



Entwicklerhandbuch

Amazon Personalize



Amazon Personalize: Entwicklerhandbuch

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Personalize?	1
Preise für Amazon Personalize	2
Anleitung für Erstbenutzer	3
Entdecken von Amazon Personalize mit dem Magic Movie Machine	3
Navigieren in diesem Handbuch	3
Verwandte AWS Services und Lösungen	4
Services von Drittanbietern	5
Weitere Informationen	6
Amazon Personalize und generative KI	7
Empfehlungen mit Themen aus dem Content Generator	7
Metadaten für Empfehlungen	8
Vorkonfigurierter Code LangChain für die Personalisierung	9
Funktionsweise	11
Zusammenfassung des Amazon Personalize Personalize-Workflows	11
Nutzungsbedingungen von Amazon Personalize	13
Datenimport und -verwaltung	13
Training	15
Modellbereitstellung und Empfehlungen	17
Amazon Personalize Sie Daten	19
Daten zu Interaktionen	19
Artikeldaten	19
Benutzerdaten	20
Daten zu Aktionen	20
Aktionen, Interaktionen, Daten.	20
Einrichten von Amazon Personalize	21
Registrieren Sie sich für ein AWS-Konto	21
Erstellen eines Administratorbenutzers	22
Regionen und Endpunkte	23
Einrichten von Berechtigungen	23
Benutzern die Erlaubnis geben, auf Amazon Personalize zuzugreifen	24
Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen	26
Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren	30
Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden	36
Einrichten der AWS CLI	38

Einrichten der AWS SDKs	40
Erste Schritte	41
Voraussetzungen für den Einstieg	42
Erstellung der Trainingsdaten (Domain-Datensatzgruppe)	42
Die Trainingsdaten erstellen (Gruppe „Benutzerdefinierter Datensatz“)	44
Erste Schritte mit einer Domain-Datensatzgruppe	44
Erste Schritte mit einer Domain-Datensatzgruppe (Konsole)	45
Erste Schritte mit einer Domain-Datensatzgruppe (SDK für Java 2.x)	55
Erste Schritte mit einer Domain-Datensatzgruppe (SDK for Python (Boto3))	64
Erste Schritte mit einer Domain-Datensatzgruppe (SDK für JavaScript v3)	70
Erste Schritte mit einer benutzerdefinierten Datensatzgruppe	78
Erste Schritte (Konsole)	79
Erste Schritte (AWS CLI)	90
Erste Schritte (SDK for Python (Boto3))	101
Erste Schritte (SDK für Java 2.x)	108
Bereinigen von Ressourcen	120
Säuberung domänenbasierter Ressourcen	120
Bereinigen benutzerdefinierter Ressourcen	121
Datensätze und Schemas	123
Datensätze	124
Datensatz mit Interaktionen zwischen Elementen	125
Benutzerdatensatz	131
Elementedatensatz	132
Datensatz „Aktionen“	135
Datensatz „Aktionen“: Interaktionen	138
Schemata	139
Anforderungen an die Schemaformatierung	140
Domain-Datasets und Schemas	142
Benutzerdefinierte Datensätze und Schemas	159
Erstellen eines Schemas mit Python	174
Richtlinien für Datenformate	174
Richtlinien und Anforderungen für das Format von Massendaten	175
Beispiel für Interaktionsdaten	176
Formatieren von expliziten Eindrücken	178
Formatieren von kategorialen Daten	178
Domain-Anwendungsfälle und benutzerdefinierte Rezepte	180

Anwendungsfall- und Rezeptfunktionen	180
Personalisierung in Echtzeit	180
Exploration (Erkundung)	181
Automatische Updates	182
Auswahl eines Anwendungsfalls	184
Anwendungsfälle für VIDEO_ON_DEMAND	185
ECOMMERCE-Anwendungsfälle	189
Ein Rezept wählen	194
Amazon Personalize Sie Rezepttypen nach Anwendungsfall	194
Rezepte von Amazon Personalize	196
Verfügbare Amazon Personalize Personalize-Rezepte anzeigen	198
USER_PERSONALIZATION	199
BELIEBTE_ARTIKEL	236
PERSONALISIERTE RANGFOLGE	240
RELATED_ITEMS	247
PERSONALISIERTE_AKTIONEN	259
USER_SEGMENTIERUNG	264
Checkliste für die Bereitschaft	270
Haben Sie Ihre Anwendungsfälle mit Amazon-Personalize-Ressourcen abgeglichen?	270
Verfügen Sie über genügend Daten zur Elementinteraktion?	271
Verfügen Sie über eine Echtzeit-Event-Streaming-Architektur?	271
Sind Ihre Daten für Amazon Personalize optimiert?	272
Sammeln Sie optionale Daten, die Empfehlungen verbessern können?	272
Haben Sie einen Plan, Ihre Empfehlungen zu testen?	273
Haben Sie zusätzliche Geschäftsziele?	273
Amazon Personalize Personalize-Arbeitsablauf	274
Schritt 1: Erstellen einer Dataset-Gruppe	275
Erstellen einer Dataset-Gruppe in (Konsole)	276
Eine Datensatzgruppe erstellen (AWS CLI)	277
Erstellen einer Datensatzgruppe (AWSSDKs)	278
Schritt 2: Daten vorbereiten und importieren	280
Massendaten vorbereiten und importieren	281
Einzelne Datensätze importieren	318
Schritt 3: Erstellen von Empfehlungen oder benutzerdefinierten Ressourcen	335
Domäne erstellen	335
Benutzerdefinierte Ressourcen erstellen	369

Schritt 4: Empfehlungen einholen	426
Punktzahlen für Empfehlungen	427
Empfehlungen in Echtzeit erhalten	427
Batch-Empfehlungen und Benutzersegmente (benutzerdefinierte Ressourcen)	457
Aufrechterhaltung der Relevanz von Empfehlungen	495
Aktualisierungen von Datensätzen	495
Verwalten von Domain-Empfehlungen	495
Wartung von benutzerdefinierten Lösungen	496
Ereignisse aufzeichnen	498
Wie Ereignisse in Echtzeit Empfehlungen beeinflussen	499
Interaktionsereignisse mit Objekten aufzeichnen	499
Anforderungen für die Aufzeichnung von Interaktionsereignissen mit Objekten und das Trainieren eines Modells	500
Einen Event-Tracker für Artikelinteraktionen erstellen	501
Die PutEvents Operation verwenden	504
Ereignismetriken und Attributionsberichte	512
Aufzeichnen von Aktionsinteraktionsereignissen	514
Anforderungen für die Aufzeichnung von Ereignissen im Zusammenhang mit Aktionen	515
ID des Aktions-Interaktions-Event-Trackers	515
Die PutActionInteractions Operation wird verwendet	516
Ereignisse für anonyme Benutzer aufzeichnen	519
Aufbau eines kontinuierlichen Ereignisverlaufs für anonyme Benutzer	520
Event-Tracking-Dienste von Drittanbietern	520
Beispielimplementierungen	521
Daten verwalten	522
Aktualisieren von Daten	522
Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken	522
Das Schema eines Datensatzes ersetzen	526
Aktualisierung vorhandener Massendaten	530
Aktualisieren von Daten mit einzelnen Importvorgängen	536
Analysieren von Daten in Datensätzen	537
Erforderliche Berechtigungen für die Analyse von Daten	538
Dateneinblicke	538
Anzeigen von Erkenntnissen und Statistiken zu Datensätzen	541
Exportieren eines Datensatzes	542
Anforderungen an die Berechtigungen für den Export von Datensätzen	544

Erstellen eines Datensatz-Exportauftrags (Konsole)	545
Einen Datensatz-Exportauftrag erstellen (AWS CLI)	547
Einen Datensatz-Exportauftrag (AWSSDKs) erstellen	548
Löschen von Daten	552
Löschen eines Datensatzes (Konsole)	552
Löschen eines Datensatzes (AWS CLI)	553
Löschen eines Datensatzes (AWSSDKs)	553
Filtern von Ergebnissen	554
Filterausdrücke	555
Richtlinien und Anforderungen	557
Filterausdrucksstruktur und -elemente	558
Beispiele für Filterausdrücke	561
Filtern von Empfehlungen in Echtzeit	567
Filtern von Empfehlungen in Echtzeit (Konsole)	567
Filtern von Empfehlungen in Echtzeit (AWS CLI)	573
Filtern von Echtzeitempfehlungen (AWS SDKs)	575
Filtern von Batch-Empfehlungen und Benutzersegmenten (benutzerdefinierte Ressourcen)	581
Bereitstellen von Filterwerten in Ihrem Eingabe-JSON	582
Filtern von Batch-Workflows (Konsole)	583
Filtern von Batch-Workflows (AWS -SDKs)	583
Messung der Wirkung von Empfehlungen	585
Messung der Wirkung von Empfehlungen mit einer metrischen Zuordnung	585
Richtlinien und Anforderungen	586
Eine metrische Zuordnung erstellen	590
Verwaltung einer metrischen Zuordnung	597
Ergebnisse veröffentlichen und anzeigen	606
Wie Sie den Einfluss von Empfehlungen mit A/B-Tests messen	613
Bewährte Methoden für A/B-Tests	614
A/B-Tests mit CloudWatch Evidently-	615
Personalisierung von Suchergebnissen von OpenSearch	619
Beispiel für einen Anwendungsfall	620
Personalisierter Such-Workflow	620
So funktioniert das Amazon Personalize Search Ranking-Plugin	622
Zusätzliche Informationen	623
Richtlinien und Anforderungen	623
Anforderungen an das Plugin	623

Amazon OpenSearch Service-Berechtigungen einrichten	624
Open-Source-Berechtigungen einrichten OpenSearch	629
Einrichtung OpenSearch und Installation des Plug-ins	631
Amazon OpenSearch Service einrichten	632
Open Source einrichten OpenSearch	633
Konfiguration des Plug-ins	635
Felder für den Antwortprozessor <code>personalized_search_ranking</code>	636
Eine Pipeline mit Amazon OpenSearch Service erstellen	637
Eine Pipeline mit Open Source erstellen OpenSearch	638
Anwenden des Plugins auf OpenSearch Abfragen	639
Anwenden des Plug-ins auf Amazon OpenSearch Service-Abfragen	639
Anwenden des Plug-ins auf Abfragen in Open Source OpenSearch	641
OpenSearch Ergebnisse mit Ergebnissen aus dem Plugin vergleichen	642
Ergebnisse mit Amazon OpenSearch Service vergleichen	643
Vergleich der Ergebnisse mit Open Source OpenSearch	645
Überwachung des Plugins	646
Überwachung des Plugins mit Amazon OpenSearch Service	646
Überwachung des Plugins mit Open Source OpenSearch	647
Beispiel für Pipeline-Kennzahlen	647
Markieren von Ressourcen	649
Richtlinien und Anforderungen	650
Zusätzliche Informationen	650
Hinzufügen von Tags zu Amazon-Personalize-Ressourcen	651
Hinzufügen von Tags (Konsole)	651
Hinzufügen von Tags (AWS CLI)	652
Hinzufügen von Tags (AWS SDKs)	653
Entfernen von Tags aus Amazon-Personalize-Ressourcen	656
Entfernen von Tags (Konsole)	656
Entfernen von Tags (AWS CLI)	657
Entfernen von Tags (AWS SDKs)	657
Verwenden von Tags in IAM-Richtlinien	658
Fehlerbehebung	660
Häufig gestellte Fragen	660
Datenimport und -verwaltung	660
Eine benutzerdefinierte Lösung und Lösungsversion erstellen	662
Modellbereitstellung (benutzerdefinierte Kampagnen)	662

Empfehlungen	663
Empfehlungen zum Filtern	664
Fehlermeldungen	664
Datenimport und -verwaltung	665
Eine Lösung und eine Lösungsversion erstellen (benutzerdefinierte Ressourcen)	666
Modellbereitstellung (benutzerdefinierte Kampagnen)	667
Empfehlungsgeber (Domain-Datensatzgruppen)	667
Empfehlungen	667
Empfehlungen zum Filtern	668
Angaben von Ressourcen mitAWS CloudFormation	669
Amazon Personalize undAWS CloudFormationVorlagen	669
BeispielAWS CloudFormation-Vorlagen für Amazon Personalize Resources	670
CreateDatasetGruppe	670
CreateDataset	671
CreateSchema	672
CreateSolution	673
Weitere Informationen zu AWS CloudFormation	673
Sicherheit	675
Datenschutz	676
Datenverschlüsselung	677
Identitäts- und Zugriffsverwaltung	678
Zielgruppe	678
Authentifizierung mit Identitäten	679
Verwalten des Zugriffs mit Richtlinien	683
So funktioniert Amazon Personalize mit IAM	686
Serviceübergreifende Confused-Deputy-Prävention	694
Beispiele für identitätsbasierte Richtlinien	695
Fehlerbehebung	701
Protokollierung und Überwachung	703
Überwachen	704
CloudWatch Metriken für Amazon Personalize	708
Protokollieren Amazon Personalize Personalize-API-Aufrufen mit AWS CloudTrail	713
Compliance-Validierung	715
Ausfallsicherheit	716
Sicherheit der Infrastruktur	717
VPC-Endpunkte (AWS PrivateLink)	717

Überlegungen zu Amazon Personalize VPC-Endpunkten	718
Erstellen eines VPC-Schnittstellen-Endpunkts für Amazon Personalize	718
Erstellen einer VPC-Endpunktrichtlinie für Amazon Personalize	719
Endpunkte und Kontingente	721
Amazon Personalize Endpunkte und Regionen	721
-Compliance	721
Service Quotas	721
Beantragen einer Kontingenterhöhung	729
API-Referenz	731
Aktionen	731
Amazon Personalize	734
Amazon Personalize Events	965
Amazon Personalize Runtime	980
Datentypen	998
Amazon Personalize	1001
Amazon Personalize Events	1152
Amazon Personalize Runtime	1168
Häufige Fehler	1173
Geläufige Parameter	1175
Dokumentverlauf	1179
AWS-Glossar	1197
.....	mcxcviii

Was ist Amazon Personalize?

Amazon Personalize ist ein vollständig verwalteter Machine-Learning-Service, der Ihre Daten verwendet, um Elementempfehlungen für Ihre Benutzer zu generieren. Es kann auch Benutzersegmente generieren, die auf der Affinität der Benutzer für bestimmte Elemente oder Elementmetadaten basieren.

Zu den häufigsten Anwendungsfällen gehören die folgenden:

- Anpassen einer Video-Streaming-App – Sie können vorkonfigurierte oder anpassbare Amazon-Personalize-Ressourcen verwenden, um Ihrer Streaming-App mehrere Arten personalisierter Videoempfehlungen hinzuzufügen. Zum Beispiel Top-Auswahlen für Sie, Mehr wie X und beliebtesten Videoempfehlungen.
- Hinzufügen von Produktempfehlungen zu einer E-Commerce-App – Sie können vorkonfigurierte oder anpassbare Amazon-Personalize-Ressourcen verwenden, um Ihrer Einzelhandels-App mehrere Arten personalisierter Produktempfehlungen hinzuzufügen. Zum Beispiel Für Sie empfohlen, Häufig zusammen gekauft und Kunden, die X angesehen haben, haben sich auch Produktempfehlungen angesehen.
- Hinzufügen von Empfehlungen zu bewährten Aktionen in Echtzeit zu Ihrer App – Sie können anpassbare Amazon-Personalize-Ressourcen verwenden, um die Aktionen zu empfehlen, die Ihre Benutzer höchstwahrscheinlich basierend auf ihrem Verhalten ausführen werden. Sie können beispielsweise Empfehlungen in Echtzeit hinzuzufügen, um sich für Ihr Treueprogramm anzumelden, Ihre mobile App herunterzuladen oder sich für Werbe-E-Mails anzumelden.
- Erstellen personalisierter E-Mails – Sie können anpassbare Amazon-Personalize-Ressourcen verwenden, um Batch-Empfehlungen für alle Benutzer in einer E-Mail-Liste zu generieren. Anschließend können Sie einen [AWS -Service](#) oder [einen Drittanbieterservice](#) verwenden, um Benutzern personalisierte E-Mails zu senden, die Elemente in Ihrem Katalog empfehlen.
- Erstellen einer gezielten Marketingkampagne – Sie können Amazon Personalize verwenden, um Segmente von Benutzern zu generieren, die höchstwahrscheinlich mit Elementen in Ihrem Katalog interagieren werden. Anschließend können Sie einen [AWS -Service](#) oder [einen Drittanbieterservice](#) verwenden, um eine gezielte Marketingkampagne zu erstellen, die verschiedene Elemente zu verschiedenen Benutzersegmenten hochstuft.
- Personalisieren von Suchergebnissen – Sie können anpassbare Amazon-Personalize-Ressourcen verwenden, um Suchergebnisse für Ihre Benutzer zu personalisieren. Amazon Personalize kann beispielsweise Suchergebnisse, die Sie mit generieren, neu sortieren [OpenSearch](#).

Für die meisten Anwendungsfälle generiert Amazon Personalize Empfehlungen, die hauptsächlich auf Elementinteraktionsdaten basieren. Die Daten zur Elementinteraktion stammen von Ihren Benutzern, die mit Elementen in Ihrem Katalog interagieren. Zum Beispiel Benutzer, die auf verschiedene Elemente klicken. Ihre Elementinteraktionsdaten können sowohl aus Ihren historischen Masseninteraktionsdatensätzen in einer CSV-Datei als auch aus Echtzeitereignissen Ihrer Benutzer stammen, während sie mit Ihrem Katalog interagieren. In einigen Fällen verwendet Amazon Personalize auch Daten von Elementen und Benutzern wie Genre, Preis oder Geschlecht. Und für Szenarien mit den nächsten bewährten Aktionen werden Aktionen und Aktionsinteraktionsdaten verwendet.

Wenn Sie Massendaten importieren, können Sie Amazon SageMaker Data Wrangler verwenden, um Daten aus mehr als 40 Quellen zu importieren und für Amazon Personalize vorzubereiten. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

Amazon Personalize umfasst API-Operationen für die Echtzeit-Personalisierung und Batch-Operationen für Massenempfehlungen und Benutzersegmente. Sie können schnell mit für Anwendungsfälle optimierten Empfehlungen für Ihre Geschäftsdomäne beginnen oder Ihre eigenen konfigurierbaren benutzerdefinierten Ressourcen erstellen.

Themen

- [Preise für Amazon Personalize](#)
- [Anleitung für Erstbenutzer von Amazon Personalize](#)
- [Verwandte AWS Services und Lösungen](#)
- [Services von Drittanbietern](#)
- [Weitere Informationen](#)

Preise für Amazon Personalize

Bei Amazon Personalize fallen keine Mindestgebühren und keine Vorabverpflichtungen an. Das [AWS kostenlose Kontingent](#) von bietet ein monatliches Kontingent von bis zu 20 GB Datenverarbeitung pro verfügbarer AWS Region, bis zu 100 Stunden Trainingszeit pro AWS berechtigter Region und bis zu 180.000 Empfehlungsanforderungen. Das kostenlose Kontingent ist für die ersten zwei Monate der Nutzung gültig.

Eine vollständige Liste der Gebühren und Preise finden Sie unter [Amazon Personalize – Preise](#).

Anleitung für Erstbenutzer von Amazon Personalize

Wenn Sie Amazon Personalize zum ersten Mal verwenden, können Ihnen die folgenden Ressourcen den Einstieg erleichtern.

Themen

- [Entdecken von Amazon Personalize mit dem Magic Movie Machine](#)
- [Navigieren in diesem Handbuch](#)

Entdecken von Amazon Personalize mit dem Magic Movie Machine

Der Magic Movie Machine ist ein interaktives Lernerlebnis. Es hilft Ihnen dabei, Amazon-Personalize-Funktionen zu entdecken und mehr über die Generierung von Empfehlungen zu erfahren. Eine kurze Einführung finden Sie im folgenden Video. Testen Sie dann die [Magic Movie Machine](#).

[Erste Schritte mit Amazon Personalize](#)

Navigieren in diesem Handbuch

Wir empfehlen Ihnen, der Reihe nach die folgenden Abschnitte zu lesen:

1. [Funktionsweise](#) – In diesem Abschnitt wird der Amazon-Personalize-Workflow vorgestellt und Sie werden durch die Schritte geführt, um personalisierte Erlebnisse für Ihre Benutzer zu schaffen. Dieser Abschnitt enthält auch allgemeine Amazon-Personalize-Begriffe und ihre Definitionen. Beginnen Sie mit diesem Abschnitt, um sicherzustellen, dass Sie die Workflows und Begriffe von Amazon Personalize gut verstehen, bevor Sie Empfehlungen erhalten.
2. [Einrichten von Amazon Personalize](#) – In diesem Abschnitt richten Sie Ihr ein AWS-Konto, richten die erforderlichen Berechtigungen für die Verwendung von Amazon Personalize ein und richten die AWS CLI und die AWS SDKs für die Verwendung und Verwaltung von Amazon Personalize ein.
3. [Erste Schritte](#) – In diesem Abschnitt beginnen Sie mit der Verwendung von Amazon Personalize mit einem einfachen Filmdatensatz. Schließen Sie diese Tutorials ab, um praktische Erfahrungen mit Amazon Personalize zu sammeln. Sie können entweder mit einer Domain-Datensatzgruppe oder einer benutzerdefinierten Datensatzgruppe beginnen:
 - Um mit der Erstellung einer Domain-Datensatzgruppe zu beginnen, schließen Sie die ab [Voraussetzungen für den Einstieg](#) und starten Sie dann die Tutorials in [Erste Schritte mit einer Domain-Datensatzgruppe](#).

- Um mit einer benutzerdefinierten Datensatzgruppe zu beginnen, schließen Sie die ab [Voraussetzungen für den Einstieg](#) und starten Sie dann die Tutorials in [Erste Schritte mit einer Domain-Datensatzgruppe](#).
4. [Domain-Anwendungsfälle und benutzerdefinierte Rezepte](#) – Erfahren Sie mehr über die Domänenanwendungsfälle und benutzerdefinierten Rezepte, mit denen Sie ein Modell in Amazon Personalize trainieren können. Verwenden Sie diese Informationen, um Ihren Anwendungsfall mit Ressourcen in Amazon Personalize abzugleichen.
 5. [Checkliste für die Bereitschaft](#) – Überprüfen Sie die Checkliste für die Bereitschaft, um mit der Verwendung von Amazon Personalize mit Ihren eigenen Daten zu beginnen. Diese Checkliste enthält Listen von Amazon-Personalize-Funktionen, -Anforderungen und -Datenleitfäden. Es kann Ihnen bei der Planung helfen oder Sie können es als Referenz verwenden, wenn Sie Ressourcen in Amazon Personalize erstellen.
 6. [Amazon Personalize Personalize-Arbeitsablauf](#) – Dieser Abschnitt führt Sie durch den gesamten Amazon-Personalize-Workflow. Es enthält step-by-step Anweisungen zum Erstellen einer Domain-Datensatzgruppe oder einer benutzerdefinierten Datensatzgruppe, zum Vorbereiten und Importieren von Daten, zum Erstellen von Empfehlungsgebern oder benutzerdefinierten Ressourcen und zum Abrufen von Empfehlungen.
 7. [Ereignisse aufzeichnen](#) – In diesem Abschnitt wird beschrieben, wie Sie Ereignisse zur Elementinteraktion und zur Aktionsinteraktion in Echtzeit aufzeichnen. Nachdem Sie Ihre Amazon-Personalize-Ressourcen eingerichtet haben, füllen Sie diesen Abschnitt aus, um zu erfahren, wie Sie Ihre Datensätze über das Verhalten Ihrer Benutzer auf dem Laufenden halten.
 8. [Filtern von Empfehlungen und Benutzersegmenten](#) – In diesem Abschnitt wird beschrieben, wie Sie Empfehlungen filtern. Füllen Sie diesen Abschnitt aus, um zu erfahren, wie Sie Filterausdrücke erstellen, um Empfehlungen basierend auf benutzerdefinierten Kriterien zu filtern. Beispielsweise möchten Sie vielleicht keine Produkte empfehlen, die ein Benutzer bereits gekauft hat, oder keine Filme empfehlen, die ein Benutzer bereits gesehen hat.

Verwandte AWS Services und Lösungen

Amazon Personalize lässt sich nahtlos in andere - AWS Services und -Lösungen integrieren. Beispielsweise ist Folgendes möglich:

- Verwenden Sie Amazon SageMaker Data Wrangler (Data Wrangler), um Daten aus mehr als 40 Quellen in einen Amazon-Personalize-Datensatz zu importieren. Data Wrangler ist eine Funktion von Amazon SageMaker Studio, die eine end-to-end Lösung zum Importieren, Vorbereiten,

Transformieren und Analysieren von Daten bietet. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

- Verwenden Sie AWS Amplify , um Ereignisse zur Elementinteraktion aufzuzeichnen. Amplify enthält eine JavaScript Bibliothek zum Aufzeichnen von Ereignissen aus Webclient-Anwendungen. Und sie enthält eine Bibliothek zum Aufzeichnen von Ereignissen im Servercode. Weitere Informationen finden Sie in der [Amplify-Dokumentation](#).
- Automatisieren und planen Sie Amazon-Personalize-Aufgaben mit [Aufrechterhaltung persönlicher Erlebnisse mit Machine Learning](#). Diese AWS Lösungsimplementierung automatisiert den Amazon-Personalize-Workflow, einschließlich Datenimport, Training von Lösungsversionen und Batch-Workflows.
- Verwenden Sie Amazon CloudWatch Evidently, um A/B-Tests mit Amazon-Personalize-Empfehlungen durchzuführen. Weitere Informationen finden Sie unter [A/B-Tests mit CloudWatch Evidently](#).
- Verwenden Sie Amazon Pinpoint, um gezielte Marketingkampagnen zu erstellen. Ein Beispiel, das zeigt, wie Sie Amazon Pinpoint und Amplify verwenden, um Amazon-Personalize-Empfehlungen zu einer Marketing-E-Mail-Kampagne und einer Webanwendung hinzuzufügen, finden Sie unter [Web Analytics mit Amplify](#).

Services von Drittanbietern

Amazon Personalize funktioniert gut mit verschiedenen Services von Drittanbietern.

- Amplitude – Sie können Amplitude verwenden, um Benutzeraktionen zu verfolgen, die Ihnen helfen, das Verhalten Ihrer Benutzer zu verstehen. Informationen zur Verwendung von Amplitude und Amazon Personalize finden Sie im folgenden Blogbeitrag des AWS Partnernetzwerks (APN): [Messung der Effektivität der Personalisierung mit Amplitude und Amazon Personalize](#).
- geschweift – Sie können geschweift verwenden, um Benutzern personalisierte E-Mails zu senden, in denen Artikel in Ihrem Katalog empfohlen werden. Tarze ist eine marktführerische Messaging-Plattform (E-Mail, Push, SMS). Einen Workshop, der zeigt, wie Amazon Personalize und Brasilien integriert werden, finden Sie unter [Amazon-Personalize-Workshop](#).
- mParticle – Sie können mParticle verwenden, um Ereignisdaten aus Ihrer App zu sammeln. Ein Beispiel, das zeigt, wie Sie mParticle und Amazon Personalize verwenden, um personalisierte Produktempfehlungen zu implementieren, finden Sie unter [Wie man die Leistungsfähigkeit eines CDP für Machine Learning nutzt: Teil 2](#).

- **Optimiert** – Sie können Optimizely verwenden, um A/B-Tests mit Amazon-Personalize-Empfehlungen durchzuführen. Informationen zur Verwendung von Optimizely und Amazon Personalize finden Sie unter [Optimieren der Integration von Amazon Personalize, um leistungsstarkes Machine Learning mit Experimenten zu kombinieren](#).
- **Segment** – Sie können Segment verwenden, um Ihre Daten an Amazon Personalize zu senden. Weitere Informationen zur Integration von Segment in Amazon Personalize finden Sie unter [Amazon-Personalize-Ziel](#).

Eine vollständige Liste der Partner finden Sie unter [Amazon-Personalize-Partner](#).

Weitere Informationen

Die folgenden Ressourcen enthalten zusätzliche Informationen zu Amazon Personalize:

- Eine kurze Referenz, anhand derer Sie feststellen können, ob Amazon Personalize zu Ihrem Anwendungsfall passt, finden Sie im [Amazon Personalize C Spickblatt](#) im [Amazon Personalize-Beispiel](#)-Repository.
- Eine Reihe von Videos zur Verwendung von Amazon Personalize finden Sie in der [Amazon Personalize Deep Dive Video Serie](#) auf YouTube.
- Ausführliche Tutorials und Codebeispiele finden Sie im [amazon-personalize-samples GitHub Repository](#).

Amazon Personalize und generative KI

Amazon Personalize funktioniert gut mit generativer künstlicher Intelligenz (generative KI). Amazon Personalize Content Generator kann mithilfe generativer KI den Batch-Empfehlungen für verwandte Artikel ansprechende Themen hinzufügen. Content Generator ist eine generative KI-Funktion, die von Amazon Personalize verwaltet wird.

Sie können auch die Amazon Personalize-Empfehlungen verwenden, um Amazon Personalize in Ihren generativen KI-Workflow zu integrieren und die Benutzererfahrung zu verbessern. Sie können beispielsweise Empfehlungen zu generativen KI-Eingabeaufforderungen hinzufügen, um Marketinginhalte zu erstellen, die auf die Interessen der einzelnen Benutzer zugeschnitten sind. Sie können auch kurze Zusammenfassungen für empfohlene Inhalte erstellen oder Produkte oder Inhalte über Chat-Bots empfehlen.

Das folgende Video zeigt, wie Sie Empfehlungen mit Amazon Personalize und generativer KI verbessern können.

[Verbessern Sie Ihre Empfehlungen mit Amazon Personalize und Generative KI](#)

Die folgenden Funktionen von Amazon Personalize verwenden generative KI oder können Ihnen helfen, generative KI-Lösungen zu entwickeln, die personalisierte Inhalte erstellen.

Themen

- [Empfehlungen mit Themen aus dem Content Generator](#)
- [Metadaten für Empfehlungen](#)
- [Vorkonfigurierter Code LangChain für die Personalisierung](#)

Empfehlungen mit Themen aus dem Content Generator

Amazon Personalize Content Generator kann den Batch-Empfehlungen beschreibende Themen hinzufügen. Content Generator ist eine generative KI-Funktion, die von Amazon Personalize verwaltet wird.

Wenn Sie Stapelempfehlungen mit Themen erhalten, fügt Amazon Personalize Content Generator für jeden Satz ähnlicher Artikel ein beschreibendes Thema hinzu. Wenn Sie beispielsweise Empfehlungen für ähnliche Artikel für ein Frühstück erhalten, generiert Amazon Personalize möglicherweise ein Thema wie Rise and Shine oder Morning Essentials. Sie könnten das Thema

verwenden, um einen generischen Karusselltitel wie „Häufig zusammen gekauft“ zu ersetzen. Oder Sie könnten das Thema in eine Werbe-E-Mail oder eine Marketingkampagne für neue Menüoptionen integrieren.

Um Designs zu generieren, importieren Sie Daten in Artikelinteraktionen und Artikeldatensätze, erstellen eine benutzerdefinierte Lösung mit dem Rezept „Ähnliche Artikel“ und generieren Chargenempfehlungen. Ihre Artikeldaten müssen Artikelbeschreibung und Titelinformationen enthalten. Detaillierte Artikelbeschreibungen und Titel helfen Content Generator dabei, genauere und ansprechendere Themen zu erstellen.

- Informationen zum Amazon Personalize finden Sie unter [Amazon Personalize Personalize-Arbeitsablauf](#).
- Informationen zu Batch-Empfehlungen finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).
- Informationen zum Generieren von Empfehlungen mit Themen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Metadaten für Empfehlungen

Wenn Sie Empfehlungen erhalten, können Sie Amazon Personalize veranlassen, Metadaten zu jedem empfohlenen Artikel aus Ihrem Artikeldatensatz zurückzugeben. Sie können diese Metadaten zusammen mit den Empfehlungen von Amazon Personalize zu Ihren generativen KI-Eingabeaufforderungen hinzufügen, um überzeugendere Inhalte zu generieren.

Beispielsweise könnten Sie generative KI verwenden, um Marketing-E-Mails zu erstellen. Sie können Amazon Personalize und ihre Metadaten, z. B. Filmgenres, als Teil von Prompt Engineering für generative KI verwenden. Mit personalisierten Eingabeaufforderungen können Sie generative KI verwenden, um ansprechende Marketing-E-Mails zu erstellen, die auf die jeweiligen Interessen Ihrer Kunden zugeschnitten sind.

Um Empfehlungsmetadaten zu erhalten, schließen Sie zunächst den Amazon Personalize ab, um Daten zu importieren und Domain- oder benutzerdefinierte Ressourcen zu erstellen. Wenn Sie eine Amazon Personalize Personalize-Empfehlung oder eine Kampagne erstellen, aktivieren Sie die Option, Metadaten in Empfehlungen aufzunehmen. Wenn Sie Empfehlungen erhalten, können Sie angeben, welche Spalten mit Artikeldaten Sie einbeziehen möchten.

- Informationen zum Amazon Personalize finden Sie unter [Amazon Personalize Personalize-Arbeitsablauf](#).

- Informationen zur Aktivierung von Metadaten für einen Empfehlungsgeber finden Sie unter [Aktivieren von Metadaten in Empfehlungen \(Domain-Ressourcen\)](#).
- Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Aktivieren von Metadaten in Empfehlungen \(benutzerdefinierte Ressourcen\)](#).
- Weitere Informationen darüber, wie Sie Amazon Personalize mit generativer KI zur Erstellung von Marketingkampagnen verwenden können, finden Sie unter Verbessern [Sie Ihre Marketinglösungen mit Amazon Personalize und](#) generativer KI.

Vorkonfigurierter Code LangChain für die Personalisierung

LangChain ist ein Framework für die Entwicklung von Anwendungen, die auf Sprachmodellen basieren. Es enthält Code, der für Amazon Personalize entwickelt wurde. Sie können diesen Code verwenden, um Amazon Personalize in Ihre generative KI-Lösung zu integrieren.

Sie können beispielsweise den folgenden Code verwenden, um Amazon Personalize für einen Benutzer zu Ihrer Kette hinzuzufügen.

```
from aws_langchain import AmazonPersonalize
from aws_langchain import AmazonPersonalizeChain
from langchain.llms.bedrock import Bedrock

recommender_arn="RECOMMENDER ARN"

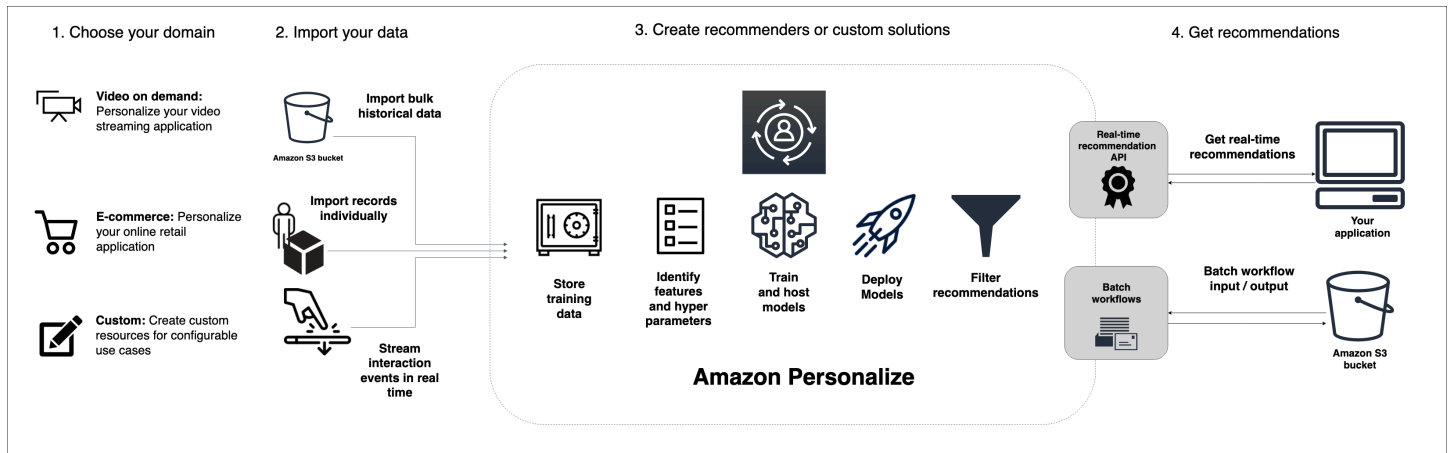
bedrock_llm = Bedrock(model_id="anthropic.claude-v2", region_name="us-west-2")
client=AmazonPersonalize(credentials_profile_name="default",region_name="us-west-2",recommender_arn=recommender_arn)
# Create personalize chain
# Use return_direct=True if you do not want summary
chain = AmazonPersonalizeChain.from_llm(
    llm=bedrock_llm,
    client=client,
    return_direct=False
)
response = chain({'user_id': '1'})
print(response)
```

- Informationen zu den ersten Schritten mit finden Sie LangChain unter [Erste Schritte](#).

- Informationen zur Verwendung von für Amazon Personalize erstelltem LangChain Code, einschließlich komplexerer Codebeispiele, finden Sie unter [Amazon Personalize LangChain Personalize-Erweiterungen](#) im [AWS Beispiel-Repository](#).

Funktionsweise

Amazon Personalize verwendet Ihre Daten, um domänenbasierte oder anpassbare Empfehlungsmodelle zu trainieren. Sie verwenden in Ihrer Anwendung eine private Empfehlungs-API, um Empfehlungen in Echtzeit anzufordern. Amazon Personalize unterstützt auch Batch-Workflows zum Abrufen von Artikelempfehlungen und Benutzersegmenten.



Themen

- [Zusammenfassung des Amazon Personalize Personalize-Workflows](#)
- [Nutzungsbedingungen von Amazon Personalize](#)
- [Datentypen, die Amazon Personalize verwenden kann](#)

Zusammenfassung des Amazon Personalize Personalize-Workflows

Der Amazon Personalize Personalize-Workflow sieht wie folgt aus:

1. [Erstellen Sie eine Datensatzgruppe](#)

Eine Datensatzgruppe ist ein Container für Amazon Personalize Personalize-Ressourcen. Die Art der Datensatzgruppe, die Sie erstellen, bestimmt die Ressourcen, die Sie in Schritt 3 des Amazon Personalize erstellen können.

- Mit einer Domain-Datensatzgruppe können Sie Empfehlungen für Anwendungsfälle von VIDEO_ON_DEMAND- oder E-COMMERCE-Domains erstellen. Amazon Personalize verwaltet die Konfiguration, Schulung und Aktualisierung dieser Empfehlungen. Wenn Sie mit einer

Domain-Datensatzgruppe beginnen, können Sie trotzdem benutzerdefinierte Ressourcen hinzufügen.

- Mit einer benutzerdefinierten Datensatzgruppe können Sie nur benutzerdefinierte Ressourcen erstellen. Dazu gehören Lösungen, Lösungsversionen und Kampagnen. Bei diesen Ressourcen haben Sie mehr Kontrolle über Konfigurationen, Updates und Umschulungen.

2. [Daten vorbereiten und importieren](#)

Sie importieren Interaktions-, Artikel-, Benutzer-, Aktions- und Aktionsinteraktionsdatensätze in Datensätze (Amazon Personalize Container für Daten). Sie können Datensätze in großen Mengen oder einzeln importieren. Wenn Sie Massendaten importieren, können Sie Amazon SageMaker Data Wrangler verwenden, um Daten aus mehr als 40 Quellen zu importieren und für Amazon Personalize vorzubereiten. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

3. [Erstellen Sie Domain-Empfehlungen oder benutzerdefinierte Ressourcen](#)

Nachdem Sie Ihre Daten importiert haben, erstellen Sie Domain-Empfehlungen (für Domain-Datensatzgruppen) oder benutzerdefinierte Ressourcen (für benutzerdefinierte Datensatzgruppen), um ein Modell anhand Ihrer Daten zu trainieren. Sie verwenden diese Ressourcen, um Empfehlungen zu generieren.

4. [Holen Sie sich Empfehlungen](#)

Verwenden Sie Ihren Empfehlungsdienst oder Ihre benutzerdefinierte Kampagne, um Empfehlungen zu erhalten. Mit einer benutzerdefinierten Datensatzgruppe können Sie auch Batch-Empfehlungen oder Benutzersegmente abrufen.

Nachdem Sie den Amazon Personalize-Workflow zum ersten Mal abgeschlossen haben, sollten Sie Ihre Daten auf dem neuesten Stand halten und alle benutzerdefinierten Lösungen regelmäßig neu trainieren. Auf diese Weise kann Ihr Modell aus den jüngsten Aktivitäten Ihres Benutzers lernen und die Relevanz von Empfehlungen aufrechterhalten und verbessern. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Nutzungsbedingungen von Amazon Personalize

In diesem Abschnitt werden die in Amazon Personalize verwendeten Begriffe vorgestellt.

Themen

- [Datenimport und -verwaltung](#)
- [Training](#)
- [Modellbereitstellung und Empfehlungen](#)

Datenimport und -verwaltung

Die folgenden Begriffe beziehen sich auf das Importieren, Exportieren und Formatieren von Daten in Amazon Personalize.

Kontextmetadaten

Interaktionsdaten, die Sie über den Browserkontext eines Benutzers (z. B. verwendetes Gerät oder Standort) sammeln, wenn ein Ereignis (z. B. ein Klick) eintritt. Kontextabhängige Metadaten können die Relevanz von Empfehlungen für neue und bestehende Benutzer verbessern.

dataset

Ein Container für Daten, die Sie auf Amazon Personalize hochladen. Es gibt drei Arten von Amazon-Personalize-Datensätzen: Benutzer, Elemente und Interaktionen.

Datensatzgruppe

Ein Container für Amazon Personalize Personalize-Ressourcen, einschließlich Datensätzen, Domain-Empfehlungen und benutzerdefinierten Ressourcen. Eine Datensatzgruppe organisiert Ihre Ressourcen in unabhängigen Sammlungen, wobei Ressourcen aus einer Datensatzgruppe keine Ressourcen in einer anderen Datensatzgruppe beeinflussen können. Eine Datensatzgruppe kann entweder eine Domain-Datensatzgruppe oder eine benutzerdefinierte Datensatzgruppe sein.

Domain-Datensatzgruppe

Eine Datensatzgruppe mit vorkonfigurierten Ressourcen für verschiedene Geschäftsdomänen und Anwendungsfälle. Amazon Personalize verwaltet den Lebenszyklus von Schulungsmodellen und deren Bereitstellung. Wenn Sie eine Domain-Datensatzgruppe erstellen, wählen Sie Ihre Geschäftsdomain aus, importieren Ihre Daten und erstellen Empfehlungen für jeden Ihrer Anwendungsfälle. Sie verwenden Ihren Empfehlungsdienst in Ihrer Anwendung, um Empfehlungen für den GetRecommendations Vorgang zu erhalten.

Wenn Sie mit einer Domain-Datensatzgruppe beginnen, können Sie trotzdem benutzerdefinierte Ressourcen wie Lösungen und Lösungsversionen hinzufügen, die mit Rezepten für benutzerdefinierte Anwendungsfälle trainiert wurden.

Benutzerdefinierte Datensatzgruppe

Eine Datensatzgruppe, die nur benutzerdefinierte Ressourcen enthält, darunter Lösungen, Lösungsversionen, Filter, Kampagnen und Batch-Inferenzjobs. Sie verwenden eine Kampagne, um Empfehlungen für den GetRecommendations Vorgang zu erhalten. Sie verwalten den Lebenszyklus von Schulungsmodellen und deren Einsatz. Wenn Sie mit einer benutzerdefinierten Datensatzgruppe beginnen, können Sie sie später keiner Domäne zuordnen. Erstellen Sie stattdessen eine neue Domain-Datensatzgruppe.

Auftrag zum Exportieren von Datensätzen

Ein Tool zum Exportieren von Datensätzen, das die Datensätze in einem Datensatz in eine oder mehrere CSV-Dateien in einem Amazon S3 S3-Bucket ausgibt. Die CSV-Ausgabedatei enthält eine Kopfzeile mit Spaltennamen, die den Feldern im Schema des Datensatzes entsprechen.

Job zum Importieren von Datensätzen

Ein Massenimport-Tool, das Ihren Amazon Personalize-Datensatz mit Daten aus einer CSV-Datei in Ihrem Amazon S3 S3-Bucket füllt.

event

Eine Benutzeraktion — wie ein Klick, ein Kauf oder eine Videowiedergabe —, die Sie aufzeichnen und in einen Datensatz mit Amazon Personalize Personalize-Artikelinteraktionen hochladen. Sie importieren Ereignisse in großen Mengen aus einer CSV-Datei, inkrementell mit der Amazon Personalize Personalize-Konsole und in Echtzeit.

Explizite Impressionen

Eine Liste von Artikeln, die Sie manuell zu einem Amazon Personalize-Datensatz mit Artikelinteraktionen hinzufügen. Im Gegensatz zu impliziten Impressionen, die Amazon Personalize automatisch aus Ihren Empfehlungsdaten ableitet, entscheiden Sie, was in explizite Impressionen enthalten sein soll.

Implizite Impressionen

Die Empfehlungen, die Ihre Anwendung einem Benutzer zeigt. Im Gegensatz zu expliziten Impressionen, die Sie manuell zu einem Datensatz mit Artikelinteraktionen hinzufügen, leitet Amazon Personalize automatisch implizite Impressionen aus Ihren Empfehlungsdaten ab.

Impressionsdaten

Die Liste der Artikel, die Sie einem Benutzer präsentiert haben, als dieser mit einem bestimmten Artikel interagiert hat, indem er darauf geklickt, ihn angesehen, gekauft usw. hat. Amazon Personalize verwendet Impressionsdaten, um die Relevanz neuer Elemente für einen Benutzer zu berechnen, basierend darauf, wie häufig Benutzer das gleiche Element ausgewählt oder ignoriert haben.

Datensatz „Interaktionen“

Ein Container für historische Daten und Echtzeitdaten, die Sie aus Interaktionen zwischen Benutzern und Elementen (sogenannten [Ereignissen](#)) sammeln. Interaktionsdaten können [Impressionsdaten](#) und [kontextuelle Metadaten](#) beinhalten.

Artikeldatensatz

Ein Container für Metadaten zu Ihren Artikeln, z. B. Preis, Genre oder Verfügbarkeit.

schema

Ein JSON-Objekt im [Apache Avro-Format](#), das Amazon Personalize über die Struktur Ihrer Daten informiert. Amazon Personalize verwendet Ihr Schema, um Ihre Daten zu analysieren.

Benutzerdatensatz

Ein Container für Metadaten über Ihre Nutzer, z. B. Alter, Geschlecht oder Treuemitgliedschaft.

Training

Die folgenden Begriffe beziehen sich auf das Trainieren eines Modells in Amazon Personalize.

item-to-item Rezept für Ähnlichkeiten (SIMS)

Ein [RELATED_ITEMS-Rezept](#), das die Daten aus einem Interactions-Datensatz verwendet, um Empfehlungen für Elemente abzugeben, die einem bestimmten Element ähnlich sind. Das SIMS-Rezept berechnet Ähnlichkeit auf der Grundlage der Art und Weise, wie Benutzer mit Artikeln interagieren, anstatt anhand von Artikelmetadaten wie Preis oder Farbe abzugleichen.

Artikelaffinität

Ein `USER_SEGMENTATION`-Rezept, das die Daten aus einem Datensatz mit Artikelinteraktionen und einem Artikeldatensatz verwendet, um Benutzersegmente für jedes von Ihnen angegebene Element zu erstellen, und zwar auf der Grundlage der Wahrscheinlichkeit, dass die Benutzer mit dem Element interagieren.

item-attribute-affinity

Ein `USER_SEGMENTATION`-Rezept, das die Daten aus einem Datensatz mit Artikelinteraktionen und einem Artikeldatensatz verwendet, um für jedes von Ihnen angegebene Artikelattribut ein Benutzersegment zu erstellen, das auf der Wahrscheinlichkeit basiert, dass die Benutzer mit Elementen interagieren, die das Attribut enthalten.

Rezept personalized-ranking

Ein Rezept von [PERSONALIZED_RANKING](#), das eine Sammlung von Artikeln, die Sie bereitstellen, auf der Grundlage des prognostizierten Interesses für einen bestimmten Benutzer einordnet. Verwenden Sie das Rezept für ein personalisiertes Ranking, um die Reihenfolge der kuratierten Listen mit Elementen oder Suchergebnissen, die für einen bestimmten Benutzer personalisiert sind, zu personalisieren.

popularity-count-Rezept

Ein [USER_PERSONALIZATION-Rezept](#), das die Elemente empfiehlt, die die meisten Interaktionen mit einzelnen Benutzern hatten.

Empfehlungsgeber

Ein Tool für Domain-Datensatz-Gruppen, das Empfehlungen generiert. Sie erstellen einen Empfehlungsgeber für eine Domain-Datensatzgruppe und verwenden ihn in Ihrer Anwendung, um mithilfe der `GetRecommendations` API Empfehlungen in Echtzeit zu erhalten. Wenn Sie einen Empfehlungsgeber erstellen, geben Sie einen Anwendungsfall an und Amazon Personalize trainiert die Modelle, die den Empfehlungsgeber unterstützen, mit den besten Konfigurationen für den Anwendungsfall.

recipe

Ein Amazon Personalize Personalize-Algorithmus, der vorkonfiguriert ist, um die Artikel vorherzusagen, mit denen ein Benutzer interagieren wird (für `USER_PERSONALIZATION`-Rezepte), Artikel zu berechnen, die bestimmten Artikeln ähneln, an denen ein Benutzer Interesse gezeigt hat (für `RELATED_ITEMS` Rezepte), oder eine Sammlung von Artikeln, die Sie anbieten, auf der Grundlage des prognostizierten Interesses für einen bestimmten Benutzer zu bewerten (für `PERSONALIZED_RANKING`-Rezepte).

Lösung

Das Rezept, die benutzerdefinierten Parameter und die trainierten Modelle (Lösungsversionen), die Amazon Personalize zur Generierung von Empfehlungen verwendet.

Lösungsversion

Ein trainiertes Modell, das Sie als Teil einer Lösung in Amazon Personalize erstellen. Sie stellen eine Lösungsversion in einer Kampagne bereit, um die Personalisierungs-API zu aktivieren, mit der Sie Empfehlungen anfordern.

Trainingsweise

Der Umfang der Schulung, die bei der Erstellung einer Lösungsversion durchgeführt werden muss. Es gibt zwei verschiedene Modi: FULL und UPDATE. Der FULL-Modus erstellt eine völlig neue Lösungsversion, die auf der Gesamtheit der Trainingsdaten aus den Datensätzen in Ihrer Datensatzgruppe basiert. UPDATE aktualisiert die bestehende Lösungsversion inkrementell, um neue Elemente zu empfehlen, die Sie seit der letzten Schulung hinzugefügt haben.

Note

Mit User-Personalization oder Next-Best-Action aktualisiert Amazon Personalize automatisch die neueste Lösungsversion, die im FULL-Trainingsmodus trainiert wurde. Siehe [Automatische Updates](#).

Rezept user-personalization

Ein auf dem Hierarchical Recurrent Neural Network (HRNN) basierendes [USER_PERSONALIZATION-Rezept, das vorhersagt, mit welchen Elementen ein Benutzer interagieren wird](#). Das user-personalization-Rezept kann mithilfe von Elementerkundung und Impressionsdaten Empfehlungen für neue Elemente generieren.

Modellbereitstellung und Empfehlungen

Die folgenden Begriffe beziehen sich auf die Bereitstellung und Verwendung eines Modells in Amazon Personalize.

Batch-Inferenz-Job

Ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket importiert, Ihre Lösungsversion verwendet, um Empfehlungen zu generieren, und die Empfehlungen in einen Amazon S3 S3-Bucket exportiert. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3 S3-)

Bucket). Verwenden Sie einen Batch-Inferenzjob, um Empfehlungen für große Datensätze zu erhalten, für die keine Aktualisierungen in Echtzeit erforderlich sind.

Batch-Segmentauftrag

Ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket importiert, Ihre Lösungsversion verwendet, um Benutzersegmente zu erstellen, und die Benutzersegmente in einen Amazon S3 S3-Bucket exportiert. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3 S3-Bucket). Verwenden Sie einen Batch-Segmentauftrag mit einer Lösung, die auf einem USER_SEGMENTATION-Rezept basiert, um Benutzersegmente auf der Grundlage der Wahrscheinlichkeit zu erstellen, dass der Benutzer mit verschiedenen Elementen oder Elementen mit unterschiedlichen Artikelattributen interagiert.

Kampagne

Eine bereitgestellte Lösungsversion (trainiertes Modell) mit dedizierter Transaktionskapazität zur Erstellung von Empfehlungen für Ihre Anwendungsbutzer in Echtzeit. Nachdem Sie eine Kampagne erstellt haben, verwenden Sie die `getPersonalizedRanking` API-Operationen `getRecommendations` oder, um Empfehlungen zu erhalten.

Erkundung von Elementen

Bei der Erkundung umfassen die Empfehlungen einige Elemente oder Aktionen, bei denen die Wahrscheinlichkeit, dass sie dem Benutzer empfohlen werden, in der Regel geringer ist, z. B. neue Artikel oder Aktionen, Artikel oder Aktionen mit wenigen Interaktionen oder Artikel oder Aktionen, die für den Benutzer aufgrund seines vorherigen Verhaltens weniger relevant sind.

Empfehlungen

Eine Liste von Elementen, mit denen Amazon Personalize prognostiziert, dass ein Benutzer interagieren wird. Je nach verwendetem Amazon Personalize Personalize-Rezept können Empfehlungen entweder eine Artikelliste (USER_PERSONALIZATION-Rezepte und RELATED_ITEMS-Rezepte) oder eine Rangfolge einer von Ihnen bereitgestellten Artikelsammlung (PERSONALIZED_RANKING-Rezepte) sein.

Benutzersegmente

Listen von Benutzern, von denen Amazon Personalize prognostiziert, dass sie mit Ihrem Katalog interagieren werden. Je nach verwendetem USER_SEGMENTATION-Rezept erstellen Sie Benutzersegmente auf der Grundlage von Artikeln (Item-Affinity-Rezept), Artikelmetadaten (Item-Attribute-Affinity-Rezept). Benutzersegmente erstellen Sie mit einem Batch-Segmentauftrag.

Datentypen, die Amazon Personalize verwenden kann

In den folgenden Themen werden die verschiedenen Datentypen vorgestellt, die Sie in Amazon Personalize importieren können.

Themen

- [Daten zu Interaktionen](#)
- [Artikeldaten](#)
- [Benutzerdaten](#)
- [Daten zu Aktionen](#)
- [Aktionen, Interaktionen, Daten.](#)

Daten zu Interaktionen

Eine Interaktion ist ein Ereignis, das Sie aufzeichnen und dann als Trainingsdaten importieren. Amazon Personalize generiert Empfehlungen hauptsächlich auf der Grundlage der Interaktionsdaten. Interaktionsdaten können Folgendes beinhalten:

- Daten zu Ereignistyp und Ereigniswert
- Kontextuelle Metadaten
- Daten zu Eindrücken

Sie importieren Interaktionsdaten in einen Datensatz mit Artikelinteraktionen. Weitere Informationen zu Interaktionsdatensätzen finden Sie unter [Datensatz mit Interaktionen zwischen Elementen](#).

Artikeldaten

Zu den Artikelmetadaten, die Amazon Personalize verwenden kann, gehören:

- Numerische Daten zu jedem Artikel, z. B. dessen Preis.
- Kategorische Metadaten zu jedem Artikel, z. B. das Genre oder die Farbe des Artikels.
- Zeitstempeldaten der Erstellung für jedes Element.
- Unstrukturierte Textmetadaten, wie Produktbeschreibungen oder Filmzusammenfassungen.

Sie importieren Metadaten zu Ihren Artikeln in einen Artikeldatensatz. Weitere Informationen zu Artikel-Datensätzen finden Sie unter [Elementedatensatz](#).

Benutzerdaten

Zu den Benutzermetadaten, die Amazon Personalize verwenden kann, gehören:

- Numerische Daten über jeden Benutzer, z. B. sein Alter.
- Kategorische Metadaten zu jedem Benutzer, z. B. Geschlecht oder Mitgliedschaftsstatus.

Sie importieren Metadaten über Ihre Benutzer in einen Benutzerdatensatz. Weitere Informationen zu Benutzerdatensätzen finden Sie unter [Benutzerdatensatz](#).

Daten zu Aktionen

Zu den Aktionsdaten, die Amazon Personalize verwenden kann, gehören:

- Der geschäftliche Wert oder die Bedeutung jeder Aktion.
- Kategorische Metadaten für jede Aktion, z. B. Saisonalität oder Aktionsexklusivität.
- Zeitstempeldaten zum Ablauf der Aktion, die angeben, wann Amazon Personalize aufhören soll, jede Aktion zu empfehlen.
- Daten zur Wiederholungshäufigkeit, die angeben, wie lange Amazon Personalize warten soll, bevor jede Aktion empfohlen wird, nachdem ein Benutzer damit interagiert hat.

Sie importieren Daten über Ihre Aktionen in einen Aktionsdatensatz. Weitere Informationen zu Aktions-Datensätzen finden Sie unter [Datensatz „Aktionen“](#).

Aktionen, Interaktionen, Daten.

Zu den Daten, die Amazon Personalize aus Benutzerinteraktionen mit Aktionen verwenden kann, gehören:

- Daten zum Ereignistyp
- Kategorische Metadaten

Sie importieren Interaktionsdaten in einen Datensatz mit Aktionsinteraktionen. Weitere Informationen zu Datensätzen für Aktionsinteraktionen finden Sie unter [Datensatz „Aktionen“: Interaktionen](#).

Einrichten von Amazon Personalize

Bevor Sie Amazon Personalize verwenden, benötigen Sie ein Amazon Web Services (AWS)-Konto mit einem Administratorbenutzer. Nachdem Sie die erforderlichen Berechtigungen eingerichtet haben, können Sie über die Amazon-Personalize-Konsole, die AWS Command Line Interface (AWS CLI) oder die AWS SDKs auf Amazon Personalize zugreifen.

Themen

- [Registrieren Sie sich für ein AWS-Konto](#)
- [Erstellen eines Administratorbenutzers](#)
- [Regionen und Endpunkte](#)
- [Einrichten von Berechtigungen](#)
- [Einrichten der AWS CLI](#)
- [Einrichten der AWS SDKs](#)

Registrieren Sie sich für ein AWS-Konto

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

So registrieren Sie sich für ein AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für ein registrieren AWS-Konto, Root-Benutzer des AWS-Kontos wird ein erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem [Administratorbenutzer Administratorzugriff](#) zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff](#) erfordern.

AWS sendet Ihnen eine Bestätigungs-E-Mail, nachdem der Registrierungsprozess abgeschlossen ist. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen eines Administratorbenutzers

Nachdem Sie sich für ein registriert haben AWS-Konto, sichern Sie Ihr Root-Benutzer des AWS-Kontos, aktivieren AWS IAM Identity Center und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Ihrer Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei der [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu .

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Administratorbenutzers

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter [Konfigurieren des Benutzerzugriffs mit dem Standard IAM-Identity-Center-Verzeichnis-](#) im AWS IAM Identity Center -Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM-Identity-Center-Benutzer finden Sie unter [Anmelden beim - AWS Zugriffsportal](#) im AWS-Anmeldung -Benutzerhandbuch.

Regionen und Endpunkte

Ein Endpunkt ist eine URL, die als Eintrittspunkt für einen Webservice fungiert. Jeder Endpunkt ist mit einer bestimmten AWS -Region verknüpft. Achten Sie auf die Standardregionen der Amazon-Personalize-Konsole, der AWS CLI und der Amazon-Personalize-SDKs, da alle Amazon-Personalize-Komponenten einer bestimmten Kampagne (Datensatz, Lösung, Kampagne, Ereignis-Tracker) in derselben Region erstellt werden müssen. Die von Amazon Personalize unterstützten Regionen und Endpunkte finden Sie unter [Regionen und Endpunkte](#).

Einrichten von Berechtigungen

Sie müssen Benutzern, Gruppen oder Rollen die Berechtigung erteilen, mit Amazon-Personalize-Ressourcen zu interagieren. Und Sie müssen Amazon Personalize die Berechtigung erteilen, auf die Ressourcen zuzugreifen, die Sie in Amazon Personalize erstellen, und Aufgaben in Ihrem Namen auszuführen.

So richten Sie Berechtigungen ein

1. Erteilen Sie Ihren Benutzern, Gruppen oder Rollen die Berechtigung, mit Amazon-Personalize-Ressourcen zu interagieren und eine Rolle an Amazon Personalize zu übergeben. Siehe [Benutzern die Erlaubnis geben, auf Amazon Personalize zuzugreifen](#).
2. Erteilen Sie Amazon Personalize die Berechtigung zum Zugriff auf Ihre Ressourcen in Amazon Personalize und die Berechtigung zum Ausführen von Aufgaben in Ihrem Namen. Siehe [Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen](#).
3. Ändern Sie die Vertrauensrichtlinie Ihrer Amazon-Personalize-Servicerolle, damit das [Confused-Deputy-Problem](#) verhindert wird. Ein Beispiel für eine Vertrauensbeziehungsrichtlinie finden Sie unter [Serviceübergreifende Confused-Deputy-Prävention](#). Informationen zum Ändern der Vertrauensrichtlinie einer Rolle finden Sie unter [Ändern einer Rolle](#).

4. Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon-Personalize-IAM-Service-Rolle die Berechtigung zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).
5. Führen Sie die Schritte unter aus [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#), um Amazon Personalize Zugriff auf Ihre Amazon S3-Ressourcen zu gewähren.

Themen

- [Benutzern die Erlaubnis geben, auf Amazon Personalize zuzugreifen](#)
- [Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen](#)
- [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#)
- [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#)

Benutzern die Erlaubnis geben, auf Amazon Personalize zuzugreifen

Um Ihren Benutzern Zugriff auf Amazon Personalize zu gewähren, erstellen Sie eine IAM-Richtlinie, die Ihnen den Zugriff auf Ihre Amazon Personalize-Ressourcen gewährt und eine Rolle an Amazon Personalize weitergibt. Sie verwenden diese Richtlinie dann, wenn Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzufügen.

Erstellen IAM-Richtlinie für Ihre Benutzer

Erstellen Sie eine IAM-Richtlinie, die Amazon Personalize vollen Zugriff auf Ihre Amazon Personalize-Ressourcen gewährt.

So verwenden Sie den JSON-Richtlinienditor zum Erstellen einer Richtlinie

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite Policies (Richtlinien).


Wenn Sie zum ersten Mal Policies (Richtlinien) auswählen, erscheint die Seite Welcome to Managed Policies (Willkommen bei verwalteten Richtlinien). Wählen Sie Get Started.

3. Wählen Sie oben auf der Seite Create policy (Richtlinie erstellen) aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.

5. Geben Sie folgendes JSON-Richtliniendokument ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "personalize.amazonaws.com"
        }
      }
    }
  ]
}
```

6. Wählen Sie Weiter.

 Note

Sie können jederzeit zwischen den Editoroptionen Visual und JSON wechseln. Wenn Sie jedoch Änderungen vornehmen oder im Visual-Editor Next (Weiter) wählen, strukturiert IAM Ihre Richtlinie möglicherweise um, um sie für den visuellen Editor zu optimieren. Weitere Informationen finden Sie unter [Richtlinienrestrukturierung](#) im IAM-Benutzerhandbuch.

7. Geben Sie auf der Seite unter Name einen Namen und unter Description eine Beschreibung für die Richtlinie ein, die Sie erstellen. Überprüfen Sie Permissions defined in this policy (In dieser Richtlinie definierte Berechtigungen), um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden.

8. Wählen Sie `Create policy` (Richtlinie erstellen) aus, um Ihre neue Richtlinie zu speichern.

Um nur die zum Ausführen einer Aufgabe in Amazon Personalize erforderlichen Berechtigungen zu erteilen, ändern Sie die vorherige Richtlinie ein. Eine vollständige Liste der Amazon Personalize-Aktionen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Personalize](#).

Bereitstellung des Zugriffs auf Amazon Personalize

Hängen Sie die neue IAM-Richtlinie an, wenn Sie Ihren Benutzern Berechtigungen gewähren.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter [Erstellen einer Rolle für einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen

Um Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen zu erteilen, erstellen Sie eine IAM-Richtlinie, die Amazon Personalize vollen Zugriff auf Ihre Amazon-Personalize-Ressourcen gewährt. Oder Sie können die - `AWS` verwaltete `AmazonPersonalizeFullAccess` Richtlinie verwenden. `AmazonPersonalizeFullAccess` bietet mehr Berechtigungen, als erforderlich sind. Wir empfehlen, eine neue IAM-Richtlinie zu erstellen, die nur die erforderlichen Berechtigungen gewährt. Weitere Informationen über verwaltete Richtlinien finden Sie unter [AWS verwaltete Richtlinien](#).

Nachdem Sie eine Richtlinie erstellt haben, erstellen Sie eine IAM-Rolle für Amazon Personalize und fügen ihr die neue Richtlinie an.

Themen

- [Erstellen einer neuen IAM-Richtlinie für Amazon Personalize](#)
- [Erstellen einer IAM-Rolle für Amazon Personalize](#)

Erstellen einer neuen IAM-Richtlinie für Amazon Personalize

Erstellen Sie eine IAM-Richtlinie, die Amazon Personalize vollen Zugriff auf Ihre Amazon-Personalize-Ressourcen bietet.

So verwenden Sie den JSON-Richtlinienditor zum Erstellen einer Richtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite Policies (Richtlinien).

Wenn Sie zum ersten Mal Policies (Richtlinien) auswählen, erscheint die Seite Welcome to Managed Policies (Willkommen bei verwalteten Richtlinien). Wählen Sie Get Started.

3. Wählen Sie oben auf der Seite Create policy (Richtlinie erstellen) aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie folgendes JSON-Richtliniendokument ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    }
  ]
}
```

6. Wählen Sie Weiter aus.

Note

Sie können jederzeit zwischen den Editoroptionen Visual und JSON wechseln. Wenn Sie jedoch Änderungen vornehmen oder im Visual-Editor Weiter wählen, strukturiert IAM Ihre Richtlinie möglicherweise um, um sie für den visuellen Editor zu optimieren. Weitere Informationen finden Sie unter [Richtlinienrestrukturierung](#) im IAM-Benutzerhandbuch.

7. Geben Sie auf der Seite Prüfen und erstellen unter Richtliniennamen einen Namen und unter Beschreibung (optional) eine Beschreibung für die Richtlinie ein, die Sie erstellen. Überprüfen Sie Permissions defined in this policy (In dieser Richtlinie definierte Berechtigungen), um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden.
8. Wählen Sie Create policy (Richtlinie erstellen) aus, um Ihre neue Richtlinie zu speichern.

Erstellen einer IAM-Rolle für Amazon Personalize


Um Amazon Personalize verwenden zu können, müssen Sie eine - AWS Identity and Access Management Servicerolle für Amazon Personalize erstellen. Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch. Nachdem Sie eine Servicerolle für Amazon Personalize erstellt haben, erteilen Sie der Rolle bei Bedarf zusätzliche Berechtigungen, [Zusätzliche Berechtigungen für Servicerollen](#) die unter aufgeführt sind.

So erstellen Sie die Servicerolle für Amazon Personalize (IAM-Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie für Vertrauenswürdige Entität die Option AWS-Service aus.
4. Wählen Sie für Service oder Anwendungsfall Amazon Personalize und dann den Personalize-Anwendungsfall aus.
5. Wählen Sie Weiter aus.
6. Wählen Sie die Richtlinie aus, die Sie im vorherigen Verfahren erstellt haben.

7. (Optional) Legen Sie eine [Berechtigungsgrenze](#) fest. Dies ist ein erweitertes Feature, das für Servicerollen verfügbar ist, aber nicht für servicegebundene Rollen.
 - a. Öffnen Sie den Abschnitt Berechtigungsgrenze festlegen und wählen Sie dann Berechtigungsgrenze verwenden aus, um die maximalen Rollenberechtigungen zu steuern.

IAM enthält eine Liste der AWS von verwalteten und vom Kunden verwalteten Richtlinien in Ihrem Konto.
 - b. Wählen Sie die Richtlinie aus, die für eine Berechtigungsgrenze verwendet werden soll.
8. Wählen Sie Weiter aus.
9. Geben Sie einen Rollennamen oder ein Suffix für den Rollennamen ein, um den Zweck der Rolle zu identifizieren.

 **Important**

Beachten Sie beim Benennen einer Rolle Folgendes:

- Rollennamen müssen innerhalb Ihres eindeutig sein AWS-Kontound können nicht durch Groß- und Kleinschreibung eindeutig gemacht werden.

Erstellen Sie beispielsweise keine Rollen mit den Namen **PRODRÖLE** und **prodrole**. Wenn ein Rollename in einer Richtlinie oder als Teil eines ARN verwendet wird, wird bei dem Rollennamen die Groß- und Kleinschreibung beachtet. Wenn Kunden in der Konsole jedoch ein Rollename angezeigt wird, z. B. während des Anmeldevorgangs, wird die Groß- und Kleinschreibung des Rollennamens nicht beachtet.

- Sie können den Namen der Rolle nach ihrer Erstellung nicht mehr bearbeiten, da andere Entitäten möglicherweise auf die Rolle verweisen.

10. (Optional) Geben Sie unter Beschreibung eine Beschreibung für die Rolle ein.
11. (Optional) Um die Anwendungsfälle und Berechtigungen für die Rolle zu bearbeiten, wählen Sie in den Abschnitten Schritt 1: Vertrauenswürdige Entitäten auswählen oder Schritt 2: Berechtigungen hinzufügen die Option Bearbeiten aus.
12. (Optional) Um die Rolle zu identifizieren, zu organisieren oder nach ihr zu suchen, fügen Sie Tags als Schlüssel-Wert-Paare hinzu. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter [Markieren von IAM-Ressourcen](#) im IAM-Benutzerhandbuch.
13. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Nachdem Sie eine Rolle für Amazon Personalize erstellt haben, können Sie ihr [Zugriff auf Ihren Amazon S3-Bucket](#) und [alle AWS KMS Schlüssel](#) gewähren.

Zusätzliche Berechtigungen für Servicerollen

Nachdem Sie die Rolle erstellt und ihr Berechtigungen für den Zugriff auf Ihre Ressourcen in Amazon Personalize erteilt haben, gehen Sie wie folgt vor:

1. Ändern Sie die Vertrauensrichtlinie Ihrer Amazon-Personalize-Servicerolle, damit das [Confused-Deputy-Problem](#) verhindert wird. Ein Beispiel für eine Vertrauensbeziehungsrichtlinie finden Sie unter [Serviceübergreifende Confused-Deputy-Prävention](#). Informationen zum Ändern der Vertrauensrichtlinie einer Rolle finden Sie unter [Ändern einer Rolle](#).
2. Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon-Personalize-IAM-Servicerolle die Berechtigung zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren

Gehen Sie wie folgt vor, um Amazon Personalize Zugriff auf Ihren Amazon S3-Bucket zu gewähren:

1. Falls Sie dies noch nicht getan haben, folgen Sie den Schritten [Einrichten von Berechtigungen](#) zur Einrichtung von Berechtigungen, damit Amazon Personalize in Ihrem Namen auf Ihre Ressourcen in Amazon Personalize zugreifen kann.
2. Fügen Sie der Amazon Personalize-Servicerolle eine Richtlinie hinzu (siehe [Erstellen einer IAM-Rolle für Amazon Personalize](#)), die den Zugriff auf Ihren Amazon S3-Bucket ermöglicht. Weitere Informationen finden Sie unter [Ihrer Amazon Personalize-Servicerolle eine Amazon S3-Richtlinie zuordnen](#).
3. Hängen Sie eine Bucket-Richtlinie an den Amazon S3-Bucket an, der Ihre Datendateien enthält, damit Amazon Personalize darauf zugreifen kann. Weitere Informationen finden Sie unter [Eine Amazon Personalize-Zugriffsrichtlinie an Ihren Amazon S3-Bucket anhängen](#).
4. Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis erteilen, Ihren Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Note

Da Amazon Personalize nicht mit AWS VPCs kommuniziert, kann Amazon Personalize nicht mit Amazon S3-Buckets interagieren, die nur VPC-Zugriff zulassen.

Themen

- [Ihrer Amazon Personalize-Servicerolle eine Amazon S3-Richtlinie zuordnen](#)
- [Eine Amazon Personalize-Zugriffsrichtlinie an Ihren Amazon S3-Bucket anhängen](#)

Ihrer Amazon Personalize-Servicerolle eine Amazon S3-Richtlinie zuordnen

Gehen Sie wie folgt vor, um Ihrer Amazon Personalize-Rolle eine Amazon S3-Richtlinie hinzuzufügen:

1. Melden Sie sich bei der IAM-Konsole an (<https://console.aws.amazon.com/iam/>).
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create Policy (Richtlinie erstellen) aus.
3. Wählen Sie die Registerkarte „JSON“ aus und aktualisieren Sie die Richtlinie wie folgt. Ersetzen Sie bucket-name durch den Namen von Ihrem Bucket. Wenn Sie einen Batch-Workflow verwenden, benötigt Amazon Personalize zusätzliche Berechtigungen. Siehe [Richtlinie für Servicerollen für Batch-Workflows](#).

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

```
]
}
```

4. Wählen Sie Next: Markierungen (Weiter: Markierungen). Fügen Sie optional beliebige Tags hinzu und wählen Sie Überprüfen.
5. Geben Sie der Richtlinie einen Namen.
6. (Optional) Geben Sie für Beschreibung einen kurzen Satz ein, der diese Richtlinie beschreibt, zum Beispiel **Allow Amazon Personalize to access its Amazon S3 bucket**.
7. Wählen Sie Create Policy (Richtlinie erstellen) aus.
8. Wählen Sie im Navigationsbereich Rollen und dann die Rolle aus, die Sie für Amazon Personalize erstellt haben. Siehe [Erstellen einer IAM-Rolle für Amazon Personalize](#).
9. Wählen Sie für Berechtigungen die Option Richtlinien anhängen aus.
10. Um eine Richtlinie in der Liste anzuzeigen, geben Sie einen Teil des Richtliniennamens in das Filterfeld Filter policies (Filterrichtlinien) ein.
11. Aktivieren Sie das Kontrollkästchen neben der Richtlinie, die Sie zuvor in diesem Verfahren erstellt haben.
12. Wählen Sie Attach policy (Richtlinie anfügen) aus.

Bevor Ihre Rolle für die Verwendung mit Amazon Personalize bereit ist, müssen Sie auch eine Bucket-Richtlinie an den Amazon S3-Bucket anhängen, der Ihre Daten enthält. Siehe [Eine Amazon Personalize-Zugriffsrichtlinie an Ihren Amazon S3-Bucket anhängen](#).

Richtlinie für Servicerollen für Batch-Workflows

Um einen Batch-Workflow abzuschließen, benötigt Amazon Personalize die Erlaubnis, auf Ihren Amazon S3-Bucket zuzugreifen und Dateien zu diesem hinzuzufügen. Gehen Sie wie oben beschrieben vor, um die folgende Richtlinie mit Ihrer Amazon Personalize-Rolle zu verknüpfen. Ersetzen Sie bucket -name durch den Namen von Ihrem Bucket. Weitere Informationen zu Batch-Workflows finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
```

```

    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"
    ]
  }
]
}

```

Service-Rollenrichtlinie für den Export eines Datensatzes

Um einen Datensatz zu exportieren, benötigt Ihre Amazon Personalize-Service-Rolle die Erlaubnis, die PutObject und ListBucket Actions in Ihrem Amazon S3-Bucket zu verwenden. Die folgende Beispielformel gewährt Amazon Personalize PutObject und ListBucket Berechtigungen. Ersetzen Sie es bucket-name durch den Namen Ihres Buckets und fügen Sie die Richtlinie Ihrer Service-Rolle für Amazon Personalize hinzu. Informationen zum Anhängen von Richtlinien an eine Service-Rolle finden Sie unter [Ihrer Amazon Personalize-Service-Rolle eine Amazon S3-Richtlinie zuordnen](#).

```

{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

Eine Amazon Personalize-Zugriffsrichtlinie an Ihren Amazon S3-Bucket anhängen

Amazon Personalize benötigt eine Genehmigung für den Zugriff auf den S3-Bucket. Für Workflows, die keine Batch-Verarbeitung sind, fügen Sie Ihrem Bucket die folgende Richtlinie hinzu. Ersetzen Sie `bucket-name` durch den Namen von Ihrem Bucket. Informationen zu Batch-Workflows finden Sie unter [Amazon S3-Bucket-Richtlinie für Batch-Workflows](#).

Weitere Informationen zu Amazon S3-Bucket-Richtlinien finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#) .

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

Amazon S3-Bucket-Richtlinie für Batch-Workflows

Für Batch-Workflows benötigt Amazon Personalize die Erlaubnis, auf Ihren Amazon S3-Bucket zuzugreifen und Dateien zu diesem hinzuzufügen. Fügen Sie die folgende Richtlinie zu Ihrem Bucket hinzu: Ersetzen Sie `bucket-name` durch den Namen von Ihrem Bucket.

Weitere Informationen zum Hinzufügen einer Amazon S3-Bucket-Richtlinie zu einem Bucket finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#) . Weitere Informationen zu Batch-Workflows finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

Amazon S3-Bucket-Richtlinie für den Export eines Datensatzes

Um einen Datensatz zu exportieren, benötigt Amazon Personalize die Erlaubnis, die `PutObject` and `ListBucket` Actions in Ihrem Amazon S3-Bucket zu verwenden. Die folgende Beispielrichtlinie gewährt das Amazon Personalize-Prinzip `PutObject` und die `ListBucket` entsprechenden Berechtigungen. Ersetzen Sie es `bucket-name` durch den Namen Ihres Buckets und fügen Sie die Richtlinie Ihrem Bucket hinzu. Informationen zum Hinzufügen einer Amazon S3-Bucket-Richtlinie zu einem Bucket finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#) im Amazon Simple Storage Service-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      }
    }
  ]
}
```

```
    },
    "Action": [
      "s3:PutObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"
    ]
  }
]
```

Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden

Wenn Sie einen AWS Key Management Service (AWS KMS) -Schlüssel angeben, wenn Sie die Amazon Personalize-Konsole oder die APIs verwenden, oder wenn Sie Ihren AWS KMS Schlüssel verwenden, um einen Amazon S3-Bucket zu verschlüsseln, müssen Sie Amazon Personalize die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Um Berechtigungen zu gewähren, müssen Ihre AWS KMS Schlüsselrichtlinie und Ihre IAM-Richtlinie, die Ihrer Servicerolle zugeordnet ist, Amazon Personalize die Erlaubnis zur Verwendung Ihres Schlüssels gewähren. Dies gilt für die Erstellung der folgenden Elemente in Amazon Personalize.

- Datensatzgruppen
- Auftrag zum Import von Datensätzen (nur die AWS KMS Schlüsselrichtlinie muss Berechtigungen gewähren)
- Aufträge zum Exportieren von Datensätzen
- Batch-Inferenzanträge
- Aufträge stapelweise segmentieren
- Metrische Zuschreibungen

Ihre AWS KMS wichtigsten Richtlinien und IAM-Richtlinien müssen Berechtigungen für die folgenden Aktionen gewähren:

- Decrypt
- GenerateDataKey

- DescribeKey
- CreateGrant(nur in wichtigen Richtlinien erforderlich)
- ListGrants

Der Widerruf AWS KMS wichtiger Berechtigungen nach dem Erstellen einer Ressource kann zu Problemen beim Erstellen eines Filters oder beim Abrufen von Empfehlungen führen. Weitere Informationen zu AWS KMS Richtlinien finden Sie unter [Verwenden wichtiger Richtlinien in AWS KMS](#) im AWS Key Management Service-Entwicklerhandbuch. Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch. Informationen zum Anhängen einer IAM-Richtlinie an eine Rolle finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im IAM-Benutzerhandbuch.

Themen

- [Wichtiges politisches Beispiel](#)
- [Beispiel für eine IAM-Richtlinie](#)

Wichtiges politisches Beispiel

Das folgende wichtige Richtlinienbeispiel gewährt Amazon Personalize und Ihrer Rolle die Mindestberechtigungen für die vorherigen Amazon Personalize-Vorgänge. Wenn Sie beim Erstellen einer Datensatzgruppe einen Schlüssel angeben und Daten aus einem Datensatz exportieren möchten, muss Ihre Schlüsselrichtlinie die `GenerateDataKeyWithoutPlaintext` Aktion beinhalten.

```
{
  "Version": "2012-10-17",
  "Id": "key-policy-123",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<account-id>:role/<personalize-role-name>",
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
```

```

        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:ListGrants"
    ],
    "Resource": "*"
}
]
}

```

Beispiel für eine IAM-Richtlinie

Das folgende Beispiel für eine IAM-Richtlinie gewährt einer Rolle die AWS KMS Mindestberechtigungen, die für die vorherigen Amazon Personalize-Vorgänge erforderlich sind. Bei Aufträgen zum Import von Datensätzen muss nur die AWS KMS Schlüsselrichtlinie Berechtigungen gewähren.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey",
        "kms:ListGrants"
      ],
      "Resource": "*"
    }
  ]
}

```

Einrichten der AWS CLI

Die AWS Command Line Interface (AWS CLI) ist ein einheitliches Entwicklertool für die Verwaltung von - AWS Services, einschließlich Amazon Personalize. Wir empfehlen Ihnen, sie zu installieren.

1. Um die zu installieren AWS CLI, folgen Sie den Anweisungen unter [Installieren der AWS Command Line Interface](#) im AWS Command Line Interface -Schnittstellen-Benutzerhandbuch.

2. Um die zu konfigurieren AWS CLI und ein Profil für den Aufruf der einzurichten AWS CLI, folgen Sie den Anweisungen unter [Konfigurieren der AWS CLI](#) im AWS Command Line Interface - Benutzerhandbuch.
3. Führen Sie den folgenden Befehl aus, um zu bestätigen, dass das AWS CLI Profil ordnungsgemäß konfiguriert ist.

```
aws configure --profile default
```

Wenn das Profil korrekt konfiguriert wurde, wird die Ausgabe etwa wie folgt aussehen:

```
AWS Access Key ID [*****52FQ]:  
AWS Secret Access Key [*****xgyZ]:  
Default region name [us-west-2]:  
Default output format [json]:
```

4. Führen Sie die folgenden Befehle aus, um zu überprüfen, ob für die Verwendung mit Amazon Personalize konfiguriert AWS CLI ist.

```
aws personalize help
```

and

```
aws personalize-runtime help
```

and

```
aws personalize-events help
```

Wenn der korrekt konfiguriert AWS CLI ist, wird eine Liste der unterstützten AWS CLI Befehle für Amazon-Personalize-, Amazon-Personalize-Laufzeit- und Amazon-Personalize-Ereignisse angezeigt.

Wenn Sie die einrichten AWS CLI und sie die Befehle für Amazon Personalize nicht erkennt, aktualisieren Sie die AWS CLI. Führen Sie den folgenden Befehl aus AWS CLI, um die zu aktualisieren.

```
pip3 install awscli --upgrade --user
```

Weitere Informationen finden Sie unter [Installieren der AWS CLI mit pip](#).

Einrichten der AWS SDKs

Laden Sie die zu verwendenden AWS SDKs herunter und installieren Sie sie. Dieses Handbuch enthält Beispiele für SDK for Python (Boto3), SDK for Java 2.x und SDK for JavaScript v3.

Weitere Informationen zu anderen AWS SDKs finden Sie unter [Tools für Amazon Web Services](#).

Informationen zum Einrichten von Amplify finden Sie unter [AWS Amplify](#).

- [AWS SDK for Python \(Boto3\)](#)

Um das SDK for Python (Boto3) zu installieren, folgen Sie den [Schnellstartanweisungen](#) in der Boto3-Dokumentation.

- [SDK für Java 2.x](#)

Weitere Informationen zum Einrichten des SDK for Java 2.x finden Sie im Thema [Erste Schritte mit dem SDK for Java 2.x](#) im AWS SDK for Java 2.x -Entwicklerhandbuch.

Codebeispiele für Amazon Personalize finden Sie unter [Amazon-Personalize-Java-Codebeispiele](#) im [AWS SDK-Beispiel](#)-Repository.

- [AWS SDK for JavaScript v3](#)

Weitere Informationen zum Einrichten des -SDK für JavaScript v3 finden Sie im [Thema Erste Schritte mit dem AWS SDK for JavaScript](#) im AWS SDK for JavaScript -Entwicklerhandbuch.

Codebeispiele für Amazon Personalize finden Sie unter [Amazon-Personalize-Codebeispiele für SDK für JavaScript v3](#) im Repository mit [AWS SDK-Beispielen](#).

Erste Schritte

Die folgenden Abschnitte helfen Ihnen bei den ersten Schritten mit Amazon Personalize mit der Amazon Personalize Personalize-Konsole und AWS den AWS CLI SDKs. Die Tutorials verwenden historische Daten, die aus 100.000 Filmbewertungen von 9.700 Filmen von 600 Benutzern bestehen.

Um die Tutorials zu vereinfachen:

- Wir verwenden einen kleinen Datensatz. Dies könnte sich negativ auf alle von Ressourcen generierten Kennzahlen auswirken. Die Tutorials dienen als Einführung in den Amazon Personalize Personalize-Workflow und generieren nicht unbedingt die leistungsstärksten Modelle.
- Wir erstellen nur einen Datensatz mit Artikelinteraktionen und verlassen uns auf die Tatsache, dass ein Benutzer einen Film gesehen hat, und nicht darauf, wie er den Film bewertet hat. Dies vereinfacht die Vorbereitung der Schulungsdaten.
- Wir erfassen keine Live-Benutzerinteraktions-Ereignisse. Weitere Informationen zum Erfassen von Benutzerereignissen finden Sie unter [Ereignisse aufzeichnen](#).

Sie können wählen, ob Sie mit einer Domain-Datensatzgruppe oder einer benutzerdefinierten Datensatzgruppe beginnen möchten:

- Domain-Datensatzgruppen stellen Ressourcen bereit, die je nach Domain für unterschiedliche Anwendungsfälle optimiert sind. Um mit der Erstellung einer Domain-Datensatzgruppe zu beginnen, führen Sie zunächst den [Voraussetzungen für den Einstieg](#) und anschließend das Tutorial unter durch [Erste Schritte mit einer Domain-Datensatzgruppe](#).
- Mit benutzerdefinierten Datensatzgruppen können Sie nur benutzerdefinierte Ressourcen erstellen und konfigurieren. Um mit der Bereitstellung personalisierter Filmempfehlungen für Ihre Benutzer mithilfe von benutzerdefinierten Ressourcen und dem Rezept für die Benutzerpersonalisierung zu beginnen, führen Sie die Tutorials unter durch [Voraussetzungen für den Einstieg](#) und beginnen Sie dann mit den Tutorials. [Erste Schritte mit einer benutzerdefinierten Datensatzgruppe](#)

Wenn Sie die Übung Erste Schritte abgeschlossen haben, gehen Sie wie unter Löschen der von Ihnen erstellten Ressourcen beschrieben vor, um unnötige Kosten [Bereinigen von Ressourcen](#) zu vermeiden.

Themen

- [Voraussetzungen für den Einstieg](#)

- [Erste Schritte mit einer Domain-Datensatzgruppe](#)
- [Erste Schritte mit einer benutzerdefinierten Datensatzgruppe](#)
- [Bereinigen von Ressourcen](#)

Voraussetzungen für den Einstieg

Die folgenden Schritte sind Voraussetzungen für die Übungen zu den ersten Schritten.

1. Richten Sie Berechtigungen ein, damit Amazon Personalize in Ihrem Namen auf Ihre Ressourcen zugreifen kann. Dazu gehört die Erstellung einer Servicerolle für Amazon Personalize und die Gewährung des Zugriffs auf Amazon Personalize Personalize-Ressourcen mit einer IAM-Richtlinie. Weitere Informationen finden Sie unter [Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen](#).
2. Bereiten Sie Ihre Trainingsdaten vor und laden Sie die Daten in Ihren Amazon S3 S3-Bucket hoch:
 - Tutorials zu Domain-Datensatzgruppen finden Sie unter [Erstellung der Trainingsdaten \(Domain-Datensatzgruppe\)](#).
 - Tutorials zu benutzerdefinierten Datensatzgruppen finden Sie unter [Die Trainingsdaten erstellen \(Gruppe „Benutzerdefinierter Datensatz“\)](#).
3. Erteilen Sie Ihrer Amazon Personalize Personalize-Servicerolle die Erlaubnis, auf Ihre Amazon S3 S3-Ressourcen zuzugreifen, wie unter beschrieben [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

Erstellung der Trainingsdaten (Domain-Datensatzgruppe)

Um Trainingsdaten zu erstellen, laden Sie die Filmbewertungsdaten herunter, ändern und speichern Sie sie in einem Amazon Simple Storage Service (Amazon S3) -Bucket. Erteilen Sie Amazon Personalize dann die Erlaubnis, aus dem Bucket zu lesen.

Um die Trainingsdaten zu erstellen

1. Laden Sie die ZIP-Datei mit Filmeinstufungen herunter und entpacken Sie sie ([ml-latest-small.zip](#)) [MovieLens](#) von der für Bildung und Entwicklung empfohlenen Datei (F. Maxwell Harper und Joseph A. Konstan. 2015). Die MovieLens Datensätze: Geschichte und Kontext.

ACM-Transaktionen auf interaktiven intelligenten Systemen (TII) 5, 4:19:1 — 19:19. <https://doi.org/10.1145/2827872>).

2. Öffnen Sie die `ratings.csv` Datei. Diese Datei enthält die Interaktionsdaten für dieses Tutorial.
 - a. Löschen Sie die `rating`-Spalte.
 - b. Benennen Sie die `movieId` Spalten `userId` und `ITEM_ID` jeweils in `USER_ID` und um.
 - c. Fügen Sie eine `EVENT_TYPE`-Spalte hinzu und setzen Sie den Wert für jeden Datensatz auf `watch`. Wenn Sie Microsoft Excel verwenden, können Sie den `EVENT_TYPE` für jeden Datensatz festlegen, indem Sie ihn `watch` in die erste Zelle der Spalte eingeben und dann auf die untere rechte Ecke der Zelle doppelklicken. Ihre Kopfzeile sollte wie folgt aussehen:

USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE

Diese Spalten müssen exakt der Abbildung entsprechen, damit Amazon Personalize die Daten erkennt. Die ersten Zeilen Ihrer Daten sollten wie folgt aussehen:

```
USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE
1, 1, 964982703, watch
1, 3, 964981247, watch
1, 6, 964982224, watch
1, 47, 964983815, watch
1, 50, 964982931, watch
....
....
```

Speichern Sie die `ratings.csv`-Datei.

3. Laden Sie `ratings.csv` es in Ihren Amazon S3 S3-Bucket hoch. Weitere Informationen finden Sie unter [Hochladen von Dateien und Ordern per Drag & Drop](#) im Amazon Simple Storage Service-Benutzerhandbuch.
4. Erteilen Sie Amazon Personalize die Erlaubnis, die Daten im Bucket zu lesen. Weitere Informationen finden Sie unter [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

Die Trainingsdaten erstellen (Gruppe „Benutzerdefinierter Datensatz“)

Um Trainingsdaten zu erstellen, laden Sie die Filmbewertungsdaten herunter, ändern und speichern Sie sie in einem Amazon Simple Storage Service (Amazon S3) -Bucket. Erteilen Sie Amazon Personalize dann die Erlaubnis, aus dem Bucket zu lesen.

1. Laden Sie die ZIP-Datei mit Filmeinstufungen herunter und entpacken Sie sie ([ml-latest-small.zip](#)) aus der [MovieLens](#) Kategorie „Für Bildung und Entwicklung empfohlen“ (F. Maxwell Harper und Joseph A. Konstan. 2015). Die MovieLens Datensätze: Geschichte und Kontext. ACM-Transaktionen auf interaktiven intelligenten Systemen (TII) 5, 4:19:1 — 19:19. <https://doi.org/10.1145/2827872>).
2. Öffnen Sie die `ratings.csv` Datei. Diese Datei enthält die Interaktionsdaten für dieses Tutorial.
 - a. Löschen Sie die `rating`-Spalte.
 - b. Ersetzen Sie die Überschriftenzeile durch Folgendes:

USER_ID, ITEM_ID, TIMESTAMP

Diese Header müssen genau den abgebildeten entsprechen, damit Amazon Personalize die Daten erkennt.

Speichern Sie die `ratings.csv`-Datei.

3. Laden Sie `ratings.csv` es in Ihren Amazon S3 S3-Bucket hoch. Weitere Informationen finden Sie unter [Hochladen von Dateien und Ordnern per Drag & Drop](#) im Amazon Simple Storage Service-Benutzerhandbuch.
4. Erteilen Sie Amazon Personalize die Erlaubnis, die Daten im Bucket zu lesen. Weitere Informationen finden Sie unter [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

Erste Schritte mit einer Domain-Datensatzgruppe

In diesem Tutorial für die ersten Schritte erstellen Sie eine Domain-Datensatzgruppe für die Domäne VIDEO_ON_DEMAND, importieren Interaktionsdaten aus einer CSV-Datei und erstellen eine Empfehlung mit den Top-Picks für Ihren Anwendungsfall. Anschließend verwenden Sie den Empfehlungsdienst, um personalisierte Filmempfehlungen für einen Benutzer zu erhalten. Das

Tutorial verwendet historische Daten, die aus 100.000 Filmbewertungen von 9.700 Filmen von 600 Benutzern bestehen.

Füllen Sie zunächst das aus [Voraussetzungen für den Einstieg](#) und fahren Sie dann, je nachdem, wie Sie Amazon Personalize Personalize-Ressourcen erstellen möchten, mit [Erste Schritte mit einer Domain-Datensatzgruppe \(Konsole\)](#), [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK for Python \(Boto3\)\)](#), [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK für Java 2.x\)](#), oder [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK für JavaScript v3\)](#) fort.

Wenn Sie die Übung Erste Schritte abgeschlossen haben, folgen Sie den Schritten unter Löschen der von Ihnen erstellten Ressourcen, um unnötige Kosten [Bereinigen von Ressourcen](#) zu vermeiden.

Themen

- [Erste Schritte mit einer Domain-Datensatzgruppe \(Konsole\)](#)
- [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK für Java 2.x\)](#)
- [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK for Python \(Boto3\)\)](#)
- [Erste Schritte mit einer Domain-Datensatzgruppe \(SDK für JavaScript v3\)](#)

Erste Schritte mit einer Domain-Datensatzgruppe (Konsole)

In dieser Übung verwenden Sie die Amazon-Personalize-Konsole, um eine Domain-Datensatzgruppe und einen Empfehler zu erstellen, der Filmempfehlungen für einen bestimmten Benutzer zurückgibt.

Bevor Sie mit dieser Übung beginnen, lesen Sie die [Voraussetzungen für den Einstieg](#).

Wenn Sie mit der Übung „Erste Schritte“ fertig sind, führen Sie die Schritte unter aus, um die von Ihnen erstellten Ressourcen [Bereinigen von Ressourcen](#) zu löschen, um unnötige Gebühren zu vermeiden.

Schritt 1: Erstellen einer Domain-Datensatzgruppe

In diesem Verfahren erstellen Sie eine Domain-Datensatzgruppe für die VIDEO_ON_DEMAND-Domäne, erstellen einen Elementinteraktionen-Datensatz mit dem Standardschema für die VIDEO_ON_DEMAND-Domäne und importieren die Elementinteraktionsdaten, die Sie in erstellt haben [Erstellung der Trainingsdaten \(Domain-Datensatzgruppe\)](#).

So erstellen Sie eine Domain-Datensatzgruppe

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie im Navigationsbereich Datensatzgruppe erstellen aus.
3. Geben Sie unter Datensatzgruppendetails einen Namen für Ihre Datensatzgruppe an.
4. Wählen Sie für Domain die Option Video on demand aus. Die von Ihnen gewählte Domain bestimmt das Standardschema, das Sie beim Importieren von Daten verwenden. Sie bestimmt auch, welche Anwendungsfälle für Empfehlungsgeber verfügbar sind. Ihr Bildschirm sollte wie folgt aussehen.

[Amazon Personalize](#) > [Create dataset group](#)

Create dataset group [Info](#)

A dataset group is a container for Amazon Personalize resources, including datasets, domain recommenders, and custom resources.

Dataset group details

Name
The name you enter here distinguishes this dataset group from others.

The dataset group name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Domain
Choose a domain for your use cases.

<input type="radio"/> E-commerce Grow your business by recommending the right products at the right time.	<input checked="" type="radio"/> Video on demand Increase engagement by recommending relevant content to your users.	<input type="radio"/> Custom Create and manage custom resources for your use cases.
---	--	---

► **Tags - optional (0)** [Info](#)
A tag is an administrative label that you assign to resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your costs.

[Cancel](#) [Create group](#)

5. Wählen Sie **Create dataset group** (Datensatz-Gruppe erstellen). Die Seite **Übersicht** wird angezeigt. Fahren Sie mit [Schritt 2: Importieren von Daten](#) fort.

Schritt 2: Importieren von Daten

In diesem Verfahren erstellen Sie einen Datensatz zu Elementinteraktionen mit dem Standard-Domainschema VIDEO_ON_DEMAND. Anschließend importieren Sie die Elementinteraktionsdaten, die Sie in [Erstellung der Trainingsdaten \(Domain-Datensatzgruppe\)](#) erstellt haben.

So importieren Sie Daten

1. Auf der Übersichtsseite in Schritt 1. Erstellen Sie Datensätze und importieren Sie Daten , wählen Sie **Datensatz erstellen** und wählen Sie **Elementinteraktionsdatensatz** aus.
2. Wählen Sie **Daten direkt in Amazon-Personalize-Datensätze importieren** und dann **Weiter** aus.
3. Geben Sie auf der Seite **Schema für Elementinteraktionen konfigurieren** für **Datensatzname** einen Namen für Ihren Datensatz für Elementinteraktionen an.
4. Wählen Sie für **Datensatzschema** die Option **Neues Domänenschema erstellen** aus, indem Sie das vorhandene Standardschema für Ihre Domäne ändern, und geben Sie einen Namen für das Schema ein. Die Schemadefinition wird aktualisiert, um das Standardschema für die Domäne VIDEO_ON_DEMAND anzuzeigen. Lassen Sie das Schema unverändert. Ihr Bildschirm sollte wie folgt aussehen.

Configure item interactions schema [Info](#)

Dataset details

Dataset name

The name you enter here can help you distinguish this dataset import job from others.

The dataset name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Dataset schema

The schema you provide allows Amazon Personalize to understand and import your data.

- Create a new domain schema by modifying the existing default schema for your domain
- Use an existing domain related schema

Schema name

The name you enter here can help you distinguish this schema from others.

The schema name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Schema definition

Verify your data structure matches the following schema.

```
1 {
2   "type": "record",
3   "name": "Interactions",
4   "namespace": "com.amazonaws.personalize.schema",
5   "fields": [
6     {
7       "name": "USER_ID",
8       "type": "string"
9     },
10    {
11      "name": "ITEM_ID",
12      "type": "string"
13    },
14    {
15      "name": "TIMESTAMP",
16      "type": "long"
17    },
18  ]
19 }
```

5. Wählen Sie Weiter aus. Die Seite Importauftrag für den Datensatz für Elementinteraktionen konfigurieren wird angezeigt.
6. Lassen Sie auf der Seite Importauftrag für den Datensatz für Elementinteraktionen konfigurieren die Datenquelle unverändert als Daten aus S3 importieren.
7. Geben Sie Ihrem Importauftrag unter Name des Datensatzimportauftrags einen Namen.
8. Geben Sie unter Datenquelle an, wo Ihre Daten in Amazon Simple Storage Service (S3) gespeichert sind. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

9. Wählen Sie in IAM-Rolle für IAM-Servicerolle die Option Benutzerdefinierten IAM-Rollen-ARN eingeben und geben Sie den Amazon-Ressourcennamen (ARN) der Rolle ein, die Sie in erstellt haben [Erstellen einer IAM-Rolle für Amazon Personalize](#). Ihr Bildschirm sollte wie folgt aussehen.

Configure item interactions dataset import job Info

Dataset import job details

Data import source

Import data from S3
Specify the location where your data is stored in S3.

Incrementally import data with APIs
Incrementally import item interactions data with the event ingestion SDK.


Dataset import job name

The name you enter here can help you distinguish this dataset import job from others.

`my-dataset-import-job-name`

The dataset import job name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ (hyphen).

Data import source

 **Additional S3 bucket policy required**
In addition to the IAM service role defined above, Amazon Personalize also requires you to add a bucket policy to the S3 bucket containing your data files so that it can process them. Follow the instructions [described here](#) to add the required bucket policy to your S3 bucket.

Data location Info

Choose the S3 location of your data.

`s3://bucket/path-to-your-data/`

Your file needs to be in a CSV format and reflect the schema.

IAM Role

IAM service role

Amazon Personalize requires permissions to access your S3 bucket. Choose an existing role with access or create a role in the IAM console with the [AmazonPersonalizeFullAccess](#) IAM policy attached.

Enter a custom IAM role ARN ▼

Custom IAM role ARN

`arn:aws:iam::YourAccountID:role/YourRole`

10. Wählen Sie Import starten, um Daten zu importieren. Die Übersichtsseite für Ihre Domain-Datensatzgruppe wird angezeigt. Notieren Sie sich den Status des Imports im Abschnitt Datensätze einrichten. Wenn der Status lautet, fahren Sie mit Interaction data active fort [Schritt 3: Erstellen eines Empfehlens](#).

Schritt 3: Erstellen eines Empfehlens

In diesem Verfahren erstellen Sie einen Empfehlens für die Top-Auswahlen für Ihren Anwendungsfall für die Domain VIDEO_ON_DEMAND.

So erstellen Sie einen Empfehlens

1. Wählen Sie auf der Übersichtsseite für Ihre Domain-Datensatzgruppe in Schritt 3 die Registerkarte Video-on-Demand-Empfehlungen verwenden und dann Empfehlensprogramme erstellen aus.
2. Wählen Sie auf der Seite Anwendungsfall auswählen die Option Top-Auswahlen für Sie aus und geben Sie einen Empfehlensnamen an. Ihr Bildschirm sollte in etwa wie folgt aussehen.

Choose use case [Info](#)

You use recommenders to get recommendations for specific e-commerce use cases. Amazon Personalize trains the models backing each recommender with the optimal configurations for these use cases.

Video on demand recommenders

Video on demand use cases to create recommenders for

Amazon Personalize will create a recommender for each selected use case.

Because you watched X

Get recommendations for videos that other users also watched based on a video you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

More like X

Get recommendations for videos that are similar to a video you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Most popular

Get recommendations for videos that have been watched by the most users.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Top picks for you

Get personalized content recommendations for a user you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Trending now - new

Get recommendations for videos that are trending now.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

3. Wählen Sie Weiter aus.
4. Lassen Sie die Felder auf der Seite Erweiterte Konfiguration unverändert und wählen Sie Weiter aus.
5. Überprüfen Sie die Empfehlungsdetails und wählen Sie Empfehlungsprogramme erstellen aus, um Ihren Empfehlungsgeber zu erstellen.

Sie können den Status jedes Empfehlers auf der Seite Empfehlungen überwachen. Wenn Ihr Empfehlungsstatus Aktiv lautet, können Sie ihn verwenden, um Empfehlungen in zu erhalten [Schritt 4: Abrufen von Empfehlungen](#).

Schritt 4: Abrufen von Empfehlungen

In diesem Verfahren verwenden Sie den Empfehlungsgeber, den Sie im vorherigen Schritt erstellt haben, um Empfehlungen zu erhalten.

So erhalten Sie Empfehlungen

1. Wählen Sie auf der Übersichtsseite für Ihre Domain-Datensatzgruppe im Navigationsbereich Empfehlung aus.
2. Wählen Sie auf der Seite Empfehlungen Ihren Empfehler aus.
3. Wählen Sie oben rechts Testen aus.
4. Geben Sie unter Empfehlungsparameter eine Benutzer-ID ein. Lassen Sie die anderen Felder unverändert.
5. Wählen Sie Get recommendations (Empfehlungen erhalten). Eine Tabelle mit den 25 empfohlenen Elementen des Benutzers wird angezeigt. Ihr Bildschirm sollte wie folgt aussehen.

Test recommender

Recommendation parameters

User ID

This is the USER_ID you want to get personalized re-ranked item recommendations for. This USER_ID needs to be present in your user-interactions or user dataset.

Filter name- *optional*

Choose an existing filter to apply to your recommendations or create a new filter.

[View](#)

[Create new filter](#)

► **Promotion - *optional* info**

Define additional business rules to promote a subset of items in recommendations. The promotion filter you specify applies to these items instead of any filter you specify above.

[Cancel](#)[Get recommendations](#)

Recommendations (25)

Up to 25 recommendations are displayed. If you applied a promotion, promoted items are distributed randomly.

Recommendation ID

RID-4d12cd84-7d83-4dd9-b849-158b3e8f9ab8

Item ID

592

380

2571

590

150

296

318

780

Erste Schritte mit einer Domain-Datensatzgruppe (SDK für Java 2.x)

In diesem Tutorial erfahren Sie, wie Sie mit dem SDK for Java 2.x eine Domain-Datensatzgruppe für die Domain VIDEO_ON_DEMAND erstellen. In diesem Tutorial erstellen Sie einen Empfehler für den Anwendungsfall Top-Auswahlen.

Um unnötige Gebühren zu vermeiden, [Bereinigen von Ressourcen](#) finden Sie nach Abschluss der Übung „Erste Schritte“ unter Informationen zum Löschen der Ressourcen, die Sie im Tutorial erstellen.

Voraussetzungen

Die folgenden Schritte sind für die Durchführung dieses Tutorials erforderlich:

- Füllen Sie die aus [Voraussetzungen für den Einstieg](#), um die erforderlichen Berechtigungen einzurichten und die Trainingsdaten zu erstellen. Wenn Sie auch die abgeschlossen haben [Erste Schritte mit einer Domain-Datensatzgruppe \(Konsole\)](#), können Sie dieselben Quelldaten wiederverwenden. Wenn Sie Ihre eigenen Quelldaten verwenden, stellen Sie sicher, dass Ihre Daten wie in den Voraussetzungen formatiert sind.
- Richten Sie Ihre SDK for Java 2.x-Umgebung und - AWS Anmeldeinformationen ein, wie unter [Einrichten des AWS SDK for Java 2.x](#) Verfahrens im AWS SDK for Java 2.x -Entwicklerhandbuch angegeben.

Tutorial

In den folgenden Schritten richten Sie Ihr Projekt für die Verwendung von Amazon-Personalize-Paketen ein und erstellen Amazon-Personalize-SDK-für-Java-2.x-Clients. Anschließend importieren Sie Daten, erstellen einen Empfehlungsgeber für den Anwendungsfall Top-Auswahlen und erhalten Empfehlungen.

Schritt 1: Einrichten Ihres Projekts für die Verwendung von Amazon-Personalize-Paketen

Nachdem Sie die Voraussetzungen erfüllt haben, fügen Sie Ihrer pom.xml-Datei Amazon-Personalize-Abhängigkeiten hinzu und importieren Sie Amazon-Personalize-Pakete.

1. Fügen Sie die folgenden Abhängigkeiten zu Ihrer pom.xml-Datei hinzu. Die neuesten Versionsnummern können sich vom Beispielcode unterscheiden.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
```

```
<artifactId>personalize</artifactId>
<version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeruntime</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeevents</artifactId>
  <version>2.16.83</version>
</dependency>
```

2. Fügen Sie Ihrem Projekt die folgenden Importanweisungen hinzu.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
  software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
  software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// recommender packages
import software.amazon.awssdk.services.personalize.model.CreateRecommenderRequest;
import software.amazon.awssdk.services.personalize.model.CreateRecommenderResponse;
import software.amazon.awssdk.services.personalize.model.DescribeRecommenderRequest;
// get recommendations packages
import
  software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
  software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
```

```
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

Schritt 2: Erstellen von Amazon-Personalize-Clients

Nachdem Sie Ihrer pom.xml-Datei Amazon-Personalize-Abhängigkeiten hinzugefügt und die erforderlichen Pakete importiert haben, erstellen Sie die folgenden Amazon-Personalize-Clients:

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
    .region(region)
    .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
    .region(region)
    .build();
```

Schritt 3: Importieren von Daten

Nachdem Sie Ihre Amazon-Personalize-Clients initialisiert haben, importieren Sie die historischen Daten, die Sie erstellt haben, als Sie die abgeschlossen haben [Voraussetzungen für den Einstieg](#). Gehen Sie wie folgt vor, um historische Daten in Amazon Personalize zu importieren:

1. Speichern Sie das folgende Avro-Schema als JSON-Datei in Ihrem Arbeitsverzeichnis. Dieses Schema entspricht den Spalten in der CSV-Datei, die Sie erstellt haben, als Sie abgeschlossen haben [Erstellung der Trainingsdaten \(Domain-Datensatzgruppe\)](#).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {

```

```
        "name": "EVENT_TYPE",
        "type": "string"
    },
    {
        "name": "TIMESTAMP",
        "type": "long"
    }
],
"version": "1.0"
}
```

2. Verwenden Sie die folgende `createDomainSchema` Methode, um ein Domänenschema in Amazon Personalize zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Namen für Ihr Schema, `VIDEO_ON_DEMAND` für die Domain und den Dateipfad für die Schema-JSON-Datei, die Sie im vorherigen Schritt erstellt haben. Die Methode gibt den Amazon-Ressourcennamen (ARN) Ihres neuen Schemas zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createDomainSchema(PersonalizeClient personalizeClient,
String schemaName, String domain,
String filePath) {

String schema = null;
try {
    schema = new String(Files.readAllBytes(Paths.get(filePath)));
} catch (IOException e) {
    System.out.println(e.getMessage());
}

try {
    CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
        .name(schemaName)
        .domain(domain)
        .schema(schema)
        .build();

String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

System.out.println("Schema arn: " + schemaArn);

return schemaArn;
}
```

```
    } catch (PersonalizeException e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
    return "";  
}
```

- Erstellen Sie eine Dataset-Gruppe. Verwenden Sie die folgende `createDomainDatasetGroup` Methode, um eine Domain-Datensatzgruppe zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, einen Namen für die Datensatzgruppe und übergeben Sie `VIDEO_ON_DEMAND` für die Domain. Die Methode gibt den ARN Ihrer neuen Datensatzgruppe zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createDomainDatasetGroup(PersonalizeClient  
personalizeClient,  
    String datasetGroupName,  
    String domain) {  
  
    try {  
        CreateDatasetGroupRequest createDatasetGroupRequest =  
CreateDatasetGroupRequest.builder()  
            .name(datasetGroupName)  
            .domain(domain)  
            .build();  
  
        return  
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();  
    } catch (PersonalizeException e) {  
        System.out.println(e.awsErrorDetails().errorMessage());  
    }  
    return "";  
}
```

- Erstellen Sie einen Datensatz für Elementinteraktionen. Verwenden Sie die folgende `createDataset` Methode, um einen Datensatz für Elementinteraktionen zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Namen für Ihren Datensatz, den ARN Ihres Schemas, den ARN Ihrer Datensatzgruppe und Interactions für den Datensatztyp. Die Methode gibt den ARN Ihres neuen Datensatzes zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createDataset(PersonalizeClient personalizeClient,  
    String datasetName,  
    String datasetGroupArn,
```

```

        String datasetType,
        String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
            .name(datasetName)
            .datasetGroupArn(datasetGroupArn)
            .datasetType(datasetType)
            .schemaArn(schemaArn)
            .build();

        String datasetArn = personalizeClient.createDataset(request)
            .datasetArn();
        System.out.println("Dataset " + datasetName + " created.");
        return datasetArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

5. Importieren Sie Ihre Daten mit einem Datensatz-Importauftrag. Verwenden Sie die folgende `createPersonalizeDatasetImportJob` Methode, um einen Datensatz-Importauftrag zu erstellen.

Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, einen Namen für den Auftrag und den ARN Ihres Interaktionsdatensatzes. Übergeben Sie den Amazon S3-Bucket-Pfad (`s3://bucket name/folder name/ratings.csv`), in dem Sie die Trainingsdaten gespeichert haben, und den ARN Ihrer Servicerolle. Sie haben diese Rolle als Teil der erstellt [Voraussetzungen für den Einstieg](#). Die Methode gibt den ARN Ihres Datensatz-Importauftrags zurück. Speichern Sie es optional zur späteren Verwendung.

```

    public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
        String jobName,
        String datasetArn,
        String s3BucketPath,
        String roleArn) {

        long waitInMilliseconds = 60 * 1000;
        String status;

```

```
String datasetImportJobArn;

try {
    DataSource importDataSource = DataSource.builder()
        .dataLocation(s3BucketPath)
        .build();

    CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
        .datasetArn(datasetArn)
        .dataSource(importDataSource)
        .jobName(jobName)
        .roleArn(roleArn)
        .build();

    datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
        .datasetImportJobArn();

    DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
        .datasetImportJobArn(datasetImportJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        DatasetImportJob datasetImportJob = personalizeClient
            .describeDatasetImportJob(describeDatasetImportJobRequest)
            .datasetImportJob();

        status = datasetImportJob.status();
        System.out.println("Dataset import job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return datasetImportJobArn;
}
```

```
    } catch (PersonalizeException e) {  
        System.out.println(e.awsErrorDetails().errorMessage());  
    }  
    return "";  
}
```

Schritt 4: Erstellen eines Empfehlens

Nachdem Ihr Datensatz-Importauftrag abgeschlossen ist, können Sie einen Empfehler erstellen. Verwenden Sie die folgende `createRecommender` Methode, um einen Empfehler zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, einen Namen für den Empfehler, den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe und `arn:aws:personalize:::recipe/aws-vod-top-picks` für den Rezept-ARN. Die Methode gibt den ARN Ihres neuen Empfehlens zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createRecommender(PersonalizeClient personalizeClient,  
    String name,  
    String datasetGroupArn,  
    String recipeArn) {  
  
    long maxTime = 0;  
    long waitInMilliseconds = 30 * 1000; // 30 seconds  
    String recommenderStatus = "";  
  
    try {  
        CreateRecommenderRequest createRecommenderRequest =  
CreateRecommenderRequest.builder()  
            .datasetGroupArn(datasetGroupArn)  
            .name(name)  
            .recipeArn(recipeArn)  
            .build();  
  
        CreateRecommenderResponse recommenderResponse = personalizeClient  
            .createRecommender(createRecommenderRequest);  
        String recommenderArn = recommenderResponse.recommenderArn();  
        System.out.println("The recommender ARN is " + recommenderArn);  
  
        DescribeRecommenderRequest describeRecommenderRequest =  
DescribeRecommenderRequest.builder()  
            .recommenderArn(recommenderArn)  
            .build();
```



```

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                .status();
            System.out.println("Recommender status: " + recommenderStatus);

            if (recommenderStatus.equals("ACTIVE") ||
recommenderStatus.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return recommenderArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

Schritt 5: Abrufen von Empfehlungen

Nachdem Sie einen Empfehler erstellt haben, erhalten Sie Empfehlungen. Verwenden Sie die folgende `getRecs` Methode, um Empfehlungen für einen Benutzer zu erhalten. Übergeben Sie als Parameter einen Amazon-Personalize-Laufzeit-Client, den Amazon-Ressourcennamen (ARN) des Empfehlers, den Sie im vorherigen Schritt erstellt haben, und eine Benutzer-ID (z. B. 123). Die Methode gibt die Liste der empfohlenen Elemente auf dem Bildschirm aus.

```

    public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
String recommenderArn,
        String userId) {

        try {

```

```
        GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
        .recommenderArn(recommenderArn)
        .numResults(20)
        .userId(userId)
        .build();

        GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
        .getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();

        for (PredictedItem item : items) {
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
        }
    } catch (AwsServiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Erste Schritte mit einer Domain-Datensatzgruppe (SDK for Python (Boto3))

Dieses Tutorial zeigt Ihnen, wie Sie das SDK for Python (Boto3) verwenden, um eine Domain-Datensatzgruppe für die Domäne VIDEO_ON_DEMAND zu erstellen. In diesem Tutorial erstellen Sie eine Empfehlung für den Anwendungsfall „Top Picks for you“.

Um unnötige Kosten zu vermeiden, löschen Sie nach Abschluss dieser Übung „Erste Schritte“ die Ressourcen, die Sie in diesem Tutorial erstellt haben. Weitere Informationen finden Sie unter [Bereinigen von Ressourcen](#).

Themen

- [Voraussetzungen](#)
- [Tutorial](#)
- [Erste Schritte mit Amazon Personalize Personalize-APIs mit Jupyter \(IPython\) -Notebooks](#)

Voraussetzungen

Im Folgenden werden die Voraussetzungen für die Verwendung der Python-Beispiele in diesem Handbuch erläutert:

- Füllen Sie das aus [Voraussetzungen für den Einstieg](#), um die erforderlichen Berechtigungen einzurichten und die Trainingsdaten zu erstellen. Wenn Sie Ihre eigenen Quelldaten verwenden, stellen Sie sicher, dass Ihre Daten wie in den Voraussetzungen beschrieben formatiert sind.
- Richten Sie Ihre AWS SDK for Python (Boto3) Umgebung wie unter beschrieben ein. [Einrichten der AWS SDKs](#)

Tutorial

In den folgenden Schritten verifizieren Sie Ihre Umgebung und erstellen SDK für Python-Clients (Boto3) für Amazon Personalize. Anschließend importieren Sie Daten, erstellen eine Empfehlung für den für Sie am besten geeigneten Anwendungsfall und erhalten Empfehlungen.

Schritt 1: Verifizieren Sie Ihre Python-Umgebung und erstellen Sie Boto3-Clients

Wenn Sie die Schritte für die Voraussetzungen abgeschlossen haben, führen Sie das folgende Python-Beispiel aus, um zu überprüfen, ob Ihre Umgebung korrekt konfiguriert ist. Dieser Code erstellt auch die Amazon Personalize Personalize-Boto3-Clients, die Sie in diesem Tutorial verwenden. Wenn Ihre Umgebung korrekt konfiguriert ist, wird eine Liste der verfügbaren Rezepte angezeigt, und Sie können die anderen Beispiele in diesem Tutorial ausführen.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

Schritt 2: Daten importieren

Nachdem Sie Amazon Personalize Boto3-Clients erstellt und Ihre Umgebung verifiziert haben, importieren Sie die historischen Daten, die Sie beim Abschluss des erstellt haben. [Voraussetzungen für den Einstieg](#) Gehen Sie wie folgt vor, um historische Daten in Amazon Personalize zu importieren:

1. Verwenden Sie den folgenden Code, um ein Schema in Amazon Personalize zu erstellen. `gs-domain-interactions-schema` Ersetzen Sie es durch einen Namen für das Schema.

```
import json
schema = {
    "type": "record",
    "name": "Interactions",
    "namespace": "com.amazonaws.personalize.schema",
    "fields": [
        {
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "EVENT_TYPE",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ],
    "version": "1.0"
}

create_interactions_schema_response = personalize.create_schema(
    name='gs-domain-interactions-schema',
    schema=json.dumps(schema),
    domain='VIDEO_ON_DEMAND'
)

interactions_schema_arn = create_interactions_schema_response['schemaArn']
print(json.dumps(create_interactions_schema_response, indent=2))
```

2. Erstellen Sie eine Dataset-Gruppe mit dem folgenden Code. `dataset group name` Durch einen Namen für die Datensatzgruppe ersetzen.

```
response = personalize.create_dataset_group(
    name = 'dataset group name',
```

```

    domain = 'VIDEO_ON_DEMAND'
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])

```

3. Erstellen Sie einen Datensatz mit Artikelinteraktionen in Ihrer neuen Datensatzgruppe mit dem folgenden Code. Geben Sie dem Datensatz einen Namen und geben Sie die `schema_arn` Daten und `dataset_group_arn` aus den vorherigen Schritten an.

```

response = personalize.create_dataset(
    name = 'interactions-dataset-name',
    schemaArn = interactions_schema_arn,
    datasetGroupArn = dsg_arn,
    datasetType = 'INTERACTIONS'
)

dataset_arn = response['datasetArn']

```

4. Importieren Sie Ihre Daten mit einem Datensatz-Importjob mit dem folgenden Code. Der Code verwendet die Methode `describe_dataset_import_job`, um den Status des Jobs zu verfolgen.

Übergeben Sie Folgendes als Parameter: einen Namen für den Job, den Namen `dataset_arn` aus dem vorherigen Schritt, den Amazon S3 S3-Bucket-Pfad (`s3://bucket name/folder name/ratings.csv`), in dem Sie die Trainingsdaten gespeichert haben, und den ARN Ihrer IAM-Servicerolle. Sie haben diese Rolle als Teil von erstellt. [Voraussetzungen für den Einstieg](#) Amazon Personalize benötigt eine Genehmigung für den Zugriff auf den Bucket. Informationen zur Gewährung von Zugriff finden Sie unter [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

```

import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn'
)

```

```

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)

```

Schritt 4: Erstellen Sie einen Empfehlungsgeber

Nachdem Ihr Datensatz-Importjob abgeschlossen ist, können Sie eine Empfehlung erstellen. Verwenden Sie den folgenden Code, um einen Empfehlungsgeber zu erstellen. Übergeben Sie Folgendes als Parameter: einen Namen für den Empfehlungsgeber, den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe und `arn:aws:personalize:::recipe/aws-vod-top-picks` für den Rezept-ARN. Der Code verwendet die Methode `describe_recommender`, um den Status des Empfehlungsgebers zu verfolgen.

```

import time
create_recommender_response = personalize.create_recommender(
    name = 'gs-python-top-picks',
    recipeArn = 'arn:aws:personalize:::recipe/aws-vod-top-picks',
    datasetGroupArn = ds_group_arn
)
recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
max_time = time.time() + 3*60*60 # 3 hours

```

```
while time.time() < max_time:

    version_response = personalize.describe_recommender(
        recommenderArn = recommender_arn
    )
    status = version_response["recommender"]["status"]

    if status == "ACTIVE":
        print("Creation succeeded for {}".format(recommender_arn))

    elif status == "CREATE FAILED":
        print("Creation failed for {}".format(recommender_arn))

    if status == "ACTIVE":
        break
    else:
        print("Recommender creation is still in progress")

    time.sleep(60)
```

Schritt 5: Empfehlungen einholen

Nachdem Sie einen Empfehlungsgeber erstellt haben, verwenden Sie ihn, um Empfehlungen mit dem folgenden Code abzurufen. Übergeben Sie als Parameter den Amazon-Ressourcennamen (ARN) des Empfehlungsgebers, den Sie im vorherigen Schritt erstellt haben, und eine Benutzer-ID (z. B.123). Die Methode druckt die Liste der empfohlenen Artikel.

```
response = personalizeRt.get_recommendations(
    recommenderArn = "arn:aws:personalize:us-west-2:014025156336:recommender/gs-python-
top-picks-89",
    userId = '123'
)
print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Erste Schritte mit Amazon Personalize Personalize-APIs mit Jupyter (IPython) - Notebooks

[Um mit der Erstellung von Domain-Datensatzgruppen mit Jupyter-Notizbüchern zu beginnen, klonen oder laden Sie eine Reihe von Notizbüchern herunter, die sich im Ordner `notebooks_managed_domains` des Amazon Personalize Personalize-Beispiel-Repositorys](#)

[befinden](#). Die Notizbücher führen Sie durch das Importieren von Trainingsdaten, das Erstellen einer Empfehlung und das Abrufen von Empfehlungen mit Amazon Personalize.

Note

Bevor Sie mit den Notizbüchern beginnen, stellen Sie sicher, dass Sie Ihre Umgebung gemäß den Schritten in der README.md erstellt haben

Erste Schritte mit einer Domain-Datensatzgruppe (SDK für JavaScript v3)

In diesem Tutorial erfahren Sie, wie Sie mit AWS SDK for JavaScript Version 3 eine Domain-Datensatzgruppe für die Domäne VIDEO_ON_DEMAND erstellen. In diesem Tutorial erstellen Sie eine Empfehlung für die für Sie am besten geeigneten Anwendungsfall.

Den in diesem Tutorial verwendeten Code finden Sie in den [Amazon Personalize Personalize-Codebeispielen für SDK for JavaScript v3](#) im Repository für AWSSDK-Codebeispiele. GitHub

Um unnötige Kosten zu vermeiden, löschen Sie nach Abschluss dieser Übung „Erste Schritte“ die Ressourcen, die Sie in diesem Tutorial erstellt haben. Weitere Informationen finden Sie unter [Bereinigen von Ressourcen](#).

Themen

- [Voraussetzungen](#)
- [Tutorial](#)

Voraussetzungen

Für die Durchführung dieses Tutorials sind die folgenden Schritte erforderlich:

- Gehen Sie wie folgt vor [Voraussetzungen für den Einstieg](#), um die erforderlichen Berechtigungen einzurichten und die Trainingsdaten zu erstellen. Wenn Sie auch das abgeschlossen haben [Erste Schritte mit einer Domain-Datensatzgruppe \(Konsole\)](#), können Sie dieselben Quelldaten wiederverwenden. Wenn Sie Ihre eigenen Quelldaten verwenden, stellen Sie sicher, dass Ihre Daten wie in den Voraussetzungen beschrieben formatiert sind.
- Richten Sie das SDK für JavaScript und die AWS Anmeldeinformationen ein, wie im JavaScript Verfahren [zum Einrichten des SDK für](#) im AWS SDK for JavaScript Entwicklerhandbuch beschrieben.

Tutorial

In den folgenden Schritten installieren Sie die erforderlichen Abhängigkeiten. Anschließend erstellen Sie eine Datensatzgruppe, importieren Daten, erstellen eine Empfehlung für den für Sie am besten geeigneten Anwendungsfall und erhalten Empfehlungen.

Wenn Sie Node.js verwenden, können Sie jedes Codebeispiel ausführen, indem Sie das Beispiel als JavaScript Datei speichern und dann ausführen `node <fileName.js>`.

Schritt 1: Amazon Personalize installieren

Nachdem Sie die Voraussetzungen erfüllt haben, installieren Sie die folgenden Amazon Personalize Personalize-Abhängigkeiten:

- `@aws-sdk/client-personalize`
- `@aws-sdk/client-personalize-runtime`
- `@aws-sdk/client-personalize-events` (optional für dieses Tutorial, aber erforderlich, wenn Sie [Ereignisse aufzeichnen möchten, nachdem Sie Ihren Empfehlungsgeber erstellt](#) haben)

Das Folgende ist ein Beispiel für eine `package.json` Datei, die Sie verwenden können. Um die Abhängigkeiten mit Node.js zu installieren, navigieren Sie zu dem Speicherort, an dem Sie die `package.json` Datei gespeichert haben, und führen Sie den Befehl `npm install`.

```
{
  "name": "personalize-js-project",
  "version": "1.0.0",
  "description": "personalize operations",
  "type": "module",
  "author": "Author Name <email@address.com>",
  "license": "ISC",
  "dependencies": {
    "@aws-sdk/client-personalize": "^3.350.0",
    "@aws-sdk/client-personalize-events": "^3.350.0",
    "@aws-sdk/client-personalize-runtime": "^3.350.0",
    "fs": "^0.0.1-security"
  },
  "compilerOptions": {
    "resolveJsonModule": true,
    "esModuleInterop": true
  }
}
```

```
}
```

Schritt 2: Amazon Personalize erstellen

Nachdem Sie die Abhängigkeiten installiert haben, erstellen Sie Ihre Amazon Personalize Personalize-Clients. In diesem Tutorial gehen die Codebeispiele davon aus, dass Sie die Clients in einer Datei mit dem Namen erstellen, die in einem Verzeichnis mit dem Namen `personalizeClients.js` gespeichert ist.

Im Folgenden wird ein Beispiel für eine `personalizeClient.js`-Datei dargestellt.

```
import { PersonalizeClient } from "@aws-sdk/client-personalize";
import { PersonalizeRuntimeClient } from "@aws-sdk/client-personalize-runtime";
import { PersonalizeEventsClient } from "@aws-sdk/client-personalize-events";
// Set your AWS region.
const REGION = "region"; //e.g. "us-east-1"

const personalizeClient = new PersonalizeClient({ region: REGION});
const personalizeEventsClient = new PersonalizeEventsClient({ region: REGION});
const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: REGION});

export { personalizeClient, personalizeEventsClient, personalizeRuntimeClient };
```

Schritt 3: Daten importieren

Nachdem Sie Ihre Amazon Personalize-Kunden erstellt haben, importieren Sie die historischen Daten, die Sie beim Abschluss des [Voraussetzungen für den Einstieg](#) erstellt haben. Gehen Sie wie folgt vor, um historische Daten in Amazon Personalize zu importieren:

1. Speichern Sie das folgende Avro-Schema als JSON-Datei in Ihrem Arbeitsverzeichnis. Dieses Schema entspricht den Spalten in der CSV-Datei, die Sie beim Ausfüllen von [Erstellung der Trainingsdaten \(Domain-Datensatzgruppe\)](#) erstellt haben.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    }
  ]
}
```

```

    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}

```

- Erstellen Sie ein Domain-Schema in Amazon Personalize mit dem folgenden `createDomainSchema.js` Code. `SCHEMA_PATH` ersetzen Sie es durch den Pfad zur `schema.json`-Datei, die Sie gerade erstellt haben. Aktualisieren Sie `createSchemaParam`, um einen Namen für das Schema anzugeben, und um zu spezifizieren. `domain VIDEO_ON_DEMAND`

```

// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from 'fs';

let schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = 'TEST' // for unit tests.
}

// Set the domain schema parameters.
export const createDomainSchemaParam = {

```

```

    name: 'NAME', /* required */
    schema: mySchema, /* required */
    domain: 'DOMAIN' /* required for a domain dataset group, specify ECOMMERCE or
    VIDEO_ON_DEMAND */
  };

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateSchemaCommand(createDomainSchemaParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

3. Erstellen Sie eine Domain-Datensatzgruppe in Amazon Personalize mit dem folgenden `createDomainDatasetGroup.js` Code. Aktualisieren Sie `dendomainDatasetGroupParams`, um einen Namen für die Datensatzgruppe anzugeben, und um zu `domain` spezifizieren `VIDEO_ON_DEMAND`.

```

// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: 'NAME', /* required */
  domain: 'DOMAIN' /* required for a domain dsG, specify ECOMMERCE or
  VIDEO_ON_DEMAND */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateDatasetGroupCommand(domainDatasetGroupParams));
    console.log("Success", response);
  }
};

```

```

    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

4. Erstellen Sie einen Datensatz mit Artikelinteraktionen in Amazon Personalize mit dem folgenden `createDataset.js` Code. Aktualisieren Sie `createDatasetParam`, um den Amazon-Ressourcennamen (ARN) der Datensatzgruppe und des Schemas anzugeben, die Sie gerade erstellt haben, geben Sie dem Datensatz einen Namen und geben Sie für `datasetType` `anInteractions`.

```

// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
  datasetType: 'DATASET_TYPE', /* required */
  name: 'NAME', /* required */
  schemaArn: 'SCHEMA_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateDatasetCommand(createDatasetParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

5. Importieren Sie Ihre Daten mit dem folgenden `createDatasetImportJob.js` Code. Aktualisieren Sie `datasetImportJobParam`, um Folgendes anzugeben:

- Geben Sie einen Namen für den Job und den ARN Ihres Interactions-Datensatzes an.
- Geben Sie für `dataLocation` den Amazon S3 S3-Bucket-Pfad (`s3://bucket name/folder name/ratings.csv`) an, in dem Sie die Trainingsdaten gespeichert haben.
- `roleArn` Geben Sie den Amazon-Ressourcennamen für Ihre Amazon Personalize-Servicerolle an. Sie haben diese Rolle als Teil von erstellt. [Voraussetzungen für den Einstieg](#)

```
// Get service clients module and commands using ES6 syntax.
import {CreateDatasetImportJobCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN', /* required */
  dataSource: { /* required */
    dataLocation: 'S3_PATH'
  },
  jobName: 'NAME', /* required */
  roleArn: 'ROLE_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
  CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Schritt 4: Erstellen Sie einen Empfehlungsgeber

Nachdem Ihr Datensatz-Importjob abgeschlossen ist, können Sie eine Empfehlung erstellen.

Verwenden Sie den folgenden `createRecommender.js` Code, um einen Empfehlungsgeber zu

erstellen. Aktualisieren Sie das `createRecommenderParam` wie folgt: Geben Sie einen Namen für den Empfehlungsgeber an, geben Sie den ARN Ihrer Datensatzgruppe an und `recipeArn` geben Sie `anarn:aws:personalize:::recipe/aws-vod-top-picks`.

```
// Get service clients module and commands using ES6 syntax.
import { CreateRecommenderCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the recommender's parameters.
export const createRecommenderParam = {
  name: 'NAME', /* required */
  recipeArn: 'RECIPE_ARN', /* required */
  datasetGroupArn: 'DATASET_GROUP_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
  CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Schritt 5: Empfehlungen einholen

Nachdem Sie einen Empfehlungsgeber erstellt haben, verwenden Sie ihn, um Empfehlungen zu erhalten. Verwenden Sie den folgenden `getRecommendations.js` Code, um Empfehlungen für einen Benutzer abzurufen. Aktualisieren Sie `dengetRecommendationsParam`, um den ARN des Empfehlungsgebers anzugeben, den Sie im vorherigen Schritt erstellt haben, und geben Sie eine Benutzer-ID an (z. B.123).

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
  "@aws-sdk/client-personalize-runtime";
```

```
import { personalizeRuntimeClient } from "../libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  recommenderArn: 'RECOMMENDER_ARN', /* required */
  userId: 'USER_ID', /* required */
  numResults: 15 /* optional */
}

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
  GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Erste Schritte mit einer benutzerdefinierten Datensatzgruppe

In dieser Anleitung für die ersten Schritte erfahren Sie, wie Sie Ihren Benutzern personalisierte Filmempfehlungen mit einer benutzerdefinierten Datensatzgruppe und dem Rezept für die Benutzerpersonalisierung geben können. Das Tutorial verwendet historische Daten, die aus 100.000 Filmbewertungen zu 9.700 Filmen von 600 Benutzern bestehen.

Füllen Sie zunächst das aus [Voraussetzungen für den Einstieg](#) und fahren Sie dann mit entweder [Erste Schritte \(Konsole\)](#), [Erste Schritte \(AWS CLI\)](#) oder [Erste Schritte \(SDK for Python \(Boto3\)\)](#), oder fort [Erste Schritte \(SDK für Java 2.x\)](#).

Wenn Sie die Übung „Erste Schritte“ abgeschlossen haben, folgen Sie den Schritten unter So löschen Sie die von Ihnen erstellten Ressourcen, [Bereinigen von Ressourcen](#) um unnötige Kosten zu vermeiden.

Themen

- [Erste Schritte \(Konsole\)](#)
- [Erste Schritte \(AWS CLI\)](#)

- [Erste Schritte \(SDK for Python \(Boto3\)\)](#)
- [Erste Schritte \(SDK für Java 2.x\)](#)

Erste Schritte (Konsole)

In dieser Übung verwenden Sie die Amazon Personalize Personalize-Konsole, um eine benutzerdefinierte Datensatzgruppe mit einer Lösung zu erstellen, die Filmempfehlungen für einen bestimmten Benutzer zurückgibt. Bevor Sie mit dieser Übung beginnen, lesen Sie die [Voraussetzungen für den Einstieg](#).

Wenn Sie die Übung Erste Schritte abgeschlossen haben, gehen Sie wie unter Löschen der von Ihnen erstellten Ressourcen beschrieben vor, [Bereinigen von Ressourcen](#) um unnötige Kosten zu vermeiden.

Schritt 1: Erstellen Sie eine Datensatzgruppe und einen Datensatz

In diesem Verfahren erstellen Sie zunächst ein Dataset-Gruppe. Als Nächstes erstellen Sie einen Datensatz mit Amazon Personalize Personalize-Artikelinteraktionen in der Datensatzgruppe.

Um eine Datensatzgruppe und einen Datensatz zu erstellen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie Create dataset group (Dataset-Gruppe erstellen).
3. Geben Sie für Dataset group details (Details zur Dataset-Gruppe) in Dataset group name (Dataset-Gruppenname) einen Namen für die Dataset-Gruppe an.
4. Wählen Sie für Domain die Option Benutzerdefiniert. Ihr Bildschirm sollte wie folgt aussehen:

Create dataset group Info

Dataset group details

Name
The name you enter here distinguishes this dataset group from others.

The dataset group name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Domain
Choose a domain for your use cases.

<input type="radio"/> E-commerce Grow your business by recommending the right products at the right time.	<input type="radio"/> Video on demand Increase engagement by recommending relevant content to your users.	<input checked="" type="radio"/> Custom Create and manage custom resources for your use cases.
---	---	--

Cancel **Create dataset group and continue**

5. Wählen Sie Create dataset group (Dataset-Gruppe erstellen). Die Übersichtsseite wird angezeigt.
6. Wählen Sie unter Datensätze einrichten die Option Datensatz erstellen und anschließend Datensatz mit Artikelinteraktionen aus.
7. Wählen Sie Daten direkt in Amazon Personalize Personalize-Datensätze importieren und dann Weiter.
8. Geben Sie auf der Seite „Datensatz mit Artikelinteraktionen konfigurieren“ für Datensatzname einen Namen für Ihren Datensatz ein.
9. Wählen Sie für Datensatzschema die Option Neues Schema erstellen aus. Im Abschnitt Schemafelder wird ein Interactions Minimalschema angezeigt. Das Schema entspricht den Headern, die Sie der ratings.csv Datei zuvor hinzugefügt haben, sodass Sie keine Änderungen vornehmen müssen. Wenn Sie die Trainingsdaten noch nicht erstellt haben, finden Sie weitere Informationen unter [Voraussetzungen für den Einstieg](#).
10. Geben Sie unter Schemaname einen Namen für das neue Schema an. Ihr Bildschirm sollte wie folgt aussehen:

Configure item interactions schema [Info](#)

Dataset details

Dataset name

The name you enter here can help you distinguish this dataset import job from others.

The dataset name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Dataset schema

The schema you provide allows Amazon Personalize to understand and import your data.

- Create new schema
- Use an existing schema

Schema name

The name you enter here can help you distinguish this schema from others.

The schema name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Schema definition

Verify your data structure matches the following schema.

```
1 {
2   "type": "record",
3   "name": "Interactions",
4   "namespace": "com.amazonaws.personalize.schema",
5   "fields": [
6     {
7       "name": "USER_ID",
8       "type": "string"
9     },
10    {
11      "name": "ITEM_ID",
12      "type": "string"
13    },
14    {
15      "name": "TIMESTAMP",
16      "type": "long"
17    }
18  ],
19  "version": "1.0"
20 }
```

11. Wählen Sie Weiter. Die Auftragsseite für den Import von Datensätzen für Artikelinteraktionen konfigurieren wird angezeigt. Schließen Sie als Nächstes [Schritt 2: Daten zu Artikelinteraktionen importieren](#) den Import der Interaktionsdaten ab.

Schritt 2: Daten zu Artikelinteraktionen importieren

Nachdem Sie einen Datensatz erstellt haben, ist es an der Zeit, Daten zu Artikelinteraktionen in den Datensatz zu importieren.

Um Daten zu Artikelinteraktionen zu importieren

1. Wählen Sie auf der Jobseite „Datensatz für Artikelinteraktionen konfigurieren“ für Datenimportquelle die Option Daten aus S3 importieren aus.
2. Geben Sie unter Name des Datensatz-Import-Jobs einen Namen für Ihren Importjob an.
3. Wenn Sie Amazon Personalize Personalize-Berechtigungen im Dialogfeld Zusätzliche S3-Bucket-Richtlinie erforderlich nicht erteilt haben, folgen Sie den Anweisungen, um [die erforderliche Amazon S3 S3-Bucket-Richtlinie hinzuzufügen](#).
4. Geben Sie unter Datenspeicherort an, wo Ihre Filmdatendatei in Amazon Simple Storage Service (S3) gespeichert ist. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

5. Behalten Sie im Abschnitt IAM-Rolle für die IAM-Dienstrolle die Standardauswahl Enter a custom IAM role ARN bei.
6. Geben Sie für Custom IAM role ARN (ARN der benutzerdefinierten IAM-Rolle) die Rolle an, die Sie im Thema [Erstellen einer IAM-Rolle für Amazon Personalize](#) angelegt haben.

Die Abschnitte „Auftragsdetails für den Dataset-Import“ und „IAM-Rolle“ sollten wie folgt aussehen:

Dataset import job details

Data import source

Import data from S3
Specify the location where your data is stored in S3.

Incrementally import data with APIs
Incrementally import interactions data with the event ingestion SDK.

Dataset import job name

The name you enter here can help you distinguish this dataset import job from others.

The dataset import job name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).



Additional S3 bucket policy required

In addition to the IAM service role defined above, Amazon Personalize also requires you to add a bucket policy to the S3 bucket containing your data files so that it can process them. Follow the instructions [described here](#) to add the required bucket policy to your S3 bucket.

Data location **Info**

Choose the S3 location of your data.

Your file needs to be in a CSV format and reflect the schema.

IAM Role

IAM service role

Amazon Personalize requires permissions to access your S3 bucket. Choose an existing role with access or create a role in the IAM console with the [AmazonPersonalizeFullAccess](#) IAM policy attached.

Custom IAM role ARN

7. Wählen Sie Finish (Abschließen). Der Datenimportjob wird gestartet und die Übersichtsseite wird angezeigt. Anfänglich lautet der Status „Erstellung ausstehend“ (gefolgt von „Erstellung in Bearbeitung“), und die Schaltfläche „Lösung erstellen“ ist deaktiviert.

Die Zeit, die für den Import der Daten benötigt wird, hängt von der Größe des Datasets ab. Wenn der Datenimportauftrag abgeschlossen ist, ändert sich der Status in Aktiv und die Schaltfläche Lösung erstellen ist aktiviert.

8. Nachdem der Importauftrag abgeschlossen ist, klicken Sie auf die Schaltfläche Lösung erstellen. Die Seite Create solution (Lösung erstellen) wird angezeigt. Nachdem Sie die Daten importiert haben, können Sie nun eine Lösung in erstellen [Schritt 3: Erstellen Sie eine Lösung](#).

Schritt 3: Erstellen Sie eine Lösung

In diesem Verfahren verwenden Sie den Datensatz, in den Sie importiert haben, [Schritt 2: Daten zu Artikelinteraktionen importieren](#) um ein Modell zu trainieren. Ein geschultes Modell wird als eine Lösungsversion bezeichnet.

So erstellen Sie eine Lösung

1. Wählen Sie auf der Übersichtsseite für Ihre Datensatzgruppe unter Benutzerdefinierte Ressourcen verwenden die Option Lösung erstellen aus.
2. Wählen Sie als Lösungstyp die Option Artikelempfehlung aus, um Artikelempfehlungen für Ihre Benutzer zu erhalten.
3. Geben Sie für Solution name (Name der Lösung) einen Namen für die Lösung an.
4. Wählen Sie als Lösungstyp die Option Artikelempfehlungen aus.
5. Wählen Sie für Rezept die Option aws-user-personalization.

Ihr Bildschirm sollte wie folgt aussehen:

Specify solution details [Info](#)

Choose your solution type and choose the recipe to use in training.

Solution details

Solution name

The solution name that you enter here can help you distinguish this solution from others.

The solution name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Solution type [Info](#)

Choose the type of solution you want to create. The type determines what recipes are available for solution creation.

Item recommendation

Create a solution that generates item recommendations.

Supports Generative AI

Action recommendation - new

Create a solution that predicts the best next course of actions for your users.

User segmentation



Create a solution that predicts groups of users based on item input data.

Recipe

Recipes are preconfigured algorithms tailored to specific use cases.

Predicts items a user will interact with and performs exploration on cold items. Based on Hierarchical...

► **Tags - optional** (0) [Info](#)

A tag is an administrative label that you assign to  resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your  costs.

Cancel

Next

6. Wählen Sie Weiter. Lassen Sie die optionalen Felder für die erweiterte Konfiguration unverändert.
7. Wählen Sie Weiter und überprüfen Sie die Details der Lösung.
8. Wählen Sie Create solution (Lösung erstellen) aus. Die Schulung zur Lösungsversion beginnt und die Übersichtsseite wird angezeigt.
9. Erweitern Sie im Navigationsbereich Benutzerdefinierte Ressourcen und wählen Sie Lösungen und Rezepte aus.
10. Wählen Sie im Bereich Lösungen Ihre Lösung aus. Die Detailseite für die Lösungsseite wird angezeigt. Auf der Seite mit den Lösungsversionen wird der Status Ihres Modells aufgeführt.

Wenn der Status der Lösungsversion Aktiv lautet, können Sie zu wechseln [Schritt 4: Erstellen einer Kampagne](#).

Schritt 4: Erstellen einer Kampagne

In diesem Verfahren erstellen Sie eine Kampagne, die die Lösungsversion bereitstellt, die Sie im vorherigen Schritt erstellt haben.

So erstellen Sie eine Kampagne

1. Erweitern Sie im Navigationsbereich den Bereich Benutzerdefinierte Ressourcen und wählen Sie Kampagnen aus.
2. Wählen Sie Create campaign (Kampagne erstellen). Die Seite Neue Kampagne erstellen wird angezeigt.
3. Geben Sie in Campaign details für Campaign name einen Namen für Ihre Kampagne an.
4. Wählen Sie für Lösung die Lösung aus, die Sie im vorherigen Schritt erstellt haben, und behalten Sie für Lösungsversions-ID die Standardeinstellung bei.
5. Behalten Sie für Minimum provisioned transactions per second (Mindestens bereitgestellte Transaktionen pro Sekunde) den Standardwert von 1 bei. Lassen Sie die Felder für die Kampagnenkonfiguration unverändert.

Ihr Bildschirm sollte wie folgt aussehen:

Create new campaign

Campaign details

Campaign name

The text you enter here appears in the Campaign dashboard and detail page. It can help you distinguish this campaign from others.

The campaign name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Solution

The selected solution is used to generate the recommendations provided in your campaign.

Solution version ID

The selected solution version is used to generate the recommendations provided in your campaign.

Tue, 26 Apr 2022 18:22:43 GMT

Minimum provisioned transactions per second [Info](#)

The minimum amount of throughput in transactions per second (TPS) that is provisioned for this campaign.

Enter a number from 1-500.

Campaign configuration

Exploration weight [Info](#)

Configure how frequently recommendations include items with less interactions data or relevance. The greater the value (closer to 1), the more exploration. At 0, no exploration occurs.



Choose a value between 0 and 1.

Exploration item age cut off

Enter the item age cut off, in days since the latest interaction, to define the scope of item exploration.

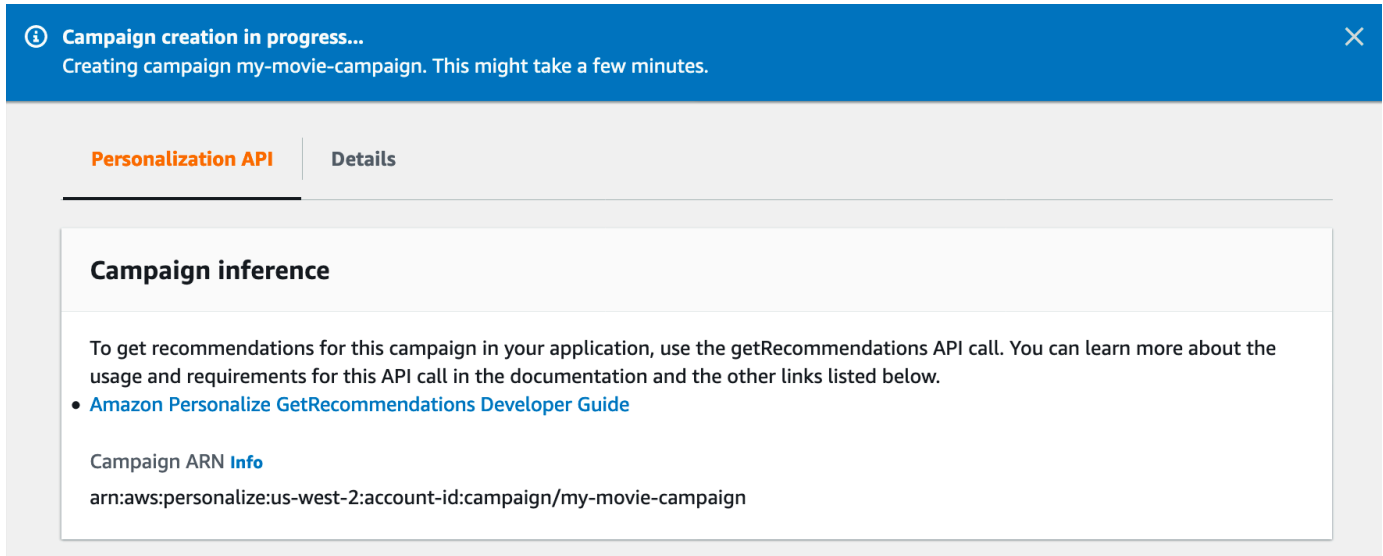
Choose a positive value.

► Tags - optional (0) [Info](#)

A tag is an administrative label that you assign to resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your costs.

- Wählen Sie **Create campaign** (Kampagne erstellen). Die Kampagnenerstellung wird gestartet und die Seiten mit den Kampagnendetails und dem Abschnitt Personalisierungs-API werden angezeigt.

Ihr Bildschirm sollte wie folgt aussehen:



Das Erstellen einer Kampagne kann einige Minuten dauern. Nachdem Amazon Personalize Ihre Kampagne erstellt hat, wird die Seite aktualisiert und zeigt nun den Abschnitt Ergebnisse der Testkampagne an. Ihr Bildschirm sollte wie folgt aussehen:

my-movie-campaign Delete Update

Personalization API Details

Campaign inference

To get recommendations for this campaign in your application, use the `getRecommendations` API call. You can learn more about the usage and requirements for this API call in the documentation and the other links listed below.

- [Amazon Personalize GetRecommendations Developer Guide](#)

Campaign ARN [Info](#)
arn:aws:personalize:us-west-2:acct-id:campaign/my-movie-campaign

Test campaign results

User ID Info
This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your user-interactions or user dataset.

Filter name - optional
To filter recommendations using a filter with parameters and a campaign you deployed or updated before November 10, 2020, you must update the campaign or create a new campaign.

None ▼ Refresh

To view filter details, [go to the filter page](#).

Context - optional
Provide context as key/value pairs.

Key	Value	
<input type="text" value="Enter a metadata attribute"/>	<input type="text" value="Enter value"/>	Remove

Add new context

Get recommendations

Schritt 5: Holen Sie sich Empfehlungen

Bei diesem Verfahren verwenden Sie die Kampagne, die Sie im vorherigen Schritt erstellt haben, um Empfehlungen zu erhalten.

So erhalten Sie Empfehlungen

- Geben Sie im Feld Ergebnisse der Testkampagne für Benutzer-ID einen Wert aus dem Bewertungsdatensatz an, **83** z. B. Behalten Sie für den Filternamen die Standardauswahl Keine bei und lassen Sie die Kontextfelder leer.
- Wählen Sie `Get recommendations` (Empfehlungen erhalten). Im Bereich „Empfehlungen“ werden die Artikel-IDs und Punktzahlen für die empfohlenen Artikel aufgeführt.

Ihr Bildschirm sollte wie folgt aussehen:

Test campaign results

User ID [Info](#)
This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your user-interactions or user dataset.

Filter name - optional
To filter recommendations using a filter with parameters and a campaign you deployed or updated before November 10, 2020, you must update the campaign or create a new campaign.

None

To view filter details, [go to the filter page](#).

Context - optional
Provide context as key/value pairs.

Key	Value	
<input type="text" value="Enter a metadata attribute"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new context"/>		

Recommendations

Recommendation ID
RID-example-ID-number-123

Item ID	Score
1391	0.0117211
1302	0.0077976
2012	0.0072628
1676	0.0061814

Erste Schritte (AWS CLI)

In dieser Übung verwenden Sie AWS Command Line Interface (AWS CLI), um Amazon Personalize zu erkunden. Sie erstellen eine Kampagne, die Filmempfehlungen für eine bestimmte Benutzer-ID zurückgibt.

Führen Sie die folgenden Schritte durch, bevor Sie mit dieser Übung beginnen:

- Lesen Sie „Erste Schritte mit [Voraussetzungen für den Einstieg](#)“.

- Richten Sie die AWS CLI wie unter [Einrichten der AWS CLI](#) angegeben ein.

Wenn Sie mit der Übung Erste Schritte fertig sind, folgen Sie den Schritten unter Löschen der von Ihnen erstellten Ressourcen, [Bereinigen von Ressourcen](#) um unnötige Kosten zu vermeiden.

Note

Die AWS CLI Befehle in dieser Übung wurden unter Linux getestet. Informationen zur Verwendung der AWS CLI Befehle unter Windows finden Sie AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch [unter Angeben von Parameterwerten für](#).

Schritt 1: Trainingsdaten importieren

Befolgen Sie die Schritte zum Erstellen einer Dataset-Gruppe, fügen Sie ein Dataset zur Gruppe hinzu und aktualisieren Sie das Dataset dann mit den Filmbewertungsdaten.

1. Sie erstellen eine Dataset-Gruppe, indem Sie den folgenden Befehl ausführen. Sie können die Datensatzgruppe verschlüsseln, indem Sie einen [AWS Key Management Service](#) Schlüssel-ARN und den ARN einer IAM-Rolle, die über Zugriffsberechtigungen für diesen Schlüssel verfügt, als Eingabeparameter übergeben. Weitere Informationen zur API finden Sie unter [CreateDatasetGroup](#).

```
aws personalize create-dataset-group --name MovieRatingDatasetGroup --kms-key-arn arn:aws:kms:us-west-2:01234567890:key/1682a1e7-a94d-4d92-bbdf-837d3b62315e --role-arn arn:aws:iam::01234567890:KMS-key-access
```

Der Dataset-Gruppen-ARN wird angezeigt, zum Beispiel:

```
{
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/MovieRatingDatasetGroup"
}
```


Zeigen Sie die von Ihnen erstellte Dataset-Gruppe mit dem Befehl `describe-dataset-group` an, indem Sie den zurückgegebenen Dataset-Gruppen-ARN angeben.

```
aws personalize describe-dataset-group \<
```

```
--dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/  
MovieRatingDatasetGroup
```

Die Dataset-Gruppe und ihre Eigenschaften werden angezeigt, z. B.:

```
{  
  "datasetGroup": {  
    "name": "MovieRatingDatasetGroup",  
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/  
MovieRatingDatasetGroup",  
    "status": "ACTIVE",  
    "creationDateTime": 1542392161.262,  
    "lastUpdatedDateTime": 1542396513.377  
  }  
}
```

 Note

Warten Sie, bis für den status der Dataset-Gruppe ACTIVE angezeigt wird, bevor Sie ein Dataset in der Gruppe erstellen. Diese Operation läuft in der Regel schnell ab.

Wenn Sie den ARN der Dataset-Gruppe vergessen haben, verwenden Sie den Befehl `list-dataset-groups`, um alle von Ihnen erstellten Dataset-Gruppen zusammen mit ihren ARNs anzuzeigen.

```
aws personalize list-dataset-groups
```

 Note

Die `list-objects` Befehle `describe-object` und `list-objects` sind für die meisten Amazon Personalize verfügbar. Diese Befehle werden im weiteren Verlauf dieser Übung nicht angezeigt, stehen aber zur Verfügung.

- Erstellen Sie eine Schemadatei im JSON-Format, indem Sie den folgenden Code in einer Datei mit dem Namen `MovieRatingSchema.json` speichern. Das Schema stimmt mit den Überschriften überein, die Sie `ratings.csv` vorher hinzugefügt haben. Der Schemaname

lautet `Interactions`, was einem der von Amazon Personalize erkannten Datensatztypen entspricht. Weitere Informationen finden Sie unter [Schemata](#).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

- Erstellen Sie ein Schema, indem Sie den folgenden Befehl ausführen. Geben Sie die im vorherigen Schritt gespeicherte Datei an. Das Beispiel zeigt die Datei als Teil des aktuellen Ordners. Weitere Informationen zur API finden Sie unter [CreateSchema](#).

```
aws personalize create-schema \
  --name MovieRatingSchema \
  --schema file://MovieRatingSchema.json
```

Der Amazon-Ressourcenname (ARN) des Schemas wird angezeigt, z. B.:

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema"
}
```

- Erstellen Sie ein leeres Dataset, indem Sie den folgenden Befehl ausführen. Geben Sie den Dataset-Gruppen-ARN und den Schema-ARN an, die in den vorherigen Schritten zurückgegeben

wurden. Der `dataset-type` muss mit dem Schema name aus dem vorherigen Schritt übereinstimmen. Weitere Informationen zur API finden Sie unter [CreateDataset](#).

```
aws personalize create-dataset \  
  --name MovieRatingDataset \  
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/  
MovieRatingDatasetGroup \  
  --dataset-type Interactions \  
  --schema-arn arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema
```

Der Dataset-ARN wird angezeigt, zum Beispiel:

```
{  
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/  
MovieRatingDatasetGroup/INTERACTIONS"  
}
```

5. Fügen Sie dem Dataset die Schulungsdaten hinzu.

- a. Erstellen Sie den Dataset-Importauftrag, indem Sie den folgenden Befehl ausführen. Geben Sie den Dataset-ARN und den Amazon S3 S3-Bucket-Namen an, die in den vorherigen Schritten zurückgegeben wurden. Geben Sie den ARN für die Rolle AWS Identity and Access Management (IAM) an, in [Erstellen einer IAM-Rolle für Amazon Personalize](#) dem Sie ihn erstellt haben. Weitere Informationen zur API finden Sie unter [CreateDatasetImportJob](#).

```
aws personalize create-dataset-import-job \  
  --job-name MovieRatingImportJob \  
  --dataset-arn arn:aws:personalize:us-west-2:acct-id:dataset/  
MovieRatingDatasetGroup/INTERACTIONS \  
  --data-source dataLocation=s3://bucketname/ratings.csv \  
  --role-arn roleArn
```

Der Dataset-Importauftrags-ARN wird angezeigt, z. B.:

```
{  
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-  
job/MovieRatingImportJob"  
}
```



- b. Überprüfen Sie mit dem Befehl `describe-dataset-import-job` den Status. Geben Sie den Dataset-Importauftrag-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [DescribeDatasetImportJob](#).

```
aws personalize describe-dataset-import-job \  
  --dataset-import-job-arn arn:aws:personalize:us-west-2:acct-id:dataset-  
import-job/MovieRatingImportJob
```

Die Eigenschaften des Dataset-Importauftrags, einschließlich des Erstellungsstatus, werden angezeigt. Anfänglich wird für den status CREATE PENDING angezeigt, z. B.:

```
{  
  "datasetImportJob": {  
    "jobName": "MovieRatingImportJob",  
    "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-  
import-job/MovieRatingImportJob",  
    "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/  
MovieRatingDatasetGroup/INTERACTIONS",  
    "dataSource": {  
      "dataLocation": "s3://<bucketname>/ratings.csv"  
    },  
    "roleArn": "role-arn",  
    "status": "CREATE PENDING",  
    "creationDateTime": 1542392161.837,  
    "lastUpdatedDateTime": 1542393013.377  
  }  
}
```

Der Dataset-Importvorgang ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Danach sind Sie zur Schulung des Modells anhand des angegebenen Datensets bereit.

 Note

Das Importieren ist zeitaufwändig. Warten Sie, bis der Dataset-Importvorgang abgeschlossen ist, bevor Sie das Modell mit dem Dataset schulen.

Schritt 2: Erstellen Sie eine Lösung (trainieren Sie das Modell)

Zwei Schritte sind erforderlich, um ein Modell erstmals zu schulen. Zuerst erstellen Sie die Konfiguration für die Schulung des Modells mithilfe der Operation [CreateSolution](#). Zweitens schulen Sie das Modell mithilfe der Operation [CreateSolutionVersion](#).

Sie trainieren ein Modell anhand eines Rezepts und Ihrer Trainingsdaten. Amazon Personalize bietet eine Reihe vordefinierter Rezepte. Weitere Informationen finden Sie unter [Ein Rezept wählen](#). Für diese Übung verwenden Sie das Rezept für die Benutzerpersonalisierung.

1. Erstellen Sie die Konfiguration für die Schulung eines Modells, indem Sie den folgenden Befehl ausführen.

```
aws personalize create-solution \  
  --name MovieSolution \  
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/  
MovieRatingDatasetGroup \  
  --recipe-arn arn:aws:personalize:::recipe/aws-user-personalization
```

Der Lösungs-ARN wird angezeigt, z. B.:

```
{  
  "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution"  
}
```

2. Überprüfen Sie den Status create mithilfe des Befehls `describe-solution`. Geben Sie den Lösungs-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [DescribeSolution](#).

```
aws personalize describe-solution \  
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

Die Eigenschaften der Lösung und der status der Erstellung werden angezeigt. Anfänglich wird der Status als CREATE PENDING angezeigt, z. B.:

```
{  
  "solution": {  
    "name": "MovieSolution",  
    "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/  
MovieSolution",
```

```

    "performHPO": false,
    "performAutoML": false,
    "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
    "solutionConfig": {},
    "status": "ACTIVE",
    "creationDateTime": "2021-05-12T16:27:59.819000-07:00",
    "lastUpdatedDateTime": "2021-05-12T16:27:59.819000-07:00"
  }
}

```

3. Wenn die Lösung AKTIV ist, trainieren Sie das Modell, indem Sie den folgenden Befehl ausführen.

```

aws personalize create-solution-version \
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution

```

Der ARN der Lösungsversion wird angezeigt, z. B.:

```

{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>"
}

```

Überprüfen Sie den Schulungsstatus der Lösungsversion, indem Sie den Befehl `describe-solution-version` verwenden. Geben Sie den Lösungsversions-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [DescribeSolutionVersion](#).

```

aws personalize describe-solution-version \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/version-id

```

Die Eigenschaften der Lösungsversion und des Schulungsstatus werden angezeigt. Anfänglich wird der Status als CREATE PENDING angezeigt, z. B.:

```

{
  "solutionVersion": {

```

```

    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
    ...,
    "status": "CREATE PENDING"
  }
}

```

4. Wenn die Lösungsversion **AKTIV** status ist, ist die Schulung abgeschlossen.

Jetzt können Sie die Trainingsmetriken überprüfen und mithilfe der Lösungsversion eine Kampagne erstellen.

Note

Die Schulung ist zeitaufwändig. Warten Sie, bis die Schulung abgeschlossen ist (der Schulungsstatus der Lösungsversion wird als **AKTIV** angezeigt), bevor Sie diese Version der Lösung in einer Kampagne verwenden.

5. Sie können die Leistung der Lösungsversion validieren, indem Sie ihre Metriken überprüfen. Mit dem folgenden Befehl können Sie die Metriken für die Lösungsversion abrufen. Geben Sie den Lösungsversions-ARN an, der vorher zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [GetSolutionMetrics](#).

```

aws personalize get-solution-metrics \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/version-id

```

Es wird eine Beispielantwort angezeigt:

```

{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/www-
solution/<version-id>",
  "metrics": {
    "coverage": 0.0485,
    "mean_reciprocal_rank_at_25": 0.0381,
    "normalized_discounted_cumulative_gain_at_10": 0.0363,
    "normalized_discounted_cumulative_gain_at_25": 0.0984,
    "normalized_discounted_cumulative_gain_at_5": 0.0175,
    "precision_at_10": 0.0107,
    "precision_at_25": 0.0207,
    "precision_at_5": 0.0107
  }
}

```

```
}  
}
```

Schritt 3: Erstellen Sie eine Kampagne (stellen Sie die Lösung bereit)

Bevor Sie Empfehlungen erhalten können, müssen Sie eine Lösungsversion bereitstellen. Das Bereitstellen einer Lösung wird auch als Erstellen einer Kampagne bezeichnet. Wenn Sie Ihre Kampagne erstellt haben, kann Ihre Client-Anwendung mithilfe der [GetRecommendations](#)-API-Empfehlungen erhalten.

1. Erstellen Sie eine Kampagne, indem Sie den folgenden Befehl ausführen. Geben Sie den Lösungsversions-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [CreateCampaign](#).

```
aws personalize create-campaign \  
  --name MovieRecommendationCampaign \  
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/  
MovieSolution/version-id \  
  --min-provisioned-tps 1
```

Es wird eine Beispielantwort angezeigt:

```
{  
  "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/  
MovieRecommendationCampaign"  
}
```

2. Überprüfen Sie den Bereitstellungsstatus, indem Sie den folgenden Befehl ausführen. Geben Sie den Kampagnen-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [DescribeCampaign](#).

```
aws personalize describe-campaign \  
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/  
MovieRecommendationCampaign
```

Es wird eine Beispielantwort angezeigt:

```
{  
  "campaign": {
```

```
    "name": "MovieRecommendationCampaign",
    "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign",
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
    "minProvisionedTPS": "1",
    "creationDateTime": 1543864775.923,
    "lastUpdatedDateTime": 1543864791.923,
    "status": "CREATE_IN_PROGRESS"
  }
}
```

Note

Warten Sie, bis für den status ACTIVE angezeigt wird, bevor Sie Empfehlungen von der Kampagne erhalten.

Schritt 4: Holen Sie sich Empfehlungen

Sie können durch Ausführen des Befehls `get-recommendations` Empfehlungen erhalten. Geben Sie den Kampagnen-ARN an, der im vorherigen Schritt zurückgegeben wurde. Geben Sie in der Anforderung eine Benutzer-ID aus dem Filmbewertungs-Dataset an. Weitere Informationen zur API finden Sie unter [GetRecommendations](#).

Note

Nicht alle Rezepte unterstützen die `GetRecommendations`-API. Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

Der AWS CLI-Befehl `personalize-runtime`, den Sie in diesem Schritt aufrufen, unterscheidet sich von dem in vorherigen Schritten.

```
aws personalize-runtime get-recommendations \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign \
  --user-id 123
```

In der Antwort gibt die Kampagne eine Liste von Elementempfehlungen (Film-IDs) zurück, die für den Benutzer interessant sein könnten. Die Liste ist in absteigender Reihenfolge nach der Bedeutung für den Benutzer sortiert.

```
{
  "itemList": [
    {
      "itemId": "14"
    },
    {
      "itemId": "15"
    },
    {
      "itemId": "275"
    },
    {
      "itemId": "283"
    },
    {
      "itemId": "273"
    },
    ...
  ]
}
```

Erste Schritte (SDK for Python (Boto3))

Dieses Tutorial zeigt Ihnen, wie Sie den Amazon Personalize Personalize-Workflow mit dem SDK for Python (Boto3) von Anfang bis Ende abschließen.

Um unnötige Kosten zu vermeiden, löschen Sie nach Abschluss dieser Übung „Erste Schritte“ die Ressourcen, die Sie in diesem Tutorial erstellt haben. Weitere Informationen finden Sie unter [Bereinigen von Ressourcen](#).

Themen

- [Voraussetzungen](#)
- [Tutorial](#)
- [Erste Schritte mit Amazon Personalize Personalize-APIs mit Jupyter \(IPython\) -Notebooks](#)

Voraussetzungen

Im Folgenden werden die Voraussetzungen für die Verwendung der Python-Beispiele in diesem Handbuch erläutert:

- Füllen Sie das aus [Voraussetzungen für den Einstieg](#), um die erforderlichen Berechtigungen einzurichten und die Trainingsdaten zu erstellen. Wenn Sie Ihre eigenen Quelldaten verwenden, stellen Sie sicher, dass Ihre Daten entsprechend den Voraussetzungen formatiert sind.
- Richten Sie Ihre AWS SDK for Python (Boto3) Umgebung wie unter beschrieben ein. [Einrichten der AWS SDKs](#)

Tutorial

In den folgenden Schritten verifizieren Sie Ihre Umgebung und erstellen SDK für Python-Clients (Boto3) für Amazon Personalize. Anschließend importieren Sie Daten, erstellen und implementieren eine Lösungsversion mit einer Kampagne und erhalten Empfehlungen.

Schritt 1: Verifizieren Sie Ihre Python-Umgebung und erstellen Sie Boto3-Clients

Wenn Sie die Schritte für die Voraussetzungen abgeschlossen haben, führen Sie das folgende Python-Beispiel aus, um zu überprüfen, ob Ihre Umgebung korrekt konfiguriert ist. Dieser Code erstellt auch die Amazon Personalize Personalize-Boto3-Clients, die Sie in diesem Tutorial verwenden. Wenn Ihre Umgebung korrekt konfiguriert ist, wird eine Liste der verfügbaren Rezepte angezeigt, und Sie können die anderen Beispiele in diesem Tutorial ausführen.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

Schritt 2: Daten importieren

Nachdem Sie Amazon Personalize Boto3-Clients erstellt und Ihre Umgebung verifiziert haben, importieren Sie die historischen Daten, die Sie beim Abschluss des erstellt haben. [Voraussetzungen für den Einstieg](#) Gehen Sie wie folgt vor, um historische Daten in Amazon Personalize zu importieren:

1. Verwenden Sie den folgenden Code, um ein Schema in Amazon Personalize zu erstellen. `getting-started-schema` Ersetzen Sie es durch einen Namen für das Schema.

```
import json
schema = {
    "type": "record",
    "name": "Interactions",
    "namespace": "com.amazonaws.personalize.schema",
    "fields": [
        {
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ],
    "version": "1.0"
}

create_interactions_schema_response = personalize.create_schema(
    name='getting-started-schema',
    schema=json.dumps(schema)
)

interactions_schema_arn = create_interactions_schema_response['schemaArn']
print(json.dumps(create_interactions_schema_response, indent=2))
```

2. Erstellen Sie eine Dataset-Gruppe mit dem folgenden Code. `dataset_group_name` Durch einen Namen für die Datensatzgruppe ersetzen.

```
response = personalize.create_dataset_group(name = 'dataset_group_name')
dataset_group_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dataset_group_arn)
['datasetGroup']

print('Name: ' + description['name'])
```

```
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

- Erstellen Sie einen Datensatz mit Artikelinteraktionen in Ihrer neuen Datensatzgruppe mit dem folgenden Code. Geben Sie dem Datensatz einen Namen und geben Sie die `schema_arn` Daten und `dataset_group_arn` aus den vorherigen Schritten an.

```
response = personalize.create_dataset(
    name = 'datase_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'Interactions'
)

dataset_arn = response['datasetArn']
```

- Importieren Sie Ihre Daten mit einem Datensatz-Importjob mit dem folgenden Code. Der Code verwendet die Methode `describe_dataset_import_job`, um den Status des Jobs zu verfolgen.

Übergeben Sie Folgendes als Parameter: einen Namen für den Job, den Namen `dataset_arn` aus dem vorherigen Schritt, den Amazon S3 S3-Bucket-Pfad (`s3://bucket name/folder name/ratings.csv`), in dem Sie die Trainingsdaten gespeichert haben, und den ARN Ihrer IAM-Servicerolle. Sie haben diese Rolle als Teil von erstellt. [Voraussetzungen für den Einstieg](#) Amazon Personalize benötigt eine Genehmigung für den Zugriff auf den Bucket. Siehe [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

```
import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
```

```

print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)

```

Schritt 3: Erstellen Sie eine Lösung

Nach dem Import Ihrer Daten erstellen Sie wie folgt eine Lösung und eine Lösungsversion. Die Lösung enthält die Konfigurationen zum Trainieren eines Modells, und eine Lösungsversion ist ein trainiertes Modell.

1. Erstellen Sie eine neue Lösung mit dem folgenden Code. Übergeben Sie Folgendes als Parameter: das `dataset_group_arn` von früher, einen Namen für die Lösung und den ARN für das Benutzerpersonalisierungsrezept (`arn:aws:personalize:::recipe/aws-user-personalization`). Speichern Sie den ARN Ihrer neuen Lösung für die spätere Verwendung.

```

create_solution_response = personalize.create_solution(
    name='solution name',
    recipeArn= 'arn:aws:personalize:::recipe/aws-user-personalization',
    datasetGroupArn = 'dataset group arn'
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)

```

2. Erstellen Sie eine Lösungsversion mit dem folgenden Code. Übergeben Sie als Parameter den `solution_arn` aus dem vorherigen Schritt. Der folgende Code erstellt eine Lösungsversion. Während des Trainings verwendet der Code den [DescribeSolutionVersion](#) Vorgang, um den Status der Lösungsversion abzurufen. Wenn das Training abgeschlossen ist, gibt die Methode den ARN Ihrer neuen Lösungsversion zurück. Speichern Sie es für eine spätere Verwendung.

```
import time
import json

create_solution_version_response = personalize.create_solution_version(
    solutionArn = solution_arn
)

solution_version_arn = create_solution_version_response['solutionVersionArn']
print(json.dumps(create_solution_version_response, indent=2))

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_solution_version_response = personalize.describe_solution_version(
        solutionVersionArn = solution_version_arn
    )
    status = describe_solution_version_response["solutionVersion"]["status"]
    print("SolutionVersion: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

Schritt 4: Erstellen einer Kampagne

Nachdem Sie Ihre Lösungsversion erstellt haben, stellen Sie sie mit einer Amazon Personalize Personalize-Kampagne bereit. Verwenden Sie den folgenden Code, um eine Kampagne zu erstellen, die Ihre Lösungsversion bereitstellt. Übergeben Sie Folgendes als Parameter: `thesolution_version_arn`, und einen Namen für die Kampagne. Die Methode gibt den Amazon-Ressourcennamen (ARN) Ihrer neuen Kampagne zurück. Speichern Sie es für eine spätere Verwendung.

```
response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn'
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
```

```
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

Schritt 5: Empfehlungen einholen

Nachdem Sie eine Kampagne erstellt haben, können Sie sie verwenden, um Empfehlungen zu erhalten. Der folgende Code zeigt, wie Sie Empfehlungen aus einer Kampagne erhalten und die ID jedes empfohlenen Artikels ausdrucken können. Übergeben Sie den ARN der Kampagne, die Sie im vorherigen Schritt erstellt haben. Als Benutzer-ID übergeben Sie die ID eines Benutzers, der aus den Trainingsdaten stammt, z. 123 B.

```
response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = '123',
    numResults = 10
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Erste Schritte mit Amazon Personalize Personalize-APIs mit Jupyter (IPython) - Notebooks

[Um mit der Verwendung von Amazon Personalize mithilfe von Jupyter-Notizbüchern zu beginnen, klonen oder laden Sie eine Reihe von Notizbüchern herunter, die sich im Ordner `getting_started` des Amazon Personalize Personalize-Beispiel-Repositorys befinden.](#) Die Notizbücher führen Sie durch den Import von Trainingsdaten, die Erstellung einer Lösung, die Erstellung einer Kampagne und das Einholen von Empfehlungen mithilfe von Amazon Personalize.

Note

[Bevor Sie mit den Notizbüchern beginnen, stellen Sie sicher, dass Sie Ihre Umgebung gemäß den Schritten in der README.md erstellt haben](#)

Erste Schritte (SDK für Java 2.x)

In diesem Tutorial erfahren Sie, wie Sie den Amazon-Personalize-Workflow von Anfang bis Ende mit dem abschließen AWS SDK for Java 2.x.

Um unnötige Gebühren zu vermeiden, führen Sie nach Abschluss der Übung „Erste Schritte“ die Schritte unter aus, [Bereinigen von Ressourcen](#) um die Ressourcen zu löschen, die Sie im Tutorial erstellen.

Weitere Beispiele finden Sie unter [Abschließen des Amazon-Personalize-Projekts](#).

Themen

- [Voraussetzungen](#)
- [Abschließen des Amazon-Personalize-Projekts](#)

Voraussetzungen

Die folgenden Schritte sind für die Durchführung dieses Tutorials erforderlich:

- Füllen Sie die aus [Voraussetzungen für den Einstieg](#), um die erforderlichen Berechtigungen einzurichten und die Trainingsdaten zu erstellen. Sie können dieselben Quelldaten verwenden, die in den - [Erste Schritte \(Konsole\)](#) oder - [Erste Schritte \(AWS CLI\)](#) Übungen verwendet werden. Wenn Sie Ihre eigenen Quelldaten verwenden, stellen Sie sicher, dass Ihre Daten wie in den Voraussetzungen formatiert sind.
- Richten Sie Ihre SDK for Java 2.x-Umgebung und - AWS Anmeldeinformationen ein, wie unter [Einrichten des AWS SDK for Java 2.x](#) Verfahrens im AWS SDK for Java 2.x -Entwicklerhandbuch angegeben.

Tutorial

In den folgenden Schritten richten Sie Ihr Projekt für die Verwendung von Amazon-Personalize-Paketen und die Erstellung von Amazon-Personalize-SDK-für-Java-2.x-Clients ein. Anschließend importieren Sie Daten, erstellen und stellen eine Lösungsversion mit einer Kampagne bereit und erhalten Empfehlungen.

Schritt 1: Einrichten Ihres Projekts für die Verwendung von Amazon-Personalize-Paketen

Nachdem Sie die Voraussetzungen erfüllt haben, fügen Sie Ihrer pom.xml-Datei Amazon-Personalize-Abhängigkeiten hinzu und importieren Sie Amazon-Personalize-Pakete.

1. Fügen Sie die folgenden Abhängigkeiten zu Ihrer pom.xml-Datei hinzu. Die neuesten Versionsnummern können sich vom Beispielcode unterscheiden.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalize</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeruntime</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeevents</artifactId>
  <version>2.16.83</version>
</dependency>
```

2. Fügen Sie Ihrem Projekt die folgenden Importanweisungen hinzu.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
  software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
  software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// solution packages
import software.amazon.awssdk.services.personalize.model.CreateSolutionRequest;
import software.amazon.awssdk.services.personalize.model.CreateSolutionResponse;
```

```
// solution version packages
import software.amazon.awssdk.services.personalize.model.DescribeSolutionRequest;
import
    software.amazon.awssdk.services.personalize.model.CreateSolutionVersionRequest;
import
    software.amazon.awssdk.services.personalize.model.CreateSolutionVersionResponse;
import
    software.amazon.awssdk.services.personalize.model.DescribeSolutionVersionRequest;
// campaign packages
import software.amazon.awssdk.services.personalize.model.CreateCampaignRequest;
import software.amazon.awssdk.services.personalize.model.CreateCampaignResponse;
// get recommendations packages
import
    software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
    software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

Schritt 2: Erstellen von Amazon-Personalize-Clients

Nachdem Sie Ihrer pom.xml-Datei Amazon-Personalize-Abhängigkeiten hinzugefügt und die erforderlichen Pakete importiert haben, erstellen Sie die folgenden Amazon-Personalize-Clients:

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
    .region(region)
    .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
    .region(region)
    .build();
```

Schritt 3: Importieren von Daten

Nachdem Sie Ihre Amazon-Personalize-Clients initialisiert haben, importieren Sie die historischen Daten, die Sie erstellt haben, als Sie die abgeschlossen haben [Voraussetzungen für den Einstieg](#). Gehen Sie wie folgt vor, um historische Daten in Amazon Personalize zu importieren:

1. Speichern Sie das folgende Avro-Schema als JSON-Datei in Ihrem Arbeitsverzeichnis. Dieses Schema entspricht den Spalten in der CSV-Datei, die Sie erstellt haben, als Sie abgeschlossen haben [Voraussetzungen für den Einstieg](#).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

2. Verwenden Sie die folgende `createSchema` Methode, um ein Schema in Amazon Personalize zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Namen für Ihr Schema und den Dateipfad für die Schema-JSON-Datei, die Sie im vorherigen Schritt erstellt haben. Die Methode gibt den Amazon-Ressourcennamen (ARN) Ihres neuen Schemas zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createSchema(PersonalizeClient personalizeClient, String
schemaName, String filePath) {

    String schema = null;
    try {
        schema = new String(Files.readAllBytes(Paths.get(filePath)));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    try {
```

```

        CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
            .name(schemaName)
            .schema(schema)
            .build();

        String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

        System.out.println("Schema arn: " + schemaArn);

        return schemaArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

3. Erstellen Sie eine Dataset-Gruppe. Verwenden Sie die folgende `createDatasetGroup` Methode, um eine Datensatzgruppe zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client und den Namen für die Datensatzgruppe. Die Methode gibt den ARN Ihrer neuen Datensatzgruppe zurück. Speichern Sie es für eine spätere Verwendung.

```

public static String createDatasetGroup(PersonalizeClient personalizeClient,
String datasetGroupName) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .build();

        return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}

```

4. Erstellen eines `createDataset`. Verwenden Sie die folgende `createDataset` Methode, um ein `createDataset` zu erstellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Namen für Ihren Datensatz, den ARN Ihres Schemas, den ARN Ihrer Datensatzgruppe und

Interactions für den Datensatztyp. Die Methode gibt den ARN Ihres neuen Datensatzes zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createDataset(PersonalizeClient personalizeClient,
    String datasetName,
    String datasetGroupArn,
    String datasetType,
    String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
            .name(datasetName)
            .datasetGroupArn(datasetGroupArn)
            .datasetType(datasetType)
            .schemaArn(schemaArn)
            .build();

        String datasetArn = personalizeClient.createDataset(request)
            .datasetArn();
        System.out.println("Dataset " + datasetName + " created.");
        return datasetArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

5. Importieren Sie Ihre Daten mit einem Datensatz-Importauftrag. Verwenden Sie die folgende `createPersonalizeDatasetImportJob` Methode, um einen Datensatz-Importauftrag zu erstellen.

Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, einen Namen für den Auftrag, den ARN Ihres , den Amazon S3-Bucket-Pfad (s3://*bucket name*/*folder name*/ratings.csv), in dem Sie die Trainingsdaten gespeichert haben, und den ARN Ihrer Servicerolle (Sie haben diese Rolle als Teil von erstellt [Voraussetzungen für den Einstieg](#)). Die Methode gibt den ARN Ihres Datensatz-Importauftrags zurück. Speichern Sie es optional zur späteren Verwendung.

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
    personalizeClient,
    String jobName,
```

```
        String datasetArn,
        String s3BucketPath,
        String roleArn) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
        CreateDatasetImportJobRequest.builder()
            .datasetArn(datasetArn)
            .dataSource(importDataSource)
            .jobName(jobName)
            .roleArn(roleArn)
            .build();

        datasetImportJobArn =
        personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
            .datasetImportJobArn();

        DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
        DescribeDatasetImportJobRequest.builder()
            .datasetImportJobArn(datasetImportJobArn)
            .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetImportJob datasetImportJob = personalizeClient
                .describeDatasetImportJob(describeDatasetImportJobRequest)
                .datasetImportJob();

            status = datasetImportJob.status();
            System.out.println("Dataset import job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
        }
    }
}
```

```
        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
```

Schritt 4: Erstellen einer Lösung

Nachdem Sie Ihre Daten importiert haben, erstellen Sie wie folgt eine Lösung und eine Lösungsversion. Die Lösung enthält die Konfigurationen zum Trainieren eines Modells und eine Lösungsversion ist ein trainiertes Modell.

1. Erstellen Sie eine neue Lösung mit der folgenden `createPersonalizeSolution` Methode. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, Ihren Datensatz gruppiert den Amazon-Ressourcennamen (ARN), einen Namen für die Lösung und den ARN für das User-Personalization-Rezept (`arn:aws:personalize:::recipe/aws-user-personalization`). Die Methode gibt den ARN zurück, den Ihre neue Lösung hat. Speichern Sie es für eine spätere Verwendung.

```
public static String createPersonalizeSolution(PersonalizeClient
personalizeClient,
        String datasetGroupArn,
        String solutionName,
        String recipeArn) {

    try {
        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .build();

        CreateSolutionResponse solutionResponse =
personalizeClient.createSolution(solutionRequest);
```

```

        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

- Erstellen Sie eine Lösungsversion mit der folgenden `createPersonalizeSolutionVersion` Methode. Übergeben Sie im vorherigen Schritt als Parameter den ARN der Lösung. Der folgende Code prüft zunächst, ob Ihre Lösung bereit ist, und erstellt dann eine Lösungsversion. Während des Trainings verwendet der Code den [-DescribeSolutionVersion](#) Vorgang, um den Status der Lösungsversion abzurufen. Wenn das Training abgeschlossen ist, gibt die Methode den ARN Ihrer neuen Lösungsversion zurück. Speichern Sie es für eine spätere Verwendung.

```

public static String createPersonalizeSolutionVersion(PersonalizeClient
personalizeClient, String solutionArn) {
    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String solutionStatus = "";
    String solutionVersionStatus = "";
    String solutionVersionArn = "";

    try {
        DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
            .solutionArn(solutionArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        // Wait until solution is active.
        while (Instant.now().getEpochSecond() < maxTime) {

            solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
            System.out.println("Solution status: " + solutionStatus);

            if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
FAILED")) {
                break;
            }
        }
    }
}

```

```
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }

    if (solutionStatus.equals("ACTIVE")) {

        CreateSolutionVersionRequest createSolutionVersionRequest =
        CreateSolutionVersionRequest.builder()
            .solutionArn(solutionArn)
            .build();

        CreateSolutionVersionResponse createSolutionVersionResponse =
        personalizeClient
            .createSolutionVersion(createSolutionVersionRequest);
        solutionVersionArn =
        createSolutionVersionResponse.solutionVersionArn();

        System.out.println("Solution version ARN: " + solutionVersionArn);

        DescribeSolutionVersionRequest describeSolutionVersionRequest =
        DescribeSolutionVersionRequest.builder()
            .solutionVersionArn(solutionVersionArn)
            .build();

        while (Instant.now().getEpochSecond() < maxTime) {

            solutionVersionStatus =
            personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
                .solutionVersion().status();
            System.out.println("Solution version status: " +
            solutionVersionStatus);

            if (solutionVersionStatus.equals("ACTIVE") ||
            solutionVersionStatus.equals("CREATE_FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

```
        }
        return solutionVersionArn;
    }
} catch (PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}
```

Weitere Informationen finden Sie unter [Erstellen einer Lösung und einer Lösungsversion](#). Wenn Sie eine Lösungsversion erstellen, können Sie deren Leistung bewerten, bevor Sie fortfahren. Weitere Informationen finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#).

Schritt 5: Erstellen einer Kampagne

Nachdem Sie Ihre Lösungsversion trainiert und bewertet haben, stellen Sie sie mit einer Amazon-Personalize-Kampagne bereit. Verwenden Sie die folgende `createPersonalCampaign` Methode, um eine Lösungsversion bereitzustellen. Übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Amazon-Ressourcennamen (ARN) der Lösungsversion, die Sie im vorherigen Schritt erstellt haben, und einen Namen für die Kampagne. Die Methode gibt den ARN Ihrer neuen Kampagne zurück. Speichern Sie es für eine spätere Verwendung.

```
public static String createPersonalCampaign(PersonalizeClient personalizeClient, String
    solutionVersionArn, String name) {

    try {
        CreateCampaignRequest createCampaignRequest = CreateCampaignRequest.builder()
            .minProvisionedTPS(1)
            .solutionVersionArn(solutionVersionArn)
            .name(name)
            .build();

        CreateCampaignResponse campaignResponse =
personalizeClient.createCampaign(createCampaignRequest);
        System.out.println("The campaign ARN is "+campaignResponse.campaignArn());
        return campaignResponse.campaignArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```



```
}  
}
```

Weitere Informationen zu Amazon-Personalize-Kampagnen finden Sie unter [Eine Kampagne erstellen](#).

Schritt 6: Abrufen von Empfehlungen

Nachdem Sie eine Kampagne erstellt haben, verwenden Sie sie, um Empfehlungen zu erhalten. Verwenden Sie die folgende `getRecs` Methode, um Empfehlungen für einen Benutzer zu erhalten. Übergeben Sie als Parameter einen Amazon-Personalize-Laufzeit-Client, den Amazon-Ressourcennamen (ARN) der Kampagne, die Sie im vorherigen Schritt erstellt haben, und eine Benutzer-ID (z. B. 123) aus den von Ihnen importierten historischen Daten. Die `-Methode` gibt die Liste der empfohlenen Elemente auf dem Bildschirm aus.

```
public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,  
String campaignArn, String userId) {  
  
    try {  
        GetRecommendationsRequest recommendationsRequest =  
GetRecommendationsRequest.builder()  
            .campaignArn(campaignArn)  
            .numResults(20)  
            .userId(userId)  
            .build();  
  
        GetRecommendationsResponse recommendationsResponse =  
personalizeRuntimeClient  
            .getRecommendations(recommendationsRequest);  
        List<PredictedItem> items = recommendationsResponse.itemList();  
        for (PredictedItem item : items) {  
            System.out.println("Item Id is : " + item.itemId());  
            System.out.println("Item score is : " + item.score());  
        }  
  
    } catch (AwsServiceException e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
}
```

Abschließen des Amazon-Personalize-Projekts

Ein all-in-one Projekt, das Ihnen zeigt, wie Sie den Amazon-Personalize-Workflow mit dem SDK for Java 2.x abschließen, finden Sie in der [Amazon-Personalize-Java-App](#) auf GitHub. Dieses Projekt umfasst das Training mehrerer Lösungsversionen mit unterschiedlichen Rezepten und das Aufzeichnen von Ereignissen mit der `PutEvents` Operation.

Weitere Beispiele finden Sie unter `-Code` im [Personalisieren](#)-Ordner des AWS SDK-Beispiel-Repositorys.

Bereinigen von Ressourcen

Löschen Sie nach Abschluss der Übungen im Rahmen der ersten Schritte die erstellten Ressourcen, um unnötige Gebühren zu vermeiden. Um die Ressourcen zu löschen, verwenden Sie entweder die Amazon Personalize Personalize-Konsole oder die `Delete` APIs aus den SDKs oder AWS Command Line Interface (AWS CLI). Verwenden Sie beispielsweise die API [DeleteCampaign](#) zum Löschen einer Kampagne.

Sie können eine Ressource nicht löschen, deren Status `CREATE PENDING` oder `IN PROGRESS` ist. Der Ressourcenstatus muss `ACTIVE` oder `CREATE FAILED` sein. Überprüfen Sie den Status mithilfe der `Describe`-APIs, z. B. [DescribeCampaign](#).

Einige Ressourcen müssen vor anderen gelöscht werden, wie in der folgenden Tabelle gezeigt. Dieser Vorgang kann einige Zeit in Anspruch nehmen.

Informationen zum Löschen der von Ihnen hochgeladenen Trainingsdaten finden Sie unter [Wie lösche ich Objekte aus einem S3-Bucket? ratings.csv](#).

Themen

- [Säuberung domänenbasierter Ressourcen](#)
- [Bereinigen benutzerdefinierter Ressourcen](#)

Säuberung domänenbasierter Ressourcen

Wenn Sie eine Domain-Datensatzgruppe erstellt haben, löschen Sie Ressourcen wie folgt:

Zu löschende Ressource	Löschen Sie dies zuerst	Hinweise
Recommender		
DatasetImportJob		Kann nicht gelöscht werden.
Dataset		Kein zugehöriger <code>DatasetImportJobs</code> kann den Status <code>CREATE PENDING</code> oder <code>IN PROGRESS</code> haben. Kein zugehöriger <code>Recommenders</code> kann den Status <code>CREATE PENDING</code> oder <code>IN PROGRESS</code> haben.
DatasetSchema	Alle Datensätze, die auf das Schema verweisen.	
DatasetGroup	Alle zugehörigen Empfehlungen Alle Datasets in der Dataset-Gruppe.	

Bereinigen benutzerdefinierter Ressourcen

Wenn Sie eine benutzerdefinierte Datensatzgruppe erstellt haben, löschen Sie Ressourcen wie folgt:

Zu löschende Ressource	Löschen Sie dies zuerst	Hinweise
Campaign		
DatasetImportJob		Kann nicht gelöscht werden.
EventTracker		Das dem Ereignis-Tracker zugeordnete Dataset für Ereignisinteraktionen wird nicht gelöscht und wird von der Lösungsversion weiterhin verwendet.

Zu löschende Ressource	Löschen Sie dies zuerst	Hinweise
Dataset		Kein zugehöriger DatasetJob kann den Status CREATE PENDING oder IN PROGRESS haben. Kein zugehöriger SolutionVersion kann den Status CREATE PENDING oder IN PROGRESS haben.
DatasetSchema	Alle Datensätze, die auf das Schema verweisen.	
Solution	Alle Kampagnen, die auf der Lösungsversion basieren.	Kein zugehöriger SolutionVersion kann den Status CREATE PENDING oder IN PROGRESS haben.
SolutionVersion		Gelöscht, wenn die zugeordnete Solution gelöscht wird.
DatasetGroup	Alle zugehörigen Ereignis-Tracker. Alle zugehörigen Lösungen. Alle Datasets in der Dataset-Gruppe.	

Datensätze und Schemas

Amazon Personalize Personalize-Datensätze sind Container für Daten. Es gibt fünf Arten von Datensätzen:

- [Artikelinteraktionen](#) — Dieser Datensatz speichert historische Daten und Echtzeitdaten aus Interaktionen zwischen Benutzern und Artikeln. In Amazon Personalize ist eine Interaktion ein Ereignis, das Sie aufzeichnen und dann als Trainingsdaten importieren. Sowohl für Domain-Datensatzgruppen als auch für benutzerdefinierte Datensatzgruppen müssen Sie mindestens einen Datensatz mit Artikelinteraktionen erstellen.
- [Benutzer](#) — Dieser Datensatz speichert Metadaten über Ihre Benutzer. Dazu können Informationen wie Alter, Geschlecht, Loyalitätsmitgliedschaft oder Artikeltitel gehören.
- [Artikel](#) — In diesem Datensatz werden Metadaten zu Ihren Artikeln gespeichert. Hierzu können Informationen wie Preis, SKU-Typ oder Verfügbarkeit gehören.
- [Aktionen](#) — In diesem Datensatz werden Metadaten zu Ihren Aktionen gespeichert. Eine Aktion ist eine Interaktionsaktivität, die Sie Ihren Kunden vielleicht empfehlen möchten. Zu den Aktionen können die Installation Ihrer mobilen App, das Ausfüllen eines Mitgliedsprofils, die Teilnahme an Ihrem Treueprogramm oder die Anmeldung für Werbe-E-Mails gehören. Für das Rezept „Next-Best-Action“ ist der Datensatz „Aktionen“ erforderlich. In keinem anderen benutzerdefinierten Rezept- oder Domain-Anwendungsfall werden Actions-Daten verwendet.
- [Interaktionen mit Aktionen](#) — Dieser Datensatz speichert historische Daten und Echtzeitdaten aus Interaktionen zwischen Benutzern und Aktionen. Das Rezept für die nächste Best-Action verwendet diese Daten und die Daten in Ihrem Actions-Datensatz, um Ihren Benutzern Aktionen zu empfehlen. Kein anderes benutzerdefiniertes Rezept oder Domain-Anwendungsfall verwendet Action-Interactions-Daten.

Jede Datensatzgruppe kann nur einen Datensatz von jedem Datensatztyp haben. Amazon Personalize speichert Ihre Daten in Datensätzen, bis Sie die Datensätze löschen. Für alle Anwendungsfälle (Domain-Datensatzgruppen) und Rezepte (benutzerdefinierte Datensatzgruppen) müssen Ihre Interaktionsdaten Folgendes enthalten:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Bevor Sie einen Datensatz erstellen, definieren Sie ein Schema für diesen Datensatz. Ein Schema informiert Amazon Personalize über die Struktur Ihrer Daten und ermöglicht Amazon Personalize, die Daten zu analysieren. Ein Schema hat einen Namensschlüssel, dessen Wert dem Datensatztyp entsprechen muss. Nachdem Sie ein Schema erstellt haben, können Sie keine Änderungen am Schema vornehmen.

Für Domain-Datensatzgruppen hat jeder Datensatztyp ein Standardschema mit erforderlichen Feldern und reservierten Schlüsselwörtern. Jedes Mal, wenn Sie einen Datensatz erstellen, können Sie entweder das vorhandene Domänenschema verwenden oder ein neues erstellen, indem Sie das vorhandene Standardschema ändern. Verwenden Sie das Standardschema als Leitfaden dafür, welche Daten für Ihre Domain importiert werden sollen. Sobald Sie das Schema definiert und den Datensatz erstellt haben, können Sie keine Änderungen mehr am Schema vornehmen.

Wenn Sie Daten in großen Mengen importieren, müssen Ihre Daten im CSV-Format (Comma Separated Values) gespeichert werden. Die erste Zeile Ihrer CSV-Datei muss Spaltenüberschriften enthalten, die Ihrem Schema entsprechen müssen. Informationen zum Formatieren Ihrer Massendaten für Amazon Personalize finden Sie unter [Richtlinien für Datenformate](#).

Themen

- [Datensätze](#)
- [Schemata](#)
- [Richtlinien für Datenformate](#)

Datensätze

Die folgenden Themen enthalten detaillierte Informationen zu Amazon Personalize Personalize-Datensätzen. Jeder Datensatztyp hat unterschiedliche Datenanforderungen. Sowohl für Domain-Datensatzgruppen als auch für benutzerdefinierte Datensatzgruppen müssen Ihre Interaktionsdaten vor dem Training folgende Werte aufweisen:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Wenn Sie eine Domain-Datensatzgruppe erstellen, gelten für jeden Datensatz je nach Domäne zusätzliche Anforderungen. Wenn Sie sich nicht sicher sind, welche Art von Daten Sie benötigen, empfehlen wir, eine Domain-Datensatzgruppe zu erstellen und sich an den Standardschemas für Ihre Domain zu orientieren. Weitere Informationen zu Datensatz- und Schemaanforderungen finden Sie unter [Schemata](#).

Themen

- [Datensatz mit Interaktionen zwischen Elementen](#)
- [Benutzerdatensatz](#)
- [Elementedatensatz](#)
- [Datensatz „Aktionen“](#)
- [Datensatz „Aktionen“: Interaktionen](#)

Datensatz mit Interaktionen zwischen Elementen

Eine Artikelinteraktion ist ein Interaktionsereignis zwischen einem Benutzer und einem Artikel in Ihrem Katalog. Beispiel: Ein Nutzer schaut sich einen Film an, schaut sich ein Angebot an oder kauft ein Paar Schuhe. Sie importieren Daten über die Interaktionen Ihrer Benutzer mit Ihren Artikeln in einen Datensatz mit Artikelinteraktionen. Sie können mehrere Ereignistypen aufzeichnen, z. B. „Klicken“, „Ansehen“ oder „Gefällt mir“.

Wenn ein Benutzer beispielsweise auf einen bestimmten Artikel klickt und den Artikel dann mit „Gefällt mir“ markiert, können Sie Amazon Personalize diese Ereignisse als Trainingsdaten verwenden lassen. Für jedes Ereignis würden Sie die ID des Benutzers, die ID des Artikels, den Zeitstempel (im Unix-Zeitepochenformat) und den Ereignistyp (Klicken und Gefällt mir) aufzeichnen. Anschließend würden Sie beide Artikelinteraktionsereignisse zu einem Datensatz mit Artikelinteraktionen hinzufügen.

Für alle Anwendungsfälle (Domain-Datensatzgruppen) und Rezepte (benutzerdefinierte Ressourcen) müssen Ihre Artikelinteraktionsdaten Folgendes enthalten:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.

- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Um eine Empfehlung oder eine benutzerdefinierte Lösung zu erstellen, müssen Sie mindestens einen Datensatz mit Artikelinteraktionen erstellen. Dieser Abschnitt enthält Informationen zu den folgenden Arten von Artikelinteraktionsdaten, die Sie in Amazon Personalize importieren können.

Themen

- [Daten zu Ereignistyp und Ereigniswert](#)
- [Kontextuelle Metadaten](#)
- [Daten zu Eindrücken](#)

Daten zu Ereignistyp und Ereigniswert

Ein Datensatz mit Artikelinteraktionen kann Ereignistyp- und Ereigniswertdaten für jede Interaktion speichern. Nur benutzerdefinierte Ressourcen verwenden Daten zu Ereigniswerten.

Daten vom Ereignistyp

Amazon Personalize verwendet Ereignistypdaten wie Klick - oder Kaufdaten, um die Absicht und das Interesse der Nutzer zu ermitteln. Wenn Sie Domain-Empfehlungen erstellen, benötigen alle Anwendungsfälle Daten zum Ereignistyp. Für einige Anwendungsfälle sind bestimmte Ereignistypen erforderlich. Es steht Ihnen frei, weitere Ereignistypen zu verwenden. Weitere Informationen finden Sie unter [Auswahl eines Anwendungsfalls](#).

Wenn Sie benutzerdefinierte Ressourcen erstellen, können Sie die für die Schulung verwendeten Ereignisse nach Ereignistyp auswählen. Wenn Ihr Datensatz mehrere Ereignistypen in einer EVENT_TYPE-Spalte enthält und Sie bei der Konfiguration einer benutzerdefinierten Lösung keinen Ereignistyp angeben, verwendet Amazon Personalize alle Artikelinteraktionsdaten für das Training mit gleichem Gewicht, unabhängig vom Typ. Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

Positive und negative Ereignistypen

Amazon Personalize geht davon aus, dass jede Interaktion positiv ist. Interaktionen mit einem negativen Ereignistyp, wie z. B. Abneigung, verhindern nicht unbedingt, dass der Artikel in future Empfehlungen des Benutzers erscheint.

Es gibt folgende Möglichkeiten, wie negative Ereignisse und das Desinteresse der Nutzer die Empfehlungen beeinflussen können:

- Für alle Domain-Anwendungsfälle und das [Benutzerpersonalisierung](#) Rezept kann Amazon Personalize Impressionsdaten verwenden. Wenn ein Artikel in den Impressionsdaten erscheint und ein Benutzer ihn nicht auswählt, ist es weniger wahrscheinlich, dass der Artikel in Empfehlungen erscheint. Weitere Informationen finden Sie unter [Daten zu Eindrücken](#).
- Wenn Sie benutzerdefinierte Ressourcen verwenden und positive und negative Ereignistypen importieren, können Sie nur anhand positiver Ereignistypen trainieren und dann Elemente herausfiltern, mit denen der Nutzer negativ interagiert hat. Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#) und [Filtern von Empfehlungen und Benutzersegmenten](#).

Daten zu Ereigniswerten (benutzerdefinierte Ressourcen)

Bei Daten zum Ereigniswert kann es sich um den Prozentsatz eines Films handeln, den ein Benutzer angesehen hat, oder um eine Bewertung von 10 Punkten. Wenn Sie benutzerdefinierte Lösungen erstellen und Event-Wertdaten zusammen mit Ereignistypdaten importieren, können Sie Datensätze, die für Schulungen verwendet werden, anhand von Typ und Wert auswählen. Bei Domain-Empfehlungen verwendet Amazon Personalize keine Daten zu Ereigniswerten und Sie können Ereignisse nicht vor dem Training filtern.

Um Datensätze nach Typ und Wert auszuwählen, notieren Sie für jedes Ereignis einen Ereignistyp und einen Ereigniswert. Welchen Wert Sie für jedes Ereignis auswählen, hängt davon ab, welche Daten Sie ausschließen möchten und welche Ereignistypen Sie aufzeichnen. Sie können beispielsweise die Benutzeraktivität, z. B. den Prozentsatz der Videos, die der Benutzer angesehen hat, für die Art von Wiedergabeereignissen abgleichen.

Wenn Sie eine Lösung konfigurieren, legen Sie einen bestimmten Wert als Schwellenwert fest, um Datensätze vom Training auszuschließen. Wenn Ihre EVENT_VALUE-Daten für Ereignisse mit dem Wert EVENT_TYPE „Ansehen“ den Prozentsatz eines Videos darstellen, das ein Benutzer angesehen hat, und Sie den Schwellenwert für den Ereigniswert auf 0,5 und den anzuzeigenden Ereignistyp festlegen, trainiert Amazon Personalize das Modell, indem es nur Interaktionsereignisse verwendet, deren EVENT_VALUE größer oder gleich 0,5 ist.

Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

Kontextuelle Metadaten

Bei bestimmten Rezepten und Anwendungsfällen für Empfehlungen kann Amazon Personalize kontextbezogene Metadaten verwenden, um zugrunde liegende Muster zu identifizieren, die die relevantesten Elemente für Ihre Benutzer aufdecken. Kontextuelle Metadaten sind Interaktionsdaten, die Sie zum Zeitpunkt eines Ereignisses in der Umgebung des Benutzers sammeln, z. B. dessen Standort oder Gerätetyp.

Durch die Einbeziehung kontextbezogener Metadaten können Sie bestehenden Benutzern ein persönlicheres Erlebnis bieten. Wenn Kunden beispielsweise anders einkaufen, wenn sie über ein Telefon auf Ihren Katalog zugreifen, als über einen Computer, sollten Sie kontextbezogene Metadaten zum Gerät des Benutzers hinzufügen. Empfehlungen werden dann je nach Art und Weise, wie sie surfen, relevanter sein.

Darüber hinaus tragen kontextbezogene Metadaten dazu bei, die Kaltstartphase für neue oder unbekannte Nutzer zu verkürzen. Die Kaltstartphase bezieht sich auf den Zeitraum, in dem Ihre Empfehlungsmaschine aufgrund fehlender historischer Informationen zu diesem Benutzer weniger relevante Empfehlungen ausgibt.

Für Domain-Datensatzgruppen können in den folgenden Anwendungsfällen für Empfehlungen kontextbezogene Metadaten verwendet werden:

- [Empfohlen für Sie](#)(E-COMMERCE-Domäne)
- [Top-Auswahlen für Sie](#)(Domäne VIDEO_ON_DEMAND)

Für benutzerdefinierte Ressourcen beinhalten Rezepte, die kontextuelle Metadaten verwenden, Folgendes:

- [Benutzerpersonalisierung](#)
- [Personalisierte Rangfolge](#)

Weitere Informationen zu Kontextinformationen finden Sie im folgenden Blogbeitrag zum AWS Machine Learning: [Erhöhung der Relevanz Ihrer Amazon Personalize Personalize-Empfehlungen durch die Nutzung von](#) Kontextinformationen.

Daten zu Eindrücken

Wenn Sie eine Domain-Datensatzgruppe für die Domain VIDEO_ON_DEMAND oder E-COMMERCE erstellen oder das [Benutzerpersonalisierung](#) Rezept verwenden, kann Amazon Personalize

Impressionsdaten modellieren, die Sie in einen Datensatz mit Artikelinteraktionen hochladen. Impressions sind Listen von Elementen, die für einen Benutzer sichtbar waren, als er mit einem bestimmten Objekt interagiert (z. B. darauf klickte oder es sich ansah).

Amazon Personalize verwendet Impressionsdaten, um zu bestimmen, welche Artikel in die Erkundung einbezogen werden sollen. Bei der Erkundung umfassen die Empfehlungen einige Elemente oder Aktionen, bei denen die Wahrscheinlichkeit, dass sie dem Benutzer empfohlen werden, in der Regel geringer ist, z. B. neue Artikel oder Aktionen, Artikel oder Aktionen mit wenigen Interaktionen oder Artikel oder Aktionen, die für den Benutzer aufgrund seines vorherigen Verhaltens weniger relevant sind. Je häufiger ein Artikel in den Impressionsdaten vorkommt, desto unwahrscheinlicher ist es, dass Amazon Personalize den Artikel in die Suche einbezieht. Impressionswerte können maximal 1000 Zeichen lang sein (einschließlich des senkrechten Balkenzeichens).

Weitere Informationen zur Erkundung finden Sie unter [Exploration \(Erkundung\)](#). Amazon Personalize kann zwei Arten von Impressions modellieren: [Implizite Impressions](#) und [Explizite Eindrücke](#).

Implizite Impressions

Implizite Impressions sind die von Amazon Personalize abgerufenen Empfehlungen, die Sie dem Benutzer zeigen. Sie können sie in Ihren Empfehlungs-Workflow integrieren, indem Sie die `RecommendationId` (von den [GetPersonalizedRanking](#) Operationen [GetRecommendations](#) und zurückgegebenen) als Eingabe für future [PutEvents](#) Anfragen verwenden. Amazon Personalize leitet die impliziten Impressions auf der Grundlage Ihrer Empfehlungsdaten ab.

Möglicherweise haben Sie eine Anwendung, die Empfehlungen für das Streamen von Videos bereitstellt. Ihr Empfehlungs-Workflow, der implizite Impressions verwendet, könnte wie folgt aussehen:

1. Sie fordern mithilfe des Amazon Personalize [the section called “GetRecommendations”](#) Personalize-API-Vorgangs Videoempfehlungen für einen Ihrer Benutzer an.
2. Amazon Personalize generiert Empfehlungen für den Benutzer, der Ihr Modell (Lösungsversion) verwendet, und gibt sie mit einer Antwort `recommendationId` in der API zurück.
3. Sie zeigen Ihrem Benutzer die Videoempfehlungen in Ihrer Anwendung.
4. Wenn Ihr Benutzer mit einem Video interagiert (z. B. klickt), zeichnen Sie die Auswahl in einem [PutEvents](#) API-Aufruf auf und fügen Sie das `recommendationId` als Parameter hinzu. Ein Codebeispiel finden Sie unter [Aufzeichnen von Impressionsdaten](#).

5. Amazon Personalize verwendet die `recommendationId`, um die Impressionsdaten aus den vorherigen Videoempfehlungen abzuleiten, und verwendet dann die Impressionsdaten als Leitfaden für die Erkundung, wobei future Empfehlungen neue Videos mit weniger Interaktionsdaten oder Relevanz beinhalten.

Weitere Informationen zur Aufzeichnung von Ereignissen mit impliziten Impressionsdaten finden Sie unter [Aufzeichnen von Impressionsdaten](#)

Explizite Eindrücke

Explizite Impressions sind Impressions, die Sie manuell aufzeichnen und an Amazon Personalize senden. Verwenden Sie explizite Impressions, um die Ergebnisse von Amazon Personalize zu manipulieren. Die Reihenfolge der Artikel hat keine Auswirkung.

Möglicherweise haben Sie eine Einkaufsanwendung, die Empfehlungen für Schuhe bietet. Wenn Sie nur Schuhe empfehlen, die derzeit auf Lager sind, können Sie diese Artikel anhand expliziter Impressions angeben. Ihr Empfehlungs-Workflow mit expliziten Impressions könnte wie folgt aussehen:

1. Sie fordern mithilfe der Amazon Personalize [the section called "GetRecommendations"](#) Personalize-API Empfehlungen für einen Ihrer Benutzer an.
2. Amazon Personalize generiert Empfehlungen für den Benutzer, der Ihr Modell (Lösungsversion) verwendet, und gibt sie in der API-Antwort zurück.
3. Sie zeigen dem Benutzer nur die empfohlenen Schuhe, die auf Lager sind.
4. Beim inkrementellen Datenimport in Echtzeit zeichnen Sie, wenn Ihr Benutzer mit einem Paar Schuhen interagiert (z. B. klickt), die Auswahl in einem [PutEvents](#) API-Aufruf auf und listen die empfohlenen Artikel, die auf Lager sind, im `impression` Parameter auf. Ein Codebeispiel finden Sie unter [Aufzeichnen von Impressionsdaten](#)

Um Impressions in historische Artikelinteraktionsdaten zu importieren, können Sie explizite Impressions in Ihrer CSV-Datei auflisten und jedes Element durch ein '|' trennen. Das vertikale Balkenzeichen wird auf die Obergrenze von 1000 Zeichen angerechnet. Ein Beispiel finden Sie unter [Formatieren von expliziten Eindrücken](#).

5. Amazon Personalize verwendet die Impressionsdaten als Leitfaden für die Erkundung, wobei future Empfehlungen neue Schuhe mit weniger Interaktionsdaten oder Relevanz beinhalten.

Benutzerdatensatz

Zu den Benutzerdaten, die Sie in Amazon Personalize importieren können, gehören numerische und kategoriale Metadaten über Ihre Benutzer, wie Geschlecht oder Loyalitätsmitgliedschaft. Sie importieren Metadaten über Ihre Benutzer in einen Amazon Personalize Users-Datensatz. Die maximale Anzahl von Metadatenfeldern beträgt 25.

Dieses Thema enthält Informationen zu den folgenden Arten von Benutzerdaten:

Themen

- [Kategorische Metadaten](#)

Kategorische Metadaten

Bei einigen Rezepten und sowohl bei VIDEO_ON_DEMAND- als auch bei E-COMMERCE-Domains verwendet Amazon Personalize kategoriale Metadaten, wie z. B. das Geschlecht oder den Mitgliedschaftsstatus eines Benutzers, um zugrunde liegende Muster zu identifizieren, die die relevantesten Artikel für Ihre Benutzer aufdecken. Sie definieren Ihren eigenen Wertebereich auf der Grundlage Ihres Anwendungsfalls. Kategorische Metadaten können in jeder Sprache verfasst sein.

Bei allen Rezepten und Domänen können Sie kategoriale Metadaten importieren und sie verwenden, um Empfehlungen anhand der Attribute eines Benutzers zu filtern. Informationen zum Filtern von Empfehlungen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Kategoriale Werte können maximal 1000 Zeichen lang sein. Wenn Sie einen Benutzer mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Für benutzerdefinierte Datensatzgruppen und benutzerdefinierte Lösungen umfassen Rezepte, die kategoriale Metadaten verwenden, Folgendes:

- [Benutzerpersonalisierung](#)
- [Personalisierte Rangfolge](#)
- [Ähnliche Elemente](#)

Elementedatensatz

Zu den Artikeldaten, die Sie in Amazon Personalize importieren können, gehören numerische und kategoriale Metadaten wie Erstellungszeitstempel, Preis, Genre, Beschreibung und Verfügbarkeit. Sie importieren Metadaten zu Ihren Artikeln in einen Amazon Personalize Items-Datensatz.

Amazon Personalize verwendet beim Training keine unkategorischen Zeichenkettendaten wie Artikeltitle oder Autoren Daten. Einige Funktionen von Amazon Personalize verwenden diese Daten jedoch, um Empfehlungen zu verbessern. Weitere Informationen finden Sie unter [Nicht kategorische Zeichenkettendaten](#).

Die maximale Anzahl von Metadaten Spalten beträgt 100. Die maximale Anzahl von Elementen, die von einem Modell während des Trainings berücksichtigt werden, beträgt 750.000. Amazon Personalize berücksichtigt diese Artikel nur bei der Generierung von Empfehlungen. Für einige Domains und Rezepte ist ein Artikeldatensatz erforderlich. Weitere Informationen zu Rezeptanforderungen finden Sie unter [Ein Rezept wählen](#).

Dieses Thema enthält Informationen zu den folgenden Arten von Artikeldaten:

Themen

- [Zeitstempeldaten der Erstellung](#)
- [Kategorische Metadaten](#)
- [Unstrukturierte Textmetadaten](#)
- [Nicht kategorische Zeichenkettendaten](#)

Zeitstempeldaten der Erstellung

Amazon Personalize verwendet Zeitstempeldaten der Erstellung (im Zeitformat Unix-Epoche, in Sekunden), um das Alter eines Artikels zu berechnen und die Empfehlungen entsprechend anzupassen.

Wenn Daten zum Erstellungszeitstempel für einen oder mehrere Artikel fehlen, leitet Amazon Personalize diese Informationen aus etwaigen Interaktionsdaten ab und verwendet den Zeitstempel der ältesten Interaktionsdaten des Artikels als Erstellungszeitstempel des Artikels. Wenn ein Artikel keine Interaktionsdaten hat, wird sein Erstellungszeitstempel als Zeitstempel der letzten Interaktion im Trainingssatz festgelegt und Amazon Personalize betrachtet ihn als neuen Artikel.

Kategorische Metadaten

Bei bestimmten Rezepten und Domains verwendet Amazon Personalize kategorische Metadaten, wie z. B. das Genre oder die Farbe eines Artikels, um zugrunde liegende Muster zu identifizieren, anhand derer die relevantesten Artikel für Ihre Benutzer sichtbar werden. Sie definieren Ihren eigenen Wertebereich auf der Grundlage Ihres Anwendungsfalls. Kategorische Metadaten können in jeder Sprache verfasst sein.

Bei allen Rezepten und Domänen können Sie kategoriale Daten importieren und sie verwenden, um Empfehlungen auf der Grundlage der Attribute eines Artikels zu filtern. Informationen zum Filtern von Empfehlungen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Kategorische Werte können maximal 1000 Zeichen lang sein. Wenn Sie ein Element mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Für Domain-Datensatzgruppen verwenden sowohl die Domänen VIDEO_ON_DEMAND als auch E-COMMERCE kategoriale Metadaten. Für benutzerdefinierte Datensatzgruppen und benutzerdefinierte Lösungen umfassen Rezepte, die kategoriale Metadaten verwenden, Folgendes:

- [Benutzerpersonalisierung](#)
- [Personalisierte Rangfolge](#)
- [Ähnliche Elemente](#)
- [Element-Affinität](#)
- [Element-Attribut-Affinität](#)

Unstrukturierte Textmetadaten

Bei bestimmten Rezepten und Domains kann Amazon Personalize aussagekräftige Informationen aus unstrukturierten Textmetadaten wie Produktbeschreibungen, Produktrezensionen oder Filmzusammenfassungen extrahieren. Amazon Personalize verwendet unstrukturierten Text, um relevante Artikel für Ihre Benutzer zu identifizieren, insbesondere wenn Artikel neu sind oder weniger Interaktionsdaten enthalten. Nehmen Sie unstrukturierte Textdaten in Ihren Artikeldatensatz auf, um die Klickraten und Konversationsraten für neue Artikel in Ihrem Katalog zu erhöhen.

Um unstrukturierte Daten zu verwenden, fügen Sie Ihrem Artikelschema ein Feld mit Typ `string` hinzu und legen Sie das Feldattribut auf `textual true` fest. Nehmen Sie dann die Textdaten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Setzen Sie den Text bei CSV-Massendateien in doppelte Anführungszeichen. Verwenden Sie das `\` Zeichen, um doppelte Anführungszeichen

oder \-Zeichen in Ihren Daten zu maskieren. Sie können höchstens ein Textfeld hinzufügen. Ein Beispiel für ein Artikelschema mit einem Feld für unstrukturierte Textdaten finden Sie unter [Beispiel für ein Artikel-Dataset-Schema \(benutzerdefiniert\)](#) Amazon Personalize schneidet Textfelder bis zur Zeichenbeschränkung ab. Stellen Sie sicher, dass sich die relevantesten Informationen im Text am Anfang des Felds befinden. Informationen zum Importieren von Daten in Amazon Personalize finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Vor der Verwendung unstrukturierter Textwerte entfernt Amazon Personalize Folgendes aus dem Text:

- HTML- und XML-Tags und -Entitäten
- Neue Zeilen-, Tabulatorzeichen und zusätzliche Leerzeichen

Unstrukturierte Textwerte können in allen Sprachen außer Chinesisch und Japanisch höchstens 20.000 Zeichen enthalten. Für Chinesisch und Japanisch können Sie maximal 7.000 Zeichen verwenden. Amazon Personalize kürzt Werte, die die Zeichenbeschränkung überschreiten, auf die Zeichenbeschränkung.

Text kann in den folgenden Sprachen verfasst werden:

- Chinesisch (vereinfacht)
- Chinesisch (traditionell)
- Englisch
- Französisch
- Deutsch
- Japanisch
- Portugiesisch
- Spanisch

Sie können unstrukturierte Textelemente in mehreren Sprachen einreichen, aber der Text jedes Elements sollte nur in einer Sprache verfasst sein.

Für Domain-Datensatzgruppen verwenden sowohl die Domänen VIDEO_ON_DEMAND als auch E-COMMERCE Textmetadaten. Für benutzerdefinierte Datensatzgruppen und benutzerdefinierte Lösungen umfassen Rezepte, die Textmetadaten verwenden, Folgendes:

- [Benutzerpersonalisierung](#)

- [Personalisierte Rangfolge](#)
- [Ähnliche Elemente](#)
- [Element-Affinität](#)
- [Element-Attribut-Affinität](#)

Nicht kategorische Zeichenkettendaten

Mit Ausnahme von Artikel-IDs verwendet Amazon Personalize beim Training keine nicht kategorischen Zeichenkettendaten, wie Artikeltitel oder Autoren Daten. Amazon Personalize kann es jedoch mit den folgenden Funktionen verwenden:

- Amazon Personalize kann Artikelmetadaten in Empfehlungen aufnehmen, einschließlich nicht kategorischer Zeichenkettenwerte. Sie können Metadaten verwenden, um Empfehlungen auf Ihrer Benutzeroberfläche zu erweitern, z. B. um den Namen des Regisseurs zu einem Karussell mit Filmempfehlungen hinzuzufügen. Weitere Informationen finden Sie unter [Metadaten in Empfehlungen aktivieren](#).
- Wenn Sie dies verwenden [Ähnliche Elemente](#), können Sie Batch-Empfehlungen mit Themen generieren. Wenn Sie Batch-Empfehlungen mit Themen generieren, müssen Sie im Batch-Inferenzjob eine Spalte mit dem Elementnamen angeben. Weitere Informationen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).
- Sie können Filter erstellen, um Elemente auf der Grundlage von Zeichenkettendaten, die nicht kategorisch sind, in Empfehlungen aufzunehmen oder daraus zu entfernen. Weitere Informationen zu Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Datensatz „Aktionen“

Eine Aktion ist eine Aktivität, die Interaktionen oder Umsätze generiert und die Sie Ihren Nutzern empfehlen möchten. Zu den Aktionen können die Installation Ihrer mobilen App, das Ausfüllen eines Mitgliedsprofils, die Teilnahme an Ihrem Treueprogramm oder die Anmeldung für Werbe-E-Mails gehören. Sie importieren Daten über Ihre Aktionen in einen Amazon Personalize Actions-Datensatz. Zu den Daten für eine Aktion gehören beispielsweise eine ID für die Aktion, der geschätzte Wert der Aktion oder der Ablaufzeitstempel der Aktion.

Während des Modelltrainings berücksichtigt Amazon Personalize maximal 1000 Aktionen. Wenn Sie mehr als 1000 Aktionen importieren, entscheidet Amazon Personalize, welche Aktionen in die Schulung aufgenommen werden sollen, wobei neuen Aktionen (Aktionen, die Sie kürzlich ohne

Interaktionen hinzugefügt haben) und bestehenden Aktionen mit Daten zu aktuellen Interaktionen Priorität eingeräumt wird.

Die maximale Anzahl von Spalten ist 10. Dieses Thema enthält Informationen zu den folgenden Typen von Aktionsdaten:

Themen

- [Daten über den Zeitstempel des Ablaufs der Aktion](#)
- [Häufigkeitsdaten wiederholen](#)
- [Wertdaten](#)
- [Daten mit Zeitstempel der Erstellung](#)
- [Kategorische Metadaten](#)

Daten über den Zeitstempel des Ablaufs der Aktion

Ein Ablaufzeitstempel für eine Aktion gibt das Datum an, an dem eine Aktion nicht mehr gültig ist. Sie geben Zeitstempeldaten für den Ablauf einer Aktion im Zeitformat Unix-Epoche in Sekunden an. Wenn eine Aktion abgelaufen ist, nimmt Amazon Personalize sie nicht in Empfehlungen auf.

Geben Sie einen Ablaufzeitstempel für Ihre Aktionen an, wenn Sie deren Auftreten in Empfehlungen auf einen bestimmten Zeitraum beschränken möchten. Angenommen, Sie haben eine Anwendung, die während eines bestimmten Monats eine Mitgliederaktion durchführt. Sie könnten einen Ablaufzeitstempel für die Anmeldeaktion für das Ende dieses Monats festlegen. Amazon Personalize hört automatisch auf, diese Aktion zu empfehlen, wenn dieses Datum erreicht ist.

Wenn Sie den Ablaufzeitstempel für eine neue Aktion auf einen Zeitpunkt in der Vergangenheit setzen oder wenn Sie den Zeitstempel einer Aktion auf einen Zeitpunkt in der Vergangenheit aktualisieren, kann es bis zu 2 Stunden dauern, bis die Aktion aus den Empfehlungen entfernt ist.

Häufigkeitsdaten wiederholen

Die Daten zur Wiederholungshäufigkeit geben an, wie viele Tage Amazon Personalize warten sollte, um eine bestimmte Aktion zu empfehlen, nachdem ein Benutzer damit interagiert hat. Dies basiert auf der Historie des Benutzers in Ihrem Datensatz „Aktionsinteraktionen“. Sie geben die Wiederholungshäufigkeit einer Aktion in Tagen mit einem Maximum von 30 an.

Angenommen, Sie haben eine E-Commerce-Anwendung, bei der jeder Benutzer ein Konto und ein Profil erstellt. Wenn Sie eine `complete profile` Aktion haben und eine Woche warten möchten,

nachdem ein Benutzer mit ihr interagiert hat, bevor Sie sie erneut empfehlen, würden Sie für die Aktion 7 Tage angeben. REPEAT_FREQUENCY Nach 7 Tagen beginnt Amazon Personalize, die Aktion für Empfehlungen in Betracht zu ziehen.

Wenn Sie keine Wiederholungshäufigkeit für eine Aktion angeben, begrenzt Amazon Personalize nicht, wie oft sie in Empfehlungen erscheint.

Wertdaten

Wertdaten sind der Geschäftswert oder die Bedeutung jeder Aktion. Der Wert einer Aktion `value` kann zwischen 1 und 10 liegen, wobei 10 die wertvollste Aktion in Ihrem Datensatz ist.

Beispielsweise könnten Sie zwei Aktionen haben, eine für die Registrierung für Ihr Basisabonnement und eine für die Registrierung für Ihren Premium-Service. Für den Basisdienst können Sie einen Wert von 5 und für den Premium-Dienst einen Wert von 10 angeben.

Amazon Personalize verwendet Wertdaten als eine Eingabe bei der Bestimmung der besten Aktion, die Sie Ihren Benutzern empfehlen können. Wenn ein Benutzer beispielsweise die gleiche Wahrscheinlichkeit hat, die eine oder andere Aktion auszuführen, stuft Amazon Personalize die Aktion mit dem höchsten Wert in Bezug auf Empfehlungen höher ein.

Daten mit Zeitstempel der Erstellung

Amazon Personalize verwendet Zeitstempeldaten der Erstellung (im Zeitformat Unix-Epoche, in Sekunden), um das Alter einer Aktion zu berechnen und die Empfehlungen entsprechend anzupassen.

Wenn Sie nicht über Zeitstempeldaten für die Erstellung verfügen, leitet Amazon Personalize diese Informationen aus allen Aktionsinteraktionsdaten ab. Es verwendet den Zeitstempel der ältesten Interaktionsdaten der Aktion als Zeitstempel für die Erstellung der Aktion. Wenn eine Aktion keine Interaktionsdaten hat, wird ihr Erstellungszeitstempel als Zeitstempel der letzten Interaktion im Trainingssatz festgelegt, und Amazon Personalize betrachtet sie als neue Aktion.

Kategorische Metadaten

Amazon Personalize verwendet kategorische Metadaten zu Aktionen, wie Saisonalität oder Aktionsexklusivität, um die zugrunde liegenden Muster zu identifizieren, die die besten Aktionen für Ihre Benutzer aufzeigen. Sie definieren Ihren eigenen Wertebereich auf der Grundlage Ihres Anwendungsfalls. Kategorische Metadaten können in jeder Sprache verfasst sein.

Sie können kategoriale Daten importieren und sie verwenden, um Empfehlungen auf der Grundlage der Attribute einer Aktion zu filtern. Informationen zum Filtern von Empfehlungen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Kategorische Werte können maximal 1000 Zeichen lang sein. Wenn Sie eine Aktion mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Datensatz „Aktionen“: Interaktionen

Eine Aktionsinteraktion ist eine Interaktion, an der ein Benutzer und eine Aktion in Ihrem [Aktionsdatensatz](#) beteiligt sind. Sie importieren Aktionsinteraktionen in einen Amazon Personalize Action-Interaktions-Datensatz. Jede Aktionsinteraktion besteht aus einer Benutzer-ID, einer ActionID, einem Zeitstempel, einem Ereignistyp und allen zusätzlichen Daten über die Interaktion, wie z. B. kategorische Metadaten.

Wenn Sie beispielsweise eine Anmeldeaktion in Ihrem Aktionen-Datensatz haben und ein Benutzer diese Aktion ausführt, würden Sie die Benutzer-ID, die ID der Aktion, den Zeitstempel und für den Ereignistyp den Datensatz aufzeichnen. TAKEN Sie können Aktionsereignisse im Rahmen eines Datensatz-Import-Jobs in großen Mengen importieren oder sie mit dem [PutActionInteractions](#) API-Vorgang in Echtzeit streamen. Weitere Informationen zum Importieren von Daten finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Wenn Sie ein benutzerdefiniertes Rezept für PERSONALIZED_ACTIONS verwenden, verwendet Amazon Personalize alle Daten in Ihrem Aktionsinteraktionen-Datensatz als Eingabe, um die Aktionen vorherzusagen, die Ihre Benutzer höchstwahrscheinlich ergreifen werden. Es gibt keine Mindestanforderung für Daten zu Aktionsinteraktionen. Wir empfehlen, dass Sie diese Daten importieren, um qualitativ hochwertige Handlungsempfehlungen zu erhalten. Wenn Sie keine Daten zu Aktionsinteraktionen haben, können Sie einen leeren Datensatz für Aktionsinteraktionen erstellen und mithilfe der [PutActionInteractions](#) API-Operation die Interaktionen Ihrer Kunden mit Aktionen aufzeichnen.

Die folgenden Themen enthalten weitere Informationen zu den Aktionsinteraktionsdaten, die Amazon Personalize verwenden kann.

Themen

- [Daten vom Ereignistyp](#)

Daten vom Ereignistyp

Amazon Personalize kann Muster in Ereignistypdaten verwenden, um die Aktionen zu identifizieren, die Ihre Benutzer höchstwahrscheinlich ergreifen werden. Wenn ein Kunde beispielsweise häufig eine E-Mail-Abonnementaktion ignoriert (gekennzeichnet mit dem Ereignistyp NOT_TAKEN), passt Amazon Personalize die Empfehlungen möglicherweise an, sodass weniger dieser Aktionen angeboten werden.

Sie können nur die folgenden Ereignistypen für Aktionsereignisse verwenden. Amazon Personalize verwendet diese Ereignisse, um mehr über Ihren Benutzer zu erfahren und zu berechnen, welche Aktionen als Nächstes empfohlen werden sollten.

- **Ergriffen** — Erledigte Ereignisse aufzeichnen, wenn ein Benutzer eine empfohlene Maßnahme ergreift.
- **Nicht ausgeführt** — Ereignisse, die nicht ausgeführt wurden, aufzeichnen, wenn Ihr Benutzer sich bewusst dafür entscheidet, die Aktion nicht auszuführen, nachdem er sich die Aktion angesehen hat. Zum Beispiel, wenn sie „Nein“ wählen, wenn Sie ihnen die Aktion zeigen. Ereignisse, die nicht durchgeführt wurden, können darauf hinweisen, dass der Kunde an der Aktion nicht interessiert ist.
- **Angesehen** — Angesehene Ereignisse aufzeichnen, wenn Sie einem Benutzer eine Aktion zeigen, bevor er sich entscheidet, ob er eine Aktion ausführt oder nicht. Amazon Personalize verwendet View-Ereignisse, um mehr über die Interessen Ihrer Benutzer zu erfahren. Wenn ein Benutzer beispielsweise eine Aktion betrachtet, sie aber nicht ausführt, ist dieser Benutzer möglicherweise in future nicht an dieser Aktion interessiert.

Schemata

Ein Schema informiert Amazon Personalize über die Struktur Ihrer Daten und ermöglicht Amazon Personalize, die Daten zu analysieren. Ein Schema hat einen Namensschlüssel, dessen Wert dem Datensatztyp entsprechen muss. Nachdem Sie ein Schema erstellt haben, können Sie keine Änderungen am Schema vornehmen.

Für Domain-Datensatzgruppen hat jeder Datensatztyp ein Standardschema mit erforderlichen Feldern und reservierten Schlüsselwörtern. Jedes Mal, wenn Sie einen Datensatz erstellen, können Sie entweder das vorhandene Domänenschema verwenden oder ein neues erstellen, indem Sie das vorhandene Standardschema ändern. Verwenden Sie das Standardschema als Leitfaden dafür,

welche Daten für Ihre Domain importiert werden sollen. Sobald Sie das Schema definiert und den Datensatz erstellt haben, können Sie keine Änderungen mehr am Schema vornehmen.

Themen

- [Anforderungen an die Schemaformatierung](#)
- [Domain-Datasets und Schemas](#)
- [Benutzerdefinierte Datensätze und Schemas](#)
- [Erstellen eines Schemas mit SDK for Python \(Boto3\)](#)

Anforderungen an die Schemaformatierung

Wenn Sie ein Schema für einen Datensatz in einer Domain-Datensatzgruppe oder einer benutzerdefinierten Datensatzgruppe erstellen, müssen Sie die folgenden Richtlinien beachten:

- Sie müssen das Schema im [Avro-Format](#) definieren. Informationen zu den Avro-Datentypen, die wir unterstützen, finden Sie unter [Schema-Datentypen](#)
- Die Schemafelder können in beliebiger Reihenfolge angezeigt werden, müssen jedoch der Reihenfolge der entsprechenden Spaltenüberschriften in Ihrer CSV-Datei entsprechen.
- Schemas müssen flache JSON-Dateien ohne verschachtelte Strukturen sein. Beispielsweise kann ein Feld nicht mehreren Unterfeldern übergeordnet sein.
- Amazon Personalize Personalize-Schemas unterstützen keine komplexen Typen wie Arrays und Maps.
- Schemafelder müssen eindeutige alphanumerische Namen haben. Sie können beispielsweise nicht gleichzeitig ein `GENRES_FIELD_1` Feld und ein `GENRESFIELD1` Feld hinzufügen.
- Sie müssen Pflichtfelder als ihre erforderlichen Datentypen definieren. Für reservierte kategoriale Zeichenfolgenfelder muss das `categorical` Attribut auf `gesetzt` `true` sein, während reservierte Zeichenkettenfelder nicht kategorisch sein können. Die Schlüsselwörter dürfen nicht in Ihren Daten enthalten sein.
- Wenn Sie Ihr eigenes Metadatenfeld vom Typ hinzufügen `string` und möchten, dass Amazon Personalize es beim Training verwendet, muss es das `categorical` Attribut oder das `textual` Attribut enthalten (nur Artikelschemas unterstützen Felder mit dem Textattribut).
- Amazon Personalize kann beim Generieren von Themen, beim Zurückgeben von Metadaten in Empfehlungen und beim Filtern von Empfehlungen unkategorielle Zeichenfolgenspalten verwenden, z. B. Spalten mit Artikelnamen. Weitere Informationen finden Sie unter [Nicht kategorische Zeichenkettendaten](#).

- Amazon Personalize verwendet beim Training oder Filtern von Empfehlungen keine boolean Typdaten. Damit Amazon Personalize beim Training oder Filtern boolesche Daten verwendet, verwenden Sie ein Feld vom Typ String und verwenden Sie die Werte "True" und "False" in Ihren Daten. Oder Sie können den Typ int oder long und die Werte 0 und 1 verwenden.
- Textfelder müssen vom Typ sein string und das textual Attribut muss auf gesetzt sein true. Weitere Hinweise zu unstrukturierten Textdaten finden Sie unter [Unstrukturierte Textmetadaten](#)

Für Datensätze mit Domänen-Datensatzgruppen gelten zusätzliche Anforderungen, die sowohl auf der Domäne als auch auf dem Datensatztyp basieren. Für Datensätze mit benutzerdefinierten Datensatzgruppen gelten je nach Typ zusätzliche Anforderungen.

Schema-Datentypen

Amazon Personalize Personalize-Schemas unterstützen die folgenden Avro-Typen für Felder:

- float
- double
- int
- long
- Zeichenfolge
- Boolesch
- Null

Einige erforderliche und reservierte Felder unterstützen Nulldaten. Wenn Sie einem Feld einen null Typ hinzufügen, können Sie unvollständige Daten (z. B. Metadaten mit leeren Werten) verwenden, um Empfehlungen zu generieren. Informationen darüber, welche Felder Nulldaten unterstützen, finden Sie unter [Domain-Datasets und Schemas](#) oder [Benutzerdefinierte Datensätze und Schemas](#). Das folgende Beispiel zeigt, wie ein Nulltyp für ein GENDER-Feld hinzugefügt wird.

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
}
```

}

Domain-Datasets und Schemas

Wenn Sie eine Domain-Datensatzgruppe erstellen, bestimmt die von Ihnen gewählte Domain Ihre Datensatz- und Schemaanforderungen. Jede Domain hat ein Standardschema für jeden Datensatztyp.

Wenn Sie einen Datensatz erstellen, können Sie entweder das Standardschema verwenden oder ein neues erstellen, das auf dem Standardschema basiert. Verwenden Sie das Standardschema als Leitfaden dafür, welche Daten gesammelt und in die einzelnen Datensatztypen importiert werden sollen. In den folgenden Themen werden die Anforderungen an Datensätze und Schemas für die einzelnen Domänen erläutert.

Informationen zu den Datentypen, die Sie in Amazon Personalize importieren können, finden Sie unter [Datentypen, die Amazon Personalize verwenden kann](#).

Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Themen

- [VIDEO_ON_DEMAND-Datensätze und Schemas](#)
- [E-COMMERCE-Datensätze und -Schemas](#)

VIDEO_ON_DEMAND-Datensätze und Schemas

Wenn Sie eine Domain-Datensatzgruppe für die Domäne VIDEO_ON_DEMAND erstellen, verfügt jeder Datensatztyp über ein Standardschema mit einer Reihe von für VIDEO_ON_DEMAND spezifischen erforderlichen und empfohlenen Feldern. Sie können entweder das Standardschema verwenden oder ein neues Schema auf der Grundlage des Standardschemas erstellen. Die Daten, die Sie importieren, müssen in Format und Typ Ihrem Schema entsprechen. Verwenden Sie die in den folgenden Abschnitten aufgeführten Standard-Domänenschemas als Leitfaden, um zu bestimmen, welche Daten importiert werden müssen, um Ihre Video_on_Demand-basierte Empfehlung zu erstellen.

Es steht Ihnen frei, weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), sind die Feldnamen und Datentypen Ihre Entscheidung.

Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Die folgenden Themen enthalten Informationen zu den erforderlichen und empfohlenen Feldern der einzelnen Datensätze für die Domäne VIDEO_ON_DEMAND. Jeder Datensatzabschnitt enthält das Standardschema VIDEO_ON_DEMAND im JSON-Format.

Themen

- [Anforderungen an den Domänendatensatz und das Schema für VIDEO_ON_DEMAND](#)
- [Anforderungen an den Datensatz für Artikelinteraktionen \(Domäne VIDEO_ON_DEMAND\)](#)
- [Anforderungen an Benutzerdatensätze \(Domäne VIDEO_ON_DEMAND\)](#)
- [Anforderungen an den Artikeldatensatz \(Domäne VIDEO_ON_DEMAND\)](#)

Anforderungen an den Domänendatensatz und das Schema für VIDEO_ON_DEMAND

Jeder Datensatztyp hat die folgenden Pflichtfelder und reservierten Schlüsselwörter. Reservierte Schlüsselwörter sind optionale Felder, die keine Metadaten sind. Diese Felder gelten als reserviert, da Sie die Felder bei ihrer Verwendung als erforderlichen Datentyp definieren müssen. Reservierte kategoriale Zeichenfolgenfelder müssen auf `categorical` gesetzt sein `true`, reservierte Zeichenfolgenfelder können dagegen nicht kategorisch sein. Die Schlüsselwörter dürfen nicht in Ihren Daten enthalten sein.

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
Interaktionen mit Elementen (Standardschema)	USER_ID (string) ITEM_ID (string) TIMESTAMP (long) EVENT_TYPE (string und je nach Anwendungsfall und Watch Click Ereignistypen)	EVENT_VALUE (,) float null EINDRUCK (,) string null EMPFEHLUNG_ID (,) string null QUELLE DER EREIGNISZ UORDNUNG (,) string null
Benutzer (Standardschema)	USER_ID (string)	SUBSCRIPTION_MODEL (kategorisch,) string null

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
	1 Metadatenfeld (kategorisch string oder numerisch)	
Artikel (Standardschema)	ITEM_ID (string) CREATION_TIMESTAMP () long GENRES (kategorisch) string	PREIS (float,null) DAUER (float,null) GENRE_L2 (kategorisch,) string null GENRE_L3 (kategorisch,) string null DURCHSCHNITTLICHE BEWERTUNG float (null,) PRODUKTBESCHREIBUNG (textuell,) string null CONTENT_OWNER (kategorisch,) string null CONTENT_CLASSIFICATION (kategorisch,) string null

Anforderungen an den Datensatz für Artikelinteraktionen (Domäne VIDEO_ON_DEMAND)

Ein Datensatz mit Artikelinteraktionen speichert historische Daten und Echtzeitdaten aus Interaktionen zwischen Benutzern und Artikeln in Ihrem VIDEO_ON_DEMAND-Katalog. Weitere Informationen zu den Datentypen, die Sie in einem Interaktionsdatensatz speichern können, finden Sie unter [Datensatz mit Interaktionen zwischen Elementen](#)

Sie benötigen einen Datensatz mit Artikelinteraktionen für alle Anwendungsfälle, und Ihr Schema muss die folgenden Felder enthalten:

- USER_ID (string)
- ITEM_ID string

- `TIMESTAMP` (long)
- `EVENT_TYPE` (string und je nach [Anwendungsfall](#) und Ereignistypen) Watch Click

Ihr Schema kann auch die folgenden reservierten Schlüsselwörter enthalten:

- `EVENT_VALUE` (float,) null
- `EINDRUCK` (,) string null
- `EMPFEHLUNG_ID` (,) string null

Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen. Ein Beispiel für das Standardschema für Datensätze mit Artikelinteraktionen für VIDEO_ON_DEMAND-Domänen finden Sie unter [Standardschema für Interaktionen \(Domäne VIDEO_ON_DEMAND\)](#).

Fügen Sie optional das reservierte Schlüsselwort `EVENT_VALUE` hinzu, wenn Sie über Wertdaten für Ereignisse verfügen, z. B. den Prozentsatz der angesehenen Videos. Fügen Sie optional das reservierte Schlüsselwort `IMPRESSION` hinzu, wenn Sie explizite und implizite Impressionsdaten einbeziehen möchten. Weitere Informationen zum Aufzeichnen von Impressionsdaten finden Sie unter [Daten zu Eindrücken](#).

Die maximale Gesamtzahl optionaler Metadatenfelder, die Sie einem Datensatz mit Artikelinteraktionen hinzufügen können, zusammen mit der Gesamtzahl der unterschiedlichen Ereignistypen in Ihren Artikelinteraktionsdaten, beträgt 10. Die in dieser Anzahl enthaltenen Metadatenfelder sind die Felder `EVENT_TYPE` und `EVENT_VALUE` sowie alle benutzerdefinierten Metadatenfelder, die Sie Ihrem Schema hinzufügen. Die maximale Anzahl von Metadatenfeldern mit Ausnahme von reservierten Feldern wie `IMPRESSION` beträgt 5. Kategorische Werte können maximal 1000 Zeichen lang sein. Wenn Sie eine Interaktion mit einem kategorialen Wert mit mehr als 1000 haben, schlägt Ihr Datensatz-Importjob fehl.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Datensatz mit Artikelinteraktionen für die Domäne VIDEO_ON_DEMAND finden Sie unter [Service Quotas](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Standardschema für Interaktionen (Domäne VIDEO_ON_DEMAND)

Das Folgende ist das Standarddomänenschema VIDEO_ON_DEMAND für Datensätze mit Artikelinteraktionen.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Anforderungen an Benutzerdatensätze (Domäne VIDEO_ON_DEMAND)

Ein Benutzerdatensatz speichert Metadaten über Ihre Benutzer. Dies kann Informationen wie Alter, Geschlecht und Loyalitätsmitgliedschaft für jeden Artikel beinhalten. Informationen zu den Arten von Benutzerdaten, die Sie in Amazon Personalize importieren können, finden Sie unter [Benutzerdatensatz](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Ein Benutzerdatensatz ist für alle VIDEO_ON_DEMAND-Anwendungsfälle optional. Wenn Sie über Benutzerdaten verfügen, empfehlen wir, solche zu erstellen, um die relevantesten Empfehlungen

zu erhalten. Wenn Sie einen Benutzerdatensatz erstellen, muss Ihr Schema die folgenden Felder enthalten.

- USER_ID
- 1 Metadatenfeld (kategorisch `string` oder numerisch)

Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen. Ein Beispiel für das Standardschema für Benutzerdatensätze für VIDEO_ON_DEMAND-Domänen finden Sie unter [Standardbenutzerschema \(Domäne VIDEO_ON_DEMAND\)](#)

Ein SUBSCRIPTION_MODEL Feld ist im Standardschema enthalten. Dieses Feld ist ein optionales reserviertes Schlüsselwort und muss den Typ haben, `string` wobei kategorisch auf `true` gesetzt sein muss. Um die besten Empfehlungen zu erhalten, empfehlen wir, dieses Feld in Ihrem Schema beizubehalten, wenn Sie Informationen zum Abonnementmodell für jeden Ihrer Benutzer in Ihren Daten haben. Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen.

Verwenden von kategorialen Daten

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ hinzu `string` und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Bei Benutzern mit mehreren Kategorien trennen Sie die einzelnen Werte mithilfe des vertikalen Balkens '|' voneinander ab. Für ein Feld SUBSCRIPTION_MODEL könnten Ihre Daten für einen Benutzer beispielsweise `student|monthly|discount discount`.

Kategoriale Werte können maximal 1000 Zeichen enthalten. Wenn Sie einen Benutzer mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Standardbenutzerschema (Domäne VIDEO_ON_DEMAND)

Das Folgende ist das Standarddomänenschema VIDEO_ON_DEMAND für Benutzerdatensätze.

```
{  
  
  "type": "record",  
  "name": "Users",  
  "namespace": "com.amazonaws.personalize.schema",  
  "fields": [  

```

```
{
  {
    "name": "USER_ID",
    "type": "string"
  },
  {
    "name": "SUBSCRIPTION_MODEL",
    "type": "string",
    "categorical": true
  }
],
"version": "1.0"
}
```

Anforderungen an den Artikeldatensatz (Domäne VIDEO_ON_DEMAND)

Ein Artikeldatensatz speichert Metadaten zu Ihren Artikeln in Ihrem Katalog. Dies kann Informationen wie Preis, Genre und Verfügbarkeit für jeden Artikel beinhalten. Informationen zu den Artikeldatentypen, die Sie in Amazon Personalize importieren können, finden Sie unter [Elementedatensatz](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Für einige Anwendungsfälle ist ein Artikeldatensatz erforderlich (siehe [Anwendungsfälle für VIDEO_ON_DEMAND](#)). Falls optional, empfehlen wir dennoch, einen zu erstellen, um die relevantesten Empfehlungen zu erhalten. Wenn Sie einen Artikeldatensatz erstellen, muss Ihr Schema die folgenden Felder enthalten:

- ITEM_ID
- GENRES (kategorischstring)
- CREATION_TIMESTAMP (im Zeitformat für die Unix-Epoche)

Ihr Schema kann auch die folgenden reservierten Schlüsselwörter enthalten. Jedes Schlüsselwort listet den erforderlichen Datentyp auf und gibt an, ob es Nulldaten unterstützt. Das Hinzufügen des Nulltyps ist optional.

- PREIS (Float)
- DAUER (Float)
- GENRE_L2 (kategorisch,) string null
- GENRE_L3 (kategorisch,) string null

- DURCHSCHNITTLICHE BEWERTUNG float (null,)
- PRODUKTBESCHREIBUNG (textuell,) string null
- CONTENT_OWNER (kategorischstring,null): Das Unternehmen, dem das Video gehört. Die Werte könnten beispielsweise HBO, Paramount und NBC lauten.
- CONTENT_CLASSIFICATION (kategorischstring,null): Die Bewertung des Inhalts. Die Werte könnten beispielsweise G, PG, PG-13, R, NC-17 und nicht bewertet sein.

Um die besten Empfehlungen zu erhalten, empfehlen wir, dass Sie so viele dieser Felder in Ihrem Schema behalten, wie Sie Daten haben. Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen. Die maximale Anzahl von Metadaten spalten beträgt 100. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen.

Verwenden Sie reservierte Schlüsselwörter `GENRE_L2` und `GENRE_L3` für Elemente mit mehreren Kategorien mit mehreren Ebenen. Weitere Informationen finden Sie unter [Verwendung von kategorialen Daten](#). Informationen zu textuellen und kategorialen Metadaten finden Sie unter [Elementedatensatz](#). Ein Beispiel für das Standardschema für Artikel-Datensätze für E-COMMERCE-Domänen finden Sie unter [Schema für Standardelemente \(Domäne VIDEO_ON_DEMAND\)](#)

Verwendung von kategorialen Daten

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ hinzu `string` und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Kategoriewerte können maximal 1000 Zeichen lang sein. Wenn Sie ein Element mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Bei Elementen mit mehreren Kategorien trennen Sie die einzelnen Werte durch den senkrechten Balken `|` voneinander. Für ein `GENRES`-Feld könnten Ihre Daten für ein Element beispielsweise `Action|Crime|Biopic` Wenn Sie über mehrere Ebenen von kategorialen Daten verfügen und einige Elemente mehrere Kategorien für jede Ebene in der Hierarchie haben, fügen Sie für jede Ebene ein Feld hinzu und fügen Sie hinter jedem Feldnamen einen Ebenenindikator an: `GENRES`, `GENRE_L2`, `GENRE_L3`. Auf diese Weise können Sie Empfehlungen auf der Grundlage von Unterkategorien filtern, auch wenn ein Element zu mehreren Kategorien mit mehreren Ebenen gehört. Ein Video könnte beispielsweise die folgenden Daten für jede Kategorieebene enthalten:

- `GENRES`: Aktion | Abenteuer

- GENRE_L2: Krimi | Western
- GENRE_L3: Biopic

In diesem Beispiel handelt es sich bei dem Video um Action > Verbrechen > Biopic-Hierarchie und Abenteuer > Western > Biopic-Hierarchie. Wir empfehlen, nur bis zu L3 zu verwenden, aber Sie können bei Bedarf auch mehr Stufen verwenden. Informationen zum Erstellen und Verwenden von Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Schema für Standardelemente (Domäne VIDEO_ON_DEMAND)

Das Folgende ist das Standardschema für Artikel-Datasets für die Domäne VIDEO_ON_DEMAND.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

E-COMMERCE-Datensätze und -Schemas

Wenn Sie eine Domain-Datensatzgruppe für die E-COMMERCE-Domain erstellen, hat jeder Datensatztyp ein Standardschema mit einer Reihe von E-Commerce-spezifischen erforderlichen und empfohlenen Feldern. Sie können das Standardschema verwenden oder ein neues erstellen,

das auf dem Standardschema basiert. Die Daten, die Sie importieren, müssen in Format und Typ Ihrem Schema entsprechen. Verwenden Sie die in den folgenden Abschnitten aufgeführten Standard-Domain-Schemas als Leitfaden, um zu bestimmen, welche Daten Sie importieren müssen, um Ihre E-Commerce-basierte Empfehlung zu erstellen.

Es steht Ihnen frei, weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), sind die Feldnamen und Datentypen Ihre Entscheidung.

Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Die folgenden Themen enthalten Informationen zu den erforderlichen und empfohlenen Feldern der einzelnen Datensätze für die E-COMMERCE-Domäne. Jeder Datensatzabschnitt enthält das Standard-E-COMMERCE-Schema im JSON-Format.

Themen

- [Anforderungen an den Datensatz und das Schema der E-COMMERCE-Domain](#)
- [Anforderungen an den Datensatz für Artikelinteraktionen \(E-COMMERCE-Domain\)](#)
- [Anforderungen an Benutzerdatensätze \(E-COMMERCE-Domäne\)](#)
- [Anforderungen an Artikeldatensätze \(E-COMMERCE-Domäne\)](#)

Anforderungen an den Datensatz und das Schema der E-COMMERCE-Domain

Jeder Datensatztyp hat die folgenden Pflichtfelder und reservierten Schlüsselwörter. Reservierte Schlüsselwörter sind optionale Felder, die keine Metadaten sind. Diese Felder gelten als reserviert, da Sie die Felder bei ihrer Verwendung als erforderlichen Datentyp definieren müssen. Reservierte kategoriale Zeichenfolgenfelder müssen auf `categorical` gesetzt sein `true`, reservierte Zeichenfolgenfelder können dagegen nicht kategorisch sein. Die Schlüsselwörter dürfen nicht in Ihren Daten enthalten sein.

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
Interaktionen mit Elementen (Standardschema)	USER_ID (string) ITEM_ID (string)	EVENT_VALUE (,) float null EINDRUCK (,) string null

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
	TIMESTAMP (long) EVENT_TYPE (string und je nach Anwendungsfall und Purchase View Ereignistypen)	EMPFEHLUNG_ID (,) string null QUELLE DER EREIGNISZUORDNUNG (,) string null
Benutzer (Standardschema)	USER_ID (string) 1 Metadatenfeld (kategorisch string oder numerisch)	
Elemente (Standardschema)	ITEM_ID (string) PREIS (float) KATEGORIE_L1 (kategorisch) string	CATEGORY_L2 (kategorisch,) string null CATEGORY_L3 (kategorisch,) string null PRODUKTBESCHREIBUNG (textuell,) string null ERSTELLUNGSZEITSTEMPEL (,) long AGE_GROUP (kategorisch,) string null ERWACHSENE (kategorisch,) string null GESCHLECHT (kategorischstring,) null

Anforderungen an den Datensatz für Artikelinteraktionen (E-COMMERCE-Domain)

Ein Datensatz mit Artikelinteraktionen speichert historische Daten und Echtzeitdaten aus Interaktionen zwischen Benutzern und Artikeln in Ihrem E-COMMERCE-Katalog. Weitere Informationen zu den Datentypen, die Sie in einem Interaktions-Datensatz speichern können,

finden Sie unter [Datensatz mit Interaktionen zwischen Elementen](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Sie müssen mindestens einen Datensatz mit Artikelinteraktionen erstellen und Ihr Schema muss die folgenden Felder enthalten:

- USER_ID (string)
- ITEM_ID (string)
- TIMESTAMP (long)
- EVENT_TYPE (string und je nach [Anwendungsfall](#) und Purchase View Ereignistypen)

Ihr Schema kann auch die folgenden reservierten Schlüsselwörter enthalten:

- EVENT_VALUE (float,) null
- EINDRUCK (,) string null
- EMPFEHLUNG_ID (,) string null

Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen. Ein Beispiel für das Standardschema für Datensätze mit Artikelinteraktionen für E-COMMERCE-Domänen finden Sie unter [Standardschema für Interaktionen \(E-COMMERCE-Domäne\)](#).

Fügen Sie optional das reservierte Schlüsselwort EVENT_VALUE hinzu, wenn Sie Wertdaten für Ereignisse haben. Fügen Sie optional das reservierte Schlüsselwort IMPRESSION hinzu, wenn Sie explizite und implizite Impressionsdaten einbeziehen möchten. Weitere Informationen zum Aufzeichnen von Impressionsdaten finden Sie unter [Daten zu Eindrücken](#).

Die maximale Gesamtzahl optionaler Metadatenfelder, die Sie einem Datensatz mit Artikelinteraktionen hinzufügen können, zusammen mit der Gesamtzahl der unterschiedlichen Ereignistypen in Ihren Artikelinteraktionsdaten, beträgt 10. Die in dieser Anzahl enthaltenen Metadatenfelder sind die Felder EVENT_TYPE und EVENT_VALUE sowie alle benutzerdefinierten Metadatenfelder, die Sie Ihrem Schema hinzufügen. Die maximale Anzahl von Metadatenfeldern mit Ausnahme von reservierten Feldern wie IMPRESSION beträgt 5. Kategorische Werte können

maximal 1000 Zeichen lang sein. Wenn Sie eine Interaktion mit einem kategorialen Wert mit mehr als 1000 haben, schlägt Ihr Datensatz-Importjob fehl.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Datensatz mit Artikelinteraktionen für die E-COMMERCE-Domäne finden Sie unter [Service Quotas](#).

Standardschema für Interaktionen (E-COMMERCE-Domäne)

Das Folgende ist das Standard-E-Commerce-Domänenschema für Artikelinteraktionsdatensätze.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Anforderungen an Benutzerdatensätze (E-COMMERCE-Domäne)

Ein Benutzerdatensatz speichert Metadaten über Ihre Benutzer. Dies kann Informationen wie Alter, Geschlecht und Loyalitätsmitgliedschaft für jeden Benutzer beinhalten. Weitere Informationen zu den Arten von Benutzerdaten, die Sie in Amazon Personalize importieren können, finden Sie unter [Benutzerdatensatz](#). Informationen zu allgemeinen Amazon Personalize Personalize-

Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Ein Benutzerdatensatz ist für alle E-COMMERCE-Anwendungsfälle optional. Wenn Sie über Benutzerdaten verfügen, empfehlen wir, diese zu erstellen, um die relevantesten Empfehlungen zu erhalten. Wenn Sie einen Benutzerdatensatz erstellen, muss Ihr Schema die folgenden Felder enthalten.

- USER_ID
- 1 Metadatenfeld (kategorisch `string` oder numerisch)

Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen. Ein Beispiel für das Standardschema für Benutzerdatensätze für E-COMMERCE-Domänen finden Sie unter [Standardbenutzerschema \(E-COMMERCE-Domäne\)](#).

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Benutzerdatensatz finden Sie unter [Service Quotas](#).

Verwenden von kategorialen Daten

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ hinzu `string` und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Bei Benutzern mit mehreren Kategorien trennen Sie die einzelnen Werte mithilfe des vertikalen Balkens `|` voneinander ab. Für ein Feld `SUBSCRIPTION_MODEL` könnten Ihre Daten für einen Benutzer beispielsweise `student|monthly|discount discount`.

Kategoriale Werte können maximal 1000 Zeichen enthalten. Wenn Sie einen Benutzer mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Standardbenutzerschema (E-COMMERCE-Domäne)

Im Folgenden finden Sie das Standard-E-Commerce-Domänenschema für Benutzerdatensätze mit einem `CATEGORY`-Feld als erforderlichem Metadatenfeld.

```
{
```

```
"type": "record",
"name": "Users",
"namespace": "com.amazonaws.personalize.schema",
"fields": [
  {
    "name": "USER_ID",
    "type": "string"
  },
  {
    "name": "MEMBERSHIP_STATUS",
    "type": "string",
    "categorical": true
  }
],
"version": "1.0"
}
```

Anforderungen an Artikeldatensätze (E-COMMERCE-Domäne)

In einem Artikeldatensatz werden Metadaten zu Ihren E-COMMERCE-Artikeln gespeichert. Dies kann Informationen wie Preis, Kategorie und Produktbeschreibung für jeden Artikel beinhalten. Weitere Informationen zu den Artikeldatentypen, die Sie in Amazon Personalize importieren können, finden Sie unter [Elementedatensatz](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Schemas, unabhängig von der Domäne.

Ein Artikeldatensatz ist für alle E-COMMERCE-Anwendungsfälle optional. Wenn Sie Artikeldaten haben, empfehlen wir, einen zu erstellen, um die relevantesten Empfehlungen zu erhalten. Wenn Sie einen Artikeldatensatz erstellen, muss Ihr Schema die folgenden Felder enthalten:

- ITEM_ID
- PREIS (float)
- KATEGORIE_L1 (kategorisch) string

Ihr Schema kann auch die folgenden reservierten Schlüsselwörter enthalten. Für kategoriale Felder können Sie je nach Anwendungsfall Ihren eigenen Wertebereich definieren.

- CATEGORY_L2 (kategorisch,) string null
- CATEGORY_L3 (kategorisch,) string null

- **PRODUKTBESCHREIBUNG** (textuell,) `string null`
- **ERSTELLUNGSZEITSTEMPEL** () `float`
- **AGE_GROUP** (kategorisch`string,null`): Die Altersgruppe, für die der Artikel bestimmt ist. Werte können Neugeborene, Säuglinge, Kinder und Erwachsene sein.
- **ERWACHSENE** (kategorisch`string,null`): Gibt an, ob der Artikel ausschließlich Erwachsenen vorbehalten ist, z. B. Alkohol. Die Werte können ja oder nein sein.
- **GESCHLECHT** (kategorisch`string,null`): Das Geschlecht, für das der Artikel bestimmt ist. Die Werte können männlich, weiblich und unisex sein.

Um die besten Empfehlungen zu erhalten, empfehlen wir, dass Sie so viele dieser Felder in Ihrem Schema behalten, wie Sie Daten haben. Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen. Die Daten, die Sie importieren, müssen Ihrem Schema entsprechen. Die maximale Anzahl von Metadaten spalten beträgt 100. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen.

Verwenden Sie reservierte Schlüsselwörter `CATEGORY_L2` und `CATEGORY_L3` für Artikel mit mehreren Kategorien auf mehreren Ebenen. Weitere Informationen finden Sie unter [Verwendung von kategorialen Daten](#). Informationen zu textuellen und kategorialen Metadaten finden Sie unter [Unstrukturierte Textmetadaten](#). Ein Beispiel für das Standardschema für Artikel-Datensätze für E-COMMERCE-Domänen finden Sie unter [Standardschema für Artikel \(E-COMMERCE-Domäne\)](#)

Verwendung von kategorialen Daten

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ `string` hinzu und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Sie können Ihren eigenen Wertebereich auf der Grundlage Ihres Anwendungsfalls definieren. Kategorische Werte können maximal 1000 Zeichen enthalten. Wenn Sie ein Element mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Bei Elementen mit mehreren Kategorien trennen Sie die einzelnen Werte durch den senkrechten Balken `|` voneinander. Für ein `CATEGORY_L1`-Feld könnten Ihre Daten für ein Element beispielsweise wie folgt lauten. `Electronics|Productivity|Mouse` Wenn Sie über mehrere Ebenen von kategorialen Daten verfügen und einige Elemente mehrere Kategorien für jede Ebene in der Hierarchie haben, fügen Sie für jede Ebene ein Feld hinzu und fügen Sie hinter jedem Feldnamen

einen Ebenenindikator an: CATEGORY_L1, CATEGORY_L2, CATEGORY_L3. Auf diese Weise können Sie Empfehlungen auf der Grundlage von Unterkategorien filtern, auch wenn ein Element zu mehreren Kategorien mit mehreren Ebenen gehört. Ein Artikel könnte beispielsweise die folgenden Daten für jede Kategorieebene enthalten:

- CATEGORY_L1: Elektronik|Produktivität
- CATEGORY_L2: Produktivität|Computer
- CATEGORY_L3: Maus

In diesem Beispiel befindet sich das Element in der Hierarchie Elektronik > Produktivität > Maus und in der Hierarchie Produktivität > Computer > Maus. Wir empfehlen, nur bis zu L3 zu verwenden, aber Sie können bei Bedarf auch mehr Stufen verwenden. Informationen zum Erstellen und Verwenden von Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Standardschema für Artikel (E-COMMERCE-Domäne)

Das Folgende ist das Standardschema für Artikel-Datensätze für die E-COMMERCE-Domäne mit nur den erforderlichen Feldern.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "PRICE",
      "type": "float"
    },
    {
      "name": "CATEGORY_L1",
      "type": [
        "string"
      ],
      "categorical": true
    }
  ],
  "version": "1.0"
}
```


}

Benutzerdefinierte Datensätze und Schemas

Wenn Sie eine benutzerdefinierte Datensatzgruppe erstellen, erstellen Sie Ihre eigenen Schemas von Grund auf neu. Datensätze und Schemas für benutzerdefinierte Datensatzgruppen haben weniger Pflichtfelder und mehr Flexibilität. In den folgenden Themen werden das Schema und die Datenanforderungen für Datensätze einer benutzerdefinierten Datensatzgruppe erläutert. Jeder Datensatzabschnitt listet die erforderlichen Daten für den Datensatztyp auf und enthält ein JSON-Beispiel für ein Schema.

Informationen zu den Datentypen, die Sie in Amazon Personalize importieren können, finden Sie unter [Datensätze](#). Informationen zu allgemeinen Amazon Personalize Personalize-Schemaanforderungen, wie Formatierungsanforderungen und verfügbaren Felddatentypen, finden Sie unter [Schemata](#). Diese Anforderungen gelten für alle Amazon Personalize Personalize-Schemas.

Themen

- [Anforderungen an benutzerdefinierte Datensätze und Schemas](#)
- [Schemaanforderungen für Datensätze mit Artikelinteraktionen \(benutzerdefiniert\)](#)
- [Anforderungen an das Datensatz-Schema der Benutzer \(benutzerdefiniert\)](#)
- [Anforderungen an das Artikeldatensatz-Schema \(benutzerdefiniert\)](#)
- [Anforderungen an das Aktions-Dataset-Schema \(benutzerdefiniert\)](#)
- [Schemaanforderungen für den Datensatz „Aktionsinteraktionen“ \(benutzerdefiniert\)](#)

Anforderungen an benutzerdefinierte Datensätze und Schemas

Wenn Sie einen Datensatz für eine benutzerdefinierte Datensatzgruppe erstellen, hat jeder Datensatztyp die folgenden Pflichtfelder und reservierten Schlüsselwörter mit den erforderlichen Datentypen.

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
Interaktionen mit Elementen (Beispiel für ein Schema)	USER_ID (string) ITEM_ID (string) TIMESTAMP (long)	EVENT_TYPE (string) EREIGNISWERT (float,) null

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
		EINDRUCK (,) string null EMPFEHLUNG_ID (,) string null QUELLE DER EREIGNISZ UORDNUNG (,) string null
Benutzer (Schemabeispiel)	USER_ID (string) 1 Metadatenfeld (kategorisch string oder numerisch)	
Elemente (Beispiel für ein Schema)	ITEM_ID (string) 1 Metadatenfeld (Kategorie- oder string Textfeld oder numerisches Feld)	CREATION_TIMESTAMP () long
Aktionen (Schemabeispiel)	ACTION_ID () string	ERSTELLUNGSZEITSTEMPEL () long WERT (,) long null GEBEN SIE (string,null) EIN EXPIRATION_TIMESTAMP (,) long null WIEDERHOLUNGSHÄUFIGKEIT (long,null)

Dataset-Typ	Pflichtfelder	Reservierte Schlüsselwörter
Interaktionen mit Aktionen (Beispiel für ein Schema)	USER_ID (string) ACTION_ID () string EVENT_TYPE (string) TIMESTAMP (long)	EINDRUCK (,) string null EMPFEHLUNG_ID (,) string null

Metadaten-Felder

Metadaten enthalten Felder mit oder ohne Zeichenfolge, die nicht erforderlich sind oder kein reserviertes Schlüsselwort verwenden. Für Metadatenschemas gelten die folgenden Einschränkungen:

- Für Benutzer- und Artikelschemas ist mindestens ein Metadatenfeld erforderlich.
- Sie können maximal 25 Metadatenfelder für ein Benutzerschema, 100 Metadatenfelder für ein Artikelschema und 10 Metadatenfelder für ein Aktionsschema hinzufügen.
- Wenn Sie ein eigenes Metadatenfeld vom Typ `hinzufügenstring`, muss es das `categorical` Attribut oder das `textual` Attribut enthalten (nur Artikelschemas unterstützen Felder mit dem Textattribut). Andernfalls verwendet Amazon Personalize das Feld beim Trainieren eines Modells nicht.

Reservierte Schlüsselwörter

Reservierte Schlüsselwörter sind optionale Felder, die keine Metadaten sind. Diese Felder gelten als reserviert, da Sie die Felder bei ihrer Verwendung als erforderlichen Datentyp definieren müssen und die Schlüsselwörter nicht als Werte in Ihren Daten verwendet werden können. Reservierte kategoriale Zeichenfolgenfelder müssen auf `categorical` gesetzt sein `true`, während reservierte Zeichenfolgenfelder nicht kategorisch sein können. Die folgenden Schlüsselwörter sind reserviert:

- `EVENT_TYPE`: Verwenden Sie für Datensätze mit Artikelinteraktionen mit einem oder mehreren Ereignistypen, z. B. sowohl Klicken als auch Herunterladen, ein Feld. `EVENT_TYPE` Sie müssen ein `EVENT_TYPE`-Feld als ein definieren `string` und darf nicht als kategorial festgelegt werden.

- **EVENT_VALUE**: Verwenden Sie für Datensätze mit Artikelinteraktionen, die Wertdaten für Ereignisse enthalten, z. B. den Prozentsatz eines Videos, das ein Benutzer angesehen hat, ein Feld mit dem Typ `optional. EVENT_VALUE float null`
- **CREATION_TIMESTAMP**: Verwenden Sie für Datensätze vom Typ „Elemente“ oder „Aktionen“ mit einem Zeitstempel für das Erstellungsdatum jedes Elements ein Feld mit einem Typ `CREATION_TIMESTAMP long`. Amazon Personalize verwendet `CREATION_TIMESTAMP` Daten, um das Alter eines Artikels zu berechnen und die Empfehlungen entsprechend anzupassen. Siehe [Zeitstempeldaten der Erstellung](#).
- **IMPRESSION**: Verwenden Sie für Datensätze mit Artikelinteraktionen mit expliziten Impressionsdaten ein `IMPRESSION` Feld mit Typ `String` und optional Typ `null`. Impressions sind Listen von Elementen, die für einen Benutzer sichtbar waren, als er mit einem bestimmten Element interagiert (z. B. darauf klickte oder es sich ansah). Weitere Informationen finden Sie unter [Daten zu Eindrücken](#).
- **RECOMMENDATION_ID**: Verwenden Sie für Datensätze mit Artikelinteraktionen, die frühere Empfehlungen als implizite Impressionsdaten verwenden, optional ein `RECOMMENDATION_ID` Feld mit Typ `String` und optional Typ `null`.

Sie müssen kein `RECOMMENDATION_ID` Feld für Amazon Personalize hinzufügen, um implizite Impressions bei der Generierung von Empfehlungen zu verwenden. Sie können eine `recommendationId` [PutEvents](#) Operation auch ohne sie übergeben. Weitere Informationen finden Sie unter [Daten zu Eindrücken](#).

- **WERT**: Wenn Sie für Aktions-Datasets wertvolle Daten für einige oder alle Ihrer Aktionen haben, fügen Sie Ihrem Schema ein `VALUE` Feld hinzu. Verwenden Sie als Typ `long` und geben Sie optional ein `null`. Weitere Hinweise zu Aktionen und ihrem Wert finden Sie unter [Wertdaten](#).
- **ACTION_EXPIRATION_TIMESTAMP**: Wenn Sie für Aktions-Datasets einen Ablaufzeitstempel für einige oder alle Ihrer Aktionen haben, fügen Sie Ihrem Schema ein Feld hinzu. `ACTION_EXPIRATION_TIMESTAMP` Verwenden Sie als Typ `long` und geben Sie optional den Typ ein. `long null` Weitere Hinweise zu Ablaufzeitstempeln finden Sie unter [Daten über den Zeitstempel des Ablaufs der Aktion](#).
- **REPEAT_FREQUENCY**: Wenn Sie für Aktions-Datasets Daten zur Wiederholungshäufigkeit einiger oder aller Ihrer Aktionen haben, fügen Sie Ihrem Schema ein `REPEAT_FREQUENCY` Feld hinzu. Verwenden Sie `long` für seinen Typ und geben Sie optional den Typ ein. `null` Weitere Hinweise zu Daten zur Wiederholungshäufigkeit finden Sie unter [Häufigkeitsdaten wiederholen](#).

Schemaanforderungen für Datensätze mit Artikelinteraktionen (benutzerdefiniert)

Ein Datensatz mit Artikelinteraktionen speichert historische Daten und Echtzeitdaten aus Interaktionen zwischen Benutzern und Artikeln in Ihrem Katalog. Informationen zu den Arten von Interaktionsdaten, die Amazon Personalize verwenden kann, finden Sie unter [Datensatz mit Interaktionen zwischen Elementen](#).

Die Daten, die Sie für jede Interaktion angeben, müssen Ihrem Schema entsprechen. Abhängig von Ihrem Schema können Interaktionsmetadaten leere/Nullwerte enthalten. Sie müssen für jede Interaktion mindestens Folgendes angeben:

- Benutzer-ID
- Artikel-ID
- Zeitstempel (im Zeitformat für die Unix-Epoche)

Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen.

Die maximale Gesamtzahl optionaler Metadatenfelder, die Sie einem Datensatz mit Artikelinteraktionen hinzufügen können, zusammen mit der Gesamtzahl der unterschiedlichen Ereignistypen in Ihren Artikelinteraktionsdaten, beträgt 10. Die in dieser Anzahl enthaltenen Metadatenfelder sind die Felder `EVENT_TYPE` und `EVENT_VALUE` sowie alle benutzerdefinierten Metadatenfelder, die Sie Ihrem Schema hinzufügen. Die maximale Anzahl von Metadatenfeldern mit Ausnahme von reservierten Feldern wie `IMPRESSION` beträgt 5. Kategorische Werte können maximal 1000 Zeichen lang sein. Wenn Sie eine Interaktion mit einem kategorialen Wert mit mehr als 1000 haben, schlägt Ihr Datensatz-Importjob fehl.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Datensatz mit Artikelinteraktionen finden Sie unter [Service Quotas](#).

Beispiel für ein Interaktionsschema (benutzerdefiniert)

Das folgende Beispiel zeigt ein Schema für einen Datensatz mit Artikelinteraktionen. Die Felder `USER_ID`, `ITEM_ID` und `TIMESTAMP` sind erforderlich. Die `IMPRESSION` Felder `EVENT_TYPE` und `EVENT_VALUE`, sind optionale reservierte Schlüsselwörter, die von Amazon Personalize erkannt werden. `EVENT_TYPE` muss vom Typ „Zeichenfolge“ sein und darf nicht kategorisch sein. `LOCATION` und `DEVICE` sind optionale kontextuelle Metadatenfelder. Informationen

zu Schemaanforderungen finden Sie unter [Anforderungen an benutzerdefinierte Datensätze und Schemas](#)

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": [
        "float",
        "null"
      ]
    },
    {
      "name": "LOCATION",
      "type": "string",
      "categorical": true
    },
    {
      "name": "DEVICE",
      "type": [
        "string",
        "null"
      ],
      "categorical": true
    },
    {
      "name": "TIMESTAMP",
```

```

        "type": "long"
    },
    {
        "name": "IMPRESSION",
        "type": "string"
    }
],
"version": "1.0"
}

```

Für dieses Schema könnten die ersten Zeilen mit historischen Daten in einer CSV-Datei wie folgt aussehen. Beachten Sie, dass einige Werte für EVENT_VALUE Null sind.

```

USER_ID,ITEM_ID,EVENT_TYPE,EVENT_VALUE,LOCATION,DEVICE,TIMESTAMP,IMPRESSION
35,73,click,,Ohio,Tablet,1586731606,73|70|17|95|96|92|55|45|16|97|56|54|33|94|36|10|5|
43|19|13|51|90|65|59|38
54,35,watch,0.75,Indiana,Cellphone,1586735164,35|82|78|57|20|63|1|90|76|75|49|71|26|24|
25|6|37|85|40|98|32|13|11|54|48
9,33,click,,Oregon,Cellphone,1586735158,68|33|62|6|15|57|45|24|78|89|90|40|26|91|66|31|
47|17|99|29|27|41|77|75|14
23,10,watch,0.25,California,Tablet,1586735697,92|89|36|10|39|77|4|27|79|18|83|16|28|68|
78|40|50|3|99|7|87|49|12|57|53
27,11,watch,0.55,Indiana,Tablet,1586735763,11|7|39|95|71|1|6|40|41|28|99|53|68|76|0|65|
69|36|22|42|34|67|24|20|66
...
...

```

Anforderungen an das Datensatz-Schema der Benutzer (benutzerdefiniert)

Ein Benutzerdatensatz speichert Metadaten über Ihre Benutzer. Dies kann Informationen wie Alter, Geschlecht und Loyalitätsmitgliedschaft für jeden Artikel beinhalten. Informationen zu den Arten von Benutzerdaten, die Sie in Amazon Personalize importieren können, finden Sie unter [Benutzerdatensatz](#).

Die Daten, die Sie für jeden Benutzer angeben, müssen Ihrem Schema entsprechen. Sie müssen mindestens eine Benutzer-ID für jeden Benutzer angeben (maximale Länge 256 Zeichen). Abhängig von Ihrem Schema können Benutzermetadaten leere Werte oder Nullwerte enthalten. Ihr Benutzerschema muss mindestens ein Metadatenfeld enthalten. Wenn Sie jedoch einen null Typ hinzufügen, kann dieser Wert für den Benutzer Null sein. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht

als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen.

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ `string` hinzu und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Bei Benutzern mit mehreren Kategorien trennen Sie die einzelnen Werte mithilfe des vertikalen Balkens `|` voneinander ab. Für ein Feld `SUBSCRIPTION_MODEL` könnten Ihre Daten für einen Benutzer beispielsweise `student|monthly|discount discount`.

Kategoriale Werte können maximal 1000 Zeichen enthalten. Wenn Sie einen Benutzer mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Benutzerdatensatz finden Sie unter [Service Quotas](#).

Beispiel für ein Benutzerschema (benutzerdefiniert)

Das folgende Beispiel zeigt, wie ein Benutzerschema strukturiert wird. Das `USER_ID` Feld ist erforderlich und die `GENDER` Felder `AGE` und sind Metadaten. Es ist mindestens ein Metadatenfeld erforderlich, und Sie können maximal 25 Metadatenfelder hinzufügen. Informationen zu Schemaanforderungen finden Sie unter [Anforderungen an benutzerdefinierte Datensätze und Schemas](#).

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "AGE",
      "type": "int"
    },
    {
      "name": "GENDER",
      "type": "string",
      "categorical": true
    }
  ]
}
```



```
],  
  "version": "1.0"  
}
```

Für dieses Schema könnten die ersten Zeilen mit historischen Daten in einer CSV-Datei wie folgt aussehen.

```
USER_ID,AGE,GENDER  
5,34,Male  
6,56,Female  
8,65,Male  
...  
...
```

Anforderungen an das Artikeldatensatz-Schema (benutzerdefiniert)

Ein Artikeldatensatz speichert Metadaten zu Ihren Artikeln in Ihrem Katalog. Dies kann Informationen wie Preis, Genre und Verfügbarkeit für jeden Artikel beinhalten. Informationen zu den Artikeldatentypen, die Sie in Amazon Personalize importieren können, finden Sie unter [Elementedatensatz](#).

Die Daten, die Sie für jeden Artikel angeben, müssen Ihrem Artikeldatensatzschema entsprechen. Sie müssen mindestens eine Artikel-ID für jeden Artikel angeben (maximale Länge 256 Zeichen). Abhängig von Ihrem Schema können Elementmetadaten leere Werte oder Nullwerte enthalten. Ihr Schema muss mindestens ein Metadatenfeld enthalten. Wenn Sie jedoch einen `null` Typ hinzufügen, kann dieser Wert für das Element Null sein. Es steht Ihnen frei, je nach Ihrem Anwendungsfall und Ihren Daten weitere Felder hinzuzufügen. Solange die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#), liegen die Feldnamen und Datentypen bei Ihnen.

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ `string` hinzu und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Artikel auf. Kategoriewerte können maximal 1000 Zeichen lang sein. Wenn Sie ein Element mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Bei Elementen mit mehreren Kategorien trennen Sie die einzelnen Werte durch den senkrechten Balken `|` voneinander. Für ein `GENRES`-Feld könnten Ihre Daten für ein Element beispielsweise `Action|Crime|Biopic` Wenn Sie über mehrere Ebenen von kategorialen Daten verfügen und

einige Elemente mehrere Kategorien für jede Ebene in der Hierarchie haben, fügen Sie für jede Ebene ein Feld hinzu und fügen Sie hinter jedem Feldnamen einen Ebenenindikator an: GENRES, GENRE_L2, GENRE_L3. Auf diese Weise können Sie Empfehlungen auf der Grundlage von Unterkategorien filtern, auch wenn ein Element zu mehreren Kategorien mit mehreren Ebenen gehört (Informationen zum Erstellen und Verwenden von Filtern finden Sie unter). [Filtern von Empfehlungen und Benutzersegmenten](#) Ein Video könnte beispielsweise die folgenden Daten für jede Kategorieebene enthalten:

- GENRES: Aktion | Abenteuer
- GENRE_L2: Krimi | Western
- GENRE_L3: Biopic

In diesem Beispiel handelt es sich bei dem Video um Action > Verbrechen > Biopic-Hierarchie und Abenteuer > Western > Biopic-Hierarchie. Wir empfehlen, nur bis zu L3 zu verwenden, aber Sie können bei Bedarf auch mehr Stufen verwenden.

Während der Modellschulung berücksichtigt Amazon Personalize maximal 750.000 Artikel. Wenn Sie mehr als 750.000 Artikel importieren, entscheidet Amazon Personalize, welche Artikel in die Schulung aufgenommen werden sollen, wobei der Schwerpunkt auf der Aufnahme neuer Artikel (Artikel, die Sie kürzlich ohne Interaktionen hinzugefügt haben) und vorhandener Artikel mit aktuellen Interaktionsdaten liegt.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Artikeldatensatz finden Sie unter [Service Quotas](#)

Beispiel für ein Artikel-Dataset-Schema (benutzerdefiniert)

Das folgende Beispiel zeigt, wie ein Artikelschema strukturiert wird. Das Feld ITEM_ID ist ein Pflichtfeld. Das GENRE Feld besteht aus kategorialen Metadaten und das DESCRIPTION Feld besteht aus textuellen Metadaten. Es ist mindestens ein Metadatenfeld erforderlich. Sie können maximal 100 Metadatenfelder hinzufügen. Das CREATION_TIMESTAMP Feld ist ein reserviertes Schlüsselwort. Hinweise zu Schemaanforderungen finden Sie unter [Anforderungen an benutzerdefinierte Datensätze und Schemas](#).

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
```

```
{
  "name": "ITEM_ID",
  "type": "string"
},
{
  "name": "GENRES",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
},
{
  "name": "CREATION_TIMESTAMP",
  "type": "long"
},
{
  "name": "DESCRIPTION",
  "type": [
    "null",
    "string"
  ],
  "textual": true
},
],
"version": "1.0"
}
```

Für dieses Schema könnten die ersten Zeilen mit historischen Daten in einer CSV-Datei wie folgt aussehen.

```
ITEM_ID,GENRES,CREATION_TIMESTAMP,DESCRIPTION
1,Adventure|Animation|Children|Comedy|Fantasy,1570003267,"This is an animated movie
that features action, comedy, and fantasy. Audience is children. This movie was
released in 2004."
2,Adventure|Children|Fantasy,1571730101,"This is an adventure movie with elements of
fantasy. Audience is children. This movie was release in 2010."
3,Comedy|Romance,1560515629,"This is a romantic comedy. The movie was released in 1999.
Audience is young women."
4,Comedy|Drama|Romance,1581670067,"This movie includes elements of both comedy and
drama as well as romance. This movie was released in 2020."
...
...
```

Anforderungen an das Aktions-Dataset-Schema (benutzerdefiniert)

Eine Aktion ist eine Interaktionsaktivität, die Sie Ihren Kunden vielleicht empfehlen möchten. Zu den Aktionen können die Installation Ihrer mobilen App, das Ausfüllen eines Mitgliedsprofils, die Teilnahme an Ihrem Treueprogramm oder die Anmeldung für Werbe-E-Mails gehören. Ein Aktions-Datensatz speichert Daten über Ihre Aktionen. Informationen zu den Arten von Aktionsdaten, die Sie in Amazon Personalize importieren können, finden Sie unter [Datensatz „Aktionen“](#).

Die Daten, die Sie für jede Aktion angeben, müssen Ihrem Aktions-Datensatzschema entsprechen. Abhängig von Ihrem Schema können Aktionsmetadaten leere/Nullwerte enthalten. Ihr Schema muss mindestens ein ACTION_ID-Feld enthalten und Sie müssen für jede Aktion eine ID angeben.

Abhängig von Ihrem Anwendungsfall und Ihren Daten können Sie zusätzliche Felder hinzufügen. Sie können die Feldnamen und Datentypen auswählen, sofern die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#).

Um kategoriale Daten zu verwenden, fügen Sie ein Feld vom Typ `string` hinzu und legen Sie das kategoriale Attribut des Felds `true` in Ihrem Schema auf fest. Nehmen Sie dann die kategorialen Daten in Ihre CSV-Massendatei und die Importe einzelner Aktionen auf. Kategorische Werte können maximal 1000 Zeichen enthalten. Wenn Sie eine Aktion mit einem kategorialen Wert mit mehr als 1000 Zeichen haben, schlägt Ihr Datensatz-Importjob fehl.

Bei Aktionen mit mehreren Kategorien trennen Sie die einzelnen Werte durch den senkrechten Balken `|` voneinander. Für ein MEMBERSHIP_LEVEL-Feld könnten Ihre Daten für eine Aktion beispielsweise lauten. `Premium|DeLuxe|Exclusive`

Während des Modelltrainings berücksichtigt Amazon Personalize maximal 1000 Aktionen. Wenn Sie mehr als 1000 Aktionen importieren, entscheidet Amazon Personalize, welche Aktionen in die Schulung aufgenommen werden sollen, wobei neuen Aktionen (Aktionen, die Sie kürzlich ohne Interaktionen hinzugefügt haben) und bestehenden Aktionen mit Daten zu aktuellen Interaktionen Priorität eingeräumt wird.

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Aktionsdatensatz finden Sie unter [Service Quotas](#).

Beispiel für ein Aktions-Dataset-Schema (benutzerdefiniert)

Das folgende Beispiel zeigt, wie ein Actions-Schema strukturiert wird. Das Feld ACTION_ID ist ein Pflichtfeld. Das MEMBERSHIP_LEVEL Feld ist ein kategoriales Zeichenfolgenfeld. Die

REPEAT_FREQUENCY Felder VALUECREATION_TIMESTAMP, und sind reservierte Schlüsselwörter mit den erforderlichen Typen. Sie können maximal 10 Spalten hinzufügen. Informationen zu Schemaanforderungen finden Sie unter [Anforderungen an benutzerdefinierte Datensätze und Schemas](#).

```
{
  "type": "record",
  "name": "Actions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ACTION_ID",
      "type": "string"
    },
    {
      "name": "VALUE",
      "type": [
        "null",
        "long"
      ]
    },
    {
      "name": "MEMBERSHIP_LEVEL",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "REPEAT_FREQUENCY",
      "type": [
        "long",
        "null"
      ]
    }
  ],
}
```

```
"version": "1.0"  
}
```

Für dieses Schema könnten die ersten Zeilen mit historischen Daten in einer CSV-Datei wie folgt aussehen.

```
ACTION_ID,VALUE,MEMBERSHIP_LEVEL,CREATION_TIMESTAMP,REPEAT_FREQUENCY  
1,10,Deluxe|Premium,1510003267,7  
2,5,Basic,1580003267,7  
3,5,Preview,1590003267,3  
4,10,Deluxe|Platinum,1560003267,4  
...  
...
```

Schemaanforderungen für den Datensatz „Aktionsinteraktionen“ (benutzerdefiniert)

Ein Datensatz mit Aktionsinteraktionen speichert historische Daten und Echtzeitdaten von Interaktionen zwischen Benutzern und Aktionen in Ihrem Aktionsdatensatz. Informationen zu den Datentypen, die Amazon Personalize verwenden kann, finden Sie unter [Datensatz „Aktionen“: Interaktionen](#).

Die Daten, die Sie für jede Interaktion angeben, müssen Ihrem Schema entsprechen. Abhängig von Ihrem Schema können Interaktionsmetadaten leere/Nullwerte enthalten. Ihr Schema muss mindestens Folgendes enthalten:

- USER_ID
- ACTION_ID
- TIMESTAMP (ZEITSTEMPEL)
- EVENT_TYPE

Abhängig von Ihrem Anwendungsfall und Ihren Daten können Sie zusätzliche Felder hinzufügen. Sie können die Feldnamen und Datentypen auswählen, sofern die Felder nicht als erforderlich oder reserviert aufgeführt sind und die Datentypen unter aufgeführt sind [Schema-Datentypen](#).

Weitere Informationen zu Mindestanforderungen und maximalen Datenlimits für einen Datensatz mit Aktionsinteraktionen finden Sie unter [Service Quotas](#).

Beispiel für ein Schema für einen Datensatz mit Aktionsinteraktionen (benutzerdefiniert)

Das folgende Beispiel zeigt ein Schema für einen Datensatz mit Aktionsinteraktionen, der nur die erforderlichen Felder enthält. Hinweise zu allgemeinen Anforderungen an die Formatierung von Schemas finden Sie unter [Anforderungen an die Schemaformatierung](#).

```
{
  "type": "record",
  "name": "ActionInteractions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ACTION_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Für dieses Schema könnten die ersten Zeilen mit historischen Daten in einer CSV-Datei wie folgt aussehen. Beachten Sie, dass einige Werte für IMPRESSION Null sind.

```
USER_ID,ACTION_ID,EVENT_TYPE,TIMESTAMP
35,73,Viewed,1586731606
54,35,Not taken,1586731609
9,33,Viewed,1586735158
23,10,Taken,1586735697
27,11,Taken,1586735763
...
```

...

Erstellen eines Schemas mit SDK for Python (Boto3)

1. Definieren Sie das Schema, das Sie verwenden möchten, im Avro-Format.
2. Speichern Sie das Schema in einer JSON-Datei im Standardordner von Python.
3. Erstellen Sie das Schema mit dem folgenden Code.

```
import boto3

personalize = boto3.client('personalize')

with open('schema.json') as f:
    createSchemaResponse = personalize.create_schema(
        name = 'YourSchema'
        schema = f.read()
    )

schema_arn = createSchemaResponse['schemaArn']

print('Schema ARN:' + schema_arn )
```

4. Amazon Personalize gibt den ARN des neuen Schemas zurück. Speichern Sie es für eine spätere Verwendung.

Amazon Personalize bietet Operationen zur Verwaltung von Schemas. Beispielsweise können Sie die [ListSchemas](#)-API verwenden, um eine Liste der verfügbaren Schemata abzurufen.

Nachdem Sie ein Schema erstellt haben, verwenden Sie es mit Datasets, die dem Schema entsprechen. Weitere Informationen finden Sie unter [Richtlinien für Datenformate](#).

Richtlinien für Datenformate

Wenn Sie Daten in Amazon Personalize importieren, können Sie wählen, ob Sie Datensätze in großen Mengen, einzeln oder beides importieren möchten.

- Bei Massenimporten wird eine große Anzahl historischer Datensätze gleichzeitig importiert. Sie können Ihre Massendaten mit Data Wrangler und mehreren SageMaker Datenquellen vorbereiten

und importieren. Oder Sie können Massendaten selbst vorbereiten und sie direkt aus einer CSV-Datei in Amazon S3 in Amazon Personalize importieren.

- Bei Einzelimporten importieren Sie einzelne Datensätze mit der Amazon Personalize Personalize-Konsole und API-Vorgängen. Oder Sie können Interaktionsdaten von Live-Events in Echtzeit streamen. Weitere Informationen zu einzelnen Importen finden Sie unter [Einzelne Datensätze importieren](#).

Bevor Sie Ihre Massendaten importieren, stellen Sie sicher, dass sie richtig formatiert sind. Die folgenden Abschnitte können Ihnen beim Formatieren Ihrer Massendaten helfen. Wenn Sie sich nicht sicher sind, wie Sie Ihre Daten formatieren sollen, können Sie Amazon SageMaker Data Wrangler (Data Wrangler) verwenden, um Ihre Daten vorzubereiten. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

Themen

- [Richtlinien und Anforderungen für das Format von Massendaten](#)
- [Beispiel für Interaktionsdaten](#)
- [Formatieren von expliziten Eindrücken](#)
- [Formatieren von kategorialen Daten](#)

Richtlinien und Anforderungen für das Format von Massendaten

Mithilfe der folgenden Richtlinien und Anforderungen können Sie sicherstellen, dass Ihre Massendaten korrekt formatiert sind.

- Ihre Eingabedaten müssen sich in einer CSV-Datei (durch Kommas getrennte Werte) befinden.
- Die erste Zeile Ihrer CSV-Datei muss Ihre Spaltenüberschriften enthalten. Die Überschriften sollten nicht von Anführungszeichen (") umgeben sein.
- Stellen Sie sicher, dass Sie über die erforderlichen Felder für Ihren Datensatztyp verfügen und dass ihre Namen den Anforderungen von Amazon Personalize entsprechen. Beispielsweise könnten Ihre Artikeldaten eine Spalte ITEM_IDENTIFICATION_NUMBER mit IDs für jeden Ihrer Artikel enthalten. Um diese Spalte als ITEM_ID-Feld zu verwenden, benennen Sie die Spalte in um. ITEM_ID Wenn Sie Data Wrangler verwenden, um Ihre Daten zu formatieren, können Sie die Transformation Map columns for Amazon Personalize Data Wrangler verwenden, um sicherzustellen, dass Ihre Spalten korrekt benannt sind.

Informationen zu Pflichtfeldern finden Sie unter [Schemata](#) Informationen zur Verwendung von Data Wrangler zur Vorbereitung Ihrer Daten finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#)

- Die Namen der Spaltenüberschriften in Ihrer CSV-Datei müssen Ihrem Schema entsprechen.
- Jeder Datensatz in Ihrer CSV-Datei muss sich in einer einzigen Zeile befinden.
- Die Datentypen in jeder Spalte müssen Ihrem Schema entsprechen. Wenn Sie Data Wrangler zum Formatieren Ihrer Daten verwenden, können Sie die Data Wrangler-Transformation „[Wert als Typ analysieren](#)“ verwenden, um die Datentypen zu konvertieren.
- `TIMESTAMP` und die `CREATION_TIMESTAMP` Daten müssen im UNIX-Epochenzeitformat vorliegen. Weitere Informationen finden Sie unter [Zeitstempeldaten](#).
- Wenn Ihre Daten nicht-ASCII-kodierte Zeichen enthalten, muss Ihre CSV-Datei im UTF-8-Format codiert sein.
- Stellt sicher, dass Sie alle Textdaten wie unter beschrieben formatieren. [Unstrukturierte Textmetadaten](#)
- Achten Sie darauf, Impressionsdaten und kategoriale Daten wie unter und beschrieben zu formatieren. [Formatieren von expliziten Eindrücken](#) [Formatieren von kategorialen Daten](#)

Beispiel für Interaktionsdaten

Die folgenden Interaktionsdaten stellen historische Benutzeraktivitäten auf einer Website dar, auf der Kinokarten verkauft werden. Sie könnten die Daten verwenden, um ein Modell zu trainieren, das Filmempfehlungen auf der Grundlage der Interaktionsdaten der Benutzer bereitstellt.

```
USER_ID, ITEM_ID, EVENT_TYPE, EVENT_VALUE, TIMESTAMP
196, 242, click, 15, 881250949
186, 302, click, 13, 891717742
22, 377, click, 10, 878887116
244, 51, click, 20, 880606923
166, 346, click, 10, 886397596
298, 474, click, 40, 884182806
115, 265, click, 20, 881171488
253, 465, click, 50, 891628467
305, 451, click, 30, 886324817
```

Hier ist das zugehörige Interaktionsschema:

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    { "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": "float"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Amazon Personalize benötigt die `TIMESTAMP` Felder `USER_ID`, `ITEM_ID`, und `USER_ID` ist die Kennung für einen Benutzer Ihrer Anwendung. `ITEM_ID` ist die Kennung für einen Film. `EVENT_TYPE` und `EVENT_VALUE` sind die Identifikatoren für Benutzeraktivitäten. In den Beispieldaten `click` könnte `a` ein Kaufereignis für einen Film darstellen und `15` könnte der Kaufpreis des Films sein. `TIMESTAMP` steht für die Zeit der Unix-Zeit, in der der Film gekauft wurde.

Zeitstempeldaten

Zeitstempeldaten, wie z. B. Daten `TIMESTAMP` (für Artikelinteraktionsdatensätze) oder `CREATION_TIMESTAMP` (für Artikeldatensätze), müssen im Zeitformat der Unix-Epoche in Sekunden vorliegen. Der Epoch-Zeitstempel in Sekunden für das Datum 31. Juli 2020 lautet beispielsweise 1596238243. [Verwenden Sie einen Epoch-Konverter — Unix-Zeitstempel-Konverter, um Datumsangaben in Unix-Zeitstempel umzuwandeln.](#)

Formatieren von expliziten Eindrücken

Wenn Sie das [Benutzerpersonalisierung](#) Rezept verwenden, können Sie Impressionsdaten aufzeichnen und hochladen. Impressions sind Listen von Elementen, die für einen Benutzer sichtbar waren, wenn er mit einem bestimmten Element interagiert hat (z. B. angeklickt oder angesehen). Um Impressionsdaten in einem Massendatenimport hochzuladen, notieren Sie jede Element-ID manuell. Achten Sie darauf, die Werte als Teil Ihrer historischen Interaktionsdaten durch einen senkrechten Strich, das Zeichen '|', zu trennen. Das vertikale Balkenzeichen wird auf die Obergrenze von 1000 Zeichen für Impressionsdaten angerechnet. Weitere Informationen zu Impressionsdaten finden Sie unter [Daten zu Eindrücken](#).

Im Folgenden finden Sie einen kurzen Auszug aus einem Datensatz mit Artikelinteraktionen, der explizite Impressions in der IMPRESSION Spalte enthält.

EVENT_TYPE	EINDRUCK	ITEM_ID	TIMESTAMP	USER_ID
klicken Sie auf	73 70 17 95 96	73	1586731606	BENUTZER_1
klicken Sie auf	35 82 78 57 20 63 1 90 76 75 49 71 26 24 25 6	35	1586735164	BENUTZER_2
...

In der Anwendung wurden USER_1 Benutzerelemente 73, 70, und 17 angezeigt, 95, 96 und der Benutzer entschied sich schließlich für das Element 73. Wenn Sie auf der Grundlage dieser Daten eine neue Lösungsversion erstellen, werden die Elemente 70, 17, 95, und 96 dem Benutzer seltener empfohlen.

Formatieren von kategorialen Daten

Um mehrere Kategorien für ein einzelnes Element bei der Verwendung kategorischer Zeichenfolgedaten einzubinden, trennen Sie die Werte mit dem vertikalen Balkenzeichen '|'. Bei einem Element, das zwei Kategorien hat, würde eine Datenzeile beispielsweise wie folgt aussehen:

```
ITEM_ID,GENRE
item_123,horror|comedy
```

Nachdem Sie Ihre Daten formatiert haben, laden Sie sie in einen Amazon S3 S3-Bucket hoch, damit Sie sie in Amazon Personalize importieren können. Weitere Informationen finden Sie unter [Hochladen des Hochladen des Uploads des Uploads des Amazon S3 Pakts](#).

Domain-Anwendungsfälle und benutzerdefinierte Rezepte

Amazon Personalize bietet verschiedene Anwendungsfälle für Domänen und benutzerdefinierte Rezepte für Trainingsmodelle:

- Wenn Sie einen Empfehlungsgeber in einer Domain-Datensatzgruppe erstellen, geben Sie einen Anwendungsfall an. Amazon Personalize trainiert die Modelle, die den Empfehlungsgeber unterstützen, mit den besten Konfigurationen für den jeweiligen Anwendungsfall.
- Wenn Sie eine benutzerdefinierte Lösung in einer benutzerdefinierten Datensatzgruppe oder einer Domänen-Datensatzgruppe erstellen, geben Sie ein Rezept an und konfigurieren Trainingsparameter. Wenn Sie eine Lösungsversion für die Lösung erstellen, trainiert Amazon Personalize die Modelle, die der Lösungsversion zugrunde liegen, auf der Grundlage des Rezepts und der Trainingskonfiguration.

Themen

- [Anwendungsfall- und Rezeptfunktionen](#)
- [Auswahl eines Anwendungsfalls](#)
- [Ein Rezept wählen](#)

Anwendungsfall- und Rezeptfunktionen

Bei einigen Anwendungsfällen und Rezepten verwendet Amazon Personalize die folgenden Funktionen, um relevantere Empfehlungen zu generieren und die Elementerkennung und das Engagement zu verbessern.

Themen

- [Personalisierung in Echtzeit](#)
- [Exploration \(Erkundung\)](#)
- [Automatische Updates](#)

Personalisierung in Echtzeit

Bei einigen Anwendungsfällen und Rezepten verwendet Amazon Personalize Echtzeit-Personalisierung, um Empfehlungen zu aktualisieren und an das sich entwickelnde Interesse

eines Benutzers anzupassen. Es aktualisiert Empfehlungen für einen Benutzer, während Sie seine Interaktionen mit Elementen oder Aktionen aufzeichnen, die im letzten vollständigen Training vorhanden sind. Sie zeichnen diese Interaktionen mit einem Ereignis-Tracker und der [-PutEvents](#) Operation oder bei Interaktionen mit Aktionen die [- PutActionInteractions](#) Operation auf.

Weitere Informationen zur Aufzeichnung von Ereignissen finden Sie unter [Ereignisse aufzeichnen](#). Informationen zu neuen Daten, die Empfehlungen in Echtzeit beeinflussen, einschließlich Echtzeit-Personalisierung, finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Die folgenden Anwendungsfälle und Rezepte unterstützen Echtzeit-Personalisierung:

- [Für Sie empfohlen \(Anwendungsfall ECOMMERCE\)](#)
- [Top-Auswahlen für Sie \(Anwendungsfall VIDEO_ON_DEMAND\)](#)
- [User-Personalization-Rezept](#)
- [Personalisiertes Ranking-Rezept](#)
- [Nächstes Best-Action-Rezept](#)

Exploration (Erkundung)

Für einige Domain-Anwendungsfälle und benutzerdefinierte Rezepte verwendet Amazon Personalize die Erkundung, wenn es Elemente empfiehlt. Bei der Erkundung umfassen Empfehlungen einige Elemente oder Aktionen, die normalerweise weniger wahrscheinlich für den Benutzer empfohlen werden würden, z. B. neue Elemente oder Aktionen, Elemente oder Aktionen mit wenigen Interaktionen oder Elemente oder Aktionen, die für den Benutzer aufgrund ihres vorherigen Verhaltens weniger relevant sind. Dies verbessert die Erkennung und Interaktion von Elementen, wenn Sie einen sich schnell ändernden Katalog haben oder wenn neue Elemente, wie Nachrichtenartikel oder Werbeaktionen, für Benutzer relevanter sind, da sie aktuell sind.

Wenn Ihr Anwendungsfall oder Rezept Erkundung verwendet, wenn Sie einen Empfehler oder eine benutzerdefinierte Kampagne erstellen oder wenn Sie einen Batch-Inferenzauftrag (benutzerdefinierte Ressourcen) erstellen, können Sie die Erkundung mit den folgenden Feldern konfigurieren:

- Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht) – Konfigurieren Sie, wie viel Sie untersuchen möchten. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen

oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).

- Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Elementexploration basierend auf dem Elementalter. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Elementalter bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum ist 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

Anwendungsfälle und Rezepte, die Erkundung verwenden

Weitere Informationen zu den einzelnen Anwendungsfällen oder Rezepten, die Erkundungen verwenden, finden Sie im Folgenden:

- [Für Sie empfohlen \(Anwendungsfall ECOMMERCE\)](#)
- [Top-Auswahlen für Sie \(Anwendungsfall VIDEO_ON_DEMAND\)](#)
- [User-Personalization-Rezept](#)
- [Nächstes Best-Action-Rezept](#)

Automatische Updates

Für einige Anwendungsfälle und benutzerdefinierte Rezepte aktualisiert Amazon Personalize automatisch Ihre Empfehlungs- oder Lösungsversion, um neue Elemente oder Aktionen für Empfehlungen zu berücksichtigen. Für automatische Updates fallen keine Kosten an. Eine Liste der Anwendungsfälle und Rezepte mit automatischen Updates finden Sie unter [Domain-Anwendungsfälle und benutzerdefinierte Rezepte mit automatischen Updates](#).

Automatische Updates funktionieren wie folgt:

- Wann Amazon Personalize Ihre Lösungsversion oder Ihren Empfehler automatisch aktualisiert, hängt davon ab, wie Sie Empfehlungen erhalten:

- Für Echtzeitempfehlungen aktualisiert Amazon Personalize die Lösungsversion oder den Empfehler alle zwei Stunden.
- Wenn Sie bei Empfehlungen für Batch-Elemente einen Batch-Inferenzantrag erstellen und die neueste vollständig trainierte Lösungsversion für Ihre Lösung angeben, aktualisiert Amazon Personalize die Lösungsversion automatisch, um neue Elemente während der Erkundung zu berücksichtigen. Wenn Sie nicht die neueste Lösungsversion angeben, erfolgt kein Update.
- Mit jedem Update beginnt Amazon Personalize mit der Aufnahme neuer Elemente in Empfehlungen mithilfe von [Exploration \(Erkundung\)](#). Wenn Sie ein neues Element oder eine neue Aktion in Betracht ziehen, berücksichtigt Amazon Personalize alle Metadaten für das Element. Diese Daten haben jedoch erst dann größere Auswirkungen auf Empfehlungen, wenn Sie Interaktionen für das Element aufgezeichnet und vollständig neu trainiert haben.
- Automatische Updates sind kein vollständiges erneutes Training. Stattdessen können Sie mit automatischen Updates Ihre neuen Elemente vor dem nächsten vollständigen erneuten Training in Empfehlungen veröffentlichen. Ein vollständiges Training kann erfolgen, nachdem das wöchentliche automatische Retraining Ihres Domain-Empfehlungsgebers abgeschlossen ist. Oder es kann sein, nachdem Sie eine neue Lösungsversion erstellt haben, bei der auf `trainingMode` gesetzt ist `FULL`.
- Damit eine Aktualisierung durchgeführt werden kann, müssen Sie neue Aktions-, Element- oder Interaktionsdaten seit der letzten automatischen Aktualisierung oder dem vollständigen erneuten Training angeben.
- Amazon Personalize berücksichtigt neue Elemente, bis Sie 750.000 Elemente importieren. Dies ist die maximale Anzahl von Elementen, die während des Trainings berücksichtigt werden.

Zusätzliche Richtlinien und Anforderungen für benutzerdefinierte Ressourcen

Wenn Sie benutzerdefinierte Ressourcen verwenden, finden Sie im Folgenden Richtlinien und Anforderungen für automatische Updates:

- Ihre Lösungsversion muss in einer Kampagne bereitgestellt werden. Ihre Kampagne verwendet automatisch die aktualisierte Lösungsversion.
- Automatische Updates sind kein vollständiges erneutes Training. Sie sollten weiterhin wöchentlich eine neue Lösungsversion trainieren, bei der auf `trainingMode` gesetzt ist, `FULL` damit das Modell aus dem Verhalten Ihrer Benutzer lernen kann.
- Wenn Sie manuell eine neue Lösungsversion erstellen, aktualisiert Amazon Personalize ältere Lösungsversionen nicht automatisch, selbst wenn Sie sie in einer Kampagne bereitgestellt haben.

- Wenn nicht alle zwei Stunden häufig genug ist, können Sie manuell eine Lösungsversion erstellen, bei der auf `trainingMode` gesetzt ist, `UPDATE` um diese neuen Elemente in Empfehlungen aufzunehmen. Denken Sie einfach daran, dass Amazon Personalize automatisch nur Ihre neueste vollständig trainierte Lösungsversion aktualisiert. Die manuell aktualisierte Lösungsversion wird in Zukunft nicht automatisch aktualisiert. Um Aktualisierungen fortzusetzen, erstellen Sie eine neue Lösung, bei der der Trainingsmodus auf `setzt` ist, `FULL` und stellen Sie sie in einer Kampagne bereit.

Domain-Anwendungsfälle und benutzerdefinierte Rezepte mit automatischen Updates

Weitere Informationen zu den einzelnen Anwendungsfällen oder Rezepten mit automatischen Updates finden Sie im Folgenden:

- [Empfohlen für Sie \(Anwendungsfall ECOMMERCE\)](#)
- [Top-Auswahlen für Sie \(Anwendungsfall VIDEO_ON_DEMAND\)](#)
- [User-Personalization-Rezept](#)
- [Nächstes Best-Action-Rezept](#)

Auswahl eines Anwendungsfalls

Wenn Sie einen Empfehlungsgeber in einer Domain-Datensatzgruppe erstellen, geben Sie einen Anwendungsfall an. Amazon Personalize trainiert die Modelle, die den Recommender unterstützen, mit den besten Konfigurationen für den jeweiligen Anwendungsfall. Jede Domain hat unterschiedliche Anwendungsfälle. Wenn Sie beispielsweise `VIDEO_ON_DEMAND` für Ihre Domain-Datensatzgruppe angeben, sind nur `VIDEO_ON_DEMAND`-Anwendungsfälle verfügbar. Für jeden Anwendungsfall gelten unterschiedliche Anforderungen für das Abrufen von Empfehlungen. Für einige Anwendungsfälle sind bestimmte Ereignistypen erforderlich. Es steht Ihnen frei, weitere Ereignistypen hinzuzufügen.

Für alle Anwendungsfälle müssen Ihre Interaktionsdaten Folgendes enthalten:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Themen

- [Anwendungsfälle für VIDEO_ON_DEMAND](#)
- [ECOMMERCE-Anwendungsfälle](#)

Anwendungsfälle für VIDEO_ON_DEMAND

In den folgenden Abschnitten werden die Anforderungen und der Amazon-Ressourcenname (ARN) für jeden Anwendungsfall von VIDEO_ON_DEMAND aufgeführt. Für alle Anwendungsfälle müssen Ihre Interaktionsdaten Folgendes enthalten:

- Mindestens 1 000 Datensätze für Elementinteraktionen von Benutzern, die mit Elementen in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Elementinteraktionen.

Für Qualitätsempfehlungen empfehlen wir Ihnen, mindestens 50.000 Elementinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Elementinteraktionen durchzuführen.

Note

Wenn Sie die [CreateRecommender](#) API verwenden, geben Sie den hier aufgeführten ARN für den Rezept-ARN an.

Themen

- [Da Sie X angesehen haben](#)
- [Mehr wie X](#)
- [Beliebtste](#)
- [Trending jetzt](#)
- [Top-Auswahlen für Sie](#)

Da Sie X angesehen haben

Erhalten Sie Empfehlungen für Videos, die sich andere Benutzer auch angesehen haben, basierend auf einem von Ihnen angegebenen Video. In diesem Anwendungsfall filtert Amazon Personalize automatisch Videos, die der Benutzer angesehen hat, basierend auf der von Ihnen angegebenen `userId` und `Watch` Ereignissen. Wenn Sie Ihren eigenen Filter anwenden, wird Ihr Filter angewendet, nachdem die Videos, die der Benutzer angesehen hat, herausgefiltert wurden.

Beim Filtern berücksichtigt Amazon Personalize maximal 100 Elementinteraktionen pro Benutzer und Ereignistyp. Dies gilt für alle automatischen oder benutzerdefinierten Filter. Sie können die [Service Quotas-Konsole](#) verwenden, um eine Erhöhung dieses Limits anzufordern. Weitere Informationen finden Sie im Abschnitt [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch für Service Quotas.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-vod-because-you-watched-x`
- `GetRecommendations` API-Anforderungen:

`userId`: Erforderlich

`itemId`: Erforderlich

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: mindestens 1 000 `Watch` Ereignisse.

Mehr wie X

Erhalten Sie Empfehlungen für Videos, die einem von Ihnen angegebenen Video ähneln. In diesem Anwendungsfall filtert Amazon Personalize automatisch Videos, die der Benutzer angesehen hat, basierend auf der von Ihnen angegebenen `userId` und `Watch` Ereignissen. Wenn Sie Ihren eigenen Filter anwenden, wird Ihr Filter angewendet, nachdem die Videos, die der Benutzer angesehen hat, herausgefiltert wurden.

Beim Filtern berücksichtigt Amazon Personalize maximal 100 Elementinteraktionen pro Benutzer und Ereignistyp. Dies gilt für alle automatischen oder benutzerdefinierten Filter. Sie können die [Service Quotas-Konsole](#) verwenden, um eine Erhöhung dieses Limits anzufordern. Weitere Informationen finden Sie im Abschnitt [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch für Service Quotas.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-vod-more-like-x`
- GetRecommendations API-Anforderungen:
 - `userId`: Erforderlich
 - `itemId`: Erforderlich
- Datensätze, die beim Training verwendet werden:
 - Interaktionen (erforderlich)
 - Elemente (erforderlich)
- Erforderliche Anzahl von Ereignissen: Mindestens 1 000 Ereignisse eines beliebigen Typs.
- Empfohlene Ereignistypen: - Watch und -ClickEreignisse.

Beliebtste

Erhalten Sie Empfehlungen für Videos, die von den meisten Benutzern angesehen wurden.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-vod-most-popular`
- GetRecommendations -Anforderungen:
 - `userId`: Erforderlich
 - `itemId`: nicht verwendet
- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: Mindestens 1 000 Watch Ereignisse.

Trending jetzt

Erhalten Sie Empfehlungen für Videos, die derzeit im Trend sind. Trending-Videos sind Elemente, die bei Ihren Benutzern schnell beliebter werden. Alle zwei Stunden wertet Amazon Personalize Ihre Interaktionsdaten automatisch aus und identifiziert Trendelemente.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-vod-trending-now`
- GetRecommendations API-Anforderungen:
 - `userId`: Nur erforderlich, wenn Sie nach Elementen filtern `CurrentUser`, mit denen ein Benutzer interagiert hat

`itemId`: nicht verwendet

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Anzahl von Ereignissen: Mindestens 1 000 Ereignisse eines beliebigen Typs.

Top-Auswahlen für Sie

Erhalten Sie personalisierte Inhaltsempfehlungen für einen von Ihnen angegebenen Benutzer. In diesem Anwendungsfall filtert Amazon Personalize automatisch Videos, die der Benutzer angesehen hat, basierend auf der von Ihnen angegebenen `userId` und `Watch` Ereignissen. Wenn Sie Ihren eigenen Filter anwenden, wird Ihr Filter angewendet, nachdem die Videos, die der Benutzer angesehen hat, herausgefiltert wurden.

Beim Filtern berücksichtigt Amazon Personalize maximal 100 Elementinteraktionen pro Benutzer und Ereignistyp. Dies gilt für alle automatischen oder benutzerdefinierten Filter. Sie können die [Service Quotas-Konsole](#) verwenden, um eine Erhöhung dieses Limits anzufordern. Weitere Informationen finden Sie im Abschnitt [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch für Service Quotas.

Bei der Empfehlung von Elementen verwendet dieser Anwendungsfall [real-time-personalization](#) und die [Erkundung von](#) . Und es verwendet [automatische Updates](#), um neue Elemente für Empfehlungen zu berücksichtigen.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-vod-top-picks`
- `GetRecommendations` -Anforderungen:

`userId`: Erforderlich

`itemId`: nicht verwendet

- Datensätze, die beim Training verwendet werden:
 - Interaktionen (erforderlich)
 - Elemente (optional)
 - Benutzer (optional)
- Erforderliche Anzahl von Ereignissen: mindestens 1000 Ereignisse.
- Empfohlene Ereignistypen: - `Click` und -`Watch`Ereignisse.

- Konfigurationsparameter für die Nutzung: Wenn Sie einen Empfehler erstellen, können Sie die Erkundung wie folgt konfigurieren.
- Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht) – Konfigurieren Sie, wie viel untersucht werden soll. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).
- Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Articlexploration auf der Grundlage des Artikelalters. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Elementalter bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum beträgt 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

ECOMMERCE-Anwendungsfälle

In den folgenden Abschnitten werden die Anforderungen und der Amazon-Ressourcenname (ARN) für jeden ECOMMERCE-Anwendungsfall aufgeführt. Für alle Anwendungsfälle müssen Ihre Interaktionsdaten Folgendes haben:

- Mindestens 1 000 Datensätze für Elementinteraktionen von Benutzern, die mit Elementen in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Elementinteraktionen.

Für Qualitätsempfehlungen empfehlen wir Ihnen, mindestens 50.000 Elementinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Elementinteraktionen durchzuführen.

Note

Wenn Sie die [CreateRecommender](#) API verwenden, geben Sie den hier aufgeführten ARN für den Rezept-ARN an.

Themen

- [Am häufigsten angesehen](#)
- [Bewährte Verkäufer](#)
- [Häufig zusammen gekauft](#)
- [Kunden, die X auch angesehen haben](#)
- [Empfohlen für Sie](#)

Am häufigsten angesehen

Erhalten Sie Empfehlungen für beliebte Artikel basierend darauf, wie oft Ihre Kunden einen Artikel angesehen haben.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-views`
- GetRecommendations -Anforderungen:

`userId`: Erforderlich

`itemId`: nicht verwendet

`inputList`: ENTF.

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: mindestens 1 000 View Ereignisse.

Bewährte Verkäufer

Erhalten Sie Empfehlungen für beliebte Artikel, je nachdem, wie oft Ihre Kunden einen Artikel gekauft haben.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-purchases`

- **GetRecommendations -Anforderungen:**

`userId`: Erforderlich

`itemId`: nicht verwendet

`inputList`: ENTF.

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: mindestens 1 000 Purchase Ereignisse.

Häufig zusammen gekauft

Erhalten Sie Empfehlungen für Artikel, die Kunden häufig kaufen, zusammen mit einem von Ihnen angegebenen Artikel.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-ecomm-frequently-bought-together`
- **GetRecommendations -Anforderungen:**

`userId`: Nur erforderlich, wenn Sie nach filtern `CurrentUser`

`itemId`: Erforderlich

`inputList`: ENTF.

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: mindestens 1 000 Purchase Ereignisse.

Kunden, die X auch angesehen haben

Erhalten Sie Empfehlungen für Elemente, die Kunden auch auf der Grundlage eines von Ihnen angegebenen Elements angesehen haben. In diesem Anwendungsfall filtert Amazon Personalize automatisch Elemente, die der Benutzer gekauft hat, basierend auf der von Ihnen angegebenen `userId` und `Purchase` Ereignissen. Wenn Sie Ihren eigenen Filter anwenden, wird Ihr Filter angewendet, nachdem die Elemente, die der Benutzer bereits gekauft hat, herausgefiltert wurden.

Beim Filtern berücksichtigt Amazon Personalize maximal 100 Elementinteraktionen pro Benutzer und Ereignistyp. Dies gilt für alle automatischen oder benutzerdefinierten Filter. Sie können die [Service Quotas-Konsole](#) verwenden, um eine Erhöhung dieses Limits anzufordern. Weitere Informationen finden Sie im Abschnitt [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch für Service Quotas.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-ecomm-customers-who-viewed-x-also-viewed`
- GetRecommendations -Anforderungen:

`userId`: Erforderlich

`itemId`: Erforderlich

`inputList`: ENTF.

- Datensätze, die beim Training verwendet werden: Nur Datensatz zu Elementinteraktionen (erforderlich)
- Erforderliche Ereignistypen: mindestens 1 000 View Ereignisse.
- Empfohlene Ereignistypen: Purchase Ereignisse.

Empfohlen für Sie

Erhalten Sie personalisierte Empfehlungen für Elemente basierend auf einem von Ihnen angegebenen Benutzer. In diesem Anwendungsfall filtert Amazon Personalize automatisch Elemente heraus, die der Benutzer basierend auf der von Ihnen angegebenen `userId` und Purchase Ereignissen erworben hat. Wenn Sie Ihren eigenen Filter anwenden, wird Ihr Filter angewendet, nachdem die Elemente, die der Benutzer bereits gekauft hat, herausgefiltert wurden.

Beim Filtern berücksichtigt Amazon Personalize maximal 100 Elementinteraktionen pro Benutzer und Ereignistyp. Dies gilt für alle automatischen oder benutzerdefinierten Filter. Sie können die [Service Quotas-Konsole](#) verwenden, um eine Erhöhung dieses Limits anzufordern. Weitere Informationen finden Sie im Abschnitt [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch für Service Quotas.

Bei der Empfehlung von Elementen verwendet dieser Anwendungsfall [real-time-personalization](#) und die [Erkundung von](#) . Und es verwendet [automatische Updates](#), um neue Elemente für Empfehlungen zu berücksichtigen.

- Rezept-ARN: `arn:aws:personalize:::recipe/aws-ecomm-recommended-for-you`
- GetRecommendations -Anforderungen:

`userId`: Erforderlich

`itemId`: nicht verwendet

`inputList`: ENTF.

- Datensätze, die beim Training verwendet werden:
 - Interaktionen (erforderlich)
 - Elemente (optional)
 - Benutzer (optional)
- Erforderliche Anzahl von Ereignissen: mindestens 1000 Ereignisse.
- Empfohlene Ereignistypen: - View und -PurchaseEreignisse.
- Konfigurationsparameter für die Nutzung: Wenn Sie einen Empfehler erstellen, können Sie die Erkundung wie folgt konfigurieren.
 - Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht) – Konfigurieren Sie, wie viel untersucht werden soll. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei Null findet keine Untersuchung statt und Empfehlungen basieren auf aktuellen Daten (Relevanz).
 - Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Elementexploration basierend auf dem Elementalter. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Alter von Elementen bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum beträgt 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

Ein Rezept wählen

Wenn Sie eine benutzerdefinierte Lösung erstellen, geben Sie ein Rezept an und konfigurieren die Trainingsparameter. Rezepte sind Amazon Personalize Personalize-Algorithmen, die für bestimmte Anwendungsfälle vorbereitet wurden. Amazon Personalize bietet Rezepte für Trainingsmodelle, die auf gängigen Anwendungsfällen basieren. Wenn Sie eine Lösungsversion für die Lösung erstellen, trainiert Amazon Personalize die Modelle, die die Lösungsversion unterstützen, basierend auf dem Rezept und der Trainingskonfiguration.

In den Rezepten von Amazon Personalize wird beim Training Folgendes verwendet:

- Vordefinierte Attribute Ihrer Daten
- Vordefinierte Funktionstransformationen
- Vordefinierte Algorithmen
- Anfängliche Parametereinstellungen für die Algorithmen

Um Ihr Modell zu optimieren, können Sie viele dieser Parameter überschreiben, wenn Sie eine Lösung erstellen. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Themen

- [Amazon Personalize Sie Rezepttypen nach Anwendungsfall](#)
- [Rezepte von Amazon Personalize](#)
- [Verfügbare Amazon Personalize Personalize-Rezepte anzeigen](#)
- [USER_PERSONALIZATION](#)
- [BELIEBTE_ARTIKEL](#)
- [PERSONALISIERTE RANGFOLGE](#)
- [RELATED_ITEMS](#)
- [PERSONALISIERTE AKTIONEN](#)
- [USER_SEGMENTIERUNG](#)

Amazon Personalize Sie Rezepttypen nach Anwendungsfall

Um Ihr Rezept auszuwählen, wählen Sie zunächst Ihren Anwendungsfall aus den folgenden Optionen aus und notieren Sie sich den entsprechenden Rezepttyp.

- Artikel für Benutzer empfehlen (USER_PERSONALIZATION-Rezepte)

Um Ihren Benutzern personalisierte Empfehlungen zu geben, trainieren Sie Ihr Modell mit einem USER_PERSONALIZATION-Rezept. Personalisierte Empfehlungen tragen zu einer besseren Kundenbindung und Konversionsrate bei.

- Artikel für einen Benutzer einordnen (PERSONALIZED_RANKING-Rezepte)

Um die Reihenfolge der kuratierten Listen oder Suchergebnisse für Ihre Benutzer zu personalisieren, trainieren Sie Ihr Modell mit einem PERSONALIZED_RANKING-Rezept. Mit den Rezepten von PERSONALIZED_RANKING wird eine personalisierte Liste erstellt, indem eine Sammlung von Eingabeelementen auf der Grundlage des prognostizierten Interesses für einen bestimmten Benutzer neu geordnet wird. Personalisierte Listen verbessern das Kundenerlebnis und erhöhen die Kundenbindung und das Kundenengagement.

- Empfehlen von trendigen oder beliebten Artikeln (POPULAR_ITEMS Rezepte)

Verwenden Sie ein Rezept von POPULAR_ITEMS, um trendige oder beliebte Artikel zu empfehlen. Sie können ein POPULAR_ITEMS verwenden, wenn Ihre Kunden Wert darauf legen, mit was andere Benutzer interagieren. Zu den häufigsten Verwendungszwecken gehören die Empfehlung viraler Inhalte in sozialen Medien, aktuelle Nachrichtenartikel oder aktuelle Sportvideos.

- Empfehlen ähnlicher Artikel (RELATED_ITEMS-Rezepte)

Um ähnliche Artikel zu empfehlen, z. B. Artikel, die häufig zusammen gekauft wurden, oder Filme, die sich auch andere Nutzer angesehen haben, sollten Sie ein RELATED_ITEMS-Rezept verwenden. Wenn Sie ähnliche Artikel empfehlen, können Sie Ihren Kunden helfen, Artikel zu finden, und die Konversionsrate der Nutzer erhöhen.

- Empfehlung der nächstbesten Aktion (PERSONALIZED_ACTIONS-Rezepte)

Um Ihren Benutzern in Echtzeit die nächstbeste Aktion zu empfehlen, z. B. die Registrierung für Ihr Treueprogramm oder die Beantragung einer Kreditkarte, sollten Sie ein Rezept für PERSONALIZED_ACTIONS verwenden. Wenn Sie die nächstbeste Aktion empfehlen, können Sie die Kundenbindung erhöhen, mehr Umsatz generieren und das Nutzererlebnis verbessern.

- Benutzersegmente abrufen (USER_SEGMENTATION-Rezepte)

Um Benutzersegmente auf der Grundlage von Artikeleingabedaten zu erhalten, z. B. Benutzer, die höchstwahrscheinlich mit Elementen mit einem bestimmten Attribut interagieren, sollten Sie ein USER_SEGMENTATION-Rezept verwenden. Durch das Abrufen von Benutzersegmenten können Sie erweiterte Marketingkampagnen erstellen, mit denen verschiedene Artikel in verschiedenen

Benutzersegmenten je nach der Wahrscheinlichkeit, dass sie eine Aktion ergreifen, beworben werden.

Rezepte von Amazon Personalize

Amazon Personalize bietet die folgenden Arten von Rezepten. Neben verhaltensbedingten Unterschieden gelten für jeden Typ andere Anforderungen, um Empfehlungen zu erhalten, wie aus der folgenden Tabelle hervorgeht.

Rezepttyp	Rezepte	API	API-Anforderungen
USER_PERSONALIZATION	Benutzerpersonalisierung HRNN-Rezept (veraltet) HRNN-Metadaten-Rezept (veraltet) HRNN-Coldstart-Rezept (veraltet)	GetRecommendations	userId: Erforderlich itemId: nicht verwendet inputList : ENTF.
BELIEBTE_ARTIKEL	Trending – Jetzt Beliebtheit-Anzahl	GetRecommendations	userId: Nur erforderlich, wenn Sie einen Filter anwenden, der dies erfordert itemId: nicht verwendet inputList : ENTF.
PERSONALISIERTE_RANGFOLGE	Personalisierte Rangfolge	GetPersonalizedRanking	userId: Erforderlich

Rezepttyp	Rezepte	API	API-Anforderungen
			<p>itemId: ENTF.</p> <p>inputList : Liste von itemIds</p>
RELATED_ITEMS	<p>Ähnliche Elemente</p> <p>SIMS</p>	GetRecommendations	<p>userId: Nur erforderlich, wenn Sie einen Filter anwenden, der dies erfordert</p> <p>itemId: Erforderlich</p> <p>inputList : ENTF.</p>
PERSONALIZED_ACTIONS	Nächste Best-Action	GetActionRecommendations	<p>userId: Erforderlich</p> <p>actionId: nicht verwendet</p> <p>itemId: nicht verwendet</p> <p>inputList : ENTF.</p>

Rezepttyp	Rezepte	API	API-Anforderungen
BENUTZERSEGMENTIERUNG	Element-Affinität Element-Attribut-Affinität	CreateBatchSegmentJob	Informationen zu den Anforderungen für Batch-Workflows finden Sie unter. Erstellen eines Batch-Segmentauftrags

Verfügbare Amazon Personalize Personalize-Rezepte anzeigen

Anzeigen einer Liste verfügbarer Rezepte:

- Wählen Sie in der Amazon Personalize Personalize-Konsole eine Datensatzgruppe aus. Wählen Sie im Navigationsbereich Solutions and recipes (Lösungen und Rezepte) und anschließend die Registerkarte Recipes (Rezepte) aus.
- Rufen Sie mit AWS SDK for Python (Boto3) die [ListRecipes](#) API auf.
- Verwenden Sie mit dem AWS CLI den folgenden Befehl.

```
aws personalize list-recipes
```

Rufen Sie die API auf, um Informationen über ein Rezept mit dem SDK for Python (Boto3) zu erhalten. [DescribeRecipe](#) Verwenden Sie den folgenden Befehl AWS CLI, um Informationen über ein Rezept mit dem zu erhalten.

```
aws personalize describe-recipe --recipe-arn recipe_arn
```


USER_PERSONALIZATION

Die Rezepte von USER_PERSONALIZATION sagen anhand der Datensätze Interaktionen, Artikel und Benutzer voraus, mit welchen Elementen ein Benutzer interagieren wird. Wenn Sie ein Empfehlungssystem aufbauen, das personalisierte Empfehlungen für jeden Ihrer Benutzer bereitstellt, sollten Sie Ihr Modell mit einem USER_PERSONALIZATION-Rezept trainieren.

Die Rezepte für USER_PERSONALIZATION lauten wie folgt:

- [User-Personalization-Rezept](#)
- [Ältere Rezepte für die Benutzerpersonalisierung](#)

User-Personalization-Rezept

Das User-Personalization (aws-user-personalization)-Rezept ist für alle personalisierten Empfehlungsszenarien optimiert. Es prognostiziert die Elemente, mit denen ein Benutzer höchstwahrscheinlich interagieren wird. Sie können User-Personalization verwenden, um personalisierte Filmempfehlungen für eine Streaming-App oder personalisierte Produktempfehlungen für eine Einzelhandels-App zu generieren.

Mit User-Personalization generiert Amazon Personalize Empfehlungen, die hauptsächlich auf Benutzerelementinteraktionsdaten in einem Elementinteraktionsdatensatz basieren. Es kann auch beliebige Element- und Benutzermetadaten in Ihren Datensätzen für Elemente und Benutzer verwenden. Weitere Informationen zu den verwendeten Daten finden Sie unter [Erforderliche und optionale Datensätze](#).

Themen

- [Rezept-Features](#)
- [Erforderliche und optionale Datensätze](#)
- [Eigenschaften und Hyperparameter](#)
- [Training mit dem User-Personalization-Rezept \(Konsole\)](#)
- [Training mit dem User-Personalization-Rezept \(Python SDK\)](#)
- [Abrufen von Empfehlungen und Aufzeichnungsimpessionen \(SDK für Python \(Boto3\)\)](#)
- [Beispiel für ein Jupyter-Notebook](#)

Rezept-Features

User-Personalization verwendet beim Generieren von Elementempfehlungen die folgenden Amazon-Personalize-Rezeptfunktionen:

- Personalisierung in Echtzeit – Mit der Personalisierung in Echtzeit aktualisiert und passt Amazon Personalize Elementempfehlungen an das sich entwickelnde Interesse eines Benutzers an. Weitere Informationen finden Sie unter [Personalisierung in Echtzeit](#).
- Nutzung – Bei der Erkundung umfassen Empfehlungen neue Elemente oder Elemente mit weniger Interaktionsdaten. Dies verbessert die Erkennung und Interaktion von Elementen, wenn Sie über einen sich schnell ändernden Katalog verfügen oder wenn neue Elemente, wie Nachrichtenartikel oder Werbeaktionen, für Benutzer relevanter sind, wenn sie auf dem neuesten Stand sind. Weitere Informationen zur Erkundung finden Sie unter [Exploration \(Erkundung\)](#).
- Automatische Updates – Bei automatischen Updates aktualisiert Amazon Personalize automatisch alle zwei Stunden das neueste Modell (Lösungsversion), um neue Elemente für Empfehlungen zu berücksichtigen. Weitere Informationen finden Sie unter [Automatische Updates](#).

Erforderliche und optionale Datensätze

Um die User-Personalization verwenden zu können, müssen Sie ein erstellen [Datensatz mit Interaktionen zwischen Elementen](#) und mindestens 1000 Elementinteraktionen importieren. Amazon Personalize generiert Empfehlungen hauptsächlich auf der Grundlage von Elementinteraktionsdaten.

Mit User-Personalization kann Amazon Personalize Daten zu Elementinteraktionen verwenden, die Folgendes enthalten:

- Ereignistyp- und Ereigniswertdaten – Amazon Personalize verwendet Ereignistypdaten wie Klick- oder Watch-Ereignistypen, um die Absicht und das Interesse von Benutzern anhand von Mustern in ihrem Verhalten zu identifizieren. Außerdem können Sie Ereignistyp- und Ereigniswertdaten verwenden, um Datensätze vor dem Training zu filtern. Weitere Informationen finden Sie unter [Daten zu Ereignistyp und Ereigniswert](#).
- Kontextbezogene Metadaten – Kontextbezogene Metadaten sind Interaktionsdaten, die Sie zum Zeitpunkt eines Ereignisses in der Umgebung des Benutzers sammeln, z. B. seinen Standort oder Gerätetyp. Weitere Informationen finden Sie unter [Kontextuelle Metadaten](#).
- Impressionsdaten – Impressions sind Listen von Elementen, die für einen Benutzer sichtbar waren, als er mit einem bestimmten Element interagiert hat (angeklickt, angesehen, gekauft usw.). Weitere Informationen finden Sie unter [Daten zu Eindrücken](#).

Die folgenden Datensätze sind optional und können Empfehlungen verbessern:

- Benutzerdatensatz – Amazon Personalize kann Daten in Ihrem Benutzerdatensatz verwenden, um Ihre Benutzer und ihre Interessen besser zu verstehen. Sie können auch Daten in einem Benutzerdatensatz verwenden, um Empfehlungen zu filtern. Informationen zu den Benutzerdaten, die Sie importieren können, finden Sie unter [Benutzerdatensatz](#).
- Artikeldatensatz – Amazon Personalize kann Daten in Ihrem Artikeldatensatz verwenden, um Verbindungen und Muster in ihrem Verhalten zu identifizieren. Dies hilft Amazon Personalize, Ihre Benutzer und ihre Interessen zu verstehen. Sie können auch Daten in einem Elementdatensatz verwenden, um Empfehlungen zu filtern. Informationen zu den Elementdaten, die Sie importieren können, finden Sie unter [Elementedatensatz](#).

Eigenschaften und Hyperparameter

Das User-Personalization-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-user-personalization`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-user-personalization`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-user-personalization`

Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

In der folgenden Tabelle werden die Hyperparameter für das User-Personalization-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Algorithmus-Hyperparameter	
<p><code>hidden_dimension</code></p>	<p>Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Datensatz zu Elementinteraktionen kompliziertere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung. Um den besten Wert zu wählen, verwenden Sie HPO. Um HPO zu verwenden, legen Sie <code>performHPO</code> auf <code>true</code> fest, wenn Sie die Operationen CreateSolutionVersion und CreateSolution aufrufen.</p> <p>Standardwert</p> <p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
<p><code>bptt</code></p>	<p>Legt fest, ob die Technik „Backpropagation über Zeit“ verwendet werden soll. Die Backpropagation über Zeit ist eine Technik, die Gewichtungen in wiederkehrenden neuronalen netzwerkbasierten Algorithmen aktualisiert. Verwenden Sie <code>bptt</code> für langfristige Guthaben, um verzögerte Belohnungen mit frühen Ereignissen zu verbinden. Eine verzögerte Belohnung kann beispielsweise ein Kauf nach mehreren Klicks sein. Ein frühes Ereignis kann ein erster Klick sein. Selbst bei denselben Ereignistypen wie einem Klick empfiehlt es sich, langfristige Effekte zu berücksichtigen und die Gesamtelohnungen zu maximieren.</p>

Name	Beschreibung
	<p>Um Langzeiteffekte zu berücksichtigen, verwenden Sie höhere <code>bptt</code>-Werte. Die Verwendung eines höheren <code>bptt</code>-Werts erfordert größere Datasets und mehr Zeit für die Verarbeitung.</p> <p>Standardwert: 32</p> <p>Bereich: [2, 32]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
<code>recency_mask</code>	<p>Bestimmt, ob das Modell die neuesten Beliebtheits-trends im Datensatz zu Elementinteraktionen berücksichtigen soll. Neueste Beliebtheitstrends können plötzliche Änderungen in den zugrunde liegenden Mustern von Interaktionsereignissen umfassen. Um ein Modell zu schulen, das aktuelle Ereignisse stärker gewichtet, legen Sie <code>recency_mask</code> auf <code>true</code> fest. Um ein Modell zu schulen, das alle vorherigen Interaktionen gleichermaßen abwägt, legen Sie <code>recency_mask</code> auf <code>false</code> fest. Um gute Empfehlungen mit gleicher Gewichtung zu erhalten, benötigen Sie möglicherweise ein größeres Schulungsdataset.</p> <p>Standardwert: <code>True</code></p> <p>Bereich: <code>True</code> oder <code>False</code></p> <p>Werttyp: Boolescher Wert</p> <p>HPO-optimierbar: Ja</p>

Hyperparameter für die Auslastung

Name	Beschreibung
<code>min_user_history_length_percentile</code>	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen, da die Daten für diese Benutzer in der Regel Rauschen enthalten. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.99</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Hyperparameter der Kampagnenkonfiguration für die Elementexploration

Name	Beschreibung
exploration_weight	<p>Bestimmt, wie häufig Empfehlungen Elemente mit weniger Daten oder Relevanz zur Elementinteraktion enthalten. Je näher der Wert bei 1,0 liegt, desto mehr Erkundungen. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz). Weitere Informationen finden Sie unter the section called "CampaignConfig".</p> <p>Standardwert: 0.3</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<p>exploration_item_age_cut_of_f</p>	<p>Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Artikelexploration auf der Grundlage des Artikelalters. Amazon Personalize bestimmt das Alter eines Elements anhand seines Erstellungszeitstempels oder, falls Erstellungszeitstempeldaten fehlen, anhand von Elementinteraktionsdaten. Weitere Informationen dazu, wie Amazon Personalize das Alter eines Artikels bestimmt, finden Sie unter Zeitstempeldaten der Erstellung.</p> <p>Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum beträgt 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.</p> <p>Standardwert: 30.0</p> <p>Bereich: Positive Gleitkommazahlen</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Training mit dem User-Personalization-Rezept (Konsole)

Um das User-Personalization-Rezept zum Generieren von Empfehlungen in der Konsole zu verwenden, trainieren Sie zunächst eine neue Lösungsversion mit dem -Rezept. Stellen Sie dann eine Kampagne mit der Lösungsversion bereit und verwenden Sie die Kampagne, um Empfehlungen zu erhalten.

Training einer neuen Lösungsversion mit dem User-Personalization-Rezept (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Erstellen Sie eine benutzerdefinierte Datensatzgruppe mit einem neuen Schema und laden Sie Ihren Datensatz mit Impressionsdaten hoch. Fügen Sie optional [CREATION_TIMESTAMP](#) und [Unstrukturierte Textmetadaten](#) Daten in Ihren Artikel-Datensatz ein, damit Amazon Personalize das Alter eines Artikels genauer berechnen und kalte Elemente identifizieren kann.

Weitere Informationen zum Importieren von Daten finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

3. Wählen Sie auf der Seite Datensatzgruppen die neue Datensatzgruppe aus, die den Datensatz oder Datensätze mit Impressionsdaten enthält.
4. Wählen Sie im Navigationsbereich Lösungen und Rezepte und dann Lösung erstellen aus.
5. Geben Sie auf der Seite Lösung erstellen für den Lösungsnamen den Namen Ihrer neuen Lösung ein.
6. Wählen Sie für Lösungstyp die Option Elementempfehlung aus, um Elementempfehlungen für Ihre Benutzer zu erhalten.
7. Wählen Sie für Rezept aws-user-personalization. Der Abschnitt Lösungskonfiguration wird mit mehreren Konfigurationsoptionen angezeigt.
8. Wenn Ihr Datensatz zu Elementinteraktionen in der Lösungskonfiguration `EVENT_TYPE` oder die Spalten `EVENT_TYPE` und `EVENT_VALUE` enthält, verwenden Sie optional die Schwellenwertfelder Ereignistyp und Ereigniswert, um die Daten zu Elementinteraktionen auszuwählen, die Amazon Personalize beim Training des Modells verwendet.

Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

9. Konfigurieren Sie optional Hyperparameter für Ihre Lösung. Eine Liste der Eigenschaften und Hyperparameter von User-Personalization-Rezepten finden Sie unter [Eigenschaften und Hyperparameter](#).
10. Wählen Sie Lösung erstellen und trainieren, um mit dem Training zu beginnen. Die Seite Dashboard wird angezeigt.

Sie können zur Seite mit den Lösungsdetails navigieren, um den Trainingsfortschritt im Abschnitt Lösungsversionen zu verfolgen. Wenn das Training abgeschlossen ist, lautet der Status Aktiv .

Erstellen einer Kampagne und Abrufen von Empfehlungen (Konsole)

Wenn der Status Ihrer Lösung Aktiv lautet, können Sie Ihre Kampagne erstellen und Empfehlungen wie folgt erhalten:

1. Wählen Sie entweder auf der Seite mit den Lösungsdetails oder auf der Seite Kampagnen die Option Neue Kampagne erstellen aus.
2. Geben Sie auf der Seite Neue Kampagne erstellen für Kampagnendetails die folgenden Informationen an:
 - Kampagnenname: Geben Sie den Namen der Kampagne ein. Der Text, den Sie hier eingeben, wird auf der Seite Kampagnen-Dashboard und Details angezeigt.
 - Lösung: Wählen Sie die Lösung aus, die Sie gerade erstellt haben.
 - Lösungsversions-ID: Wählen Sie die ID der Lösungsversion aus, die Sie gerade erstellt haben.
 - Minimale bereitgestellte Transaktionen pro Sekunde: Legen Sie die minimalen bereitgestellten Transaktionen pro Sekunde fest, die Amazon Personalize unterstützt. Weitere Informationen finden Sie unter dem Vorgang [CreateCampaign](#).
3. Geben Sie für Kampagnenkonfiguration die folgenden Informationen an:
 - Auslastungsgewicht: Konfigurieren Sie, wie viel untersucht werden soll, wobei Empfehlungen Elemente mit weniger Elementinteraktionsdaten oder Relevanzen enthalten, je häufiger Sie die Untersuchung angeben. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).
 - Altersabbruch für Nutzungselemente: Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion ein, um den Umfang der Elementexploration zu definieren. Um die Anzahl der Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein.

Wenn Sie beispielsweise 10 eingeben, werden nur Elemente mit Elementinteraktionsdaten aus den 10 Tagen seit der letzten Interaktion im Datensatz während der Untersuchung berücksichtigt.

Note


Empfehlungen können Elemente ohne Daten zur Elementinteraktion von außerhalb dieses Zeitrahmens enthalten. Dies liegt daran, dass diese Elemente für die Interessen

der Benutzer relevant sind und keine Untersuchung erforderlich war, um sie zu identifizieren.

4. Wählen Sie Create campaign (Kampagne erstellen).
5. Wenn der Kampagnenstatus auf der Seite mit den Kampagnendetails Aktiv lautet, können Sie die Kampagne verwenden, um Empfehlungen zu erhalten und Impressionen aufzuzeichnen. Weitere Informationen finden Sie unter [Schritt 5: Holen Sie sich Empfehlungen](#) unter „Erste Schritte“.

Amazon Personalize aktualisiert Ihre neueste Lösungsversion automatisch alle zwei Stunden, um neue Daten aufzunehmen. Ihre Kampagne verwendet automatisch die aktualisierte Lösungsversion. Weitere Informationen finden Sie unter [Automatische Updates](#).

Um die Kampagne manuell zu aktualisieren, erstellen und trainieren Sie zunächst mithilfe der Konsole oder der [-CreateSolutionVersion](#) Operation eine neue Lösungsversion, wobei auf `trainingMode` gesetzt ist `update`. Anschließend aktualisieren Sie die Kampagne manuell auf der Seite Kampagnen der Konsole oder mithilfe der [-UpdateCampaign](#) Operation.

 Note

Amazon Personalize aktualisiert Lösungsversionen, die Sie vor dem 17. November 2020 erstellt haben, nicht automatisch.

Training mit dem User-Personalization-Rezept (Python SDK)

Wenn Sie eine Datensatzgruppe erstellt und Ihren Datensatz(e) mit Impressionsdaten hochgeladen haben, können Sie eine Lösung mit dem User-Personalization-Rezept trainieren. Nehmen Sie optional [CREATION_TIMESTAMP](#) und [Unstrukturierte Textmetadaten](#) Daten in Ihren Artikel-Datensatz auf, damit Amazon Personalize das Alter eines Artikels genauer berechnen und kalte Elemente identifizieren kann. Weitere Informationen zum Erstellen von Datensatzgruppen und zum Hochladen von Trainingsdaten finden Sie unter [Schemata](#).

So trainieren Sie eine Lösung mit dem User-Personalization-Rezept unter Verwendung des AWS SDK

1. Erstellen Sie eine neue Lösung mit der `-create_solution` Methode.

Ersetzen Sie durch `solution name` Ihren Lösungsnamen und `dataset group arn` durch den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe.

```
import boto3

personalize = boto3.client('personalize')

print('Creating solution')
create_solution_response = personalize.create_solution(name = 'solution name',
                                                       recipeArn = 'arn:aws:personalize::recipe/aws-user-
personalization',
                                                       datasetGroupArn = 'dataset group arn',
                                                       )
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

Eine Liste der `aws-user-personalization` Rezeptigenschaften und Hyperparameter finden Sie unter [Eigenschaften und Hyperparameter](#).

- Erstellen Sie eine neue Lösungsversion mit den aktualisierten Trainingsdaten und legen `trainingMode` Sie auf fest, FULL indem Sie den folgenden Codeausschnitt verwenden. Ersetzen Sie durch `solution arn` den ARN Ihrer Lösung.

```
import boto3

personalize = boto3.client('personalize')

create_solution_version_response = personalize.create_solution_version(solutionArn
                               = 'solution arn',
                               trainingMode='FULL')

new_solution_version_arn = create_solution_version_response['solutionVersionArn']
print('solution_version_arn:', new_solution_version_arn)
```

- Wenn Amazon Personalize mit der Erstellung Ihrer Lösungsversion fertig ist, erstellen Sie Ihre Kampagne mit den folgenden Parametern:
 - Geben Sie ein neues `campaign name` und das in Schritt 2 `solution version arn` generierte an.
 - Ändern Sie den Hyperparameter für die Konfiguration der `explorationWeight` Elementexploration, um zu konfigurieren, wie viel untersucht werden soll. Elemente mit

weniger Elementinteraktionsdaten oder Relevanz werden häufiger empfohlen, je näher der Wert bei 1,0 liegt. Der Standardwert ist 0,3.

- Ändern Sie den Hyperparameterparameterparameter der `explorationItemAgeCutOff` Elementexplorationskonfiguration, um die maximale Dauer in Tagen im Vergleich zur letzten Interaktion anzugeben, für die Elemente untersucht werden sollen. Je größer der Wert ist, desto mehr Elemente werden bei der Erkundung berücksichtigt.

Verwenden Sie den folgenden Python-Ausschnitt, um eine neue Kampagne zu erstellen, deren Schwerpunkt auf der Erkundung mit dem Explorations-Cutoff nach 30 Tagen liegt. Das Erstellen einer Kampagne dauert in der Regel einige Minuten, kann aber auch über eine Stunde dauern.

```
import boto3

personalize = boto3.client('personalize')

create_campaign_response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn',
    minProvisionedTPS = 1,
    campaignConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
"explorationItemAgeCutOff": "30"}}
)

campaign_arn = create_campaign_response['campaignArn']
print('campaign_arn:', campaign_arn)
```

Mit User-Personalization aktualisiert Amazon Personalize Ihre Lösungsversion automatisch alle zwei Stunden, um neue Daten aufzunehmen. Ihre Kampagne verwendet automatisch die aktualisierte Lösungsversion. Weitere Informationen finden Sie unter [Automatische Updates](#).

Um die Kampagne manuell zu aktualisieren, erstellen und trainieren Sie zunächst mithilfe der Konsole oder der [-CreateSolutionVersion](#) Operation eine neue Lösungsversion, wobei auf `trainingMode` gesetzt ist `update`. Anschließend aktualisieren Sie die Kampagne manuell auf der Seite Kampagnen der Konsole oder mithilfe der [-UpdateCampaign](#) Operation.

 Note

Amazon Personalize aktualisiert Lösungsversionen, die Sie vor dem 17. November 2020 erstellt haben, nicht automatisch.

Abrufen von Empfehlungen und Aufzeichnungsimpressionen (SDK für Python (Boto3))

Wenn Ihre Kampagne erstellt wird, können Sie sie verwenden, um Empfehlungen für einen Benutzer zu erhalten und Impressionen aufzuzeichnen. Informationen zum Abrufen von Batch-Empfehlungen mithilfe der - AWS SDKs finden Sie unter [Erstellen eines Batch-Inferenzauftrags \(AWS SDKs\)](#).

So erhalten Sie Empfehlungen und nehmen Impressionen auf

1. Rufen Sie die `get_recommendations`-Methode auf. Ändern Sie `campaign_arn` in den ARN Ihrer neuen Kampagne und `user_id` in die `userId` des Benutzers.

```
import boto3

rec_response = personalize_runtime.get_recommendations(campaignArn = 'campaign_arn',
                                                       userId = 'user id')
print(rec_response['recommendationId'])
```

2. Erstellen Sie einen neuen Ereignis-Tracker zum Senden von PutEvents Anfragen. Ersetzen Sie `event_tracker_name` durch den Namen Ihres Ereignis-Trackers und `dataset_group_arn` durch den ARN Ihrer Datensatzgruppe.

```
import boto3

personalize = boto3.client('personalize')

event_tracker_response = personalize.create_event_tracker(
    name = 'event tracker name',
    datasetGroupArn = 'dataset group arn'
)
event_tracker_arn = event_tracker_response['eventTrackerArn']
event_tracking_id = event_tracker_response['trackingId']
print('eventTrackerArn:{},\n eventTrackingId:{}'.format(event_tracker_arn,
                                                       event_tracking_id))
```

3. Verwenden Sie die `recommendationId` aus Schritt 1 und die `event tracking id` aus Schritt 2, um eine neue `PutEvents` Anforderung zu erstellen. Diese Anforderung protokolliert die neuen Impressionsdaten aus der Sitzung des Benutzers. Ändern Sie in `user id` die ID des Benutzers.

```
import boto3

personalize_events.put_events(
    trackingId = 'event tracking id',
    userId= 'user id',
    sessionId = '1',
    eventList = [{
        'sentAt': datetime.now().timestamp(),
        'eventType' : 'click',
        'itemId' : rec_response['itemList'][0]['itemId'],
        'recommendationId': rec_response['recommendationId'],
        'impression': [item['itemId'] for item in rec_response['itemList']],
    }]
)
```

Beispiel für ein Jupyter-Notebook

Ein Beispiel für ein Jupyter-Notebook, das zeigt, wie das User-Personalization-Rezept verwendet wird, finden Sie unter [User Personalization with Exploration](#).

Ältere Rezepte für die Benutzerpersonalisierung

Note

Ältere HRNN-Rezepte sind nicht mehr verfügbar. Diese Dokumentation dient zu Referenzzwecken.

Wir empfehlen, das Rezept `aws-user-personalization` (Benutzerpersonalisierung) gegenüber den älteren HRNN-Rezepten zu verwenden. Die Benutzerpersonalisierung verbessert und vereinheitlicht die Funktionalität der HRNN-Rezepte. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

Im Folgenden finden Sie ältere `USER_PERSONALIZATION`-Rezepte.

- [HRNN-Rezept \(veraltet\)](#)
- [HRNN-Coldstart-Rezept \(veraltet\)](#)
- [HRNN-Metadaten-Rezept \(veraltet\)](#)

HRNN-Rezept (veraltet)

Note

Ältere HRNN-Rezepte sind nicht mehr verfügbar. Diese Dokumentation dient zu Referenzzwecken.

Wir empfehlen, das Rezept `aws-user-personalization` (Benutzerpersonalisierung) gegenüber den älteren HRNN-Rezepten zu verwenden. Die Benutzerpersonalisierung verbessert und vereinheitlicht die Funktionalität der HRNN-Rezepte. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

Das Rezept für hierarchisches wiederkehrendes neuronales Netzwerk (HRNN) von Amazon Personalize modelliert Änderungen im Benutzerverhalten, um während einer Sitzung Empfehlungen zu geben. Eine Sitzung ist eine Reihe von Benutzerinteraktionen innerhalb eines bestimmten Zeitraums mit dem Ziel, ein bestimmtes Element zu finden, um z. B. eine Anforderung zu erfüllen. Indem Sie die jüngsten Interaktionen eines Benutzers höher gewichten, können Sie während einer Sitzung relevantere Empfehlungen geben.

HRNN berücksichtigt Benutzerabsicht und -interessen, die sich mit der Zeit ändern können. Damit werden geordnete Benutzerhistorien automatisch gewichtet, um bessere Inferenzen zu erzielen. HRNN verwendet einen Gating-Mechanismus, um die reduzierten Gewichtungen als lernbare Funktion von Elementen mit Zeitstempel zu modellieren.

Amazon Personalize leitet die Funktionen für jeden Benutzer aus Ihrem Datensatz ab. Findet eine Echtzeit-Datenintegration statt, werden diese Funktionen entsprechend den Benutzeraktivitäten in Echtzeit aktualisiert. Um eine Empfehlung zu erhalten, geben Sie nur die `USER_ID` an. Wenn Sie auch eine `ITEM_ID` angeben, ignoriert Amazon Personalize sie.

Das HRNN-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-hrnn`)
- Rezept Amazon-Ressourcenname (ARN) — `arn:aws:personalize:::recipe/aws-hrnn`
- Algorithmus ARN — `arn:aws:personalize:::algorithm/aws-hrnn`

- Merkmalstransformation ARN — `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Art des Rezepts — `USER_PERSONALIZATION`

In der folgenden Tabelle werden die Hyperparameter für das HRNN-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Hyperparameter des Algorithmus	
<code>hidden_dimension</code>	<p>Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Datensatz für Artikelinteraktionen kompliziertere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung. Bestimmen Sie den optimalen Wert mithilfe der HPO. Um HPO zu verwenden, legen Sie <code>performHPO</code> auf <code>true</code> fest, wenn Sie die Operationen CreateSolutionVersion und CreateSolution aufrufen.</p> <p>Standardwert: 43</p>

Name	Beschreibung
	<p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
bptt	<p>Legt fest, ob die Technik „Backpropagation über Zeit“ verwendet werden soll. Die Backpropagation über Zeit ist eine Technik, die Gewichtungen in wiederkehrenden neuronalen netzwerkbasierten Algorithmen aktualisiert. Verwenden Sie <code>bptt</code> für langfristige Guthaben, um verzögerte Belohnungen mit frühen Ereignissen zu verbinden. Eine verzögerte Belohnung kann beispielsweise ein Kauf nach mehreren Klicks sein. Ein frühes Ereignis kann ein erster Klick sein. Selbst bei denselben Ereignistypen wie einem Klick empfiehlt es sich, langfristige Effekte zu berücksichtigen und die Gesamtelohnungen zu maximieren. Um Langzeiteffekte zu berücksichtigen, verwenden Sie höhere <code>bptt</code>-Werte. Die Verwendung eines höheren <code>bptt</code>-Werts erfordert größere Datasets und mehr Zeit für die Verarbeitung.</p> <p>Standardwert: 32</p> <p>Bereich: [2, 32]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

Name	Beschreibung
recency_mask	<p>Legt fest, ob das Modell die neuesten Beliebtheitstrends im Datensatz Artikelinteraktionen berücksichtigen soll. Neueste Beliebtheitstrends können plötzliche Änderungen in den zugrunde liegenden Mustern von Interaktionsereignissen umfassen. Um ein Modell zu schulen, das aktuelle Ereignisse stärker gewichtet, legen Sie <code>recency_mask</code> auf <code>true</code> fest. Um ein Modell zu schulen, das alle vorherigen Interaktionen gleichermaßen abwägt, legen Sie <code>recency_mask</code> auf <code>false</code> fest. Um gute Empfehlungen mit gleicher Gewichtung zu erhalten, benötigen Sie möglicherweise ein größeres Schulungsdataset.</p> <p>Standardwert: <code>True</code></p> <p>Bereich: <code>True</code> oder <code>False</code></p> <p>Werttyp: Boolescher Wert</p> <p>HPO-optimierbar: Ja</p>

Hyperparameter zur Featureinstellung

Name	Beschreibung
<code>min_user_history_length_percentile</code>	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen, da die Daten für diese Benutzer in der Regel Rauschen enthalten. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min__user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.99</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

HRNN-Metadaten-Rezept (veraltet)

Note

Ältere HRNN-Rezepte sind nicht mehr verfügbar. Diese Dokumentation dient zu Referenzzwecken.

Wir empfehlen, das Rezept `aws-user-personalization` (Benutzerpersonalisierung) gegenüber den älteren HRNN-Rezepten zu verwenden. Die Benutzerpersonalisierung verbessert und vereinheitlicht die Funktionalität der HRNN-Rezepte. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

Das HRNN-Metadaten-Rezept prognostiziert die Elemente, mit denen ein Benutzer interagiert. Dies ähnelt dem [HRNN](#)-Rezept mit zusätzlichen Funktionen, die aus Kontext-, Benutzer- und Elementmetadaten abgeleitet sind (aus Datensätzen für Interaktionen, Benutzer und Elemente). HRNN-Metadaten bietet Genauigkeitsvorteile gegenüber Nicht-Metadatenmodellen, wenn qualitativ hochwertige Metadaten verfügbar sind. Die Verwendung dieses Rezepts erfordert möglicherweise längere Schulungszeiten.

Das HRNN-Metadaten-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-hrnn-metadata`)
- Rezept Amazon-Ressourcenname (ARN) — `arn:aws:personalize:::recipe/aws-hrnn-metadata`
- Algorithmus ARN — `arn:aws:personalize:::algorithm/aws-hrnn-metadata`
- Merkmalstransformation ARN — `arn:aws:personalize:::feature-transformation/featurize_metadata`
- Art des Rezepts — `USER_PERSONALIZATION`

In der folgenden Tabelle werden die Hyperparameter für das HRNN-Metadaten-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Algorithmus-Hyperparameter	
hidden_dimension	<p>Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Datensatz mit Artikelinteraktionen kompliziertere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung. Bestimmen Sie den optimalen Wert mithilfe der HPO. Um HPO zu verwenden, legen Sie <code>performHPO</code> auf <code>true</code> fest, wenn Sie die Operationen CreateSolutionVersion und CreateSolution aufrufen.</p> <p>Standardwert: 43</p> <p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
bptt	<p>Legt fest, ob die Technik „Backpropagation über Zeit“ verwendet werden soll. Die Backpropagation über Zeit ist eine Technik, die Gewichtungen in wiederkehrenden neuronalen netzwerkbasierenden Algorithmen aktualisiert. Verwenden Sie <code>bptt</code> für langfristige Guthaben, um verzögerte Belohnungen mit frühen Ereignissen zu verbinden. Eine verzögerte Belohnung</p>

Name	Beschreibung
	<p>kann beispielsweise ein Kauf nach mehreren Klicks sein. Ein frühes Ereignis kann ein erster Klick sein. Selbst bei denselben Ereignistypen wie einem Klick empfiehlt es sich, langfristige Effekte zu berücksichtigen und die Gesamtbelohnungen zu maximieren. Um Langzeiteffekte zu berücksichtigen, verwenden Sie höhere <code>bptt</code>-Werte. Die Verwendung eines höheren <code>bptt</code>-Werts erfordert größere Datasets und mehr Zeit für die Verarbeitung.</p> <p>Standardwert: 32</p> <p>Bereich: [2, 32]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

Name	Beschreibung
recency_mask	<p>Legt fest, ob das Modell die neuesten Beliebtheitstrends im Datensatz Artikelinteraktionen berücksichtigen soll. Neueste Beliebtheitstrends können plötzliche Änderungen in den zugrunde liegenden Mustern von Interaktionsereignissen umfassen. Um ein Modell zu schulen, das aktuelle Ereignisse stärker gewichtet, legen Sie <code>recency_mask</code> auf <code>true</code> fest. Um ein Modell zu schulen, das alle vorherigen Interaktionen gleichermaßen abwägt, legen Sie <code>recency_mask</code> auf <code>false</code> fest. Um gute Empfehlungen mit gleicher Gewichtung zu erhalten, benötigen Sie möglicherweise ein größeres Schulungsdataset.</p> <p>Standardwert: <code>True</code></p> <p>Bereich: <code>True</code> oder <code>False</code></p> <p>Werttyp: Boolescher Wert</p> <p>HPO-optimierbar: Ja</p>

Hyperparameter zur Featureinstellung

Name	Beschreibung
<code>min_user_history_length_percentile</code>	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen, da die Daten für diese Benutzer in der Regel Rauschen enthalten. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min__user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.99</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

HRNN-Coldstart-Rezept (veraltet)

Note

Ältere HRNN-Rezepte sind nicht mehr verfügbar. Diese Dokumentation dient zu Referenzzwecken.

Wir empfehlen, das Rezept `aws-user-personalization` (Benutzerpersonalisierung) gegenüber den älteren HRNN-Rezepten zu verwenden. Die Benutzerpersonalisierung verbessert und vereinheitlicht die Funktionalität der HRNN-Rezepte. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

Verwenden Sie das HRNN-ColdStart-Rezept, um vorherzusagen, mit welchen Elementen ein Benutzer interagieren wird, wenn Sie häufig neue Elemente und Interaktionen hinzufügen, und Sie möchten sofort Empfehlungen für diese Elemente erhalten. Das HRNN-Kaltstartrezept ähnelt dem [HRNN-Metadata](#)-Rezept, aber es ermöglicht Ihnen, Empfehlungen aus neuen Elementen zu erhalten.

Darüber hinaus können Sie das HRNN-Coldstart-Rezept verwenden, wenn Sie es von Schulungselementen ausschließen möchten, die über eine lange Liste von Interaktionen verfügen, entweder aufgrund eines aktuellen Beliebheitstrends oder weil die Interaktionen äußerst ungewöhnlich sind und zu Störungen bei der Schulung führen. Mit HRNN-Coldstart können Sie weniger relevante Elemente herausfiltern, um eine Teilmenge für die Schulung zu erstellen. Bei der Teilmenge der Elemente, den so genannten „kalten Elementen“, handelt es sich um Elemente, für die im Datensatz Artikelinteraktionen verwandte Interaktionsereignisse vorhanden sind. Ein Element wird als kaltes Element betrachtet, wenn Folgendes zutrifft:

- Es hat weniger Interaktionen als eine angegebene Anzahl von maximalen Interaktionen. Sie geben diesen Wert im `cold_start_max_interactions`-Hyperparameter des Rezepts an.
- Es hat eine kürzere relative Dauer als die maximale Dauer. Sie geben diesen Wert im `cold_start_max_duration`-Hyperparameter des Rezepts an.

Um die Anzahl der Kaltelemente zu reduzieren, legen Sie einen niedrigeren Wert für `cold_start_max_interactions` oder `cold_start_max_duration` fest. Um die Anzahl der Kaltelemente zu erhöhen, legen Sie einen höheren Wert für `cold_start_max_interactions` oder `cold_start_max_duration` fest.

Beim HRNN-Kaltstart gelten folgende Limits für Kaltelemente:

- Maximum cold start items: 80.000
- Minimum cold start items: 100

Wenn die Anzahl der Kaltelemente außerhalb dieses Bereichs liegt, schlagen Versuche, eine Lösung zu erstellen, fehl.

Das HRNN-Kaltstartrezept hat die folgenden Eigenschaften:

- Name (Name – aws-hrnn-coldstart)
- Rezept Amazon-Ressourcenname (ARN) — `arn:aws:personalize:::recipe/aws-hrnn-coldstart`
- Algorithmus ARN — `arn:aws:personalize:::algorithm/aws-hrnn-coldstart`
- Merkmalstransformation ARN — `arn:aws:personalize:::feature-transformation/featurize_coldstart`
- Art des Rezepts — `USER_PERSONALIZATION`

Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

In der folgenden Tabelle werden die Hyperparameter für das HRNN-Kaltstartrezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Hyperparameter des Algorithmus	

Name	Beschreibung
hidden_dimension	<p>Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Datensatz für Artikelinteraktionen kompliziertere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung. Bestimmen Sie den optimalen Wert mithilfe der HPO. Um HPO zu verwenden, legen Sie <code>performHPO</code> auf <code>true</code> fest, wenn Sie die Operationen CreateSolutionVersion und CreateSolution aufrufen.</p> <p>Standardwert</p> <p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

Name	Beschreibung
bptt	<p>Legt fest, ob die Technik „Backpropagation über Zeit“ verwendet werden soll. Die Backpropagation über Zeit ist eine Technik, die Gewichtungen in wiederkehrenden neuronalen netzwerkbasierten Algorithmen aktualisiert. Verwenden Sie <code>bptt</code> für langfristige Guthaben, um verzögerte Belohnungen mit frühen Ereignissen zu verbinden. Eine verzögerte Belohnung kann beispielsweise ein Kauf nach mehreren Klicks sein. Ein frühes Ereignis kann ein erster Klick sein. Selbst bei denselben Ereignistypen wie einem Klick empfiehlt es sich, langfristige Effekte zu berücksichtigen und die Gesamtelohnungen zu maximieren. Um Langzeiteffekte zu berücksichtigen, verwenden Sie höhere <code>bptt</code>-Werte. Die Verwendung eines höheren <code>bptt</code>-Werts erfordert größere Datasets und mehr Zeit für die Verarbeitung.</p> <p>Standardwert: 32</p> <p>Bereich: [2, 32]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

Name	Beschreibung
recency_mask	<p>Legt fest, ob das Modell die neuesten Beliebtheitstrends im Datensatz Artikelinteraktionen berücksichtigen soll. Neueste Beliebtheitstrends können plötzliche Änderungen in den zugrunde liegenden Mustern von Interaktionsereignissen umfassen. Um ein Modell zu schulen, das aktuelle Ereignisse stärker gewichtet, legen Sie <code>recency_mask</code> auf <code>true</code> fest. Um ein Modell zu schulen, das alle vorherigen Interaktionen gleichermaßen abwägt, legen Sie <code>recency_mask</code> auf <code>false</code> fest. Um gute Empfehlungen mit gleicher Gewichtung zu erhalten, benötigen Sie möglicherweise ein größeres Schulungsdataset.</p> <p>Standardwert: <code>True</code></p> <p>Bereich: <code>True</code> oder <code>False</code></p> <p>Werttyp: Boolescher Wert</p> <p>HPO-optimierbar: Ja</p>

Hyperparameter zur Featureinstellung

cold_start_max_interactions	<p>Die maximale Anzahl von Benutzer-Element-Interaktionen, die ein Element als Kaltelement haben kann.</p> <p>Standardwert: 15</p> <p>Bereich: Positive Ganzzahlen</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Nein</p>
-----------------------------	--

Name	Beschreibung
<code>cold_start_max_duration</code>	<p>Die maximale Dauer in Tagen relativ zum Ausgangspunkt einer Benutzer-Element-Interaktion, die als Kaltstartelement betrachtet wird. Um den Ausgangspunkt der Benutzer-Element-Interaktion festzulegen, legen Sie den <code>cold_start_relative_from</code> - Hyperparameter fest.</p> <p>Standardwert: 5.0</p> <p>Bereich: Positive Gleitkommazahlen</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>
<code>cold_start_relative_from</code>	<p>Legt den Ausgangspunkt für das HRNN-Coldstart-Rezept fest, um <code>cold_start_max_duration</code> zu berechnen. Wählen Sie <code>currentTime</code> , um die aktuelle Zeit zu berechnen.</p> <p>Um <code>cold_start_max_duration</code> anhand des Zeitstempels des letzten Elements im Datensatz Artikelinteraktionen zu berechnen, wählen Sie <code>latestItem</code> . Diese Einstellung ist nützlich, wenn Sie häufig neue Elemente hinzufügen.</p> <p>Standardwert: <code>latestItem</code></p> <p>Bereich: <code>currentTime</code> , <code>latestItem</code></p> <p>Werttyp: Zeichenfolge</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>min_user_history_length_percentile</code>	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile</code> to <code>0.05</code> und <code>max_user_history_length_percentile</code> to <code>0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen, da die Daten für diese Benutzer in der Regel Rauschen enthalten. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min__user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.99</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Verwenden von AutoML zur Auswahl eines HRNN-Rezepts (nur API)

Amazon Personalize kann auf der Grundlage seiner Analyse der Eingabedaten automatisch das am besten geeignete Rezept für hierarchisches wiederkehrendes neuronales Netzwerk (HRNN) auswählen. Diese Option wird als AutoML bezeichnet. Zum Ausführen von AutoML setzen Sie den `performAutoML`-Parameter auf `true`, wenn Sie die [CreateSolution](#)-API aufrufen.

Sie können auch die Liste der Rezepte angeben, die Amazon Personalize anhand einer von Ihnen angegebenen Metrik untersucht, um das optimale Rezept zu ermitteln. In diesem Fall rufen Sie die `CreateSolution` Operation auf, spezifizieren `true` für den `performAutoML` Parameter, lassen den `recipeArn` Parameter weg und schließen den `solutionConfig` Parameter ein, wobei Sie `metricName` und `recipeList` als Teil des `autoMLConfig` Objekts angeben.

Wie ein Rezept ausgewählt wird, ist in der folgenden Tabelle dargestellt. Es muss entweder `performAutoML` oder `recipeArn` angegeben werden, aber nicht beides. AutoML erfolgt nur mithilfe der HRNN-Rezepte.

<code>performAutoML</code>	<code>recipeArn</code>	<code>solutionConfig</code>	Ergebnis
<code>true</code>	weglassen	ausgelassen	Amazon Personalize wählt das Rezept
<code>true</code>	weglassen	<code>autoMLConfig</code> : <code>metricName</code> und <code>recipeList</code> werden angegeben	Amazon Personalize wählt aus der Liste ein Rezept aus, das die Metrik optimiert
weglassen	angegeben	ausgelassen	Sie müssen das Rezept angeben.
weglassen	angegeben	angegeben	Sie geben das Rezept an und überschreiben die Standard-Schulungseigenschaften

Note

Wenn `performAutoML true` ist, werden alle Parameter des `solutionConfig`-Objekts mit Ausnahme von `autoMLConfig` ignoriert.

BELIEBTE_ARTIKEL

Verwenden Sie ein `POPULAR_ITEMS`-Rezept, um trendige oder beliebte Artikel zu empfehlen, z. B. aktuelle Nachrichtenartikel oder beliebte Inhalte in sozialen Medien. Verwenden Sie das [Trending-Now-Rezept](#) Rezept, um Empfehlungen für Artikel zu generieren, die bei Ihren Benutzern immer beliebter werden. Um eine Ausgangsbasis für Vergleichszwecke zu erstellen, empfehlen wir, das [Beliebtheit-Anzahl](#) Rezept zu verwenden. Dieses `POPULAR_ITEMS`-Rezept empfiehlt die beliebtesten Artikel auf der Grundlage der Zählung von Interaktionen.

Die Rezepte von `POPULAR_ITEMS` lauten wie folgt:

- [Trending-Now-Rezept](#)
- [Popularity-Count-Rezept](#)

Trending-Now-Rezept

Das `Trending-Now-Rezept` (`aws-trending-now`) generiert Empfehlungen für Elemente, die bei Ihren Benutzern schnell beliebter werden. Sie können das `Trending-Now-Rezept` verwenden, wenn Elemente, die an Popularität zunehmen, für Ihre Kunden relevanter sind. Beispielsweise schätzen Ihre Kunden möglicherweise sehr, womit andere Benutzer interagieren. Zu den gängigsten Anwendungsfällen gehören die Empfehlung von Social-Media-Inhalten, Breaking-News-Artikeln oder aktuelle Sportvideos.

`Trending-Now` identifiziert automatisch die wichtigsten Trendelemente, indem es die Zunahme der Interaktionen berechnet, die jedes Element in konfigurierbaren Zeitintervallen aufweist. Die Elemente mit der höchsten Erhöhungsrate werden als Trendelemente betrachtet. Die Zeit basiert auf Zeitstempeldaten in Ihrem Datensatz für Elementinteraktionen. Die betrachteten Elemente stammen aus den Interaktionsdaten, die Sie in großen Mengen und inkrementell importiert haben. Sie müssen keine neue Lösungsversion manuell erstellen, damit `Trending-Now` neue Elemente in Interaktionsdaten berücksichtigt.

Sie können das Zeitintervall angeben, indem Sie `Trend discovery frequency` bei der Erstellung Ihrer Lösung angeben. Wenn Sie beispielsweise `30 minutes` für `Trend discovery frequency` angeben, identifiziert Amazon Personalize für alle 30 Minuten an Daten die Elemente mit der größten Zunahme der Interaktionen seit der letzten Auswertung. Mögliche Häufigkeiten sind 30 Minuten, 1 Stunde, 3 Stunden und 1 Tag. Wählen Sie eine Häufigkeit aus, die der Verteilung Ihrer Interaktionsdaten entspricht. Fehlende Daten über das von Ihnen gewählte Intervall können die Empfehlungsgenauigkeit verringern. Wenn Sie in den letzten zwei Zeitintervallen keine Interaktionen importieren, empfiehlt Amazon Personalize nur beliebte Elemente, anstatt Elemente zu trenden.

Mit `Trending-Now` rufen Sie den [GetRecommendations](#) Vorgang auf oder erhalten Empfehlungen auf der Seite Testkampagne der Amazon-Personalize-Konsole. Amazon Personalize gibt die wichtigsten Trendelemente zurück. Sie übergeben einen `userId` in Ihrer Anfrage, wenn Sie einen Filter anwenden, der ihn erfordert. Mit der `GetRecommendations` API können Sie die Anzahl der Trendelemente konfigurieren, die mit dem `numResults` Parameter zurückgegeben werden. Mit dem `Trending-Now`-Rezept können Sie keine Batch-Empfehlungen erhalten.

Um `Trending-Now` verwenden zu können, müssen Sie einen Datensatz für Elementinteraktionen mit mindestens 1000 eindeutigen historischen Interaktionen und Ereignisinteraktionen zusammen erstellen (nach Filterung nach `eventType` und `eventValueThreshold`, falls angegeben). Beim Generieren von Empfehlungen für Trendelemente verwendet `Trending-Now` keine Daten in Element- oder Benutzerdatensätzen. Sie können Empfehlungen jedoch weiterhin basierend auf Daten in diesen Datensätzen filtern. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Themen

- [Eigenschaften und Hyperparameter](#)
- [Erstellen einer Lösung \(SDK für Python \(Boto3\)\)](#)
- [Beispiel für ein Jupyter-Notebook](#)

Eigenschaften und Hyperparameter

Das `Trending-Now`-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-trending-now`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-trending-now`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-trending-now-custom`

Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

In der folgenden Tabelle werden die Hyperparameter für das Trending-Now-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Hyperparameter für die Feature-Transformation	
Trend discovery frequency	<p>Geben Sie an, wie oft Amazon Personalize Ihre Interaktionsdaten auswertet und Trendelemente identifiziert. Wenn Sie beispielsweise 30 minutes für <code>Trend discovery frequency</code> angeben, identifiziert Amazon Personalize alle 30 Minuten Elemente mit der größten Zunahme der Interaktionen über 30-minütige Intervalle.</p> <p>Zu den verfügbaren Häufigkeiten gehören 30 Minuten, 1 Stunde, 3 Stunden und 1 Tag. Wählen Sie eine Häufigkeit aus, die der Verteilung Ihrer Interaktionsdaten entspricht. Fehlende Daten über das von Ihnen gewählte Intervall können die Empfehlungsgenauigkeit verringern. Wenn Sie keinen Wert angeben, wird standardmäßig alle 2 Stunden verwendet.</p> <p>Standardwert: 2 Stunden</p>

Name	Beschreibung
	Mögliche Werte: 30 Minuten, 1 Stunde, 3 Stunden und 1 Tag. Werttyp: Zeichenfolge HPO-optimierbar: Nein

Erstellen einer Lösung (SDK für Python (Boto3))

Der folgende Code zeigt, wie Sie eine Lösung mit dem Trending-Now-Rezept unter Verwendung des SDK for Python (Boto3) erstellen. Mögliche Werte für `trend_discovery_frequency` sind `30 minutes`, `1 hour`, `3 hours`, und `1 day`. Informationen zum Erstellen einer Lösung mit der Konsole finden Sie unter [Erstellen einer Lösung \(Konsole\)](#).

```
import boto3

personalize = boto3.client("personalize")

create_solution_response = personalize_client.create_solution(
    name="solution name",
    recipeArn="arn:aws:personalize::recipe/aws-trending-now",
    datasetGroupArn="dataset group ARN",
    solutionConfig={
        "featureTransformationParameters": {
            "trend_discovery_frequency": "1 hour"
        }
    }
)
print(create_solution_response['solutionArn'])
```

Beispiel für ein Jupyter-Notebook

Ein Beispiel für ein Jupyter-Notebook, das zeigt, wie das Trending-Now-Rezept verwendet wird, finden Sie unter [trending_now_example.ipynb](#) im Amazon-Personalize-Beispiel GitHub -Repository.

Popularity-Count-Rezept

Popularity-Count empfiehlt die beliebtesten Elemente auf der Grundlage Ihrer Interaktionsdaten. Die beliebtesten Elemente sind die Elemente mit den meisten Interaktionsdaten eindeutiger Benutzer.

Das Rezept gibt für alle Benutzer dieselben beliebten Elemente zurück. Popularity-Count ist eine gute Grundlage für den Vergleich mit anderen Rezepten unter Verwendung der Auswertungsmetriken, die Amazon Personalize generiert, wenn Sie eine Lösungsversion erstellen. Weitere Informationen finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#).

Nachdem Sie eine Lösungsversion erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungsversion und Daten auf dem neuesten Stand halten. Mit Popularity-Count müssen Sie manuell eine neue Lösungsversion erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungsversion aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Dieses vordefinierte Rezept besitzt die folgenden Eigenschaften:

- Name (Name – aws-popularity-count)
- Rezept-ARN – arn:aws:personalize:::recipe/aws-popularity-count
- Algorithmus-ARN – arn:aws:personalize:::algorithm/aws-popularity-count
- ARN der Feature-Transformation – arn:aws:personalize:::feature-transformation/sims
- Rezepttyp – USER_PERSONALIZATION

Popularity-Count hat keine offengelegten Hyperparameter.

PERSONALISIERTE RANGFOLGE

Das PERSONALIZED_RANKING-Rezept, Personalized-Ranking, enthält Empfehlungen in Rangfolge auf der Grundlage des prognostizierten Zinsniveaus.

[Personalisiertes Ranking](#)

Das Rezept für personalisiertes Ranking ist ein Rezept für ein hierarchisches wiederkehrendes neuronales Netzwerk (HRNN), mit dem auch Ergebnisse gefiltert und neu eingestuft werden können. Die personalisierte Rangfolge stellt eine Liste der besten Empfehlungen bereit. Verwenden Sie das Rezept für personalisiertes Ranking, wenn Sie die Ergebnisse für Ihre Nutzer personalisieren, z. B. bei personalisierten Suchergebnissen oder kuratierten Listen.

Um ein Modell zu trainieren, verwendet das Rezept für personalisiertes Ranking die Daten in Ihrem Datensatz mit Artikelinteraktionen und, falls Sie sie erstellt haben, den Datensätzen „Artikel“ und „Benutzer“ in Ihrer Datensatzgruppe.

Personalisiertes Ranking-Rezept

Das Personalized-Ranking-Rezept generiert personalisierte Ranglisten von Elementen. Eine personalisierte Rangfolge ist eine Liste empfohlener Elemente, die für einen bestimmten Benutzer neu nach Rang angeordnet sind. Dies ist nützlich, wenn Sie eine Sammlung von geordneten Elementen haben, z. B. Suchergebnisse, Werbeaktionen oder kuratierte Listen, und Sie für jeden Ihrer Benutzer eine personalisierte Neurangierung bereitstellen möchten. Mit Personalized-Ranking kann Amazon Personalize beispielsweise Suchergebnisse, die Sie mit generieren, neu sortieren [OpenSearch](#).

Um ein Modell zu trainieren, verwendet das Personalized-Ranking-Rezept die Daten in Ihrem Datensatz für Elementinteraktionen. Wenn Sie sie erstellt haben, werden der Datensatz Elemente und der Datensatz Benutzer in Ihrer Datensatzgruppe verwendet (diese Datensätze sind optional). Mit Personalized-Ranking kann Ihr Artikel-Datensatz [Unstrukturierte Textmetadaten](#) und Ihr Artikel-Interaktionen-Datensatz enthalten [Kontextuelle Metadaten](#). Verwenden Sie die [-GetPersonalizedRanking](#) API, um eine personalisierte Rangfolge zu erhalten.

Nachdem Sie eine Lösungsversion erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungsversion und Daten auf dem neuesten Stand halten. Mit Personalized-Ranking müssen Sie manuell eine neue Lösungsversion erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungsversion aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Note

Wenn Sie Elemente ohne Interaktionsdaten für die Rangfolge angeben, gibt Amazon Personalize diese Elemente ohne Empfehlungswert in der `GetPersonalizedRanking` API-Antwort zurück.

Das Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-personalized-ranking`)

- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-personalized-ranking`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-personalized-ranking`
- ARN der Feature-Transformation – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- Rezepttyp – `PERSONALIZED_RANKING`

Hyperparameter

Die folgende Tabelle beschreibt die Hyperparameter für das Personalisierte-Rangfolgen-Rezept. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Algorithmus-Hyperparameter	
<code>hidden_dimension</code>	Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Datensatz zu Elementinteraktionen kompliziertere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung. Bestimmen Sie den optimalen Wert mithilfe der HPO.

Name	Beschreibung
	<p>Um HPO zu verwenden, legen Sie <code>performHPO</code> auf <code>true</code> fest, wenn Sie die Operationen CreateSolutionVersion und CreateSolution aufrufen.</p> <p>Standardwert</p> <p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
bptt	<p>Legt fest, ob die Technik „Backpropagation über Zeit“ verwendet werden soll. Die Backpropagation über Zeit ist eine Technik, die Gewichtungen in wiederkehrenden neuronalen netzwerkbasierten Algorithmen aktualisiert. Verwenden Sie <code>bptt</code> für langfristige Guthaben, um verzögerte Belohnungen mit frühen Ereignissen zu verbinden. Eine verzögerte Belohnung kann beispielsweise ein Kauf nach mehreren Klicks sein. Ein frühes Ereignis kann ein erster Klick sein. Selbst bei denselben Ereignistypen wie einem Klick empfiehlt es sich, langfristige Effekte zu berücksichtigen und die Gesamtelohnungen zu maximieren. Um Langzeiteffekte zu berücksichtigen, verwenden Sie höhere <code>bptt</code>-Werte. Die Verwendung eines höheren <code>bptt</code>-Werts erfordert größere Datensets und mehr Zeit für die Verarbeitung.</p> <p>Standardwert: 32</p> <p>Bereich: [2, 32]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

Name	Beschreibung
<code>recency_mask</code>	<p>Bestimmt, ob das Modell die neuesten Beliebtheits-trends im Datensatz zu Elementinteraktionen berücksichtigen soll. Neueste Beliebtheits-trends können plötzliche Änderungen in den zugrunde liegenden Mustern von Interaktionsereignissen umfassen. Um ein Modell zu schulen, das aktuelle Ereignisse stärker gewichtet, legen Sie <code>recency_mask</code> auf <code>true</code> fest. Um ein Modell zu schulen, das alle vorherigen Interaktionen gleichermaßen abwägt, legen Sie <code>recency_mask</code> auf <code>false</code> fest. Um gute Empfehlungen mit gleicher Gewichtung zu erhalten, benötigen Sie möglicherweise ein größeres Schulungsdataset.</p> <p>Standardwert: <code>True</code></p> <p>Bereich: <code>True</code> oder <code>False</code></p> <p>Werttyp: Boolescher Wert</p> <p>HPO-optimierbar: Ja</p>

Hyperparameter für die Auslastung

Name	Beschreibung
<code>min_user_history_length_percentile</code>	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Historienlänge ist die Gesamtmenge der Daten zu einem Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen, da die Daten für diese Benutzer in der Regel Rauschen enthalten. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min__user_history_length_percentile to 0.05</code> und <code>max_user_history_length_percentile to 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0.99</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Beispiel-Notebook mit personalisiertem Rang

Ein Beispiel für ein Jupyter-Notebook, das zeigt, wie das Personalized-Ranking-Rezept verwendet wird, finden Sie unter [Personalize Ranking-Beispiel](#).

RELATED_ITEMS

Note

Alle RELATED_ITEMS-Rezepte verwenden Interaktionsdaten. Wählen Sie das Rezept für ähnliche Artikel, wenn Sie auch Artikelmetadaten haben und möchten, dass Amazon Personalize diese verwendet, um ähnliche Artikel zu finden. Oder wählen Sie das SIMS-Rezept, wenn Sie mehr Hyperparameter für das Modell konfigurieren möchten.

Die RELATED_ITEMS-Rezepte geben Elemente zurück, die einem Element ähneln, das Sie angeben, wenn Sie Empfehlungen erhalten. Die RELATED_ITEMS-Rezepte lauten wie folgt:

- [Ähnliche Elemente-Rezepte](#)
- [SIMS-Rezept](#)

[Ähnliche Artikel](#)

Das Rezept „Ähnliche Artikel“ generiert Empfehlungen für Artikel, die einem von Ihnen angegebenen Artikel ähnlich sind. Es berechnet die Ähnlichkeit sowohl auf der Grundlage von Interaktionsdaten als auch, falls Sie diese angeben, auf Artikelmetadaten. Wenn Amazon Personalize die Artikelnummer, die Sie in Ihrer Empfehlungsanfrage angegeben haben, nicht finden kann, gibt das Rezept beliebte Artikel als Empfehlungen zurück. Similar-Items verwendet bei der Generierung von Empfehlungen keine Daten aus einem Benutzerdatensatz. Sie können jedoch weiterhin Empfehlungen auf der Grundlage von Daten in einem Benutzerdatensatz filtern. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

[SIMS](#)

Das Rezept für item-to-item Ähnlichkeiten (SIMS) generiert Elemente, die einem bestimmten Objekt ähnlich sind, und zwar auf der Grundlage des gleichzeitigen Auftretens des Elements in der Benutzerhistorie in Ihrem Datensatz mit Artikelinteraktionen. Wenn keine ausreichenden Benutzerverhaltensdaten für ein Element verfügbar sind oder wenn die angegebene Element-ID nicht gefunden wird, gibt das Rezept beliebte Elemente als Empfehlungen zurück.

Ähnliche Elemente-Rezepte

Note

Alle RELATED_ITEMS-Rezepte verwenden Interaktionsdaten. Wählen Sie Ähnliche Elemente, wenn Sie auch Elementmetadaten haben und möchten, dass Amazon Personalize sie verwendet, um ähnliche Elemente zu finden. Oder wählen Sie die aus [SIMS-Rezept](#), wenn Sie weitere Hyperparameter für das Modell konfigurieren möchten.

Das Rezept Ähnliche Elemente (aws-similar-items) generiert Empfehlungen für Elemente, die einem von Ihnen angegebenen Element ähneln. Verwenden Sie ähnliche Elemente, um Kunden dabei zu helfen, neue Elemente in Ihrem Katalog auf der Grundlage ihres vorherigen Verhaltens und ihrer Elementmetadaten zu entdecken. Die Empfehlung ähnlicher Elemente kann das Benutzerinteraktionen, die Klickrate und die Konvertierungsrate für Ihre Anwendung erhöhen.

Ähnliche Elemente berechnet die Ähnlichkeit auf der Grundlage von Interaktionsdaten und allen von Ihnen bereitgestellten Elementmetadaten. Es berücksichtigt das gleichzeitige Auftreten des Elements in Benutzerverläufen in Ihrem Interaktionsdatensatz und alle Ähnlichkeiten der Elementmetadaten. Bei ähnlichen Elementen könnte Amazon Personalize beispielsweise Artikel empfehlen, die Kunden häufig zusammen mit einem ähnlichen Stil ([Kategorische Metadaten](#)) gekauft haben, oder Filme, die verschiedene Benutzer ebenfalls mit einer ähnlichen Beschreibung () angesehen haben [Unstrukturierte Textmetadaten](#).

Bei ähnlichen Elementen geben Sie eine Element-ID in einer [-GetRecommendations](#) Operation (oder der Amazon-Personalize-Konsole) an und Amazon Personalize gibt eine Liste ähnlicher Elemente zurück. Oder Sie können einen Batch-Workflow verwenden, um ähnliche Elemente für alle Elemente in Ihrem Bestand abzurufen (siehe [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#)). Wenn Sie ähnliche Elemente erhalten, können Sie die Elemente basierend auf einem Attribut des Elements filtern, das Sie in Ihrer Anforderung angeben. Dazu fügen Sie Ihrem Filter ein `CurrentItem.attribute`-Element hinzu. Ein Beispiel finden Sie unter [item data filter examples](#).

Um Ähnliche Elemente zu verwenden, müssen Sie einen Datensatz für Elementinteraktionen mit mindestens 1000 eindeutigen historischen Interaktionen und Ereignisinteraktionen (zusammen) erstellen. Für genauere Vorhersagen empfehlen wir Ihnen, auch einen Artikel-Datensatz zu erstellen und Metadaten zu Artikeln in Ihren Katalog zu importieren. Ähnliche Elemente verwenden beim Generieren von Empfehlungen keine Daten in einem Benutzerdatensatz. Sie können Empfehlungen

weiterhin basierend auf Daten in einem Benutzerdatensatz filtern. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wenn Sie einen Elementdatensatz mit Textdaten und Elementtiteldaten haben, können Sie Designs für verwandte Elemente in Batch-Empfehlungen generieren. Weitere Informationen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Sie können Empfehlungen für Elemente erhalten, die einem kalten Element ähneln (einem Element mit weniger als fünf Interaktionen). Wenn Amazon Personalize die Element-ID, die Sie in Ihrer Empfehlungsanforderung oder Batch-Eingabedatei angeben, nicht finden kann, gibt das Rezept beliebte Elemente als Empfehlungen zurück.

Nachdem Sie eine Lösungsversion erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungsversion und Daten auf dem neuesten Stand halten. Bei ähnlichen Elementen müssen Sie manuell eine neue Lösungsversion erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungsversion aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Eigenschaften und Hyperparameter

Das Rezept für ähnliche Elemente hat die folgenden Eigenschaften:

- Name (Name – `aws-similar-items`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-similar-items`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-similar-items`

Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

In der folgenden Tabelle werden die Hyperparameter für das Rezept Ähnliche Elemente beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]

- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Algorithmus-Hyperparameter	
<code>popularity_discount_factor</code>	<p>Konfigurieren Sie, wie sich die Popularität auf Empfehlungen auswirkt. Geben Sie einen Wert näher Null an, um beliebtere Elemente einzuschließen. Geben Sie einen Wert an, der näher an eins liegt, um weniger Wert auf die Beliebtheit zu legen.</p> <p>Standardwert: 0.0</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>
<code>item_id_hidden_dim</code>	<p>Die Anzahl der ausgeblendeten Variablen, die Amazon Personalize verwendet, um Element-ID-Einbettungen basierend auf Interaktionsdaten zu modellieren. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Um zu verwenden <code>item_id_hidden_dim</code>, müssen Sie HPO verwenden und Werte für den minimalen und maximalen Bereich angeben. Amazon Personalize verwendet HPO, um den besten Wert innerhalb des von Ihnen angegebenen Bereichs zu finden. Geben Sie einen größeren Maximalwert an, wenn Sie einen großen Datensatz für Elementinteraktionen haben. Die Verwendung eines größeren Maximalwerts erfordert mehr Zeit für die Verarbeitung.</p>

Name	Beschreibung
	<p>Um HPO zu verwenden, setzen Sie <code>performHPO</code> auf <code>true</code> wenn Sie die -CreateSolution Operation aufrufen.</p> <p>Standardwert: 100</p> <p>Bereich: [30, 200]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
<code>item_metadata_hidden_dim</code>	<p>Die Anzahl der ausgeblendeten Variablen, die Amazon Personalize zum Modellieren von Elementmetadaten verwendet. Um zu verwenden <code>item_metadata_hidden_dim</code>, müssen Sie HPO verwenden und Werte für den minimalen und maximalen Bereich angeben. Amazon Personalize verwendet HPO, um den besten Wert innerhalb des von Ihnen angegebenen Bereichs zu finden. Geben Sie einen größeren Maximalwert an, wenn Sie einen großen Datensatz für Elementinteraktionen haben. Die Verwendung eines größeren Maximums erfordert mehr Zeit für die Verarbeitung.</p> <p>Um HPO zu verwenden, setzen Sie <code>performHPO</code> auf <code>true</code> wenn Sie die -CreateSolution Operation aufrufen.</p> <p>Standardwert: 100</p> <p>Bereich: [30, 200]</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>

SIMS-Rezept

Note

Alle RELATED_ITEMS-Rezepte verwenden Interaktionsdaten. Wählen Sie SIMS, wenn Sie weitere Hyperparameter für das Modell konfigurieren möchten. Wählen Sie die aus [Ähnliche Elemente-Rezepte](#), wenn Sie Elementmetadaten haben und möchten, dass Amazon Personalize sie verwendet, um ähnliche Elemente zu finden.

Das Item-to-item Ähnlichkeiten (SIMS)-Rezept verwendet kollaborative Filterung, um Elemente zu empfehlen, die einem Element am ähnlichsten sind, das Sie angeben, wenn Sie Empfehlungen erhalten. SIMS verwendet Ihren Artikelinteraktionsdatensatz, nicht Elementmetadaten wie Farbe oder Preis, um die Ähnlichkeit zu ermitteln. SIMS identifiziert das gleichzeitige Auftreten des Elements in Benutzerverläufen in Ihrem Interaktionsdatensatz, um ähnliche Elemente zu empfehlen. Mit SIMS könnte Amazon Personalize beispielsweise empfehlen, Shop-Artikel zu probieren, die Kunden häufig zusammen gekauft haben, oder Filme, die verschiedene Benutzer ebenfalls angesehen haben.

Wenn Sie ähnliche Elementempfehlungen erhalten, können Sie die Elemente basierend auf einem Attribut des Elements filtern, das Sie in Ihrer Anforderung angeben. Dazu fügen Sie Ihrem Filter ein `CurrentItem.attribute`-Element hinzu. Ein Beispiel finden Sie unter [item data filter examples](#).

Um SIMS verwenden zu können, müssen Sie einen Datensatz mit Elementinteraktionen mit mindestens 1000 eindeutigen historischen Interaktionen und Ereignisinteraktionen (zusammen) erstellen. SIMS verwendet beim Generieren von Empfehlungen keine Daten in einem Benutzer- oder Elementdatensatz. Sie können Empfehlungen weiterhin basierend auf Daten in diesen Datensätzen filtern. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wenn nicht genügend Benutzerverhaltensdaten für ein Element vorhanden sind oder die von Ihnen angegebene Element-ID nicht gefunden wird, empfiehlt SIMS beliebte Elemente. Nachdem Sie eine Lösungsversion erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungsversion und Daten auf dem neuesten Stand halten. Mit SIMS müssen Sie manuell eine neue Lösungsversion erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungsversion aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Das SIMS-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-sims`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-sims`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-sims`
- ARN der Feature-Transformation – `arn:aws:personalize:::feature-transformation/sims`
- Rezepttyp – `RELATED_ITEMS`

In der folgenden Tabelle werden die Hyperparameter für das SIMS-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden. Die Auswahl des besten Werts für einen Hyperparameter wird als Hyperparameteroptimierung (HPO) bezeichnet. Weitere Informationen finden Sie unter [Hyperparameter und HPO](#).

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO optimierbar: Kann der Parameter an der Hyperparameteroptimierung (HPO) teilnehmen?

Name	Beschreibung
Algorithmus-Hyperparameter	
<code>popularity_discount_factor</code>	<p>Konfigurieren Sie, wie sich die Popularität auf Empfehlungen auswirkt. Geben Sie einen Wert näher Null an, um beliebtere Elemente einzuschließen. Geben Sie einen Wert an, der näher an eins liegt, um weniger Wert auf die Beliebtheit zu legen.</p> <p>Standardwert: 0.5</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p>

Name	Beschreibung
	HPO-optimierbar: Ja
min_cointeraction_count	<p>Die Mindestanzahl von Co-Interaktionen, die Sie benötigen, um die Ähnlichkeit zwischen einem Elementpaar zu berechnen. Ein Wert von 3 bedeutet beispielsweise, dass Sie drei oder mehr Benutzer benötigen, die mit beiden Elementen interagiert haben, um ihre Ähnlichkeit zu berechnen.</p> <p>Standardwert: 3</p> <p>Bereich: [0, 10].</p> <p>Werttyp: Ganzzahl</p> <p>HPO-optimierbar: Ja</p>
Hyperparameter für die Auslastung	

Name	Beschreibung
min_user_history_length_percentile	<p>Das Mindestperzentil der Benutzerhistorienlängen, das in die Modellschulung einbezogen werden muss. Die Historienlänge ist die Gesamtmenge der verfügbaren Daten für einen Benutzer. Verwenden Sie <code>min_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit kurzen Historienlängen auszuschließen. Benutzer mit einer kurzen Historie zeigen häufig Muster basierend auf der Beliebtheit von Elementen anstelle der persönlichen Bedürfnisse oder Wünsche des Benutzers. Wenn Sie diese entfernen , können Sie Modelle schulen, die sich stärker auf die zugrunde liegenden Muster in Ihren Daten konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Standardwert: 0.005</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_user_history_length_percentile</code>	<p>Das maximale Perzentil der Benutzerhistorienlänge, das in die Modellschulung einbezogen werden kann. Die Historienlänge ist die Gesamtmenge der verfügbaren Daten für einen Benutzer. Verwenden Sie <code>max_user_history_length_percentile</code> , um einen Prozentsatz von Benutzern mit langen Historienlängen auszuschließen. Benutzer mit einer langen Historie neigen dazu, Rauschen zu erzeugen. Beispielsweise kann ein Roboter über eine lange Liste automatisierter Interaktionen verfügen. Durch das Entfernen dieser Benutzer werden Störungen der Schulung eingeschränkt. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Benutzer beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_hist_length_percentile = 0.05</code> und <code>max_hist_length_percentile = 0.95</code> festlegen, werden alle Benutzer berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Historienlängen.</p> <p>Standardwert: 0,995</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>min_item_interaction_count_percentile</code>	<p>Das minimale Perzentil der Element-Interaktionsanzahl, das in die Modellschulung einbezogen werden kann. Verwenden Sie <code>min_item_interaction_count_percentile</code>, um einen Prozentsatz von Elementen mit einer kurzen Historie von Interaktionen auszuschließen. Elemente mit einer kurzen Historie sind häufig neue Elemente. Wenn Sie sie entfernen, können Sie Modelle schulen, die sich stärker auf Elemente mit einer bekannten Historie konzentrieren. Wählen Sie einen geeigneten Wert aus, nachdem Sie die Längen der Benutzerhistorien mithilfe eines Histogramms oder eines ähnlichen Tools überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Elemente beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Standardwert: 0.01</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

Name	Beschreibung
<code>max_item_interaction_count_percentile</code>	<p>Das maximale Perzentil der Element-Interaktionen zählt, um in die Modellschulung einbezogen zu werden. Verwenden Sie <code>max_item_interaction_count_percentile</code>, um einen Prozentsatz von Elementen mit einer langen Historie von Interaktionen auszuschließen. Elemente mit einer langen Historie sind in der Regel älter und können veraltet sein. Beispiel: Eine Film-DVD, die vergriffen ist. Das Entfernen dieser Elemente führt dazu, dass man sich auf relevantere Elemente konzentriert. Wählen Sie einen geeigneten Wert, nachdem Sie die Längen der Benutzerhistorien mit einem Histogramm oder einem ähnlichen Tool überprüft haben. Wir empfehlen, den Wert so festzulegen, dass die meisten Elemente beibehalten, aber die Grenzfälle entfernt werden.</p> <p>Wenn Sie beispielsweise <code>min_item_interaction_count_percentile = 0.05</code> und <code>max_item_interaction_count_percentile = 0.95</code> festlegen, werden alle Elemente berücksichtigt, mit Ausnahme der unteren und oberen 5 % in Bezug auf ihre Interaktionsanzahl.</p> <p>Standardwert: 0.9</p> <p>Bereich: [0,0, 1,0].</p> <p>Werttyp: Gleitkommazahl</p> <p>HPO-optimierbar: Nein</p>

SIMS-Beispiel-Notebook

Ein Beispiel eines Jupyter-Notizbuchs zur Verwendung des SIMS-Rezepts finden Sie unter [Finding similar items + HPO \(Suchen von ähnlichen Elementen + HPO\)](#).

PERSONALISIERTE_AKTIONEN

Verwenden Sie ein Rezept für `PERSONALIZED_ACTIONS`, um Ihren Benutzern in Echtzeit die nächstbeste Aktion zu empfehlen, z. B. sich für Ihr Treueprogramm anzumelden, Ihre App herunterzuladen oder eine Kreditkarte zu beantragen. Wenn Sie die nächstbeste Aktion empfehlen, können Sie die Kundenbindung erhöhen, mehr Umsatz generieren und das Nutzererlebnis verbessern.

Die Rezepte von `PERSONALIZED_ACTIONS` lauten wie folgt:

- [Nächstes Best-Action-Rezept](#)

Nächstes Best-Action-Rezept

Das Next-Best-Action (`aws-next-best-action`)-Rezept generiert Empfehlungen in Echtzeit für die nächstbesten Aktionen für Ihre Benutzer. Die nächste beste Aktion für einen Benutzer ist die Aktion, die er höchstwahrscheinlich ergreifen wird. Zum Beispiel die Registrierung für Ihr Treueprogramm, das Herunterladen Ihrer App oder das Anfordern einer Kreditkarte.

Mit Next-Best-Action können Sie Ihren Benutzern personalisierte Aktionsempfehlungen geben, während sie Ihre Anwendung verwenden. Das Vorschlagen der richtigen Aktion für einen Benutzer kann dazu führen, dass mehr Benutzer Ihre Aktionen ausführen. Abhängig von den Aktionen, die Sie empfehlen möchten, können Sie die Kundenbindung erhöhen, mehr Umsatz generieren und das Benutzererlebnis Ihrer Anwendung verbessern. Ein Anwendungsfallbeispiel, das beschreibt, wie personalisierte Aktionsempfehlungen einer E-Commerce-Anwendung zugute kommen können, finden Sie unter [Anwendungsfallbeispiel](#).

Amazon Personalize prognostiziert die nächstbeste Aktion aus den Aktionen, die Sie in Ihren Aktionsdatensatz importieren. Es identifiziert die Aktionen, die ein Benutzer höchstwahrscheinlich auf der Grundlage seiner Interaktionen mit Aktionen und Elementen ausführen wird. Wenn Ihre Aktionsdaten den Wert der Aktion enthalten, wird Amazon Personalize für den Wert der Aktion berücksichtigt. Wenn es ebenso wahrscheinlich ist, dass ein Benutzer zwei verschiedene Aktionen ausführt, stuft Amazon Personalize die Aktion mit dem höheren Wert ein. Weitere Informationen zu den Daten, die das Next-Best-Action-Rezept verwendet, finden Sie unter [Erforderliche und optionale Datensätze](#).

Wenn Sie Empfehlungen für Echtzeitaktionen für einen Benutzer erhalten, gibt Amazon Personalize eine Liste der Aktionen zurück, die der Benutzer höchstwahrscheinlich innerhalb eines konfigurierbaren Zeitraums (`action optimization period`) ausführen wird. Zum Beispiel

die Aktionen, die sie in den nächsten 14 Tagen höchstwahrscheinlich ergreifen werden. Die Liste wird in absteigender Reihenfolge nach Wahrscheinlichkeitswert sortiert. Dieser Wert gibt die Wahrscheinlichkeit an, dass der Benutzer die Aktion ausführt.

Wenn Sie eine Lösung mit dem Next-Best-Action-Rezept erstellen, können Sie das Zeitfenster konfigurieren, das Amazon Personalize bei der Vorhersage von Aktionen verwendet, indem Sie den Funktionsparameter `action optimization period`-Hyperparameter verwenden. Weitere Informationen finden Sie unter [Eigenschaften und Hyperparameter](#).

Themen

- [Anwendungsfallbeispiel](#)
- [Rezept-Features](#)
- [Erforderliche und optionale Datensätze](#)
- [Eigenschaften und Hyperparameter](#)

Anwendungsfallbeispiel

Das Vorschlagen der richtigen Aktion für einen Benutzer kann dazu führen, dass mehr Benutzer Ihre Aktionen ausführen. Abhängig von den Aktionen, die Sie empfehlen möchten, können Sie möglicherweise die Kundenbindung erhöhen, mehr Umsatz generieren und das Benutzererlebnis Ihrer Anwendung verbessern.

Sie könnten beispielsweise eine E-Commerce-Anwendung haben, die die folgenden verschiedenen Aktionen vorschlägt:

- Abonnieren eines Treueprogramms
- Herunterladen der mobilen App
- Kaufen in der Kategorie Jewelry
- Kaufen in der Kategorie Bol und Pflege

Möglicherweise haben Sie einen Benutzer, der häufig an Ihrem Standort arbeitet und wiederholt die Kaufaktionen Jewelry und Bol und Grooming ausgeführt hat. Für diesen Benutzer können die Empfehlungen für Amazon-Personalize-Aktionen und ihre Ergebnisse Folgendes beinhalten:

- Abonnieren eines Treueprogramms

Propensity-Score – 1,00

- Kaufen in der Kategorie Jewelry

Propensity-Score – 0,86

- Kaufen der Kategorie in Bol und Pflege

Propensity-Score – 0,85

Mit diesen Aktionsempfehlungen wissen Sie, dass Sie den Benutzer auffordern, sich für Ihr Treueprogramm anzumelden. Diese Aktion hat den höchsten Propensitätswert und ist die Aktion, die der Benutzer höchstwahrscheinlich ausführen wird. Dies liegt daran, dass der Benutzer häufig in Ihrem Geschäft vorkommt und wahrscheinlich die Vorteile Ihres Treueprogramms nutzen wird.

Rezept-Features

Das Next-Best-Action-Rezept verwendet beim Generieren von Aktionsempfehlungen die folgenden Amazon-Personalize-Rezeptfunktionen:

- **Echtzeit-Personalisierung:** Amazon Personalize verwendet Echtzeit-Personalisierung, um Aktionsempfehlungen entsprechend dem sich entwickelnden Interesse eines Benutzers zu aktualisieren und anzupassen. Weitere Informationen finden Sie unter [Personalisierung in Echtzeit](#).
- **Exploration:** Bei der Erkundung umfassen Empfehlungen neue Aktionen oder Aktionen mit weniger Interaktionsdaten. Weitere Informationen zur Erkundung finden Sie unter [Exploration \(Erkundung\)](#).
- **Automatische Updates:** Mit automatischen Updates aktualisiert Amazon Personalize automatisch alle zwei Stunden das neueste Modell (Lösungsversion), um neue Aktionen während der Untersuchung in Empfehlungen aufzunehmen. Weitere Informationen finden Sie unter [Automatische Updates](#).

Erforderliche und optionale Datensätze

Um das Next-Best-Action-Rezept zu verwenden, müssen Sie die folgenden Datensätze erstellen:

- **Aktionen:** Sie importieren Daten zu Ihren Aktionen, z. B. deren Wert, in einen Amazon-Personalize-Actions-Datensatz.

In Ihren Aktionsdaten können Sie für jede Aktion einen EXPIRATION_TIMESTAMP angeben. Wenn eine Aktion abgelaufen ist, nimmt Amazon Personalize sie nicht in Empfehlungen auf. Sie können auch für jede Aktion ein REPEAT_FREQUENCY angeben. Dies gibt an, wie lange Amazon Personalize warten soll, bevor eine Aktion erneut empfohlen wird, nachdem ein Benutzer damit

interagiert hat. Informationen zu den Daten, die ein Aktionsdatensatz speichern kann, finden Sie unter [Datensatz „Aktionen“](#).

- Elementinteraktionen: Ihr Datensatz für Elementinteraktionen muss mindestens 1000 Elementinteraktionen haben. Amazon Personalize verwendet Elementinteraktionen, um den aktuellen Status Ihrer Benutzer und ihre Interessen zu verstehen. Informationen zu den Daten zu Elementinteraktionen finden Sie unter [Datensatz mit Interaktionen zwischen Elementen](#).

Die folgenden Datensätze sind optional:

- Datensatz für Aktionsinteraktionen: Eine Aktionsinteraktion ist eine Interaktion, an der ein Benutzer und eine Aktion in Ihrem Aktionsdatensatz beteiligt sind. Sie können Interaktionen mit ergriffenen, nicht durchgeführten und angezeigten Aktionen importieren. Obwohl diese Daten optional sind, empfehlen wir Ihnen, Aktionsinteraktionsdaten für Qualitätsempfehlungen zu importieren. Wenn Sie keine Aktionsinteraktionsdaten haben, können Sie einen leeren Datensatz für Aktionsinteraktionen erstellen und die Interaktionen Ihrer Kunden mit Aktionen mithilfe der [PutActionInteractions](#) API-Operation aufzeichnen.

Bis Sie Aktionsinteraktionsdaten importieren, empfiehlt Amazon Personalize Aktionen in Ihrem Aktionsdatensatz ohne Personalisierung und die Wahrscheinlichkeitswerte liegen bei 0,0.

Informationen zu den Aktionsinteraktionsdaten, die Sie importieren können, finden Sie unter [Datensatz „Aktionen“: Interaktionen](#). Informationen zur Aufzeichnung von Aktionsinteraktionsereignissen finden Sie unter [Aufzeichnen von Aktionsinteraktionsereignissen](#).

Note

Mit Next-Best-Action verwendet Amazon Personalize keine Impressionsdaten oder Kontextmetadaten in einem Aktionsinteraktionsdatensatz.

- Benutzer: Amazon Personalize verwendet alle Daten in Ihrem Benutzerdatensatz, um Ihre Benutzer und ihre Interessen besser zu verstehen. Sie können auch Daten in einem Benutzerdatensatz verwenden, um Aktionsempfehlungen zu filtern. Informationen zu den Benutzerdaten, die Sie importieren können, finden Sie unter [Benutzerdatensatz](#).
- Elemente: Amazon Personalize verwendet alle Daten in Ihrem Elementdatensatz zusammen mit Ihrem Elementinteraktionsdatensatz, um Verbindungen und Muster in ihrem Verhalten zu identifizieren. Dies hilft Amazon Personalize, Ihre Benutzer und ihre Interessen zu

verstehen. Informationen zu den Elementdaten, die Sie importieren können, finden Sie unter [Elementedatensatz](#).

Eigenschaften und Hyperparameter

Das Next-Best-Action-Rezept unterstützt keine Hyperparameter-Optimierung. Das Next-Best-Action-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-next-best-action`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-next-best-action`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-next-best-action`

In der folgenden Tabelle werden die Feature-Hyperparameter für das `aws-next-best-action` Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Hyperparameter zur Entwicklung von Funktionen steuern, wie die Daten für die Schulung gefiltert werden.

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)
- HPO-Tunable: Ob der Parameter an HPO teilnehmen kann

Name	Beschreibung
Hyperparameter für die Auslastung	
<code>action_optimization_period</code>	<p>Das Zeitfenster, das Amazon Personalize bei der Vorhersage der nächstbesten Aktionen für einen Benutzer verwendet. Zum Beispiel die Aktionen, die der Benutzer in den nächsten 14 Tagen wahrscheinlich ausführen wird.</p> <p>Wenn Sie nicht viele Aktionsinteraktionsdaten haben, geben Sie einen größeren Wert an. Wenn Sie sich</p>

Name	Beschreibung
	nicht sicher sind, welchen Wert Sie angeben sollen, verwenden Sie den Standardwert. Standardwert: 14 Bereich: [7, 28] Werttyp: Ganzzahl HPO-optimierbar: Nein

USER_SEGMENTIERUNG

USER_SEGMENTATION-Rezepte generieren Benutzersegmente auf der Grundlage von Artekeleingabedaten. Jedes Benutzersegment wird in absteigender Reihenfolge sortiert, basierend auf der Wahrscheinlichkeit, dass jeder Benutzer mit Artikeln in Ihrem Inventar interagiert. Verwenden Sie ein USER_SEGMENTATION-Rezept, um Benutzersegmente zu erstellen, die aufgrund ihrer Einstellungen für Artikel oder Artikelattribute höchstwahrscheinlich mit Ihrem Katalog interagieren werden. Möglicherweise möchten Sie beispielsweise eine gezielte Marketingkampagne für Nutzer erstellen, die sich höchstwahrscheinlich einen bestimmten Film ansehen oder ein bestimmtes Produkt nach Marke kaufen würden.

[Affinität zum Artikel](#)

Das Item-Affinity (aws-item-affinity) -Rezept ist ein USER_SEGMENTATION-Rezept, das für jedes von Ihnen angegebene Element ein Benutzersegment erstellt.

Um ein Modell zu trainieren, verwendet das Item-Affinity-Rezept die Datensätze Interactions und Items in Ihrer Datensatzgruppe. [Um Benutzersegmente zu erstellen, trainieren Sie eine Lösungsversion mit dem Item-Affinity-Rezept und erstellen dann einen Batch-Segment-Job.](#)

[Artikel-Attribut-Affinität](#)

Das Rezept Item-Attribute-Affinity (aws-item-attribute-affinity) ist ein USER_SEGMENTATION-Rezept, das für jedes von Ihnen angegebene Artikelattribut ein Benutzersegment erstellt.

Um ein Modell zu trainieren, verwendet das Item-Attribute-Affinity-Rezept den Interactions-Datensatz und den Artikel-Datensatz aus einer Datensatzgruppe. [Um Benutzersegmente zu erstellen, trainieren](#)

[Sie eine Lösungsversion mit dem Item-Attribute-Affinity-Rezept und erstellen dann einen Batch-Segmentjob.](#)

Item-Affinity-Rezept

Das Item-Affinity (aws-item-affinity)-Rezept ist ein USER_SEGMENTATION-Rezept, das ein Benutzersegment (Benutzergruppe) für jedes von Ihnen angegebene Element erstellt. Dies sind die Benutzer, von denen Amazon Personalize voraussagt, dass sie höchstwahrscheinlich mit jedem Element interagieren werden. Verwenden Sie Item-Affinity, um mehr über Ihre Benutzer zu erfahren und Aktionen basierend auf ihren jeweiligen Benutzersegmenten durchzuführen.

Sie können beispielsweise eine Marketingkampagne für Ihre Einzelhandelsanwendung erstellen, die auf den Benutzereinstellungen für Artikel in Ihrem Katalog basiert. Die Element-Affinität würde für jedes Element ein Benutzersegment erstellen, das auf Daten in Ihren Interaktionen- und Elementdatensätzen basiert. Sie können dies verwenden, um verschiedene Elemente auf der Grundlage der Wahrscheinlichkeit, dass sie eine Aktion ausführen, zu verschiedenen Benutzersegmenten hochzustufen (z. B. auf ein Element klicken oder ein Element kaufen). Andere Verwendungsmöglichkeiten können z. B. verkaufsübergreifende Produkte für verschiedene Benutzergruppen oder die Identifizierung potenzieller Stellenbewerber sein.

Um Benutzersegmente auf der Grundlage von Elementen zu erhalten, erstellen Sie eine Lösung und eine Lösungsversion mit dem Element-Affinitäts-Rezept, fügen dann eine Liste von Elementen im JSON-Format zu einem Amazon S3-Bucket hinzu und erstellen einen [Batch-Segmentauftrag](#). Amazon Personalize gibt für jedes Element ein Benutzersegment an Ihren Ausgabespeicherort in Amazon S3 aus. Ihre Eingabedaten können maximal 500 Elemente enthalten, für die Sie Benutzersegmente abrufen können. Informationen zum Vorbereiten von Eingabedaten für einen Batch-Segmentauftrag finden Sie unter [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#).

Sie müssen über einen Datensatz für Elementinteraktionen verfügen, um die Elementffinität verwenden zu können. Elemente und Benutzer-Datensätze sind optional. Sie können Benutzersegmente mit Batch-Segmentaufträgen abrufen. Weitere Informationen finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

Nachdem Sie eine Lösungsversion erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungsversion und Daten auf dem neuesten Stand halten. Mit Item-Affinity müssen Sie manuell eine neue Lösungsversion erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungsversion aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#). Um ein

Benutzersegment für ein Element abzurufen, muss das Element beim Erstellen der Lösungsversion vorhanden sein.

Das Item-Affinity-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-item-affinity`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-item-affinity`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-item-affinity`
- ARN der Feature-Transformation – `arn:aws:personalize:::feature-transformation/item-affinity`
- Rezepttyp – `USER_SEGMENTATION`

In der folgenden Tabelle werden die Hyperparameter für das Item-Affinity-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Sie können die Hyperparameteroptimierung (HPO) nicht mit dem Item-Affinity-Rezept verwenden.

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)

Name	Beschreibung
Algorithmus-Hyperparameter	
<code>hidden_dimension</code>	Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Interaktionen-Dataset komplexere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung.

Name	Beschreibung
	Standardwert
	Bereich: [32, 256]
	Werttyp: Ganzzahl

Item-Attribute-Affinity-Rezept

Das Item-Attribute-Affinity (aws-item-attribute-affinity)-Rezept ist ein USER_SEGMENTATION-Rezept, das ein Benutzersegment (Benutzergruppe) für jedes von Ihnen angegebene Elementattribut erstellt. Dies sind die Benutzer, von denen Amazon Personalize voraussagt, dass sie höchstwahrscheinlich mit Elementen mit dem bestimmten Attribut interagieren werden. Verwenden Sie Item-Attribute-Affinity, um mehr über Ihre Benutzer zu erfahren und Aktionen basierend auf ihren jeweiligen Benutzersegmenten durchzuführen.

Beispielsweise möchten Sie vielleicht eine Marketingkampagne für Ihre Einzelhandelsanwendung erstellen, die auf den Benutzereinstellungen für Warteschlangentypen in Ihrem Katalog basiert. Item-Attribute-Affinity würde ein Benutzersegment für jede auf dem Nachrichtentyp basierende Daten in Ihren Interaktionen- und Elementdatensätzen erstellen. Sie könnten dies verwenden, um verschiedene Telefonie für verschiedene Benutzersegmente hochzustufen, basierend auf der Wahrscheinlichkeit, dass sie Maßnahmen ergreifen werden (z. B. auf einen Arzt oder den Kauf eines Arzts). Andere Verwendungsmöglichkeiten können die Förderung verschiedener Filmgenres für verschiedene Benutzer oder die Identifizierung potenzieller Stellenbewerber basierend auf dem Auftragstyp sein.

Um Benutzersegmente basierend auf Elementattributen abzurufen, erstellen Sie eine Lösung und eine Lösungsversion mit dem Elementattribut-Affinitätsrezept, fügen dann eine Liste von Elementattributen im JSON-Format zu einem Amazon S3-Bucket hinzu und erstellen einen [Batch-Segmentauftrag](#). Amazon Personalize gibt für jedes Element ein Benutzersegment an Ihren Ausgabespeicherort in Amazon S3 aus. Ihre Eingabedaten können maximal 10 Abfragen haben, wobei jede Abfrage ein oder mehrere Elementattribute umfasst. Informationen zum Vorbereiten von Eingabedaten für einen Batch-Segmentauftrag finden Sie unter [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#).

Sie müssen über einen Datensatz für Elementinteraktionen und einen Datensatz für Elemente verfügen, um Item-Attribute-Affinität verwenden zu können. Ihr Elementdatensatz muss mindestens

eine Spalte enthalten, die eine nicht textbasierte, nicht reservierte Metadaten­spalte ist. Sie können Benutzersegmente mit Batch-Segmentaufträgen abrufen. Weitere Informationen finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

Nachdem Sie eine Lösungs­version erstellt haben, stellen Sie sicher, dass Sie Ihre Lösungs­version und Daten auf dem neuesten Stand halten. Mit Item-Attribute-Affinity müssen Sie manuell eine neue Lösungs­version erstellen (das Modell neu trainieren), damit Amazon Personalize neue Elemente für Empfehlungen berücksichtigt und das Modell mit dem neuesten Verhalten Ihres Benutzers aktualisiert. Anschließend müssen Sie jede Kampagne mit der Lösungs­version aktualisieren. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#). Um ein Benutzersegment für ein Elementattribut abzurufen, muss das Elementattribut beim Erstellen der Lösungs­version vorhanden sein.

Das Item-Attribute-Affinity-Rezept hat die folgenden Eigenschaften:

- Name (Name – `aws-item-attribute-affinity`)
- Amazon-Ressourcenname (ARN) für Rezepte – `arn:aws:personalize:::recipe/aws-item-attribute-affinity`
- Algorithmus-ARN – `arn:aws:personalize:::algorithm/aws-item-attribute-affinity`
- ARN der Feature-Transformation – `arn:aws:personalize:::feature-transformation/item-attribute-affinity`
- Rezepttyp – `USER_SEGMENTATION`

In der folgenden Tabelle werden die Hyperparameter für das Item-Attribute-Affinity-Rezept beschrieben. Ein Hyperparameter ist ein Algorithmusparameter, den Sie anpassen können, um die Modellleistung zu verbessern. Algorithmus-Hyperparameter steuern die Leistung des Modells. Sie können die Hyperparameteroptimierung (HPO) nicht mit dem Item-Attribute-Affinity-Rezept verwenden.

Die Tabelle enthält auch die folgenden Informationen für jeden Hyperparameter:

- Bereich: [Untergrenze, Obergrenze]
- Werttyp: Ganzzahl, kontinuierlich (float), kategorisch (boolean, Liste, Zeichenfolge)

Name	Beschreibung
Algorithmus-Hyperparameter	
hidden_dimension	<p>Anzahl der ausgeblendeten Variablen im Modell. Ausgeblendete Variablen erstellen die Kaufhistorie und Elementstatistiken der Benutzer neu, um die Rangfolge zu generieren. Geben Sie eine größere Anzahl ausgeblendeter Dimensionen an, wenn Ihr Interaktionen-Dataset komplexere Muster enthält. Je mehr ausgeblendete Dimensionen verwendet werden, desto größer muss das Dataset sein und desto länger dauert die Verarbeitung.</p> <p>Standardwert</p> <p>Bereich: [32, 256]</p> <p>Werttyp: Ganzzahl</p>

Checkliste für die Bereitschaft

Nachdem Sie die Funktionsweise von Amazon Personalize überprüft und die Übung „Erste Schritte“ abgeschlossen haben, können Sie mit der Verwendung von Amazon Personalize mit Ihren eigenen Daten beginnen. Diese Checkliste enthält Listen von Amazon-Personalize-Funktionen, -Anforderungen und -Datenleitfäden. Es kann Ihnen bei der Planung helfen, oder Sie können es als Referenz verwenden, wenn Sie Ressourcen in Amazon Personalize erstellen.

Themen

- [Haben Sie Ihre Anwendungsfälle mit Amazon-Personalize-Ressourcen abgeglichen?](#)
- [Verfügen Sie über genügend Daten zur Elementinteraktion?](#)
- [Verfügen Sie über eine Echtzeit-Event-Streaming-Architektur?](#)
- [Sind Ihre Daten für Amazon Personalize optimiert?](#)
- [Sammeln Sie optionale Daten, die Empfehlungen verbessern können?](#)
- [Haben Sie einen Plan, Ihre Empfehlungen zu testen?](#)
- [Haben Sie zusätzliche Geschäftsziele?](#)

Haben Sie Ihre Anwendungsfälle mit Amazon-Personalize-Ressourcen abgeglichen?

Amazon-Personalize-Empfehlungen können sich mit den folgenden Anwendungsfällen befassen:

- Generieren personalisierter Empfehlungen für einen Benutzer
- Empfehlung ähnlicher oder verwandter Elemente
- Empfehlung zu Trenden oder beliebten Artikeln
- Empfehlung der nächsten besten Aktionen für einen Benutzer
- Neuordnen nach Relevanz (nur mit benutzerdefinierten Ressourcen)
- Generieren von Benutzersegmenten (nur mit benutzerdefinierten Ressourcen)

Amazon-Personalize-Features domänenbasierte Ressourcen und benutzerdefinierte Ressourcen, die für diese Anwendungsfälle konfiguriert sind. Erstellen Sie zunächst eine Domain-Datensatzgruppe oder eine benutzerdefinierte Datensatzgruppe:

- Mit einer Domain-Datensatzgruppe erstellen Sie Ressourcen, die für die Domains VIDEO_ON_DEMAND oder ECOMMERCE vorkonfiguriert und optimiert sind.

Wenn Sie über eine Streaming-Video- oder E-Commerce-Anwendung verfügen, empfehlen wir Ihnen, mit einer Domain-Datensatzgruppe zu beginnen. Sie können weiterhin benutzerdefinierte Ressourcen hinzufügen, z. B. Lösungen und Lösungsversionen, die für benutzerdefinierte Anwendungsfälle trainiert wurden. Und Sie können weiterhin benutzerdefinierte Ressourcen verwenden, um Batch-Empfehlungen zu erhalten.

- Bei einer benutzerdefinierten Datensatzgruppe wählen Sie ein Rezept aus, das Ihrem Anwendungsfall entspricht. Anschließend trainieren und stellen Sie nur konfigurierbare Lösungen und Lösungsversionen (trainierte Empfehlungsmodelle von Amazon Personalize) bereit. Wenn Sie bereit sind, können Sie die Lösungsversion in einer Kampagne für Echtzeitempfehlungen bereitstellen. Oder Sie können Batch-Empfehlungen ohne Kampagne erhalten.

Wenn Sie kein Streaming-Video oder keine E-Commerce-Anwendung haben, empfehlen wir Ihnen, eine benutzerdefinierte Datensatzgruppe zu erstellen. Andernfalls beginnen Sie mit einer Domain-Datensatzgruppe und fügen bei Bedarf benutzerdefinierte Ressourcen hinzu.

Informationen zu den Anwendungsfällen und benutzerdefinierten Rezepten, die in Amazon Personalize verfügbar sind, finden Sie unter [Domain-Anwendungsfälle und benutzerdefinierte Rezepte](#).

Verfügen Sie über genügend Daten zur Elementinteraktion?

Für alle Anwendungsfälle und Rezepte müssen Sie mindestens 1 000 Elementinteraktionen für 25 eindeutige Benutzer mit jeweils mindestens zwei Interaktionen haben. Für Qualitätsempfehlungen empfehlen wir Ihnen, mindestens 50.000 Elementinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Elementinteraktionen durchzuführen.

Wenn Sie sich nicht sicher sind, ob Sie über genügend Daten verfügen, können Sie diese mit der Amazon-Personalize-Konsole importieren und analysieren. Weitere Informationen finden Sie unter [Analysieren von Daten in Datensätzen](#).

Verfügen Sie über eine Echtzeit-Event-Streaming-Architektur?

Wenn Sie nicht über genügend Elementinteraktionsdaten verfügen, können Sie Amazon Personalize verwenden, um zusätzliche Echtzeit-Ereignisdaten zu sammeln. Bei einigen Rezepten und

Anwendungsfällen kann Amazon Personalize aus den neuesten Aktivitäten Ihres Benutzers lernen und Empfehlungen aktualisieren, während er Ihre Anwendung verwendet.

Informationen zur Aufzeichnung von Ereignissen, einschließlich der Auswirkungen von Ereignissen auf Empfehlungen, einer Liste der Services zur Ereignisverfolgung von Drittanbietern und Beispielimplementierungen, finden Sie unter [Ereignisse aufzeichnen](#).

Sind Ihre Daten für Amazon Personalize optimiert?

Wir empfehlen Ihnen, in Ihren Daten nach Folgendem zu suchen:

- Überprüfen Sie, ob Werte fehlen. Wir empfehlen, dass mindestens 70 % Ihrer Datensätze Daten für jedes Attribut enthalten. Wir empfehlen Spalten, die Nullwerte zulassen, mindestens zu 70 % vollständig zu sein.
- Korrigieren Sie alle Ungenauigkeiten oder Probleme in Ihren Daten, z. B. inkonsistente Namenskonventionen, doppelte Kategorien für ein Element, nicht übereinstimmende IDs zwischen Datensätzen oder doppelte IDs. Diese Probleme können sich negativ auf Empfehlungen auswirken oder zu unerwartetem Verhalten führen. Beispielsweise könnten Sie sowohl „N/A“ als auch „Nicht zutreffend“ in Ihren Daten haben, aber Empfehlungen herausfiltern, die nur auf „N/A“ basieren. Elemente, die mit „Nicht zutreffend“ gekennzeichnet sind, werden vom Filter nicht entfernt.
- Wenn ein Element, ein Benutzer oder eine Aktion mehrere Kategorien haben kann, z. B. einen Film mit mehreren Genres, kombinieren Sie die kategorischen Werte in einem Attribut und trennen Sie jeden Wert durch den Operator |. Die GENRES-Daten eines Films könnten beispielsweise Aktion | Aktion | Thriller sein.
- Vermeiden Sie mehr als 1000 mögliche Kategorien für eine Spalte (es sei denn, die Spalte enthält Daten nur zu Filterzwecken).

Eine vollständige Liste der Datenempfehlungen und Anweisungen zur Verwendung von Amazon Personalize zur Identifizierung von Problemen finden Sie unter [Analysieren von Daten in Datensätzen](#).

Sammeln Sie optionale Daten, die Empfehlungen verbessern können?

Die folgenden Daten können dazu beitragen, Ihre Empfehlungsrelevanz zu verbessern.

- Ereignistyp (erforderlich für alle Anwendungsfälle der Domain-Datensatzgruppe)
- Ereigniswert
- Kontextuelle Metadaten
- Element- und Benutzermetadaten
- Aktionsinteraktionsdaten (nur von PERSONALIZED_ACTIONS-Rezepten verwendet)

Weitere Informationen zu den Datentypen, die Amazon Personalize verwenden kann, finden Sie unter [Datentypen, die Amazon Personalize verwenden kann](#).

Haben Sie einen Plan, Ihre Empfehlungen zu testen?

Sie können A/B-Tests verwenden, um die Ergebnisse verschiedener Benutzergruppen zu vergleichen, die mit Empfehlungen aus verschiedenen Modellen interagieren. A/B-Tests können Ihnen helfen, verschiedene Empfehlungsstrategien zu vergleichen und festzustellen, ob Empfehlungen Ihnen helfen, Ihre Geschäftsziele zu erreichen. Weitere Informationen finden Sie unter [Wie Sie den Einfluss von Empfehlungen mit A/B-Tests messen](#).

Haben Sie zusätzliche Geschäftsziele?

In einigen Fällen haben Sie möglicherweise Ziele, zusätzlich zur Generierung relevanter Empfehlungen für Ihre Benutzer. Sie können beispielsweise den Umsatz maximieren oder bestimmte Arten von Elementen aus einer bestimmten Kategorie hochstufen. Die folgenden Amazon-Personalize-Funktionen können hilfreich sein:

- Werbeaktionen: Sie können Werbeaktionen verwenden, um sicherzustellen, dass ein bestimmter Prozentsatz der Artikel Ihren Geschäftsanforderungen entspricht. Weitere Informationen finden Sie unter [Werbung für Artikel in Empfehlungen](#).
- Optimierung für das Geschäftsziel: Bei einigen Rezepten für benutzerdefinierte Datensatzgruppen können Sie eine Lösung für ein benutzerdefiniertes Ziel optimieren, z. B. die Maximierung von Streaming-Minuten oder die Umsatzsteigerung. Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).
- Filtern von Empfehlungen. Verwenden Sie Filter, um Geschäftsregeln auf Empfehlungen anzuwenden. Sie können Filter verwenden, um bestimmte Arten von Elementen in Empfehlungen ein- oder auszuschließen. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Amazon Personalize Personalize-Arbeitsablauf

Nachdem Sie das überprüft haben [Checkliste für die Bereitschaft](#), können Sie mit dem Abschluss des Amazon Personalize beginnen:

1. [Erstellen Sie eine Datensatzgruppe](#)

Eine Datensatzgruppe ist ein Container für Amazon Personalize Personalize-Ressourcen. Die Art der Datensatzgruppe, die Sie erstellen, bestimmt die Ressourcen, die Sie in Schritt 3 des Amazon Personalize erstellen können.

- Mit einer Domain-Datensatzgruppe können Sie Empfehlungen erstellen, die für Anwendungsfälle von VIDEO_ON_DEMAND- oder E-COMMERCE-Domains konfiguriert sind. Sie verwenden die Empfehlungen, um Empfehlungen zu erhalten. Amazon Personalize verwaltet deren Konfiguration, Schulung und Updates. Wenn Sie mit einer Domain-Datensatzgruppe beginnen, können Sie trotzdem benutzerdefinierte Ressourcen hinzufügen.
- Mit einer benutzerdefinierten Datensatzgruppe können Sie nur benutzerdefinierte Ressourcen erstellen. Dazu gehören Lösungen, Lösungsversionen und Kampagnen. Bei diesen Ressourcen haben Sie mehr Kontrolle über Konfigurationen, Updates und Umschulungen.

2. [Daten vorbereiten und importieren](#)

Sie importieren Artikelinteraktionen, Aktionsinteraktionen, Artikel-, Benutzer- und Aktionsdatensätze in Datensätze (Amazon Personalize Personalize-Datencontainer). Sie können Datensätze in großen Mengen oder einzeln importieren. Wenn Sie Massendaten importieren, können Sie Amazon SageMaker Data Wrangler verwenden, um Daten aus mehr als 40 Quellen zu importieren und für Amazon Personalize vorzubereiten. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

3. [Erstellen Sie Domain-Empfehlungen oder benutzerdefinierte Ressourcen](#)

Nachdem Sie Ihre Daten importiert haben, erstellen Sie Domain-Empfehlungen (für Domain-Datensatzgruppen) oder benutzerdefinierte Ressourcen (für benutzerdefinierte Datensatzgruppen), um ein Modell anhand Ihrer Daten zu trainieren. Sie verwenden diese Ressourcen, um Empfehlungen zu generieren.

4. [Holen Sie sich Empfehlungen](#)

Verwenden Sie Ihren Empfehlungsdienst oder Ihre benutzerdefinierte Kampagne, um Empfehlungen zu erhalten. Mit einer benutzerdefinierten Datensatzgruppe können Sie auch Batch-Empfehlungen oder Benutzersegmente abrufen.

Nachdem Sie den Amazon Personalize-Workflow zum ersten Mal abgeschlossen haben, sollten Sie die Daten auf dem neuesten Stand halten und alle benutzerdefinierten Lösungen regelmäßig neu trainieren. Auf diese Weise kann Ihr Modell aus den jüngsten Aktivitäten Ihres Benutzers lernen und die Relevanz von Empfehlungen aufrechterhalten und verbessern. Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Schritt 1: Erstellen einer Dataset-Gruppe

Wenn Sie anfangen, Amazon Personalize zu verwenden, erstellen Sie eine Datensatzgruppe. Eine Datensatzgruppe ist ein Container für Amazon Personalize-Ressourcen, einschließlich Datensätze, Domain-Empfehlungen und benutzerdefinierte Ressourcen. Eine Datensatzgruppe organisiert Ihre Ressourcen in unabhängigen Sammlungen, sodass Ressourcen aus einer Datensatzgruppe keine Auswirkungen auf Ressourcen in anderen Datensatzgruppen haben.

Sie erstellen eine Datensatzgruppe für jede Ihrer Geschäftsdomänen. Beispielsweise haben Sie möglicherweise eine Anwendung, die Empfehlungen für das Streamen von Videos gibt, und eine andere, die Empfehlungen für Hörbücher bereitstellt. In Amazon Personalize würden Sie für jede Anwendung eine Datensatzgruppe erstellen. Auf diese Weise haben die Daten aus einer Anwendung keinen Einfluss auf die Empfehlungen, die Amazon Personalize für die andere Anwendung generiert.

Sie können eine Domain-Dataset-Gruppe oder eine benutzerdefinierte Dataset-Gruppe erstellen:

- Mit einer Domain-Dataset-Gruppe erstellen Sie Ressourcen, die für verschiedene Anwendungsfälle vorkonfiguriert und optimiert sind. Wenn Sie eine Datensatzgruppe erstellen, machen Sie sie zu einer Domain-Datensatzgruppe, indem Sie eine Domäne von VIDEO_ON_DEMAND oder ECOMMERCE angeben.

Wenn Sie eine Streaming-Video- oder E-Commerce-Anwendung haben, empfehlen wir Ihnen, eine Domain-Dataset-Gruppe zu erstellen. Sie können weiterhin benutzerdefinierte Ressourcen hinzufügen, z. B. Lösungen und Lösungsversionen, die für benutzerdefinierte Anwendungsfälle geschult wurden.

- Eine benutzerdefinierte Datensatzgruppe umfasst nur benutzerdefinierte Ressourcen, die Sie je nach Anwendungsfall konfigurieren. Mit benutzerdefinierten Ressourcen trainieren und implementieren Sie konfigurierbare Lösungen und Lösungsversionen (ein trainiertes Amazon Personalize-Empfehlungsmodell), die auf Ihre Geschäftsanforderungen zugeschnitten sind. Wenn Sie keine VIDEO_ON_DEMAND- oder E-COMMERCE-Anwendung haben, empfehlen wir Ihnen, eine benutzerdefinierte Datensatzgruppe zu erstellen. Andernfalls empfehlen wir, mit einer Domain-Dataset-Gruppe zu beginnen und bei Bedarf benutzerdefinierte Ressourcen hinzuzufügen.

Sie können eine Datensatzgruppe mit der Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.

Themen

- [Erstellen einer Dataset-Gruppe in \(Konsole\)](#)
- [Eine Datensatzgruppe erstellen \(AWS CLI\)](#)
- [Erstellen einer Datensatzgruppe \(AWS SDKs\)](#)

Erstellen einer Dataset-Gruppe in (Konsole)

Erstellen Sie eine Datensatzgruppe, indem Sie den Namen der Datensatzgruppe in der Amazon Personalize-Konsole angeben.

Um eine Dataset-Gruppe zu erstellen

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie Create dataset group (Dataset-Gruppe erstellen).
3. Wenn Sie Amazon Personalize zum ersten Mal verwenden, wählen Sie auf der Seite Datensatzgruppe erstellen unter Neue Datensatzgruppe die Option Erste Schritte aus.
4. Geben Sie für Dataset group details (Details zur Dataset-Gruppe) in Dataset group name (Dataset-Gruppenname) einen Namen für die Dataset-Gruppe an.
5. Wählen Sie Ihre Domain:
 - Wählen Sie E-Commerce, um eine E-COMMERCE-Domain-Datensatzgruppe zu erstellen.
 - Wählen Sie Video on Demand, um eine VIDEO_ON_DEMAND Domain-Datensatzgruppe zu erstellen.

- Wählen Sie Benutzerdefiniert, um eine benutzerdefinierte Datensatzgruppe mit nur benutzerdefinierten Ressourcen wie Lösungen, Kampagnen und Batch-Inferenzaufträgen zu erstellen.
6. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen über das Markieren von Amazon Personalize-Ressourcen mit Tags finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).
 7. Wählen Sie Create dataset group (Dataset-Gruppe erstellen). Die Übersichtsseite wird angezeigt. Sie können nun Daten importieren. Siehe [Schritt 2: Daten vorbereiten und importieren](#).

Eine Datensatzgruppe erstellen (AWS CLI)

Verwenden Sie die `create-dataset-group` Operation, um eine Datensatzgruppe zu erstellen. Um eine Domain-Datensatzgruppe zu erstellen, geben Sie für die Domäne `ECOMMERCE` oder `anVIDEO_ON_DEMAND`. Um eine benutzerdefinierte Datensatzgruppe zu erstellen, geben Sie keine Domäne an. Sie können den Parameter `Tags` verwenden, um Ressourcen in Amazon Personalize optional zu taggen. Ein Beispiel finden Sie unter [Hinzufügen von Tags \(AWS CLI\)](#).

Der folgende Code erstellt eine Domain-Dataset-Gruppe für die `VIDEO_ON_DEMAND` Domäne.

```
aws personalize create-dataset-group \  
--name dataset-group-name \  
--domain VIDEO_ON_DEMAND
```

Wenn der Befehl erfolgreich ausgeführt wurde, wird die Dataset-Gruppe Amazon Reser-Gruppe (ARN) wie folgt angezeigt.

```
{  
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/  
DatasetGroupName"  
}
```

Notieren Sie sich diesen Wert für future Verwendung. Um die von Ihnen erstellte Datensatzgruppe anzuzeigen, verwenden Sie den `describe-dataset-group` Befehl und geben Sie die zurückgegebene Datensatzgruppe ARN an.

```
aws personalize describe-dataset-group \  
--dataset-group-arn dataset group arn
```

Die Datensatzgruppe und ihre Eigenschaften werden wie folgt angezeigt.

```
{
  "datasetGroup": {
    "name": "DatasetGroupName",
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
DatasetGroupName",
    "status": "ACTIVE",
    "creationDateTime": 1542392161.262,
    "lastUpdatedDateTime": 1542396513.377
  }
}
```

Wenn die Datensatzgruppe AKTIVstatus ist, sind Sie bereit, Daten zu importieren. Weitere Informationen finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Erstellen einer Datensatzgruppe (AWSSDKs)

Verwenden Sie den folgenden Code, um eine Domäne-Datensatz-Gruppe zu erstellen. Geben Sie der Domain-Dataset-Gruppe einen Namen und geben Sie für `domain` entweder `ECOMMERCE` oder `anVIDEO_ON_DEMAND`. Um eine benutzerdefinierte Datensatzgruppe zu erstellen, ändern Sie den Code, um den Domain-Parameter zu entfernen.

Weitere Informationen zum API-Betrieb finden Sie [CreateDatasetGroup](#) unter API-Referenz. Sie können den Parameter `Tags` verwenden, um Ressourcen in Amazon Personalize optional zu taggen. Ein Beispiel finden Sie unter [Hinzufügen von Tags \(AWS SDKs\)](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
    name = 'dataset group name',
    domain = 'business domain'
)
dsg_arn = response['datasetGroupArn']
```



```

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])

```

SDK for Java 2.x

```

public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                              String datasetGroupName,
                                              String domain) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
        CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .domain(domain)
            .build();

        return
        personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
    name: 'NAME', /* required */
    domain: 'DOMAIN' /* required for a domain dsG, specify ECOMMERCE or
    VIDEO_ON_DEMAND */
}

```

```
export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateDatasetGroupCommand(domainDatasetGroupParams));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Die [DescribeDatasetGroup](#) Operation gibt den Status `datasetGroupArn` und den Status des Vorgangs zurück. Wenn die Datensatzgruppe `AKTIV` status ist, sind Sie bereit, Daten zu importieren. Weitere Informationen finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Schritt 2: Daten vorbereiten und importieren

Amazon Personalize verwendet Ihre Daten, um Empfehlungen für Ihre Benutzer und Benutzersegmente zu generieren. Amazon Personalize speichert Ihre Daten in Datensätzen, bis Sie die Datensätze löschen. Für alle Anwendungsfälle (Domain-Datensatzgruppen) und Rezepte (benutzerdefinierte Ressourcen) müssen Ihre Interaktionsdaten Folgendes enthalten:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Wenn Sie Daten importieren, können Sie wählen, ob Datensätze in großen Mengen, einzeln oder in beiden Fällen importiert werden sollen.

- Bei Massenimporten wird eine große Anzahl historischer Datensätze gleichzeitig importiert. Sie können Ihre Artikelinteraktions-, Benutzer- und Artikel-Massendaten mit Data Wrangler und mehreren SageMaker Datenquellen vorbereiten und importieren. Oder Sie können Massendaten

selbst vorbereiten und sie direkt aus einer CSV-Datei in Amazon S3 in Amazon Personalize importieren. Informationen zum Formatieren Ihrer Massendaten für Amazon Personalize finden Sie unter [Richtlinien für Datenformate](#).

- Bei Einzelimporten importieren Sie einzelne Datensätze mit der Amazon Personalize Personalize-Konsole und API-Vorgängen. Oder Sie können Interaktionsdaten von Live-Events in Echtzeit importieren.

Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Wenn Ihr Katalog wächst, aktualisieren Sie Ihre historischen Daten mit zusätzlichen Import von Massen- oder Einzeldaten. Um Empfehlungen in Echtzeit zu erhalten, sollten Sie Ihren Datensatz mit Artikelinteraktionen über das Verhalten Ihrer Nutzer auf dem neuesten Stand halten. Sie tun dies, indem Sie [Interaktionsergebnisse](#) in Echtzeit mit einem Event-Tracker und dem [PutEvents](#) Vorgang aufzeichnen. Weitere Informationen finden Sie unter [Ereignisse aufzeichnen](#)

Themen

- [Massendaten vorbereiten und importieren](#)
- [Einzelne Datensätze importieren](#)

Massendaten vorbereiten und importieren

Wenn Sie einen Datensatz erstellt haben, können Sie mit dem Import Ihrer historischen Massendaten in Amazon Personalize beginnen. Sie haben zwei Möglichkeiten, Ihre Massendatensätze zu importieren:

- Für Artikelinteraktionen, Benutzer und Artikeldatensätze können Sie Amazon SageMaker Data Wrangler verwenden, um Ihre Daten aus mehr als 40 Quellen zu importieren, Visualisierungen und Amazon Personalize-spezifische Erkenntnisse zu generieren und sie so zu transformieren, dass sie die Anforderungen von Amazon Personalize erfüllen.
- Für alle Datensatztypen können Sie Massendaten direkt in Datensätze importieren. Wenn Sie direkt importieren, formatieren Sie Ihre Daten manuell, um die Anforderungen von Amazon Personalize zu erfüllen, und laden sie auf Amazon S3 hoch. Anschließend erstellen Sie ein Schema und einen Datensatz und importieren die Daten mit einem Datensatz-Importjob direkt in den Datensatz.

Mithilfe der folgenden Richtlinien können Sie sicherstellen, dass Ihre Massendaten korrekt formatiert sind.

- Ihre Eingabedaten müssen sich in einer CSV-Datei (durch Kommas getrennte Werte) befinden.
- Die erste Zeile Ihrer CSV-Datei muss Ihre Spaltenüberschriften enthalten. Die Überschriften sollten nicht von Anführungszeichen (") umgeben sein.
- Stellen Sie sicher, dass Sie über die erforderlichen Felder für Ihren Datensatztyp verfügen und dass ihre Namen den Anforderungen von Amazon Personalize entsprechen. Beispielsweise könnten Ihre Artikeldaten eine Spalte `ITEM_IDENTIFICATION_NUMBER` mit IDs für jeden Ihrer Artikel enthalten. Um diese Spalte als `ITEM_ID`-Feld zu verwenden, benennen Sie die Spalte in `um. ITEM_ID`. Wenn Sie Data Wrangler verwenden, um Ihre Daten zu formatieren, können Sie die Transformation `Map columns for Amazon Personalize Data Wrangler` verwenden, um sicherzustellen, dass Ihre Spalten korrekt benannt sind.

Informationen zu Pflichtfeldern finden Sie unter [Schemata](#). Informationen zur Verwendung von Data Wrangler zur Vorbereitung Ihrer Daten finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

- Die Namen der Spaltenüberschriften in Ihrer CSV-Datei müssen Ihrem Schema entsprechen.
- Jeder Datensatz in Ihrer CSV-Datei muss sich in einer einzigen Zeile befinden.
- Die Datentypen in jeder Spalte müssen Ihrem Schema entsprechen. Wenn Sie Data Wrangler zum Formatieren Ihrer Daten verwenden, können Sie die Data Wrangler-Transformation „[Wert als Typ analysieren](#)“ verwenden, um die Datentypen zu konvertieren.
- `TIMESTAMP` und die `CREATION_TIMESTAMP` Daten müssen im UNIX-Zeitformat für die Epoche vorliegen. Weitere Informationen finden Sie unter [Zeitstempeldaten](#).
- Wenn Ihre Daten nicht-ASCII-kodierte Zeichen enthalten, muss Ihre CSV-Datei im UTF-8-Format codiert sein.
- Stellt sicher, dass Sie alle Textdaten wie unter beschrieben formatieren. [Unstrukturierte Textmetadaten](#)
- Achten Sie darauf, Impressionsdaten und kategoriale Daten wie unter und beschrieben zu formatieren. [Formatieren von expliziten Eindrücken](#) [Formatieren von kategorialen Daten](#)

Weitere Informationen zu den Anforderungen an die Formatierung von Massendaten für Amazon Personalize finden Sie unter [Richtlinien für Datenformate](#).

Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Wenn Sie bereits eine Empfehlung erstellt oder eine benutzerdefinierte Lösungsversion mit einer Kampagne bereitgestellt haben, hängt die Art und Weise, wie sich neue Masseneinträge auf Empfehlungen auswirken, vom verwendeten Domain-Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).


Filtern Sie Updates nach Bulk-Datensätzen

Innerhalb von 20 Minuten nach Abschluss eines Massenimports aktualisiert Amazon Personalize alle Filter, die Sie in der Datensatzgruppe erstellt haben, mit Ihren neuen Artikel- und Benutzerdaten. Mit diesem Update kann Amazon Personalize die neuesten Daten verwenden, um Empfehlungen für Ihre Benutzer zu filtern.

Themen

- [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#)
- [Daten direkt in Amazon Personalize Personalize-Datensätze importieren](#)

Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler

 **Important**

Wenn Sie Data Wrangler verwenden, entstehen SageMaker Ihnen Kosten. Eine vollständige Liste der Gebühren und Preise finden Sie auf der Registerkarte Data Wrangler in der [SageMaker Amazon-Preisübersicht](#). Um zusätzliche Gebühren zu vermeiden, fahren Sie Ihre Data Wrangler-Instance herunter, wenn Sie fertig sind. Weitere Informationen finden Sie unter Data Wrangler [herunterfahren](#).

Nachdem Sie eine Datensatzgruppe erstellt haben, können Sie Amazon SageMaker Data Wrangler (Data Wrangler) verwenden, um Daten aus mehr als 40 Quellen in einen Amazon Personalize-Datensatz zu importieren. Data Wrangler ist eine Funktion von Amazon SageMaker Studio Classic, die eine end-to-end Lösung zum Importieren, Vorbereiten, Transformieren und Analysieren von Daten bietet. Sie können Data Wrangler nicht verwenden, um Daten vorzubereiten und in einen Actions-Datensatz oder einen Action-Interaktions-Datensatz zu importieren.

Wenn Sie Data Wrangler zum Vorbereiten und Importieren von Daten verwenden, verwenden Sie einen Datenfluss. Ein Datenfluss definiert eine Reihe von Datenvorbereitungsschritten für maschinelles Lernen, beginnend mit dem Import von Daten. Jedes Mal, wenn Sie Ihrem Flow einen Schritt hinzufügen, ergreift Data Wrangler eine Aktion an Ihren Daten, z. B. transformiert sie oder generiert eine Visualisierung.

Im Folgenden sind einige der Schritte aufgeführt, die Sie zu Ihrem Schema hinzufügen können, um Daten für Amazon Personalize vorzubereiten:

- **Einblicke:** Sie können Ihrem Flow spezifische Insight-Schritte von Amazon Personalize hinzufügen. Diese Erkenntnisse können Ihnen helfen, mehr über Ihre Daten zu erfahren und zu erfahren, welche Maßnahmen Sie ergreifen können, um sie zu verbessern.
- **Visualisierungen:** Sie können Visualisierungsschritte hinzufügen, um Grafiken wie Histogramme und Streudiagramme zu erstellen. Mithilfe von Diagrammen können Sie Probleme in Ihren Daten erkennen, z. B. Ausreißer oder fehlende Werte.
- **Transformationen:** Sie können spezifische und allgemeine Transformationsschritte von Amazon Personalize verwenden, um sicherzustellen, dass Ihre Daten die Anforderungen von Amazon Personalize erfüllen. Die Amazon Personalize Personalize-Transformation hilft Ihnen, Ihre Datenspalten je nach Amazon Personalize Personalize-Datensatztyp den erforderlichen Spalten zuzuordnen.

Wenn Sie Data Wrangler verlassen müssen, bevor Sie Daten in Amazon Personalize importieren, können Sie zu der Stelle zurückkehren, an der Sie aufgehört haben, indem Sie denselben Datensatztyp wählen, wenn Sie [Data Wrangler von der Amazon Personalize Personalize-Konsole aus starten](#). Oder Sie können direkt über Studio Classic auf Data Wrangler zugreifen. SageMaker

Wir empfehlen Ihnen, Daten aus Data Wrangler wie folgt in Amazon Personalize zu importieren. Die Transformations-, Visualisierungs- und Analyseschritte sind optional, wiederholbar und können in beliebiger Reihenfolge ausgeführt werden.

1. [Berechtigungen einrichten](#) — Richten Sie Berechtigungen für Amazon Personalize und SageMaker Servicerollen ein. Und richten Sie Berechtigungen für Ihre Benutzer ein.
2. [Data Wrangler in SageMaker Studio Classic von der Amazon Personalize Personalize-Konsole aus starten](#) — [Verwenden Sie die Amazon Personalize Personalize-Konsole](#), um eine SageMaker Domain zu konfigurieren und Data Wrangler in Studio Classic zu starten. SageMaker

3. [Importieren Sie Ihre Daten in Data Wrangler — Importieren Sie Daten aus über 40 Quellen in Data Wrangler](#). Zu den Quellen gehören AWS Dienste wie Amazon Redshift, Amazon EMR oder Amazon Athena sowie Drittanbieter wie Snowflake oder. DataBricks
4. [Transformieren Sie Ihre Daten](#) — Verwenden Sie Data Wrangler, um Ihre Daten so zu transformieren, dass sie die Anforderungen von Amazon Personalize erfüllen.
5. [Visualisieren und analysieren Sie Ihre Daten](#) — Verwenden Sie Data Wrangler, um Ihre Daten zu visualisieren und sie mithilfe von Amazon Personalize-spezifischen Erkenntnissen zu analysieren.
6. [Daten verarbeiten und in Amazon Personalize importieren](#) — Verwenden Sie ein SageMaker Studio Classic Jupyter-Notizbuch, um Ihre verarbeiteten Daten in Amazon Personalize zu importieren.

Zusätzliche Informationen

Die folgenden Ressourcen bieten zusätzliche Informationen zur Verwendung von Amazon SageMaker Data Wrangler und Amazon Personalize.

- Ein Tutorial, das Sie durch die Verarbeitung und Transformation eines Beispieldatensatzes führt, finden Sie unter [Demo: Data Wrangler Titanic Dataset Walkthrough](#) im Amazon Developer Guide. SageMaker In diesem Tutorial werden die Felder und Funktionen von Data Wrangler vorgestellt.
- Informationen zum Onboarding bei SageMaker Amazon-Domains finden Sie unter [Schnelles Onboarding bei Amazon SageMaker Domain](#) im Amazon SageMaker Developer Guide.
- Informationen zu den Datenanforderungen von Amazon Personalize finden Sie unter [Richtlinien für Datenformate](#) und [Schemata](#).

Einrichten von Berechtigungen

Um Daten mit Data Wrangler vorzubereiten, müssen Sie die folgenden Berechtigungen einrichten:

- Erstellen Sie eine Servicerolle für Amazon Personalize: Falls Sie dies noch nicht getan haben, folgen Sie den Anweisungen unter [Einrichten von Amazon Personalize](#) So erstellen Sie eine IAM-Servicerolle für Amazon Personalize. Diese Rolle muss über `GetObject` `ListBucket` Berechtigungen für die Amazon S3 S3-Buckets verfügen, in denen Ihre verarbeiteten Daten gespeichert werden. Und sie muss berechtigt sein, alle AWS KMS Schlüssel zu verwenden.

Informationen darüber, wie Sie Amazon Personalize Zugriff auf Ihre Amazon S3 S3-Buckets gewähren, finden Sie unter. [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#)
Informationen darüber, wie Sie Amazon Personalize Zugriff auf Ihre AWS KMS Schlüssel

gewähren, finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

- Erstellen Sie einen Administratorbenutzer mit SageMaker Berechtigungen: Ihr Administrator muss vollen Zugriff auf eine SageMaker Domain haben SageMaker und diese erstellen können. Weitere Informationen finden Sie im Amazon SageMaker Developer Guide unter [Create a Administrative User and Group](#).
- Eine SageMaker Ausführungsrolle erstellen: Erstellen Sie eine SageMaker Ausführungsrolle mit Zugriff auf SageMaker Ressourcen und Amazon Personalize Personalize-Datenimportoperationen. Der SageMaker Ausführungsrolle muss die [AmazonSageMakerFullAccess](#) Richtlinie angehängt sein. Wenn Sie detailliertere Data Wrangler-Berechtigungen benötigen, finden Sie weitere Informationen unter [Data Wrangler Security and Permissions](#) im Amazon Developer Guide. SageMaker [Weitere Informationen zu Rollen finden Sie unter Rollen SageMaker . SageMaker](#)

Um Zugriff auf Amazon Personalize Personalize-Datenimportvorgänge zu gewähren, fügen Sie der SageMaker Ausführungsrolle die folgende IAM-Richtlinie hinzu. Diese Richtlinie gewährt die erforderlichen Berechtigungen, um Daten in Amazon Personalize zu importieren und eine Richtlinie an Ihren Amazon S3 S3-Bucket anzuhängen. Und es gewährt PassRole Berechtigungen, wenn es sich bei dem Service um Amazon Personalize handelt. Aktualisieren Sie Amazon S3 bucket-name auf den Namen des Amazon S3 S3-Buckets, den Sie als Ziel für Ihre formatierten Daten verwenden möchten, nachdem Sie sie mit Data Wrangler vorbereitet haben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:Create*",
        "personalize:List*",
        "personalize:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutBucketPolicy"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",

```



```
        "arn:aws:s3:::bucket-name/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "personalize.amazonaws.com"
        }
    }
}
]
```

Informationen zur Erstellung einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch](#). Informationen zum Anhängen einer IAM-Richtlinie an eine Rolle finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch](#).

Data Wrangler von Amazon Personalize aus starten

Um Data Wrangler von Amazon Personalize aus zu starten, verwenden Sie die Amazon Personalize Personalize-Konsole, um eine SageMaker Domain zu konfigurieren und Data Wrangler zu starten.

Um Data Wrangler von Amazon Personalize aus zu starten

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie unter Datensätze einrichten die Option Datensatz erstellen und wählen Sie den Typ des zu erstellenden Datensatzes aus. Sie können Data Wrangler nicht verwenden, um einen Aktionsdatensatz oder einen Aktionsinteraktionen-Datensatz vorzubereiten.
4. Wählen Sie Daten mithilfe von Data Wrangler importieren und dann Weiter.
5. Wählen Sie für SageMaker Domain aus, ob Sie eine bestehende Domain verwenden oder eine neue erstellen möchten. Sie benötigen eine SageMaker Domain, um in SageMaker Studio

Classic auf Data Wrangler zuzugreifen. Informationen zu Domains und Benutzerprofilen finden Sie unter [SageMaker Domain](#) im Amazon SageMaker Developer Guide.

6. Um eine bestehende Domain zu verwenden, wählen Sie eine SageMaker Domain und ein Benutzerprofil aus, um die Domain zu konfigurieren.
7. Um eine neue Domain zu erstellen:
 - Geben Sie der neuen Domain einen Namen.
 - Wählen Sie einen Benutzerprofilnamen.
 - Wählen Sie für die Ausführungsrolle die Rolle aus, in der Sie sie erstellt haben [Einrichten von Berechtigungen](#). Oder, falls Sie über die CreateRole erforderlichen Berechtigungen verfügen, erstellen Sie mit dem Assistenten zur Rollenerstellung eine neue Rolle. Der Rolle, die Sie verwenden, muss die AmazonSageMakerFullAccess Richtlinie angehängt sein.
8. Wählen Sie Weiter. Wenn Sie eine neue Domain erstellen, SageMaker beginnt die Erstellung Ihrer Domain. Dies kann bis zu zehn Minuten dauern.
9. Überprüfen Sie die Details für Ihre SageMaker Domain.
10. Wählen Sie Daten mit Data Wrangler importieren. SageMaker Studio Classic beginnt mit der Erstellung Ihrer Umgebung. Wenn Sie fertig sind, wird die Datenflusseite von Data Wrangler in SageMaker Studio Classic auf einer neuen Registerkarte geöffnet. Es kann bis zu fünf Minuten dauern, bis SageMaker Studio Classic die Erstellung Ihrer Umgebung abgeschlossen hat. Wenn der Vorgang abgeschlossen ist, können Sie mit dem Import von Daten in Data Wrangler beginnen. Weitere Informationen finden Sie unter [Daten in Data Wrangler importieren](#).

Daten in Data Wrangler importieren

Nachdem Sie eine SageMaker Domain konfiguriert und Data Wrangler auf einer neuen Registerkarte gestartet haben, sind Sie bereit, Daten aus Ihrer Quelle in Data Wrangler zu importieren. Wenn Sie Data Wrangler verwenden, um Daten für Amazon Personalize vorzubereiten, importieren Sie jeweils einen Datensatz. Wir empfehlen, mit einem Datensatz für Artikelinteraktionen zu beginnen. Sie können Data Wrangler nicht verwenden, um einen Aktionsdatensatz oder einen Datensatz mit Aktionsinteraktionen vorzubereiten.

Sie beginnen auf der Datenflusseite. Die Seite sollte in etwa wie folgt aussehen.

The screenshot shows the Amazon Personalize Data Wrangler interface. At the top right, it indicates '16 vCPU + 64 GiB' and a 'Get help' button. A notification banner states: 'You can use Data Wrangler to import data from your data source into Amazon Personalize datasets. You start by specifying your data source and importing your data into Data Wrangler. Then you can analyze it, transform it and import it into Amazon Personalize. For information about importing data, see [Importing data using Data Wrangler](#).' Below this, the 'Data Flow' section is active, showing a workflow diagram with three steps: 'Import Data', 'Prepare', and 'Process'. Below the diagram are two buttons: 'Import data' and 'Use sample dataset'.

Um mit dem Import von Daten zu beginnen, wählen Sie Daten importieren und geben Ihre Datenquelle an. Data Wrangler unterstützt mehr als 40 Quellen. Dazu gehören AWS Dienste wie Amazon Redshift, Amazon EMR oder Amazon Athena sowie Dritte wie Snowflake oder. DataBricks. Verschiedene Datenquellen haben unterschiedliche Verfahren zum Verbinden und Importieren von Daten.

Eine vollständige Liste der verfügbaren Quellen und step-by-step Anweisungen zum Importieren von Daten finden Sie unter [Import](#) im Amazon SageMaker Developer Guide.

Nachdem Sie Daten in Data Wrangler importiert haben, sind Sie bereit, sie zu transformieren. Hinweise zur Transformation von Daten finden Sie unter. [Daten transformieren](#)

Daten transformieren

Um Daten in Data Wrangler zu transformieren, fügen Sie Ihrem Datenfluss einen Transformationsschritt hinzu. Data Wrangler umfasst über 300 Transformationen, mit denen Sie Ihre Daten aufbereiten können, darunter eine Transformation mit Kartenspalten für Amazon Personalize. Und Sie können die allgemeinen Data Wrangler-Transformationen verwenden, um Probleme wie Ausreißer, Typprobleme und fehlende Werte zu beheben.

Nachdem Sie Ihre Daten transformiert haben, können Sie sie mit Data Wrangler analysieren. Oder, wenn Sie mit der Vorbereitung Ihrer Daten in Data Wrangler fertig sind, können Sie sie verarbeiten und in Amazon Personalize importieren. Informationen zur Analyse von Daten finden Sie unter [Generierung von Visualisierungen und Dateneinblicken](#). Hinweise zum Verarbeiten und Importieren von Daten finden Sie unter [Daten verarbeiten und in Amazon Personalize importieren](#).

Themen

- [Zuordnen von Spalten für Amazon Personalize](#)
- [Allgemeine Daten: Wrangler-Transformationen](#)

Zuordnen von Spalten für Amazon Personalize

Um Ihre Daten so zu transformieren, dass sie die Anforderungen von Amazon Personalize erfüllen, fügen Sie die Transformation Map-Spalten für Amazon Personalize hinzu und ordnen Ihre Spalten den erforderlichen und optionalen Feldern für Amazon Personalize zu.

So verwenden Sie die Map-Spalten für die Amazon Personalize Personalize-Transformation

1. Wählen Sie + für Ihre letzte Transformation und wählen Sie Transformation hinzufügen. Wenn Sie keine Transformation hinzugefügt haben, wählen Sie + für die Transformation der Datentypen. Data Wrangler fügt diese Transformation automatisch zu Ihrem Flow hinzu.
2. Wählen Sie Schritt hinzufügen.
3. Wählen Sie Transforms for Amazon Personalize. Die Transformation Kartenspalten für Amazon Personalize ist standardmäßig ausgewählt.
4. Verwenden Sie die Transformationsfelder, um Ihre Daten den erforderlichen Amazon Personalize-Attributen zuzuordnen.
 1. Wählen Sie den Datensatztyp, der Ihren Daten entspricht (Interaktionen, Artikel oder Benutzer).

2. Wählen Sie Ihre Domain (E-COMMERCE, VIDEO_ON_DEMAND oder benutzerdefiniert). Die von Ihnen gewählte Domain muss mit der Domain übereinstimmen, die Sie bei der Erstellung Ihrer Datensatzgruppe angegeben haben.
3. Wählen Sie die Spalten aus, die den erforderlichen und optionalen Feldern für Amazon Personalize entsprechen. Wählen Sie beispielsweise für die Spalte item_ID die Spalte in Ihren Daten aus, in der die eindeutigen Identifikationsinformationen für jeden Ihrer Artikel gespeichert sind.

Jedes Spaltenfeld wird nach Datentyp gefiltert. Nur die Spalten in Ihren Daten, die die Amazon Personalize Personalize-Datentypanforderungen erfüllen, sind verfügbar. Wenn Ihre Daten nicht dem erforderlichen Typ entsprechen, können Sie die Wrangler-Transformation „[Wert als Typ analysieren](#)“ verwenden, um sie zu konvertieren.

Allgemeine Daten: Wrangler-Transformationen

Die folgenden allgemeinen Data Wrangler-Transformationen können Ihnen bei der Vorbereitung von Daten für Amazon Personalize helfen:

- Datentypkonvertierung: Wenn Ihr Feld in der Transformation Map-Spalten für Amazon Personalize nicht als mögliche Option aufgeführt ist, müssen Sie möglicherweise seinen Datentyp konvertieren. Die Data Wrangler-Transformation „[Wert als Typ analysieren](#)“ kann Ihnen bei der Konvertierung Ihrer Daten helfen. Oder Sie können die Datentypentransformation verwenden, die Data Wrangler standardmäßig hinzufügt, wenn Sie einen Flow erstellen. Um diese Transformation zu verwenden, wählen Sie den Datentyp aus den Dropdownlisten Typ aus, wählen Vorschau und dann Aktualisieren aus.

Informationen zu den erforderlichen Datentypen für Felder finden Sie im Abschnitt für Ihre Domain und Ihren Datensatztyp unter [Schemata](#).

- Umgang mit fehlenden Werten und Ausreißern: Wenn Sie Erkenntnisse über fehlende Werte oder Ausreißer generieren, können Sie die Data Wrangler-Transformationen „[Ausreißer behandeln](#)“ und „[Fehlende Werte behandeln](#)“ verwenden, um diese Probleme zu lösen.
- Benutzerdefinierte Transformationen: Mit Data Wrangler können Sie Ihre eigenen Transformationen mit Python (benutzerdefinierte Funktion) PySpark, Pandas oder (SQL) erstellen. PySpark Sie können eine benutzerdefinierte Transformation verwenden, um Aufgaben wie das Löschen doppelter Spalten oder das Gruppieren nach Spalten auszuführen. Weitere Informationen finden Sie unter [Benutzerdefinierte Transformationen](#) im Amazon SageMaker Developer Guide.

Generierung von Visualisierungen und Dateneinblicken

Nachdem Sie Ihre Daten in Data Wrangler importiert haben, können Sie sie verwenden, um Visualisierungen und Dateneinblicke zu generieren.

- [Visualisierungen](#): Data Wrangler kann verschiedene Arten von Diagrammen erstellen, z. B. Histogramme und Streudiagramme. Sie können beispielsweise ein Histogramm erstellen, um Ausreißer in Ihren Daten zu identifizieren.
- [Dateneinblicke](#): Sie können einen Datenqualitäts- und Insights-Bericht für Amazon Personalize verwenden, um anhand von Dateneinblicken sowie Spalten- und Zeilenstatistiken mehr über Ihre Daten zu erfahren. Dieser Bericht kann Sie darüber informieren, ob Sie irgendwelche Typprobleme mit Ihren Daten haben. Außerdem erfahren Sie, welche Maßnahmen Sie ergreifen können, um Ihre Daten zu verbessern. Diese Maßnahmen können Ihnen helfen, die Ressourcenanforderungen von Amazon Personalize zu erfüllen, z. B. die Anforderungen an die Modellschulung, oder sie können zu verbesserten Empfehlungen führen.

Nachdem Sie mithilfe von Visualisierungen und Erkenntnissen mehr über Ihre Daten erfahren haben, können Sie diese Informationen verwenden, um zusätzliche Transformationen zur Verbesserung Ihrer Daten anzuwenden. Oder, wenn Sie mit der Vorbereitung Ihrer Daten fertig sind, können Sie sie verarbeiten und in Amazon Personalize importieren. Informationen zur Transformation Ihrer Daten finden Sie unter [Daten transformieren](#). Hinweise zum Verarbeiten und Importieren von Daten finden Sie unter [Daten verarbeiten und in Amazon Personalize importieren](#).

Generieren von Visualisierungen

Sie können Data Wrangler verwenden, um verschiedene Arten von Diagrammen zu erstellen, z. B. Histogramme und Streudiagramme. Sie können beispielsweise ein Histogramm erstellen, um Ausreißer in Ihren Daten zu identifizieren. Um eine Datenvisualisierung zu generieren, fügen Sie Ihrem Schema einen Analyseschritt hinzu und wählen unter Analysetyp die Visualisierung aus, die Sie erstellen möchten.

Weitere Informationen zum Erstellen von Visualisierungen in Data Wrangler finden Sie unter [Analysieren und Visualisieren](#) im Amazon Developer Guide. SageMaker

Generierung von Dateneinblicken

Sie können Data Wrangler verwenden, um einen Datenqualitäts- und Insights-Bericht für Amazon Personalize zu erstellen, der für Ihren Datensatztyp spezifisch ist. Bevor Sie den Bericht erstellen, empfehlen wir Ihnen, Ihre Daten so zu transformieren, dass sie die Anforderungen von Amazon

Personalize erfüllen. Dies wird zu relevanteren Erkenntnissen führen. Weitere Informationen finden Sie unter [Daten transformieren](#).

Themen

- [Inhalt melden](#)
- [Der Bericht wird generiert](#)

Inhalt melden

Der Datenqualitäts- und Insights-Bericht für Amazon Personalize umfasst die folgenden Abschnitte:

- Zusammenfassung: Die Berichtszusammenfassung umfasst Datensatzstatistiken und Warnungen mit hoher Priorität:
 - Datensatzstatistiken: Dazu gehören spezifische Statistiken von Amazon Personalize, wie die Anzahl der eindeutigen Benutzer in Ihren Interaktionsdaten, und allgemeine Statistiken, wie die Anzahl fehlender Werte oder Ausreißer.
 - Warnungen mit hoher Priorität: Dies sind spezifische Erkenntnisse von Amazon Personalize, die sich am stärksten auf Schulungen oder Empfehlungen auswirken. Jede Warnung enthält eine empfohlene Maßnahme, die Sie ergreifen können, um das Problem zu lösen.
- Doppelte Zeilen und Unvollständige Zeilen: Diese Abschnitte enthalten Informationen darüber, in welchen Zeilen Werte fehlen und welche Zeilen in Ihren Daten doppelt vorkommen.
- Zusammenfassung der Funktionen: Dieser Abschnitt enthält den Datentyp für jede Spalte, ungültige oder fehlende Dateninformationen und die Anzahl der Warnungen.
- Funktionsdetails: Dieser Abschnitt enthält Unterabschnitte mit detaillierten Informationen für jede Ihrer Datenspalten. Jeder Unterabschnitt enthält Statistiken für die Spalte, z. B. die Anzahl der kategorialen Werte und Informationen zu fehlenden Werten. Und jeder Unterabschnitt enthält spezifische Einblicke und Handlungsempfehlungen von Amazon Personalize für Datenspalten. Ein Einblick könnte beispielsweise darauf hindeuten, dass eine Spalte mehr als 30 mögliche Kategorien hat.

Probleme mit dem Datentyp

Der Bericht identifiziert Spalten, die nicht den richtigen Datentyp haben, und gibt den erforderlichen Typ an. Um Einblicke in diese Funktionen zu erhalten, müssen Sie den Datentyp der Spalte konvertieren und den Bericht erneut generieren. Um den Typ zu konvertieren, können Sie die Data Wrangler-Transformation „[Wert als Typ analysieren](#)“ verwenden.

Einblicke in Amazon Personalize

Die Erkenntnisse von Amazon Personalize beinhalten ein Ergebnis und eine vorgeschlagene Maßnahme. Die Aktion ist optional. Der Bericht könnte beispielsweise Erkenntnisse und Maßnahmen in Bezug auf die Anzahl der Kategorien für eine Spalte mit kategorialen Daten enthalten. Wenn Sie nicht glauben, dass es sich bei der Spalte um eine kategorische Spalte handelt, können Sie diese Erkenntnis ignorieren und keine Maßnahmen ergreifen.

Abgesehen von geringfügigen Formulierungsunterschieden sind die spezifischen Erkenntnisse von Amazon Personalize dieselben wie die Erkenntnisse aus einzelnen Datensätzen, die Sie möglicherweise generieren, wenn Sie Ihre Daten mit Amazon Personalize analysieren. Der Insights-Bericht in Data Wrangler enthält beispielsweise Erkenntnisse wie „Der Datensatz mit Artikelinteraktionen hat nur X eindeutige Benutzer mit zwei oder mehr Interaktionen“. Er enthält jedoch keine Erkenntnisse wie „X% der Elemente im Artikeldatensatz haben keine Interaktionen im Datensatz Artikelinteraktionen“.

Eine Liste möglicher Amazon Personalize Personalize-spezifischer Erkenntnisse finden Sie in den Erkenntnissen, die nicht auf mehrere Datensätze verweisen. [Dateneinblicke](#)

Berichtsbeispiele

Das Erscheinungsbild des Amazon Personalize entspricht dem allgemeinen Insights-Bericht in Data Wrangler. Beispiele für den allgemeinen Insights-Bericht finden [Sie unter Get Insights On Data and Data Quality](#) im Amazon SageMaker Developer Guide. Das folgende Beispiel zeigt, wie der Zusammenfassungsbereich eines Berichts für einen Datensatz mit Artikelinteraktionen aussieht. Er enthält Datensatzstatistiken und einige mögliche Warnungen zu Datensätzen mit hoher Priorität zu Artikelinteraktionen.


SUMMARY

Dataset statistics


Key	Value	Feature type	Count
Number of features	6	numeric	2
Number of rows	31	categorical	0
Missing	0%	text	4
Valid	100%	datetime	0
Duplicate rows	6.45%	binary	0
Users with sufficient int...	0	unknown	0
Number of unique users	30		
Number of unique items	30		
Sparse rows	0%		
Distinct rows	30		

High Priority Warnings


4 high severity warnings were detected. See the list below.

 Duplicate rows High

We found that 6.45% of the data are duplicate. Some data sources could include valid duplicates and in other cases these duplicates could point to problems in data collection. Duplicate samples resulting from faulty data collection, could derail machine learning processes that rely on splitting to independent training and validation folds. For example quick model scores, prediction power estimation and automatic hyper parameter tuning. Duplicate samples could be removed from the dataset using the **Drop duplicates** transform under **Manage rows**.

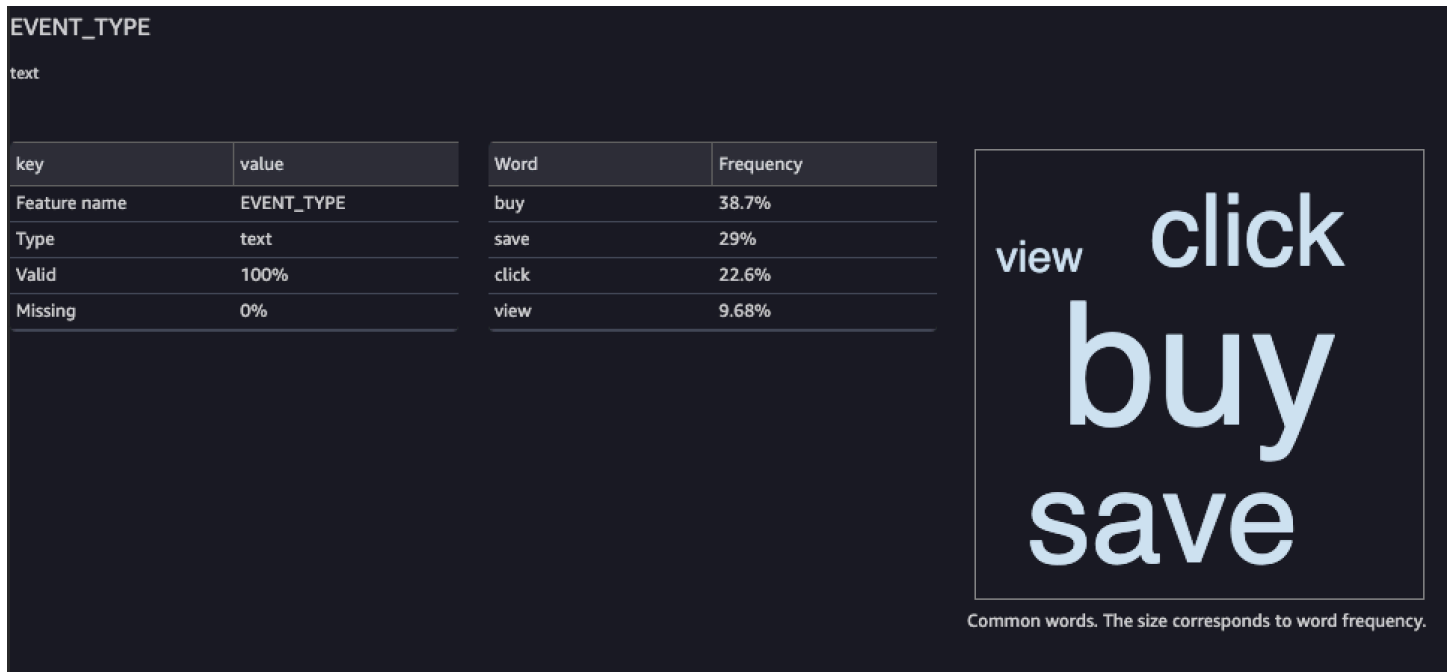
 Insufficient interactions High

The Interactions dataset has only 30 interactions. Model training requires a minimum of 1,000 interactions. We recommend at least 50,000. Import 49970 additional unique interactions records before training a model.

 Insufficient Users High

The Interactions dataset has only 0 unique users with two or more interactions. Model training requires at least 25 such users. We recommend at least 1,000. Import at least 2 interactions records each for 1000 additional users.

Das folgende Beispiel zeigt, wie der Abschnitt mit den Feature-Details für eine EVENT_TYPE-Spalte eines Datensatzes mit Artikelinteraktionen in einem Bericht erscheinen könnte.



Der Bericht wird generiert

Um den Datenqualitäts- und Insights-Bericht für Amazon Personalize zu erstellen, wählen Sie Dateneinblicke für Ihre Transformation abrufen und eine Analyse erstellen.

Um einen Datenqualitäts- und Insights-Bericht für Amazon Personalize zu erstellen

1. Wählen Sie die Option + für die Transformation, die Sie analysieren. Wenn Sie keine Transformation hinzugefügt haben, wählen Sie + für die Transformation der Datentypen. Data Wrangler fügt diese Transformation automatisch zu Ihrem Flow hinzu.
2. Wählen Sie Get Data Insights aus. Das Fenster Analyse erstellen wird angezeigt.
3. Wählen Sie als Analysetyp Data Quality and Insights Report for Amazon Personalize aus.
4. Wählen Sie unter Datensatztyp den Typ des Amazon Personalize Personalize-Datensatzes aus, den Sie analysieren.
5. Wählen Sie optional Auf vollständigen Daten ausführen aus. Standardmäßig generiert Data Wrangler nur Erkenntnisse aus einer Stichprobe Ihrer Daten.
6. Wählen Sie Create (Erstellen) aus. Wenn die Analyse abgeschlossen ist, wird der Bericht angezeigt.

Daten verarbeiten und in Amazon Personalize importieren

Wenn Sie mit der Analyse und Transformation Ihrer Daten fertig sind, können Sie sie verarbeiten und in Amazon Personalize importieren.

- [Daten verarbeiten](#) — Die Verarbeitung der Daten wendet Ihre Transformation auf Ihren gesamten Datensatz an und gibt ihn an ein von Ihnen festgelegtes Ziel aus. In diesem Fall geben Sie einen Amazon S3 S3-Bucket an.
- [Daten in Amazon Personalize importieren](#) — Um verarbeitete Daten in Amazon Personalize zu importieren, führen Sie ein in Studio Classic bereitgestelltes Jupyter Notebook aus. SageMaker Dieses Notizbuch erstellt Ihre Amazon Personalize Personalize-Datensätze und importiert Ihre Daten in sie.

Verarbeiten von Daten

Bevor Sie Daten in Amazon Personalize importieren, müssen Sie Ihre Transformation auf Ihren gesamten Datensatz anwenden und ihn in einem Amazon S3 S3-Bucket ausgeben. Dazu erstellen Sie einen Zielknoten, dessen Ziel auf einen Amazon S3 S3-Bucket festgelegt ist, und starten dann einen Verarbeitungsjob für die Transformation.

step-by-step Anweisungen zur Angabe eines Ziels und zum Starten eines Prozessauftrags finden Sie unter [Starten von Verarbeitungsaufträgen mit wenigen Klicks mithilfe von Amazon SageMaker Data Wrangler](#). Wenn Sie ein Ziel hinzufügen, wählen Sie Amazon S3. Sie verwenden diesen Speicherort, wenn Sie die verarbeiteten Daten in Amazon Personalize importieren.

Wenn Sie mit der Verarbeitung Ihrer Daten fertig sind, können Sie sie aus dem Amazon S3 S3-Bucket in Amazon Personalize importieren.

Daten in Amazon Personalize importieren

Nachdem Sie Ihre Daten verarbeitet haben, können Sie sie in Amazon Personalize importieren. Um verarbeitete Daten in Amazon Personalize zu importieren, führen Sie ein in Studio Classic bereitgestelltes Jupyter Notebook aus. SageMaker Dieses Notizbuch erstellt Ihre Amazon Personalize Personalize-Datensätze und importiert Ihre Daten in sie.

Um verarbeitete Daten in Amazon Personalize zu importieren

1. Wählen Sie für die Transformation, die Sie exportieren möchten, Exportieren nach und dann Amazon Personalize (über Jupyter Notebook).

2. Ändern Sie das Notizbuch, um den Amazon S3 S3-Bucket anzugeben, den Sie als Datenziel für den Verarbeitungsjob verwendet haben. Geben Sie optional die Domain für Ihre Datensatzgruppe an. Standardmäßig erstellt das Notizbuch eine benutzerdefinierte Datensatzgruppe.
3. Überprüfen Sie die Notizbuchzellen, aus denen das Schema erstellt wurde. Stellen Sie sicher, dass die Schemafelder die erwarteten Typen und Attribute haben, bevor Sie die Zelle ausführen.
 - Stellen Sie sicher, dass Felder, die Nulldaten unterstützen, in der Typenliste `null` aufgeführt sind. Das folgende Beispiel zeigt, wie Daten `null` für ein Feld hinzugefügt werden.

```
{
  "name": "GENDER",
  "type": [
    "null",
    "string"
  ],
  "categorical": true
}
```

- Stellen Sie sicher, dass das kategoriale Attribut für kategoriale Felder auf `true` gesetzt ist. Das folgende Beispiel zeigt, wie ein Feld als kategorisch markiert wird.

```
{
  "name": "SUBSCRIPTION_MODEL",
  "type": "string",
  "categorical": true
}
```

- Stellen Sie sicher, dass bei Textfeldern das Textattribut auf `true` gesetzt ist. Das folgende Beispiel zeigt, wie ein Feld als Textfeld markiert wird.

```
{
  "name": "DESCRIPTION",
  "type": [
    "null",
    "string"
  ],
  "textual": true
}
```

4. Führen Sie das Notizbuch aus, um ein Schema und einen Datensatz zu erstellen und Ihre Daten in den Amazon Personalize-Datensatz zu importieren. Sie führen das Notizbuch genauso aus wie ein Notizbuch außerhalb von SageMaker Studio Classic. [Informationen zum Ausführen von Jupyter-Notebooks finden Sie unter Running Code.](#) Informationen zu Notebooks in SageMaker Studio Classic finden Sie unter [Use Amazon SageMaker Notebooks](#) im Amazon SageMaker Developer Guide.

Wenn Sie das Notizbuch fertiggestellt haben und Interaktionsdaten importiert haben, können Sie Empfehlungen oder benutzerdefinierte Ressourcen erstellen. Sie können den Vorgang auch mit einem Artikeldatensatz oder einem Benutzerdatensatz wiederholen. Informationen zum Erstellen von Empfehlungen oder benutzerdefinierten Ressourcen finden Sie unter [Schritt 3: Erstellen von Empfehlungen oder benutzerdefinierten Ressourcen.](#)

Daten direkt in Amazon Personalize Personalize-Datensätze importieren

Nachdem Sie einen Datensatz erstellt haben, können Sie Massendatensätze aus einer großen CSV-Datei in einen Amazon Personalize importieren.

Um Daten direkt in Amazon Personalize importieren, gehen Sie wie folgt vor:

1. Erstellen Sie eine JSON-Schema-Datei auf der Grundlage Ihrer Daten. Schemaanforderungen und Beispiele finden Sie unter [Schemata](#).
2. Vergewissern Sie sich, dass Ihre Daten korrekt formatiert sind. Die Spaltennamen müssen Ihrem Schema entsprechen. Ihre Daten müssen sich in einer CSV-Datei befinden. Richtlinien für das Datenformat finden Sie unter [Richtlinien für Datenformate](#).
3. Laden Sie Ihre CSV-Dateien in einen Amazon Simple Storage Service (Amazon S3) -Bucket hoch und gewähren Sie Amazon Personalize Zugriff auf Ihre Amazon S3-Ressourcen.
4. Erstellen Sie ein Amazon Personalize Personalize-Schema mit der JSON-Datei aus Schritt eins. Und erstellen Sie einen Amazon Personalize Personalize-Datensatz.
5. Erstellen Sie einen Job zum Importieren von Datensätzen, der den Datensatz mit Daten aus Ihrem Amazon S3 S3-Bucket füllt. Um einen Job zum Importieren von Datensätzen für Interaktionsdatensätze zu erstellen, muss Ihre CSV-Datei mindestens 1000 Interaktionsdatensätze enthalten.

Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Themen

- [Hochladen des Hochladen des Uploads des Uploads des Amazon S3 Pakts](#)
- [Einen Datensatz und ein Schema erstellen](#)
- [Import von Massendatensätzen mit einem Datensatz-Importjob](#)

Hochladen des Hochladen des Uploads des Uploads des Amazon S3 Pakts

Nachdem Sie Ihre historischen Eingabedaten formatiert haben (siehe [Richtlinien für Datenformate](#)), müssen Sie die CSV-Datei in einen Amazon S3 S3-Bucket hochladen und Amazon Personalize die Erlaubnis erteilen, auf Ihre Amazon S3 S3-Ressourcen zuzugreifen:

1. Falls Sie dies noch nicht getan haben, folgen Sie den Schritten unter Berechtigungen einrichten, damit Amazon Personalize in [Einrichten von Berechtigungen](#) Ihrem Namen auf Ihre Amazon Personalize Personalize-Ressourcen zugreifen kann.
2. Hochladen des CSV-Dateien Ihre CSV-Dateien in einen Amazon-S3 (Amazon S3) -Bucket hoch. Dies ist der Ort, von dem Amazon Personalize Ihre Daten importiert. Weitere Informationen finden Sie unter Hochladen des [Hochladen des Uploads von Dateien und Ordnern mit Drag-and-Drop](#) im Amazon-Simple-Storage-Service unter Hochladen des Amazon-Supports des Amazon-Supports des Amazon-Supports.
3. Geben Amazon Personalize Zugriff auf Ihre Amazon-S3-Ressourcen, indem Sie Zugriffsrichtlinien an Ihren Amazon-S3-Bucket und die Amazon-S3-Bucket mit Amazon-S3-Bucket und Amazon-S3-Bucket Zugriffsrichtlinien anhängen. Siehe [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis erteilen, Ihren Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Nachdem Sie Ihre Daten in einen Amazon-S3-Bucket hochzuladen und Amazon Personalize Zugriff auf Amazon-S3-Bucket Zugriff auf Amazon-S3-Bucket Zugriff auf Amazon-S3-Bucket gewähren. Siehe [Massendaten vorbereiten und importieren](#).

Einen Datensatz und ein Schema erstellen

Nachdem Sie den Vorgang abgeschlossen haben [Schritt 1: Erstellen einer Dataset-Gruppe](#), können Sie einen Datensatz erstellen. Datensätze sind Amazon Personalize Personalize-Container für Daten. Wenn Sie einen Datensatz erstellen, erstellen Sie auch ein Schema für den Datensatz. Ein Schema informiert Amazon Personalize über die Struktur Ihrer Daten und ermöglicht Amazon Personalize, die Daten zu analysieren.

Sie erstellen Datensätze mit der Amazon Personalize Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs. Informationen zu den verschiedenen Arten von Datensätzen sowie zu den Anforderungen an Datensätze und Schemas finden Sie unter [Datensätze und Schemas](#)

Themen

- [Einen Datensatz und ein Schema erstellen \(Konsole\)](#)
- [Einen Datensatz und ein Schema erstellen \(AWS CLI\)](#)
- [Einen Datensatz und ein Schema \(AWSSDKs\) erstellen](#)

Einen Datensatz und ein Schema erstellen (Konsole)

Wenn dies Ihr erster Datensatz in Ihrer Datensatzgruppe ist, ist Ihr erster Datensatztyp ein Datensatz mit Artikelinteraktionen. Um Ihren Datensatz mit Artikelinteraktionen in der Konsole zu erstellen, geben Sie den Datensatznamen und anschließend ein JSON-Schema im [Avro-Format](#) an. Wenn es nicht Ihr erster Datensatz in dieser Datensatzgruppe ist, wählen Sie den Datensatztyp und geben Sie dann einen Namen und ein Schema an.

Informationen zu Amazon Personalize Personalize-Datensätzen und Schemaanforderungen finden Sie unter [Datensätze und Schemas](#)

Note

Wenn Sie den Vorgang gerade abgeschlossen haben [Schritt 1: Erstellen einer Dataset-Gruppe](#) und bereits dabei sind, Ihren Datensatz zu erstellen, fahren Sie mit Schritt 4 dieses Verfahrens fort.

Um einen Datensatz und ein Schema zu erstellen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen die Datensatzgruppe aus, in [Schritt 1: Erstellen einer Dataset-Gruppe](#) der Sie erstellt haben.
3. Wählen Sie unter Datensätze einrichten die Option Datensatz erstellen und wählen Sie den Typ des zu erstellenden Datensatzes aus.
4. Wählen Sie Daten direkt in Amazon Personalize Personalize-Datensätze importieren und dann Weiter.
5. Geben Sie unter Datensatzdetails für Datensatzname einen Namen für Ihren Datensatz an.
6. Wählen Sie unter Schemadetails für Schemaauswahl entweder ein vorhandenes Schema aus, oder wählen Sie Neues Schema erstellen aus.
7. Wenn Sie ein neues Schema erstellen, fügen Sie unter Schemadefinition das Schema-JSON ein, das Ihren Daten entspricht. Verwenden Sie die Beispiele unter [Schemata](#) als Leitfaden. Nachdem Sie ein Schema erstellt haben, können Sie keine Änderungen am Schema vornehmen.
8. Geben Sie für New schema name (Name des neuen Schemas) einen Namen für das neue Schema an.
9. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#)
10. Wählen Sie Weiter und folgen Sie den Anweisungen unter, um Ihre [Massendaten vorbereiten und importieren](#) Daten zu importieren.

Einen Datensatz und ein Schema erstellen (AWS CLI)

Um einen Datensatz und ein Schema mit dem zu erstellenAWS CLI, definieren Sie zunächst ein Schema im [Avro-Format](#) und fügen es mithilfe der [CreateSchema](#) Operation zu Amazon Personalize hinzu. Erstellen Sie dann mithilfe der [CreateDataset](#) Operation einen Datensatz. Informationen zu Amazon Personalize Personalize-Datensätzen und Schemaanforderungen finden Sie unter [Datensätze und Schemas](#)

So erstellen Sie ein Schema und einen Datensatz

1. Erstellen Sie eine Schemadatei im Avro-Format und speichern Sie sie als JSON-Datei. Diese Datei sollte auf dem Datasettyp basieren, den Sie erstellen, z. B. Interaktionen.

Das Schema muss mit den Spalten in Ihren Daten übereinstimmen und das Schema name muss mit einem der von Amazon Personalize erkannten Datensatztypen übereinstimmen. Das Folgende ist ein Beispiel für ein minimales Datensatzschema für Artikelinteraktionen. Weitere Beispiele finden Sie unter [Schemata](#).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

2. Erstellen Sie ein Schema in Amazon Personalize, indem Sie den folgenden Befehl ausführen. Nachdem Sie ein Schema erstellt haben, können Sie keine Änderungen am Schema vornehmen. `schemaName` ersetzen Sie es durch den Namen des Schemas und `file://SchemaName.json` ersetzen Sie es durch den Speicherort der JSON-Datei, die Sie im vorherigen Schritt erstellt haben. Das Beispiel zeigt die Datei als Teil des aktuellen Ordners.

Wenn Sie ein Schema für einen Datensatz in einer Domain-Datensatzgruppe erstellen, fügen Sie den `domain` Parameter hinzu und setzen Sie ihn auf `ECOMMERCE` oder `VIDEO_ON_DEMAND`. Weitere Informationen zur API finden Sie unter [CreateSchema](#).

```
aws personalize create-schema \
  --name SchemaName \
  --schema file://SchemaName.json
```

Das Schema Amazon Resource Name (ARN) wird angezeigt, wie im folgenden Beispiel gezeigt:

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/SchemaName"
}
```

- Erstellen Sie ein leeres Dataset, indem Sie den folgenden Befehl ausführen. Geben Sie die Datensatzgruppe Amazon Resource Name (ARN) von [Eine Datensatzgruppe erstellen \(AWS CLI\)](#) und den Schema-ARN aus dem vorherigen Schritt an. Die Werte für den Datensatztyp können InteractionsUsers,Items,Actions, oder seinAction_Interactions. Weitere Informationen zur API finden Sie unter [CreateDataset](#).

```
aws personalize create-dataset \
  --name Dataset Name \
  --dataset-group-arn Dataset Group ARN \
  --dataset-type Dataset Type \
  --schema-arn Schema Arn
```

Der Dataset-ARN wird angezeigt, wie im folgenden Beispiel gezeigt.

```
{
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/DatasetName/
INTERACTIONS"
}
```

- Notieren Sie den ARN des Datensatzes für die spätere Verwendung. Nachdem Sie einen Datensatz erstellt haben, können Sie Ihre Trainingsdaten importieren. Siehe [Massendaten vorbereiten und importieren](#).

Einen Datensatz und ein Schema (AWSSDKs) erstellen

Um einen Datensatz und ein Schema mithilfe der AWS SDKs zu erstellen, definieren Sie zunächst ein Schema im [Avro-Format](#) und fügen es mithilfe der Operation zu Amazon Personalize hinzu. [CreateSchema](#) Nachdem Sie ein Schema erstellt haben, können Sie keine Änderungen am Schema vornehmen. Erstellen Sie dann mithilfe der [CreateDataset](#) Operation einen Datensatz. Informationen zu Amazon Personalize Personalize-Datensätzen und Schemaanforderungen finden Sie unter [Datensätze und Schemas](#)

So erstellen Sie ein Schema und einen Datensatz

1. Erstellen Sie eine Schemadatei im Avro-Format und speichern Sie sie als JSON-Datei in Ihrem Arbeitsverzeichnis.

Das Schema muss mit den Spalten in Ihren Daten übereinstimmen und das Schema name muss einem von Amazon Personalize erkannten Datasetyp entsprechen. Das Folgende ist ein Beispiel für ein minimales Datensatzschema für Artikelinteraktionen. Weitere Beispiele finden Sie unter [Schemata](#).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

2. Erstellen Sie ein Schema mit dem folgenden Code. Geben Sie den Namen für Ihr Schema und den Dateipfad für Ihre Schema-JSON-Datei an.

Wenn Sie ein Schema für einen Datensatz in einer Domain-Dataset-Gruppe erstellen, fügen Sie den `domain` Parameter hinzu und setzen Sie ihn auf `ECOMMERCE` oder `VIDEO_ON_DEMAND`. Weitere Informationen zur API finden Sie unter [CreateSchema](#).

SDK for Python (Boto3)

```
import boto3
```

```
personalize = boto3.client('personalize')

with open('schemaFile.json') as f:
    createSchemaResponse = personalize.create_schema(
        name = 'schema name',
        schema = f.read()
    )

schema_arn = createSchemaResponse['schemaArn']

print('Schema ARN:' + schema_arn )
```

SDK for Java 2.x

```
public static String createSchema(PersonalizeClient personalizeClient, String
schemaName, String filePath) {

    String schema = null;

    try {
        schema = new String(Files.readAllBytes(Paths.get(filePath)));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    try {
        CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
            .name(schemaName)
            .schema(schema)
            .build();

        String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();
        System.out.println("Schema arn: " + schemaArn);

        return schemaArn;

    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from 'fs';

let schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = 'TEST' // For unit tests.
}

// Set the schema parameters.
export const createSchemaParam = {
  name: 'NAME', /* required */
  schema: mySchema /* required */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateSchemaCommand(createSchemaParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

Amazon Personalize gibt den ARN des neuen Schemas zurück. Notieren Sie es, weil Sie es im nächsten Schritt benötigen werden.

- Erstellen Sie mithilfe der [CreateDataset](#) Operation einen Datensatz. Der folgende Code zeigt, wie ein Datensatz erstellt wird. Geben Sie den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe, den Schema-ARN aus dem vorherigen Schritt und den Datensatztyp an. Die Werte für den Datensatztyp können InteractionsUsers,Items,Actions, oder seinAction_Interactions. Hinweise zu den verschiedenen Typen von Datensätzen finden Sie unter [Datensätze und Schemas](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset(
    name = 'dataset_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'dataset_type'
)

print ('Dataset Arn: ' + response['datasetArn'])
```

SDK for Java 2.x

```
public static String createDataset(PersonalizeClient personalizeClient,
                                   String datasetName,
                                   String datasetGroupArn,
                                   String datasetType,
                                   String schemaArn) {

    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
            .name(datasetName)
            .datasetGroupArn(datasetGroupArn)
            .datasetType(datasetType)
            .schemaArn(schemaArn).build();

        String datasetArn =
personalizeClient.createDataset(request).datasetArn();
        System.out.println("Dataset " + datasetName + " created. Dataset ARN: "
+ datasetArn);

        return datasetArn;
    }
}
```

```
    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
    datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
    datasetType: 'DATASET_TYPE', /* required */
    name: 'NAME', /* required */
    schemaArn: 'SCHEMA_ARN' /* required */
}

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
        CreateDatasetCommand(createDatasetParam));
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

Nachdem Sie einen Datensatz erstellt haben, können Sie Ihre Trainingsdaten importieren. Siehe [Massendaten vorbereiten und importieren](#).

Import von Massendatensätzen mit einem Datensatz-Importjob

Nachdem Sie Ihre Eingabedaten formatiert (siehe [Richtlinien für Datenformate](#)), sie in einen Amazon Simple Storage Service (Amazon S3) -Bucket (siehe [Hochladen des Hochladen des Uploads des Uploads des Amazon S3 Pakts](#)) hochgeladen und abgeschlossen haben [Einen Datensatz und ein Schema erstellen](#), importieren Sie die Massendatensätze in den Datensatz, indem Sie einen Datensatz-Importjob erstellen.

Ein Datensatz-Importjob ist ein Massenimport-Tool, das Ihren Datensatz mit Daten aus Ihrem Amazon S3 S3-Bucket füllt. Sie können einen Job zum Importieren von Datensätzen mithilfe der Amazon Personalize Personalize-Konsole AWS Command Line Interface (AWS CLI) oder der AWS SDKs erstellen.

Wenn Sie zuvor einen Datensatz-Importauftrag für einen Datensatz erstellt haben, können Sie einen neuen Datensatz-Importauftrag verwenden, um die vorhandenen Massendaten zu ergänzen oder zu ersetzen. Weitere Informationen finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Themen

- [Massendatensätze importieren \(Konsole\)](#)
- [Massendatensätze importieren \(AWS CLI\)](#)
- [Import von Massendatensätzen \(AWSSDKs\)](#)

Massendatensätze importieren (Konsole)

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Hinweise zum Aktualisieren vorhandener Daten finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Um mit der Amazon Personalize-Konsole Massendatensätze in einen Datensatz zu importieren, erstellen Sie einen Datensatz-Import-Job mit einem Namen, der IAM-Servicerolle und dem Speicherort Ihrer Daten.

Wenn Sie Ihren Datensatz gerade in erstellt haben [Einen Datensatz und ein Schema erstellen](#), fahren Sie mit Schritt 5 fort. Wenn Sie bereits einen ersten Importauftrag abgeschlossen haben

und Ihre Daten aktualisieren möchten, finden Sie weitere Informationen unter [Aktualisierung von Massendatensätzen \(Konsole\)](#).

So importieren Sie Massendatensätze (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus. Die Übersicht der Datensatzgruppe wird angezeigt.
3. Wählen Sie im Navigationsbereich Datensätze und dann den Datensatz aus, in den Sie Massendaten importieren möchten.
4. Wählen Sie unter Datensatz-Importaufträge die Option Datensatz-Importjob erstellen aus.
5. Wählen Sie unter Auftragsdetails für Datensatz-Import für Datenimportquelle die Option Daten aus S3 importieren aus.
6. Geben Sie unter Name des Datensatz-Import-Jobs einen Namen für Ihren Importjob an.
7. Geben Sie unter Eingabequelle für S3-Standort an, wo Ihre Datendatei in Amazon S3 gespeichert ist. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

Wenn sich Ihre CSV-Dateien in einem Ordner in Ihrem Amazon S3 S3-Bucket befinden und Sie mit einem Datensatz-Importjob mehrere CSV-Dateien in einen Datensatz hochladen möchten, können Sie den Pfad zu dem Ordner angeben. Amazon Personalize verwendet nur die Dateien in der ersten Ebene Ihres Ordners, es verwendet keine Daten in Unterordnern. Verwenden Sie die folgende Syntax mit einem / nach dem Ordnernamen:

s3://<name of your S3 bucket>/<folder path>/

8. Wählen Sie unter IAM-Rolle aus, ob Sie entweder eine neue Rolle erstellen oder eine vorhandene verwenden möchten. Wenn Sie die Voraussetzungen erfüllt haben, wählen Sie Bestehende Servicerolle verwenden aus und geben Sie die Rolle an, in der Sie sie erstellt haben. [Erstellen einer IAM-Rolle für Amazon Personalize](#)
9. Wenn Sie eine Metrik-Attribution erstellt haben und Metriken zu diesem Job in Amazon S3 veröffentlichen möchten, wählen Sie unter Event-Metriken in S3 veröffentlichen die Option Metriken für diesen Importjob veröffentlichen aus.

Wenn Sie noch keine erstellt haben und Metriken für diesen Job veröffentlichen möchten, wählen Sie Metrik-Attribution erstellen aus, um eine neue Zuordnung auf einer anderen Registerkarte zu

erstellen. Nachdem Sie die Metrik-Attribution erstellt haben, können Sie zu diesem Bildschirm zurückkehren und die Erstellung des Importjobs abschließen.

Weitere Informationen zu metrischen Attributionen finden Sie unter [Messung der Wirkung von Empfehlungen](#)

10. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#)
11. Wählen Sie Finish (Abschließen). Der Datenimportauftrag beginnt und die Seite Dashboard Overview (Dashboard-Übersicht) wird angezeigt. Der Dataset-Importvorgang ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Nachdem Sie Ihre Daten importiert haben, sind Sie bereit, eine Lösung zu erstellen. Weitere Informationen finden Sie unter [Erstellen einer Lösung und einer Lösungsversion](#).

Massendatensätze importieren (AWS CLI)

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Hinweise zum Aktualisieren vorhandener Daten finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Um Massendatensätze mit dem zu importierenAWS CLI, erstellen Sie mit dem [CreateDatasetImportJob](#) Befehl einen Job zum Importieren von Datensätzen. Wenn Sie zuvor einen Datensatz-Importauftrag für einen Datensatz erstellt haben, können Sie mithilfe des Importmodus-Parameters angeben, wie die neuen Daten hinzugefügt werden sollen. Ein Codebeispiel finden Sie unter [Massendatensätze werden aktualisiert \(AWS CLI\)](#).

Massendatensätze importieren (AWS CLI)

1. Erstellen Sie den Dataset-Importauftrag, indem Sie den folgenden Befehl ausführen. Geben Sie den Amazon-Ressourcennamen (ARN) für Ihren Datensatz an und geben Sie den Pfad zu Ihrem

Amazon S3-Bucket an, in dem Sie die Trainingsdaten gespeichert haben. Verwenden Sie die folgende Syntax für den Pfad:

```
s3://<name of your S3 bucket>/<folder path>/<CSV filename>
```

Wenn sich Ihre CSV-Dateien in einem Ordner in Ihrem Amazon S3 S3-Bucket befinden und Sie mit einem Datensatz-Importjob mehrere CSV-Dateien in einen Datensatz hochladen möchten, können Sie den Pfad zu dem Ordner angeben. Amazon Personalize verwendet nur die Dateien in der ersten Ebene Ihres Ordners, es verwendet keine Daten in Unterordnern. Verwenden Sie die folgende Syntax mit einem / nach dem Ordernamen:

```
s3://<name of your S3 bucket>/<folder path>/
```

Geben Sie die AWS Identity and Access Management (IAM) -Rolle Amazon Resource Name (ARN) an, in [Erstellen einer IAM-Rolle für Amazon Personalize](#) der Sie erstellt haben. Der Standardwert `import-mode` ist `FULL`. Weitere Informationen finden Sie unter [Aktualisierung vorhandener Massendaten](#). Weitere Informationen über den Vorgang finden Sie im Abschnitt [CreateDatasetImportJob](#).

```
aws personalize create-dataset-import-job \  
--job-name dataset import job name \  
--dataset-arn dataset arn \  
--data-source dataLocation=s3://bucketname/filename \  
--role-arn roleArn \  
--import-mode FULL
```

Der ARN des Datensatz-Importauftrags wird angezeigt, wie im folgenden Beispiel gezeigt.

```
{  
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/  
DatasetImportJobName"  
}
```

- Überprüfen Sie mit dem Befehl `describe-dataset-import-job` den Status. Geben Sie den Dataset-Importauftrag-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen über den Vorgang finden Sie im Abschnitt [DescribeDatasetImportJob](#).

```
aws personalize describe-dataset-import-job \  
--dataset-import-job-arn dataset import job arn
```

Die Eigenschaften des Dataset-Importauftrags, einschließlich des Erstellungstatus, werden angezeigt. Anfänglich wird das als CREATE PENDING status angezeigt.

```
{
  "datasetImportJob": {
    "jobName": "Dataset Import job name",
    "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/DatasetImportJobArn",
    "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/DatasetGroupName/INTERACTIONS",
    "dataSource": {
      "dataLocation": "s3://<bucketname>/ratings.csv"
    },
    "importMode": "FULL",
    "roleArn": "role-arn",
    "status": "CREATE PENDING",
    "creationDateTime": 1542392161.837,
    "lastUpdatedDateTime": 1542393013.377
  }
}
```

Der Dataset-Importvorgang ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Nachdem Sie Ihre Daten in die entsprechenden Datensätze in der Datensatzgruppe importiert haben, können Sie eine Lösungsversion (trainiertes Modell) erstellen. Weitere Informationen finden Sie unter [Erstellen einer Lösung und einer Lösungsversion](#).

Import von Massendatensätzen (AWSSDKs)

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Hinweise zum Aktualisieren vorhandener Daten finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Um Daten zu importieren, erstellen Sie mit der [CreateDatasetImportJob](#) Operation einen Job zum Importieren von Datensätzen. Der folgende Code zeigt, wie Sie einen Datensatz-Importjob erstellen.

Geben Sie den Jobnamen, `datasetArn` den Amazon-Ressourcennamen (ARN) Ihres Datensatzes und den Pfad `dataLocation` zu Ihrem Amazon S3-Bucket ein, in dem Sie die Trainingsdaten gespeichert haben. Verwenden Sie die folgende Syntax für den Pfad:

```
s3://<name of your S3 bucket>/<folder path>/<CSV filename>.csv
```

Wenn sich Ihre CSV-Dateien in einem Ordner in Ihrem Amazon S3 S3-Bucket befinden und Sie mit einem Datensatz-Importjob mehrere CSV-Dateien in einen Datensatz hochladen möchten, können Sie den Pfad zu dem Ordner angeben. Amazon Personalize verwendet nur die Dateien in der ersten Ebene Ihres Ordners, es verwendet keine Daten in Unterordnern. Verwenden Sie die folgende Syntax mit einem / nach dem Ordernamen:

```
s3://<name of your S3 bucket>/<folder path>/
```

Geben Sie für die die `roleArn` AWS Identity and Access Management (IAM) -Rolle an, die Amazon Personalize Berechtigungen für den Zugriff auf Ihren S3-Bucket erteilt. Siehe [Erstellen einer IAM-Rolle für Amazon Personalize](#). Der Standardwert `importMode` ist FULL. Weitere Informationen finden Sie unter [Aktualisierung von Massendatensätzen \(AWSSDKs\)](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']
```

```
print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,

                                                    String jobName,
                                                    String datasetArn,
                                                    String s3BucketPath,
                                                    String roleArn,
                                                    ImportMode importMode) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
            .datasetArn(datasetArn)
            .dataSource(importDataSource)
            .jobName(jobName)
            .roleArn(roleArn)
            .importMode(importMode)
            .build();

        datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
            .datasetImportJobArn();

        DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
            .datasetImportJobArn(datasetImportJobArn)
            .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;
```

```
while (Instant.now().getEpochSecond() < maxTime) {

    DatasetImportJob datasetImportJob = personalizeClient
        .describeDatasetImportJob(describeDatasetImportJobRequest)
        .datasetImportJob();

    status = datasetImportJob.status();
    System.out.println("Dataset import job status: " + status);

    if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
        break;
    }
    try {
        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
    datasetArn: 'DATASET_ARN', /* required */
    dataSource: {
```

```
    dataLocation: 's3://<name of your S3 bucket>/<folderName>/<CSVfilename>.csv' /*
required */
  },
  jobName: 'NAME',          /* required */
  roleArn: 'ROLE_ARN',     /* required */
  importMode: "FULL"       /* optional, default is FULL */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Die Antwort auf den [DescribeDatasetImportJob](#) Vorgang beinhaltet den Status des Vorgangs.

Sie müssen warten, bis sich der Status in ACTIVE ändert, bevor Sie mithilfe der Daten ein Modell schulen können.

Der Dataset-Importvorgang ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Nachdem Sie Daten in einen Amazon Personalize Personalize-Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Weitere Informationen finden Sie unter [Daten verwalten](#).

Nachdem Sie Ihre Daten in die entsprechenden Datensätze in der Datensatzgruppe importiert haben, können Sie eine Lösungsversion (trainiertes Modell) erstellen. Weitere Informationen finden Sie unter [Erstellen einer Lösung und einer Lösungsversion](#).

Einzelne Datensätze importieren

Nachdem Sie den Vorgang abgeschlossen haben [Einen Datensatz und ein Schema erstellen](#), können Sie einzelne Datensätze, einschließlich Artikelinteraktionen, Benutzer, Artikel, Aktionen oder Aktionsinteraktionen, in einen vorhandenen Datensatz importieren. Wenn Sie Daten einzeln importieren, können Sie Ihren Amazon Personalize-Datensätzen kleine Stapel von Datensätzen

hinzufügen, wenn Ihr Katalog wächst. Sie können bis zu 10 Datensätze pro einzelnen Importvorgang importieren.

Wenn Sie Apache Kafka verwenden, können Sie den Kafka-Konnektor für Amazon Personalize verwenden, um Daten in Echtzeit zu Amazon Personalize zu streamen. Weitere Informationen finden Sie unter [Kafka Connector for Amazon Personalize](#) im `personalize-kafka-connector` GitHub-Repository.

Wenn Sie über eine große Menge an historischen Datensätzen verfügen, empfehlen wir, dass Sie zuerst Daten in großen Mengen importieren und dann bei Bedarf Daten einzeln importieren. Siehe [Daten direkt in Amazon Personalize Personalize-Datensätze importieren](#).

Filtern Sie Aktualisierungen nach Importen einzelner Datensätze

Amazon Personalize aktualisiert alle Filter, die Sie in der Datensatzgruppe erstellt haben, innerhalb von 20 Minuten nach dem letzten Einzelimport mit Ihren neuen Interaktions-, Artikel- und Benutzerdaten. Dieses Update ermöglicht es Ihren Kampagnen, Ihre neuesten Daten beim Filtern von Empfehlungen für Ihre Nutzer zu verwenden.

Wenn Sie bereits einen Empfehlungsgeber erstellt oder eine benutzerdefinierte Lösungsversion mit einer Kampagne bereitgestellt haben, hängt die Art und Weise, wie neue individuelle Datensätze die Empfehlungen beeinflussen, vom verwendeten Domain-Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Themen

- [Interaktionen einzeln importieren](#)
- [Einzelnes Importieren von Benutzern](#)
- [Einzelnes Importieren von Elementen](#)
- [Aktionen einzeln importieren](#)

Interaktionen einzeln importieren

Nachdem Sie [Einen Datensatz und ein Schema erstellen](#) die Erstellung eines Datensatzes mit Artikelinteraktionen abgeschlossen haben, können Sie ein oder mehrere neue Ereignisse einzeln in den Datensatz importieren. Um [Interaktionsereignisse](#) einzeln zu importieren, erstellen Sie einen [Event-Tracker](#) und importieren dann ein oder mehrere Ereignisse in Ihren Datensatz mit Artikelinteraktionen. Sie können historische einzelne Interaktionsereignisse mit der Amazon

Personalize Personalize-Konsole importieren oder historische Ereignisse oder Echtzeitereignisse mithilfe der AWS Command Line Interface (AWS CLI) oder der AWS SDKs importieren.

Dieser Abschnitt enthält Informationen zum Importieren von Ereignissen mit der Amazon Personalize Personalize-Konsole. Wir empfehlen, die Amazon Personalize-Konsole zu verwenden, um nur historische Ereignisse zu importieren. Informationen zur Verwendung der AWS CLI AWS SDKs zur Aufzeichnung von Ereignissen in Echtzeit finden Sie unter [Ereignisse aufzeichnen](#)

Informationen darüber, wie Amazon Personalize Filter für neue Datensätze aktualisiert und wie neue Datensätze Empfehlungen beeinflussen, finden Sie unter [Einzelne Datensätze importieren](#).

Themen

- [Einen Event-Tracker \(Konsole\) erstellen](#)
- [Ereignisse einzeln importieren \(Konsole\)](#)

Einen Event-Tracker (Konsole) erstellen

Note

Wenn Sie einen Event-Tracker erstellt haben, können Sie zu springen [Ereignisse einzeln importieren \(Konsole\)](#).

Bevor Sie ein Ereignis in einen Interactions-Datensatz importieren können, müssen Sie einen [Event-Tracker](#) für die Datensatzgruppe erstellen.

Um einen Event-Tracker (Konsole) zu erstellen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen die Datensatzgruppe mit dem Datensatz Artikelinteraktionen aus, in die Sie Ereignisse importieren möchten.
3. Wählen Sie im Dashboard für die Datensatzgruppe unter Install Event Ingestion SDK die Option Start aus.
4. Geben Sie auf der Seite Tracker konfigurieren unter Tracker-Konfigurationen für Tracker-Name einen Namen für den Event-Tracker ein und wählen Sie Weiter aus.

5. Auf der Seite SDK installieren werden die Tracking-ID für den neuen Event-Tracker sowie Anweisungen zur Verwendung AWS Amplify oder AWS Lambda zum Streamen von Ereignisdaten angezeigt.

Sie können diese Informationen ignorieren, da Sie die Amazon Personalize-Konsole verwenden, um Veranstaltungsdaten hochzuladen. Wenn Sie mit AWS Amplify oder AWS Lambda in future Eventdaten streamen möchten, können Sie sich diese Informationen ansehen, indem Sie auf der Seite Event-Tracker den Event-Tracker auswählen.

6. Wählen Sie Finish (Abschließen). Sie können jetzt Ereignisse mit der Konsole importieren (Ereignisse anzeigen) [Ereignisse einzeln importieren \(Konsole\)](#) oder mithilfe der PutEvents Operation in Echtzeit aufzeichnen (siehe [Ereignisse aufzeichnen](#)).

Ereignisse einzeln importieren (Konsole)

Nachdem Sie einen Event-Tracker erstellt haben, können Sie Ereignisse einzeln in einen Datensatz mit Artikelinteraktionen importieren. Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits einen Datensatz mit Artikelinteraktionen erstellt haben. Hinweise zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Um Ereignisse einzeln zu importieren (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen die Datensatzgruppe mit dem Datensatz Artikelinteraktionen aus, in die Sie Ereignisse importieren möchten.
3. Wählen Sie im Navigationsbereich Datensätze aus.
4. Wählen Sie auf der Seite „Datensätze“ den Datensatz Interaktionen aus.
5. Wählen Sie oben rechts auf der Seite mit den Datensatz-Details die Option Datensatz ändern und anschließend Datensatz erstellen aus.
6. Geben Sie auf der Seite Benutzerelementinteraktionsdatensätze erstellen für Datensatzeingabe die Ereignisdetails im JSON-Format ein. Die Feldnamen und Werte des Ereignisses müssen dem Schema entsprechen, das Sie bei der Erstellung des Datensatzes „Artikelinteraktionen“ verwendet haben. Amazon Personalize stellt eine JSON-Vorlage mit Feldnamen und Datentypen aus diesem Schema bereit. Sie können bis zu 10 Ereignisse gleichzeitig importieren.
7. Wählen Sie Datensatz (e) erstellen. Als Antwort wird das Ergebnis des Imports aufgeführt und eine Erfolgs- oder Fehlschlagsmeldung angezeigt.

Einzelnes Importieren von Benutzern

Nachdem Sie [Einen Datensatz und ein Schema erstellen](#) die Erstellung eines Benutzerdatensatzes abgeschlossen haben, können Sie einzeln einen oder mehrere neue Benutzer in den Datensatz importieren. Durch den individuellen Import von Benutzern können Sie Ihren Benutzerdatensatz mit kleinen Batch-Importen auf dem neuesten Stand halten, wenn Ihr Katalog wächst. Sie können bis zu 10 Benutzer gleichzeitig importieren. Wenn Sie eine große Anzahl neuer Benutzer haben, empfehlen wir Ihnen, zuerst Daten in großen Mengen zu importieren und dann die Benutzerdaten nach Bedarf einzeln zu importieren. Siehe [Daten direkt in Amazon Personalize Personalize-Datensätze importieren](#).

Sie können die Amazon-Personalize-Konsole, die AWS Command Line Interface (AWS CLI) oder AWS SDKs verwenden, um Benutzer zu importieren. Wenn Sie einen Benutzer mit dem importieren, das `userId` sich bereits in Ihrem Benutzerdatensatz befindet, ersetzt Amazon Personalize den Benutzer durch den neuen. Sie können bis zu 10 Benutzer gleichzeitig importieren.

Informationen darüber, wie Amazon Personalize Filter für neue Datensätze aktualisiert und wie sich neue Datensätze auf Empfehlungen auswirken, finden Sie unter [Einzelne Datensätze importieren](#).

Themen

- [Einzelnes Importieren von Benutzern \(Konsole\)](#)
- [Einzelnes Importieren von Benutzern \(AWS CLI\)](#)
- [Einzelnes Importieren von Benutzern \(AWS SDKs\)](#)

Einzelnes Importieren von Benutzern (Konsole)

Sie können bis zu 10 Benutzer gleichzeitig importieren. Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits einen Benutzerdatensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

So importieren Sie Benutzer einzeln (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen die Datensatzgruppe mit dem Benutzerdatensatz aus, in den Sie den Benutzer importieren möchten.
3. Wählen Sie im Navigationsbereich Datensätze aus.
4. Wählen Sie auf der Seite Datensätze den Datensatz Benutzer aus.

5. Wählen Sie auf der Datensatzdetailseite oben rechts Datensatz ändern und dann Datensatz erstellen aus.
6. Geben Sie auf der Seite Benutzerdatensatz(e) erstellen für Datensatzeingabe die Benutzerdetails im JSON-Format ein. Die Feldnamen und -werte des Benutzers müssen mit dem Schema übereinstimmen, das Sie beim Erstellen des Datensatzes Benutzer verwendet haben. Amazon Personalize stellt eine JSON-Vorlage mit Feldnamen und Datentypen aus diesem Schema bereit.
7. Wählen Sie Datensatz(e) erstellen aus. In Antwort wird das Ergebnis des Imports aufgeführt und eine Erfolgs- oder Fehlermeldung angezeigt.

Einzelnes Importieren von Benutzern (AWS CLI)

Fügen Sie Ihrem Benutzerdatensatz mit der [-PutUsers](#) Operation einen oder mehrere Benutzer hinzu. Sie können bis zu 10 Benutzer mit einem einzigen PutUsers Aufruf importieren. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Benutzerdatensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Verwenden Sie den folgenden `put-users` Befehl, um einen oder mehrere Benutzer mit der hinzuzufügen AWS CLI. Ersetzen Sie `dataset arn` durch den Amazon-Ressourcennamen (ARN) Ihres Datensatzes und `user Id` durch die ID des Benutzers. Wenn sich ein Benutzer mit demselben bereits in Ihrem Benutzerdatensatz `userId` befindet, ersetzt Amazon Personalize ihn durch den neuen.

Ersetzen Sie `properties` für jedes Feld in Ihrem Benutzerdatensatz die durch `propertyName` den Feldnamen aus Ihrem Schema in Groß-/Kleinschreibung. `GENDER` wäre beispielsweise `gender` und `MEMBERSHIP_TYPE` wäre `membershipType`. Ersetzen Sie durch `user data` die Daten für den Benutzer. Um bei kategorialen Zeichenfolgendaten mehrere Kategorien für eine einzelne Eigenschaft einzuschließen, trennen Sie jede Kategorie durch eine Pipe (`|`). Zum Beispiel `"Premium Class | Legacy Member"`.

```
aws personalize-events put-users \
  --dataset-arn dataset arn \
  --users '[{
    "userId": "user Id",
    "properties": "{\"propertyName\": \"\i>user data\"}"
  },
  {
    "userId": "user Id",
    "properties": "{\"propertyName\": \"\i>user data\"}"
```

```
}]'
```

Einzelnes Importieren von Benutzern (AWS SDKs)

Fügen Sie Ihrem Benutzerdatensatz mit der [-PutUsers](#) Operation einen oder mehrere Benutzer hinzu. Wenn sich ein Benutzer mit demselben bereits in Ihrem Benutzerdatensatz `userId` befindet, ersetzt Amazon Personalize ihn durch den neuen. Sie können bis zu 10 Benutzer mit einem einzigen `PutUsers` Aufruf importieren. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Benutzerdatensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Der folgende Code zeigt, wie Sie Ihrem Benutzerdatensatz einen oder mehrere Benutzer hinzufügen. Übergeben Sie für jeden Eigenschaftsnamenparameter den Feldnamen aus Ihrem Schema in Groß- und Kleinschreibung. `GENDER` wäre beispielsweise `gender` und `MEMBERSHIP_TYPE` wäre `membershipType`. Übergeben Sie für jeden Eigenschaftswertparameter die Daten für den Benutzer.

Für kategoriale Zeichenfolgendaten, um mehrere Kategorien für eine einzelne Eigenschaft einzuschließen, trennen Sie jede Kategorie durch eine Pipe (`|`). Zum Beispiel `"Premium class | Legacy Member"`.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_users(
    datasetArn = 'dataset arn',
    users = [{
        'userId': 'user ID',
        'properties': "{\"propertyName\": \"user data\"}"
    },
    {
        'userId': 'user ID',
        'properties': "{\"propertyName\": \"user data\"}"
    }
    ]
)
```

SDK for Java 2.x

```
public static int putUsers(PersonalizeEventsClient personalizeEventsClient,
```

```
        String datasetArn,
        String user1Id,
        String user1PropertyName,
        String user1PropertyValue,
        String user2Id,
        String user2PropertyName,
        String user2PropertyValue) {

    int responseCode = 0;
    ArrayList<User> users = new ArrayList<>();

    try {
        User user1 = User.builder()
            .userId(user1Id)
            .properties(String.format("{\\"%1$s\\": \\"%2$s\\"}", user1PropertyName,
user1PropertyValue))
            .build();

        users.add(user1);

        User user2 = User.builder()
            .userId(user2Id)
            .properties(String.format("{\\"%1$s\\": \\"%2$s\\"}", user2PropertyName,
user2PropertyValue))
            .build();

        users.add(user2);

        PutUsersRequest putUsersRequest = PutUsersRequest.builder()
            .datasetArn(datasetArn)
            .build();

        responseCode =
personalizeEventsClient.putUsers(putUsersRequest).sdkHttpResponse().statusCode();
        System.out.println("Response code: " + responseCode);
        return responseCode;

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return responseCode;
}
```

SDK for JavaScript v3

```
import {
  PutUsersCommand,
  PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
  region: "REGION",
});

// set the put users parameters
var putUsersParam = {
  datasetArn:
    "DATASET ARN",
  users: [
    {
      userId: "userId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
    {
      userId: "userId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
  ],
};

export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutUsersCommand(putUsersParam)
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```


Einzelnes Importieren von Elementen

Nachdem Sie [Einen Datensatz und ein Schema erstellen](#) die Erstellung eines Artikeldatensatzes abgeschlossen haben, können Sie ein oder mehrere neue Elemente einzeln in den Datensatz importieren. Durch den individuellen Import von Elementen können Sie Ihren Artikeldatensatz mit kleinen Batch-Importen auf dem neuesten Stand halten, wenn Ihr Katalog wächst. Sie können bis zu 10 Elemente gleichzeitig importieren. Wenn Sie eine große Menge neuer Elemente haben, empfehlen wir Ihnen, zuerst Daten in großen Mengen zu importieren und dann Elementdaten nach Bedarf einzeln zu importieren. Siehe [Daten direkt in Amazon Personalize Personalize-Datensätze importieren](#).

Sie können die Amazon-Personalize-Konsole, die AWS Command Line Interface (AWS CLI) oder AWS SDKs verwenden, um Elemente zu importieren. Wenn Sie ein Element importieren, das dem `itemId` Element entspricht, das sich bereits in Ihrem Artikel-Datensatz befindet, ersetzt Amazon Personalize es durch das neue Element.

Informationen darüber, wie Amazon Personalize Filter für neue Datensätze aktualisiert und wie sich neue Datensätze auf Empfehlungen auswirken, finden Sie unter [Einzelne Datensätze importieren](#).

Themen

- [Einzelnes Importieren von Elementen \(Konsole\)](#)
- [Einzelnes Importieren von Elementen \(AWS CLI\)](#)
- [Einzelnes Importieren von Elementen \(AWS SDKs\)](#)

Einzelnes Importieren von Elementen (Konsole)

Sie können bis zu 10 Elemente gleichzeitig in einen Artikeldatensatz importieren. Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits einen Artikel-Datensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

So importieren Sie Elemente einzeln (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen die Datensatzgruppe mit dem Elementdatensatz aus, in den Sie die Elemente importieren möchten.

3. Wählen Sie im Navigationsbereich Datensätze aus.
4. Wählen Sie auf der Seite Datensätze den Datensatz Elemente aus.
5. Wählen Sie oben rechts auf der Datensatzdetailseite Datensatz ändern und dann Datensatz erstellen aus.
6. Geben Sie auf der Seite Elementdatensatz(e) erstellen für Datensatzeingabe die Elementdetails im JSON-Format ein. Die Feldnamen und -werte des Elements müssen mit dem Schema übereinstimmen, das Sie beim Erstellen des Elemente-Datensatzes verwendet haben. Amazon Personalize stellt eine JSON-Vorlage mit Feldnamen und Datentypen aus diesem Schema bereit.
7. Wählen Sie Datensatz(e) erstellen aus. In Antwort wird das Ergebnis des Imports aufgeführt und eine Erfolgs- oder Fehlermeldung angezeigt.

Einzelnes Importieren von Elementen (AWS CLI)

Fügen Sie Ihrem Elementdatensatz mithilfe der [PutItems](#) Operation ein oder mehrere Elemente hinzu. Sie können bis zu 10 Elemente mit einem einzigen `PutItems` Aufruf importieren. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Artikel-Datensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Verwenden Sie den folgenden `put-items` Befehl, um ein oder mehrere Elemente mit der hinzuzufügen AWS CLI. Ersetzen Sie `dataset arn` durch den Amazon-Ressourcennamen (ARN) Ihres Datensatzes und `item Id` durch die ID des Elements. Wenn sich ein Element mit demselben bereits in Ihrem Artikel-Datensatz `itemId` befindet, ersetzt Amazon Personalize es durch das neue.

Ersetzen Sie `properties` für jedes Feld in Ihrem Artikel-Datensatz die durch `propertyName` den Feldnamen aus Ihrem Schema in camel-Fall. `GENRES` wäre beispielsweise `genres` und `CREATION_TIMESTAMP` wäre `creationTimestamp`. Ersetzen Sie durch `item data` die Daten für das Element. `-CREATION_TIMESTAMP` Daten müssen im [Unix-Zeitformat](#) und in Sekunden vorliegen. Um bei kategorialen Zeichenfolgendaten mehrere Kategorien für eine einzelne Eigenschaft einzuschließen, trennen Sie jede Kategorie durch eine Pipe (`|`). Zum Beispiel `\\"Horror|Action\\"`.

```
aws personalize-events put-items \  
  --dataset-arn dataset arn \  
  --items '[{  
    "itemId": "item Id",  
    "properties": "{\"propertyName\": \"\iitem data\\""}
```

```
},  
{  
  "itemId": "item Id",  
  "properties": "{\"propertyName\": \"item data\"}"  
}]'
```

Einzelnes Importieren von Elementen (AWS SDKs)

Fügen Sie Ihrem Elementdatensatz mithilfe der [-PutItems](#) Operation ein oder mehrere Elemente hinzu. Sie können bis zu 10 Elemente mit einem einzigen PutItems Aufruf importieren. Wenn sich ein Element mit demselben bereits in Ihrem Artikel-Datensatz `itemId` befindet, ersetzt Amazon Personalize es durch das neue. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Artikel-Datensatz erstellt haben. Informationen zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Der folgende Code zeigt, wie Sie Ihrem Artikel-Datensatz ein oder mehrere Elemente hinzufügen. Übergeben Sie für jeden Eigenschaftsnamenparameter den Feldnamen aus Ihrem Schema in Groß- und Kleinschreibung. GENRES wäre beispielsweise `genres` und `CREATION_TIMESTAMP` wäre `creationTimestamp`. Übergeben Sie für jeden Eigenschaftswertparameter die Daten für das Element. `-CREATION_TIMESTAMP` Daten müssen im [Unix-Zeitformat](#) und in Sekunden vorliegen.

Um bei kategorialen Zeichenfolgendaten mehrere Kategorien für eine einzelne Eigenschaft einzuschließen, trennen Sie jede Kategorie durch eine Pipe (`|`). Zum Beispiel `"Horror|Action"`.

SDK for Python (Boto3)

```
import boto3  
  
personalize_events = boto3.client(service_name='personalize-events')  
  
personalize_events.put_items(  
    datasetArn = 'dataset arn',  
    items = [{  
        'itemId': 'item ID',  
        'properties': "{\"propertyName\": \"item data\"}"  
    },  
    {  
        'itemId': 'item ID',  
        'properties': "{\"propertyName\": \"item data\"}"  
    }  
])
```

SDK for Java 2.x

```
public static int putItems(PersonalizeEventsClient personalizeEventsClient,
                          String datasetArn,
                          String item1Id,
                          String item1PropertyName,
                          String item1PropertyValue,
                          String item2Id,
                          String item2PropertyName,
                          String item2PropertyValue) {

    int responseCode = 0;
    ArrayList<Item> items = new ArrayList<>();

    try {
        Item item1 = Item.builder()
            .itemId(item1Id)
            .properties(String.format("{\"%1$s\": \"%2$s\"}",
                                     item1PropertyName, item1PropertyValue))
            .build();

        items.add(item1);

        Item item2 = Item.builder()
            .itemId(item2Id)
            .properties(String.format("{\"%1$s\": \"%2$s\"}",
                                     item2PropertyName, item2PropertyValue))
            .build();

        items.add(item2);

        PutItemsRequest putItemsRequest = PutItemsRequest.builder()
            .datasetArn(datasetArn)
            .items(items)
            .build();

        responseCode =
personalizeEventsClient.putItems(putItemsRequest).sdkHttpResponse().statusCode();
        System.out.println("Response code: " + responseCode);
        return responseCode;

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

```
    return responseCode;
  }
```

SDK for JavaScript v3

```
import {
  PutItemsCommand,
  PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
  region: "REGION",
});

// set the put items parameters
var putItemsParam = {
  datasetArn:
    "DATASET ARN",
  items: [
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
  ],
};
export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutItemsCommand(putItemsParam)
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Aktionen einzeln importieren

Nachdem Sie [Einen Datensatz und ein Schema erstellen](#) die Erstellung eines [Aktionsdatensatzes](#) abgeschlossen haben, können Sie eine oder mehrere neue Aktionen einzeln in den Datensatz importieren. Wenn Sie Aktionen einzeln importieren, halten Sie Ihren Aktionsdatensatz mit kleinen Batch-Importen auf dem neuesten Stand, wenn Ihr Katalog wächst. Sie können bis zu 10 Aktionen gleichzeitig importieren. Wenn Sie über eine große Anzahl neuer Aktionen verfügen, empfehlen wir, dass Sie zuerst Daten in großen Mengen importieren und dann die Aktionsdaten bei Bedarf einzeln importieren. Siehe [Daten direkt in Amazon Personalize Personalize-Datensätze importieren](#).

Sie können die Amazon Personalize-Konsole, die AWS Command Line Interface (AWS CLI) oder AWS SDKs verwenden, um Aktionen zu importieren. Wenn Sie eine Aktion importieren, die mit einer Aktion `actionId` identisch ist, die sich bereits in Ihrem Aktionen-Datensatz befindet, ersetzt Amazon Personalize sie durch die neue Aktion.

Informationen darüber, wie neue Datensätze Empfehlungen beeinflussen, finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Themen

- [Aktionen einzeln importieren \(Konsole\)](#)
- [Aktionen einzeln importieren \(AWS CLI\)](#)
- [Aktionen einzeln importieren \(AWSSDKs\)](#)

Aktionen einzeln importieren (Konsole)

Sie können bis zu 10 Aktionen gleichzeitig in einen Aktionen-Datensatz importieren. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Actions-Datensatz erstellt haben. Hinweise zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Um Aktionen einzeln zu importieren (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatz-Gruppen die Datensatz-Gruppe mit dem Action-Datensatz aus, zu dem Sie etwas hinzufügen möchten.

3. Wählen Sie im Navigationsbereich Datensätze aus.
4. Wählen Sie auf der Seite Datensätze den Datensatz Aktionen aus.
5. Wählen Sie oben rechts auf der Seite mit den Datensatz-Details die Option Datensatz ändern und anschließend Datensatz erstellen aus.
6. Geben Sie auf der Seite Aktionsdatensätze erstellen für Datensatzeingabe die Aktionsdetails im JSON-Format ein. Die Feldnamen und Werte der Aktion müssen dem Schema entsprechen, das Sie bei der Erstellung des Actions-Datasets verwendet haben. Amazon Personalize stellt eine JSON-Vorlage mit Feldnamen und Datentypen aus diesem Schema bereit.
7. Wählen Sie Datensatz (e) erstellen. Als Antwort wird das Ergebnis des Imports aufgeführt und eine Erfolgs- oder Fehlschlagsmeldung angezeigt.

Aktionen einzeln importieren (AWS CLI)

Fügen Sie mithilfe der `PutActions` API-Operation eine oder mehrere Aktionen zu Ihrem Aktionsdatensatz hinzu. Sie können bis zu 10 Aktionen gleichzeitig importieren. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Actions-Datensatz erstellt haben. Hinweise zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Verwenden Sie den folgenden `put-actions` Befehl, um eine oder mehrere Aktionen mit dem AWS CLI hinzuzufügen. `dataset arn` Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) Ihres Datensatzes und `actionId` durch die ID der Aktion. Wenn sich eine Aktion mit derselben bereits in Ihrem Aktionen-Datensatz `actionId` befindet, ersetzt Amazon Personalize sie durch die neue.

Denn `properties` ersetzen Sie für jedes Feld in Ihrem Actions-Datensatz das `propertyName` durch den Feldnamen aus Ihrem Schema in Camelcase. Zum Beispiel wäre `ACTION_EXPIRATION_TIMESTAMP` und `CREATION_TIMESTAMP` wäre `CreationTimeStamp`. `actionExpirationTimestamp property data` Ersetzen Sie es durch die Daten für die Eigenschaft.

```
aws personalize-events put-actions \  
  --dataset-arn dataset arn \  
  --actions '[{  
    "actionId": "actionId",  
    "properties": "{\"propertyName\": \"\property data\"}"  
  },  
  {  
    "actionId": "actionId",  
    "properties": "{\"propertyName\": \"\property data\"}"
```

```
}]'
```

Aktionen einzeln importieren (AWSSDKs)

Fügen Sie mithilfe der `PutActions` Operation eine oder mehrere Aktionen zu Ihrem Aktionen-Datensatz hinzu. Sie können bis zu 10 Aktionen mit einem einzigen `PutActions` Aufruf importieren. Wenn sich eine Aktion mit derselben bereits in Ihrem Aktionen-Datensatz `actionId` befindet, ersetzt Amazon Personalize sie durch die neue. In diesem Abschnitt wird davon ausgegangen, dass Sie bereits einen Actions-Datensatz erstellt haben. Hinweise zum Erstellen von Datensätzen finden Sie unter [Einen Datensatz und ein Schema erstellen](#).

Der folgende Code zeigt, wie Sie Ihrem Aktionen-Datensatz eine oder mehrere Aktionen hinzufügen. Geben Sie für jede Aktion den `actionId`. Wenn sich eine Aktion mit derselben bereits in Ihrem Aktionen-Datensatz `actionId` befindet, ersetzt Amazon Personalize sie durch die neue. Denn `properties` ersetzen Sie für jedes weitere Feld in Ihrem Actions-Datensatz das `propertyName` durch den Feldnamen aus Ihrem Schema in Camelcase. Zum Beispiel wäre `ACTION_EXPIRATION_TIMESTAMP` und `CREATION_TIMESTAMP` wäre `CreationTimeStamp`. `actionExpirationTimestamp property data` Ersetzen Sie es durch die Daten für die Eigenschaft.

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_actions(
    datasetArn = 'dataset arn',
    actions = [{
        'actionId': 'actionId',
        'properties': "{\\"propertyName\": \\"property value\\"}"
    },
    {
        'actionId': 'actionId',
        'properties': "{\\"propertyName\": \\"property value\\"}"
    }
    ]
)
```


Schritt 3: Erstellen von Empfehlungen oder benutzerdefinierten Ressourcen

Nachdem Sie Daten importiert haben, können Sie Empfehlungen oder benutzerdefinierte Ressourcen erstellen. Verwenden Sie diese Ressourcen, um Empfehlungen zu erhalten. Die Ressourcen, die Sie erstellen, hängen vom Typ Ihrer Datensatzgruppe ab:

- Für Domain-Dataset-Gruppen erstellen Sie Empfehlungen für vordefinierte Anwendungsfälle, die auf Ihrer Domain basieren. Sie verwenden die Empfehlungen, um Empfehlungen zu erhalten. Weitere Informationen zu den verfügbaren Anwendungsfällen finden Sie unter [Auswahl eines Anwendungsfalls](#). Sie können einer Domäne-Datensatz-Datensatz-Gruppe auch benutzerdefinierte Ressourcen hinzufügen. Dazu gehören Lösungen und Lösungsversionen, die für benutzerdefinierte Anwendungsfälle geschult wurden.
- Für benutzerdefinierte Datensatzgruppen konfigurieren Sie eine Lösung mit einem Rezept. Dann erstellen Sie eine Lösungsversion (trainieren Sie ein Modell). Weitere Informationen zu den verfügbaren Rezepten finden Sie unter [Ein Rezept wählen](#).

Für Empfehlungen in Echtzeit setzen Sie die Lösungsversion in einer Kampagne ein. Um stapelweise Empfehlungen und Benutzersegmente zu erstellen, benötigen Sie keine Kampagne.

Themen

- [Domäne erstellen](#)
- [Benutzerdefinierte Ressourcen erstellen](#)

Domäne erstellen

Nachdem Sie Daten importiert haben, können Sie mit der Erstellung, Auswertung und Verwaltung von Empfehlungen in Ihrer Domain-Dataset-Gruppe beginnen. Ein Empfehlungsgeber ist eine Ressource für Domain-Dataset-Gruppen, die Empfehlungen generiert. Sie verwenden es in Ihrer Anwendung, um in Echtzeit Empfehlungen für den [GetRecommendations](#) Vorgang zu erhalten.

Themen

- [Erstellen von Empfehlungsgebern](#)
- [Bewertung eines Empfehlungsgebers](#)
- [Verwalten von Empfehlungsgebern](#)

Erstellen von Empfehlungsgebern

Nachdem Sie eine Domain-Datensatzgruppe erstellt und Daten importiert haben, können Sie Empfehlungsprogramme für Ihre Domain-Anwendungsfälle erstellen. Ein Empfehler ist eine Domain-Datensatzgruppenressource, die Empfehlungen generiert. Sie verwenden einen Empfehler in Ihrer Anwendung, um Empfehlungen in Echtzeit mit der [-GetRecommendations](#) Operation zu erhalten.

Wenn Sie einen Empfehler erstellen, geben Sie einen Anwendungsfall an und Amazon Personalize trainiert die Modelle, die den Empfehler unterstützen, mit den besten Konfigurationen für den Anwendungsfall. Jeder Anwendungsfall hat unterschiedliche API-Anforderungen für den Erhalt von Empfehlungen. Eine Liste der Anwendungsfälle für Empfehler nach Domain finden Sie unter [Auswahl eines Anwendungsfalls](#). Sie können maximal 15 Empfehler pro Region erstellen.

Amazon Personalize trainiert die Modelle, die Ihre Empfehlungsgeber unterstützen, automatisch alle 7 Tage neu. Dies ist ein vollständiges Neutraining, bei dem völlig neue Modelle auf der Grundlage der gesamten Daten in Ihren Datensätzen erstellt werden. Mit den wichtigsten Auswahlmöglichkeiten für Sie und Empfohlen für Ihre Anwendungsfälle aktualisiert Amazon Personalize die vorhandenen Modelle alle zwei Stunden, um neue Elemente in Empfehlungen für die Erkundung einzubeziehen.

Wenn Sie einen Empfehler erstellen, können Sie Elementmetadaten in Empfehlungen aktivieren. Weitere Informationen finden Sie unter [Aktivieren von Metadaten in Empfehlungen](#).

Sie können Empfehler mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.

Empfehlungsstatus

Ein Empfehler kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Um den Empfehlungsstatus zu erhalten, navigieren Sie zur Seite Empfehlungen in der Amazon-Personalize-Konsole oder verwenden Sie die [-DescribeRecommender](#) Operation.

Themen

- [Minimale Empfehlungsanforderungen pro Sekunde und Auto Scaling](#)
- [Aktivieren von Metadaten in Empfehlungen](#)

- [Erstellen von Empfehlern \(Konsole\)](#)
- [Erstellen von Empfehlungsgebern \(AWS CLI\)](#)
- [Erstellen von Empfehlern \(AWS SDKs\)](#)

Minimale Empfehlungsanforderungen pro Sekunde und Auto Scaling

Important

Ein hoher `minRecommendationRequestsPerSecond` Wert erhöht Ihre Rechnung. Wir empfehlen, mit 1 für `minRecommendationRequestsPerSecond` (Standard) zu beginnen. Verfolgen Sie Ihre Nutzung mithilfe von Amazon- CloudWatch Metriken und erhöhen Sie die `minRecommendationRequestsPerSecond` nach Bedarf.

Wenn Sie einen Empfehler erstellen, können Sie die Mindestempfehlungsanforderungen pro Sekunde des Empfehlers konfigurieren. Die minimalen Empfehlungsanforderungen pro Sekunde (`minRecommendationRequestsPerSecond`) geben den von Amazon Personalize bereitgestellten Durchsatz der Basisempfehlungsanforderung an. Der Standardwert `minRecommendationRequestsPerSecond` ist 1. Eine Empfehlungsanfrage ist eine einzelne `GetRecommendations` Operation. Der Anforderungsdurchsatz wird in Anforderungen pro Sekunde gemessen und Amazon Personalize verwendet Ihre Anforderungen pro Sekunde, um Ihre Anforderungen pro Stunde und den Preis Ihrer Empfehlungsnutzung abzuleiten.

Wenn Ihre Anforderungen pro Sekunde über `minRecommendationRequestsPerSecond` hinausgehen, skaliert Amazon Personalize die bereitgestellte Kapazität automatisch nach oben und unten, jedoch niemals unter `minRecommendationRequestsPerSecond`. Es gibt eine kurze Verzögerung, während die Kapazität erhöht wird, die zum Verlust von Anfragen führen kann.

Ihre Rechnung ist entweder der größere der Mindestanforderungen pro Stunde (basierend auf `minRecommendationRequestsPerSecond`) oder die tatsächliche Anzahl der Anforderungen. Der tatsächliche verwendete Anforderungsdurchsatz wird als durchschnittliche Anforderungen/ Sekunde innerhalb eines einstündigen Zeitfensters berechnet. Wir empfehlen, mit der Standard- zu beginnen `minRecommendationRequestsPerSecond`, Ihre Nutzung mithilfe von Amazon- CloudWatch Metriken zu verfolgen und dann die `minRecommendationRequestsPerSecond` nach Bedarf zu erhöhen.

Aktivieren von Metadaten in Empfehlungen

Important

Wenn Sie Metadaten in Empfehlungen aktivieren, fallen zusätzliche Kosten an. Weitere Informationen finden Sie unter [Amazon Personalize – Preise](#).

Wenn Sie einen Empfehler erstellen, können Sie die Option aktivieren, um Elementmetadaten aus Ihrem Artikeldatensatz in Empfehlungsergebnisse aufzunehmen. Wenn diese Option aktiviert ist, können Sie die Spalten aus Ihrem Artikel-Datensatz in Ihrer Empfehlungsanfrage angeben. Amazon Personalize gibt diese Daten für jedes Element in der Empfehlungsantwort zurück.

Sie können Metadaten verwenden, um Empfehlungen in Ihrer Benutzeroberfläche zu ergänzen, z. B. das Hinzufügen der Genre für Filme zu Karussells. Oder Sie können es verwenden, um die Empfehlungsqualität visuell zu bewerten. Wenn Sie generative KI in Ihrer App verwenden, können Sie die Metadaten in KI-Eingabeaufforderungen einfügen, um relevantere Inhalte zu generieren. Weitere Informationen zur Verwendung von Amazon Personalize mit generativer KI finden Sie unter [Amazon Personalize und generative KI](#).

- Um Metadaten mit der Amazon-Personalize-Konsole zu aktivieren, wählen Sie beim Erstellen des Empfehlers die Option Elemente in Metadaten in Empfehlungsergebnissen zurückgeben in Erweiterte Konfiguration aus. Weitere Informationen finden Sie unter [Erstellen von Empfehlern \(Konsole\)](#).
- Um Metadaten mit den - AWS SDKs oder `enableMetadataWithRecommendations` zu aktivieren AWS CLI, verwenden Sie die [CreateRecommender](#)-API-Operation und in der auf `recommenderConfig` eingestellten `true`. Weitere Informationen finden Sie unter [Erstellen von Empfehlungsgebern \(AWS CLI\)](#) oder [Erstellen von Empfehlern \(AWS SDKs\)](#).

Um Empfehlungen Metadaten hinzuzufügen, benötigen Sie einen Elementdatensatz mit einer Spalte mit Metadaten. Sie müssen die Metadaten beim Training nicht verwenden. Informationen zum Erstellen eines Datensatzes finden Sie unter [Einen Datensatz und ein Schema erstellen](#). Informationen zum Verwalten und Aktualisieren von Daten finden Sie unter [Daten verwalten](#).

Erstellen von Empfehlern (Konsole)

Important

Ein hoher `minRecommendationRequestsPerSecond` Wert erhöht Ihre Rechnung. Wir empfehlen, mit 1 für `minRecommendationRequestsPerSecond` (Standard) zu beginnen. Verfolgen Sie Ihre Nutzung mithilfe von Amazon- CloudWatch Metriken und erhöhen Sie die `minRecommendationRequestsPerSecond` nach Bedarf. Weitere Informationen finden Sie unter [Minimale Empfehlungsanforderungen pro Sekunde und Auto Scaling](#).

Erstellen Sie mit der Amazon-Personalize-Konsole wie folgt Empfehlungsprogramme für jeden Ihrer Anwendungsfälle. Wenn Sie gerade Ihre Domain-Datensatzgruppe erstellt haben und sich bereits auf der Übersichtsseite befinden, fahren Sie mit Schritt 3 fort.

So erstellen Sie Empfehlungsgeber

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Domänendatensatzgruppe aus.
3. Wählen Sie in Schritt 3 <Domänenname>-Empfehlungen verwenden und dann Empfehlungen erstellen aus.
4. Wählen Sie auf der Seite Anwendungsfälle auswählen die Anwendungsfälle aus, für die Sie Empfehlungsprogramme erstellen möchten, und geben Sie jedem einen Empfehlungsanbieternamen . Amazon Personalize erstellt einen Empfehler für jeden von Ihnen ausgewählten Anwendungsfall. Die verfügbaren Anwendungsfälle hängen von Ihrer Domain ab. Informationen zur Auswahl eines Anwendungsfalls finden Sie unter [Auswahl eines Anwendungsfalls](#).
5. Wählen Sie Weiter aus.
6. Konfigurieren Sie auf der Seite Erweiterte Konfiguration jeden Empfehler entsprechend Ihren Geschäftsanforderungen:
 - Für jeden Datensatz, der vom Anwendungsfall des Empfehlungsgebers verwendet wird, können Sie die Spalten auswählen, die Amazon Personalize beim Training der Modelle berücksichtigt, die Ihren Empfehlungsgeber unterstützen. Standardmäßig verwendet Amazon Personalize alle Spalten, die beim Training verwendet werden können. Spalten mit dem booleschen Datentyp und benutzerdefinierten Zeichenfolgenfeldern, die nicht kategorisch oder

textuell sind, werden beim Training nicht verwendet. Sie können EVENT_TYPE-Spalten nicht ausschließen.

Sie können die beim Training verwendeten Spalten ändern, um zu steuern, welche Daten Amazon Personalize beim Erstellen Ihres Empfehlens verwendet. Sie können dies tun, um mit verschiedenen Kombinationen von Trainingsdaten zu experimentieren. Oder Sie schließen Spalten ohne aussagekräftige Daten aus. Möglicherweise haben Sie eine Spalte, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte vom Training ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

- Sie können Mindestempfehlungsanforderungen pro Sekunde ändern, um eine neue Mindestanforderungskapazität für Ihren Empfehler anzugeben. Ein hoher `minRecommendationRequestsPerSecond` Wert erhöht Ihre -Rechnung. Wir empfehlen, mit 1 (Standard) zu beginnen. Verfolgen Sie Ihre Nutzung mithilfe von Amazon- CloudWatch Metriken und erhöhen Sie die `minRecommendationRequestsPerSecond` nach Bedarf. Weitere Informationen finden Sie unter [Minimale Empfehlungsanforderungen pro Sekunde und Auto Scaling](#).
- Wenn Sie die Möglichkeit haben möchten, Metadaten von Elementdatensätzen in Empfehlungen aufzunehmen, wählen Sie Metadaten von Elementen in Empfehlungsergebnisse zurückgeben aus. Wenn diese Option aktiviert ist, können Sie die Spalten aus Ihrem Artikeldatensatz in Ihrer Anfrage nach Empfehlungen oder personalisierter Rangfolge angeben. Amazon Personalize gibt diese Daten für jedes Element in der Empfehlungsantwort zurück.

Um Metadaten zu aktivieren, müssen Sie über einen Elementdatensatz mit einer Spalte von Metadaten verfügen.

- Nehmen Sie für `- Top picks for you` oder `-Recommended for you` Anwendungsfälle optional Änderungen an der `-Explorationskonfiguration` vor. Die Nutzung umfasst das Testen verschiedener Elementempfehlungen, um zu erfahren, wie Benutzer mit sehr wenigen Interaktionsdaten auf Elemente reagieren. Verwenden Sie die folgenden Felder, um die Untersuchung zu konfigurieren:
 - Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht) – Konfigurieren Sie, wie viel untersucht werden soll. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei

Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).

- Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Artikelexploration auf der Grundlage des Artikelalters. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Elementalter bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum beträgt 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

- Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Markieren von Amazon-Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).

7. Um Empfehlungsprogramme für jeden Ihrer Anwendungsfälle zu erstellen, wählen Sie Empfehlungsprogramme erstellen aus.

Sie können den Status jedes Empfehlers auf der Seite Empfehlungen überwachen. Wenn Ihr Empfehlungsstatus Aktiv lautet, können Sie ihn in Ihrer Anwendung verwenden, um Empfehlungen zu erhalten.

Erstellen von Empfehlungsgebern (AWS CLI)

Nachdem Sie eine Domain-Datensatzgruppe erstellt und Daten importiert haben, können Sie Empfehlungsprogramme für Ihre Domain-Anwendungsfälle erstellen. Ein Empfehler ist eine Domain-Datensatzgruppenressource, die Empfehlungen generiert.

Für - Top picks for your oder -Recommended for you Anwendungsfälle verwendet Amazon Personalize die Erkundung, wenn es Elemente empfiehlt. Weitere Informationen finden Sie unter [Konfigurieren der Erkundung](#).

Themen

- [Erstellen eines Empfehlers](#)

- [Konfigurieren von Spalten, die beim Training verwendet werden](#)
- [Konfigurieren der Erkundung](#)
- [Aktivieren von Metadaten in Empfehlungen](#)

Erstellen eines Empfehlens

Verwenden Sie den folgenden AWS CLI Code, um einen Empfehler für einen Domain-Anwendungsfall zu erstellen. Führen Sie diesen Code für jeden Ihrer Domain-Anwendungsfälle aus. `recipeArn` Geben Sie für den Amazon-Ressourcennamen (ARN) für Ihren Anwendungsfall an. Die verfügbaren Anwendungsfälle hängen von Ihrer Domain ab. Eine Liste der Anwendungsfälle und ihrer ARNs finden Sie unter [Auswahl eines Anwendungsfalls](#).

```
aws personalize create-recommender \  
--name recommender name \  
--dataset-group-arn dataset group ARN \  
--recipe-arn recipe ARN
```

Konfigurieren von Spalten, die beim Training verwendet werden

Sie können die Spalten ändern, die Amazon Personalize beim Training der Modelle berücksichtigt, die Ihren Empfehler unterstützen. Standardmäßig verwendet Amazon Personalize alle Spalten, die beim Training verwendet werden können. Spalten mit dem booleschen Datentyp und benutzerdefinierten nichtkategorialen Zeichenfolgenfeldern werden nicht verwendet. Sie können EVENT_TYPE-Spalten nicht ausschließen.

Sie können die beim Training verwendeten Spalten ändern, um zu steuern, welche Daten Amazon Personalize beim Erstellen Ihres Empfehlens verwendet. Sie können dies tun, um mit verschiedenen Kombinationen von Trainingsdaten zu experimentieren. Oder Sie schließen Spalten ohne aussagekräftige Daten aus. Möglicherweise haben Sie eine Spalte, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte vom Training ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

Um Spalten vom Training auszuschließen, geben Sie das `excludedDatasetColumns` Objekt in der `trainingDataConfig` als Teil der Empfehlungskonfiguration an. Geben Sie für jeden Schlüssel im Objekt den Datensatztyp an. Geben Sie für jeden Wert die Liste der auszuschließenden Spalten an.

```
aws personalize create-recommender \  
--name recommender name \  
--dataset-group-arn dataset group ARN \  
--excluded-dataset-columns excluded dataset columns
```



```
--recipe-arn recipe ARN \  
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":  
{ \"datasetType\" : [ \"column1Name\", \"column2Name\" ]}}}"
```

Konfigurieren der Erkundung

Für - Top picks for your oder -Recommended for you Anwendungsfälle verwendet Amazon Personalize die Erkundung, wenn es Elemente empfiehlt. Bei der Nutzung werden verschiedene Elementempfehlungen getestet, um zu erfahren, wie Benutzer mit sehr wenigen Interaktionsdaten auf Elemente reagieren. Sie können die Erkundung wie folgt konfigurieren:

- Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht) – Konfigurieren Sie, wie viel Sie untersuchen möchten. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).
- Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Elementexploration basierend auf dem Elementalter. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Elementalter bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum ist 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

Der folgende Code zeigt, wie Sie die Untersuchung konfigurieren, wenn Sie einen Empfehler für den Top picks for you Anwendungsfall erstellen. Das Beispiel verwendet die Standardwerte.

Wenn Sie über einen Elementdatensatz verfügen und die Option Metadaten einschließen möchten, wenn Sie Empfehlungen erhalten, aktualisieren Sie die , recommender-config um ein enableMetadataWithRecommendations Feld hinzuzufügen, und setzen Sie sie auf true.

```
aws personalize create-recommender \  

```

```
--name recommender name \  
--dataset-group-arn dataset group ARN \  
--recipe-arn arn:aws:personalize:::recipe/aws-vod-top-picks \  
--recommender-config "{\"itemExplorationConfig\":{\"explorationWeight\": \"0.3\",  
\"explorationItemAgeCutOff\": \"30\"}}"
```

Aktivieren von Metadaten in Empfehlungen

Wenn Sie über einen Artikel-Datensatz verfügen und die Option Metadaten einschließen möchten, wenn Sie Empfehlungen erhalten, legen Sie `true` in der `enableMetadataWithRecommendations` auf `festrecommender-config`.

```
aws personalize create-recommender \  
--name recommender name \  
--dataset-group-arn dataset group \  
--recipe-arn recipe ARN \  
--recommender-config "{\"enableMetadataWithRecommendations\": \"true\"}"
```

Erstellen von Empfehlern (AWS SDKs)

Nachdem Sie eine Domain-Datensatzgruppe erstellt und Daten importiert haben, können Sie Empfehlungsprogramme für Ihre Domain-Anwendungsfälle erstellen. Ein Empfehler ist eine Domain-Datensatzgruppenressource, die Empfehlungen generiert.

Für alle Anwendungsfälle können Sie die beim Training verwendeten Spalten konfigurieren. Weitere Informationen finden Sie unter [Konfigurieren von Spalten, die beim Training verwendet werden](#). Für `- Top picks for your` oder `-Recommended for you` Anwendungsfälle verwendet Amazon Personalize die Erkundung, wenn es Elemente empfiehlt. Weitere Informationen finden Sie unter [Konfigurieren der Erkundung](#).

Themen

- [Erstellen eines Empfehlens](#)
- [Konfigurieren der Erkundung](#)
- [Konfigurieren von Spalten, die beim Training verwendet werden](#)
- [Aktivieren von Metadaten](#)

Erstellen eines Empfehlers

Erstellen Sie einen Empfehler für einen Domain-Anwendungsfall mit dem folgenden Code. Geben Sie Ihrem Empfehler einen Namen und geben Sie den Amazon-Ressourcennamen (ARN) Ihrer Domain-Datensatzgruppe an. `recipeArn` Geben Sie für den ARN für Ihren Anwendungsfall an. Führen Sie diesen Code für jeden Ihrer Domain-Anwendungsfälle aus. Die verfügbaren Anwendungsfälle hängen von Ihrer Domain ab. Eine Liste der Anwendungsfälle, ihrer ARNs und ihrer Anforderungen finden Sie unter [Auswahl eines Anwendungsfalls](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'recipe ARN',
    datasetGroupArn = 'dataset group ARN'
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for Java 2.x

```
public static String createRecommender(PersonalizeClient personalizeClient,
    String name,
    String datasetGroupArn,
    String recipeArn) {

    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String recommenderStatus = "";

    try {
        CreateRecommenderRequest createRecommenderRequest =
        CreateRecommenderRequest.builder()
            .datasetGroupArn(datasetGroupArn)
            .name(name)
            .recipeArn(recipeArn)
```

```
        .build();

        CreateRecommenderResponse recommenderResponse = personalizeClient
            .createRecommender(createRecommenderRequest);
        String recommenderArn = recommenderResponse.recommenderArn();
        System.out.println("The recommender ARN is " + recommenderArn);

        DescribeRecommenderRequest describeRecommenderRequest =
DescribeRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                .status();
            System.out.println("Recommender status: " + recommenderStatus);

            if (recommenderStatus.equals("ACTIVE") ||
recommenderStatus.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return recommenderArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
```

```
import { CreateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",           /* required */
  recipeArn: "RECIPE_ARN",           /* required */
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Konfigurieren der Erkundung

Für -Top picks for your oder -Recommended for you Anwendungsfälle verwendet Amazon Personalize die Erkundung, wenn es Elemente empfiehlt. Bei der Nutzung werden verschiedene Elementempfehlungen getestet, um zu erfahren, wie Benutzer mit sehr wenigen Interaktionsdaten auf Elemente reagieren. Sie können die Erkundung wie folgt konfigurieren:

- **Hervorhebung der Untersuchung weniger relevanter Elemente (Explorationsgewicht)** – Konfigurieren Sie, wie viel Sie untersuchen möchten. Geben Sie einen Dezimalwert zwischen 0 und 1 an. Der Standardwert ist 0,3. Je näher der Wert bei 1 liegt, desto mehr Erkundung. Bei mehr Erkundung umfassen Empfehlungen mehr Elemente mit weniger Daten zu Elementinteraktionen oder Relevanz auf der Grundlage des vorherigen Verhaltens. Bei Null findet keine Untersuchung statt und die Empfehlungen basieren auf aktuellen Daten (Relevanz).

- Grenzwert für das Alter des Nutzungselements – Geben Sie das maximale Elementalter in Tagen seit der letzten Interaktion über alle Elemente im Datensatz Elementinteraktionen an. Dies definiert den Umfang der Elementexploration basierend auf dem Elementalter. Amazon Personalize bestimmt das Elementalter anhand seines Erstellungszeitstempels oder, falls Zeitstempeldaten für die Erstellung fehlen, anhand von Daten zu Elementinteraktionen. Weitere Informationen dazu, wie Amazon Personalize das Elementalter bestimmt, finden Sie unter [Zeitstempeldaten der Erstellung](#).

Um die Elemente zu erhöhen, die Amazon Personalize während der Erkundung berücksichtigt, geben Sie einen größeren Wert ein. Das Minimum ist 1 Tag und der Standardwert ist 30 Tage. Zu den Empfehlungen können Elemente gehören, die älter als der von Ihnen angegebene Grenzwert für das Elementalter sind. Dies liegt daran, dass diese Elemente für den Benutzer relevant sind und die Untersuchung sie nicht identifiziert hat.

Der folgende Code zeigt, wie Sie die Untersuchung konfigurieren, wenn Sie einen Empfehler erstellen. Das Beispiel verwendet die Standardwerte.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'arn:aws:personalize::recipe/aws-vod-top-picks',
    datasetGroupArn = 'dataset group ARN',
    recommenderConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
"explorationItemAgeCutOff": "30"}}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";
```

```
// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",          /* required */
  recipeArn: "RECIPE_ARN",          /* required */
  datasetGroupArn: "DATASET_GROUP_ARN", /* required */
  recommenderConfig: {
    itemExplorationConfig: {
      explorationWeight: "0.3",
      explorationItemAgeCutOff: "30"
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Konfigurieren von Spalten, die beim Training verwendet werden

Sie können die Spalten ändern, die Amazon Personalize beim Training der Modelle berücksichtigt, die Ihren Empfehler unterstützen. Standardmäßig verwendet Amazon Personalize alle Spalten, die beim Training verwendet werden können. Spalten mit dem booleschen Datentyp und benutzerdefinierten nichtkategorialen Zeichenfolgenfeldern werden nicht verwendet. Sie können EVENT_TYPE-Spalten nicht ausschließen.

Sie können die beim Training verwendeten Spalten ändern, um zu steuern, welche Daten Amazon Personalize beim Erstellen Ihres Empfehlers verwendet. Sie können dies tun, um mit verschiedenen Kombinationen von Trainingsdaten zu experimentieren. Oder Sie schließen Spalten

ohne aussagekräftige Daten aus. Möglicherweise haben Sie eine Spalte, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte vom Training ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

Um Spalten vom Training auszuschließen, geben Sie das `excludedDatasetColumns` Objekt in der `trainingDataConfig` als Teil der Empfehlungskonfiguration an. Geben Sie für jeden Schlüssel den Datensatztyp an. Geben Sie für jeden Wert die Liste der auszuschließenden Spalten an. Der folgende Code zeigt, wie Sie Spalten vom Training ausschließen, wenn Sie einen Empfehler erstellen.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'recipe name',
    datasetGroupArn = 'dataset group ARN',
    recommenderConfig = {
        "trainingDataConfig": {
            "excludedDatasetColumns": {
                "datasetType": ["COLUMN_A", "COLUMN_B"]
            }
        }
    }
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
```



```
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",          /* required */
  recipeArn: "RECIPE_ARN",           /* required */
  datasetGroupArn: "DATASET_GROUP_ARN", /* required */
  recommenderConfig: {
    trainingDataConfig: {
      excludedDatasetColumns: {
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Aktivieren von Metadaten

Wenn Sie über einen Artikel-Datensatz verfügen und die Option Metadaten einschließen möchten, wenn Sie Empfehlungen erhalten, legen Sie `true` in der `enableMetadataWithRecommendations` auf `festrecommender-config`.

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
  name = 'recommender name',
  recipeArn = 'recipe name',
  datasetGroupArn = 'dataset group ARN',
```

```
recommenderConfig = {"enableMetadataWithRecommendations": True}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

Bewertung eines Empfehlungsgebers

Sie können die Leistung Ihres Empfehlungsgebers anhand von Offline- und Online-Kennzahlen bewerten. Online-Metriken sind die empirischen Ergebnisse, die Sie bei den Interaktionen Ihrer Nutzer mit Empfehlungen in Echtzeit beobachten. Sie können beispielsweise die Klickrate Ihrer Benutzer aufzeichnen, während sie Ihren Katalog durchsuchen. Sie sind dafür verantwortlich, alle Online-Metriken zu generieren und aufzuzeichnen.

Offline-Metriken sind die Metriken, die Amazon Personalize generiert, wenn Sie einen Empfehlungsgeber erstellen. Sie können Offline-Metriken verwenden, um die Leistung des dem Empfehlungsgeber zugrunde liegenden Modells zu bewerten. Mithilfe von Offline-Metriken können Sie das Modell mit anderen Modellen vergleichen, die mit denselben Daten trainiert wurden. Für den Rest dieses Abschnitts bezieht sich der Begriff Metriken auf Offline-Metriken.

Um Leistungskennzahlen zu erhalten, teilt Amazon Personalize die eingegebenen Interaktionsdaten in einen Trainingsatz und einen Testatz auf. Das Trainingsset besteht zu 90% aus Ihren Benutzern und deren Interaktionsdaten. Das Testset besteht aus den verbleibenden 10% der Benutzer und ihren Interaktionsdaten.

Amazon Personalize erstellt dann den Empfehlungsgeber mithilfe des Trainingssets. Nach Abschluss der Schulung gibt Amazon Personalize dem neuen Empfehlungsgeber die ältesten 90% der Daten jedes Benutzers aus dem Testatz als Eingabe. Amazon Personalize berechnet dann Kennzahlen, indem es die Empfehlungen, die der Empfehlungsgeber generiert, mit den tatsächlichen Interaktionen in den neuesten 10% der Nutzerdaten aus dem Testset vergleicht.

Themen

- [Metriken abrufen](#)
- [Metrische Definitionen](#)
- [Beispiel](#)
- [Weitere Ressourcen](#)

Metriken abrufen

Nachdem Ihr Empfehlungsgeber aktiv ist, können Sie die Metriken für den Empfehlungsgeber in der Amazon Personalize-Konsole einsehen oder Metriken abrufen, indem Sie den Vorgang aufrufen.

[DescribeRecommender](#)

Themen

- [Metriken anzeigen \(Konsole\)](#)
- [Metriken abrufen \(\) AWS CLI](#)
- [Metriken abrufen \(AWSSDKs\)](#)

Metriken anzeigen (Konsole)

Um die Empfehlungsmetriken in der Konsole einzusehen, navigieren Sie zur Detailseite Ihres Empfehlungsgebers.

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Domain-Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Empfehlungen aus.
4. Wählen Sie aus der Liste der Empfehlungsgeber den Anbieter aus, um die zugehörigen Kennzahlen einzusehen.

Metriken abrufen () AWS CLI

Der folgende Code zeigt, wie Sie mit dem Metriken für einen Empfehlungsgeber abrufen können.
AWS CLI

```
aws personalize describe-recommender \  
--recommender-arn recommender arn
```

Im Folgenden finden Sie ein Beispiel für die Metriken, die von einem Empfehlungsgeber ausgegeben wurden, der für den Anwendungsfall „Top picks for you“ für die Domäne VIDEO_ON_DEMAND erstellt wurde.

```
{  
  "recommender": {
```

```

    "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
    "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
    "name": "name123",
    "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
    "modelMetrics": {
        "coverage": 0.27,
        "mean_reciprocal_rank_at_25": 0.0379,
        "normalized_discounted_cumulative_gain_at_5": 0.0405,
        "normalized_discounted_cumulative_gain_at_10": 0.0513,
        "normalized_discounted_cumulative_gain_at_25": 0.0828,
        "precision_at_5": 0.0136,
        "precision_at_10": 0.0102,
        "precision_at_25": 0.0091,
    }
    "recommenderConfig": {},
    "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
    "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
    "status": "ACTIVE",
}
}
}

```

Metriken abrufen (AWSSDKs)

Der folgende Code zeigt, wie mit dem SDK für Python (Boto3) Metriken für einen Empfehlungsgeber abgerufen werden.

```

import boto3

personalize = boto3.client('personalize')

response = personalize.describe_recommender(
    recommenderArn = 'recommender_arn'
)
print(response['recommender']['modelMetrics'])

```

Im Folgenden finden Sie ein Beispiel für die Metriken, die von einem Empfehlungsgeber ausgegeben wurden, der für den Anwendungsfall „Top picks for you“ für die Domäne VIDEO_ON_DEMAND erstellt wurde.

```
{
```

```
"recommender": {
  "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
  "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
  "name": "name123",
  "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
  "modelMetrics": {
    "coverage": 0.27,
    "mean_reciprocal_rank_at_25": 0.0379,
    "normalized_discounted_cumulative_gain_at_5": 0.0405,
    "normalized_discounted_cumulative_gain_at_10": 0.0513,
    "normalized_discounted_cumulative_gain_at_25": 0.0828,
    "precision_at_5": 0.0136,
    "precision_at_10": 0.0102,
    "precision_at_25": 0.0091,
  }
  "recommenderConfig": {},
  "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
  "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
  "status": "ACTIVE",
}
}
```

Metrische Definitionen

Die Metriken, die Amazon Personalize für Empfehlungsgeber generiert, werden im Folgenden anhand der folgenden Begriffe beschrieben:

- Eine relevante Empfehlung ist eine Empfehlung für ein Element, mit dem der Benutzer tatsächlich interagiert hat. Diese Elemente stammen aus den neuesten 10% der Interaktionsdaten jedes Benutzers aus dem Testset.
- Rang bezieht sich auf die Position eines empfohlenen Elements in der Liste der Empfehlungen. Position 1 (ganz oben in der Liste) ist möglicherweise am relevantesten für den Benutzer.

Für jede Metrik sind höhere Zahlen (näher an 1) besser. Weitere Informationen finden Sie in den Ressourcen, die unter aufgeführt sind [Weitere Ressourcen](#).

Abdeckung

Der Wert für die Reichweite gibt an, wie viele einzigartige Artikel Amazon Personalize möglicherweise an der Gesamtzahl der einzigartigen Artikel in den Datensätzen Interaktionen und Artikel empfiehlt. Ein höherer Deckungswert bedeutet, dass Amazon Personalize mehr Ihrer Artikel empfiehlt, anstatt dieselben wenigen Artikel wiederholt für verschiedene Benutzer zu verwenden. Anwendungsfälle, in denen Artikel untersucht werden, wie z. B. die Top-Picks für Sie (VIDEO_ON_DEMAND) und Für Sie empfohlen (E-COMMERCE), haben eine höhere Reichweite als solche, in denen dies nicht der Fall ist.

Mittlerer wechselseitiger Rang bei 25

Diese Metrik gibt Aufschluss über die Fähigkeit eines Modells, auf der obersten Position eine relevante Empfehlung zu generieren. Sie können ein Modell mit einem hohen mittleren reziproken Rang von 25 wählen, wenn Sie relevante Suchergebnisse für einen Benutzer generieren und nicht erwarten, dass der Benutzer ein Element auswählt, das weiter unten auf der Liste steht. Beispielsweise wählen Benutzer häufig das erste Kochrezept in den Suchergebnissen aus.

Amazon Personalize berechnet diese Metrik anhand des durchschnittlichen reziproken Ranking-Scores für Empfehlungsanfragen. Jeder reziproke Rangwert wird wie folgt berechnet: $1 / \text{the rank of the highest item interacted with by the user}$, wobei die mögliche Gesamtwertung 25 beträgt. Andere Elemente mit niedrigerem Rang, mit denen der Benutzer interagiert, werden ignoriert. Wenn der Benutzer das erste Element ausgewählt hat, ist die Punktzahl 1. Wenn sie keine Artikel auswählen, ist die Punktzahl 0.

Sie könnten beispielsweise drei verschiedenen Benutzern jeweils 25 Empfehlungen anzeigen:

- Wenn Benutzer 1 auf Rang 4 auf das Element und auf Rang 10 auf das Element klickt, beträgt sein reziproker Rangwert $1/4$.
- Wenn Benutzer 2 auf ein Element auf Rang 2, auf ein Element auf Rang 4 und auf ein Element auf Rang 12 klickt, beträgt sein reziproker Rangwert $1/2$.
- Wenn Benutzer 3 auf Rang 6 auf ein einzelnes Element klickt, beträgt sein reziproker Rangwert $1/6$.

Der mittlere reziproke Rang aller Empfehlungsanfragen (in diesem Fall 3) wird berechnet als.

$$(1/4 + 1/2 + 1/6) / 3 = .3056$$

normalisierter diskontierter kumulativer Gewinn (NDCG) bei K (5, 10 oder 25)

Diese Metrik gibt Aufschluss darüber, wie gut Ihr Modell Empfehlungen einstuft, wobei K ein Stichprobenumfang von 5, 10 oder 25 Empfehlungen ist. Diese Metrik ist nützlich, wenn Sie am

meisten an der Rangfolge der Empfehlungen interessiert sind, die nicht nur für den Artikel mit dem höchsten Rang gelten (Informationen dazu finden Sie unter `mean_reciprocal_rank_at_25`). Die Punktzahl für `NDCG_at_10` wäre beispielsweise nützlich, wenn Sie eine Anwendung haben, die bis zu 10 Filme gleichzeitig in einem Karussell anzeigt.

Amazon Personalize berechnet den NDCG, indem Empfehlungen auf der Grundlage ihrer Rangposition für jeden Benutzer im Testsatz Gewicht zugewiesen werden. Jede Empfehlung wird um einen Faktor abgezinst (mit geringerem Gewicht), der von ihrer Position abhängt. Die endgültige Metrik ist der Durchschnitt aller Benutzer im Testset. Der normalisierte, abgewertete kumulative Zuwachs bei K geht davon aus, dass Empfehlungen, die in einer Liste weiter unten stehen, weniger relevant sind als Empfehlungen, die weiter oben stehen.

Amazon Personalize verwendet einen Gewichtungsfaktor von $1/\log(1 + \text{position})$, wobei die Position ganz oben auf der Liste steht. 1

Präzision bei K

Diese Metrik gibt an, wie relevant die Empfehlungen Ihres Modells auf der Grundlage einer Stichprobengröße von K (5, 10 oder 25) Empfehlungen sind.

Amazon Personalize berechnet diese Metrik auf der Grundlage der Anzahl relevanter Empfehlungen aus den Top-K-Empfehlungen für jeden Benutzer im Testsatz, geteilt durch K, wobei K 5, 10 oder 25 ist. Die endgültige Metrik ist der Durchschnitt aller Benutzer im Testsatz.

Wenn Sie beispielsweise einem Benutzer 10 Elemente empfehlen und der Benutzer mit 3 von ihnen interagiert, beträgt die Genauigkeit bei K 3 korrekt vorhergesagte Elemente geteilt durch die insgesamt 10 empfohlenen Elemente: $3 / 10 = .30$.

Diese Kennzahl belohnt die genaue Empfehlung relevanter Artikel. Je näher die Punktzahl an eins liegt, desto genauer ist das Modell.

Beispiel

Im Folgenden finden Sie ein einfaches Beispiel für einen Empfehlungsgeber, der eine Liste mit Empfehlungen für einen bestimmten Benutzer erstellt. Die zweite und fünfte Empfehlung stimmt mit Datensätzen in den Testdaten für diesen Benutzer überein. Dies sind die relevanten Empfehlungen. Wenn K auf 5 gesetzt ist, werden die folgenden Metriken für den Benutzer generiert.

`reciprocal_rank`

Berechnung: $1/2$

Ergebnis: 0,5000

normalized_discounted_cumulative_gain_at_5

Berechnung: $(1/\log(1 + 2) + 1/\log(1 + 5)) / (1/\log(1 + 1) + 1/\log(1 + 2))$

Ergebnis: 0,6241

precision_at_5

Berechnung: $2/5$

Ergebnis: 0,4000

Weitere Ressourcen

Weitere Informationen zu den verschiedenen Arten von Metriken für Empfehlungssysteme finden Sie in den folgenden externen Ressourcen:

- [MRR gegen MAP gegen NDCG: Ranking-Asive-Bewertungsmetriken und wann sie verwendet werden sollten](#)
- [Vergünstigter kumulativer Gewinn: Die Ranking-Kennzahlen, über die Sie Bescheid wissen sollten](#)
- [Rückruf und Präzision auf höchstem Niveau für Empfehlungssysteme](#)
- [Bewertungsmetriken für Empfehlungssysteme](#)

Verwalten von Empfehlungsgebern

Sie müssen die Modelle, die Ihre Empfehlungsgeber unterstützen, nicht verwalten. Amazon Personalize trainiert sie automatisch alle 7 Tage neu. Dies ist ein vollständiges erneutes Training, bei dem völlig neue Modelle basierend auf der Gesamtheit der Daten in Ihren Datensätzen erstellt werden. Für Top-Auswahlen für Sie und Empfohlen für Sie aktualisiert Amazon Personalize die vorhandenen Modelle alle zwei Stunden, um neue Artikel für Empfehlungen zur Erkundung zu berücksichtigen. Weitere Informationen finden Sie unter [Automatische Updates](#).

Die Verwaltung von Empfehlungsgebern umfasst Folgendes:

- Anhalten und Starten von Empfehlungsgebern – Wenn Sie die Abrechnung für einen aktiven Empfehlungsgeber anhalten möchten, können Sie den Empfehlungsgeber anhalten und später neu starten. Weitere Informationen finden Sie unter [Einen Empfehlungsgeber beenden und starten](#).

- Aktualisieren der Konfiguration des Empfehlungsgebers – Sie können die Spalten aktualisieren, die der Empfehlungsgeber beim Training verwendet, und die Anforderungskapazität des Empfehlungsgebers aktualisieren. Weitere Informationen finden Sie unter [Einen Empfehlungsgeber aktualisieren](#).
- Löschen eines Empfehlungsgebers – Sie können Empfehlungsgeber mit der [-DeleteRecommender](#) Operation löschen. Oder Sie können einen Empfehler auf der Seite mit den Empfehlungsdetails in der Amazon-Personalize-Konsole löschen.

Themen

- [Einen Empfehlungsgeber aktualisieren](#)
- [Einen Empfehlungsgeber beenden und starten](#)

Einen Empfehlungsgeber aktualisieren

Nachdem Sie einen Empfehlungsgeber erstellt haben, können Sie die Konfiguration des Empfehlungsgebers aktualisieren:

- Sie können die Spalten aktualisieren, die der Empfehlungsgeber im Training verwendet. Wenn Sie die beim Training verwendeten Spalten ändern, beginnt Amazon Personalize automatisch mit einer vollständigen Neuschulung der Modelle, die Ihren Empfehlungsgeber unterstützen. Während das Update abgeschlossen ist, können Sie immer noch Empfehlungen vom Empfehlungsgeber erhalten. Der Empfehlungsgeber verwendet die vorherige Konfiguration, bis das Update abgeschlossen ist. Um den Status dieses Updates zu verfolgen, verwenden Sie den im [DescribeRecommender](#) Vorgang `latestRecommenderUpdate` zurückgegebenen Wert. Wenn Sie dieselben Spalten angeben, die Sie bei der Erstellung des Empfehlungsgebers angegeben haben, erfolgt keine Aktualisierung.
- Sie können die Mindestanzahl der Empfehlungsanfragen pro Sekunde aktualisieren. Dies gibt den grundlegenden Durchsatz von Empfehlungsanfragen an, der von Amazon Personalize bereitgestellt wird. Ein hoher Wert erhöht Ihre Rechnung. Wir empfehlen, mit 1 zu beginnen. Verfolgen Sie Ihre Nutzung mithilfe von CloudWatch Amazon-Metriken und erhöhen Sie sie bei Bedarf. Weitere Informationen finden Sie unter [Minimale Empfehlungsanforderungen pro Sekunde und Auto Scaling](#).
- Für die Anwendungsfälle „Top-Tipps für dich“ und „Für dich empfohlen“ kannst du die Erkundungskonfiguration aktualisieren, indem du den Schwerpunkt auf die Erkundung relevanter Gegenstände und das Mindestalter für Erkundungsgegenstände änderst. Informationen zur Erkundung finden Sie im Abschnitt für Ihren Anwendungsfall unter [Auswahl eines Anwendungsfalls](#).

Sie können Empfehlungen mit der Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs aktualisieren.

Themen

- [Einen Empfehlungsgeber aktualisieren \(Amazon Personalize-Konsole\)](#)
- [Einen Empfehlungsgeber aktualisieren \(\) AWS CLI](#)
- [Aktualisierung eines Empfehlungsschreibers \(AWS SDKs\)](#)

Einen Empfehlungsgeber aktualisieren (Amazon Personalize-Konsole)

Nachdem Sie einen Empfehlungsgeber erstellt haben, können Sie ihn aktualisieren. Sie können die Spalten, die der Empfehler in Schulungen verwendet, und die Mindestanzahl der Empfehlungsanfragen des Empfehlungsgebers pro Sekunde aktualisieren. Sie können die Explorationskonfiguration aktualisieren, um die für Sie am besten geeigneten Anwendungsfälle und die für Sie empfohlenen Anwendungsfälle zu finden. Gehen Sie wie folgt vor, um einen Empfehlungsgeber mit der Konsole zu aktualisieren.

So aktualisieren Sie die Konfiguration eines Empfehlungsgebers (Konsole)

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Domain-Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Empfehlungen aus.
4. Wählen Sie auf der Seite Empfehlungen den Empfehlungsgeber aus, den Sie aktualisieren möchten.
5. Wählen Sie in der Empfehlungskonfiguration die Option Bearbeiten.
6. Ändern Sie die Konfiguration des Empfehlungsgebers und wählen Sie Aktualisieren. Informationen zu den verschiedenen Konfigurationsoptionen finden Sie unter [Erstellen von Empfehlern \(Konsole\)](#).

Einen Empfehlungsgeber aktualisieren () AWS CLI

Um den Empfehlungsgeber mit dem zu aktualisieren AWS CLI, verwenden Sie den `update-recommender` Befehl. Geben Sie den Amazon-Ressourcennamen (ARN) für die Empfehlung und die aktualisierte Konfiguration an. Der folgende Code zeigt, wie die Spalten aktualisiert werden, die ein Empfehlungsgeber für Schulungen verwendet.

```
aws personalize update-recommender \  
--dataset-group-arn dataset group ARN \  
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":  
  { \">datasetType\": [ \">column1Name\", \">column2Name\"]}}}"
```

Wenn Sie die im Training verwendeten Spalten ändern, beginnt Amazon Personalize automatisch mit einer vollständigen Neuschulung der Modelle, die Ihren Empfehlungsgeber unterstützen. Während das Update abgeschlossen ist, können Sie immer noch Empfehlungen vom Empfehlungsgeber erhalten. Der Empfehlungsgeber verwendet die vorherige Konfiguration, bis das Update abgeschlossen ist. Um den Status dieses Updates zu verfolgen, verwenden Sie den im [DescribeRecommender](#) Vorgang `latestRecommenderUpdate` zurückgegebenen Wert.

Weitere Informationen zu den verschiedenen Konfigurationen, die Sie ändern können, finden Sie unter [RecommenderConfig](#).

Aktualisierung eines Empfehlungsschreibers (AWSSDKs)

Um den Empfehlungsgeber mit dem zu aktualisieren AWS, verwenden Sie die [UpdateRecommender](#) Operation. Geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber an und geben Sie die neue Konfiguration an. Der folgende Code zeigt, wie die Spalten aktualisiert werden, die ein Empfehlungsgeber für Schulungen verwendet.

SDK for Python (Boto3)

```
import boto3  
  
personalize = boto3.client('personalize')  
  
update_recommender_response = personalize.update_recommender(  
    recommenderArn = 'dataset group ARN',  
    recommenderConfig = {  
        "trainingDataConfig": {  
            "excludedDatasetColumns": {  
                "datasetType": ["COLUMN_A", "COLUMN_B"]  
            }  
        }  
    }  
)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { UpdateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request's parameters
export const updateRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN", /* required */
  recommenderConfig: {
    trainingDataConfig: {
      excludedDatasetColumns: {
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
UpdateRecommenderCommand(updateRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Wenn Sie die im Training verwendeten Spalten in `excludedDatasetColumns` der `updateRecommenderConfig` ändern, beginnt Amazon Personalize automatisch mit einer vollständigen Neuschulung der Modelle, die Ihren Empfehlungsgeber unterstützen. Während das Update abgeschlossen ist, können Sie immer noch Empfehlungen vom Empfehlungsgeber erhalten. Der Empfehlungsgeber verwendet die vorherige Konfiguration, bis das Update abgeschlossen ist. Um

den Status dieses Updates zu verfolgen, verwenden Sie den im [DescribeRecommender](#) Vorgang `latestRecommenderUpdate` zurückgegebenen Wert.

Weitere Informationen zu den verschiedenen Konfigurationen, die Sie ändern können, finden Sie unter [RecommenderConfig](#).

Einen Empfehlungsgeber beenden und starten

Nachdem Ihr Empfehlungsgeber aktiv ist, können Sie einen Empfehlungsgeber beenden und ihn später starten. Auf diese Weise können Sie die Abrechnung von Empfehlungsgebern unterbrechen und nur dann bezahlen, wenn Sie sie nutzen. Beispielsweise müssen Sie möglicherweise nur an bestimmten Wochentagen Empfehlungen einholen. Sie können den Empfehlungsgeber an den Tagen beenden, an denen Sie ihn nicht benötigen, und ihn dann an den Tagen starten, an denen Sie ihn benötigen.

Nachdem Sie einen Empfehlungsgeber beendet haben, können Sie ihn nicht mehr verwenden, um Empfehlungen zu erhalten. Wenn Sie einen Empfehlungsgeber beenden, werden die Abrechnung und Umschulung des Empfehlungsgebers unterbrochen. Wenn Sie einen Empfehlungsgeber beenden, wird dieser jedoch nicht gelöscht. Sie können es jederzeit neu starten und weiterhin Empfehlungen erhalten. Wenn Sie einen Empfehlungsgeber starten, wird kein neuer Empfehlungsgeber mit Ihren Daten erstellt. Vielmehr nimmt es die Abrechnung der Empfehlungsgeber und die Umschulung alle 7 Tage wieder auf.

Sie können einen Empfehlungsgeber mit der Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI), den AWS SDKs oder den API-Vorgängen [StartRecommender](#) und [StopRecommender](#) starten.

Der Empfehlungsgeber gibt an

Wenn Sie einen Empfehlungsgeber beenden, wechselt der Empfehlungsstatus in der folgenden Reihenfolge von AKTIV zu INAKTIV:

AKTIV > AUSSTEHEND BEENDEN > LAUFENDER STOPP > INAKTIV

Wenn Sie einen Empfehlungsgeber starten, wechselt der Empfehlungsstatus in der folgenden Reihenfolge von INAKTIV zu AKTIV:

INAKTIV > START STEHT AUS > START LÄUFT > AKTIV

Themen

- [Einen Empfehlungsgeber beenden und starten \(Konsole\)](#)
- [Einen Empfehlungsgeber beenden und neu starten \(\) AWS CLI](#)
- [Stoppen und Neustarten eines Empfehlungssystems \(AWSSDKs\)](#)

Einen Empfehlungsgeber beenden und starten (Konsole)

Sie können Amazon Personalize verwenden, um einen Empfehlungsgeber zu beenden und neu zu starten.

Themen

- [Einen Empfehlungsgeber stoppen \(Konsole\)](#)
- [Einen Empfehlungsgeber starten \(Konsole\)](#)

Einen Empfehlungsgeber stoppen (Konsole)

Sie können die Amazon Personalize-Konsole verwenden, um einen aktiven Empfehlungsgeber wie folgt zu stoppen.

Um einen Empfehlungsgeber zu stoppen

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Domain-Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Empfehlungen aus.
4. Wählen Sie auf der Seite Empfehlungen den Empfehlungsgeber aus, den Sie beenden möchten.
5. Wählen Sie oben rechts die Option Empfehlung beenden und bestätigen Sie dies im angezeigten Fenster.

Wenn der Empfehlungsstatus inaktiv ist, hat Ihr Empfehlungsgeber aufgehört. Dadurch werden jegliche Fakturierung und Umschulung von Empfehlungsgebern eingestellt. Sie können den Empfehlungsgeber erst verwenden, wenn Sie ihn starten.

Einen Empfehlungsgeber starten (Konsole)

Sie können die Amazon Personalize-Konsole verwenden, um einen inaktiven Empfehlungsgeber wie folgt zu starten.

Um einen Empfehlungsgeber zu starten

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Domain-Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Empfehlungen aus.
4. Wählen Sie auf der Seite Empfehlungen den Empfehlungsgeber aus, den Sie starten möchten.
5. Wählen Sie oben rechts die Option Empfehlung starten und bestätigen Sie in dem daraufhin angezeigten Fenster, dass Sie ihn starten möchten.

Wenn der Empfehlungsstatus aktiv ist, können Sie weiterhin Empfehlungen von ihm erhalten. Die Abrechnung der Empfehlungsgeber und die automatische Umschulung werden ebenfalls wieder aufgenommen.

Einen Empfehlungsgeber beenden und neu starten () AWS CLI

Um einen aktiven Empfehlungsgeber mit dem zu beendenAWS CLI, verwenden Sie den `stop-recommender` Befehl und geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber wie folgt an:

```
aws personalize stop-recommender --recommender-arn "recommender arn"
```

Um einen inaktiven Empfehlungsgeber mit dem zu startenAWS CLI, verwenden Sie den `start-recommender` Befehl und geben Sie den ARN für den gestoppten Empfehlungsgeber wie folgt an:

```
aws personalize start-recommender --recommender-arn "recommender arn"
```

Weitere Informationen zu den API-Operationen finden Sie unter [StartRecommender](#) und [StopRecommender](#).

Stoppen und Neustarten eines Empfehlungssystems (AWSSDKs)

Sie können die AWS SDKs verwenden, um einen aktiven Empfehlungsgeber zu starten oder einen inaktiven Empfehlungsgeber zu beenden. Weitere Informationen zu den API-Operationen finden Sie unter [StartRecommender](#) und [StopRecommender](#).

Themen

- [Einen Empfehlungsgeber stoppen \(AWSSDKs\)](#)

- [Einen Empfehlungsgeber starten \(AWSSDKs\)](#)

Einen Empfehlungsgeber stoppen (AWSSDKs)

Der folgende Code zeigt, wie Sie einen aktiven Empfehlungsgeber mit den AWS SDKs beenden können. Durch das Beenden werden die Abrechnung aller Empfehlungsgeber und die automatische Umschulung gestoppt. Sie können den Empfehlungsgeber erst verwenden, wenn Sie ihn neu starten.

SDK for Python (Boto3)

Um einen aktiven Empfehlungsgeber mit dem SDK für Python (Boto3) zu stoppen, verwenden Sie die `stop_recommender` Methode und geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber wie folgt an.

```
import boto3
personalize = boto3.client('personalize')

stop_recommender_response = personalize.stop_recommender(
    recommenderArn = "recommenderARN"
)
print(stop_recommender_response)
```

SDK for Java 2.x

Um einen aktiven Empfehlungsgeber mit dem SDK für Java 2.x zu stoppen, verwenden Sie die `stopRecommender` Methode und geben Sie den ARN für den Empfehlungsgeber wie folgt an.

```
public static void stopRecommender(PersonalizeClient personalizeClient,
                                   String datasetGroupArn) {

    try {

        StopRecommenderRequest stopRecommenderRequest =
        StopRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();
        personalizeClient.stopRecommender(stopRecommenderRequest);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```



```
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { StopRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request params
export const stopRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN" /* required */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new StopRecommenderCommand(stopRecommenderParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Einen Empfehlungsgeber starten (AWSSDKs)

Der folgende Code zeigt, wie Sie einen inaktiven Empfehlungsgeber mit den AWS SDKs starten. Wenn der Empfehlungsstatus aktiv ist, können Sie weiterhin Empfehlungen von ihm erhalten. Gleichzeitig werden die Abrechnung der Empfehlungsgeber und die automatische Umschulung wieder aufgenommen.

SDK for Python (Boto3)

Um einen inaktiven Empfehlungsgeber mit dem SDK für Python (Boto3) zu starten, verwenden Sie die `start_recommender` Methode und geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber wie folgt an.

```
import boto3
personalize = boto3.client('personalize')

start_recommender_response = personalize.start_recommender(
    recommenderArn = "recommenderARN"
)
print(start_recommender_response)
```

SDK for Java 2.x

Um einen inaktiven Empfehlungsgeber mit dem SDK für Java 2.x zu starten, verwenden Sie die `startRecommender` Methode und geben Sie den ARN für den Empfehlungsgeber wie folgt an.

```
public static void startRecommender(PersonalizeClient personalizeClient,
                                   String datasetGroupArn) {

    try {

        StartRecommenderRequest startRecommenderRequest =
        StartRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();
        personalizeClient.startRecommender(startRecommenderRequest);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { StartRecommenderCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
```

```
    region: "REGION"
  });

  // set the request params
  export const startRecommenderParam = {
    recommenderArn: "RECOMMENDER_ARN" /* required */
  };

  export const run = async () => {
    try {
      const response = await personalizeClient.send(
        new StartRecommenderCommand(startRecommenderParam)
      );
      console.log("Success", response);
      return response; // For unit tests.
    } catch (err) {
      console.log("Error", err);
    }
  };
  run();
```

Benutzerdefinierte Ressourcen erstellen

Nachdem Sie Ihre Daten importiert haben, können Sie die benutzerdefinierten Ressourcen erstellen, die Sie zum Abrufen von Empfehlungen verwenden. Gehen Sie wie folgt vor, um die benutzerdefinierten Ressourcen zu erstellen, die Empfehlungen generieren:

1. Eine Lösung erstellen und konfigurieren: Passen Sie die Lösungsparameter und rezeptspezifischen Hyperparameter an, sodass das Modell Ihren spezifischen Geschäftsanforderungen entspricht.
2. Lösungsversion erstellen: Erstellen Sie eine Lösungsversion (trainieren Sie ein Modell). Die Lösungsversion generiert Amazon Personalize oder Benutzersegmente.
3. Stellen Sie die Lösungsversion mit einer Kampagne bereit (nur für Empfehlungen in Echtzeit): Erstellen Sie eine Kampagne, um Ihre Lösungsversion bereitzustellen. Sie verwenden die Kampagne, wenn Sie Empfehlungen in Echtzeit anfordern. Wenn Sie mehrere Empfehlungen erhalten, müssen Sie keine Kampagne erstellen.

Themen

- [Erstellen einer Lösung und einer Lösungsversion](#)

- [Eine Kampagne erstellen](#)

Erstellen einer Lösung und einer Lösungsversion

Nachdem Sie mit dem Importieren von Daten fertig sind, können Sie eine Lösung erstellen. Eine Lösung bezieht sich auf die Kombination aus einem Amazon-Personalize-Rezept, benutzerdefinierten Parametern und einer oder mehreren Lösungsversionen (trainierte Modelle).

Gehen Sie wie folgt vor, um eine Lösung in Amazon Personalize zu erstellen:

1. Lösung erstellen – Passen Sie Lösungsparameter und rezeptspezifische Hyperparameter an, damit das Modell Ihren spezifischen Geschäftsanforderungen entspricht. Siehe [Erstellen und Konfigurieren einer Lösung](#). Eine Liste der verfügbaren Rezepte finden Sie unter [Ein Rezept wählen](#).
2. Erstellen einer Lösungsversion (Modell trainieren) – Trainieren Sie das Machine-Learning-Modell, das Amazon Personalize verwendet, um Empfehlungen für Ihre Kunden zu generieren. Siehe [Eine Lösungsversion erstellen](#).
3. Bewerten der Lösungsversion – Verwenden Sie die Metriken, die Amazon Personalize aus der neuen Lösungsversion generiert, um die Leistung des Modells zu bewerten. Siehe [Auswerten einer Lösungsversion mit Metriken](#).

Themen

- [Erstellen und Konfigurieren einer Lösung](#)
- [Eine Lösungsversion erstellen](#)
- [Auswerten einer Lösungsversion mit Metriken](#)

Erstellen und Konfigurieren einer Lösung

Nachdem Sie mit dem Importieren von Daten fertig sind, können Sie eine Lösung erstellen. Eine Lösung bezieht sich auf die Kombination aus einem Amazon-Personalize-Rezept, benutzerdefinierten Parametern und einer oder mehreren Lösungsversionen (trainierte Modelle).

Wenn Sie eine Lösung erstellen, können Sie sie so konfigurieren, dass sie Ihren spezifischen Geschäftsanforderungen entspricht:

- Wenn Sie eine Lösung konfigurieren, geben Sie das Rezept an, das Ihrem Anwendungsfall entspricht. Weitere Informationen zu verfügbaren Rezepten finden Sie unter [Ein Rezept wählen](#).

- Wenn das von Ihnen gewählte Rezept Elementempfehlungen oder Benutzersegmente generiert, können Sie die Spalten ändern, die Amazon Personalize beim Training eines Modells berücksichtigt (Erstellen einer Lösungsversion). Weitere Informationen finden Sie unter [Konfiguration der beim Training verwendeten Spalten](#).
- Sie können Hyperparameter konfigurieren, um das Modell basierend auf Ihren Rezept- und Geschäftsanforderungen zu optimieren. Unterschiedliche Rezepte verwenden unterschiedliche Hyperparameter. Informationen zur Konfiguration von Hyperparametern finden Sie unter [Hyperparameter und HPO](#). Die verfügbaren Hyperparameter für Ihr Rezept finden Sie auf der Seite für Ihr Rezept in [Ein Rezept wählen](#).
- Wenn Sie entweder das - [User-Personalization-Rezept](#) oder das - [Personalisiertes Ranking-Rezept](#) verwenden, können Sie Ihre Lösung zusätzlich zur Relevanz für ein Ziel optimieren. Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).
- Wenn Sie über Ereignistyp- und Ereigniswertdaten verfügen, können Sie damit die Elementinteraktionsdaten auswählen, die Amazon Personalize während des Trainings berücksichtigt. Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

Wenn Sie bereits über eine Lösung verfügen, können Sie die Amazon-Personalize-Konsole verwenden, um die Lösung zu klonen. Wenn Sie eine Lösung klonen, können Sie die Konfiguration der vorhandenen Lösung als Ausgangspunkt verwenden, z. B. das Rezept und die Hyperparameter, und bei Bedarf Änderungen vornehmen. Weitere Informationen finden Sie unter [Klonen einer Lösung \(Konsole\)](#). Nachdem Sie eine Lösung erstellt haben, können Sie die Konfigurationsdetails der Lösung auf der Detailseite der Lösung der Amazon-Personalize-Konsole oder mit dem [-DescribeSolution](#)Vorgang anzeigen.

Sie können eine Lösung mit der Konsole, der AWS Command Line Interface (AWS CLI) oder dem AWS SDK erstellen und konfigurieren.

Themen

- [Erstellen einer Lösung \(Konsole\)](#)
- [Erstellen einer Lösung \(AWS CLI\)](#)
- [Erstellen einer Lösung \(AWS SDKs\)](#)
- [Konfiguration der beim Training verwendeten Spalten](#)
- [Optimierung einer Lösung für ein zusätzliches Ziel](#)
- [Hyperparameter und HPO](#)

- [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#)
- [Klonen einer Lösung \(Konsole\)](#)

Erstellen einer Lösung (Konsole)

Um eine Lösung in der Konsole zu erstellen, wählen Sie die Datensatzgruppe aus, die den Datensatz enthält, den Sie verwenden möchten, und geben Sie dann einen Lösungsnamen, ein Rezept und optionale rezeptspezifische Hyperparameter an.

Wenn Ihr Rezept Elementempfehlungen oder Benutzersegmente generiert, können Sie die Spalten so ändern, dass sie mit verschiedenen Kombinationen von Trainingsdaten experimentieren, oder Spalten ohne aussagekräftige Daten aus dem Training ausschließen. Beispielsweise enthält möglicherweise eine Spalte, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte vom Training ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

So konfigurieren Sie eine Lösung (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Auf der Übersichtsseite für Schritt 3:
 - Wenn Sie eine Domain-Datensatzgruppe erstellt haben, wählen Sie Benutzerdefinierte Ressourcen verwenden und dann Lösungen erstellen aus.
 - Wenn Sie eine benutzerdefinierte Datensatzgruppe erstellt haben, wählen Sie Lösungen erstellen aus.
4. Geben Sie für Solution name (Name der Lösung) einen Namen für die Lösung an.
5. Wählen Sie für Lösungstyp den Lösungstyp aus, den Sie erstellen möchten. Der von Ihnen gewählte Typ bestimmt, welche Rezepte verfügbar sind.
 - Wählen Sie Elementempfehlung, um Elementempfehlungen für Ihre Benutzer zu erhalten. Zum Beispiel personalisierte Filmempfehlungen.
 - Wählen Sie Aktionsempfehlung, um Aktionsempfehlungen für Ihre Benutzer zu erhalten. Generieren Sie beispielsweise die nächstbeste Aktion, z. B. das Herunterladen Ihrer App, für einen Benutzer.

- Wählen Sie Benutzersegmentierung aus, um Benutzersegmente (Benutzergruppen) basierend auf Ihren Elementdaten abzurufen.
6. Wählen Sie für Rezept ein Rezept aus (siehe [Ein Rezept wählen](#)).
 7. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Markieren von Amazon-Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).
 8. Wählen Sie Weiter aus.
 9. Wenn Ihr Rezept auf der Seite Erweiterte Konfiguration Elementempfehlungen oder Benutzersegmente generiert, verwenden Sie optional Spalten für das Training, um die Spalten auszuwählen, die Amazon Personalize bei den Versionen der Schulungslösung berücksichtigt.

Standardmäßig verwendet Amazon Personalize alle Spalten, die im Training verwendet werden können. Es werden nur Datensätze aufgeführt, die von Ihrem ausgewählten Rezept verwendet werden. Spalten mit dem booleschen Datentyp und nichtkategorialen Zeichenfolgenfeldern werden nicht verwendet. Sie können EVENT_TYPE-Spalten nicht ausschließen.

Weitere Informationen finden Sie unter [Konfiguration der beim Training verwendeten Spalten](#).

10. Konfigurieren Sie in der Hyperparameter-Konfiguration alle Hyperparameter-Optionen basierend auf Ihrem Rezept und Ihren Geschäftsanforderungen. Unterschiedliche Rezepte verwenden unterschiedliche Hyperparameter. Die verfügbaren Hyperparameter finden Sie in den einzelnen Rezepten unter [Ein Rezept wählen](#).
11. Wenn Ihr Datensatz zu Elementinteraktionen in der Lösungskonfiguration EVENT_TYPE oder sowohl EVENT_TYPE als auch EVENT_VALUE enthält, verwenden Sie optional die Schwellenwertfelder Ereignistyp und Ereigniswert, um die Elementinteraktionsdaten auszuwählen, die Amazon Personalize beim Training des Modells verwendet.

Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

12. Wenn Sie entweder das - [User-Personalization-Rezept](#) oder das - [Personalisiertes Ranking-Rezept](#) verwenden, geben Sie optional ein -Ziel an und wählen Sie eine -Zielsensibilität, um Ihre Lösung zusätzlich zur Relevanz für ein Ziel zu optimieren. Die Zielsensibilität konfiguriert, wie Amazon Personalize anhand von Interaktionsdaten ein ausgewogenes Verhältnis zwischen Empfehlungselementen auf der Grundlage Ihres Ziels und Ihrer Relevanz ausgleicht. Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).

13. Wählen Sie Weiter und überprüfen Sie die Lösungsdetails. Sie können die Konfiguration Ihrer Lösung nicht mehr ändern, nachdem Sie sie erstellt haben.
14. Wählen Sie Create solution (Lösung erstellen) aus. Amazon Personalize beginnt mit der Erstellung Ihrer ersten Lösungsversion und die Seite Lösungen und Rezepte wird angezeigt. Um den Status zu überwachen, wählen Sie Ihre Lösung aus und zeigen Sie den Status im Abschnitt Lösungsversionen an.

Wenn es aktiv ist, können Sie es in einer Kampagne bereitstellen und dann Empfehlungen in Echtzeit erhalten. Je nach Rezept können Sie Batch-Empfehlungen erhalten, ohne eine Kampagne zu erstellen.

Weitere Informationen zu Kampagnen finden Sie unter [Eine Kampagne erstellen](#). Weitere Informationen zu Batch-Empfehlungen finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

Erstellen einer Lösung (AWS CLI)

Verwenden Sie die folgende `create-solution` Operation AWS CLI, um eine Lösung mit der zu erstellen. Geben Sie `solution name`, `dataset group arn` und `recipe arn`.

```
aws personalize create-solution \  
  --name solution name \  
  --dataset-group-arn dataset group arn \  
  --recipe-arn recipe arn
```

Der Amazon-Ressourcenname (ARN) der Lösung wird angezeigt, zum Beispiel:

```
{  
  "solutionArn": "arn:aws:personalize:<region>:solution/<solution name>"  
}
```

Notieren Sie sich den Lösungs-ARN für die zukünftige Verwendung und fahren Sie mit fort [Eine Lösungsversion erstellen \(AWS CLI\)](#).

Je nach Rezept können Sie den obigen Code ändern, um die Rezepteigenschaften und Hyperparameter zu optimieren (siehe [Hyperparameter und HPO](#)), Spalten auswählen, die im Training verwendet werden (siehe [Konfiguration der beim Training verwendeten Spalten \(\) AWS CLI](#)), oder die Interaktionsdaten filtern, die für das Training verwendet werden (siehe [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#)).

Wenn Sie entweder das - [User-Personalization-Rezept](#) oder das - [Personalisiertes Ranking-Rezept](#) Rezept verwenden, können Sie Ihre Lösung zusätzlich zur Relevanz für ein Ziel optimieren. Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).

Erstellen einer Lösung (AWS SDKs)

Der folgende Code zeigt, wie Sie eine Amazon-Personalize-Lösung erstellen. Geben Sie der Lösung einen Namen und geben Sie den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe sowie den ARN des zu verwendenden Rezepts an. Weitere Informationen zu Rezepten finden Sie unter [Ein Rezept wählen](#).

Abhängig von Ihrem Rezept können Sie den folgenden Code ändern, um die Eigenschaften und Hyperparameter von Rezepten zu optimieren (siehe [Hyperparameter und HPO](#)), konfigurieren, welche Spalten für das Training verwendet werden (siehe [Konfiguration der beim Training verwendeten Spalten \(AWSSDKs\)](#)), oder die Interaktionsdaten filtern, die für das Training verwendet werden (siehe [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#)).

Wenn Sie entweder das - [User-Personalization-Rezept](#) oder das - [Personalisiertes Ranking-Rezept](#) Rezept verwenden, können Sie Ihre Lösung zusätzlich zur Relevanz für ein Ziel optimieren. Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

print('Creating solution')
create_solution_response = personalize.create_solution(
    name='solution name',
    recipeArn= 'recipe arn',
    datasetGroupArn = 'dataset group arn'
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for Java 2.x

```
public static String createPersonalizeSolution(PersonalizeClient
personalizeClient,
        String datasetGroupArn,
```

```

        String solutionName,
        String recipeArn) {

    try {
        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .build();

        CreateSolutionResponse solutionResponse =
personalizeClient.createSolution(solutionRequest);
        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateSolutionCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution parameters.
export const createSolutionParam = {
    datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
    recipeArn: 'RECIPE_ARN', /* required */
    name: 'NAME' /* required */
}

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
CreateSolutionCommand(createSolutionParam));
        console.log("Success", response);
        return response; // For unit tests.
    }
}

```

```
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

Notieren Sie sich den Lösungs-ARN für die zukünftige Verwendung und fahren Sie mit fort [Eine Lösungsversion \(SDKs\) erstellen AWS](#).

Konfiguration der beim Training verwendeten Spalten

Wenn Ihr Rezept Artikelempfehlungen oder Benutzersegmente generiert, können Sie die Spalten ändern, die Amazon Personalize bei der Erstellung einer Lösungsversion berücksichtigt (Training eines Modells). Standardmäßig verwendet Amazon Personalize alle Spalten, die beim Training verwendet werden können. Spalten mit dem booleschen Datentyp und benutzerdefinierte Zeichenkettenfelder, die nicht kategorisch oder textuell sind, werden beim Training nicht verwendet. Sie können EVENT_TYPE-Spalten nicht ausschließen.

Sie können die beim Training verwendeten Spalten ändern, um zu steuern, welche Daten Amazon Personalize beim Trainieren eines Modells verwendet (Erstellen einer Lösungsversion). Sie können dies tun, um mit verschiedenen Kombinationen von Trainingsdaten zu experimentieren. Oder Sie könnten Spalten ohne aussagekräftige Daten ausschließen. Sie könnten beispielsweise eine Spalte haben, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte von der Schulung ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

Wenn Sie bereits eine Lösung erstellt haben und die beim Training verwendeten Spalten ändern möchten, können Sie die Lösung klonen. Wenn Sie eine Lösung klonen, können Sie die Konfiguration der vorhandenen Lösung als Ausgangspunkt verwenden, z. B. das Rezept und die Hyperparameter, und bei Bedarf Änderungen vornehmen. Weitere Informationen finden Sie unter [Klonen einer Lösung \(Konsole\)](#).

Sie können die Spalten konfigurieren, die Amazon Personalize beim Training mit der Amazon Personalize Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS dem SDK verwendet. Informationen zur Auswahl von Spalten mit der Amazon Personalize Personalize-Konsole finden Sie in [Erstellen einer Lösung \(Konsole\)](#) den erweiterten Konfigurationsschritten unter. Nachdem Sie eine Lösung erstellt haben, können Sie die Spalten, die die Lösung verwendet, auf der Detailseite der Lösung in der Amazon Personalize Personalize-Konsole oder zusammen mit dem [DescribeSolution](#) Vorgang anzeigen.

Themen

- [Konfiguration der beim Training verwendeten Spalten \(\) AWS CLI](#)
- [Konfiguration der beim Training verwendeten Spalten \(AWSSDKs\)](#)

Konfiguration der beim Training verwendeten Spalten () AWS CLI

Um Spalten vom Training auszuschließen, geben Sie das `excludedDatasetColumns` Objekt in der `trainingDataConfig` als Teil der Lösungskonfiguration an. Geben Sie für jeden Schlüssel den Datensatztyp an. Geben Sie für jeden Wert die Liste der auszuschließenden Spalten an. Der folgende Code zeigt, wie Sie Spalten vom Training ausschließen, wenn Sie eine Lösung mit dem erstellenAWS CLI.

```
aws personalize create-solution \  
--name solution name \  
--dataset-group-arn dataset group ARN \  
--recipe-arn recipe ARN \  
--solution-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":  
  { \"datasetType\" : [ \"column1Name\", \"column2Name\" ] } } }"
```

Konfiguration der beim Training verwendeten Spalten (AWSSDKs)

Um Spalten vom Training auszuschließen, geben Sie das `excludedDatasetColumns` Objekt in der `trainingDataConfig` als Teil der Lösungskonfiguration an. Geben Sie für jeden Schlüssel den Datensatztyp an. Geben Sie für jeden Wert die Liste der auszuschließenden Spalten an. Der folgende Code zeigt, wie Sie Spalten vom Training ausschließen, wenn Sie eine Lösung mit dem SDK for Python (Boto3) erstellen.

```
import boto3  
  
personalize = boto3.client('personalize')  
  
create_solution_response = personalize.create_solution(  
    name = 'solution name',  
    recipeArn = 'recipe ARN',  
    datasetGroupArn = 'dataset group ARN',  
    solutionConfig = {  
        "trainingDataConfig": {  
            "excludedDatasetColumns": {  
                "datasetType": ["COLUMN_A", "COLUMN_B"]  
            }  
        }  
    }  
)
```

```
    }  
  }  
}  
)  
solution_arn = create_solution_response['solutionArn']  
print('solution_arn: ', solution_arn)
```

Optimierung einer Lösung für ein zusätzliches Ziel

Wenn Sie das User-Personalization-Rezept oder das Personalized-Ranking-Rezept verwenden, können Sie vor dem Training eine Amazon-Personalize-Lösung für ein Ziel zusätzlich zur maximalen Relevanz optimieren, z. B. die Umsatzmaximierung.

Bei Articleempfehlungsrezepten besteht das Hauptziel von Amazon Personalize darin, die relevantesten Elemente für Ihre Benutzer auf der Grundlage historischer und Echtzeit-Elementinteraktionsdaten vorherzusagen. Dies sind die Elemente, mit denen Ihre Benutzer höchstwahrscheinlich interagieren werden (z. B. die Elemente, auf die sie höchstwahrscheinlich klicken werden). Wenn Sie ein zusätzliches Ziel haben, z. B. die Maximierung von Streaming-Minuten oder die Umsatzsteigerung, können Sie eine Lösung erstellen, die Empfehlungen generiert, die sowohl auf der Relevanz als auch auf Ihrem Ziel basieren.

Um eine Lösung für ein zusätzliches Ziel zu optimieren, erstellen Sie eine neue Lösung mit dem User-Personalization-Rezept oder dem Personalized-Ranking-Rezept und wählen Sie die numerische Metadaten-Spalte in Ihrem Artikeldatensatz aus, die mit Ihrem Ziel zusammenhängt. Beim Generieren von Empfehlungen gibt Amazon Personalize Elementen mit höheren Werten für diese Datenspalte mehr Bedeutung. Sie können beispielsweise eine VIDEO_LENGTH-Spalte auswählen, um die Streaming-Minuten zu maximieren, oder eine PRICE-Spalte, um den Umsatz zu maximieren.

Sie können die Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs verwenden. Informationen zur Verwendung der Amazon-Personalize-Konsole finden Sie unter [Erstellen einer Lösung \(Konsole\)](#).

Themen

- [Richtlinien und Anforderungen](#)
- [Ausrichtung und Relevanz des Gleichgewichtsziels](#)
- [Messung der Optimierungsleistung](#)
- [Optimieren einer Lösung \(AWS CLI\)](#)
- [Optimieren einer Lösung \(AWS SDKs\)](#)

- [Beispiel für ein Jupyter-Notebook](#)

Richtlinien und Anforderungen

Die Zielanforderungen lauten wie folgt:

- Sie können nur eine Spalte für Ihr Ziel auswählen.
- Die Spalte muss einen numerischen Typ in Ihrem Schema haben.
- Die Spalte darf keinen null Typ in Ihrem Schema haben.

Weitere Informationen zu Schemata und Datentypen finden Sie unter [Schemata](#).

Ausrichtung und Relevanz des Gleichgewichtsziels

Es kann einen Kompromiss geben, wenn Sie Elemente empfehlen, die mehr auf Ihrem Ziel als auf Relevanz basieren. Wenn Sie beispielsweise den Umsatz durch Empfehlungen steigern möchten, könnten Empfehlungen nur für teure Artikel die Elemente für Ihre Benutzer weniger relevant machen und die Benutzerinteraktion und -konvertierung verringern.

Um das Gleichgewicht zwischen Relevanz und Ihrem Ziel zu konfigurieren, wählen Sie bei der Erstellung der Lösung eine der folgenden Zielsensibilitätsstufen aus:

- Aus: Amazon Personalize verwendet hauptsächlich Elementinteraktionsdaten, um die relevantesten Elemente für Ihren Benutzer vorherzusagen.
- Niedrig: Amazon Personalize legt weniger Wert auf Ihr Ziel. Die Bedeutung von Elementinteraktionsdaten ist wichtiger.
- Mittel: Amazon Personalize konzentriert sich anhand von Daten zu Elementinteraktionen gleich auf Ihr Ziel und Ihre Relevanz.
- Hoch: Amazon Personalize legt mehr Wert auf Ihr Ziel. Die Bedeutung von Elementinteraktionsdaten ist weniger wichtig.

Messung der Optimierungsleistung

Wenn Sie eine Lösungsversion für eine Lösung mit einem Optimierungsziel erstellen (Modell trainieren), generiert Amazon Personalize eine `-average_rewards_at_k` Metrik. Der Wert für `average_rewards_at_k` gibt an, wie gut die Lösungsversion abschneidet, um Ihr Ziel zu erreichen.

Um diese Metrik zu berechnen, berechnet Amazon Personalize die Belohnungen für jeden Benutzer wie folgt:

```
rewards_per_user = total rewards from the user's interactions with their  
top 25 reward generating recommendations / total rewards from the user's  
interactions with recommendations
```

Der Endwert `average_rewards_at_k` ist der Durchschnitt aller `rewards_per_user` normalisierten als Dezimalwert kleiner oder gleich 1 und größer als 0. Je näher der Wert bei 1 liegt, desto mehr Gewinne pro Benutzer können Sie von Empfehlungen erwarten.

Wenn Ihr Ziel beispielsweise darin besteht, den Umsatz durch Klicks zu maximieren, berechnet Amazon Personalize jede Benutzerbewertung, indem der Gesamtumsatz durch die Elemente geteilt wird, auf die der Benutzer aus seinen 25 teuersten Empfehlungen geklickt hat, durch den Umsatz aus allen empfohlenen Elementen, auf die der Benutzer geklickt hat. Amazon Personalize gibt dann einen normalisierten Durchschnitt aller Benutzerbewertungen zurück. Je näher der Wert bei 1 `average_rewards_at_k` liegt, desto mehr Umsatz können Sie durchschnittlich pro Benutzer aus Empfehlungen erwarten.

Weitere Informationen zum Generieren von Metriken finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#).

Optimieren einer Lösung (AWS CLI)

Sie können ein Ziel nur mit dem User-Personalization- oder Personalized-Ranking-Rezept optimieren. Um eine Lösung für ein zusätzliches Ziel mit der zu optimieren AWS CLI, erstellen Sie eine neue Lösung und geben Sie Ihre Zieldetails mit dem `optimizationObjective` Schlüssel im `-solutionConfig` Objekt an. Der `optimizationObjective` hat die folgenden Felder:

- `itemAttribute`: Geben Sie den Namen der numerischen Metadaten­spalte aus dem Datensatz Elemente an, der sich auf Ihr Ziel bezieht.
- `objectiveSensitivity`: Geben Sie den Schwerpunkt an, den die Lösung Ihrem Ziel bei der Generierung von Empfehlungen beimisst. Die Zielsensibilitätsstufe konfiguriert, wie Amazon Personalize anhand von Daten zur Elementinteraktion ein ausgewogenes Verhältnis zwischen Artikeln auf der Grundlage Ihres Ziels und Ihrer Relevanz sucht. Der `objectiveSensitivity` kann OFF, LOW MEDIUM oder sein HIGH. Weitere Informationen finden Sie unter [Ausrichtung und Relevanz des Gleichgewichtsziels](#).

Im Folgenden finden Sie ein Beispiel für den `create-solution` AWS CLI -Befehl. Ersetzen Sie die `recipe arn` Werte `solution namedataset group arn`, und durch Ihre eigenen Werte.

Ersetzen Sie für `COLUMN_NAME` den numerischen `optimizationObjective` Metadaten spaltennamen aus dem Artikeldatensatz, der sich auf Ihr Ziel bezieht. `objectiveSensitivity` Geben Sie für OFF, LOW, MEDIUM oder HIGH an.

```
aws personalize create-solution \  
--name solution name \  
--dataset-group-arn dataset group arn \  
--recipe-arn recipe arn \  
--solution-config "{\"optimizationObjective\":{\"itemAttribute\":\"COLUMN_NAME\",  
\"objectiveSensitivity\":{\"MEDIUM\"}}}"
```

Wenn Ihre Lösung bereit ist, erstellen Sie eine neue Lösungsversion (ein Beispielbefehl finden Sie unter [Erstellen einer Lösung \(AWS CLI\)](#)). Sobald Sie eine Lösungsversion erstellt haben, können Sie die Optimierungsleistung mit den Metriken der Lösungsversion anzeigen. Siehe [Messung der Optimierungsleistung](#).

Optimieren einer Lösung (AWS SDKs)

Sie können ein Ziel nur mit dem User-Personalization- oder Personalized-Ranking-Rezept optimieren.

Um eine Lösung für ein zusätzliches Ziel mithilfe der - AWS SDKs zu optimieren, erstellen Sie eine neue Lösung und geben Sie Ihre Zieldetails mithilfe des `optimizationObjective` Schlüssels im `-solutionConfig` Objekt für die Lösung an. Der `optimizationObjective` hat die folgenden Felder:

- `itemAttribute`: Geben Sie den Namen der numerischen Metadaten spalte aus dem Datensatz Elemente der Datensatzgruppe an, der sich auf Ihr Ziel bezieht.
- `objectiveSensitivity`: Geben Sie den Schwerpunkt an, den die Lösung Ihrem Ziel bei der Generierung von Empfehlungen beimisst. Die Zielsensibilitätsstufe konfiguriert, wie Amazon Personalize anhand von Daten zur Elementinteraktion ein ausgewogenes Verhältnis zwischen Artikeln auf der Grundlage Ihres Ziels und Ihrer Relevanz sucht. Der `objectiveSensitivity` kann OFF, LOW MEDIUM oder seinHIGH. Weitere Informationen finden Sie unter [Ausrichtung und Relevanz des Gleichgewichtsziels](#).

Verwenden Sie den folgenden Code, um eine Lösung mit einem zusätzlichen Ziel mit der AWS SDK for Python (Boto3) oder der zu erstellen AWS SDK for Java 2.x.

Wenn Ihre Lösung bereit ist, erstellen Sie eine neue Lösungsversion (Beispielcode finden Sie unter [Eine Lösungsversion \(SDKs\) erstellen AWS](#)). Sobald Sie eine Lösungsversion erstellt haben, können Sie die Optimierungsleistung mit den Metriken der Lösungsversion anzeigen. Siehe [Messung der Optimierungsleistung](#).

SDK for Python (Boto3)

Verwenden Sie die folgende `create_solution` Methode, um eine Lösung zu erstellen, die für ein zusätzliches Ziel optimiert ist. Ersetzen Sie die `recipe_arn` Werte `solution_name` `dataset_group_arn`, und durch Ihre eigenen Werte.

Ersetzen Sie für `COLUMN_NAME` den numerischen `optimizationObjective` Metadaten spaltennamen aus dem Artikeldatensatz, der sich auf Ihr Ziel bezieht. `objectiveSensitivity` Geben Sie für OFF, LOW, MEDIUM oder HIGH an.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name= 'solution name',
    recipeArn = 'recipe arn',
    datasetGroupArn = 'dataset group arn',
    solutionConfig = {
        "optimizationObjective": {
            "itemAttribute": "COLUMN_NAME",
            "objectiveSensitivity": "MEDIUM"
        }
    }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for Java 2.x

Um eine Lösung zu erstellen, die für ein zusätzliches Ziel optimiert ist, verwenden Sie die folgende `createPersonalizeSolution` Methode und übergeben Sie Folgendes als Parameter: einen Amazon-Personalize-Service-Client, den Amazon-Ressourcennamen (ARN) der Datensatzgruppe, einen Lösungsnamen, den Rezept-ARN, das Elementattribut und den Zielsensibilitätsgrad.

```
public static String createPersonalizeSolution(PersonalizeClient personalizeClient,
                                              String datasetGroupArn,
                                              String solutionName,
                                              String recipeArn,
                                              String itemAttribute,
                                              String objectiveSensitivity) {

    try {
        OptimizationObjective optimizationObjective =
        OptimizationObjective.builder()
            .itemAttribute(itemAttribute)
            .objectiveSensitivity(objectiveSensitivity)
            .build();

        SolutionConfig solutionConfig = SolutionConfig.builder()
            .optimizationObjective(optimizationObjective)
            .build();

        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .solutionConfig(solutionConfig)
            .build();

        CreateSolutionResponse solutionResponse =
        personalizeClient.createSolution(solutionRequest);

        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateSolutionCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create the personalizeClient
```

```
const personalizeClient = new PersonalizeClient({ region: "REGION"});

// set the solution parameters.
export const createSolutionParam = {
  datasetGroupArn: 'DATASET_GROUP_ARN',          /* required */
  recipeArn: 'RECIPE_ARN',                       /* required */
  name: 'NAME',                                  /* required */
  solutionConfig: {
    optimizationObjective: {
      itemAttribute: "COLUMN_NAME",              /* specify the numerical column from
the Items dataset related to your objective */
      objectiveSensitivity: "MEDIUM"            /* specify OFF, LOW, MEDIUM, or HIGH
*/
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateSolutionCommand(createSolutionParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

Beispiel für ein Jupyter-Notebook

Ein Beispiel für ein Jupyter-Notebook, das zeigt, wie eine Lösung erstellt wird, die für zusätzliche zielbasierte Elementmetadaten optimiert ist, finden Sie im Ordner [Objective_optimization](#) des [Amazon-Personalize-Beispiel](#) GitHub -Repositorys.

Hyperparameter und HPO

Sie geben vor dem Training Hyperparameter an, um das trainierte Modell für Ihren speziellen Anwendungsfall zu optimieren. Im Gegensatz dazu stehen Modellparameter, deren Werte während des Schulungsprozesses ermittelt werden.

Hyperparameter werden über den `algorithmHyperParameters`-Schlüssel angegeben, der Teil des `SolutionConfig`-Objekts ist, das an die `CreateSolution`-Operation übergeben wird.

Eine komprimierte Version des `CreateSolution`-Anfrage ist im Folgenden. Zum Beispiel gehört das `solutionConfig`-Objekt. Du benutzt `solutionConfig` zum Überschreiben der Standardparameter eines Rezepts.

```
{
  "name": "string",
  "recipeArn": "string",
  "eventType": "string",
  "solutionConfig": {
    "optimizationObjective": {
      "itemAttribute": "string",
      "objectiveSensitivity": "string"
    },
    "eventValueThreshold": "string",
    "featureTransformationParameters": {
      "string" : "string"
    },
    "algorithmHyperParameters": {
      "string" : "string"
    },
    "hpoConfig": {
      "algorithmHyperParameterRanges": {
        ...
      },
      "hpoResourceConfig": {
        "maxNumberOfTrainingJobs": "string",
        "maxParallelTrainingJobs": "string"
      }
    }
  },
},
}
```

Unterschiedliche Rezepte verwenden unterschiedliche Hyperparameter. Die verfügbaren Hyperparameter finden Sie in den einzelnen Rezepten unter [Ein Rezept wählen](#).

Aktivieren der Hyperparameter-Optimierung

Hyperparameter-Optimierung (Hyper-Parameter Optimization, HPO), oder Feineinstellung, bezeichnet die Wahl optimaler Hyperparameter für ein bestimmtes Lernziel. Die optimalen

Hyperparameter werden ermittelt, indem viele Schulungen mit unterschiedlichen Werten aus den vorgegebenen Möglichkeiten durchgeführt werden. Standardmäßig führt Amazon Personalize keine HPO aus. Um HPO zu verwenden, stellen Sie `performHPO` auf `true` ein und binden Sie das `hpoConfig`-Objekt ein.

Hyperparameter können kategorische, kontinuierliche oder Ganzzahl-Werte sein. Das `hpoConfig`-Objekt verfügt über Schlüssel, die jedem dieser Typen entsprechen, wobei Sie die Hyperparameter und deren Bereiche angeben. Sie müssen jeden Typ in Ihrer Anfrage angeben, aber wenn ein Rezept keinen Parameter eines Typs hat, können Sie ihn leer lassen. Zum Beispiel hat die Benutzerpersonalisierung keinen einstellbaren Hyperparameter vom kontinuierlichen Typ. Also zum `continuousHyperParameterRange` verwenden, würden Sie ein leeres Array übergeben.

Der folgende Code zeigt, wie Sie eine Lösung erstellen, bei der HPO mithilfe des SDK for Python (Boto3) aktiviert ist. Die Lösung im Beispiel verwendet den [User-Personalization-Rezept](#) und hat HPO auf `true` aus. Der Code liefert einen Wert für `hidden_dimension` und die `categoricalHyperParameterRanges` und `integerHyperParameterRanges` aus. Die `continuousHyperParameterRange` leer ist und der `hpoResourceConfig` setzt den `maxNumberOfTrainingJobs` und `maxParallelTrainingJobs` aus.

```
create_solution_response = personalize.create_solution(
    name = solutionName,
    datasetGroupArn = 'arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName',
    recipeArn = 'arn:aws:personalize:::recipe/aws-user-personalization',
    performHPO = True,
    solutionConfig = {
        "algorithmHyperParameters": {
            "hidden_dimension": "55"
        },
        "hpoConfig": {
            "algorithmHyperParameterRanges": {
                "categoricalHyperParameterRanges": [
                    {
                        "name": "recency_mask",
                        "values": [ "true", "false" ]
                    }
                ],
            "integerHyperParameterRanges": [
                {
                    "name": "bptt",
                    "minValue": 2,
```

```

        "maxValue": 22
      }
    ],
    "continuousHyperParameterRanges": [
      ]
  },
  "hpoResourceConfig": {
    "maxNumberOfTrainingJobs": "4",
    "maxParallelTrainingJobs": "2"
  }
}
)

```

Weitere Informationen zu HPO finden Sie unter [Automatische Modelloptimierung](#) aus.

Anzeigen von Hyperparametern

Sie können die Hyperparameter der Lösung anzeigen, indem Sie den [DescribeSolution](#) verwenden. Das folgende Beispiel zeigt eine `DescribeSolution` Ausgabe. Nach dem Erstellen einer Lösungsversion (Training eines Modells) können Sie Hyperparameter auch mit dem [DescribeSolutionVersion](#) verwenden.

```

{
  "solution": {
    "name": "hpo_config_solution",
    "solutionArn": "arn:aws:personalize:region:accountId:solution/solutionName",
    "performHPO": true,
    "performAutoML": false,
    "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
    "datasetGroupArn": "arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName",
    "eventType": "click",
    "solutionConfig": {
      "hpoConfig": {
        "hpoResourceConfig": {
          "maxNumberOfTrainingJobs": "4",
          "maxParallelTrainingJobs": "2"
        },
        "algorithmHyperParameterRanges": {
          "integerHyperParameterRanges": [
            {

```

```

        "name": "training.bptt",
        "minValue": 2,
        "maxValue": 22
    }
],
"continuousHyperParameterRanges": [],
"categoricalHyperParameterRanges": [
    {
        "name": "data.recency_mask",
        "values": [
            "true",
            "false"
        ]
    }
]
}
},
"algorithmHyperParameters": {
    "hidden_dimension": "55"
}
},
"status": "ACTIVE",
"creationDateTime": "2022-07-08T12:12:48.565000-07:00",
"lastUpdatedDateTime": "2022-07-08T12:12:48.565000-07:00"
}
}

```

Auswahl der für das Training verwendeten Artikelinteraktionsdaten

Sie können die Ereignisse in einem Datensatz mit Artikelinteraktionen auswählen, den Amazon Personalize beim Erstellen einer Lösungsversion (Training eines Modells) verwendet. Wenn Sie vor dem Training Artikelinteraktionsdaten auswählen, können Sie nur eine relevante Teilmenge Ihrer Daten für das Training verwenden oder Rauschen entfernen, um ein optimierteres Modell zu trainieren. Weitere Informationen zu Datensätzen zur Interaktion mit Gegenständen finden Sie unter [Schemata](#) und [Datensatz mit Interaktionen zwischen Elementen](#)

Sie können Daten zur Artikelinteraktion wie folgt auswählen:

- Datensätze nach Typ auswählen — Wenn Sie eine Lösung konfigurieren und Ihr Datensatz mit Artikelinteraktionen Ereignistypen in einer EVENT_TYPE-Spalte enthält, können Sie optional einen Ereignistyp angeben, der im Training verwendet werden soll. Wenn Ihr Datensatz mit Artikelinteraktionen beispielsweise die Ereignistypen Kauf, Klick und Ansehen umfasst und Sie

möchten, dass Amazon Personalize das Modell nur mit Zuschauerereignissen trainiert, würden Sie bei der Konfiguration Ihrer Lösung `watch as the` bereitstellen, `event_type` das Amazon Personalize in Schulungen verwendet.

Wenn Ihr Datensatz mit Artikelinteraktionen mehrere Ereignistypen in einer `EVENT_TYPE`-Spalte enthält und Sie bei der Konfiguration Ihrer Lösung keinen Ereignistyp angeben, verwendet Amazon Personalize alle Artikelinteraktionsdaten für ein Training mit gleichem Gewicht, unabhängig vom Typ.

- Wählen Sie Datensätze nach Typ und Wert aus — Wenn Sie eine Lösung konfigurieren und Ihr Datensatz mit Artikelinteraktionen die Felder `EVENT_TYPE` und `EVENT_VALUE` enthält, können Sie einen bestimmten Wert als Schwellenwert festlegen, um Datensätze vom Training auszuschließen. Wenn Ihre `EVENT_VALUE`-Daten für Ereignisse mit dem Wert `EVENT_TYPE` „Ansehen“ den Prozentsatz eines Videos darstellen, das ein Benutzer angesehen hat, und Sie den Schwellenwert für den Ereigniswert auf 0,5 und den anzuzeigenden Ereignistyp festlegen, trainiert Amazon Personalize das Modell, indem es nur Interaktionsereignisse verwendet, deren `EVENT_VALUE` größer oder gleich 0,5 ist.

Filtern von Datensätzen nach Ereigniswert und Ereignistyp (SDK) AWS

Im folgenden Verfahren verwenden Sie das AWS SDK for Python (Boto3), um ein Interaktionsschema zu erstellen, das ein Schulungsdataset filtert. Sie können ein Jupyter-Notebook (iPython) verwenden, um dieselbe Aufgabe auszuführen. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Personalize Personalize-APIs mit Jupyter \(IPython\) -Notebooks](#).

Voraussetzungen: Erfüllen Sie die Voraussetzungen und stellen Sie sicher, dass Ihre Python-Umgebung wie in [Erste Schritte \(SDK for Python \(Boto3\)\)](#) beschrieben eingerichtet ist.

So filtern Sie Datensätze, die in einem Schulungsdataset verwendet werden, nach Ereigniswert oder Ereignistyp

1. Erstellen Sie ein Interaktionsschema und schließen Sie die `EVENT_VALUE` Felder `EVENT_TYPE` "name" und mithilfe von "type" Schlüssel-Wert-Paaren ein, wie in der Abbildung gezeigt.

```
import boto3
import json

personalize = boto3.client('personalize')
```



```
# Create a name for your schema
schema_name = 'YourSchemaName'

# Define the schema for your dataset
schema = {
    "type": "record",
    "name": "Interactions",
    "namespace": "com.amazonaws.personalize.schema",
    "fields": [
        {
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "EVENT_VALUE",
            "type": "float"
        },
        {
            "name": "EVENT_TYPE",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ],
    "version": "1.0"
}

# Create the schema for Amazon Personalize
create_schema_response = personalize.create_schema(
    name = schema_name,
    schema = json.dumps(schema)
)

#To get the schema ARN, use the following lines
schema_arn = create_schema_response['schemaArn']
print('Schema ARN:' + schema_arn )
```

2. Formatieren Sie Ihre Eingabedaten entsprechend Ihrem Schema. Ein Codebeispiel finden Sie unter [Richtlinien für Datenformate](#).
3. Laden Sie Ihre Daten in einen Amazon Simple Storage Service (Amazon S3) -Bucket hoch. Ein Codebeispiel finden Sie unter [Hochladen des Hochladen des Uploads des Uploads des Amazon S3 Pakts](#).
4. Importieren Sie Ihre Daten mit der [CreateDatasetImportJob](#) API in Amazon Personalize. Achten Sie darauf, den Amazon-Ressourcennamen (ARN) Ihrer Dataset-Gruppe aufzuzeichnen, da Sie ihn beim erstellen der Lösung benötigen. Ein Codebeispiel finden Sie unter [Import von Massendatensätzen \(AWSSDKs\)](#).
5. Rufen Sie den ARN des Rezepts ab, das Sie verwenden möchten, wenn Sie Ihre Lösung erstellen. Sie benötigen ihn zum Erstellen der Lösung.

```
# Display the ARNs of the recipes
recipe_list = personalize.list_recipes()
for recipe in recipe_list['recipes']:
    print(recipe['recipeArn'])

# Store the ARN of the recipe that you want to use
recipe_arn = "arn:aws:personalize:::recipe/aws-recipe-name"
```

6. Rufen Sie die [CreateSolution](#)-API auf. Wenn Sie beispielsweise den Ereignistyp angeben möchten "purchase", legen Sie ihn im eventType Parameter fest. Wenn Sie einen Ereigniswert angeben möchten, z. B. 10, legen Sie ihn im Parameter eventValueThreshold fest. Sie können auch einen Ereignistyp und einen Ereigniswert angeben.

```
# Create the solution
create_solution_response = personalize.create_solution(
    name = "your-solution-name",
    datasetGroupArn = dataset_group_arn,
    recipeArn = recipe_arn,
    eventType = 'watched',
    solutionConfig = {
        "eventValueThreshold": "0.5"
    }
)

# Store the solution ARN
solution_arn = create_solution_response['solutionArn']

# Use the solution ARN to get the solution status
```

```
solution_description = personalize.describe_solution(solutionArn = solution_arn)
['solution']
print('Solution status: ' + solution_description['status'])
```

7. Wenn Sie die Lösung haben, verwenden Sie sie, um ein Modell zu schulen, indem Sie seinen Lösungs-ARN in einer [CreateSolutionVersion](#)-Anforderung angeben.

```
# Create a solution version
create_solution_version_response = personalize.create_solution_version(solutionArn
    = solution_arn)

# Store the solution version ARN
solution_version_arn = create_solution_version_response['solutionVersionArn']

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

Die Schulung ist abgeschlossen, wenn der Status ACTIVE ist. Weitere Informationen finden Sie unter [Erstellen einer Lösung und einer Lösungsversion](#).

Nachdem Sie ein Modell geschult haben, sollten Sie seine Leistung bewerten. Um Ihr Modell zu optimieren, können Sie den `eventValueThreshold`-Wert oder andere Hyperparameter anpassen. Weitere Informationen finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#).

Klonen einer Lösung (Konsole)

Wenn Sie eine neue Lösung erstellen, können Sie die Amazon Personalize-Konsole verwenden, um eine Lösung zu klonen. Wenn Sie eine Lösung klonen, können Sie die Konfiguration der vorhandenen Lösung als Ausgangspunkt verwenden, z. B. das Rezept und die Hyperparameter, und bei Bedarf Änderungen vornehmen. Dies ist nützlich, wenn Sie eine Änderung an einer Lösung vornehmen, aber alle anderen Eigenschaften unverändert lassen möchten. Fügen Sie beispielsweise Ihrem Datensatz eine neue Spalte mit Trainingsdaten hinzu. In diesem Fall würden Sie eine Lösung klonen, der Lösung einen Namen geben, die beim Training verwendeten Spalten ändern und alle anderen Eigenschaften unverändert lassen.

Klonen einer Lösung

Um eine Lösung zu klonen, wählen Sie die vorhandene Lösung und dann die Option Lösung klonen. Geben Sie dann der neuen Lösung einen Namen und ändern Sie die entsprechenden Felder.

Um eine Lösung zu klonen

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie Benutzerdefinierte Ressourcen und dann Lösungen.
4. Wählen Sie die Lösung, die Sie klonen möchten.
5. Wählen Sie Aktionen und dann Lösung klonen.
6. Geben Sie der neuen Lösung einen Namen.
7. Nehmen Sie alle Änderungen an den Lösungsdetails und der erweiterten Konfiguration vor. Amazon Personalize füllt diese Felder vorab mit Werten aus der vorhandenen Lösung aus. Informationen zu den einzelnen Feldern finden Sie unter [Erstellen und Konfigurieren einer Lösung](#).

Eine Lösungsversion erstellen

Nachdem Sie den Vorgang abgeschlossen haben [Erstellen und Konfigurieren einer Lösung](#), können Sie eine Lösungsversion erstellen. Eine Lösungsversion bezieht sich auf ein trainiertes Modell für maschinelles Lernen. Eine Lösungsversion ist eine benutzerdefinierte Ressource. Sie können eine Lösungsversion mit einer Amazon Personalize bereitstellen. Sie verwenden die Kampagne, um Empfehlungen für Benutzer zu erhalten. Und Sie können eine Lösungsversion verwenden, um Batch-Empfehlungen zu erhalten.

Sie können eine Lösungsversion mithilfe der Konsole AWS Command Line Interface (AWS CLI) oder der AWS SDKs erstellen. Wenn Ihre Lösungsversion den Status CREATE_PENDING oder CREATE_IN_PROGRESS hat, können Sie den [the section called "StopSolutionVersionCreation"](#) Vorgang verwenden, um die Erstellung der Lösungsversion zu beenden. Siehe [Stoppen der Erstellung einer Lösungsversion](#).

Themen

- [Eine Lösungsversion erstellen \(Konsole\)](#)
- [Eine Lösungsversion erstellen \(AWS CLI\)](#)

- [Eine Lösungsversion \(SDKs\) erstellen AWS](#)
- [Stoppen der Erstellung einer Lösungsversion](#)

Eine Lösungsversion erstellen (Konsole)

Wenn Sie Ihre Lösung zum ersten Mal mit der Amazon Personalize Personalize-Konsole erstellen, erstellen Sie auch eine Lösungsversion. Auf der Seite mit den Lösungsdetails können Sie den Trainingsfortschritt im Abschnitt Lösungsversionen verfolgen. Wenn die Schulung abgeschlossen ist, lautet der Status Aktiv und Sie sind bereit, eine Kampagne zu starten und Empfehlungen zu erhalten. Siehe [Eine Kampagne erstellen](#).

Wenn Sie eine zusätzliche Lösungsversion für eine bestehende Lösung erstellen möchten, erstellen Sie auf der Lösungsübersichtsseite wie folgt eine neue Lösungsversion.

Um eine neue Lösungsversion zu erstellen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Navigieren Sie zur Seite mit den Datensatzgruppen und wählen Sie die Datensatzgruppe mit Ihrer neuen Lösung aus.
3. Wählen Sie im Navigationsbereich unter Benutzerdefinierte Ressourcen die Option Lösungen und Rezepte aus.
4. Wählen Sie auf der Seite Lösung und Rezepte die Lösung aus, für die Sie eine Lösungsversion erstellen möchten.
5. Wählen Sie auf der Seite mit der Lösungsübersicht die Option Lösungsversion erstellen aus, um mit dem Training eines neuen Modells zu beginnen.

Auf der Seite mit den Lösungsdetails können Sie den Trainingsfortschritt im Abschnitt Lösungsversionen verfolgen. Wenn die Schulung abgeschlossen ist, lautet der Status Aktiv. Sie können sie anhand der von Amazon Personalize bereitgestellten Kennzahlen auswerten. Weitere Informationen finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#).

Wenn die Schulung aufgrund eines Fehlers nicht abgeschlossen werden kann, wird Ihnen die Schulung nicht in Rechnung gestellt. Wenn Ihre Lösungsversion den Status CREATE_PENDING oder CREATE_IN_PROGRESS hat, können Sie den Prozess zur Erstellung der Lösungsversion beenden. Um die Erstellung der Lösungsversion zu beenden, navigieren Sie zur Seite mit den

Lösungsversionsdetails und wählen Sie Beenden aus. Siehe [Stoppen der Erstellung einer Lösungsversion](#).

Eine Lösungsversion erstellen (AWS CLI)

Wenn Ihre Lösung AKTIV ist, trainieren Sie das Modell, indem Sie den folgenden Befehl ausführen. `solution` arn Ersetzen Sie durch die Lösung Amazon Resource Name (ARN) von [Erstellen und Konfigurieren einer Lösung](#).

```
aws personalize create-solution-version \  
  --solution-arn solution arn
```

Der ARN der Lösungsversion wird angezeigt, z. B.:

```
{  
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/  
<version-id>"  
}
```

Überprüfen Sie den Trainingsstatus der Lösungsversion mithilfe des `describe-solution-version` Befehls. Geben Sie den Lösungsversions-ARN an, der im vorherigen Schritt zurückgegeben wurde. Weitere Informationen zur API finden Sie unter [DescribeSolutionVersion](#).

```
aws personalize describe-solution-version \  
  --solution-version-arn solution version arn
```

Die Eigenschaften der Lösungsversion und des Schulungsstatus werden angezeigt. Anfänglich wird der Status als CREATE PENDING angezeigt, z. B.:

```
{  
  "solutionVersion": {  
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/  
solutionName/<version-id>",  
    ...,  
    "status": "CREATE PENDING"  
  }  
}
```

Die Schulung ist abgeschlossen, wenn sie abgeschlossen status ist, ACTIVE und Sie können sie anhand der von Amazon Personalize bereitgestellten Kennzahlen auswerten. Weitere Informationen

finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#). Wenn die Schulung aufgrund eines Fehlers nicht abgeschlossen werden kann, wird Ihnen die Schulung nicht in Rechnung gestellt.

Wenn Ihre Lösungsversion den Status `CREATE_PENDING` oder `CREATE_IN_PROGRESS` hat, können Sie den [StopSolutionVersionCreation](#) Vorgang verwenden, um die Erstellung der Lösungsversion zu beenden. Siehe [Stoppen der Erstellung einer Lösungsversion](#).

Eine Lösungsversion (SDKs) erstellen AWS

Wenn Ihre Lösung **AKTIV** ist, verwenden Sie den folgenden Code, um eine Lösungsversion zu erstellen. Geben Sie den Amazon-Ressourcennamen (ARN) von an [Erstellen und Konfigurieren einer Lösung](#). Verwenden Sie den [DescribeSolutionVersion](#) Vorgang, um den Status der Lösungsversion abzurufen.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')
# Store the solution ARN
solution_arn = 'solution_arn'

# Use the solution ARN to get the solution status.
solution_description = personalize.describe_solution(solutionArn = 'solution_arn')
['solution']
print('Solution status: ' + solution_description['status'])

# Use the solution ARN to create a solution version.
print ('Creating solution version')
response = personalize.create_solution_version(solutionArn = solution_arn)
solution_version_arn = response['solutionVersionArn']
print('Solution version ARN: ' + solution_version_arn)

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeSolutionVersion(PersonalizeClient
    personalizeClient, String solutionArn) {
```

```
    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String solutionStatus = "";
    String solutionVersionStatus = "";
    String solutionVersionArn = "";

    try {
        DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
            .solutionArn(solutionArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        // Wait until solution is active.
        while (Instant.now().getEpochSecond() < maxTime) {

            solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
            System.out.println("Solution status: " + solutionStatus);

            if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }

        // Once the solution is active, start creating a solution version.

        if (solutionStatus.equals("ACTIVE")) {

            CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
                .solutionArn(solutionArn)
                .build();

            CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient.createSolutionVersion(createSolutionVersionRequest);
```



```
        solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

        System.out.println("Solution version ARN: " + solutionVersionArn);

        DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
            .solutionVersionArn(solutionVersionArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            // Use the solution version ARN to get the solution version
status.
            solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest).solutionVersion()
            System.out.println("Solution version status: " +
solutionVersionStatus);

            if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return solutionVersionArn;
    }
} catch(PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSolutionVersionCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution version parameters.
export const solutionVersionParam = {
  solutionArn: 'SOLUTION_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateSolutionVersionCommand(solutionVersionParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Um den aktuellen Status der Lösungsversion zu überprüfen, rufen Sie die [DescribeSolutionVersion](#)-Operation auf und übergeben Sie den ARN der von der `CreateSolutionVersion`-Operation zurückgegebenen Lösungsversion. Die Schulung ist abgeschlossen, wenn sie abgeschlossen status ist, `ACTIVE` und Sie können sie anhand der von Amazon Personalize bereitgestellten Kennzahlen auswerten. Weitere Informationen finden Sie unter [Auswerten einer Lösungsversion mit Metriken](#). Wenn die Schulung aufgrund eines Fehlers nicht abgeschlossen werden kann, wird Ihnen die Schulung nicht in Rechnung gestellt.

Wenn Ihre Lösungsversion den Status `CREATE_PENDING` oder `CREATE_IN_PROGRESS` hat, können Sie den [StopSolutionVersionCreation](#) Vorgang verwenden, um die Erstellung der Lösungsversion zu beenden. Siehe [Stoppen der Erstellung einer Lösungsversion](#).

Stoppen der Erstellung einer Lösungsversion

Wenn Ihre Lösungsversion den Status `CREATE_PENDING` oder `CREATE_IN_PROGRESS` hat, können Sie die Amazon Personalize Console oder die [StopSolutionVersionCreation](#)-Operation, um die Erstellung der Lösungsversion zu beenden (beenden Sie das Training eines Modells). Sie können das Erstellen einer Lösungsversion nicht fortsetzen, nachdem sie beendet wurde. Ressourcen werden Ihnen bis zu dem Zeitpunkt in Rechnung gestellt, an dem die Erstellung der Lösungsversion beendet wurde.

Das Stoppen der Erstellung einer Lösungsversion beendet das Modelltraining, löscht jedoch nicht die Lösungsversion. Sie können die Details zur Lösungsversion weiterhin in der Amazon Personalize Personalisierungs-Konsole und mit der [DescribeSolutionVersion](#) verwenden.

Sie können den Erstellungsprozess der Lösungsversion mit der Amazon Personalize Personalisierungs-Konsole stoppen, AWS Command Line Interface (AWS CLI) oder das AWS SDKs.

Themen

- [Stoppen der Erstellung einer Lösungsversion \(Konsole\)](#)
- [Stoppen der Erstellung einer Lösungsversion \(AWS CLI\)](#)
- [Stoppen der Erstellung einer Lösungsversion \(AWS SDKs\)](#)

Stoppen der Erstellung einer Lösungsversion (Konsole)

Wenn Ihre Lösungsversion den Status `CREATE_PENDING` oder `CREATE_IN_PROGRESS` hat, können Sie die Erstellung einer Lösungsversion beenden (das Training eines Modells beenden).

So beenden Sie das Erstellen einer Lösungsversion (Konsole)

1. Öffnen Sie die Amazon Personalize Personalisierungs-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Auf der Datensatzgruppen Wählen Sie die Datensatzgruppe mit der Lösungsversion, die Sie anhalten möchten.
3. Wählen Sie im Navigationsbereich Lösungen und Rezeptenaus.
4. Auf der Lösung und Rezepten Wählen Sie die Lösung mit der Lösungsversion, die Sie anhalten möchten.
5. In :Lösungsversion Wählen Sie die Lösungsversion, die Sie anhalten möchten.

6. Wählen Sie auf der Seite zu den LösungsversionErstellen beendenaus. Abhängig vom ursprünglichen Status der Lösungsversion ändert sich der Status der Lösungsversion wie folgt:
- CREATE_PENDING wechselt zu CREATE_STOPPED.
 - CREATE_IN_PROGRESS wechselt zu CREATE_STOPPING und dann zu CREATE_STOPPING.

Stoppen der Erstellung einer Lösungsversion (AWS CLI)

Wenn Ihre Lösungsversion den Status CREATE_PENDING oder CREATE_IN_PROGRESS hat, können Sie die Erstellung einer Lösungsversion beenden (das Training eines Modells beenden). Verwenden Sie Folgendesstop-solution-version-creationBefehl, um das Erstellen der Lösungsversion mit demAWS CLIaus. Ersetzen `solution version arn` mit dem Amazon-Ressourcennamen (ARN) der Lösungsversion, die Sie anhalten möchten. Ihnen werden Ressourcen in Rechnung gestellt, die bis zu dem Zeitpunkt verwendet wurden, an dem die Erstellung der Lösungsversion gestoppt wurde.

```
aws personalize stop-solution-version-creation \  
  --solution-version-arn solution version arn
```

Überprüfen Sie den Schulungsstatus der Lösungsversion mit derdescribe-solution-versionbefehl.

```
aws personalize describe-solution-version \  
  --solution-version-arn solution version arn
```

Abhängig vom ursprünglichen Status der Lösungsversion ändert sich der Status der Lösungsversion wie folgt:

- CREATE_PENDING wechselt zu CREATE_STOPPED.
- CREATE_IN_PROGRESS wechselt zu CREATE_STOPPING und dann CREATE_STOPPING

Stoppen der Erstellung einer Lösungsversion (AWSSDKs)

Wenn Ihre Lösungsversion den Status CREATE_PENDING oder CREATE_IN_PROGRESS hat, können Sie die Erstellung einer Lösungsversion beenden (das Training eines Modells beenden). Der folgende Code zeigt, wie Sie das Erstellen einer Lösungsversion mit derAWS SDK for Python

(Boto3)oderAWS SDK for Java 2.xaus. Ihnen werden Ressourcen in Rechnung gestellt, die bis zu dem Zeitpunkt verwendet wurden, an dem die Erstellung der Lösungsversion gestoppt wurde.

SDK for Python (Boto3)

Verwenden Sie Folgendes`stop_solution_version_creation`-Methode, um die Erstellung einer Lösungsversion zu stoppen. Ersetzen`solution_version_arn`mit dem Amazon-Ressourcennamen (ARN) der Lösungsversion, die Sie anhalten möchten. Die Methode verwendet das[DescribeSolutionVersion](#)-Operation, um den Status der Lösungsversion abzurufen.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.stop_solution_version_creation(
    solutionVersionArn = solution_version_arn
)

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

Verwenden Sie Folgendes`stopSolutionVersionCreation`-Methode, um das Erstellen einer Lösungsversion zu beenden. Übergeben Sie als Parameter einen Amazon-Personalisierungs-Service-Client und den Amazon-Ressourcennamen (ARN) der Lösungsversion, die Sie nicht erstellen möchten. Der folgende Code verwendet den[DescribeSolutionVersion](#)-Operation, um den Status der Lösungsversion abzurufen.

```
public static void stopSolutionVersionCreation(PersonalizeClient personalizeClient,
String solutionVersionArn) {
    String solutionVersionStatus = "";

    StopSolutionVersionCreationRequest stopSolutionVersionCreationRequest =
StopSolutionVersionCreationRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .build();
```

```
personalizeClient.stopSolutionVersionCreation(stopSolutionVersionCreationRequest);

// Use the solution version ARN to get the solution version status.
DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
    .solutionVersionArn(solutionVersionArn)
    .build();

solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
    .solutionVersion()
    .status();
System.out.println("Solution version status: " + solutionVersionStatus);
}
```

Abhängig vom ursprünglichen Status der Lösungsversion ändert sich der Status der Lösungsversion wie folgt:

- CREATE_PENDING wechselt zu CREATE_STOPPED.
- CREATE_IN_PROGRESS wechselt zu CREATE_STOPPING und dann zu CREATE_STOPPING.

Auswerten einer Lösungsversion mit Metriken

Sie können die Leistung Ihrer Lösungsversion anhand von Offline- und Online-Metriken bewerten. Online-Metriken sind die empirischen Ergebnisse, die Sie bei der Interaktion Ihrer Benutzer mit Echtzeitempfehlungen beobachten. Sie können beispielsweise die Klickrate Ihrer Benutzer aufzeichnen, während sie Ihren Katalog durchsuchen. Sie sind dafür verantwortlich, alle Online-Metriken zu generieren und aufzuzeichnen.

Offline-Metriken sind die Metriken, die Amazon Personalize generiert, wenn Sie eine Lösungsversion trainieren. Sie können Offline-Metriken verwenden, um die Leistung des Modells zu bewerten, bevor Sie eine Kampagne erstellen und Empfehlungen geben. Offline-Metriken ermöglichen es Ihnen, die Auswirkungen der Änderung der Hyperparameter einer Lösung anzuzeigen oder Ergebnisse von Modellen zu vergleichen, die mit denselben Daten trainiert wurden. Für den Rest dieses Abschnitts bezieht sich der Begriff Metriken auf Offline-Metriken .

Um Leistungsmetriken zu erhalten, teilt Amazon Personalize die Eingabeinteraktionsdaten in einen Trainingssatz, einen Testsatz und für PERSONALIZED_ACTIONS, einen Validierungssatz, auf. Die Aufteilung hängt von der Art des ausgewählten Rezepts ab:

- Bei USER_SEGMENTATION-Rezepten besteht der Trainingssatz aus 80 % der Interaktionsdaten jedes Benutzers und der Testsatz aus 20 % der Interaktionsdaten jedes Benutzers.
- Für alle anderen Rezepttypen besteht der Trainingssatz aus 90 % Ihrer Benutzer und ihren Interaktionsdaten. Der Testsatz besteht aus den verbleibenden 10 % der Benutzer und ihren Interaktionsdaten.

Amazon Personalize erstellt dann die Lösungsversion mithilfe des Trainingssatzes. Nach Abschluss des Trainings gibt Amazon Personalize der neuen Lösungsversion die ältesten 90 % der Daten jedes Benutzers aus dem Testsatz als Eingabe. Amazon Personalize berechnet dann Metriken, indem es die Empfehlungen, die die Lösungsversion generiert, mit den tatsächlichen Interaktionen in den neuesten 10 % der Daten jedes Benutzers aus dem Testsatz vergleicht.

Um eine Baseline für Vergleichszwecke zu generieren, empfehlen wir die Verwendung des [-Beliebtheit-Anzahl](#) Rezepts, das die beliebtesten K-Elemente empfiehlt.

Themen

- [Abrufen von Metriken zur Lösungsversion](#)
- [Metrikdefinitionen](#)
- [Beispiel](#)
- [Weitere Ressourcen](#)

Abrufen von Metriken zur Lösungsversion

Nachdem Sie eine Lösungsversion erstellt haben, können Sie Metriken verwenden, um deren Leistung zu bewerten. Sie können Metriken für eine Lösungsversion mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) und AWS SDKs abrufen.

Themen

- [Abrufen von Metriken zur Lösungsversion \(Konsole\)](#)
- [Abrufen von Metriken zur Lösungsversion \(AWS CLI\)](#)
- [Abrufen von Lösungsversionsmetriken \(AWS SDKs\)](#)

Abrufen von Metriken zur Lösungsversion (Konsole)

Um Empfehlungsmetriken in der Konsole anzuzeigen, navigieren Sie zur Detailseite für Ihre Lösungsversion.

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre benutzerdefinierte Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Benutzerdefinierte Ressourcen und dann Lösungen und Rezepte aus.
4. Wählen Sie Ihre Lösung aus.
5. Wählen Sie unter Lösungsversionen Ihre Lösungsversion aus, um die Detailseite anzuzeigen. Die Metriken werden auf der Registerkarte Metriken der Lösungsversion im unteren Bereich aufgeführt. Definitionen von Metriken finden Sie unter [Metrikdefinitionen](#).

Nachdem Sie Ihre Lösungsversion ausgewertet haben, können Sie eine Kampagne erstellen, indem Sie die Lösungsversion mit den besten Metriken für Ihren Anwendungsfall bereitstellen. Weitere Informationen zur Bereitstellung einer Lösung finden Sie unter [Eine Kampagne erstellen](#).

Abrufen von Metriken zur Lösungsversion (AWS CLI)

Sie erhalten die Metriken für eine bestimmte Lösungsversion, indem Sie die [GetSolutionMetrics](#) Operation aufrufen. Der folgende Code zeigt, wie Metriken mit der abgerufen werden AWS CLI.

```
personalize get-solution-metrics --solution-version-arn solution version ARN
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Lösungsversion, die mit dem [Benutzerpersonalisierung](#) Rezept mit einem zusätzlichen Optimierungsziel erstellt wurde.

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>",
  "metrics": {
    "coverage": 0.27,
    "mean_reciprocal_rank_at_25": 0.0379,
    "normalized_discounted_cumulative_gain_at_5": 0.0405,
    "normalized_discounted_cumulative_gain_at_10": 0.0513,
```



```
    "normalized_discounted_cumulative_gain_at_25": 0.0828,  
    "precision_at_5": 0.0136,  
    "precision_at_10": 0.0102,  
    "precision_at_25": 0.0091,  
    "average_rewards_at_k": 0.653  
  }  
}
```

Eine Erläuterung der einzelnen Metriken finden Sie unter [Metrikdefinitionen](#). Nachdem Sie Ihre Lösungsversion ausgewertet haben, können Sie eine Kampagne erstellen, indem Sie die Lösungsversion mit den besten Metriken für Ihren Anwendungsfall bereitstellen. Weitere Informationen zur Bereitstellung einer Lösung finden Sie unter [Eine Kampagne erstellen](#).

Abrufen von Lösungsversionsmetriken (AWS SDKs)

Sie erhalten die Metriken für eine bestimmte Lösungsversion, indem Sie die [GetSolutionMetrics](#) Operation aufrufen. Verwenden Sie den folgenden Code zum Abrufen von Metriken.

SDK for Python (Boto3)

```
import boto3  
  
personalize = boto3.client('personalize')  
  
response = personalize.get_solution_metrics(  
    solutionVersionArn = 'solution version arn')  
  
print(response['metrics'])
```

SDK for Java 2.x

```
public static void getSolutionVersionMetrics(PersonalizeClient personalizeClient,  
    String solutionVersionArn) {  
  
    try {  
        GetSolutionMetricsRequest request = GetSolutionMetricsRequest.builder()  
            .solutionVersionArn(solutionVersionArn)  
            .build();  
        Map<String, Double> metrics =  
personalizeClient.getSolutionMetrics(request).metrics();  
        metrics.forEach((key, value) -> System.out.println(key + " " + value));  
    }  
}
```

```
} catch (PersonalizeException e ) {  
    System.err.println(e.awsErrorDetails().errorMessage());  
    System.exit(1);  
}  
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Lösungsversion, die mit dem [Benutzerpersonalisierung](#) Rezept mit einem zusätzlichen Optimierungsziel erstellt wurde.

```
{  
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/  
MovieSolution/<version-id>",  
  "metrics": {  
    "coverage": 0.27,  
    "mean_reciprocal_rank_at_25": 0.0379,  
    "normalized_discounted_cumulative_gain_at_5": 0.0405,  
    "normalized_discounted_cumulative_gain_at_10": 0.0513,  
    "normalized_discounted_cumulative_gain_at_25": 0.0828,  
    "precision_at_5": 0.0136,  
    "precision_at_10": 0.0102,  
    "precision_at_25": 0.0091,  
    "average_rewards_at_k": 0.653  
  }  
}
```

Eine Erläuterung der einzelnen Metriken finden Sie unter [Metrikdefinitionen](#). Nachdem Sie Ihre Lösungsversion ausgewertet haben, können Sie eine Kampagne erstellen, indem Sie die Lösungsversion mit den besten Metriken für Ihren Anwendungsfall bereitstellen. Weitere Informationen zur Bereitstellung einer Lösung finden Sie unter [Eine Kampagne erstellen](#).

Metrikdefinitionen

Die Metriken, die Amazon Personalize für Lösungsversionen generiert, werden unten unter Verwendung der folgenden Begriffe beschrieben:

- Relevante Empfehlung ist eine Empfehlung für ein Element, mit dem der Benutzer tatsächlich interagiert hat. Diese Elemente stammen aus den neuesten 10 % der Interaktionsdaten jedes Benutzers aus dem Testsatz.
- Rang bezieht sich auf die Position eines empfohlenen Elements in der Liste der Empfehlungen. Position 1 (ganz oben in der Liste) ist möglicherweise am relevantesten für den Benutzer.

Für jede Metrik sind höhere Zahlen (naher an 1) besser. Weitere Informationen finden Sie in den Ressourcen unter [Weitere Ressourcen](#).

Abdeckung

Der Wert für die Abdeckung gibt den Anteil der eindeutigen Elemente (für Elementempfehlungen), Aktionen (für Aktionsempfehlungen) oder Benutzer (für Benutzersegmentempfehlungen) an, den Amazon Personalize aus der Gesamtzahl der eindeutigen Datensätze in Ihren Datensätzen empfehlen könnte.

Eine höhere Abdeckungsbewertung bedeutet, dass Amazon Personalize mehr von Ihrem Katalog empfiehlt als die Datensätze wiederholt. Rezepte, bei denen Elementexploration wie User-Personalization stattfindet, haben eine höhere Abdeckung als solche, die dies nicht tun, z. B. Ähnliche Elemente.

Mittlerer wechselseitiger Rang bei 25

Diese Metrik informiert Sie über die Fähigkeit eines Modells, relevante Elementempfehlungen an der obersten Position zu generieren.

Sie können ein Modell mit einem hohen mittleren reziproken Rang bei 25 wählen, wenn Sie Elementsuchergebnisse für einen Benutzer generieren und nicht erwarten, dass der Benutzer ein Element unten in der Liste auswählt. Beispielsweise wählen Benutzer in den Suchergebnissen häufig das erste Rezept aus. Amazon Personalize generiert diese Metrik nicht für PERSONALIZED_ACTIONS- oder USER_SEGMENTATION-Rezepte.

Amazon Personalize berechnet diese Metrik anhand des durchschnittlichen reziproken Rangwerts für Anfragen nach Empfehlungen. Jeder reziproke Rangwert wird wie folgt berechnet: $1 / \text{the rank of the highest item interacted with by the user}$, wobei die insgesamt mögliche Rangfolge 25 beträgt. Andere Elemente mit niedrigerem Rang, mit denen der Benutzer interagiert, werden ignoriert. Wenn der Benutzer das erste Element ausgewählt hat, ist der Wert 1. Wenn sie keine Elemente auswählen, ist der Wert 0.

Sie können beispielsweise drei verschiedenen Benutzern jeweils 25 Empfehlungen anzeigen:

- Wenn Benutzer 1 auf das Element mit Rang 4 und auf das Element mit Rang 10 klickt, liegt ihr reziproscher Rang bei $1/4$.
- Wenn Benutzer 2 auf ein Element mit Rang 2, ein Element mit Rang 4 und ein Element mit Rang 12 klickt, beträgt ihr gegenseitiger Rangwert $1/2$.
- Wenn Benutzer 3 auf ein einzelnes Element mit Rang 6 klickt, liegt ihr reziproscher Rang bei $1/6$.

Der mittlere reziproke Rang für alle Anfragen nach Empfehlungen (in diesem Fall 3) wird als berechnet $(1/4 + 1/2 + 1/6) / 3 = .3056$.

normalisierter abgewerteter kumulativer Gewinn (NDCG) bei K (5/10/25)

Diese Metrik informiert Sie darüber, wie gut Ihr Modell Element- oder Aktionsempfehlungen bewertet, wobei K eine Stichprobengröße von 5, 10 oder 25 Empfehlungen ist. Diese Metrik ist nützlich, wenn Sie am meisten an der Rangfolge von Empfehlungen interessiert sind, die nur über das Element oder die Aktion mit dem höchsten Rang hinausgehen (siehe dazu `mean reciprocal rank at 25`). Der Wert für NDCG at 10 wäre beispielsweise nützlich, wenn Sie eine Anwendung haben, die bis zu 10 Filme gleichzeitig in einem Karussell anzeigt.

Amazon Personalize berechnet den NDCG, indem Empfehlungen basierend auf ihrer Rangfolgeposition für jeden Benutzer im Testsatz eine Gewichtung zugewiesen wird. Jede Empfehlung wird (bei geringerer Gewichtung) um einen Faktor reduziert, der von ihrer Position abhängt. Die endgültige Metrik ist der Durchschnitt aller Benutzer im Testsatz. Der normalisierte, abgewertete kumulative Zuwachs bei K geht davon aus, dass Empfehlungen, die in einer Liste weiter unten stehen, weniger relevant sind als Empfehlungen, die weiter oben stehen.

Amazon Personalize verwendet einen Gewichtungsfaktor von $1/\log(1 + \text{position})$, wobei ganz oben in der Liste Position steht 1.

Genauigkeit bei K

Diese Metrik gibt an, wie relevant die Empfehlungen Ihres Modells sind, basierend auf einer Stichprobengröße von K (5, 10 oder 25) Empfehlungen.

Amazon Personalize berechnet diese Metrik auf der Grundlage der Anzahl relevanter Empfehlungen aus den wichtigsten K-Empfehlungen für jeden Benutzer im Testsatz geteilt durch K, wobei K 5, 10 oder 25 ist. Die endgültige Metrik ist der Durchschnitt aller Benutzer im Testsatz.

Wenn Sie beispielsweise einem Benutzer 10 Elemente empfehlen und der Benutzer mit 3 davon interagiert, beträgt die Genauigkeit bei K 3 korrekt vorhergesagte Elemente geteilt durch die insgesamt 10 empfohlenen Elemente: $3 / 10 = .30$.

Diese Metrik gewinnt eine präzise Empfehlung relevanter Elemente. Je näher der Wert an eins liegt, desto genauer ist das Modell.

precision

Wenn Sie eine Lösungsversion mit dem Next-Best-Action-Rezept trainieren, generiert Amazon Personalize eine `precision` Metrik anstelle von `precision at K`. Diese Metrik gibt an, wie gut Ihr Modell bei der Vorhersage von Aktionen ist, die Benutzer tatsächlich ergreifen werden.

Um zu berechnen `precision`, dividiert Amazon Personalize für jede Aktion in Ihrem Datensatz die Anzahl der Benutzer, von denen korrekt vorhergesagt wurde, dass sie die Aktion ausführen, durch die Gesamtzahl der empfohlenen Aktionen. Amazon Personalize berechnet dann den Durchschnitt für alle Aktionen in Ihrem Datensatz.

Wenn beispielsweise eine Aktion für 100 Benutzer empfohlen wurde und 60 Benutzer die Aktion ausgeführt haben und 40 Benutzer, die dies nicht getan haben, lautet der `precision` für die Aktion $60 / 100 = .60$. Amazon Personalize wendet diese Berechnung dann für alle Aktionen an und gibt den Durchschnitt zurück.

Diese Metrik gewinnt eine präzise Empfehlung relevanter Aktionen. Je näher der Wert an eins liegt, desto genauer ist das Modell.

average_rewards_at_k

Wenn Sie eine Lösungsversion für eine Lösung mit einem Optimierungsziel erstellen (Modell trainieren), generiert Amazon Personalize eine `-average_rewards_at_k` Metrik. Der Wert für `average_rewards_at_k` gibt an, wie gut die Lösungsversion abschneidet, um Ihr Ziel zu erreichen. Um diese Metrik zu berechnen, berechnet Amazon Personalize die Belohnungen für jeden Benutzer wie folgt:

```
rewards_per_user = total rewards from the user's interactions with their  
top 25 reward generating recommendations / total rewards from the user's  
interactions with recommendations
```

Der Endwert `average_rewards_at_k` ist der Durchschnitt aller `rewards_per_user` normalisierten als Dezimalwert kleiner oder gleich 1 und größer als 0. Je näher der Wert bei 1 liegt, desto mehr Gewinne pro Benutzer können Sie von Empfehlungen erwarten.

Wenn Ihr Ziel beispielsweise darin besteht, den Umsatz durch Klicks zu maximieren, berechnet Amazon Personalize jede Benutzerbewertung, indem der Gesamtumsatz durch die Elemente geteilt wird, auf die der Benutzer aus seinen 25 teuersten Empfehlungen geklickt hat, durch den Umsatz aus allen empfohlenen Elementen, auf die der Benutzer geklickt hat. Amazon Personalize gibt dann einen normalisierten Durchschnitt aller Benutzerbewertungen zurück. Je näher der

Wert bei 1 `average_rewards_at_k` liegt, desto mehr Umsatz können Sie durchschnittlich pro Benutzer aus Empfehlungen erwarten.

Weitere Informationen finden Sie unter [Optimierung einer Lösung für ein zusätzliches Ziel](#).

Genauigkeit der Trendvorhersage

Wenn Sie die Lösungsversion mit dem [-Trending – Jetzt](#) Rezept trainiert haben, ist dies die Steigerung der Popularität der vom Modell empfohlenen Elemente. Je höher die Genauigkeit der Trendvorhersage (je näher an 1) ist, desto besser ist es, dass das Modell Trendelemente korrekt identifiziert.

Um die Beschleunigung der Beliebtheit zu berechnen, teilt Amazon Personalize die Rate der Zunahme der Popularität auf alle empfohlenen Elemente durch die gesamte Zunahme der 25 beliebtesten Elemente auf. Diese Elemente stammen aus den tatsächlichen Interaktionen im Testsatz.

Abhängig von Ihrer Datenverteilung und dem, was Sie für die Häufigkeit der Trenderkennung wählen, kann der Wert für die Genauigkeit der Trendvorhersage 0,0 sein.

Treffer (Treffer bei K)

Wenn Sie die Lösungsversion mit einem `USER_SEGMENTATION`-Rezept trainiert haben, die durchschnittliche Anzahl der Benutzer im prognostizierten Top- relevanten K-Ergebnis, die den tatsächlichen Benutzern entsprechen. Tatsächliche Benutzer sind die Benutzer, die tatsächlich mit den Elementen im Testsatz interagiert haben. K ist die 1 % der relevantesten Benutzer. Je höher der Wert, desto genauer sind die Vorhersagen.

Erinnerung (Wiedererkennung bei K)

Wenn Sie die Lösungsversion mit einem `USER_SEGMENTATION`-Rezept trainiert haben, der durchschnittliche Prozentsatz der vorhergesagten Benutzer in den prognostizierten Top-relevanten K-Ergebnissen, die den tatsächlichen Benutzern entsprechen. Tatsächliche Benutzer sind die Benutzer, die tatsächlich mit den Elementen im Testsatz interagiert haben. K ist die 1 % der relevantesten Benutzer. Je höher der Wert, desto genauer sind die Vorhersagen.

Rückruf

Wenn Sie eine Lösungsversion mit dem Next-Best-Action-Rezept trainieren, teilt Ihnen diese Metrik mit, wie gut Ihre Lösungsversion darin ist, Aktionen zu erkennen, mit denen Benutzer interagieren werden.

Zur Berechnung von `recall` für jede Aktion in Ihrem Datensatz die Anzahl der Benutzer, von denen korrekt vorhergesagt wurde, dass sie die Aktion ausführen, durch die Gesamtzahl der Benutzer, die die Aktion im Testset tatsächlich ausführen. Amazon Personalize berechnet dann den Durchschnitt für alle Aktionen in Ihrem Datensatz.

Wenn beispielsweise 100 Benutzer eine Aktion im Testset ausführen und Amazon Personalize prognostizierte 50 dieser Benutzer die Aktion ausführen würden, lautet die `recall` für die Aktion $50 / 100 = .50$. Amazon Personalize wendet diese Berechnung dann für alle Aktionen an und gibt den Durchschnitt zurück.

Fläche unter der Kurve (AUC)

Wenn Sie die Lösungsversion mit einem `PERSONALIZED_ACTIONS`-Rezept trainiert haben, der Bereich unter der Kurve Betriebsmerkmale des Empfängers für Ihre Lösungsversion. Diese Metrik gibt an, wie gut die Lösungsversion bei der korrekten Identifizierung von Aktionen funktioniert, die Benutzer ausführen werden.

Die Kurve Receiver Operating Characteristic zeichnet die Leistung der Lösungsversion. Es zeichnet die wirklich positiven (Aktionen, die korrekt als relevant vorhergesagt wurden) und falsch positiven (Aktionen, die falsch als relevant vorhergesagt wurden) Raten bei unterschiedlichen Schwellenwerten auf. Die Fläche unter der Kurve (AUC) ist ein Wert, der die Leistung der Lösungsversion basierend auf ihrer Kurve zusammenfasst.

Die AUC einer Lösungsversion kann zwischen 0 und 1 liegen. Je näher an 1 liegt, desto besser kann das Modell relevante Aktionen für Ihre Benutzer vorhersagen.

Beispiel

Im Folgenden finden Sie ein einfaches Beispiel für eine Lösungsversion, die eine Liste von Empfehlungen für einen bestimmten Benutzer erstellt. Die zweite und fünfte Empfehlung stimmt mit Datensätzen in den Testdaten für diesen Benutzer überein. Dies sind die relevanten Empfehlungen. Wenn K auf 5 gesetzt ist, werden die folgenden Metriken für den Benutzer generiert.

`reciprocal_rank`

Berechnung: $1/2$

Ergebnis: 0,5000

`normalized_discounted_cumulative_gain_at_5`

Berechnung: $(1/\log(1 + 2) + 1/\log(1 + 5)) / (1/\log(1 + 1) + 1/\log(1 + 2))$

Ergebnis: 0,6241

precision_at_5

Berechnung: 2/5

Ergebnis: 0,4000

Weitere Ressourcen

Informationen zur Bewertung einer Lösungsversion mit A/B-Tests finden Sie unter [Verwenden von A/B-Tests zur Messung der Wirksamkeit von Empfehlungen, die von Amazon Personalize generiert werden](#). Weitere Informationen zu verschiedenen Arten von Metriken für Empfehlungssysteme finden Sie in den folgenden externen Ressourcen:

- [MRR im Vergleich zu MAP im Vergleich zu NDCG: Bewertungsmetriken mit Rangerkennung und wann sie verwendet werden sollten](#)
- [Rabattierter kumulativer Gewinn: Die Rangfestlegungsmetriken, die Sie kennen sollten](#)
- [Erinnerung und Präzision bei k für Empfehlungssysteme](#)
- [Bewertungsmetriken für Empfehlungssysteme](#)
- [Betriebsmerkmale des Empfängers](#)

Eine Kampagne erstellen

Wenn Sie Empfehlungen in Echtzeit mit benutzerdefinierten Ressourcen erhalten möchten [Eine Lösungsversion erstellen](#), können Sie nach Abschluss des Vorgangs Ihre Lösungsversion mit einer Kampagne bereitstellen.

Bei einer Kampagne wird eine Lösungsversion (trainiertes Modell) mit einer bereitgestellten Transaktionskapazität zur Generierung von Empfehlungen in Echtzeit bereitgestellt.

Nachdem Sie eine Kampagne erstellt haben, verwenden Sie die [GetRecommendations](#) oder [GetPersonalizedRanking](#) API-Operationen, um Empfehlungen zu erhalten. Wenn Sie Empfehlungen auf einmal erhalten, müssen Sie keine Kampagne erstellen. Weitere Informationen finden Sie unter [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#).

Wenn Sie eine Kampagne erstellen, können Sie Artikelmetadaten in Empfehlungen aktivieren. Weitere Informationen finden Sie unter [Metadaten in Empfehlungen aktivieren](#).

Sie können eine Kampagne mit der Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.


 **Important**

Wenn Sie Ihre Lösungsversion manuell neu trainieren oder die Einstellungen einer bestehenden Kampagne ändern möchten, z. B. die Aktivierung von Metadaten in Empfehlungen, müssen Sie Ihre Kampagne aktualisieren. Weitere Informationen finden Sie unter [Aktualisieren einer Kampagne](#).

Themen

- [Minimale Anzahl bereitgestellter Transaktionen pro Sekunde und auto-scaling](#)
- [Metadaten in Empfehlungen aktivieren](#)
- [Eine Kampagne erstellen \(Konsole\)](#)
- [Eine Kampagne erstellen \(\) AWS CLI](#)
- [Eine Kampagne \(AWSSDKs\) erstellen](#)
- [Aktualisieren einer Kampagne](#)

Minimale Anzahl bereitgestellter Transaktionen pro Sekunde und auto-scaling

 **Important**

Ein Hoch erhöht `minProvisionedTPS` Ihre Kosten. Wir empfehlen, mit 1 für zu beginnen `minProvisionedTPS` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minProvisionedTPS` diese bei Bedarf.

Wenn Sie eine Amazon Personalize Personalize-Kampagne erstellen, können Sie die Mindestanzahl an bereitgestellten Transaktionen pro Sekunde (`minProvisionedTPS`) für die Kampagne angeben. Dies ist der grundlegende Transaktionsdurchsatz für die von Amazon Personalize bereitgestellte Kampagne. Es legt die Mindestabrechnungsgebühr für die Kampagne fest, solange sie aktiv ist. Bei einer Transaktion handelt es sich um eine einzelne Transaktion `GetRecommendations` oder eine `GetPersonalizedRanking` Anfrage. Die Standardeinstellung `minProvisionedTPS` ist 1.

Wenn Ihr TPS über den Wert hinaus ansteigt `minProvisionedTPS`, skaliert Amazon Personalize die bereitgestellte Kapazität automatisch nach oben und unten, aber niemals darunter.

`minProvisionedTPS` Bei der Erhöhung der Kapazität kommt es zu einer kurzen Zeitverzögerung, die zum Verlust von Transaktionen führen kann. Wenn Ihr Traffic abnimmt, kehrt die Kapazität auf die zurück `minProvisionedTPS`.

Ihnen wird das bereitgestellte Mindest-TPS oder, falls Ihre Anfragen das TPS überschreiten `minProvisionedTPS`, das tatsächliche TPS in Rechnung gestellt. Der tatsächliche TPS ist die Gesamtzahl der von Ihnen gestellten Empfehlungsanfragen. Wir empfehlen, mit einem niedrigen Wert zu beginnen `minProvisionedTPS`, Ihre Nutzung anhand von CloudWatch Amazon-Metriken zu verfolgen und den Wert dann bei `minProvisionedTPS` Bedarf zu erhöhen.

Weitere Informationen zu den Kampagnenkosten finden Sie unter [Amazon Personalize Personalize-Preise](#).

Metadaten in Empfehlungen aktivieren

Important

Wenn Sie Metadaten in Empfehlungen aktivieren, entstehen Ihnen zusätzliche Kosten. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Preise](#).

Wenn Sie eine Kampagne erstellen, können Sie die Option aktivieren, Artikelmetadaten aus Ihrem Artikeldatensatz mit Empfehlungsergebnissen einzubeziehen. Anschließend können Sie die Spalten aus Ihrem Artikeldatensatz auswählen, die Sie in Ihre Anfrage nach Empfehlungen oder personalisiertem Ranking aufnehmen möchten. Amazon Personalize gibt diese Daten für jeden Artikel in der Empfehlungsantwort zurück.

Sie können Metadaten verwenden, um Empfehlungen in Ihrer Benutzeroberfläche zu erweitern, z. B. um Genres für Filme zu Karussells hinzuzufügen. Oder Sie können sie verwenden, um die Qualität der Empfehlungen visuell zu beurteilen. Wenn Sie generative KI in Ihrer App verwenden, können Sie die Metadaten in KI-Eingabeaufforderungen einfügen, um relevantere Inhalte zu generieren. Weitere Informationen zur Verwendung von Amazon Personalize mit generativer KI finden Sie unter [Amazon Personalize und generative KI](#).

- Um Metadaten mit der Amazon Personalize-Konsole zu aktivieren, wählen Sie bei der Erstellung der Kampagne in den Empfehlungsergebnissen in den Kampagnendetails die Option Artikelmetadaten zurücksenden aus. Weitere Informationen finden Sie unter [Eine Kampagne erstellen \(Konsole\)](#).

- Um Metadaten mit den AWS SDKs oder zu aktivieren AWS CLI, verwenden Sie die [CreateCampaign](#) API-Operation und `campaignConfig.enableMetadataInInferenceResponse` auf `true`. Weitere Informationen finden Sie unter [Eine Kampagne erstellen \(\) AWS CLI](#) oder [Eine Kampagne \(AWS SDKs\) erstellen](#).

Um Metadaten zu Empfehlungen hinzufügen zu können, benötigen Sie einen Artikeldatensatz mit einer Spalte mit Metadaten. Sie müssen die Metadaten nicht im Training verwenden. Hinweise zum Erstellen eines Datensatzes finden Sie unter [Einen Datensatz und ein Schema erstellen](#). Informationen zur Verwaltung und Aktualisierung von Daten finden Sie unter [Daten verwalten](#).

Eine Kampagne erstellen (Konsole)

Sobald der Status Ihrer Lösungsversion Aktiv lautet, können Sie sie mit einer Amazon Personalize bereitstellen.

Um eine Kampagne zu erstellen (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe mit der Lösungsversion aus, die Sie bereitstellen möchten.
3. Wählen Sie im Navigationsbereich unter Benutzerdefinierte Ressourcen die Option Kampagnen aus.
4. Wählen Sie auf der Seite Kampagnen die Option Kampagne erstellen aus.
5. Geben Sie auf der Seite Neue Kampagne erstellen für Kampagnendetails die folgenden Informationen ein:
 - Kampagnenname — Geben Sie den Namen der Kampagne ein. Der Text, den Sie hier eingeben, wird im Kampagnen-Dashboard und auf der Detailseite angezeigt.
 - Lösung — Wählen Sie die Lösung aus, die Sie gerade erstellt haben.
 - Lösungsversions-ID — Wählen Sie die ID der Lösungsversion, die Sie gerade erstellt haben.
 - Minimale Anzahl bereitgestellter Transaktionen pro Sekunde (in APIs als `minProvisionedTPS` bezeichnet) — Legen Sie die Mindestanzahl bereitgestellter Transaktionen pro Sekunde fest, die Amazon Personalize unterstützt. Ein hoher Wert erhöht Ihre Rechnung. Wir empfehlen, mit 1 zu beginnen (Standardeinstellung). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minProvisionedTPS` diese bei Bedarf. Weitere Informationen finden Sie unter [Minimale Anzahl bereitgestellter Transaktionen pro Sekunde und auto-scaling](#).

- Artikelmetadaten in den Empfehlungsergebnissen zurückgeben — Wählen Sie diese Option, wenn Sie Metadaten in die Empfehlungsergebnisse einbeziehen möchten. Wenn diese Option aktiviert ist, können Sie die Spalten aus Ihrem Artikeldatensatz in Ihrer Anfrage nach Empfehlungen oder personalisiertem Ranking angeben. Amazon Personalize gibt diese Daten für jeden Artikel in der Empfehlungsantwort zurück.

Um Metadaten zu aktivieren, benötigen Sie einen Artikeldatensatz mit einer Spalte mit Metadaten.

6. Wenn Sie das Rezept für die Benutzerpersonalisierung verwendet haben, geben Sie in der Kampagnenkonfiguration optional Werte für das Gewicht von Exploration und das Mindestalter für das Alter von Exploration-Artikeln ein. Weitere Informationen finden Sie unter [Benutzerpersonalisierung](#).
7. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).
8. Wählen Sie Create campaign (Kampagne erstellen).
9. Wenn der Kampagnenstatus auf der Seite mit den Kampagnendetails Aktiv lautet, können Sie die Kampagne verwenden, um Empfehlungen zu erhalten und Impressionen aufzuzeichnen. Weitere Informationen finden Sie unter [Schritt 4: Empfehlungen einholen](#).

Die Kampagne ist bereit, wenn ihr Status AKTIV ist. Wenn Sie Ihre Lösungsversion neu trainieren oder Ihre Kampagneneinstellungen ändern möchten, müssen Sie Ihre Kampagne aktualisieren. Weitere Informationen finden Sie unter [Aktualisieren einer Kampagne](#).

Eine Kampagne erstellen () AWS CLI

Sobald der Status Ihrer Lösungsversion Aktiv lautet, können Sie sie mit einer Amazon Personalize bereitstellen. Verwenden Sie den folgenden `create-campaign` AWS CLI Befehl, um eine Kampagne zu erstellen, die eine Lösungsversion bereitstellt, die nach dem Rezept für Benutzerpersonalisierung trainiert wurde. Geben Sie der Kampagne einen Namen und geben Sie die Lösungsversion ARN (Amazon Resource Name) an. Ändern Sie optional `denminProvisionedTPS`, falls Ihr Anwendungsfall eine höhere bereitgestellte Kapazität erfordert. Der minimale Wert beträgt 1.

Die `campaign-config` Parameter sind spezifisch für das Rezept, mit dem Sie die Lösungsversion trainiert haben (weitere Informationen zu Rezepten finden Sie unter [Ein Rezept wählen](#)).

In diesem Beispiel werden die folgenden Rezeptspezifischen `itemExplorationConfig` Felder für die Benutzerpersonalisierung mit ihren Standardwerten verwendet: `explorationWeight` und `explorationItemAgeCutOff`. Wenn Sie den `campaign-config` Parameter weglassen, gelten die Standardwerte. Weitere Informationen zu den `itemExplorationConfig` Feldern finden Sie in der [Eigenschaften und Hyperparameter Benutzerpersonalisierung](#) Rezeptur.

Wenn Sie über einen Artikeldatensatz verfügen und die Option haben möchten, Metadaten einzubeziehen, wenn Sie Empfehlungen erhalten, aktualisieren Sie den, `campaign-config` um ein `enableMetadataWithRecommendations` Feld hinzuzufügen, und setzen Sie es auf `true`.

Eine vollständige Liste aller Parameter finden Sie unter [CreateCampaign](#).

```
aws personalize create-campaign \  
--name campaign name \  
--solution-version-arn solution version arn \  
--min-provisioned-tps 1 \  
--campaign-config "{\"itemExplorationConfig\":{\"explorationWeight\": \"0.3\",  
\"explorationItemAgeCutOff\": \"30\"}}"
```

Die Kampagne ist bereit, wenn ihr Status **AKTIV** ist. Um den aktuellen Status zu erhalten, rufen Sie [DescribeCampaign](#) auf und überprüfen Sie, ob das Feld `status` auf **ACTIVE** eingestellt ist.

Wenn Sie Ihre Lösungsversion neu trainieren oder Ihre Kampagneneinstellungen ändern möchten, müssen Sie Ihre Kampagne aktualisieren. Weitere Informationen finden Sie unter [Aktualisieren einer Kampagne](#).

Amazon Personalize bietet Funktionen zur Verwaltung von Kampagnen, z. B. [ListCampaigns](#) zum Auflisten der von Ihnen erstellten Kampagnen. Sie können eine Kampagne durch Aufrufen von [DeleteCampaign](#) löschen. Wenn Sie eine Kampagne löschen, werden die Lösungsvarianten, die Teil der Kampagne sind, nicht gelöscht.

Nachdem Sie Ihre Kampagne erstellt haben, können Sie damit Empfehlungen abgeben. Weitere Informationen finden Sie unter [Schritt 4: Empfehlungen einholen](#).

Eine Kampagne (AWSSDKs) erstellen

Sobald der Status Ihrer Lösungsversion **Aktiv** lautet, können Sie sie mit einer Amazon Personalize bereitstellen. Erstellen Sie mit dem folgenden Code eine Kampagne. Geben Sie der Kampagne einen Namen, geben Sie den Amazon-Ressourcennamen (ARN) der bereitzustellenden Lösungsversion an und geben Sie optional an, welche [Minimale Anzahl bereitgestellter TPS](#) die Kampagne unterstützt

(der Standardwert für diesen Parameter ist 1). Wenn Sie das [Benutzerpersonalisierung](#) Rezept verwenden, können Sie die Artikelsuche mit den `explorationItemAgeCutOff` Parametern `itemExplorationWeight` und konfigurieren.

Wenn Sie über einen Artikeldatensatz verfügen und die Option haben möchten, Metadaten einzubeziehen, wenn Sie Empfehlungen erhalten, aktualisieren Sie den `campaignConfig` Code, sodass er ein `enableMetadataWithRecommendations` Attribut enthält, und legen Sie es fest `true`.

Eine vollständige Liste aller Parameter finden Sie unter [CreateCampaign](#).

SDK for Python (Boto3)

In diesem Beispiel sind die `explorationItemAgeCutOff` Parameter `itemExplorationWeight` und [Benutzerpersonalisierung](#) rezeptspezifisch. Die Standardeinstellung `itemExplorationWeight` ist 0,3 und die Standardeinstellung `explorationItemAgeCutOff` ist 30. Wenn Sie die Konfigurationsparameter für Kampagnen weglassen, gelten die Standardwerte.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn',
    minProvisionedTPS = 1,
    campaignConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
"explorationItemAgeCutOff": "30"}}
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

In diesem Beispiel sind die `explorationItemAgeCutOff` Parameter `itemExplorationWeight` und [Benutzerpersonalisierung](#) rezeptspezifisch. Die

Standardeinstellung `itemExplorationWeight` ist 0,3 und die Standardeinstellung `explorationItemAge CutOff` ist 30. Wenn Sie die Konfigurationsparameter für Kampagnen weglassen, gelten die Standardwerte.

```
public static void createCampaign(PersonalizeClient personalizeClient,
    String campaignName,
    String solutionVersionArn,
    Integer minProvisionedTPS,
    String itemExplorationWeight,
    String explorationItemAgeCutOff) {

    //Optional code to instantiate a HashMap and add the explorationWeight and
    explorationItemAgeCutOff values.
    //Remove if you aren't using User-Personalization.
    Map<String,String> itemExploration = new HashMap<String,String>();
    itemExploration.put("explorationWeight", itemExplorationWeight);
    itemExploration.put("explorationItemAgeCutOff", explorationItemAgeCutOff);

    try {
        // Build a User-Personalization recipe specific campaignConfig object with
        the itemExploration map.
        // CampaignConfig construction will vary by recipe.
        CampaignConfig campaignConfig = CampaignConfig.builder()
            .itemExplorationConfig(itemExploration)
            .build();

        // build the createCampaignRequest
        CreateCampaignRequest createCampaignRequest =
        CreateCampaignRequest.builder()
            .name(campaignName)
            .solutionVersionArn(solutionVersionArn)
            .minProvisionedTPS(minProvisionedTPS)
            .campaignConfig(campaignConfig) //
            .build();

        // create the campaign
        CreateCampaignResponse campaignResponse =
        personalizeClient.createCampaign(createCampaignRequest);
        String campaignArn = campaignResponse.campaignArn();

        DescribeCampaignRequest campaignRequest = DescribeCampaignRequest.builder()
            .campaignArn(campaignArn)
            .build();
    }
}
```

```

        DescribeCampaignResponse campaignResponse =
personalizeClient.describeCampaign(campaignRequest);
        Campaign newCampaign = campaignResponse.campaign();

        System.out.println("The Campaign status is " + newCampaign.status());

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateCampaignCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the campaign parameters
export const createCampaignParam = {
    solutionVersionArn: "SOLUTION_VERSION_ARN",    /* required */
    name: "CAMPAIGN_NAME",                        /* required */
    minProvisionedTPS: 1,                          /* optional */
    campaignConfig: {
        itemExplorationConfig: {
            explorationWeight: "0.3",                /* optional */
            explorationItemAgeCutOff: "30",          /* optional */
        },
    },
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new CreateCampaignCommand(createCampaignParam)
        );
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {

```



```
    console.log("Error", err);
  }
};
run();
```

Die Kampagne ist bereit, wenn ihr Status **AKTIV** ist. Um den aktuellen Status zu erhalten, rufen Sie [DescribeCampaign](#) auf und überprüfen Sie, ob das Feld `status` auf **ACTIVE** eingestellt ist.

Wenn Sie Ihre Lösungsversion manuell neu trainieren oder Ihre Kampagneneinstellungen ändern möchten, müssen Sie Ihre Kampagne aktualisieren. Weitere Informationen finden Sie unter [Aktualisieren einer Kampagne](#).

Amazon Personalize bietet Funktionen zur Verwaltung von Kampagnen, z. B. [ListCampaigns](#) zum Auflisten der von Ihnen erstellten Kampagnen. Sie können eine Kampagne durch Aufrufen von [DeleteCampaign](#) löschen. Wenn Sie eine Kampagne löschen, werden die Lösungsvarianten, die Teil der Kampagne sind, nicht gelöscht.

Nachdem Sie Ihre Kampagne erstellt haben, können Sie damit Empfehlungen abgeben. Weitere Informationen finden Sie unter [Schritt 4: Empfehlungen einholen](#).

Aktualisieren einer Kampagne

Um eine neu trainierte Lösungsversion mit einer vorhandenen Kampagne bereitzustellen oder die Konfiguration Ihrer Kampagne [Minimale Anzahl bereitgestellter TPS](#) oder Kampagne zu ändern, z. B. Metadaten in Empfehlungen zu aktivieren, müssen Sie die Kampagne manuell aktualisieren.

Mit User-Personalization oder Next-Best-Action aktualisiert Amazon Personalize automatisch alle zwei Stunden Ihre neueste Lösungsversion (trainiert mit `trainingMode` auf **FULL**), um neue Elemente oder Aktionen in Empfehlungen aufzunehmen, und Ihre Kampagne verwendet automatisch die aktualisierte Lösungsversion. Aktualisieren Sie eine Kampagne nur dann manuell, wenn Sie die Lösungsversion manuell neu trainieren, wenn auf `trainingMode` gesetzt ist **FULL**, oder wenn Sie Änderungen an der Konfiguration Ihrer Kampagne `minProvisionedTPS` oder Kampagne vornehmen möchten. Weitere Informationen zu automatischen Updates finden Sie unter [Automatische Updates](#).

Sie aktualisieren eine Kampagne manuell mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs.

Themen

- [Aktualisieren einer Kampagne \(Konsole\)](#)

- [Aktualisieren einer Kampagne \(AWS CLI\)](#)
- [Aktualisieren einer Kampagne \(AWS SDKs\)](#)

Aktualisieren einer Kampagne (Konsole)

Um eine manuell neu trainierte Lösungsversion bereitzustellen oder Änderungen an Ihrer Kampagnenkonfiguration vorzunehmen, müssen Sie Ihre Kampagne aktualisieren.

So aktualisieren Sie eine Kampagne (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe mit der Kampagne aus, die Sie aktualisieren möchten.
3. Wählen Sie im Navigationsbereich Campaigns (Kampagnen) aus.
4. Wählen Sie auf der Seite Kampagnen die Kampagne aus, die Sie aktualisieren möchten.
5. Wählen Sie auf der Seite mit den Kampagnendetails die Option Aktualisieren aus.
6. Nehmen Sie auf der Seite Kampagne aktualisieren Ihre Änderungen vor. Wenn Sie beispielsweise eine neu trainierte Lösungsversion bereitstellen, wählen Sie für Lösungsversions-ID die Identifikationsnummer für die neue Lösungsversion aus.
7. Wählen Sie Aktualisieren. Amazon Personalize aktualisiert die Kampagne, um die neue Lösungsversion und alle geänderten Konfigurationen zu verwenden.

Aktualisieren einer Kampagne (AWS CLI)

Um eine neue Lösungsversion bereitzustellen, die Ihrer Kampagne zu ändern [Minimale Anzahl bereitgestellter TPS](#) oder die Konfiguration Ihrer Kampagne zu ändern, müssen Sie Ihre Kampagne aktualisieren. Verwenden Sie den folgenden `update-campaign` Befehl, um eine Kampagne zu aktualisieren, um eine neue Lösungsversion mit zu verwenden AWS CLI.

Ersetzen Sie durch `campaign arn` den Amazon-Ressourcennamen (ARN) der Kampagne, die Sie aktualisieren möchten. Ersetzen Sie durch `new solution version arn` die Lösungsversion, die Sie bereitstellen möchten.

```
aws personalize update-campaign \  
--campaign-arn campaign arn \  
--solution-version-arn new solution version arn \  
--min-provisioned-tps 1
```

Aktualisieren einer Kampagne (AWS SDKs)

Um eine neue Lösungsversion bereitzustellen, die Ihrer Kampagne [Minimale Anzahl bereitgestellter TPS](#) oder die Konfiguration Ihrer Kampagne zu ändern, müssen Sie Ihre Kampagne aktualisieren. Verwenden Sie den folgenden Code, um eine Kampagne mit dem SDK for Python (Boto3) oder dem SDK for Java 2.x zu aktualisieren. Eine vollständige Liste der Parameter finden Sie unter [UpdateCampaign](#).

SDK for Python (Boto3)

Verwenden Sie die folgende `update_campaign` Methode, um eine neue Lösungsversion bereitzustellen. Ersetzen Sie durch `campaign_arn` den Amazon-Ressourcennamen (ARN) der Kampagne, die Sie aktualisieren möchten, ersetzen Sie durch `new_solution_version_arn` den neuen Lösungsversions-ARN und ändern Sie optional `minProvisionedTPS`.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.update_campaign(
    campaignArn = 'campaign_arn',
    solutionVersionArn = 'new_solution_version_arn',
    minProvisionedTPS = 1,
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

Verwenden Sie die folgende `updateCampaign` Methode, um eine Kampagne zu aktualisieren, um eine neue Lösungsversion zu verwenden. Übergeben Sie als Parameter einen Amazon-Personalize-Service-Client, den Amazon-Ressourcennamen (ARN) der neuen Lösungsversion und die [Minimale Anzahl bereitgestellter TPS](#).

```
public static void updateCampaign(PersonalizeClient personalizeClient,
                                String campaignArn,
                                String solutionVersionArn,
```

```
Integer minProvisionedTPS) {

    try {
        // build the updateCampaignRequest
        UpdateCampaignRequest updateCampaignRequest =
UpdateCampaignRequest.builder()
            .campaignArn(campaignArn)
            .solutionVersionArn(solutionVersionArn)
            .minProvisionedTPS(minProvisionedTPS)
            .build();

        // update the campaign
        personalizeClient.updateCampaign(updateCampaignRequest);

        DescribeCampaignRequest campaignRequest = DescribeCampaignRequest.builder()
            .campaignArn(campaignArn)
            .build();

        DescribeCampaignResponse campaignResponse =
personalizeClient.describeCampaign(campaignRequest);
        Campaign updatedCampaign = campaignResponse.campaign();

        System.out.println("The Campaign status is " + updatedCampaign.status());

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Schritt 4: Empfehlungen einholen

Abhängig von Ihren Ressourcen können Sie Empfehlungen in Echtzeit oder mit einem Batch-Workflow erhalten.

- Mit benutzerdefinierten Ressourcen können Sie Empfehlungen oder Batch-Empfehlungen in Echtzeit erhalten. Für Empfehlungen in Echtzeit müssen Sie eine benutzerdefinierte Kampagne erstellen, bevor Sie Empfehlungen erhalten. Für Batch-Empfehlungen müssen Sie keine Kampagne erstellen.
- Mit Empfehlungen in einer Domain-Datensatzgruppe können Sie nur Empfehlungen in Echtzeit erhalten.

In den folgenden Themen wird erläutert, wie und wann die einzelnen Empfehlungstypen verwendet werden.

Themen

- [Punktzahlen für Empfehlungen](#)
- [Empfehlungen in Echtzeit erhalten](#)
- [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#)

Punktzahlen für Empfehlungen

Mit maßgeschneiderten Lösungen, die mit den Rezepten User-Personalization, Personalized-Ranking und PERSONALIZED_ACTIONS erstellt wurden, berücksichtigt Amazon Personalize in Empfehlungen für jeden Artikel eine Bewertung. Diese Werte stellen die relative Sicherheit dar, die Amazon Personalize darüber hat, welchen Artikel oder welche Aktion der Benutzer als Nächstes auswählen wird. Höhere Punktzahlen bedeuten eine größere Gewissheit.

- Informationen zu Punktzahlen für die Benutzerpersonalisierung finden Sie unter. [So funktioniert die Bewertung von Empfehlungen zur Benutzerpersonalisierung \(benutzerdefinierte Ressourcen\)](#)
- Informationen zu Punktzahlen für PERSONALIZED_ACTIONS-Rezepte finden Sie unter. [So funktioniert die Bewertung von Handlungsempfehlungen](#)
- Informationen zu Punktzahlen für Empfehlungen mit personalisiertem Ranking finden Sie unter. [So funktioniert die personalisierte Bewertung von Rankings](#)

Bei Batch-Inferenzjobs werden die Artikelwerte wie unter und beschrieben berechnet. [So funktioniert die Bewertung von Empfehlungen zur Benutzerpersonalisierung \(benutzerdefinierte Ressourcen\)](#) [So funktioniert die personalisierte Bewertung von Rankings](#) Sie können die Ergebnisse in der JSON-Ausgabedatei des Batch-Inferenzjobs anzeigen.

Empfehlungen in Echtzeit erhalten

Empfehlungen in Echtzeit sind Empfehlungen, die Sie anfordern und die Sie Ihren Benutzern zeigen, während sie Ihre Anwendung verwenden. Sie können Empfehlungen von Amazon Personalize in Echtzeit mit einer Empfehlung (für Domain-Datensatzgruppen) oder einer benutzerdefinierten Kampagne erhalten.

- Für Domain-Empfehlungen können Sie während des Vorgangs Empfehlungen in Echtzeit erhalten. [the section called “GetRecommendations”](#) Oder Sie können Ihren Empfehlungsgeber mit der Amazon Personalize-Konsole testen.
- Bei benutzerdefinierten Ressourcen erhalten Sie je nach dem Rezept, das Sie zur Erstellung der Lösungsversion verwendet haben, auf der die Kampagne basiert, Empfehlungen für Ihre Benutzer mit den [the section called “GetRecommendations”](#) [the section called “GetPersonalizedRanking”](#) API-Operationen [GetActionRecommendations](#), oder. Oder Sie können Ihre Kampagne mit der Amazon Personalize-Konsole testen.

Wenn Sie Domain-Anwendungsfälle oder Rezepte verwenden, die eine Personalisierung in Echtzeit ermöglichen, wie z. B. den Anwendungsfall Top-Picks für Sie oder das Rezept für Benutzerpersonalisierung, aktualisiert Amazon Personalize die Empfehlungen auf der Grundlage der letzten Aktivitäten Ihrer Benutzer, während Sie deren Interaktionen mit Ihrem Katalog aufzeichnen. Weitere Informationen zur Aufzeichnung von Ereignissen in Echtzeit und zur Personalisierung finden Sie unter. [Ereignisse aufzeichnen](#)

Wenn Sie Ihre Kampagne so konfiguriert haben, dass Metadaten für empfohlene Artikel zurückgegeben werden, können Sie die Spalten angeben, die in Ihren [GetRecommendations](#) oder [GetPersonalizedRanking](#) API-Vorgängen enthalten sein sollen. Sie können die Spalten auch angeben, wenn Sie die Kampagne mit der Amazon Personalize-Konsole testen.

Für einige Anwendungsfälle und Rezepte können Sie in Ihrer Anfrage eine Werbeaktion angeben. Eine Werbeaktion definiert zusätzliche Geschäftsregeln, die für eine konfigurierbare Teilmenge empfohlener Artikel gelten. Weitere Informationen finden Sie unter [Werbung für Artikel in Empfehlungen](#).

Themen

- [Abrufen von Elementempfehlungen](#)
- [Handlungsempfehlungen einholen](#)
- [Ein personalisiertes Ranking erhalten \(benutzerdefinierte Ressourcen\)](#)
- [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#)

Abrufen von Elementempfehlungen

Sie können Articlempfehlungen von einem Amazon-Personalize-Empfehlungsgeber oder einer benutzerdefinierten Kampagne mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs erhalten.

Note

Wenn Sie ein benutzerdefiniertes PERSONALIZED_RANKING-Rezept verwendet haben, finden Sie weitere Informationen unter [Ein personalisiertes Ranking erhalten \(benutzerdefinierte Ressourcen\)](#).

Themen

- [So funktioniert die Bewertung von Empfehlungen zur Benutzerpersonalisierung \(benutzerdefinierte Ressourcen\)](#)
- [Abrufen von Elementempfehlungen \(Konsole\)](#)
- [Abrufen von Elementempfehlungen \(AWS CLI\)](#)
- [Abrufen von Elementempfehlungen \(AWS SDKs\)](#)
- [Werbung für Artikel in Empfehlungen](#)

So funktioniert die Bewertung von Empfehlungen zur Benutzerpersonalisierung (benutzerdefinierte Ressourcen)

Mit dem User-Personalization-Rezept generiert Amazon Personalize Ergebnisse für Elemente, die auf den Interaktionsdaten und Metadaten eines Benutzers basieren. Diese Werte stellen die relative Sicherheit dar, die Amazon Personalize bei der Frage hat, ob der Benutzer als Nächstes mit dem Element interagieren wird. Höhere Punktzahlen bedeuten eine größere Gewissheit.

Amazon Personalize bewertet alle Elemente in Ihrem Katalog relativ zueinander auf einer Skala von 0 bis 1 (beide inklusive), sodass die Summe aller Punktzahlen 1 entspricht. Wenn Sie beispielsweise Filmempfehlungen für einen Benutzer abrufen und im Artikel-Dataset drei Filme enthalten sind, könnten deren Punktzahlen 0.6, 0.3 und 0.1 lauten. Wenn sich in Ihrem Bestand 1.000 Filme befinden, besitzen die Filme mit der höchsten Bewertung dementsprechend möglicherweise eine sehr geringe Punktzahl (die durchschnittliche Punktzahl wäre .001), aber da die Bewertung relativ ist, sind die Empfehlungen weiterhin gültig.

Mathematisch gesehen werden die Ergebnisse für jedes Benutzer-Element-Paar (u, i) nach der folgenden Formel berechnet, wobei \exp die exponentielle Funktion, w_u bzw. $w_{i,j}$ Benutzer- und Elementeinbettungen sind und das Erscheinungsbuchstabe Sigma (Σ) die Summierung aller Elemente im Elementdatensatz darstellt:

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_j \exp(\bar{w}_u^\top w_j)}$$

Note

Amazon Personalize zeigt keine Ergebnisse für Domain-Empfehlungen oder die Rezepte Ähnliche Elemente, SIMS oder Popularity-Count an. Weitere Informationen zu Punktzahlen für Empfehlungen für Personalized-Ranking finden Sie unter [So funktioniert die personalisierte Bewertung von Rankings](#).

Abrufen von Elementempfehlungen (Konsole)

Um Empfehlungen mit der Amazon-Personalize-Konsole zu erhalten, geben Sie die Anforderungsinformationen auf der Detailseite eines Empfehlens (Domain-Datensatzgruppe) oder einer benutzerdefinierten Kampagne an.

So erhalten Sie Empfehlungen

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe aus, die die Kampagne oder den Empfehlungsgeber enthält, den Sie verwenden.
3. Wählen Sie im Navigationsbereich Kampagnen oder Empfehlungen aus.
4. Wählen Sie die Zielkampagne oder den Empfehlungsgeber aus.
5. Geben Sie für eine Kampagne unter Ergebnisse der Testkampagne Ihre Empfehlungsanforderungsdetails basierend auf dem von Ihnen verwendeten Rezept ein. Wählen Sie als Empfehler die Option Empfehlung testen aus und geben Sie Ihre Empfehlungsanforderungsdetails basierend auf Ihrem Anwendungsfall ein.

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` aus diesen Ereignissen angeben, als wäre es sein `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

6. Wählen Sie optional einen Filter aus. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).
7. Wenn Sie kontextbezogene Metadaten verwenden, geben Sie Daten für jeden Kontext an. Geben Sie für jeden Kontext für den Schlüssel das Metadatenfeld ein. Geben Sie für Wert die Kontextdaten ein. Weitere Informationen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#).
8. Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne oder Ihren Empfehlungsgeber aktiviert haben, wählen Sie für Elemente-Datensatzspalten die Metadatenfelder aus, die Sie in Empfehlungsergebnisse aufnehmen möchten. Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Metadaten in Empfehlungen aktivieren](#). Informationen zum Aktivieren von Metadaten für einen Empfehler finden Sie unter [Aktivieren von Metadaten in Empfehlungen](#).
9. Wenn Sie eine Teilmenge von Elementen hochstufen möchten, füllen Sie optional die Felder Hochstufung aus. Weitere Informationen finden Sie unter [Werbung für Artikel in Empfehlungen](#).
10. Wählen Sie `Get recommendations` (Empfehlungen erhalten). Eine Tabelle mit den 25 empfohlenen Top-Elementen des Benutzers wird angezeigt.

Abrufen von Elementempfehlungen (AWS CLI)

Die folgenden Codebeispiele zeigen verschiedene Varianten dafür, wie Sie Artikelempfehlungen mit der erhalten AWS CLI.

Themen

- [Abrufen von Elementempfehlungen](#)
- [Einschließen von Elementmetadaten in Empfehlungen](#)

Abrufen von Elementempfehlungen

Verwenden Sie den folgenden Code, um Empfehlungen aus einer Kampagne zu erhalten. Um Empfehlungen von einem Empfehler zu erhalten, ersetzen Sie den `campaign-arn` Parameter durch die `recommender-arn`.

Geben Sie die ID des Benutzers an, für den Sie Empfehlungen erhalten möchten, und den Amazon-Ressourcennamen (ARN) Ihrer Kampagne oder Ihres Empfehlers. Eine Liste der 10 wichtigsten empfohlenen Elemente für den Benutzer wird angezeigt. Um die Anzahl der empfohlenen Elemente zu ändern, ändern Sie den Wert für `numResults`. Der Standardwert ist 25 Elemente. Das Maximum beträgt 500 Elemente. Wenn Sie ein `RELATED_ITEMS`-Rezept verwendet haben, um die Lösungsversion zu trainieren, die die Kampagne unterstützt, ersetzen Sie den `user-id` Parameter durch `item-id` und geben Sie die Element-ID an.

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` aus diesen Ereignissen angeben, als wäre es sein `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

```
aws personalize-runtime get-recommendations \  
--campaign-arn campaign arn \  
--user-id User ID \  
--num-results 10
```

Einschließen von Elementmetadaten in Empfehlungen

Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne oder Ihren Empfehler aktiviert haben, können Sie die Metadaten spalten für den Elementdatensatz angeben, die in die Antwort aufgenommen werden sollen. Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Metadaten in Empfehlungen aktivieren](#). Informationen zum Aktivieren von Metadaten für einen Empfehler finden Sie unter [Aktivieren von Metadaten in Empfehlungen](#).

Das folgende Codebeispiel zeigt, wie Sie die Metadaten spalten als Teil Ihrer Anforderung von Empfehlungen angeben.

```
aws personalize-runtime get-recommendations \  
--campaign-arn campaign arn \  
--user-id User ID \  
--num-results 10
```

```
--num-results 10 \  
--metadata-columns "{\"ITEMS\": [\"coLumnNameA\", \"coLumnNameB\"]}"
```

Abrufen von Elementempfehlungen (AWS SDKs)

Die folgenden Codebeispiele zeigen verschiedene Varianten dafür, wie Sie Elementempfehlungen mit den - AWS SDKs erhalten.

Themen

- [Abrufen von Elementempfehlungen](#)
- [Einschließen von Elementmetadaten in Empfehlungen](#)

Abrufen von Elementempfehlungen

Der folgende Code zeigt, wie Sie Amazon-Personalize-Empfehlungen für einen Benutzer aus einer Kampagne abrufen. Um Empfehlungen von einem Empfehler zu erhalten, ersetzen Sie den `campaignArn` Parameter durch die `recommenderArn`.

Geben Sie die ID des Benutzers an, für den Sie Empfehlungen erhalten möchten, und den Amazon-Ressourcennamen (ARN) Ihrer Kampagne oder Ihres Empfehlers. Eine Liste der 10 wichtigsten empfohlenen Elemente für den Benutzer wird angezeigt. Um die Anzahl der empfohlenen Elemente zu ändern, ändern Sie den Wert für `numResults`. Der Standardwert ist 25 Elemente. Das Maximum beträgt 500 Elemente. Wenn Sie ein `RELATED_ITEMS`-Rezept verwendet haben, um die Lösungsversion zu trainieren, die die Kampagne unterstützt, ersetzen Sie den `userId` Parameter durch `itemId` und geben Sie die Element-ID an.

Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne oder Ihren Empfehler aktiviert haben, können Sie die Metadaten spalten für den Elementdatensatz angeben, die in die Antwort aufgenommen werden sollen. Ein Codebeispiel finden Sie unter [Einschließen von Elementmetadaten in Empfehlungen](#). Informationen zum Aktivieren von Metadaten finden Sie unter [Metadaten in Empfehlungen aktivieren](#).

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` aus diesen Ereignissen angeben, als wäre es sein `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

SDK for Java 2.x

```
public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
String campaignArn, String userId) {

    try {
        GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
            .campaignArn(campaignArn)
            .numResults(20)
            .userId(userId)
            .build();

        GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
            .getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();
        for (PredictedItem item : items) {
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
        }

    } catch (AwsServiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
  "@aws-sdk/client-personalize-runtime";

import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
  "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: 'CAMPAIGN_ARN', /* required */
  userId: 'USER_ID',          /* required */
  numResults: 15             /* optional */
}

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
  GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Einschließen von Elementmetadaten in Empfehlungen

Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne oder Ihren Empfehler aktiviert haben, können Sie die Metadaten­spalten für den Elementdatensatz angeben, die in die Antwort aufgenommen werden sollen. Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Metadaten in Empfehlungen aktivieren](#). Informationen zum Aktivieren von Metadaten für einen Empfehler finden Sie unter [Aktivieren von Metadaten in Empfehlungen](#).

Das folgende Codebeispiel zeigt, wie Sie die Metadaten­spalten als Teil Ihrer Anforderung von Empfehlungen angeben.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
    metadataColumns = {
        "ITEMS": ['columnNameA', 'columnNameB']
    }
)

print("Recommended items")
for item in response['itemList']:
    print(item['itemId'])
    print(item['metadata'])
```

Werbung für Artikel in Empfehlungen

Bei allen Domain-Anwendungsfällen und einigen benutzerdefinierten Rezepten können Sie eine Werbeaktion angeben, wenn Sie Empfehlungen erhalten. Eine Werbeaktion definiert zusätzliche Geschäftsregeln, die für eine konfigurierbare Teilmenge empfohlener Artikel gelten. Sie haben beispielsweise eine Streaming-App und möchten Ihre eigenen Sendungen und Filme bewerben, aber auch relevante Titel empfehlen. Sie könnten eine Werbeaktion verwenden, um festzulegen, dass ein bestimmter Prozentsatz der empfohlenen Artikel aus der Kategorie intern stammen muss. Bei den übrigen empfohlenen Artikeln handelt es sich weiterhin um relevante Empfehlungen, die auf Ihrem Rezept und allen Anforderungsfiltren basieren.

Um eine Werbeaktion in Anspruch zu nehmen, geben Sie in Ihrer Empfehlungsanfrage Folgendes an:

- Der Prozentsatz der empfohlenen Artikel, auf die der Werbefilter angewendet werden soll.
- Ein Filter, der die Beförderungskriterien festlegt. Weitere Informationen finden Sie unter [Werbefilter](#).

In der Antwort auf die Empfehlung werden beworbene Artikel nach dem Zufallsprinzip relativ zu anderen empfohlenen Artikeln positioniert, jedoch in sortierter Reihenfolge relativ zu anderen beworbenen Artikeln. Je nach Rezept werden empfohlene Artikel, die nicht Teil einer Werbeaktion sind, nach Relevanz für den Nutzer, Beliebtheit oder Ähnlichkeit sortiert. Wenn nicht genügend Artikel die Werbekriterien erfüllen, enthält das Ergebnis so viele beworbene Artikel wie möglich.

Mit der Amazon Personalize-Konsole AWS Command Line Interface (AWS CLI) oder den AWS SDKs können Sie eine Werbeaktion auf Empfehlungen anwenden.

Themen

- [Anwendungsfälle und Rezepte zur Unterstützung von Werbeaktionen](#)
- [Werbefilter](#)
- [Artikel bewerben \(Konsole\)](#)
- [Werbeartikel \(AWS CLI\)](#)
- [Werbung für Artikel \(AWSSDKs\)](#)

Anwendungsfälle und Rezepte zur Unterstützung von Werbeaktionen

Alle Anwendungsfälle unterstützen Werbeaktionen. Die folgenden benutzerdefinierten Rezepte unterstützen Werbeaktionen:

- [USER_PERSONALIZATION](#)-Rezepte
- [RELATED_ITEMS](#)-Rezepte
- [BELIEBTE_ARTIKEL](#)-Rezepte

Werbefilter

Wenn Sie eine Beförderung auf eine Empfehlungsanfrage anwenden, wählen Sie einen Filter, der die Beförderungskriterien festlegt. Sie können einen vorhandenen Filter verwenden oder einen neuen erstellen. Sie erstellen und verwalten Filter für Werbeaktionen wie andere Filter in Amazon Personalize. Hinweise zum Erstellen und Verwalten von Filtern finden Sie unter [Filtern von Ergebnissen](#).

Der einzige Unterschied zwischen einem Werbefilter und einem Filter, den Sie außerhalb der Werbeaktion auswählen (dem Anforderungsfilter), besteht darin, wie Amazon Personalize sie anwendet. Ein Werbefilter gilt nur für beworbene Artikel, während ein Anforderungsfilter nur für die verbleibenden empfohlenen Artikel gilt. Wenn Sie einen Anforderungsfilter und einen Werbefilter angeben und beide Filter auf beworbene Artikel anwenden möchten, muss der Ausdruck Ihres Werbefilters beide Ausdrücke enthalten. Die Art und Weise, wie Sie zwei Ausdrücke kombinieren, hängt von den verwendeten Datensätzen ab. Weitere Informationen zu Filterausdrücken, ihren Regeln und deren Erstellung finden Sie unter [Filterausdrücke](#).

Beispiele für Filterausdrücke

Der folgende Ausdruck umfasst nur Artikel aus der Kategorie „Inhouse“. Sie können diesen Ausdruck verwenden, wenn Sie in Ihren Empfehlungen für Ihre eigenen Inhalte werben möchten.

```
INCLUDE ItemID WHERE Items.OWNER IN ("in-house")
```

Der folgende Ausdruck umfasst nur Elemente, die vor einem von Ihnen angegebenen Zeitstempel erstellt wurden. Sie können diesen Ausdruck verwenden, um kürzlich erstellte Artikel zu bewerben.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

Der folgende Ausdruck zeigt, wie Sie einen Anforderungsfilter auf beworbene Artikel anwenden können. Es beinhaltet nur verfügbare Kleidungsstücke als Werbeartikel. In diesem Szenario `Items.AVAILABLE IN ("True")` würde der auch im Anforderungsfilterausdruck verwendet, sodass sich alle Empfehlungen auf Elemente beziehen, die verfügbar sind.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") AND Items.AVAILABLE IN ("True")
```

Eine vollständigere Liste von Filterbeispielen finden Sie unter [Beispiele für Filterausdrücke](#).

Artikel bewerben (Konsole)

Um bestimmte Artikel in Empfehlungen mit der Amazon Personalize-Konsole zu bewerben, erstellen Sie einen Filter und geben Sie dann die Werbedetails in der Empfehlungsanfrage an. Informationen zu anderen Feldern finden Sie unter [Abrufen von Elementempfehlungen \(Konsole\)](#).

Um Artikel in Empfehlungen zu bewerben

1. Öffnen Sie die Amazon Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe aus, die die von Ihnen verwendete Kampagne oder Empfehlung enthält.
3. Falls Sie dies noch nicht getan haben, erstellen Sie einen Filter, der die Beförderungskriterien festlegt. Sie erstellen Filter für Werbeaktionen auf die gleiche Weise wie Sie Anforderungsfilter erstellen. Informationen zum Erstellen und Verwalten von Filtern finden Sie unter [Filtern von Ergebnissen](#).
4. Wählen Sie im Navigationsbereich Empfehlungen oder Kampagnen aus.
5. Wählen Sie die Zielkampagne oder den Empfehlungsgeber aus.

6. Geben Sie für Kampagnen unter Kampagnenergebnisse die Details Ihrer Empfehlungsanfrage ein, die auf dem von Ihnen verwendeten Rezept basieren. Wählen Sie für Empfehlungsgeber die Option Empfehlungsgeber testen und geben Sie die Details Ihrer Empfehlungsanfrage ein.
7. Wählen Sie optional einen Filter für die Anfrage. Dieser Filter gilt nur für Artikel, die nicht beworben wurden. Informationen zum Erstellen und Verwalten von Filtern finden Sie unter [Filtern von Ergebnissen](#).
8. Wenn Sie kontextuelle Metadaten verwenden, geben Sie Daten für jeden Kontext an. Geben Sie für jeden Kontext für den Schlüssel das Metadatenfeld ein. Geben Sie für den Wert die Kontextdaten ein. Weitere Informationen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#).
9. Geben Sie für Promotion Folgendes an:
 - Prozentualer Anteil der beworbenen Artikel: Geben Sie den Prozentsatz der empfohlenen Artikel ein, auf die die Werbeaktion angewendet werden soll.
 - Filter: Wählen Sie einen Filter, der die Beförderungskriterien festlegt. Dieser Filter gilt für die beworbenen Artikel und nicht für jeden Anforderungsfilter, den Sie möglicherweise in Schritt 7 angegeben haben.
 - Filterparameter: Wenn Ihre Werbeaktion einen Filter mit Platzhalterparametern verwendet, geben Sie für jeden Parameter den Wert ein, um die Filterkriterien festzulegen. Um mehrere Werte für einen Parameter zu verwenden, trennen Sie jeden Wert durch ein Komma.
10. Wählen Sie Get recommendations (Empfehlungen erhalten). Eine Tabelle mit den 25 wichtigsten empfohlenen Artikeln des Benutzers wird angezeigt. In der Spalte Gesponserter Artikel wird angezeigt, ob der Artikel aufgrund Ihrer Werbeaktion aufgenommen wurde. Beworbene Artikel werden nach dem Zufallsprinzip relativ zu anderen empfohlenen Artikeln positioniert, jedoch in sortierter Reihenfolge relativ zu anderen beworbenen Artikeln. Je nach Anwendungsfall oder Rezept werden empfohlene Artikel, die nicht Teil einer Werbeaktion sind, nach Relevanz für den Nutzer, Beliebtheit oder Ähnlichkeit sortiert. Wenn nicht genügend Artikel die Werbekriterien erfüllen, enthält das Ergebnis so viele beworbene Artikel wie möglich.

Werbeartikel (AWS CLI)

Der folgende Code zeigt, wie Artikel in Empfehlungen mit der AWS CLI und einer benutzerdefinierten Kampagne beworben werden. Um Artikel mit einem Empfehlungsgeber zu bewerben, ersetzen Sie den `campaign-arn` Parameter durch einen `recommender-arn` und geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber an. Geben Sie für die Werbefelder Folgendes an:

- **Name:** Geben Sie der Werbeaktion einen Namen. In der Antwort auf die Empfehlung wird der Name verwendet, um beworbene Artikel zu identifizieren.
- **percent-promoted-items:** Der Prozentsatz der empfohlenen Artikel, auf die die Werbeaktion angewendet werden soll. In diesem Beispiel werden 50% der Artikel beworben.
- **filterARN:** Geben Sie den Amazon-Ressourcennamen (ARN) des Filters an, der die Beförderungskriterien definiert. Weitere Informationen finden Sie unter [Werbefilter](#).
- **Parameternamen und Werte:** Wenn Ihr Filterausdruck irgendeinen Parameter hat, geben Sie die Parameternamen (Groß- und Kleinschreibung beachten) und die Werte an. Wenn Ihr Filterausdruck beispielsweise über einen \$GENRE Parameter verfügt, geben Sie GENRE als Schlüssel und ein oder mehrere Genres wie Comedy als Wert an. Trennen Sie mehrere Werte durch ein Komma. Wenn Sie das verwenden AWS CLI, müssen Sie für jeden Wert das / Zeichen verwenden, um sowohl Anführungszeichen als auch das / Zeichen zu maskieren. Das folgende Codebeispiel zeigt, wie die Werte formatiert werden.

Der Code zeigt, wie Sie sowohl einen Anforderungsfilter als auch einen Werbefilter verwenden. Ein Werbefilter gilt nur für beworbene Artikel, während ein Anforderungsfilter nur für die verbleibenden empfohlenen Artikel gilt. Weitere Informationen finden Sie unter [Werbefilter](#).

Hinweise zu zusätzlichen Feldern finden Sie unter [Abrufen von Elementempfehlungen \(AWS SDKs\)](#) und [Abrufen eines personalisierten Rankings mithilfe von kontextuellen Metadaten](#).

```
aws personalize-runtime get-recommendations \
--campaign-arn CampaignArn \
--user-id 1 \
--num-results 10 \
--filter-arn RequestFilterArn \
--filter-values '{
  "RequestFilterParameterName": "\"value\\"",
  "RequestFilterParameterName": "\"value1\",\"value2\",\"value3\\""
}' \
--promotions "[{
  \"name\": \"promotionName\",
  \"percentPromotedItems\": 50,
  \"filterArn\": \"PromotionFilterARN\",
  \"filterValues\": {\"PromotionParameterName\":\"\\\"value1, value2\\\"\"}
}]"
```

Eine Liste der empfohlenen Artikel wird angezeigt. Beworbene Artikel werden nach dem Zufallsprinzip relativ zu anderen empfohlenen Artikeln positioniert, jedoch in sortierter Reihenfolge relativ zu

anderen beworbenen Artikeln. Je nach Rezept werden empfohlene Artikel, die nicht Teil einer Werbeaktion sind, nach Relevanz für den Nutzer, Beliebtheit oder Ähnlichkeit sortiert. Wenn nicht genügend Artikel die Werbekriterien erfüllen, enthält das Ergebnis so viele beworbene Artikel wie möglich.

```
{
  "itemList": [
    {
      "itemId1": "123",
      "score": .0117211,
      "promotionName": "promotionName"
    },
    {
      "itemId2": "456",
      "score": .0077976
    },
    {
      "itemId3": "789",
      "score": .0067171
    },
    .....
  ]
}
```

Werbung für Artikel (AWSSDKs)

Der folgende Code zeigt, wie Elemente in Empfehlungen mit dem SDK für Python (Boto3) und dem SDK für Java 2.x sowie einer benutzerdefinierten Kampagne beworben werden. Um Artikel mit einem Empfehlungsgeber zu bewerben, ersetzen Sie den `campaignArn` Parameter durch `recommenderArn` und geben Sie den Amazon-Ressourcennamen (ARN) für den Empfehlungsgeber an. Geben Sie für die Werbefelder Folgendes an:

- **Name:** Geben Sie den Namen der Werbeaktion an. Die Antwort auf die Empfehlung enthält den Namen, um beworbene Artikel zu identifizieren.
- **percentPromotedItems:** Der Prozentsatz der empfohlenen Artikel, auf die die Werbeaktion angewendet werden soll.
- **promotionFilterArn:** Der Amazon-Ressourcenname (ARN) des Filters, der die Beförderungskriterien definiert. Weitere Informationen finden Sie unter [Werbefilter](#).
- **Beliebige Parameternamen und Werte:** Wenn Ihr Filterausdruck irgendwelche Parameter hat, geben Sie für jeden Parameter in Ihrem Filterausdruck den Parameternamen (Groß- und Kleinschreibung beachten) und die Werte an. Wenn Ihr Filterausdruck beispielsweise über einen

\$GENRE Parameter verfügt, geben Sie "GENRE" als Schlüssel und ein oder mehrere Genres wie „\“ Komödie "\“ als Wert an. Trennen Sie mehrere Werte durch ein Komma. Zum Beispiel "\“ comedy \",\"drama\", \"horror\"\".

Der folgende Code zeigt, wie Sie sowohl einen Anforderungsfilter als auch einen Werbefilter verwenden. Ein Werbefilter gilt nur für beworbene Artikel, während ein Anforderungsfilter nur für die verbleibenden empfohlenen Artikel gilt. Weitere Informationen finden Sie unter [Werbefilter](#).

Hinweise zu zusätzlichen Feldern finden Sie unter [Abrufen von Elementempfehlungen \(AWS SDKs\)](#) und [Abrufen eines personalisierten Rankings mithilfe von kontextuellen Metadaten](#).

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = "CampaignARN",
    userId = '1',
    numResults = 10,
    filterArn = 'RequestFilterARN',
    filterValues = {
        "RequestFilterParameterName": "\"value1\"",
        "RequestFilterParameterName": "\"value1\", \"value2\", \"value3\""
        ....
    },
    promotions = [{
        "name" : "promotionName",
        "percentPromotedItems" : 50,
        "filterArn": "promotionFilterARN",
        "filterValues": {
            "PromotionParameterName": "\"Value1\", \"Value2\""
            ...
        }
    }]
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
    if ("promotionName" in item):
```

```
print(item['promotionName'])
```

SDK for Java 2.x

```
public static void getRecommendationsWithPromotedItems(PersonalizeRuntimeClient
personalizeRuntimeClient,
                String campaignArn,
                String userId,
                String requestFilterArn,
                String requestParameterName,
                String requestParameterValue1,
                String requestParameterValue2,
                String promotionName,
                int percentPromotedItems,
                String promotionFilterArn,
                String promotionParameterName,
                String promotionParameterValue1,
                String promotionParameterValue2) {

    try {

        Map<String, String> promotionFilterValues = new HashMap<>();

        promotionFilterValues.put(promotionParameterName, String.format("\'%1$s\'",
        \\'%2$s\'",
            promotionParameterValue1, promotionParameterValue2));

        Promotion newPromotion = Promotion.builder()
            .name(promotionName)
            .percentPromotedItems(percentPromotedItems)
            .filterArn(promotionFilterArn)
            .filterValues(promotionFilterValues)
            .build();

        List<Promotion> promotionList = new List<>();

        promotionsList.add(newPromotion);

        Map<String, String> requestfilterValues = new HashMap<>();

        requestfilterValues.put(requestParameterName, String.format("\'%1$s\'", \\'%2$s
        \\'",
            requestParameterValue1, requestParameterValue2));
```

```

    GetRecommendationsRequest recommendationsRequest =
    GetRecommendationsRequest.builder()
        .campaignArn(campaignArn)
        .numResults(20)
        .userId(userId)
        .filterArn(requestFilterArn)
        .filterValues(requestFilterValues)
        .promotions(promotionList)
        .build();

    GetRecommendationsResponse recommendationsResponse =
    personalizeRuntimeClient.getRecommendations(recommendationsRequest);
    List<PredictedItem> items = recommendationsResponse.itemList();

    for (PredictedItem item: items) {
        System.out.println("Item Id is : "+item.itemId());
        System.out.println("Item score is : "+item.score());
        System.out.println("Promotion name is : "+item.promotionName());
    }
} catch (PersonalizeRuntimeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { GetRecommendationsCommand, PersonalizeRuntimeClient } from
    "@aws-sdk/client-personalize-runtime";

// create personalizeRuntimeClient.
const personalizeRuntimeClient = new PersonalizeRuntimeClient({
    region: "REGION",
});

// set recommendation request param
export const getRecommendationsParam = {
    campaignArn: "CAMPAIGN_ARN", /* required */
    userId: "USER_ID", /* required */
    numResults: 25, /* optional */
    filterArn: "FILTER_ARN", /* provide if you are applying a custom filter */
}

```

```

    filterValues: {
      "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your filter has a placeholder
parameter */
    },
    promotions: [
      {
        name: "PROMOTION_NAME", /* specify the name of the promotion. The
recommendation response includes the name to identify promoted items. */
        percentPromotedItems: 50, /* the percentage of recommended items to apply the
promotion to. */
        filterArn:
          "PROMOTION_FILTER_ARN", /* the Amazon Resource Name (ARN) of the filter that
defines the promotion criteria. */
        filterValues: {
          "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your promotion filter has a
placeholder parameter */
        },
      },
    ],
  ],
};

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", "\nItems are: ");
    response.itemList.forEach(element => console.log(element.itemId))
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

Eine Liste der empfohlenen Artikel wird angezeigt. Beworbene Artikel werden nach dem Zufallsprinzip relativ zu anderen empfohlenen Artikeln positioniert, jedoch in sortierter Reihenfolge relativ zu anderen beworbenen Artikeln. Je nach Rezept werden empfohlene Artikel, die nicht Teil einer Werbeaktion sind, nach Relevanz für den Nutzer, Beliebtheit oder Ähnlichkeit sortiert. Wenn nicht genügend Artikel die Werbekriterien erfüllen, enthält das Ergebnis so viele beworbene Artikel wie möglich.

```
{
  "itemList": [
    {
      "itemId1": "123",
      "score": .0117211,
      "promotionName": "promotionName"
    },
    {
      "itemId2": "456",
      "score": .0077976
    },
    {
      "itemId3": "789",
      "score": .0067171
    },
    .....
  ]
}
```

Handlungsempfehlungen einholen

Wenn Sie ein Rezept für `PERSONALIZED_ACTIONS` verwenden, können Sie in Echtzeit Handlungsempfehlungen aus Ihrer Kampagne abrufen. Sie können Handlungsempfehlungen mit der Amazon Personalize Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs abrufen.

Themen

- [So funktioniert die Bewertung von Handlungsempfehlungen](#)
- [Handlungsempfehlungen abrufen \(Konsole\)](#)
- [Handlungsempfehlungen werden abgerufen \(AWS CLI\)](#)
- [Abrufen von Handlungsempfehlungen \(AWSSDKs\)](#)

So funktioniert die Bewertung von Handlungsempfehlungen

Mit dem Rezept „Next-Best-Action“ generiert Amazon Personalize Punktzahlen für Aktionen, die auf der Wahrscheinlichkeit basieren, dass der Benutzer mit der Aktion interagiert. Die Punktzahlen können zwischen 0 und 1,0 liegen. Je näher 1,0, desto wahrscheinlicher ist es, dass der Benutzer mit der Aktion interagiert.

Wenn Sie keine Aktionsinteraktionsdaten importiert haben, haben alle empfohlenen Aktionen einen Wert von 0,0. Wenn Amazon Personalize im Rahmen der Erkundung eine Aktion empfiehlt, hat der Artikel eine Bewertung von 0,0. Amazon Personalize verwendet Exploration, um Aktionen ohne Aktionsinteraktionsdaten zu empfehlen. Weitere Informationen zur Erkundung finden Sie unter [Exploration \(Erkundung\)](#).

Handlungsempfehlungen abrufen (Konsole)

Um Handlungsempfehlungen mit der Amazon Personalize-Konsole zu erhalten, geben Sie die Anforderungsinformationen auf der Detailseite Ihrer benutzerdefinierten Kampagne an.

Um Handlungsempfehlungen zu erhalten

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe aus, die die Kampagne enthält, die Sie verwenden.
3. Wählen Sie im Navigationsbereich unter Benutzerdefinierte Ressourcen die Option Kampagnen aus.
4. Wählen Sie die Zielkampagne aus.
5. Geben Sie unter Ergebnisse der Testkampagne die Details Ihrer Empfehlungsanfrage ein.

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` Ereignisse so angeben, als ob es seine eigenen wären `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

6. Wählen Sie optional einen Filter aus. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).
7. Wählen Sie Get recommendations (Empfehlungen erhalten). Eine Tabelle mit den fünf wichtigsten empfohlenen Aktionen des Benutzers wird angezeigt.

Handlungsempfehlungen werden abgerufen (AWS CLI)

Verwenden Sie den folgenden Code, um Handlungsempfehlungen aus einer Kampagne zu erhalten. Geben Sie die ID des Benutzers an, für den Sie Empfehlungen erhalten möchten, und den Amazon-Ressourcennamen (ARN) Ihrer Kampagne.

Um die Anzahl der empfohlenen Aktionen zu ändern, ändern Sie den Wert für `numResults`. Die Standardeinstellung ist 5 Aktionen. Das Maximum sind 100 Aktionen.

Um Aktionsempfehlungen nach benutzerdefinierten Kriterien zu filtern, können Sie einen Filter erstellen und ihn auf den `get-action-recommendations` Vorgang anwenden. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` Ereignisse so angeben, als ob es sich um seine eigenen Ereignisse handelt `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

```
aws personalize-runtime get-action-recommendations \  
--campaign-arn campaign arn \  
--user-id User ID \  
--num-results 10
```

Abrufen von Handlungsempfehlungen (AWSSDKs)

Der folgende Code zeigt, wie Sie Amazon Personalize für einen Benutzer aus einer Kampagne abrufen können. Geben Sie die ID des Benutzers an, für den Sie Empfehlungen erhalten möchten, und den Amazon-Ressourcennamen (ARN) Ihrer Kampagne.

Um die Anzahl der empfohlenen Aktionen zu ändern, ändern Sie den Wert für `numResults`. Die Standardeinstellung ist 5 Aktionen. Das Maximum sind 100 Aktionen.

Um Handlungsempfehlungen nach benutzerdefinierten Kriterien zu filtern, können Sie einen Filter erstellen und ihn auf die [GetActionRecommendations](#) API-Anfrage anwenden. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wenn Sie Ereignisse für einen Benutzer aufgezeichnet haben, bevor er sich angemeldet hat (ein anonymer Benutzer), können Sie Empfehlungen für diesen Benutzer erhalten, indem Sie die `sessionId` Ereignisse so angeben, als ob es seine eigenen wären `userId`. Weitere Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

```
import boto3
```

```
personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_action_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)

print("Recommended actions")
for item in response['actionList']:
    print (item['actionId'])
```

Ein personalisiertes Ranking erhalten (benutzerdefinierte Ressourcen)

Eine personalisierte Rangfolge ist eine Liste empfohlener Elemente, die für einen bestimmten Benutzer neu nach Rang angeordnet sind. Um personalisierte Rankings zu erhalten, rufen Sie den [GetPersonalizedRanking](#) API-Vorgang auf oder lassen Sie sich Empfehlungen aus einer Kampagne in der Konsole anzeigen.

Note

Die der Kampagne zugrunde liegende Lösung muss mit einem Rezept vom Typ PERSONALIZED_RANKING erstellt worden sein. Weitere Informationen finden Sie unter [Ein Rezept wählen](#).

Themen

- [So funktioniert die personalisierte Bewertung von Rankings](#)
- [Ein personalisiertes Ranking erhalten \(Konsole\)](#)
- [Ein personalisiertes Ranking erhalten \(AWS CLI\)](#)
- [Abrufen eines personalisierten Rankings \(AWSSDKs\)](#)
- [Beispielnotizbuch mit personalisiertem Ranking](#)

So funktioniert die personalisierte Bewertung von Rankings

Wie bei den Ergebnissen, die der GetRecommendations Vorgang für Lösungen zurückgibt, die mit dem Rezept für [Benutzerpersonalisierung](#) erstellt wurden, ergeben die GetPersonalizedRanking

Punktzahlen insgesamt 1, aber da die Liste der in Betracht gezogenen Artikel viel kleiner ist als Ihr vollständiger Katalog, sind die Empfehlungswerte in der Regel höher.

Mathematisch gesehen `GetPersonalizedRanking` ist die Bewertungsfunktion für identisch mit `GetRecommendations`, außer dass sie nur die Eingabelemente berücksichtigt. Dies bedeutet, dass sich mit zunehmender Annäherung an 1 die Wahrscheinlichkeit erhöht, da es immer weniger andere Möglichkeiten gibt, die Punktzahl aufzuteilen:

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_{j \in \text{input}} \exp(\bar{w}_u^\top w_j)}$$

Ein personalisiertes Ranking erhalten (Konsole)

Um ein personalisiertes Ranking für einen Benutzer über die Amazon Personalize-Konsole zu erhalten, wählen Sie die Kampagne aus, die Sie verwenden, und geben Sie dann deren Benutzer-ID an, geben Sie die Liste der Artikel an, die Sie für den Benutzer ranken möchten, wählen Sie optional einen Filter und geben Sie optional alle Kontextdaten an.

Um ein personalisiertes Ranking für einen Benutzer zu erhalten

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe aus, die die Kampagne enthält, die Sie verwenden.
3. Wählen Sie im Navigationsbereich Campaigns (Kampagnen) aus.
4. Wählen Sie auf der Seite Kampagnen die Zielkampagne aus.
5. Geben Sie unter Kampagnenergebnisse testen die Benutzer-ID des Benutzers ein, für den Sie Empfehlungen erhalten möchten.
6. Geben Sie unter Artikel-IDs die Liste der Artikel ein, die für den Benutzer eingestuft werden sollen.
7. Wählen Sie optional einen Filter aus. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).
8. Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne aktiviert haben, wählen Sie für Artikeldatensatz-Spalten die Metadaten aus, die Sie in die Empfehlungsergebnisse

aufnehmen möchten. Informationen zur Aktivierung von Metadaten finden Sie unter [Metadaten in Empfehlungen aktivieren](#).

9. Wenn Ihre Kampagne kontextbezogene Metadaten verwendet (Informationen zu den Anforderungen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#)), geben Sie optional Kontextdaten an.

Geben Sie für jeden Kontext für den Schlüssel das Metadatenfeld und für den Wert die Kontextdaten ein.

10. Wählen Sie Personalisierte Artikelranglisten abrufen aus. Es wird eine Tabelle mit den Elementen angezeigt, die in der Reihenfolge ihres voraussichtlichen Interesses für den Benutzer geordnet sind.

Ein personalisiertes Ranking erhalten (AWS CLI)

Die folgenden Codebeispiele zeigen, wie Sie mit den verschiedenen Varianten ein personalisiertes Ranking erzielen können. AWS CLI

Themen

- [Ein personalisiertes Ranking erhalten](#)
- [Aufnahme von Artikelmetadaten in ein personalisiertes Ranking](#)

Ein personalisiertes Ranking erhalten

Verwenden Sie den folgenden `get-personalized-ranking` Befehl, um ein personalisiertes Ranking mit dem zu erhalten AWS CLI. Geben Sie den Amazon-Ressourcennamen (ARN) für Ihre Kampagne und die Benutzer-ID für den Benutzer an und stellen Sie eine Liste mit Artikel-IDs für die Artikel bereit, die für den Benutzer eingestuft werden sollen (jeweils durch ein Leerzeichen getrennt). Die einzustufenden Elemente müssen in den Daten enthalten sein, die Sie zum Trainieren der Lösungsversion verwendet haben. Eine Liste der nach Rangfolge bewerteten Empfehlungen wird angezeigt. Amazon Personalize betrachtet das erste Element in der Liste, das für den Benutzer am interessantesten ist.

```
aws personalize-runtime get-personalized-ranking \  
--campaign-arn Campaign ARN \  
--user-id 12 \  
--input-list 3 4 10 8 12 7
```

Aufnahme von Artikelmetadaten in ein personalisiertes Ranking

Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne aktiviert haben, können Sie die Metadaten­spalten des Artikeldatensatzes angeben, die in die Antwort aufgenommen werden sollen. Informationen zur Aktivierung von Metadaten finden Sie unter [Metadaten in Empfehlungen aktivieren](#).

Das folgende Codebeispiel zeigt, wie Sie die Metadaten­spalten als Teil Ihrer Anfrage für ein personalisiertes Ranking angeben.

```
aws personalize-runtime get-personalized-ranking \  
--campaign-arn Campaign ARN \  
--user-id 12 \  
--input-list 3 4 10 8 12 7 \  
--metadata-columns '{"ITEMS": [{"columnNameA"}, {"columnNameB"}]}'
```

Abrufen eines personalisierten Rankings (AWSSDKs)

Die folgenden Codebeispiele zeigen, wie Sie mit den SDKs in verschiedenen Varianten ein personalisiertes Ranking erzielen können. AWS

Themen

- [Ein personalisiertes Ranking erhalten](#)
- [Aufnahme von Artikelmetadaten in ein personalisiertes Ranking](#)
- [Abrufen eines personalisierten Rankings mithilfe von kontextuellen Metadaten](#)

Ein personalisiertes Ranking erhalten

Der folgende Code zeigt, wie Sie ein personalisiertes Ranking für einen Benutzer erhalten. Geben Sie die Benutzer-ID und eine Liste von Artikel­IDs an, die für den Benutzer eingestuft werden sollen. Die Element­IDs müssen in den Daten enthalten sein, mit denen Sie die Lösungs­version trainiert haben. Es wird eine Liste von Empfehlungen mit Rangfolge zurückgegeben. Amazon Personalize betrachtet das erste Element in der Liste, das für den Benutzer am interessantesten ist.

SDK for Python (Boto3)

```
import boto3  
  
personalizeRt = boto3.client('personalize-runtime')
```

```

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = ['ItemID1', 'ItemID2']
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])

```

SDK for Java 2.x

```

public static List<PredictedItem> getRankedRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
                                                String campaignArn,
                                                String userId,
                                                ArrayList<String> items) {

    try {
        GetPersonalizedRankingRequest rankingRecommendationsRequest =
GetPersonalizedRankingRequest.builder()
            .campaignArn(campaignArn)
            .userId(userId)
            .inputList(items)
            .build();

        GetPersonalizedRankingResponse recommendationsResponse =

personalizeRuntimeClient.getPersonalizedRanking(rankingRecommendationsRequest);
        List<PredictedItem> rankedItems =
recommendationsResponse.personalizedRanking();
        int rank = 1;
        for (PredictedItem item : rankedItems) {
            System.out.println("Item ranked at position " + rank + " details");
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
            System.out.println("-----");
            rank++;
        }
        return rankedItems;
    } catch (PersonalizeRuntimeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

```
    }  
    return null;  
  }  
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.  
import { GetPersonalizedRankingCommand } from  
  "@aws-sdk/client-personalize-runtime";  
import { personalizeRuntimeClient } from "../libs/personalizeClients.js";  
// Or, create the client here.  
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:  
  "REGION"});  
  
// Set the ranking request parameters.  
export const getPersonalizedRankingParam = {  
  campaignArn: "CAMPAIGN_ARN", /* required */  
  userId: 'USER_ID',          /* required */  
  inputList: ["ITEM_ID_1", "ITEM_ID_2", "ITEM_ID_3", "ITEM_ID_4"]  
}  
  
export const run = async () => {  
  try {  
    const response = await personalizeRuntimeClient.send(new  
GetPersonalizedRankingCommand(getPersonalizedRankingParam));  
    console.log("Success!", response);  
    return response; // For unit tests.  
  } catch (err) {  
    console.log("Error", err);  
  }  
};  
run();
```

Aufnahme von Artikelmetadaten in ein personalisiertes Ranking

Wenn Sie Metadaten in Empfehlungen für Ihre Kampagne aktiviert haben, können Sie die Metadaten­spalten des Artikeldatensatzes angeben, die in die Antwort aufgenommen werden sollen. Informationen zur Aktivierung von Metadaten finden Sie unter [Metadaten in Empfehlungen aktivieren](#).

Das folgende Codebeispiel zeigt, wie Sie die Metadaten­spalten als Teil Ihrer Anfrage für ein personalisiertes Ranking angeben.


```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = [ItemID1, ItemID2],
    metadataColumns = {
        "ITEMS": [columnNameA, columnNameB]
    }
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])
    print (item['metadata'])
```

Abrufen eines personalisierten Rankings mithilfe von kontextuellen Metadaten

Verwenden Sie den folgenden Code, um ein personalisiertes Ranking basierend auf kontextbezogenen Metadaten zu erhalten. Geben Sie für jedes Schlüssel-Wert-Paar das Metadatenfeld als Schlüssel und die Kontextdaten als Wert an. `context` Im folgenden Beispielcode ist der Schlüssel `DEVICE` und der Wert ist `mobile phone`. Ersetzen Sie diese Werte und das Campaign ARN und User ID durch Ihre eigenen. Wechseln Sie auch `inputList` zu einer Liste von Element-IDs, die in den Daten enthalten sind, die Sie zum Trainieren der Lösung verwendet haben. Amazon Personalize betrachtet das erste Element in der Liste, das für den Benutzer am interessantesten ist.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign ARN",
    userId = "User ID",
    inputList = [ItemID1, ItemID2],
    context = {
        'DEVICE': 'mobile phone'
    }
)
```

```
print("Personalized Ranking")
for item in response['personalizedRanking']:
    print(item['itemId'])
```

Beispielnotizbuch mit personalisiertem Ranking

[Ein Beispiel für ein Jupyter-Notizbuch, das zeigt, wie das Rezept für personalisiertes Ranking verwendet wird, finden Sie unter Beispiel für personalisiertes Ranking.](#)

Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten

Um die Relevanz von Empfehlungen zu erhöhen, fügen Sie kontextbezogene Metadaten für einen Benutzer hinzu, z. B. seinen Gerätetyp oder die Tageszeit, wenn Sie Artikelempfehlungen oder ein personalisiertes Ranking erhalten.

Um kontextbezogene Metadaten verwenden zu können, muss das Schema des Datensatzes Artikelinteraktionen über Metadatenfelder für die Kontextdaten verfügen. Zum Beispiel ein DEVICE-Feld (siehe). [Schemata](#)

Für Domain-Datensatzgruppen können in den folgenden empfohlenen Anwendungsfällen kontextuelle Metadaten verwendet werden:

- [Empfohlen für Sie](#)(E-COMMERCE-Domäne)
- [Top-Auswahlen für Sie](#)(Domäne VIDEO_ON_DEMAND)

Für benutzerdefinierte Ressourcen beinhalten Rezepte, die kontextuelle Metadaten verwenden, Folgendes:

- [Benutzerpersonalisierung](#)
- [Personalisierte Rangfolge](#)

Weitere Informationen zu Kontextinformationen finden Sie im folgenden Blogbeitrag zum AWS Machine Learning: [Erhöhung der Relevanz Ihrer Amazon Personalize Personalize-Empfehlungen durch die Nutzung von Kontextinformationen.](#)

Sie können Empfehlungen mit kontextbezogenen Metadaten über die Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder SDKs abrufen. AWS

Empfehlungen mithilfe von kontextuellen Metadaten abrufen (AWS Python SDK)

Um die Relevanz von Empfehlungen zu erhöhen, fügen Sie kontextbezogene Metadaten für einen Benutzer hinzu, z. B. seinen Gerätetyp oder die Tageszeit, wenn Sie Artikelempfehlungen oder ein personalisiertes Ranking erhalten.

Verwenden Sie den folgenden Code, um eine Empfehlung basierend auf kontextbezogenen Metadaten abzurufen. Geben Sie für jedes Schlüssel-Wert-Paar das Metadatenfeld als Schlüssel und die Kontextdaten als Wert an. `context` Im folgenden Beispielcode ist der Schlüssel `DEVICE` und der Wert ist `mobile phone`. Ersetzen Sie diese Werte und das `Campaign ARN` und `User ID` durch Ihre eigenen. Wenn Sie einen Empfehlungsgeber erstellt haben, ersetzen Sie ihn `campaignArn` durch `recommenderArn`. Eine Liste mit empfohlenen Elementen für den Benutzer wird angezeigt.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    context = {
        'DEVICE': 'mobile phone'
    }
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

Batch-Empfehlungen und Benutzersegmente (benutzerdefinierte Ressourcen)

Mit benutzerdefinierten Ressourcen können Sie Batch-Empfehlungen oder Benutzersegmente mit einem asynchronen Batchablauf abrufen. Sie könnten beispielsweise Produktempfehlungen für alle Benutzer auf einer E-Mail-Liste oder [item-to-item Ähnlichkeiten](#) in einem Inventar erhalten. Oder mit den `USER_SEGMENTATION`-Rezepten können Sie Benutzersegmente für datengesteuerte Werbung erstellen, die auf Artikeln in Ihrem Inventar und den Interaktionen Ihrer Nutzer basieren.

- Um Batch-Empfehlungen zu erhalten, verwenden Sie einen Batch-Inferenzjob. Ein Batch-Inferenzjob ist ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket importiert,

Ihre Lösungsversion verwendet, um Articlempfehlungen zu generieren, und die Empfehlungen in einen Amazon S3 S3-Bucket exportiert.

- Um Benutzersegmente abzurufen, verwenden Sie einen Batch-Segment-Job. Ein Batch-Segment-Job ist ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket importiert, Ihre mit einem USER_SEGMENTATION-Rezept trainierte Lösungsversion verwendet, um Benutzersegmente zu generieren, und die Segmente in einen Amazon S3 S3-Bucket exportiert.

Themen

- [Abrufen von Batch-Empfehlungen](#)
- [Abrufen von Benutzersegmenten](#)

Abrufen von Batch-Empfehlungen

Mit benutzerdefinierten Ressourcen können Sie Elementempfehlungen mit einem asynchronen Batch-Flow abrufen. Sie können beispielsweise Produktempfehlungen für alle Benutzer auf einer E-Mail-Liste oder [item-to-item Ähnlichkeiten](#) in einem Bestand erhalten.

Um Batch-Empfehlungen für Elemente zu erhalten, verwenden Sie einen Batch-Inferenzauftrag. Ein Batch-Inferenzauftrag ist ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3-Bucket importiert, Ihre benutzerdefinierte Lösungsversion verwendet, um Elementempfehlungen zu generieren, und dann die Elementempfehlungen in einen Amazon S3-Bucket exportiert. Je nach Rezept sind Ihre Eingabedaten eine Liste von Benutzern oder Elementen oder eine Liste von Benutzern mit jeweils einer Sammlung von Elementen.

Wenn Ihre Lösung das Rezept Ähnliche Elemente verwendet und Sie einen Artikeldatensatz mit Textdaten und Artikeltiteldaten haben, können Sie Batch-Empfehlungen mit Designs für jede Gruppe von Elementen generieren. Weitere Informationen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Beim Generieren von Batch-Empfehlungen berücksichtigt Amazon Personalize alle zum Zeitpunkt der letzten Erstellung der Lösungsversion vorhandenen Massendaten. Diese Daten können mit dem Importmodus FULL oder INCREMENTAL importiert werden. Damit neuere Massendatensätze Batch-Empfehlungen beeinflussen können, müssen Sie eine neue Lösungsversion und dann den Batch-Inferenzauftrag erstellen.

Amazon Personalize verwendet Daten aus einzelnen Importen, wenn Batch-Empfehlungen wie folgt generiert werden:

- **Neue Interaktionen mit vorhandenen Elementen und Benutzern:** Wenn Sie die Rezepte `User-Personalization` oder `Personalized-Ranking` verwenden, berücksichtigt Amazon Personalize neue Interaktionsdaten mit vorhandenen Elementen und Benutzern innerhalb von etwa 15 Minuten nach dem Datenimport. Um sicherzustellen, dass Ereignisse berücksichtigt werden, empfehlen wir Ihnen, nach dem Import mindestens 15 Minuten zu warten, bevor Sie einen Batch-Inferenzauftrag starten. Für alle anderen Rezepte müssen Sie eine neue Lösungsversion für gestreamte Ereignisse erstellen, um Batch-Empfehlungen zu beeinflussen.
- **Neue Benutzer:** Für Benutzer ohne Interaktionsdaten gelten Empfehlungen zunächst nur für beliebte Elemente. Wenn Sie `User-Personalization` oder `Personalized-Ranking` verwenden und Ereignisse für den Benutzer aufzeichnen, werden seine Empfehlungen möglicherweise innerhalb von etwa 15 Minuten nach dem Import relevanter, ohne erneut geschult zu werden. Um sicherzustellen, dass Ereignisse berücksichtigt werden, empfehlen wir Ihnen, nach dem Import mindestens 15 Minuten zu warten, bevor Sie einen Batch-Inferenzauftrag starten. Für alle anderen Rezepte müssen Sie eine neue Lösungsversion für gestreamte Ereignisse erstellen, um Batch-Empfehlungen für Benutzer ohne Interaktionsdaten zu beeinflussen.
- **Neue Elemente:** Wenn Sie mit `User-Personalization` einen Batch-Inferenzauftrag erstellen und die neueste vollständig trainierte Lösungsversion für Ihre Lösung angeben, aktualisiert Amazon Personalize automatisch die Lösungsversion, um neue Elemente in Empfehlungen für die Erkundung aufzunehmen. Wenn Sie nicht die neueste Lösungsversion angeben, erfolgt kein Update. Für jedes andere Rezept müssen Sie eine neue Lösungsversion erstellen, damit neue Elemente in Batch-Empfehlungen vorgestellt werden können. Weitere Informationen zur Erkundung finden Sie unter [Exploration \(Erkundung\)](#).

Themen

- [Batch-Workflow](#)
- [Richtlinien und Anforderungen](#)
- [Batch-Workflow-Bewertung](#)
- [Batch-Empfehlungen mit Themen aus dem Content Generator](#)
- [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#)
- [Erstellen eines Batch-Inferenzauftrags](#)
- [Beispiele für die Ausgabe von Batch-Inferenz-Jobs](#)

Batch-Workflow

Der Batch-Workflow sieht wie folgt aus:

1. Bereiten Sie Ihre Eingabedaten im JSON-Format vor und laden Sie sie in einen Amazon S3-Bucket hoch. Das Format Ihrer Eingabedaten hängt vom verwendeten Rezept ab. Siehe [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#).
2. Erstellen Sie einen separaten Speicherort für Ihre Ausgabedaten, entweder einen Ordner oder einen anderen Amazon S3-Bucket.
3. Erstellen Sie einen Batch-Inferenzauftrag. Siehe [Erstellen eines Batch-Inferenzauftrags](#).
4. Wenn die Batch-Inferenz abgeschlossen ist, rufen Sie die Elementempfehlungen von Ihrem Ausgabespeicherort in Amazon S3 ab.

Richtlinien und Anforderungen

Im Folgenden finden Sie Richtlinien und Anforderungen für den Erhalt von Batch-Empfehlungen:

- Ihre Amazon-Personalize-IAM-Servicerolle muss über die Berechtigung zum Lesen und Hinzufügen von Dateien zu Ihren Amazon S3-Buckets verfügen. Informationen zum Erteilen von Berechtigungen finden Sie unter [Richtlinie für Servicerollen für Batch-Workflows](#). Weitere Informationen zu Bucket-Berechtigungen finden Sie unter [Beispiele für Benutzerrichtlinien](#) im Amazon Simple Storage Service-Entwicklerhandbuch. Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon-Personalize-IAM-Servicerolle die Berechtigung zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).
- Sie müssen eine benutzerdefinierte Lösung und Lösungsversion erstellen, bevor Sie einen Batch-Inferenzauftrag erstellen. Sie müssen jedoch keine Amazon-Personalize-Kampagne erstellen. Wenn Sie eine Domain-Datensatzgruppe erstellt haben, können Sie dennoch benutzerdefinierte Ressourcen erstellen.
- Um Designs mit Empfehlungen zu generieren, müssen Sie das Rezept Ähnliche Elemente verwenden. Und Sie müssen über einen Elementdatensatz mit Textdaten und Elementtiteldaten verfügen. Weitere Informationen zu Themenempfehlungen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).
- Ihre Eingabedaten müssen wie unter beschrieben formatiert sein [Vorbereiten von Eingabedaten für Benutzersegmente](#).

- Mit den Trending-Now- oder Next-Best-Action-Rezepten können Sie keine Batch-Empfehlungen erhalten.
- Wenn Sie einen Filter mit Platzhalterparametern verwenden, müssen Sie die Werte für die Parameter in Ihren Eingabedaten in ein `filterValues` Objekt aufnehmen. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).
- Wir empfehlen, dass Sie einen anderen Speicherort für Ihre Ausgabedaten (entweder einen Ordner oder einen anderen Amazon S3-Bucket) als Ihre Eingabedaten verwenden.
- Batch-Empfehlungen sind möglicherweise nicht genau mit Empfehlungen in Echtzeit identisch. Dies liegt daran, dass Batch-Inferenzanträge länger dauern und nur Daten 15 Minuten vor Beginn des Auftrags als verfügbar betrachten.

Batch-Workflow-Bewertung

Batch-Empfehlungen umfassen die folgenden Ergebnisse:

- Mit User-Personalization- und Personalized-Ranking-Rezepten berechnet Amazon Personalize die Empfehlungswerte für Batch-Inferenzanträge, wie unter [So funktioniert die Bewertung von Empfehlungen zur Benutzerpersonalisierung \(benutzerdefinierte Ressourcen\)](#) und beschrieben [So funktioniert die personalisierte Bewertung von Rankings](#). Sie können Ergebnisse in der Ausgabe-JSON-Datei des Batch-Inferenzantrags anzeigen.
- Wenn Sie mit dem Rezept Ähnliche Elemente die entsprechenden Batch-Empfehlungen erhalten, sortiert Amazon Personalize jeden Satz verwandter Elemente danach, wie relevant das Design für jedes Element ist. Jedes Element enthält eine Punktzahl von 0 bis 1. Je höher der Wert ist, desto eng ist das Element mit dem Design verwandt. Weitere Informationen zu Empfehlungen für Designs finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Batch-Empfehlungen mit Themen aus dem Content Generator

Important

Wenn Sie Batch-Empfehlungen mit Themen erhalten, entstehen Ihnen zusätzliche Kosten. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Preise](#).

Wenn Sie den verwenden [Ähnliche Elemente-Rezepte](#), kann Amazon Personalize Content Generator den Batch-Empfehlungen beschreibende Themen hinzufügen. Content Generator ist eine Funktion für generative künstliche Intelligenz (generative KI), die von Amazon Personalize verwaltet wird.

Wenn Sie Stapelempfehlungen mit Themen erhalten, fügt Amazon Personalize Content Generator für jeden Satz ähnlicher Artikel ein beschreibendes Thema hinzu. Das Thema basiert auf der Artikelbeschreibung und den Artikelnamen in Ihrem Artikeldatensatz. Amazon Personalize bezieht die Themen in die Ausgabe des Batch-Inferenz-Jobs ein. Sie können die Themen verwenden, um den Text in Ihrer Bewerbung oder Ihren Marketingbotschaften überzeugender zu gestalten.

Wenn Sie beispielsweise Empfehlungen zu ähnlichen Artikeln für ein Frühstück erhalten, generiert Amazon Personalize möglicherweise ein Thema wie Rise and Shine oder Morning Essentials. Sie könnten das Thema verwenden, um einen generischen Karusselltitel wie „Häufig zusammen gekauft“ zu ersetzen. Oder Sie könnten das Thema in eine Werbe-E-Mail oder eine Marketingkampagne für neue Menüoptionen integrieren.

AWSüberwacht keine Themen aus dem Content Generator. Um die Qualität des Themas zu überprüfen, können Sie die Ergebnisse verwenden, die für jeden empfohlenen Artikel erstellt wurden. Weitere Informationen finden Sie unter [Rangfolge und Bewertung von Batch-Empfehlungen mit bestimmten Themen](#).

Themen

- [Unterstützte -Regionen](#)
- [Richtlinien und Anforderungen](#)
- [Rangfolge und Bewertung von Batch-Empfehlungen mit bestimmten Themen](#)
- [Generierung von Batch-Empfehlungen mit Themen](#)

Unterstützte -Regionen

Amazon Personalize Content Generator ist nur in den folgenden AWS Regionen verfügbar:

- USA Ost (Nord-Virginia)
- USA West (Oregon)
- Asien-Pazifik (Tokio)

Richtlinien und Anforderungen

Im Folgenden finden Sie Richtlinien und Anforderungen für die Erstellung von Empfehlungen zu Themen:

- Ihre Eingabedatei kann bis zu 100 Elemente enthalten. Informationen zu Eingabedaten für Batch-Empfehlungen finden Sie unter [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#).
- Ihre Lösung muss die verwenden [Ähnliche Elemente-Rezepte](#).
- Sie benötigen einen Artikel-Datensatz mit den folgenden Daten. Diese Daten können dazu beitragen, relevantere Themen zu generieren.
 - Es muss ein Textfeld haben, z. B. ein BESCHREIBUNGSFELD. Hinweise zu Textdaten finden Sie unter [Unstrukturierte Textmetadaten](#)
 - Es muss eine Zeichenfolgenspalte mit Elementnamendaten enthalten, z. B. ein TITLE-Feld.

Wenn Ihr Artikeldatensatz diese Daten nicht enthält, können Sie sie hinzufügen. Informationen zum Aktualisieren vorhandener Daten finden Sie unter [Aktualisieren von Daten](#).

Rangfolge und Bewertung von Batch-Empfehlungen mit bestimmten Themen

Wenn Sie Stapelempfehlungen mit Themen erhalten, ordnet Amazon Personalize jede Artikelgruppe danach, wie relevant das Thema für jeden Artikel ist. Jeder Artikel enthält eine Punktzahl in einem ungefähren Bereich von -0,1 bis 0,6. Je höher die Punktzahl, desto enger ist der Artikel mit dem Thema verwandt. Sie können die Punktzahlen verwenden, um einen Schwellenwert festzulegen, sodass nur Elemente angezeigt werden, die in engem Zusammenhang mit dem Thema stehen.

Amazon Personalize könnte beispielsweise das Thema zurückgeben *For your sweet tooth*, und die verwandten Artikel und ihre Punktzahlen könnten lauten: harte Süßigkeiten (Punktzahl 0,19884521), Schokolade (Punktzahl 0,17664525), Apfel (Punktzahl 0,08994528), Eis am Stiel (Punktzahl 0,14294521), Süßkartoffel (Punktzahl 0,07794527) und Karotte (Punktzahl 0,04994523). In Ihrer Anwendung könnten Sie eine Regel hinzufügen, nach der nur Artikel mit einer Punktzahl von $\geq 0,10$ oder mehr berücksichtigt werden, wobei Obst und Gemüse weggelassen werden.

Das folgende Beispiel zeigt das Format der Ausgabe eines Batch-Inferenzjobs, der Filmempfehlungen mit Motiven generiert.

```

{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135314

```

```
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.138913,-
...
}
```

Generierung von Batch-Empfehlungen mit Themen

Um Batch-Empfehlungen mit Themen zu generieren, führen Sie den Batch-Workflow wie unter beschrieben durch [Batch-Workflow](#). Sie bereiten Ihre Eingabedaten auf die gleiche Weise vor, wie Sie es für ein RELATED_ITEMS Rezept tun würden. Ein Beispiel finden Sie unter [RELATED_ITEMS-Rezepte](#).

Wenn Sie den Batch-Inferenzjob erstellen, aktivieren Sie die Themengenerierung und geben die Artikeltitelspalte Ihres Artikeldatensatzes an.

- Informationen zur Verwendung der Amazon Personalize-Konsole zum Erstellen eines Batch-Inferenzjobs, der Themen generiert, finden Sie unter [Erstellen eines Batch-Inferenzauftrags](#)
- Ein Codebeispiel, das zeigt, wie Sie mit dem SDK for Python (Boto3) einen Batch-Inferenzjob erstellen, der Themes generiert, finden Sie unter [Erstellen eines Batch-Inferenzauftrags, der Designs generiert](#)

Vorbereiten von Eingabedaten für Batch-Empfehlungen

Ein Batch-Inferenzauftrag importiert Ihre Batch-Eingabe-JSON-Daten aus einem Amazon S3-Bucket, verwendet Ihre benutzerdefinierte Lösungsversion, um Empfehlungen zu generieren, und exportiert dann die Elementempfehlungen in einen Amazon S3-Bucket. Bevor Sie Batch-Empfehlungen erhalten können, müssen Sie Ihre JSON-Datei vorbereiten und in einen Amazon S3-Bucket hochladen. Wir empfehlen Ihnen, einen Ausgabeordner in Ihrem Amazon S3-Bucket zu erstellen oder einen separaten Amazon S3-Ausgabe-Bucket zu verwenden. Anschließend können Sie mehrere Batch-Inferenzaufträge mit demselben Eingabedatenspeicherort ausführen.

Wenn Sie einen Filter mit Platzhalterparametern verwenden, z. B. \$GENRE, müssen Sie die Werte für die Parameter in einem `filterValues` Objekt in Ihrem Eingabe-JSON angeben. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).

So bereiten Sie Daten vor und importieren sie

1. Formatieren Sie Ihre Batch-Eingabedaten je nach Rezept. Mit dem Trending-Now-Rezept können Sie keine Batch-Empfehlungen erhalten.

- Für USER_PERSONALIZATION-Rezepte und das Popularity-Count-Rezept sind Ihre Eingabedaten eine JSON-Datei mit einer Liste von userIds
- Für RELATED_ITEMS-Rezepte sind Ihre Eingabedaten eine Liste von itemIds
- Bei PERSONALIZED_RANKING-Rezepten sind Ihre Eingabedaten eine Liste von userIds, die jeweils mit einer Sammlung von itemIds gekoppelt sind

Trennen Sie jede Zeile durch eine neue Zeile. Beispiele für Eingabedaten finden Sie unter [Beispiele für Batch-Inferenzanfrageingaben und -ausgabe-JSON](#).

2. Laden Sie Ihren Eingabe-JSON in einen Eingabeordner in Ihrem Amazon S3-Bucket hoch. Weitere Informationen finden Sie unter [Hochladen von Dateien und Ordnern per Drag & Drop](#) im Benutzerhandbuch für Amazon Simple Storage Service.
3. Erstellen Sie einen separaten Speicherort für Ihre Ausgabedaten, entweder einen Ordner oder einen anderen Amazon S3-Bucket. Indem Sie einen separaten Speicherort für das Ausgabe-JSON erstellen, können Sie mehrere Batch-Inferenzanfragen mit demselben Eingabedatenspeicherort ausführen.
4. Erstellen Sie einen Batch-Inferenzanfrage. Amazon Personalize gibt die Empfehlungen aus Ihrer Lösungsversion an Ihren Ausgabedatenspeicherort aus.

Beispiele für Batch-Inferenzanfrageingaben und -ausgabe-JSON

So formatieren Sie Ihre Eingabedaten das von Ihnen verwendete Rezept. Wenn Sie einen Filter mit Platzhalterparametern verwenden, z. B. \$GENRE, müssen Sie die Werte für die Parameter in einem filterValues Objekt in Ihrem Eingabe-JSON angeben. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).

In den folgenden Abschnitten werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für Batch-Inferenzanfragen aufgeführt. Mit dem Trending-Now-Rezept können Sie keine Batch-Empfehlungen erhalten.

Themen

- [USER_PERSONALIZATION-Rezepte](#)
- [POP_ITEMS-Rezepte \(nur Popularity-Count\)](#)
- [Personalisierte-Rangfolgen-Rezepte](#)
- [RELATED_ITEMS-Rezepte](#)

USER_PERSONALIZATION-Rezepte

Im Folgenden werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für USER_PERSONALIZATION-Rezepte gezeigt.

Input

Trennen Sie jede `userId` Zeile wie folgt durch eine neue Zeile.

```
{"userId": "4638"}  
{"userId": "663"}  
{"userId": "3384"}  
...
```

Output

```
{"input":{"userId":"4638"},"output":{"recommendedItems":  
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","6237  
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.  
{"input":{"userId":"663"},"output":{"recommendedItems":  
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104"  
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.  
{"input":{"userId":"3384"},"output":{"recommendedItems":  
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035"  
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.  
...
```

POP_ITEMS-Rezepte (nur Popularity-Count)

Im Folgenden werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für das Popularity-Count-Rezept gezeigt. Mit dem Trending-Now-Rezept können Sie keine Batch-Empfehlungen erhalten.

Input

Trennen Sie jede `userId` Zeile wie folgt durch eine neue Zeile.

```
{"userId": "12"}  
{"userId": "105"}  
{"userId": "41"}  
...
```

Output

```
{ "input": { "userId": "12"}, "output": { "recommendedItems": ["105", "106", "441"]}}
{ "input": { "userId": "105"}, "output": { "recommendedItems": ["105", "106", "441"]}}
{ "input": { "userId": "41"}, "output": { "recommendedItems": ["105", "106", "441"]}}
...
```

Personalisierte-Rangfolgen-Rezepte

Im Folgenden werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für PERSONALIZED_RANKING-Rezepte gezeigt.

Input

Trennen Sie jede - `userId` und -Liste, die mit einer neuen Zeile eingestuft werden `itemIds` soll, wie folgt.

```
{ "userId": "891", "itemList": ["27", "886", "101"]}
{ "userId": "445", "itemList": ["527", "55", "901"]}
{ "userId": "71", "itemList": ["27", "351", "101"]}
...
```

Output

```
{ "input": { "userId": "891", "itemList": ["27", "886", "101"]}, "output":
{ "recommendedItems": ["27", "101", "886"], "scores": [0.48421, 0.28133, 0.23446]}}
{ "input": { "userId": "445", "itemList": ["527", "55", "901"]}, "output":
{ "recommendedItems": ["901", "527", "55"], "scores": [0.46972, 0.31011, 0.22017]}}
{ "input": { "userId": "71", "itemList": ["29", "351", "199"]}, "output": { "recommendedItems":
["351", "29", "199"], "scores": [0.68937, 0.24829, 0.06232]}}
...
```

RELATED_ITEMS-Rezepte

Im Folgenden werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für RELATED_ITEMS-Rezepte gezeigt.

Input

Trennen Sie jede `itemId` Zeile wie folgt durch eine neue Zeile.

```

{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...

```

Output

```

{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}
...

```

Im Folgenden werden korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für das ähnliche Elementrezept mit Designs gezeigt.

Input

Trennen Sie jede `itemId` Zeile wie folgt durch eine neue Zeile.

```

{"itemId": "40"}
{"itemId": "43"}
...

```

Output

```

{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.13891
...

```

Erstellen eines Batch-Inferenzauftrags

Erstellen Sie einen Batch-Inferenzauftrag, um Batch-Elementempfehlungen für Benutzer basierend auf Eingabedaten aus Amazon S3 zu erhalten. Die Eingabedaten können eine Liste von Benutzern oder Elementen (oder beidem) im JSON-Format sein. Sie können einen Batch-Inferenzauftrag mit

der Amazon-Personalize-Konsole, der AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.

Wenn Sie einen Batch-Inferenzauftrag erstellen, geben Sie die Amazon S3-Pfade zu Ihren Eingabe- und Ausgabespeicherorten an. Amazon S3 basiert auf Präfixen. Wenn Sie ein Präfix für den Eingabedatenspeicherort angeben, verwendet Amazon Personalize alle Dateien, die diesem Präfix entsprechen, als Eingabedaten. Wenn Sie beispielsweise angeben `s3://<name of your S3 bucket>/folderName` und Ihr Bucket auch einen Ordner mit dem Pfad `s3://<name of your S3 bucket>/folderName_test`, verwendet Amazon Personalize alle Dateien in beiden Ordnern als Eingabedaten. Um nur die Dateien in einem bestimmten Ordner als Eingabedaten zu verwenden, beenden Sie den Amazon S3-Pfad mit einem Präfixtrennzeichen, z. B. `/: s3://<name of your S3 bucket>/folderName/`. Weitere Informationen darüber, wie Amazon S3 Objekte organisiert, finden Sie unter [Organisieren, Auflisten und Arbeiten mit Ihren Objekten](#).

Weitere Informationen zum Batch-Workflow in Amazon Personalize, einschließlich Berechtigungsanforderungen, Empfehlungsbewertung und Vorbereitung und Import von Eingabedaten, finden Sie unter [Abrufen von Batch-Empfehlungen](#).

Themen

- [Erstellen eines Batch-Inferenzauftrags \(Konsole\)](#)
- [Erstellen eines Batch-Inferenzauftrags \(AWS CLI\)](#)
- [Erstellen eines Batch-Inferenzauftrags \(AWS SDKs\)](#)

Erstellen eines Batch-Inferenzauftrags (Konsole)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie einen Batch-Inferenzauftrag erstellen. Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits eine Lösung und eine Lösungsversion (trainiertes Modell) erstellt haben.

So erstellen Sie einen Batch-Inferenzauftrag (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich unter Benutzerdefinierte Ressourcen die Option Batch-Inferenzaufträge aus.
4. Wählen Sie Create batch inference job (Batch-Inferenzauftrag erstellen).

5. Wählen Sie den Auftragstyp Batch-Inferenz aus.
 - Um Elementempfehlungen ohne Designs zu generieren, wählen Sie Elementempfehlungen aus.
 - Wenn Sie das Rezept Ähnliche Elemente verwenden und beschreibende Themen zu Gruppen ähnlicher Elemente hinzufügen möchten, wählen Sie Themenempfehlungen mit Content Generator aus. Um Designs zu generieren, benötigen Sie einen Elementdatensatz mit Elementnamendaten und Textdaten. Weitere Informationen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).
6. Geben Sie unter Batch inference job details (Batch-Inferenzauftragsdetails) unter Batch inference job name (Batch-Inferenzauftragsname) einen Namen für den Batch-Inferenzauftrag an.
7. Wählen Sie für Lösung die Lösung und dann die Lösungsversions-ID aus, die Sie zum Generieren der Empfehlungen verwenden möchten.
8. Geben Sie für Anzahl der Ergebnisse optional die Anzahl der Empfehlungen für jede Zeile der Eingabedaten an. Der Standardwert ist 25.
9. Wenn Ihr Batch-Auftrag Empfehlungen mit Designs generiert, wählen Sie unter Details zu Themenempfehlungen die Spalte aus, die Namen oder Titel für die Elemente in Ihrem Artikeldatensatz enthält. Diese Daten können dazu beitragen, relevantere Themen zu generieren. Weitere Informationen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).
10. Geben Sie unter Eingabequelle den Amazon S3-Pfad zu Ihrer Eingabedatei an.

Verwenden Sie die folgende Syntax: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json**

Ihre Eingabedaten müssen das richtige Format für das Rezept haben, das Ihre Lösung verwendet. Beispiele für Eingabedaten finden Sie unter [Beispiele für Batch-Inferenzauftragseingaben und -ausgabe-JSON](#).

11. Wenn Sie für Entschlüsselungsschlüssel Ihren eigenen AWS KMS Schlüssel für die Bucket-Verschlüsselung verwenden, geben Sie den Amazon-Ressourcennamen (ARN) Ihres Schlüssels an. Amazon Personalize muss über die Berechtigung zur Verwendung Ihres Schlüssels verfügen. Informationen zum Erteilen von Berechtigungen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).
12. Geben Sie unter Ausgabeziel den Pfad zu Ihrem Ausgabespeicherort an. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket).

Verwenden Sie die folgende Syntax: **s3://<name of your S3 bucket>/<output folder name>/**

13. Wenn Sie für Verschlüsselungsschlüssel Ihren eigenen AWS KMS Schlüssel für die Verschlüsselung verwenden, geben Sie den ARN Ihres Schlüssels an. Amazon Personalize muss über die Berechtigung zur Verwendung Ihres Schlüssels verfügen. Informationen zum Erteilen von Berechtigungen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).
14. Wählen Sie für IAM-Servicerolle die IAM-Servicerolle aus, die Sie während der Einrichtung für Amazon Personalize erstellt haben. Diese Rolle muss über Lese- und Schreibzugriff auf Ihre AmazonAmazon S3-Eingabe- bzw. Ausgabe-Buckets verfügen.
15. Wählen Sie unter Filter optional einen Filter aus, um einen Filter auf die Batch-Empfehlungen anzuwenden. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).
16. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Markieren von Amazon-Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).
17. Wählen Sie Create batch inference job (Batch-Inferenzauftrag erstellen). Die Erstellung von Batch-Inferenzaufträgen wird gestartet, und die Seite Batch inference jobs (Batch-Inferenzaufträge) wird mit dem angezeigten Bereich Batch inference job detail (Batch-Inferenzauftragsdetails) angezeigt.

Wenn sich der Status des Batch-Inferenzauftrags in Aktiv ändert, können Sie die Ausgabe des Auftrags aus dem angegebenen Amazon S3-Ausgabe-Bucket abrufen. Der Name der Ausgabedatei entspricht dem Format *input-name*.out.

Erstellen eines Batch-Inferenzauftrags (AWS CLI)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie mit der Operation einen Batch-Inferenzauftrag erstellen [CreateBatchInferenceJob](#).

Themen

- [Erstellen eines Batch-Inferenzauftrags](#)
- [Erstellen eines Batch-Inferenzauftrags, der Designs generiert](#)

Erstellen eines Batch-Inferenzauftrags

Sie können den `create-batch-inference-job` Befehl verwenden, um einen Batch-Inferenzauftrag zu erstellen. Geben Sie einen Auftragsnamen an, ersetzen Sie `Solution version ARN` durch den Amazon-Ressourcennamen (ARN) Ihrer Lösungsversion und ersetzen Sie durch `IAM service role ARN` den ARN der IAM-Servicerolle, die Sie während der Einrichtung für Amazon Personalize erstellt haben. Diese Rolle muss über Lese- und Schreibzugriff auf Ihre Amazon S3-Eingabe- bzw. Ausgabe-Buckets verfügen. Geben Sie optional einen Filter-ARN an, um Empfehlungen zu filtern. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#).

Ersetzen Sie `S3 input path` und `S3 output path` durch den Amazon S3-Pfad zu Ihren Eingabedatei- und Ausgabespeicherorten. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket). Verwenden Sie die folgende Syntax für Eingabe- und Ausgabespeicherorte: **`s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json`** und **`s3://<name of your S3 bucket>/<output folder name>/`**.

Das Beispiel enthält optionale `itemExplorationConfig` Hyperparameter für das User-Personalization-Rezept: `explorationWeight` und `explorationItemAgeCutOff`. Fügen Sie optional `-explorationWeight` und `-explorationItemAgeCutOff` Werte ein, um die Untersuchung zu konfigurieren. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

```
aws personalize create-batch-inference-job \
--job-name Batch job name \
--solution-version-arn Solution version ARN \
--filter-arn Filter ARN \
--job-input s3DataSource={path=s3://S3 input path} \
--job-output s3DataDestination={path=s3://S3 output path} \
--role-arn IAM service role ARN \
--batch-inference-job-config "{\"itemExplorationConfig\":{\"explorationWeight\":  
\"0.3\", \"explorationItemAgeCutOff\": \"30\"}}"
```

Erstellen eines Batch-Inferenzauftrags, der Designs generiert

Um Designs für ähnliche Elemente zu generieren, müssen Sie das Rezept `Ähnliche Elemente` verwenden und Ihr Elementdatensatz muss ein Textfeld und eine Spalte mit Elementnamendaten

haben. Weitere Informationen zu Empfehlungen für Designs finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Der folgende Code erstellt einen Batch-Inferenzantrag, der Empfehlungen mit Designs generiert. Behalten Sie den `batch-inference-job-mode` Wert bei `THEME_GENERATION`. Ersetzen Sie durch `COLUMN_NAME` den Namen der Spalte, in der Ihre Elementnamendaten gespeichert sind.

```
aws personalize create-batch-inference-job \  
--job-name Themed batch job name \  
--solution-version-arn Solution version ARN \  
--filter-arn Filter ARN \  
--job-input s3DataSource={path=s3://S3 input path} \  
--job-output s3DataDestination={path=s3://S3 output path} \  
--role-arn IAM service role ARN \  
--batch-inference-job-mode THEME_GENERATION \  
--theme-generation-config "{\"fieldsForThemeGeneration\": {\"itemName\": \  
\"COLUMN_NAME\"}}"
```

Erstellen eines Batch-Inferenzantrags (AWS SDKs)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie mit der Operation einen Batch-Inferenzantrag erstellen [CreateBatchInferenceJob](#).

Themen

- [Erstellen eines Batch-Inferenzantrags](#)
- [Erstellen eines Batch-Inferenzantrags, der Designs generiert](#)

Erstellen eines Batch-Inferenzantrags

Sie können den folgenden Code verwenden, um einen Batch-Inferenzantrag zu erstellen. Geben Sie einen Auftragsnamen, den Amazon-Ressourcennamen (ARN) Ihrer Lösungsversion und den ARN der IAM-Service-Rolle an, die Sie während der Einrichtung für Amazon Personalize erstellt haben. Diese Rolle muss über Lese- und Schreibzugriff auf Ihre Amazon S3-Eingabe- und -Ausgabe-Buckets verfügen.

Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket). Verwenden Sie die folgende Syntax für Eingabe- und Ausgabespeicherorte: `s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json` und `s3://<name of your S3 bucket>/<output folder name>/`.

Geben Sie für die Anzahl der Elemente `annumResults`, die Amazon Personalize für jede Zeile der Eingabedaten vorhersagen soll. Geben Sie optional einen Filter-ARN an, um Empfehlungen zu filtern. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#).

SDK for Python (Boto3)

Das Beispiel enthält optionale `itemExplorationConfig` Hyperparameter für das User-Personalization-Rezept: `explorationWeight` und `explorationItemAgeCutOff`. Fügen Sie optional - `explorationWeight` und -`explorationItemAgeCutOff` Werte ein, um die Untersuchung zu konfigurieren. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM service role ARN",
    filterArn = "Filter ARN",
    batchInferenceJobConfig = {
        # optional USER_PERSONALIZATION recipe hyperparameters
        "itemExplorationConfig": {
            "explorationWeight": "0.3",
            "explorationItemAgeCutOff": "30"
        }
    },
    jobInput =
        {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json"}},
    jobOutput =
        {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder name>/"}}
)
```

SDK for Java 2.x

Das Beispiel enthält optionale für das User-Personalization-Rezept spezifische `itemExplorationConfig` Felder: `explorationWeight` und `explorationItemAgeCutOff`.

Fügen Sie optional - explorationWeight und -explorationItemAgeCutOffWerte ein, um die Untersuchung zu konfigurieren. Weitere Informationen finden Sie unter [User-Personalization-Rezept](#).

```
public static String createPersonalizeBatchInferenceJob(PersonalizeClient
personalizeClient,
                                                    String solutionVersionArn,
                                                    String jobName,
                                                    String filterArn,
                                                    String
s3InputDataSourcePath,
                                                    String
s3DataDestinationPath,
                                                    String roleArn,
                                                    String explorationWeight,
                                                    String
explorationItemAgeCutOff) {
    long waitInMilliseconds = 60 * 1000;
    String status;
    String batchInferenceJobArn;

    try {
        // Set up data input and output parameters.
        S3DataConfig inputSource = S3DataConfig.builder()
            .path(s3InputDataSourcePath)
            .build();
        S3DataConfig outputDestination = S3DataConfig.builder()
            .path(s3DataDestinationPath)
            .build();

        BatchInferenceJobInput jobInput = BatchInferenceJobInput.builder()
            .s3DataSource(inputSource)
            .build();
        BatchInferenceJobOutput jobOutputLocation = BatchInferenceJobOutput.builder()
            .s3DataDestination(outputDestination)
            .build();

        // Optional code to build the User-Personalization specific item exploration
        config.
        HashMap<String, String> explorationConfig = new HashMap<>();

        explorationConfig.put("explorationWeight", explorationWeight);
    }
}
```

```
explorationConfig.put("explorationItemAgeCutOff", explorationItemAgeCutOff);

BatchInferenceJobConfig jobConfig = BatchInferenceJobConfig.builder()
    .itemExplorationConfig(explorationConfig)
    .build();
// End optional User-Personalization recipe specific code.

CreateBatchInferenceJobRequest createBatchInferenceJobRequest =
CreateBatchInferenceJobRequest.builder()
    .solutionVersionArn(solutionVersionArn)
    .jobInput(jobInput)
    .jobOutput(jobOutputLocation)
    .jobName(jobName)
    .filterArn(filterArn)
    .roleArn(roleArn)
    .batchInferenceJobConfig(jobConfig) // Optional
    .build();

    batchInferenceJobArn =
personalizeClient.createBatchInferenceJob(createBatchInferenceJobRequest)
    .batchInferenceJobArn();
    DescribeBatchInferenceJobRequest describeBatchInferenceJobRequest =
DescribeBatchInferenceJobRequest.builder()
    .batchInferenceJobArn(batchInferenceJobArn)
    .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

// wait until the batch inference job is complete.
while (Instant.now().getEpochSecond() < maxTime) {

    BatchInferenceJob batchInferenceJob = personalizeClient
        .describeBatchInferenceJob(describeBatchInferenceJobRequest)
        .batchInferenceJob();

    status = batchInferenceJob.status();
    System.out.println("Batch inference job status: " + status);

    if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
        break;
    }
    try {
        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
```

```

        System.out.println(e.getMessage());
    }
}
return batchInferenceJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateBatchInferenceJobCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch inference job's parameters.

export const createBatchInferenceJobParam = {
    jobName: 'JOB_NAME',
    jobInput: {          /* required */
        s3DataSource: { /* required */
            path: 'INPUT_PATH', /* required */
            // kmsKeyArn: 'INPUT_KMS_KEY_ARN' /* optional */
        }
    },
    jobOutput: {        /* required */
        s3DataDestination: { /* required */
            path: 'OUTPUT_PATH', /* required */
            // kmsKeyArn: 'OUTPUT_KMS_KEY_ARN' /* optional */
        }
    },
    roleArn: 'ROLE_ARN', /* required */
    solutionVersionArn: 'SOLUTION_VERSION_ARN', /* required */
    numResults: 20 /* optional integer */
};

export const run = async () => {

```

```
try {
  const response = await personalizeClient.send(new
CreateBatchInferenceJobCommand(createBatchInferenceJobParam));
  console.log("Success", response);
  return response; // For unit tests.
} catch (err) {
  console.log("Error", err);
}
};
run();
```

Die Verarbeitung des Batchauftrags kann eine Weile dauern. Sie können den Status eines Auftrags überprüfen, indem Sie [DescribeBatchInferenceJob](#) aufrufen und als Eingabeparameter einen `batchRecommendationsJobArn` übergeben. Sie können auch alle Amazon-Personalize-Stapelinferenzanträge in Ihrer AWS Umgebung auflisten, indem Sie aufrufen [ListBatchInferenceJobs](#).

Erstellen eines Batch-Inferenzauftrags, der Designs generiert

Um Designs für ähnliche Elemente zu generieren, müssen Sie das Rezept [Ähnliche Elemente](#) verwenden und Ihr Elementdatensatz muss ein Textfeld und eine Spalte mit Elementnamendaten haben. Weitere Informationen zu Empfehlungen für Designs finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#).

Der folgende Code erstellt einen Batch-Inferenzauftrag, der Empfehlungen mit Designs generiert. Behalten Sie den `batchInferenceJobMode` Wert bei `"THEME_GENERATION"`. Ersetzen Sie `COLUMN_NAME` durch den Namen der Spalte, in der Ihre Elementnamendaten gespeichert sind.

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
  solutionVersionArn = "Solution version ARN",
  jobName = "Batch job name",
  roleArn = "IAM service role ARN",
  filterArn = "Filter ARN",
  batchInferenceJobMode = "THEME_GENERATION",
  themeGenerationConfig = {
    "fieldsForThemeGeneration": {
      "itemName": "COLUMN_NAME"
```



```

    }
  },
  jobInput =
    {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input
JSON file name>.json"}}},
  jobOutput =
    {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder
name>/"}}}
)

```

Beispiele für die Ausgabe von Batch-Inferenz-Jobs

Wenn Sie einen Batch-Inferenzjob erstellen, importiert der Job Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket, verwendet Ihre Lösungsversion, um Artikelempfehlungen zu generieren, und exportiert die Empfehlungen im JSON-Format in einen Amazon S3 S3-Bucket.

In den folgenden Abschnitten sind Beispiele für Ausgabedateien für Batch-Inferenzjobs nach Rezepttyp aufgeführt. Mit den Rezepten „Trending-Now“ oder „Next-Best-Action“ können Sie keine Batch-Empfehlungen erhalten.

Themen

- [USER_PERSONALIZATION-Rezepte](#)
- [POPULAR_ITEMS Rezepte](#)
- [Personalisierte-Rangfolgen-Rezepte](#)
- [RELATED_ITEMS-Rezepte](#)

USER_PERSONALIZATION-Rezepte

Im Folgenden finden Sie ein Beispiel für die JSON-Ausgabedatei für ein USER_PERSONALIZATION-Rezept.

```

{"input":{"userId":"4638"},"output":{"recommendedItems":
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","62374",
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.004
{"input":{"userId":"663"},"output":{"recommendedItems":
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104","4
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.008
{"input":{"userId":"3384"},"output":{"recommendedItems":
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035","2
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.005

```

...

POPULAR_ITEMS Rezepte

Das folgende Beispiel zeigt das Format der JSON-Ausgabedatei für das Popularity-Count-Rezept. Mit dem Trending-Now-Rezept können Sie keine Batch-Empfehlungen erhalten.

```
{ "input": { "userId": "12"}, "output": { "recommendedItems": ["105", "106", "441"] } }
{ "input": { "userId": "105"}, "output": { "recommendedItems": ["105", "106", "441"] } }
{ "input": { "userId": "41"}, "output": { "recommendedItems": ["105", "106", "441"] } }
...
```

Personalisierte-Rangfolgen-Rezepte

Das folgende Beispiel zeigt das Format der JSON-Ausgabedatei für ein PERSONALIZED_RANKING-Rezept.

```
{ "input": { "userId": "891", "itemList": ["27", "886", "101"] }, "output": { "recommendedItems": ["27", "101", "886"], "scores": [0.48421, 0.28133, 0.23446] } }
{ "input": { "userId": "445", "itemList": ["527", "55", "901"] }, "output": { "recommendedItems": ["901", "527", "55"], "scores": [0.46972, 0.31011, 0.22017] } }
{ "input": { "userId": "71", "itemList": ["29", "351", "199"] }, "output": { "recommendedItems": ["351", "29", "199"], "scores": [0.68937, 0.24829, 0.06232] } }
...
```

RELATED_ITEMS-Rezepte

Das folgende Beispiel zeigt das Format der JSON-Ausgabedatei für ein RELATED_ITEMS-Rezept.

```
{ "input": { "itemId": "105"}, "output": { "recommendedItems": ["106", "107", "49"] } }
{ "input": { "itemId": "106"}, "output": { "recommendedItems": ["105", "107", "49"] } }
{ "input": { "itemId": "441"}, "output": { "recommendedItems": ["2", "442", "435"] } }
...
```

Das folgende Beispiel zeigt das Format der JSON-Ausgabedatei für das Similar-Items-Rezept mit Themen. Weitere Informationen zu Empfehlungen mit Themen finden Sie unter [Batch-Empfehlungen mit Themen aus dem Content Generator](#)

```
{ "input": { "itemId": "40"}, "output": { "recommendedItems": ["36", "50", "44", "22", "21", "29", "3", "1", "2", "39"], "theme": "Movies with a strong female lead", "itemsThemeRelevanceScores": [0.19994527, 0.183059963, 0.17478035, 0.1618133, 0.1574806, 0.15468733, 0.1499242, 0.14353688, 0.135314
```

```
{"input":{"itemId":"43"},"output":{"recommendedItems":  
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of  
1995","itemsThemeRelevanceScores":  
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.138913,-  
...}
```

Abrufen von Benutzersegmenten

Um Benutzersegmente abzurufen, verwenden Sie einen Batch-Segmentauftrag. Ein Batch-Segmentauftrag ist ein Tool, das Ihre Batch-Eingabedaten aus einem Amazon S3-Bucket importiert und Ihre Lösungsversion verwendet, die mit einem USER_SEGMENTATION-Rezept trainiert wurde, um Benutzersegmente für jede Zeile mit Eingabedaten zu generieren.

Je nach Rezept sind die Eingabedaten eine Liste von Elementen oder Elementmetadatenattributen im JSON-Format. Bei Elementattributen können Ihre Eingabedaten Ausdrücke enthalten, um Benutzersegmente basierend auf mehreren Metadatenattributen zu erstellen. Ein Batch-Segmentauftrag exportiert Benutzersegmente in einen Amazon S3-Ausgabe-Bucket. Jedes Benutzersegment wird in absteigender Reihenfolge sortiert, basierend auf der Wahrscheinlichkeit, dass jeder Benutzer mit dem Element in Ihren Eingabedaten interagiert.

Beim Generieren von Benutzersegmenten berücksichtigt Amazon Personalize Daten in Datensätzen aus Massen- und einzelnen Importen:

- Für Massendaten generiert Amazon Personalize Segmente, die nur die Massendaten verwenden, die beim letzten vollständigen Training der Lösungsversion vorhanden waren. Außerdem werden nur Massendaten verwendet, die Sie mit dem Importmodus FULL importiert haben (ersetzen vorhandener Daten).
- Für Daten aus einzelnen Datenimportvorgängen generiert Amazon Personalize Benutzersegmente anhand der Daten, die beim letzten vollständigen Training der Lösungsversion vorhanden waren. Damit sich neuere Datensätze auf Benutzersegmente auswirken, erstellen Sie eine neue Lösungsversion und anschließend einen Batch-Segmentauftrag.

Das Generieren von Benutzersegmenten funktioniert wie folgt:

1. Bereiten Sie Ihre Eingabedaten im JSON-Format vor und laden Sie sie in einen Amazon S3-Bucket hoch. Das Format Ihrer Eingabedaten hängt vom verwendeten Rezept und dem Auftrag ab, den Sie erstellen. Siehe [Vorbereiten von Eingabedaten für Benutzersegmente](#).

2. Erstellen Sie einen separaten Speicherort für Ihre Ausgabedaten, entweder einen anderen Ordner oder einen anderen Amazon S3-Bucket.
3. Erstellen Sie einen Batch-Segmentauftrag. Siehe [Erstellen eines Batch-Segmentauftrags](#).
4. Wenn der Batch-Segmentauftrag abgeschlossen ist, rufen Sie die Benutzersegmente von Ihrem Ausgabespeicherort in Amazon S3 ab.

Themen

- [Richtlinien und Anforderungen](#)
- [Vorbereiten von Eingabedaten für Benutzersegmente](#)
- [Erstellen eines Batch-Segmentauftrags](#)
- [Beispiele für die Ausgabe von Batch-Segment-Jobs](#)

Richtlinien und Anforderungen

Im Folgenden finden Sie Richtlinien und Anforderungen für den Batch-Abruf von Batch-Segmenten:

- Sie müssen ein USER_SEGMENTATION-Rezept verwenden.
- Ihre Amazon-Personalize-IAM-Servicerolle benötigt die Berechtigung zum Lesen und Hinzufügen von Dateien zu Ihren Amazon S3-Buckets. Informationen zum Erteilen von Berechtigungen finden Sie unter [Richtlinie für Servicerollen für Batch-Workflows](#). Weitere Informationen zu Bucket-Berechtigungen finden Sie unter [Beispiele für Benutzerrichtlinien](#) im Amazon Simple Storage Service-Entwicklerhandbuch.

Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon-Personalize-IAM-Servicerolle die Berechtigung zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

- Sie müssen eine benutzerdefinierte Lösung und Lösungsversion erstellen, bevor Sie einen Batch-Inferenzauftrag erstellen. Sie müssen jedoch keine Amazon-Personalize-Kampagne erstellen. Wenn Sie eine Domain-Datensatzgruppe erstellt haben, können Sie dennoch benutzerdefinierte Ressourcen erstellen.
- Ihre Eingabedaten müssen wie unter beschrieben formatiert sein [Vorbereiten von Eingabedaten für Benutzersegmente](#).
- Wenn Sie das Item-Attribute-Affinity-Rezept verwenden, dürfen die Attribute in Ihren Eingabedaten keine unstrukturierten Metadaten für Textelemente enthalten, z. B. eine Produktbeschreibung.

- Wenn Sie einen Filter mit Platzhalterparametern verwenden, müssen Sie die Werte für die Parameter in Ihren Eingabedaten in ein `filterValues` Objekt aufnehmen. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).
- Wir empfehlen Ihnen, für Ihre Ausgabedaten einen anderen Speicherort (entweder einen Ordner oder einen anderen Amazon S3-Bucket) als Ihre Eingabedaten zu verwenden.

Vorbereiten von Eingabedaten für Benutzersegmente

Batch-Segmentaufträge verwenden eine Lösungsversion, um Benutzersegmente auf der Grundlage von Daten zu erstellen, die Sie in einer JSON-Eingabedatei angeben. Bevor Sie Benutzersegmente abrufen können, müssen Sie Ihre JSON-Datei vorbereiten und in einen Amazon S3-Bucket hochladen. Wir empfehlen Ihnen, einen Ausgabeordner in Ihrem Amazon S3-Bucket zu erstellen oder einen separaten Amazon S3-Ausgabe-Bucket zu verwenden. Anschließend können Sie mehrere Batch-Inferenzaufträge mit demselben Eingabedatenspeicherort ausführen.

Wenn Sie einen Filter mit Platzhalterparametern verwenden, z. B. `$GENRE`, müssen Sie die Werte für die Parameter in einem `filterValues` Objekt in Ihrem Eingabe-JSON angeben. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).

So bereiten Sie Daten vor und importieren sie

1. Formatieren Sie Ihre Batch-Eingabedaten je nach dem Rezept, das Ihre Lösung verwendet. Trennen Sie das Eingabedatenelement durch eine neue Zeile. Ihre Eingabedaten sind entweder eine Liste von `itemIds` (Item-Affinity) oder Elementattributen (Item-Attribute-Affinity).

Bei Elementattributen können Eingabedaten logische Ausdrücke mit dem `-AND` Operator enthalten, um Benutzer für mehrere Elemente oder Attribute pro Abfrage abzurufen. Weitere Informationen finden Sie unter [Angeben von Elementattributen für das Item-Attribute-Affinity-Rezept](#).

Beispiele für Eingabedaten für beide Rezepte finden Sie unter [Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON](#).

2. Laden Sie Ihren Eingabe-JSON in einen Eingabeordner in Ihrem Amazon S3-Bucket hoch. Weitere Informationen finden Sie unter [Hochladen von Dateien und Ordnern per Drag & Drop](#) im Benutzerhandbuch für Amazon Simple Storage Service.
3. Erstellen Sie einen separaten Speicherort für Ihre Ausgabedaten, entweder einen Ordner oder einen anderen Amazon S3-Bucket. Indem Sie einen separaten Speicherort für das

Ausgabe-JSON erstellen, können Sie mehrere Batch-Segmentaufträge mit demselben Eingabedatenspeicherort ausführen.

Nachdem Sie Ihre Eingabedaten vorbereitet und in einen Amazon S3-Bucket hochgeladen haben, können Sie Benutzersegmente mit einem Batch-Segmentauftrag generieren. Weitere Informationen finden Sie unter [Erstellen eines Batch-Segmentauftrags](#).

Themen

- [Angaben von Elementattributen für das Item-Attribute-Affinity-Rezept](#)
- [Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON](#)

Angaben von Elementattributen für das Item-Attribute-Affinity-Rezept

Wenn Sie das Item-Attribute-Affinity-Rezept verwenden, sind Ihre Eingabedaten eine Liste von Elementattributen. Sie können verschiedene Metadaten spalten mischen. Beispielsweise kann eine Zeile eine numerische Spalte und die nächste eine kategoriale Spalte sein. Sie können Metadaten unstrukturierter Textelemente nicht als Elementattribut verwenden.

Ihre Metadaten für Eingabeelemente können logische Ausdrücke mit dem -AND-Operator enthalten, um ein Benutzersegment für mehrere Attribute abzurufen. Eine Zeile Ihrer Eingabedaten könnte beispielsweise `{"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.genres = "\Action\""} oder sein {"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.audience = "\teen\""}.`

Wenn Sie zwei Attribute mit dem -AND-Operator kombinieren, erstellen Sie ein Benutzersegment mit Benutzern, bei denen es wahrscheinlicher ist, dass sie mit Elementen interagieren, die beide Attribute basierend auf dem Interaktionsverlauf der Benutzer haben. Im Gegensatz zu Filterausdrücken (die den -IN-Operator für die Zeichenfolgengleichheit verwenden), unterstützen Batch-Segmenteingabeausdrücke nur das = Symbol für die Gleichheit für den Zeichenfolgenabgleich.

Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON

Für einen Batch-Segmentauftrag müssen Ihre Eingabedaten entweder eine Liste von itemIds (Item-Affinity-Rezept) oder Elementattributen (Item-Attribute-Affinity) sein. Jede Zeile mit Eingabedaten ist eine separate Inferenzabfrage. Jedes Benutzersegment wird in absteigender Reihenfolge sortiert, basierend auf der Wahrscheinlichkeit, dass jeder Benutzer mit Elementen in Ihrem Bestand interagiert.

Wenn Sie einen Filter mit Platzhalterparametern verwenden, z. B. \$GENRE, müssen Sie die Werte für die Parameter in einem `filterValues` Objekt in Ihrem Eingabe-JSON angeben. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).

Im Folgenden finden Sie korrekt formatierte JSON-Eingabe- und Ausgabebeispiele für Batch-Segmentaufträge, die nach Rezepten organisiert sind.

Element-Affinität

Input

Ihre Eingabedaten können maximal 500 Elemente enthalten. Trennen Sie jede `itemId` Zeile wie folgt durch eine neue Zeile.

```
{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...
```

Output

```
{"input": {"itemId": "105"}, "output": {"recommendedUsers": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedUsers": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedUsers": ["2", "442", "435"]}}
...
```

Element-Attribut-Affinität

Input

Ihre Eingabedaten können maximal 10 Abfragen haben, wobei jede Abfrage ein oder mehrere nicht textbasierte Elementattribute ist. Trennen Sie jedes Attribut oder jeden Attributausdruck wie folgt durch eine neue Zeile.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}
{"itemAttributes": "ITEMS.genres = \"Comedy\""}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\""}
...
```

Output

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"",
  "output": {"recommendedUsers": ["25", "78", "108"]}}
{"itemAttributes": "ITEMS.genres = \"Adventure\"", "output": {"recommendedUsers":
  ["87", "31", "129"]}}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"",
  "output": {"recommendedUsers": ["8", "442", "435"]}}
...
```

Erstellen eines Batch-Segmentauftrags

Wenn Sie ein `USER_SEGMENTATION`-Rezept verwendet haben, können Sie Batch-Segmentaufträge erstellen, um Benutzersegmente mit Ihrer Lösungsversion abzurufen. Jedes Benutzersegment wird in absteigender Reihenfolge sortiert, basierend auf der Wahrscheinlichkeit, dass jeder Benutzer mit Elementen in Ihrem Bestand interagiert. Je nach Rezept müssen Ihre Eingabedaten eine Liste von Elementen ([Item-Affinity-Rezept](#)) oder Elementattributen ([Item-Attribute-Affinity-Rezept](#)) im JSON-Format sein. Sie können einen Batch-Segmentauftrag mit der Amazon Personalize-Konsole, der AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.

Wenn Sie einen Batch-Segmentauftrag erstellen, geben Sie die Amazon S3-Pfade zu Ihren Eingabe- und Ausgabespeicherorten an. Amazon S3 basiert auf Präfixen. Wenn Sie ein Präfix für den Speicherort der Eingabedaten angeben, verwendet Amazon Personalize alle Dateien, die diesem Präfix entsprechen, als Eingabedaten. Wenn Sie beispielsweise angeben `s3://<name of your S3 bucket>/folderName` und Ihr Bucket auch einen Ordner mit dem Pfad `hats3://<name of your S3 bucket>/folderName_test`, verwendet Amazon Personalize alle Dateien in beiden Ordnern als Eingabedaten. Um nur die Dateien in einem bestimmten Ordner als Eingabedaten zu verwenden, beenden Sie den Amazon S3-Pfad mit einem Präfixtrennzeichen, z. B. `./: s3://<name of your S3 bucket>/folderName/`. Weitere Informationen darüber, wie Amazon S3 Objekte organisiert, finden Sie unter [Organisieren, Auflisten und Arbeiten mit Ihren Objekten](#).

Themen

- [Erstellen eines Batch-Segmentauftrags \(Konsole\)](#)
- [Erstellen eines Batch-Segmentauftrags \(AWS CLI\)](#)
- [Erstellen eines Batch-Segmentauftrags \(AWS SDKs\)](#)

Erstellen eines Batch-Segmentauftrags (Konsole)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie einen Batch-Segmentauftrag erstellen. Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits eine Lösung und eine Lösungsversion (trainiertes Modell) mit einem USER_SEGEMENTATION-Rezept erstellt haben.

So erstellen Sie einen Batch-Segmentauftrag (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppe Ihre Datensatzgruppe aus.
3. Wählen Sie Batch-Segmentaufträge im Navigationsbereich und dann Batch-Segmentauftrag erstellen aus.
4. Geben Sie in den Details des Batchsegmentauftrags für Batchsegmentauftragsname einen Namen für Ihren Batchsegmentauftrag an.
5. Wählen Sie für Lösung die Lösung und dann die Lösungsversions-ID aus, die Sie zum Generieren der Empfehlungen verwenden möchten. Sie können Batch-Segmentaufträge nur erstellen, wenn Sie ein USER_SEGEMENTATION-Rezept verwendet haben.
6. Geben Sie für Anzahl der Benutzer optional die Anzahl der Benutzer an, die Amazon Personalize für jedes Benutzersegment generiert. Der Standardwert ist 25. Das Maximum beträgt 5 Millionen.
7. Geben Sie für Eingabequelle den Amazon S3-Pfad zu Ihrer Eingabedatei an oder verwenden Sie S3, um Ihren Amazon S3-Bucket auszuwählen.

Verwenden Sie die folgende Syntax: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json**

Ihre Eingabedaten müssen das richtige Format für das Rezept haben, das Ihre Lösung verwendet. Beispiele für Eingabedaten finden Sie unter [Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON](#).

8. Geben Sie für Ausgabeziel den Pfad zu Ihrem Ausgabespeicherort an oder verwenden Sie S3 um Ihren Amazon S3Bucket auszuwählen. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket).

Verwenden Sie die folgende Syntax: **s3://<name of your S3 bucket>/<output folder name>/**

9. Wählen Sie für IAM-Rolle eine der folgenden Optionen aus:

- Wählen Sie Neue Servicerolle erstellen und verwenden und geben Sie den Namen der Servicerolle ein, um eine neue Rolle zu erstellen, oder
- Wenn Sie bereits eine Rolle mit den richtigen Berechtigungen erstellt haben, wählen Sie Vorhandene Servicerolle verwenden und wählen Sie die IAM-Rolle aus.

Die von Ihnen verwendete Rolle muss über Lese- und Schreibzugriff auf Ihre Amazon S3-Eingabe- bzw. Ausgabe-Buckets verfügen.

10. Wählen Sie für Filterkonfiguration optional einen Filter aus, um einen Filter auf die Benutzersegmente anzuwenden. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#).
11. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Markieren von Amazon-Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).
12. Wählen Sie Batch-Segmentauftrag erstellen aus. Die Erstellung von Batchsegmentaufträgen beginnt und die Seite Batchsegmentaufträge wird angezeigt, auf der der Abschnitt Details zum Batchsegmentauftrag angezeigt wird.
13. Wenn sich der Status des Batch-Segment-Auftrags in Aktiv ändert, können Sie die Ausgabe des Auftrags aus dem angegebenen Amazon S3-Ausgabe-Bucket abrufen. Der Name der Ausgabedatei entspricht dem Format *input-name*.out.

Erstellen eines Batch-Segmentauftrags (AWS CLI)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie einen Batch-Segmentauftrag mit dem folgenden `create-batch-segment-job` Code erstellen. Geben Sie einen Auftragsnamen an, ersetzen Sie durch `solution version ARN` den Amazon-Ressourcennamen (ARN) Ihrer Lösungsversion und ersetzen Sie durch `IAM service role ARN` den ARN der IAM-Servicerolle, die Sie während der Einrichtung für Amazon Personalize erstellt haben. Diese Rolle muss über Lese- und Schreibzugriff auf Ihre Amazon Amazon S3-Eingabe- bzw. -Ausgabe-Buckets verfügen. `num-results` Geben Sie für die Anzahl der Benutzer an, die Amazon Personalize für jede Zeile der Eingabedaten vorhersagen soll. Der Standardwert ist 25. Das Maximum beträgt 5 Millionen. Geben Sie optional ein `anfilter-arn`, um Benutzersegmente zu filtern. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#).

Ersetzen Sie S3 `input path` und S3 `output path` durch den Amazon S3-Pfad zu Ihren Eingabedatei- und Ausgabespeicherorten. Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket). Verwenden Sie die folgende Syntax für Eingabe- und Ausgabespeicherorte: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json** und **s3://<name of your S3 bucket>/<output folder name>/**.

```
aws personalize create-batch-segment-job \
    --job-name Job name \
    --solution-version-arn Solution version ARN \
    --num-results The number of predicted users \
    --filter-arn Filter ARN \
    --job-input s3DataSource={path=s3://S3 input path} \
    --job-output s3DataDestination={path=s3://S3 output path} \
    --role-arn IAM service role ARN
{
  "batchSegmentJobArn": "arn:aws:personalize:us-west-2:acct-id:batch-segment-job/
batchSegmentJobName"
}
```

Erstellen eines Batch-Segmentauftrags (AWS SDKs)

Nachdem Sie abgeschlossen haben [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#), können Sie mit der Operation einen Batch-Segmentauftrag erstellen `CreateBatchSegmentJob`. Der folgende Code zeigt, wie Sie einen Batch-Segmentauftrag erstellen. Geben Sie dem Auftrag einen Namen, geben Sie den Amazon-Ressourcennamen (ARN) der zu verwendenden Lösungsversion an, geben Sie den ARN für Ihre Amazon-Personalize-IAM-Rolle an und geben Sie den Amazon S3-Pfad zu Ihrer Eingabedatei und Ihren Ausgabespeicherorten an. Ihre IAM-Servicerolle muss über Lese- und Schreibzugriff auf Ihre Amazon S3-Eingabe- bzw. Ausgabe-Buckets verfügen.

Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket). Verwenden Sie die folgende Syntax für Eingabe- und Ausgabespeicherorte: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json** und **s3://<name of your S3 bucket>/<output folder name>/**.

Geben Sie für die Anzahl der Benutzer `numResults`, die Amazon Personalize für jede Zeile der Eingabedaten vorhersagen soll. Der Standardwert ist 25. Das Maximum beträgt 5 Millionen. Geben Sie optional ein `filterArn`, um Benutzersegmente zu filtern. Wenn Ihr Filter Platzhalterparameter verwendet, stellen Sie sicher, dass die Werte für die Parameter in

Ihrem Eingabe-JSON enthalten sind. Weitere Informationen finden Sie unter [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#).

SDK for Python (Boto3)

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_segment_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Job name",
    numResults = Number of predicted users,
    filterArn = "Filter ARN",
    roleArn = "IAM service role ARN",
    jobInput =
        {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input
JSON file name>.json"}}},
    jobOutput =
        {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder
name>/"}}}}
)
```

SDK for Java 2.x

```
public static String createBatchSegmentJob(PersonalizeClient personalizeClient,
                                           String solutionVersionArn,
                                           String jobName,
                                           String filterArn,
                                           int numResults,
                                           String
s3InputDataSourcePath,
                                           String
s3DataDestinationPath,
                                           String roleArn,
                                           String explorationWeight,
                                           String
explorationItemAgeCutOff) {
    long waitInMilliseconds = 60 * 1000;
    String status;
    String batchSegmentJobArn;
}
```

```
try {
    // Set up data input and output parameters.
    S3DataConfig inputSource = S3DataConfig.builder()
        .path(s3InputDataSourcePath)
        .build();
    S3DataConfig outputDestination = S3DataConfig.builder()
        .path(s3DataDestinationPath)
        .build();

    BatchSegmentJobInput jobInput = BatchSegmentJobInput.builder()
        .s3DataSource(inputSource)
        .build();
    BatchSegmentJobOutput jobOutputLocation = BatchSegmentJobOutput.builder()
        .s3DataDestination(outputDestination)
        .build();

    CreateBatchSegmentJobRequest createBatchSegmentJobRequest =
    CreateBatchSegmentJobRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .filterArn(filterArn)
        .jobInput(jobInput)
        .jobOutput(jobOutputLocation)
        .jobName(jobName)
        .numResults(numResults)
        .roleArn(roleArn)
        .build();

    batchSegmentJobArn =
    personalizeClient.createBatchSegmentJob(createBatchSegmentJobRequest)
        .batchSegmentJobArn();
    DescribeBatchSegmentJobRequest describeBatchSegmentJobRequest =
    DescribeBatchSegmentJobRequest.builder()
        .batchSegmentJobArn(batchSegmentJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    // wait until the batch segment job is complete.
    while (Instant.now().getEpochSecond() < maxTime) {

        BatchSegmentJob batchSegmentJob = personalizeClient
            .describeBatchSegmentJob(describeBatchSegmentJobRequest)
            .batchSegmentJob();
    }
}
```

```
        status = batchSegmentJob.status();
        System.out.println("batch segment job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return batchSegmentJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchSegmentJobCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch segment job's parameters.

export const createBatchSegmentJobParam = {
    jobName: 'NAME',
    jobInput: {          /* required */
        s3DataSource: { /* required */
            path: 'INPUT_PATH', /* required */
            // kmsKeyArn: 'INPUT_KMS_KEY_ARN' /* optional */
        }
    },
    jobOutput: {        /* required */
        s3DataDestination: { /* required */
```

```
    path: 'OUTPUT_PATH', /* required */
    // kmsKeyArn: 'OUTPUT_KMS_KEY_ARN' /* optional */
  }
},
roleArn: 'ROLE_ARN', /* required */
solutionVersionArn: 'SOLUTION_VERSION_ARN', /* required */
numResults: 20 /* optional */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateBatchSegmentJobCommand(createBatchSegmentJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Die Verarbeitung des Batchauftrags kann eine Weile dauern. Sie können den Status eines Auftrags überprüfen, indem Sie [DescribeBatchSegmentJob](#) aufrufen und als Eingabeparameter einen `batchSegmentJobArn` übergeben. Sie können auch alle Amazon-Personalize-Stapelsegmentaufträge in Ihrer AWS Umgebung auflisten, indem Sie aufrufen [ListBatchSegmentJobs](#).

Beispiele für die Ausgabe von Batch-Segment-Jobs

Ein Batch-Segmentjob importiert Ihre Batch-Eingabedaten aus einem Amazon S3 S3-Bucket und verwendet Ihre Lösungsversion, die mit einem `USER_SEGMENTATION`-Rezept trainiert wurde, um zu generieren Benutzersegmente und exportiert die Segmente in einen Amazon S3 S3-Bucket.

In den folgenden Abschnitten sind korrekt formatierte JSON-Ausgabebeispiele für Batch-Segment-Jobs nach Rezept aufgeführt.

Themen

- [Artikel-Affinität](#)
- [Artikel-Attribut-Affinität](#)

Artikel-Affinität

Im folgenden Beispiel wird das Format dieses Formats gezeigt, in dem die JSON-Ausgabedatei für das Item-Affinity-Rezept gezeigt wird.

```
{"input": {"itemId": "105"}, "output": {"recommendedUsers": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedUsers": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedUsers": ["2", "442", "435"]}}
...
```

Artikel-Attribut-Affinität

Das folgende Beispiel zeigt das Format der JSON-Ausgabedatei für das Item-Attribute-Affinity-Rezept.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"", "output": {"recommendedUsers": ["25", "78", "108"]}}
{"itemAttributes": "ITEMS.genres = \"Adventure\"", "output": {"recommendedUsers": ["87", "31", "129"]}}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"", "output": {"recommendedUsers": ["8", "442", "435"]}}
...
```


Aufrechterhaltung der Relevanz von Empfehlungen

Relevante Empfehlungen können die Benutzerinteraktion, die Klickrate und die Konvertierungsrate für Ihre Anwendung erhöhen, wenn Ihr Katalog wächst. Um die Relevanz von Amazon-Personalize-Empfehlungen für Ihre Benutzer aufrechtzuerhalten und zu verbessern, sollten Sie Ihre Daten und benutzerdefinierten Ressourcen auf dem neuesten Stand halten. Auf diese Weise kann Amazon Personalize aus dem neuesten Verhalten Ihres Benutzers lernen und Ihre neuesten Elemente in Empfehlungen aufnehmen.

Themen

- [Aktualisierungen von Datensätzen](#)
- [Verwalten von Domain-Empfehlungen](#)
- [Wartung von benutzerdefinierten Lösungen](#)

Aktualisierungen von Datensätzen

Wenn Ihr Katalog wächst, aktualisieren Sie Ihre historischen Daten mit Massen- oder individuellen Datenimportvorgängen. Weitere Informationen zum Importieren historischer Daten finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#). Informationen darüber, wie sich Daten, die Sie nach dem Training eines Modells importieren, auf Empfehlungen auswirken, finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Für Anwendungsfälle und Rezepte, die personalisierte Echtzeitempfehlungen bieten, halten Sie Ihren Datensatz zu Elementinteraktionen mit dem Verhalten Ihrer Benutzer auf dem neuesten Stand. Dazu zeichnen Sie Elementinteraktionen mit einem Ereignis-Tracker und dem PutEvents -API-Vorgang auf. Amazon Personalize aktualisiert Empfehlungen basierend auf der letzten Aktivität Ihres Benutzers, während er mit Ihrem Katalog interagiert. Weitere Informationen zur Personalisierung in Echtzeit finden Sie unter [Personalisierung in Echtzeit](#). Weitere Informationen zur Aufzeichnung von Echtzeitereignissen finden Sie unter [Ereignisse aufzeichnen](#).

Verwalten von Domain-Empfehlungen

Amazon Personalize trainiert die Modelle, die Ihre Empfehlungsgeber unterstützen, automatisch alle 7 Tage neu. Dies ist ein vollständiges erneutes Training, bei dem völlig neue Modelle basierend auf der Gesamtheit der Daten in Ihren Datensätzen erstellt werden. Wenn Sie die beim Training

verwendeten Spalten ändern, startet Amazon Personalize automatisch ein vollständiges erneutes Training der Modelle, die Ihren Empfehler unterstützen.

- Für die wichtigsten Auswahlmöglichkeiten für Sie und für Sie empfohlene Anwendungsfälle aktualisiert Amazon Personalize Ihren Empfehler, um neue Elemente für Empfehlungen zu berücksichtigen. Automatische Updates sind kein vollständiges erneutes Training, bei dem das Modell aus dem Verhalten Ihrer Benutzer lernt. Stattdessen können Sie mit automatischen Updates Amazon Personalize Ihre neuen Elemente vor dem nächsten vollständigen erneuten Training des Empfehlungsgebers in Empfehlungen veröffentlichen. Informationen zu automatischen Updates finden Sie unter [Automatische Updates](#).
- Wenn Sie den Anwendungsfall Trending now verwenden, wertet Amazon Personalize Ihre Interaktionsdaten automatisch alle zwei Stunden aus und identifiziert Trendelemente. Sie müssen nicht warten, bis Ihr Empfehlungsgeber erneut trainiert wird.

Während das Empfehlungs-Neutraining läuft, können Sie dennoch Empfehlungen vom Empfehlungsgeber erhalten. Bis zum Abschluss des erneuten Trainings verwendet der Empfehler die vorherige Konfiguration und Modelle. Um Aktualisierungen zu verfolgen, können Sie den Zeitstempel für das neueste Empfehlungsupdate auf der Seite Empfehlungsdetails in der Amazon-Personalize-Konsole anzeigen. Oder Sie können die `LatestRecommenderUpdate` Details der [-DescribeRecommender](#) Operation anzeigen.

Wartung von benutzerdefinierten Lösungen

Pflegen Sie Ihre benutzerdefinierten Lösungen, indem Sie sie regelmäßig neu trainieren. Erstellen Sie eine neue Lösungsversion (trainieren Sie das Modell neu), um neue Elemente in Empfehlungen aufzunehmen, und aktualisieren Sie das Modell mit dem neuesten Verhalten Ihres Benutzers.

Die Häufigkeit des erneuten Trainings hängt von Ihren Geschäftsanforderungen und dem verwendeten Rezept ab. Für alle Rezepte empfehlen wir, mindestens wöchentlich eine neue Lösungsversion zu erstellen. Dadurch wird ein völlig neues Modell erstellt, das auf der Gesamtheit der Trainingsdaten aus den Datensätzen in Ihrer Datensatzgruppe basiert. Für User-Personalization müssen Sie `FULL` für ein vollständiges erneutes Training `trainingMode` auf festlegen.

Wenn Sie häufig neue Elemente hinzufügen, müssen Sie je nach Rezept möglicherweise häufiger neu trainieren:

- Wenn Sie kein Rezept mit automatischen Updates (z. B. User-Personalization oder Next-Best-Action) oder das Trending-Now-Rezept verwenden, müssen Sie eine neue Lösungsversion erstellen, damit Amazon Personalize die neuen Elemente für Empfehlungen berücksichtigt.
- Wenn Sie User-Personalization oder Next-Best-Action verwenden, aktualisiert Amazon Personalize automatisch Ihre neueste vollständig trainierte Lösungsversion, um neue Elemente für Empfehlungen zu berücksichtigen.

Automatische Updates sind kein vollständiges erneutes Training, bei dem das Modell aus dem Verhalten Ihrer Benutzer lernt. Stattdessen ermöglicht automatische Updates Amazon Personalize, Ihre neuen Elemente vor dem nächsten vollständigen erneuten Training in Empfehlungen zu veröffentlichen.

Sie sollten weiterhin wöchentlich eine neue Lösungsversion trainieren, wobei auf `trainingMode` gesetzt ist `FULL`. Wenn nicht alle zwei Stunden häufig genug ist, können Sie manuell eine Lösungsversion erstellen, bei der auf `trainingMode` gesetzt ist, `UPDATE` um diese neuen Elemente für Empfehlungen zu berücksichtigen. Denken Sie einfach daran, dass Amazon Personalize automatisch nur Ihre neueste vollständig trainierte Lösungsversion aktualisiert. Die manuell aktualisierte Lösungsversion wird in Zukunft nicht automatisch aktualisiert.

Weitere Informationen zu automatischen Updates, einschließlich zusätzlicher Richtlinien und Anforderungen, finden Sie unter [Automatische Updates](#).

- Wenn Sie Trending-Now verwenden, identifiziert Amazon Personalize automatisch die wichtigsten Trendelemente in Ihren Interaktionsdaten über einen konfigurierbaren Zeitraum. Sie müssen keine neue Lösungsversion manuell erstellen, damit Trending-Now neue Elemente aus Massen- oder inkrementellen Interaktionen seit dem letzten Training berücksichtigt. Weitere Informationen finden Sie unter [Trending-Now-Rezept](#).

Informationen zum Erstellen einer neuen Lösungsversion finden Sie unter [Eine Lösungsversion erstellen](#). Nachdem Sie eine neue Lösungsversion erstellt haben, müssen Sie die Kampagne aktualisieren, um sie bereitzustellen. Weitere Informationen finden Sie unter [Aktualisieren einer Kampagne](#).

Sie können Neustrainings- und Datenimportaufgaben automatisieren und planen, indem Sie Personalisierte Erfahrungen mit Machine Learning aufrechterhalten, einer Implementierung von -AWS Lösungen, die den Amazon-Personalize-Workflow automatisiert, einschließlich Datenimport, Training von Lösungsversionen und Batch-Workflows. Weitere Informationen finden Sie unter [Aufrechterhaltung persönlicher Erlebnisse mit Machine Learning](#).

Ereignisse aufzeichnen

Ein Ereignis ist eine Interaktion zwischen einem Benutzer und Ihrem Katalog. Dabei kann es sich um eine Interaktion mit einem Artikel handeln, z. B. wenn ein Benutzer einen Artikel kauft oder sich ein Video ansieht, oder um eine Aktion, z. B. das Ausführen der Aktion. Zum Beispiel die Beantragung einer Kreditkarte oder die Registrierung für ein Mitgliedschaftsprogramm.

Amazon Personalize kann Empfehlungen nur auf der Grundlage von Echtzeitereignisdaten, nur historischen Ereignisdaten oder einer Mischung aus beidem aussprechen. Zeichnen Sie Ereignisse in Echtzeit auf, während Ihre Kunden mit Handlungsempfehlungen interagieren. Dadurch werden Ihre Interaktionsdaten erweitert und Ihre Daten auf dem neuesten Stand gehalten. Und es informiert Amazon Personalize über die aktuellen Interessen Ihrer Nutzer, wodurch die Relevanz von Empfehlungen verbessert werden kann.

Wenn Ihr Domain-Anwendungsfall oder Ihr benutzerdefiniertes Rezept [Personalisierung in Echtzeit](#) unterstützt, verwendet Amazon Personalize Ereignisse in Echtzeit, um Empfehlungen zu aktualisieren und an das sich ändernde Interesse eines Benutzers anzupassen.

Wie Sie Ereignisse in Echtzeit aufzeichnen, hängt von der Art der Interaktionsdaten ab, die Sie importieren:

- Bei Artikelinteraktionen zeichnen Sie Ereignisse in Echtzeit mit dem [PutEvents](#) API-Vorgang auf. Amazon Personalize hängt diese Daten an die Gruppe [Datensatz mit Interaktionen zwischen Elementen](#) in Ihrer Datensatzgruppe an. Weitere Informationen finden Sie unter [Ereignisse aufzeichnen](#).
- Für Aktionsinteraktionen zeichnen Sie Ereignisse in Echtzeit mit dem [PutActionInteractions](#) API-Vorgang auf. Amazon Personalize hängt diese Daten an den Datensatz „[Aktionsinteraktionen](#)“ in Ihrer [Datensatzgruppe](#) an. Nur die PERSONALIZED_ACTIONS-Rezepte verwenden Daten zu Aktionsinteraktionen. Weitere Informationen finden Sie unter [Aufzeichnen von Aktionsinteraktionsereignissen](#).

Themen

- [Wie Ereignisse in Echtzeit Empfehlungen beeinflussen](#)
- [Interaktionsereignisse mit Objekten aufzeichnen](#)
- [Aufzeichnen von Aktionsinteraktionsereignissen](#)
- [Ereignisse für anonyme Benutzer aufzeichnen](#)

- [Event-Tracking-Dienste von Drittanbietern](#)
- [Beispielimplementierungen](#)

Wie Ereignisse in Echtzeit Empfehlungen beeinflussen

Wenn Ihr Rezept Personalisierung in Echtzeit unterstützt, verwendet Amazon Personalize nach dem Erstellen einer Empfehlung oder einer benutzerdefinierten Kampagne innerhalb von Sekunden nach dem Import neue aufgezeichnete Eventdaten für bestehende Artikel oder Aktionen. Die folgenden Anwendungsfälle und Rezepte unterstützen die Personalisierung in Echtzeit:

- [Für Sie empfohlen \(E-COMMERCE-Anwendungsfall\)](#)
- [Top-Tipps für Sie \(Anwendungsfall VIDEO_ON_DEMAND\)](#)
- [Rezept für Benutzerpersonalisierung](#)
- [Rezept für personalisiertes Ranking](#)
- [Rezept für die nächste beste Aktion](#)

Wenn Sie das Trending-Now-Rezept verwenden, berücksichtigt Amazon Personalize in konfigurierbaren Intervallen automatisch Artikel aus neuen Ereignisdaten. Sie müssen keine neue Lösungsversion erstellen. Weitere Informationen finden Sie unter [Trending-Now-Rezept](#).

Wenn der Artikel, die Aktion oder der Benutzer in der Veranstaltung neu ist, hängt die Art und Weise, wie Amazon Personalize die Daten verwendet, von Ihrem Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Interaktionsereignisse mit Objekten aufzeichnen

Ein Artikelinteraktionsereignis ist eine Interaktion zwischen einem Benutzer und einem Artikel in Ihrem Katalog. Zum Beispiel ein Benutzer, der Schuhe kauft oder sich einen Film ansieht.

Zeichnen Sie Interaktionen mit Artikeln in Echtzeit auf, während Sie Ihren Kunden Artikelempfehlungen zeigen. Auf diese Weise werden Ihre Interaktionsdaten erweitert und Ihre Daten auf dem neuesten Stand gehalten. Und es informiert Amazon Personalize über die aktuellen Interessen Ihrer Nutzer, wodurch die Relevanz von Empfehlungen verbessert werden kann.

Sie zeichnen Interaktionsereignisse mit Artikeln im Rahmen des [PutEvents](#) API-Vorgangs auf. Amazon Personalize hängt die Ereignisdaten an den Datensatz Artikelinteraktionen in Ihrer

Datensatzgruppe an. Wenn Sie zwei Ereignisse mit exakt demselben Zeitstempel und identischen Eigenschaften aufzeichnen, speichert Amazon Personalize nur eines der Ereignisse. Sie können Interaktionsereignisse mit den AWS SDKs, AWS Amplify oder AWS Command Line Interface () AWS CLI aufzeichnen.

Wenn Sie Apache Kafka verwenden, können Sie den Kafka-Konnektor für Amazon Personalize verwenden, um Artikelinteraktionen in Echtzeit an Amazon Personalize zu streamen. Weitere Informationen finden Sie unter [Kafka Connector for Amazon Personalize](#) im `personalize-kafka-connector` Github-Repository.

AWS Amplify umfasst eine JavaScript Bibliothek zum Aufzeichnen von Objektinteraktionsereignissen aus Webclient-Anwendungen und eine Bibliothek zum Aufzeichnen von Ereignissen im Servercode. Weitere Informationen finden Sie unter [Amplify — Analytics](#).

Themen

- [Anforderungen für die Aufzeichnung von Interaktionsereignissen mit Objekten und das Trainieren eines Modells](#)
- [Einen Event-Tracker für Artikelinteraktionen erstellen](#)
- [Die PutEvents Operation verwenden](#)
- [Ereignismetriken und Attributionsberichte](#)

Anforderungen für die Aufzeichnung von Interaktionsereignissen mit Objekten und das Trainieren eines Modells

Um Interaktionsereignisse mit Objekten aufzuzeichnen, benötigen Sie Folgendes:

- Eine Dataset-Gruppe, die ein Item `interactions`-Dataset umfasst, das leer sein kann. Wenn Sie die [Erste Schritte](#) Anleitung gelesen haben, können Sie dieselbe Datensatzgruppe und denselben Datensatz verwenden, den Sie erstellt haben. Informationen zum Erstellen einer Datensatzgruppe und eines Datensatzes finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).
- Einen Ereignis-Tracker.
- Ein Aufruf der [PutEvents](#) API-Operation.

Sie können mit einem leeren Datensatz für Artikelinteraktionen beginnen und, wenn Sie genügend Daten aufgezeichnet haben, das Modell nur mit neu aufgezeichneten Ereignissen trainieren. Für alle

Anwendungsfälle (Domain-Datensatzgruppen) und Rezepte (benutzerdefinierte Datensatzgruppen) müssen Ihre Interaktionsdaten vor dem Training Folgendes aufweisen:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Einen Event-Tracker für Artikelinteraktionen erstellen

Bevor Sie Ereignisse zur Artikelinteraktion aufzeichnen können, müssen Sie einen Event-Tracker für Artikelinteraktionen erstellen. Ein Event-Tracker leitet neue Ereignisdaten an den Datensatz Artikelinteraktionen in Ihrer Datensatzgruppe weiter.

Sie erstellen einen Event-Tracker mit der Amazon Personalize Personalize-Konsole oder dem [CreateEventTracker](#) API-Vorgang. Sie übergeben als Parameter den Amazon-Ressourcennamen (ARN) der Datensatzgruppe, die den Zieldatensatz „Artikelinteraktionen“ enthält. Anweisungen zum Erstellen eines Event-Trackers mit der Amazon Personalize Personalize-Konsole finden Sie unter [Einen Event-Tracker \(Konsole\) erstellen](#).

Ein Event-Tracker enthält eine Tracking-ID, die Sie als Parameter übergeben, wenn Sie den [PutEvents](#) Vorgang verwenden. Amazon Personalize hängt dann die neuen Ereignisdaten an den Datensatz Artikelinteraktionen der Datensatzgruppe an, die Sie in Ihrem Event-Tracker angeben.

Note

Sie können nur einen Event-Tracker für Artikelinteraktionen für eine Datensatzgruppe erstellen.

Python

```
import boto3

personalize = boto3.client('personalize')
```

```

response = personalize.create_event_tracker(
    name='MovieClickTracker',
    datasetGroupArn='arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup'
)
print(response['eventTrackerArn'])
print(response['trackingId'])

```

Der Event-Tracker ARN und die Tracking-ID werden angezeigt, zum Beispiel:

```

{
  "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
  "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"
}

```

AWS CLI

```

aws personalize create-event-tracker \
  --name MovieClickTracker \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup

```

Der Event-Tracker ARN und die Tracking-ID werden angezeigt, zum Beispiel:

```

{
  "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
  "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"
}

```

SDK for Java 2.x

```

public static String createEventTracker(PersonalizeClient personalizeClient,
                                       String eventTrackerName,
                                       String datasetGroupArn) {

    String eventTrackerId = null;
    String eventTrackerArn = null;
    long maxTime = 3 * 60 * 60;
    long waitInMilliseconds = 30 * 1000;
    String status;

```



```
try {
    CreateEventTrackerRequest createEventTrackerRequest =
CreateEventTrackerRequest.builder()
        .name(eventTrackerName)
        .datasetGroupArn(datasetGroupArn)
        .build();

    CreateEventTrackerResponse createEventTrackerResponse =
        personalizeClient.createEventTracker(createEventTrackerRequest);

    eventTrackerArn = createEventTrackerResponse.eventTrackerArn();
    eventTrackerId = createEventTrackerResponse.trackingId();

    System.out.println("Event tracker ARN: " + eventTrackerArn);
    System.out.println("Event tracker ID: " + eventTrackerId);

    maxTime = Instant.now().getEpochSecond() + maxTime;

    DescribeEventTrackerRequest describeRequest =
DescribeEventTrackerRequest.builder()
        .eventTrackerArn(eventTrackerArn)
        .build();

    while (Instant.now().getEpochSecond() < maxTime) {

        status =
personalizeClient.describeEventTracker(describeRequest).eventTracker().status();
        System.out.println("EventTracker status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return eventTrackerId;
}
catch (PersonalizeException e){
    System.out.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
```

```
    }  
    return eventTrackerId;  
}
```

Die PutEvents Operation verwenden

Nachdem Sie einen Datensatz mit Artikelinteraktionen und einen [Event-Tracker](#) für Ihre Datensatzgruppe erstellt haben, können Sie Ereignisse im Zusammenhang mit Artikelinteraktionen aufzeichnen. Um Interaktionsereignisse mit Artikeln aufzuzeichnen, verwenden Sie den [PutEvents](#) API-Vorgang. In den folgenden Abschnitten wird gezeigt, wie Sie ein einzelnes Ereignis aufzeichnen, wie Sie mehrere Ereignisse mit Ereigniswertdaten aufzeichnen und wie Sie Impressionsdaten in ein Ereignis einbeziehen.

Informationen zum Aufzeichnen von Ereignissen für anonyme Benutzer finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

Themen

- [Aufzeichnen eines Interaktionsereignisses mit einem einzelnen Objekt](#)
- [Aufzeichnen mehrerer Artikelinteraktionsereignisse mit Ereigniswertdaten](#)
- [Aufzeichnen von Impressionsdaten](#)

Aufzeichnen eines Interaktionsereignisses mit einem einzelnen Objekt

Das folgende Beispiel zeigt einen PutEvents Vorgang, bei dem ein Artikelinteraktionsereignis übergeben wird. Das entsprechende Schema wird zusammen mit einer Beispielzeile aus dem Datensatz Artikelinteraktionen angezeigt.

Ihre Anwendung generiert ein eindeutiges ZeichensessionId, wenn ein Benutzer Ihre Website zum ersten Mal besucht oder Ihre Anwendung verwendet. Sie müssen während der gesamten Sitzung sessionId bei allen Veranstaltungen dasselbe verwenden. Amazon Personalize verwendet dassessionId, um Ereignisse mit dem Benutzer zu verknüpfen, bevor er sich anmeldet (ist anonym). Weitere Informationen finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

Die Ereignisliste ist ein Array von [Event](#)-Objekten. Für jedes Ereignis eventType ist ein erforderlich, aber in diesem Beispiel werden eventType Daten nicht für das Training verwendet, da sie nicht im Schema enthalten sind. Sie können einen Platzhalterwert angeben, um die Anforderung zu erfüllen.

Der `trackingId` stammt aus dem Event-Tracker, in [Einen Event-Tracker für Artikelinteraktionen erstellen](#) dem Sie ihn erstellt haben. Die `userId`-, `itemId`- und `sentAt`-Parameter stimmen mit den `USER_ID`-, `ITEM_ID`- und `TIMESTAMP`-Feldern eines entsprechenden historischen Interactions-Datasets überein. Weitere Informationen finden Sie unter [Schemata](#).

Entsprechende Datensatzspalten

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP
Item interactions dataset data: user123, item-xyz, 1543631760
```

Code-Beispiel

Python

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'USER_ID',
    sessionId = 'session_id',
    eventList = [{
        'sentAt': TIMESTAMP,
        'eventType': 'eventTypePlaceholder',
        'itemId': 'ITEM_ID'
    }]
)
```

AWS CLI

```
aws personalize-events put-events \
  --tracking-id tracking_id \
  --user-id USER_ID \
  --session-id session_id \
  --event-list '[{
    "sentAt": TIMESTAMP,
    "eventType": "eventTypePlaceholder",
    "itemId": "ITEM_ID"
  }]'
```

SDK for Java 2.x

```
public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
                            String trackingId,
                            String sessionId,
                            String userId,
                            String itemId) {

    try {
        Event event = Event.builder()
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
            .itemId(itemId)
            .eventType("typePlaceholder")
            .build();

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(event)
            .build();

        int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
            .sdkHttpResponse()
            .statusCode();
        System.out.println("Response code: " + responseCode);

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Nach diesem Beispiel würden Sie mit dem Trainieren eines Modells fortfahren und dazu nur die erforderlichen Eigenschaften verwenden.

Aufzeichnen mehrerer Artikelinteraktionsereignisse mit Ereigniswertdaten

Das folgende Beispiel zeigt, wie mehrere Artikelinteraktionsereignisse mit unterschiedlichen Ereignistypen und unterschiedlichen Ereigniswerten aufgezeichnet werden.

Wenn Sie eine Lösung konfigurieren und Ihr Datensatz mit Artikelinteraktionen die Felder `EVENT_TYPE` und `EVENT_VALUE` enthält, können Sie einen bestimmten Wert als Schwellenwert festlegen, um Datensätze vom Training auszuschließen. Weitere Informationen finden Sie unter [Auswahl der für das Training verwendeten Artikelinteraktionsdaten](#).

Das Beispiel zeigt auch die Aufzeichnung einer zusätzlichen Eigenschaft `numRatings`, die von bestimmten Rezepten als Metadaten verwendet wird.

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE, EVENT_VALUE, NUM_RATINGS
Item interactions dataset: user123, movie_xyz, 1543531139, rating, 5, 12
                        user321, choc-ghana, 1543531760, like, 4
                        user111, choc-fake, 1543557118, like, 3
```

Python

```
import boto3
import json

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'user555',
    sessionId = 'session1',
    eventList = [{
        'eventId': 'event1',
        'sentAt': 1553631760,
        'eventType': 'like',
        'properties': json.dumps({
            'itemId': 'choc-panama',
            'eventValue': 4,
            'numRatings': 0
        })
    }, {
        'eventId': 'event2',
        'sentAt': 1553631782,
        'eventType': 'rating',
        'properties': json.dumps({
            'itemId': 'movie_ten',
            'eventValue': 3,
            'numRatings': 13
        })
    }
])
```

```
    ]]  
  )
```

AWS CLI

```
aws personalize-events put-events \  
  --tracking-id tracking_id \  
  --user-id user555 \  
  --session-id session1 \  
  --event-list '[{  
    "eventId": "event1",  
    "sentAt": 1553631760,  
    "eventType": "like",  
    "properties": "{\"itemId\": \"choc-panama\", \"eventValue\": \"true\"}"  
  }, {  
    "eventId": "event2",  
    "sentAt": 1553631782,  
    "eventType": "rating",  
    "properties": "{\"itemId\": \"movie_ten\", \"eventValue\": \"4\",  
\"numRatings\": \"13\"}"  
  }]'
```

SDK for Java 2.x

```
public static void putMultipleEvents(PersonalizeEventsClient  
personalizeEventsClient,  
    String trackingId,  
    String sessionId,  
    String userId,  
    String event1Type,  
    Float event1Value,  
    String event1ItemId,  
    int event1NumRatings,  
    String event2Type,  
    Float event2Value,  
    String event2ItemId,  
    int event2NumRatings) {  
  
    ArrayList<Event> eventList = new ArrayList<Event>();  
  
    try {  
        Event event1 = Event.builder()  
            .eventType(event1Type)
```

```
        .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
        .itemId(event1ItemId)
        .eventValue(event1Value)
        .properties("{\"numRatings\": " + event1NumRatings + "\"}")
        .build();

    eventList.add(event1);

    Event event2 = Event.builder()
        .eventType(event2Type)
        .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
        .itemId(event2ItemId)
        .eventValue(event2Value)
        .properties("{\"numRatings\": " + event2NumRatings + "\"}")
        .build();

    eventList.add(event2);

    PutEventsRequest putEventsRequest = PutEventsRequest.builder()
        .trackingId(trackingId)
        .userId(userId)
        .sessionId(sessionId)
        .eventList(eventList)
        .build();

    int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
        .sdkHttpResponse()
        .statusCode();

    System.out.println("Response code: " + responseCode);

} catch (PersonalizeEventsException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
}
```

Note

Die Eigenschaftenschlüssel verwenden Binnenmajuskel-Benennungen, die den Feldern im Interaktionsschema entsprechen. Wenn beispielsweise das Feld 'NUM_RATINGS' im Interactions-Schema definiert ist, sollte der Eigenschaftsschlüssel sein. `numRatings`

Aufzeichnen von Impressionsdaten

Wenn Sie das [Benutzerpersonalisierung](#) Rezept verwenden oder das Feld `IMPRESSIONS` zu Ihrem Schema für einen Datensatz in einer Domain-Datensatzgruppe hinzufügen, können Sie Impressionsdaten in Ihrem `PutEvents` Vorgang aufzeichnen. Impressions sind Listen von Elementen, die für einen Benutzer sichtbar waren, wenn er mit einem bestimmten Element interagiert (z. B. darauf klickte oder es sich ansah). Amazon Personalize verwendet Impressionsdaten als Leitfaden für die Erkundung, wobei Empfehlungen Artikel mit weniger Interaktionsdaten oder Relevanz enthalten. Informationen zu den impliziten und expliziten Impressions, die Amazon Personalize modellieren kann, finden Sie unter [Daten zu Eindrücken](#)

Important

Wenn Sie widersprüchliche implizite und explizite Impressionsdaten in Ihren `PutEvents` Anfragen angeben, verwendet Amazon Personalize standardmäßig die expliziten Impressions.

Um die Amazon Personalize-Empfehlungen aufzuzeichnen, die Sie Ihrem Nutzer als Impressionsdaten anzeigen, fügen Sie diese `recommendationId` in Ihre [PutEvents](#) Anfrage ein und Amazon Personalize leitet die impliziten Impressions auf der Grundlage Ihrer Empfehlungsdaten ab.

Um Impressionsdaten für ein Ereignis manuell aufzuzeichnen, listen Sie die Impressions im Eingabeparameter des [PutEvents](#) Befehls auf. `impression` Das folgende Codebeispiel zeigt, wie Sie `recommendationId` und `impression` in eine `PutEvents` Operation mit dem SDK for Python (Boto3) oder dem SDK for Java 2.x einbeziehen. Wenn Sie beide angeben, verwendet Amazon Personalize standardmäßig die expliziten Impressions.

SDK for Python (Boto3)

```
import boto3
```



```

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'userId',
    sessionId = 'sessionId',
    eventList = [{
        'eventId': 'event1',
        'eventType': 'rating',
        'sentAt': 1553631760,
        'itemId': 'item id',
        'recommendationId': 'recommendation id',
        'impression': ['itemId1', 'itemId2', 'itemId3']
    }]
)

```

SDK for Java 2.x

Verwenden Sie die folgende `putEvents` Methode, um ein Ereignis mit Impressionsdaten und einer RecommendationID aufzuzeichnen. Übergeben Sie für den Impressions-Parameter die Liste der ItemIDs als `ArrayList`

```

public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
                             String trackingId,
                             String sessionId,
                             String userId,
                             String eventType,
                             Float eventValue,
                             String itemId,
                             ArrayList<String> impressions,
                             String recommendationId) {

    try {
        Event event = Event.builder()
            .eventType(eventType)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
            .itemId(itemId)
            .eventValue(eventValue)
            .impression(impressions)
            .recommendationId(recommendationId)
            .build();
    }
}

```

```
PutEventsRequest putEventsRequest = PutEventsRequest.builder()
    .trackingId(trackingId)
    .userId(userId)
    .sessionId(sessionId)
    .eventList(event)
    .build();

int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
    .sdkHttpResponse()
    .statusCode();
System.out.println("Response code: " + responseCode);

} catch (PersonalizeEventsException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
}
```

Ereignismetriken und Attributionsberichte

Verwenden Sie CloudWatch Amazon-Metriken, um die Art und Anzahl der an Amazon Personalize gesendeten Ereignisse zu überwachen. Weitere Informationen finden Sie unter [Überwachung von Amazon Personalize](#).

Um CloudWatch Berichte zu erstellen, die die Wirkung von Empfehlungen aufzeigen, erstellen Sie eine Metrik-Zuordnung und zeichnen Sie Benutzerinteraktionen mit Empfehlungen in Echtzeit auf. Informationen zum Erstellen einer metrischen Zuordnung finden Sie unter [Messung der Wirkung von Empfehlungen](#)

Geben Sie für jedes Ereignis die Empfehlungs-ID der Empfehlungen an, die Sie dem Benutzer gezeigt haben. Oder geben Sie die Ereignisquelle an, z. B. eine dritte Partei. Importieren Sie diese Daten, um verschiedene Kampagnen, Empfehlungsgeber und Drittanbieter zu vergleichen. Sie können maximal 100 Quellen für die Zuordnung von Ereignissen importieren.

- Wenn Sie eine `recommendationId` angeben, bestimmt Amazon Personalize automatisch die Quellkampagne oder den Empfehlungsgeber und identifiziert ihn in Berichten in einer `EVENT_ATTRIBUTION_SOURCE`-Spalte.
- Wenn Sie beide Attribute angeben, verwendet Amazon Personalize nur `eventAttributionSource`.

- Wenn Sie keine Quelle angeben, kennzeichnet Amazon Personalize die Quelle `SOURCE_NAME_UNDEFINED` in Berichten.

Der folgende Code zeigt, wie ein `eventAttributionSource` für ein Ereignis in einem `PutEvents` Vorgang bereitgestellt wird.

```
response = personalize_events.put_events(  
    trackingId = 'eventTrackerId',  
    userId= 'userId',  
    sessionId = 'sessionId123',  
    eventList = [{  
        'eventId': 'event1',  
        'eventType': 'watch',  
        'sentAt': '1667260945',  
        'itemId': '123',  
        'metricAttribution': {  
            'eventAttributionSource': 'thirdPartyServiceXYZ'  
        }  
    }]  
)  
statusCode = response['ResponseMetadata']['HTTPStatusCode']  
print(statusCode)
```

Der folgende Code zeigt, wie ein `recommendationId` für ein Ereignis in einer `PutEvents` Operation bereitgestellt wird.

```
response = personalize_events.put_events(  
    trackingId = 'eventTrackerId',  
    userId= 'userId',  
    sessionId = 'sessionId123',  
    eventList = [{  
        'eventId': 'event1',  
        'eventType': 'watch',  
        'sentAt': '1667260945',  
        'itemId': '123',  
        'recommendationId': 'RID-12345678-1234-1234-1234-abcdefghijkl'  
    }]  
)  
statusCode = response['ResponseMetadata']['HTTPStatusCode']  
print(statusCode)
```

Aufzeichnen von Aktionsinteraktionsereignissen

Ein Aktionsereignis ist eine Interaktion zwischen einem Benutzer und einer Aktion. Zum Beispiel ein Benutzer, der sich für ein Mitgliedschaftsprogramm anmeldet oder eine Kreditkarte beantragt.

Wenn Sie ein benutzerdefiniertes Rezept für PERSONALIZED_ACTIONS verwenden, zeichnen Sie Aktionsereignisse in Echtzeit auf, während Ihre Kunden mit Handlungsempfehlungen interagieren. Dadurch werden Ihre Interaktionsdaten erweitert und Ihre Daten auf dem neuesten Stand gehalten. Es informiert Amazon Personalize auch über die aktuellen Interessen Ihres Benutzers, wodurch die Relevanz von Empfehlungen verbessert werden kann. Nur die benutzerdefinierten Rezepte von PERSONALIZED_ACTIONS verwenden Daten zu Aktionsinteraktionen.

Sie zeichnen Aktionsereignisse mit dem API-Vorgang auf. [PutActionInteractions](#) Amazon Personalize hängt diese Daten an den Datensatz „[Aktionsinteraktionen](#)“ in Ihrer [Datensatzgruppe](#) an.

Ein Aktions-Interaktionsereignis muss über ein Ereignistyp-Attribut verfügen, das eines der folgenden sein kann:

- **Ergriffen** — Erlegene Ereignisse aufzeichnen, wenn ein Benutzer eine empfohlene Maßnahme ergreift.
- **Nicht ausgeführt** — Ereignisse, die nicht ausgeführt wurden, aufzeichnen, wenn Ihr Benutzer sich bewusst dafür entscheidet, die Aktion nicht auszuführen, nachdem er sich die Aktion angesehen hat. Zum Beispiel, wenn sie Nein wählen, wenn Sie ihnen die Aktion zeigen. Ereignisse, die nicht durchgeführt wurden, können darauf hinweisen, dass der Kunde an der Aktion nicht interessiert ist.
- **Angesehen** — Angesehene Ereignisse aufzeichnen, wenn Sie einem Benutzer eine Aktion zeigen, bevor er sich entscheidet, ob er eine Aktion ausführt oder nicht. Amazon Personalize verwendet View-Ereignisse, um mehr über die Interessen Ihrer Benutzer zu erfahren. Wenn ein Benutzer beispielsweise eine Aktion betrachtet, sie aber nicht ausführt, ist dieser Benutzer möglicherweise in future nicht an dieser Aktion interessiert.

Sie können Ereignisse in Echtzeit mithilfe der AWS SDKs oder AWS Command Line Interface (AWS CLI) aufzeichnen. Wenn Sie zwei Ereignisse mit exakt demselben Zeitstempel und identischen Eigenschaften aufzeichnen, speichert Amazon Personalize nur eines der Ereignisse.

Themen

- [Anforderungen für die Aufzeichnung von Ereignissen im Zusammenhang mit Aktionen](#)
- [Finden Sie die ID Ihres Action-Interaktions-Event-Trackers](#)

- [Die PutActionInteractions Operation wird verwendet](#)

Anforderungen für die Aufzeichnung von Ereignissen im Zusammenhang mit Aktionen

Um Aktionsereignisse in Echtzeit aufzuzeichnen, benötigen Sie Folgendes:

- Eine Datensatzgruppe, die eine enthält `Action interactions dataset`, die leer sein kann. Hinweise zum Erstellen einer Datensatzgruppe und eines Datensatzes finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).
- Die ID Ihres Event-Trackers. Sie geben diese ID in der PutActionInteractions Operation an. Wenn Sie einen Action-Interaction-Datensatz erstellen, erstellt Amazon Personalize automatisch einen Action-Interaktions-Event-Tracker für Sie. Weitere Informationen finden Sie unter [Finden Sie die ID Ihres Action-Interaktions-Event-Trackers](#).
- Einen Aufruf der [PutActionInteractions](#)-Operation.

Finden Sie die ID Ihres Action-Interaktions-Event-Trackers

Wenn Sie einen Action-Interaction-Datensatz erstellen, erstellt Amazon Personalize automatisch einen Action-Interaktions-Event-Tracker für Sie. Sie geben die ID des Trackers im PutActionInteractions API-Vorgang an. Amazon Personalize leitet damit neue Daten an den Datensatz „Aktionsinteraktionen“ in Ihrer Datensatzgruppe weiter.

Sie finden die ID Ihres Event-Trackers auf der Detailseite Ihres Action-Interaktions-Datensatzes in der Amazon Personalize Personalize-Konsole. Und Sie können die ID finden, indem Sie den DescribeDataset API-Vorgang aufrufen. Der folgende Python-Code gibt die Tracking-ID für ein Action-Interaktions-Dataset zurück.

```
import boto3

personalize = boto3.client(service_name='personalize')

response = personalize.describe_dataset(
    datasetArn="Action interactions dataset ARN"
)

print(response['trackingId'])
```

```
return response['trackingId']
```

Die PutActionInteractions Operation wird verwendet

Nachdem Sie einen Datensatz mit Aktionsinteraktionen erstellt haben, können Sie Aktionsereignisse mit der [PutActionInteractions](#) Operation aufzeichnen. In den folgenden Abschnitten wird gezeigt, wie Sie ein einzelnes Ereignis aufzeichnen und wie Sie mehrere Ereignisse mit Ereigniswertdaten aufzeichnen.

Themen

- [Aufzeichnen eines Interaktionsereignisses mit einer einzelnen Aktion](#)
- [Aufzeichnen mehrerer Aktionsinteraktionsereignisse](#)

Aufzeichnen eines Interaktionsereignisses mit einer einzelnen Aktion

Der folgende Code zeigt einen PutActionInteractions Vorgang, der ein TAKEN-Ereignis übergibt. Sie können dieses Ereignis aufzeichnen, wenn Sie einem Benutzer Empfehlungen von Amazon Personalize anzeigen und dieser eine Maßnahme ergreift, z. B. Ihre Kreditkarte beantragt.

Das actionInteractions ist eine Reihe von ActionInteraction Objekten. Das trackingId stammt aus dem Event-Tracker Amazon Personalize, den Sie bei der Erstellung Ihres Action-Interaktions-Datensatzes erstellt haben. Weitere Informationen finden Sie unter [Finden Sie die ID Ihres Action-Interaktions-Event-Trackers](#).

Ihre Anwendung generiert ein eindeutiges ZeichensessionId, wenn ein Benutzer Ihre Website zum ersten Mal besucht oder Ihre Anwendung verwendet. Sie müssen während der gesamten Sitzung sessionId bei allen Veranstaltungen dasselbe verwenden. Amazon Personalize verwendet dassessionId, um Ereignisse mit dem Benutzer zu verknüpfen, bevor er sich anmeldet (ist anonym). Weitere Informationen finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

Die sentAt Parameter userIdactionId, und sind den Feldern USER_ID, ACTION_ID, EVENT_TYPE und TIMESTAMP des Action-Interaktions-Datensatzes zugeordnet.

Entsprechender Datensatz „Action-Interaktionen“

```
USER_ID, ACTION_ID, TIMESTAMP, EVENT_TYPE  
user123, action-xyz, 1543631760, TAKEN
```

Code-Beispiel

AWS CLI

```
aws personalize-events put-action-interactions \  
--tracking-id 12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxxx \  
--action-interactions '[{  
  "userId": "user123",  
  "sessionId": "abcdefg",  
  "timestamp": 1543631760,  
  "eventType": "TAKEN",  
  "actionId": "action-xyz"}]'
```

SDK for Python (Boto3)

```
import boto3  
  
personalize_events = boto3.client(service_name='personalize-events')  
  
response = personalize_events.put_action_interactions(  
    trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxxx',  
    actionInteractions=[{  
        'userId': 'user123',  
        'sessionId': 'abcdefg',  
        'timestamp': 1543631760,  
        'eventType': 'Taken',  
        'actionId': 'action-xyz'  
    }]  
)
```

Aufzeichnen mehrerer Aktionsinteraktionsereignisse

Der folgende Code zeigt, wie mehrere Aktionsinteraktionsereignisse für denselben Benutzer mit derselben sessionId aufgezeichnet werden.

Entsprechender Datensatz „Aktionsinteraktionen“

```
USER_ID, ACTION_ID, EVENT_TYPE, TIMESTAMP  
user123, action123, Taken, 1543531139  
user123, action345, Not Taken, 1543531139
```

AWS CLI

```
aws personalize-events put-action-interactions \
--tracking-id 6ddf6b7-cd83-4dd4-b09d-4c35ecbacfe1 \
--action-interactions '[{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Taken",
  "actionId": "action123"
},
{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Not Taken",
  "actionId": "action345"}]'
```

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

response = personalize_events.put_action_interactions(
    trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
    actionInteractions=[{
        'userId': 'user123',
        'sessionId': 'abcdefg',
        'timestamp': 1697848587,
        'eventType': 'Taken',
        'actionId': 'action123'
    },
    {
        'userId': 'user123',
        'sessionId': 'abcdefg',
        'timestamp': 1697848622,
        'eventType': 'Not Taken',
        'actionId': 'action345'
    }
    ])
)
```


Ereignisse für anonyme Benutzer aufzeichnen

Important

Wenn Sie nicht mindestens ein Ereignis mit einem `sessionId` und `userId` für einen Benutzer aufzeichnen, verwendet Amazon Personalize die aufgezeichnete Aktivität nicht nur für das `sessionId` Training. Und nach Abschluss der Schulung basieren die Empfehlungen nicht mehr auf Aktivitäten, die `sessionId` bis dahin verfolgt wurden.

Sie können Interaktionsereignisse für Benutzer mit Elementen oder Aktionen aufzeichnen, bevor diese ein Konto erstellen. Zeichnen Sie Ereignisse für anonyme Benutzer auf, um einen kontinuierlichen Ereignisverlauf mit Ereignissen vor und nach der Anmeldung zu erstellen. Dadurch erhält Amazon Personalize mehr Interaktionsdaten über den Benutzer, was dazu beitragen kann, relevantere Empfehlungen zu generieren.

Um Ereignisse für anonyme Benutzer (Benutzer, die sich nicht angemeldet haben) aufzuzeichnen, geben Sie für jedes Ereignis nur ein `sessionId` an. Ihre Anwendung generiert ein eindeutiges `ZeichensessionId`, wenn ein Benutzer Ihre Website zum ersten Mal besucht oder Ihre Anwendung verwendet. Sie müssen während der gesamten Sitzung `sessionId` bei allen Veranstaltungen dasselbe verwenden. Amazon Personalize verwendet dieses `sessionId`, um Ereignisse mit dem Benutzer zu verknüpfen, bevor er sich anmeldet.

Amazon Personalize verwendet beim Training keine Ereignisse von anonymen Benutzern, bis Sie sie mit einem `userId` verknüpfen. Weitere Informationen finden Sie unter [Aufbau eines kontinuierlichen Ereignisverlaufs für anonyme Benutzer](#).

Um anonyme Benutzer [in Echtzeit zu personalisieren](#), geben Sie das `sessionId` als `userId` in Ihrer [GetRecommendations](#) [GetActionRecommendations](#) OR-Anfrage an.

- Codebeispiele, die zeigen, wie Elementinteraktionsereignisse mit dem `PutEvents` Vorgang und einer `SessionID` und `userId` aufgezeichnet werden, finden Sie unter [Die PutEvents Operation verwenden](#)
- Codebeispiele, die zeigen, wie Aktionsereignisse mit dem `PutActionInteractions` Vorgang und einer `SessionID` und `userId` aufgezeichnet werden, finden Sie unter [Die PutActionInteractions Operation wird verwendet](#)

Aufbau eines kontinuierlichen Ereignisverlaufs für anonyme Benutzer

Um einen Ereignisverlauf für einen anonymen Benutzer zu erstellen und Amazon Personalize seine Ereignisse beim Training verwenden zu lassen, zeichnen Sie mindestens ein Ereignis mit a `sessionId` und a `userId` auf. Anschließend können Sie eine beliebige Anzahl von Ereignissen für die `userId` aufzeichnen. Nachdem Sie mit der Bereitstellung von begonnen haben `userId`, `sessionId` können sie sich ändern. Bei der nächsten vollständigen Umschulung verknüpft Amazon Personalize den `userId` mit dem anonymen Benutzerverlauf, der bis zum Original zurückverfolgt wurde. `sessionId`

Nach Abschluss der Umschulung basieren die Empfehlungen auf den Aktivitäten, die sowohl `sessionId` von den anonymen Ereignissen als auch von allen Ereignissen, die bis zu ihren Ereignissen verfolgt wurden. `userId`

Note

Wenn Ihr Benutzer kein Konto erstellt und Sie möchten, dass Amazon Personalize die Daten beim Training verwendet, können Sie das `sessionId` als Ereignis `userId` verwenden. Wenn der Benutzer jedoch irgendwann ein Konto erstellt, können Sie die Ereignisse aus seinem anonymen Surfen nicht mit seinem neuen `userId` Konto verknüpfen.

Event-Tracking-Dienste von Drittanbietern

Die folgenden Customer Data Platforms (CDPs) können Ihnen helfen, Veranstaltungsdaten aus Ihrer Anwendung zu sammeln und an Amazon Personalize zu senden.

- Amplitude — Sie können Amplitude verwenden, um Benutzeraktionen zu verfolgen und so das Verhalten Ihrer Benutzer besser zu verstehen. Informationen zur Verwendung von Amplitude und Amazon Personalize finden Sie im folgenden Blogbeitrag des AWS Partner Network (APN): [Measuring the Effectiveness of Personalization with Amplitude and Amazon Personalize](#).
- mParticle — Sie können mParticle verwenden, um Ereignisdaten aus Ihrer App zu sammeln. Ein Beispiel, das zeigt, wie mParticle und Amazon Personalize verwendet werden können, um personalisierte Produktempfehlungen umzusetzen, finden Sie unter [So nutzen Sie die Leistungsfähigkeit einer CDP für maschinelles Lernen: Teil 2](#).

- **Segment** — Sie können Segment verwenden, um Ihre Daten an Amazon Personalize zu senden. Weitere Informationen zur Integration von Segment mit Amazon Personalize finden Sie unter [Amazon Personalize Destination](#).

Beispielimplementierungen

Ein Beispiel für ein Jupyter-Notizbuch, das zeigt, wie Amazon Personalize verwendet wird, um auf das Echtzeitverhalten von Benutzern zu reagieren, die einen Event Tracker und den [PutEvents](#) Vorgang verwenden, finden Sie unter [2.view_campaign_and_interactions.ipynb](#) im Ordner [getting_started](#) des Repositorys. [amazon-personalize-samples](#) GitHub

Ein Beispiel, das zeigt, wie Ereignisse von Benutzern gestreamt werden, die mit Empfehlungen interagieren, finden Sie unter [streaming_events](#) im Amazon Personalize Personalize-Beispiel-Repository. GitHub

Ein vollständiges Beispiel, das den Quellcode und unterstützende Dateien für die Bereitstellung von Echtzeit-APIs enthält, die sich zwischen Ihren Amazon Personalize Personalize-Ressourcen und Client-Anwendungen befinden, finden Sie unter [Real-Time Personalization APIs](#) im AWS GitHub Beispiel-Repository. Dieses Projekt beinhaltet die Implementierung von Folgendem:

- Erfassung von Benutzerkontext und Benutzerereignissen
- Zwischenspeichern von Antworten
- Dekorationsempfehlungen auf der Grundlage von Artikelmetadaten
- A/B-Tests
- API-Authentifizierung

Daten verwalten

Nachdem Sie Daten in einen Datensatz importiert haben, können Sie ihn analysieren, in einen Amazon S3 S3-Bucket exportieren, aktualisieren oder löschen, indem Sie den Datensatz löschen. Bei Datensätzen vom Typ „Artikel“ und „Benutzer“ können Sie das Schema des Datensatzes ersetzen, um Datenspalten hinzuzufügen.

Themen

- [Aktualisieren von Daten](#)
- [Analysieren von Daten in Datensätzen](#)
- [Exportieren eines Datensatzes](#)
- [Löschen von Daten](#)

Aktualisieren von Daten

Nachdem Sie Daten in einen Amazon-Personalize-Datensatz importiert haben, können Sie sie mit Massen- oder einzelnen Datenimportvorgängen aktualisieren.

Wenn Sie einen vorhandenen Datensatz aktualisieren möchten, um zusätzliche Datenspalten hinzuzufügen, können Sie das Schema des Datensatzes durch ein neues ersetzen, das die hinzugefügten Spalten enthält. Anschließend können Sie die neuen Datenspalten importieren. Weitere Informationen finden Sie unter [Das Schema eines Datensatzes ersetzen](#).

Themen

- [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#)
- [Das Schema eines Datensatzes ersetzen](#)
- [Aktualisierung vorhandener Massendaten](#)
- [Aktualisieren von Daten mit einzelnen Importvorgängen](#)

Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken

Wenn Sie bereits eine Empfehlungs- oder benutzerdefinierte Lösungsversion erstellt haben, hängt es davon ab, ob das Element, der Benutzer oder die Aktion neu ist, wie sich neue Daten auf Echtzeitempfehlungen auswirken. Und es hängt vom Domain-Anwendungsfall oder vom benutzerdefinierten Rezept ab, das Sie verwenden.

Informationen darüber, wie sich neue Datensätze auf Batch-Empfehlungen auswirken, finden Sie unter [Abrufen von Batch-Empfehlungen](#). Informationen darüber, wie sich neue Datensätze auf Batch-Segmentaufträge auswirken, finden Sie unter [Abrufen von Benutzersegmenten](#).

Themen

- [Neue Interaktionen](#)
- [Neue Elemente](#)
- [Neuer Benutzer](#)
- [Neue -Aktionen](#)

Neue Interaktionen

Neue Interaktionen sind Element- oder Aktionsinteraktionen, die Sie nach dem letzten vollständigen erneuten Training importieren.

Wenn Interaktionen sowohl bei Echtzeitinteraktionen als auch bei Masseninteraktionen ein neues Element oder eine neue Aktion beinhalten, hängt Amazon Personalize, wenn das neue Element für Empfehlungen berücksichtigt, von Ihrem Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Neue Elemente](#) oder [Neue -Aktionen](#).

Ereignisse in Echtzeit

Für Anwendungsfälle und Rezepte mit Echtzeit-Personalisierung verwendet Amazon Personalize sofort Echtzeitinteraktionen zwischen einem Benutzer und vorhandenen Elementen oder Aktionen (Datensätze, die im letzten vollständigen Training vorhanden sind), wenn Empfehlungen für denselben Benutzer generiert werden. Weitere Informationen zur Personalisierung in Echtzeit finden Sie unter [Personalisierung in Echtzeit](#).

Für alle Domain-Anwendungsfälle und benutzerdefinierten Rezepte, die keine Echtzeit-Personalisierung bieten, z. B. die Empfehlung ähnlicher Elemente, lernt Ihr Modell erst nach dem nächsten vollständigen erneuten Training aus Echtzeit-Interaktionsdaten. Dies kann der Fall sein, wenn das wöchentliche automatische erneute Training Ihres Domain-Empfehlungsgebers abgeschlossen ist. Oder es kann sein, nachdem Sie eine neue Lösungsversion erstellt haben.

Masseninteraktionen

Bei Masseninteraktionen lernt Ihr Modell sowohl bei inkrementellen als auch bei vollständigen Datensatzimportaufträgen erst nach dem nächsten vollständigen erneuten Training aus

Massenelementinteraktions- oder Aktionsinteraktionsdaten. Massendaten werden nicht verwendet, um Empfehlungen für die Echtzeit-Personalisierung zu aktualisieren.

Ein vollständiges erneutes Training kann erfolgen, wenn das wöchentliche automatische erneute Training Ihres Domain-Empfehlungsgebers abgeschlossen ist. Oder es kann sein, nachdem Sie eine neue Lösungsversion erstellt haben. Für User-Personalization oder Next-Best-Action müssen Sie `trainingMode` auf `setzenFULL` setzen.

Weitere Informationen zum Aktualisieren vorhandener Massendaten finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Neue Elemente

Neue Elemente sind Elemente, die Sie nach dem letzten vollständigen erneuten Training importieren. Sie können entweder aus Interaktionsdaten oder Elementmetadaten in einem Elementdatensatz stammen.

Neue Elemente werden für Empfehlungen wie folgt berücksichtigt:

- Für Top-Auswahlen für Sie und Empfohlen für Ihre Domain-Fälle oder User-Personalization- oder Next-Best-Action-Rezepte aktualisiert Amazon Personalize das Modell automatisch alle zwei Stunden. Nach jedem Update berücksichtigt Amazon Personalize im Rahmen der Erkundung neue Elemente für Empfehlungen. Bei der Berücksichtigung des neuen Elements berücksichtigt Amazon Personalize alle Metadaten für das Element. Diese Daten haben jedoch erst dann größere Auswirkungen auf Empfehlungen, wenn Sie Interaktionen für das Element aufgezeichnet und vollständig neu trainiert haben. Weitere Informationen zu Updates finden Sie unter [Automatische Updates](#).
- Wenn Sie den Anwendungsfall Trending now verwenden, wertet Amazon Personalize Ihre Interaktionsdaten automatisch alle zwei Stunden aus und identifiziert Trendelemente. Sie müssen nicht warten, bis Ihr Empfehlungsgeber erneut trainiert wird. Wenn Sie das Trending-Now-Rezept verwenden, berücksichtigt Amazon Personalize automatisch alle neuen Elemente über konfigurierbare Intervalle. Sie müssen keine neue Lösungsversion manuell erstellen. Weitere Informationen zum Konfigurieren von Intervallen finden Sie unter [Trending-Now-Rezept](#).
- Wenn Sie das Trending-Now-Rezept nicht verwenden oder Ihr Anwendungsfall oder Rezept keine automatischen Updates unterstützt, berücksichtigt Amazon Personalize neue Elemente erst nach dem nächsten vollständigen erneuten Training. Dies kann nach Abschluss des wöchentlichen automatischen erneuten Trainings Ihres Domain-Empfehlungsgebers liegen. Oder es kann sein, nachdem Sie eine neue Lösungsversion erstellt haben.

Neuer Benutzer

Neue Benutzer sind Benutzer, die Sie nach dem letzten vollständigen erneuten Training importieren. Sie können entweder aus Interaktionsdaten oder Benutzermetadaten in einem Benutzerdatensatz stammen. Für neue, anonyme Benutzer (Benutzer ohne `userId` können Sie Ereignisse für den Benutzer mit einem `sessionId` und Amazon Personalize ordnet dem Benutzer Ereignisse zu, bevor er sich anmeldet. Weitere Informationen finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

Amazon Personalize generiert Empfehlungen für neue Benutzer wie folgt:

- Wenn Sie den Anwendungsfall `Trending now domain` oder das benutzerdefinierte `Trending-Now`-Rezept verwenden, erhalten neue Benutzer sofort Empfehlungen für die wichtigsten Trendelemente. Wenn Sie das `Popularity-Count`-Rezept verwenden, erhalten neue Benutzer sofort Empfehlungen für Elemente mit den meisten Interaktionen.
- Bei Rezepten oder Anwendungsfällen, die personalisierte Empfehlungen für Benutzer bereitstellen, basieren Empfehlungen für neue Benutzer auf den frühen Interaktionsverläufen Ihrer vorhandenen Benutzer. Die ersten Elemente oder Aktionen, mit denen diese vorhandenen Benutzer interagiert haben, werden neuen Benutzern eher empfohlen. Wenn Sie für die `User-Personalization`- oder `Personalized-Ranking`-Rezepte `recency_mask` auf `setzenttrue`, enthalten Empfehlungen auch Elemente, die auf den neuesten Beliebheitstrends in Ihren Interaktionsdaten basieren.

Folgendes kann die Relevanz von Empfehlungen für neue Benutzer erhöhen:

- **Interaktionsdaten** – Die primäre Möglichkeit, die Relevanz von Empfehlungen für einen neuen Benutzer zu verbessern, besteht darin, Daten aus seinen Interaktionen mit Ihren Elementen zu importieren. Informationen darüber, wie sich neue Interaktionsdaten auf Empfehlungen auswirken, finden Sie unter [Neue Interaktionen](#).
- **Benutzermetadaten** – Das Importieren von Benutzermetadaten wie `GENDER` oder `MEMBERSHIP_STATUS` kann die Empfehlungen verbessern. Damit Metadaten Empfehlungen beeinflussen, müssen Sie warten, bis das wöchentliche automatische erneute Training Ihres Domain-Empfehlungsgebers abgeschlossen ist. Oder Sie müssen manuell eine neue Lösungsversion erstellen.
- **Kontextbezogene Metadaten** – Wenn Ihr Anwendungsfall oder Rezept kontextbezogene Metadaten unterstützt und Ihr Datensatz zu Elementinteraktionen Metadatenfelder für kontextbezogene Daten enthält, können Sie den Kontext des Benutzers in Ihrer Empfehlungsanfrage angeben. Dies

erfordert kein erneutes Training. Weitere Informationen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#).

Neue -Aktionen

Neue Aktionen sind Aktionen, die Sie seit dem letzten vollständigen Training importieren. Sie können entweder aus Aktionsinteraktionsdaten oder aus Aktionen in einem Aktionsdatensatz stammen.

Mit dem Next-Best-Action-Rezept aktualisiert Amazon Personalize automatisch alle zwei Stunden eine Lösungsversion. Nach jedem Update berücksichtigt Amazon Personalize im Rahmen der Untersuchung neue Aktionen für Empfehlungen. Bei der Berücksichtigung der neuen Aktion berücksichtigt Amazon Personalize alle Metadaten für die Aktion. Diese Daten haben jedoch erst dann größere Auswirkungen auf Empfehlungen, wenn Sie Aktionsinteraktionen für die Aktion aufzeichnen und vollständig neu trainieren. Weitere Informationen zu Updates finden Sie unter [Automatische Updates](#).

Das Schema eines Datensatzes ersetzen

Nachdem Sie einen Items- oder Users-Datensatz erstellt haben, können Sie dessen Schema durch ein neues oder vorhandenes ersetzen. Sie können das Schema eines Datensatzes ersetzen, wenn sich Ihre Datenstruktur nach der Erstellung des Datensatzes geändert hat. Beispielsweise haben Sie möglicherweise eine neue Spalte mit Artikelmetadaten, die Amazon Personalize bei der Schulung berücksichtigen soll. Oder Sie möchten vielleicht eine Datenspalte hinzufügen, die nur beim Filtern von Empfehlungen verwendet werden soll.

Wenn Sie das Schema eines Datensatzes ersetzen, müssen Sie alle Felder im vorherigen Schema beibehalten und können deren Datentypen oder Attribute nicht ändern. Nachdem Sie das Schema eines Datensatzes ersetzt haben, schließt Amazon Personalize automatisch alle neuen Spalten vom Training für bestehende Empfehlungen oder benutzerdefinierte Lösungen aus. Weitere Richtlinien und Anforderungen finden Sie unter [Richtlinien und Anforderungen](#).

Sie können das Schema eines Datensatzes durch die Amazon Personalize Personalize-Konsole AWS Command Line Interface (AWS CLI) und AWS SDKs ersetzen.

Themen

- [Richtlinien und Anforderungen](#)
- [Das Schema eines Datensatzes ersetzen \(Konsole\)](#)

- [Das Schema eines Datensatzes ersetzen \(\) AWS CLI](#)
- [Ersetzen des Schemas eines Datensatzes \(AWSSDKs\)](#)

Richtlinien und Anforderungen

Bevor Sie das Schema für einen Datensatz ersetzen, stellen Sie sicher, dass Sie die folgenden Richtlinien und Anforderungen kennen:

- Sie können das Schema eines Datensatzes mit Artikelinteraktionen, eines Datensatzes mit Aktionsinteraktionen oder eines Datensatzes mit Aktionen nicht ersetzen.
- Sie können Ihrem Ersatzschema neue Felder hinzufügen, müssen jedoch alle Felder des vorherigen Schemas beibehalten. Und Sie können ihre Datentypen oder Attribute nicht ändern. Wenn das vorherige Schema beispielsweise ein MEMBERSHIP_STATUS Feld für kategoriale Zeichenkettendaten enthält, muss das neue Schema, das Sie verwenden, ein MEMBERSHIP_STATUS Feld mit diesen Attributen und Datentypen enthalten.
- Wenn das aktuelle Schema ein Feld enthält, das Sie umbenennen möchten, oder wenn Sie dessen Datentypen oder Attribute ändern möchten, können Sie ein neues Feld mit einem neuen Namen und geänderten Typen oder Attributen hinzufügen. Nehmen Sie dann das neue Feld in das Training auf und schließen Sie das alte Feld aus. Alle neuen Felder müssen null Daten unterstützen. Wenn das alte Feld keine Nulldaten unterstützte, können Sie beim Importieren von Daten Platzhalterdaten verwenden, um sicherzustellen, dass Ihr Import dem Schema entspricht. Hinweise zur Konfiguration der von einem Empfehlungsgeber verwendeten Spalten finden Sie unter [Einen Empfehlungsgeber aktualisieren](#) Hinweise zur Konfiguration der von einer Lösung verwendeten Spalten finden Sie unter [Konfiguration der beim Training verwendeten Spalten](#).
- Alle neuen Felder müssen null Daten unterstützen. Hinweise zum Hinzufügen eines Nulltyps zu einem Feld finden Sie unter [Schema-Datentypen](#).
- Nachdem Sie das Schema eines Datensatzes ersetzt haben, schließt Amazon Personalize automatisch alle neuen Spalten vom Training für bestehende Empfehlungen oder benutzerdefinierte Lösungen aus. Die Verwendung des geänderten Datensatzes umfasst die folgenden Aktionen:
 - Um neue Spalten im Training zu verwenden, importieren Sie Daten, die dem neuen Schema entsprechen. Aktualisieren Sie dann alle Empfehlungen, sodass sie alle neuen Spalten verwenden, oder erstellen Sie eine neue benutzerdefinierte Lösung und konfigurieren Sie die Spalten, die beim Training verwendet werden.

Hinweise zum Aktualisieren der von einem Empfehlungsgeber verwendeten Spalten finden Sie unter [Einen Empfehlungsgeber aktualisieren](#). Hinweise zur Konfiguration der von einer Lösung verwendeten Spalten finden Sie unter [Konfiguration der beim Training verwendeten Spalten](#).

- Um beim Filtern nur Spalten zu verwenden, importieren Sie Daten, die dem neuen Schema entsprechen, erstellen Sie einen Filter, der die neuen Daten verwendet, und wenden Sie Ihren Filter auf Ihre Empfehlungsanfragen an. Sie müssen keine Empfehlungen aktualisieren oder benutzerdefinierte Ressourcen erstellen oder aktualisieren.

Das Schema eines Datensatzes ersetzen (Konsole)

Um das Schema eines Datensatzes durch die Amazon Personalize-Konsole zu ersetzen, wählen Sie den Datensatz aus, den Sie ändern möchten, und entscheiden, ob Sie ihn durch ein neues Schema ersetzen oder ein vorhandenes verwenden möchten.

Um das Schema eines Datensatzes zu ersetzen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie Datensätze und dann das Optionsfeld für den Datensatz aus, den Sie ändern möchten.
4. Wählen Sie „Aktionen“ und anschließend „Schema ersetzen“.
5. Wählen Sie unter Schemadetails aus, ob Sie es durch ein neues oder ein zuvor erstelltes Schema ersetzen möchten.
6. Geben Sie das neue Schema an, das verwendet werden soll. Wenn Sie sich für Folgendes entschieden haben:
 - Ersetzen Sie es durch ein neues Schema, geben Sie dem Schema einen Namen und nehmen Sie in der Schemadefinition Ihre Änderungen am Schema JSON vor.
 - Verwenden Sie ein zuvor erstelltes Schema und wählen Sie dann für Zuvor erstelltes Schema das Schema aus, das Sie verwenden möchten. Es werden nur geeignete Schemas aufgeführt. Informationen zu Schemaanforderungen finden Sie unter [Richtlinien und Anforderungen](#).
7. Wählen Sie Replace (Ersetzen) aus. Wenn der Datensatz aktiv ist, können Sie mit dem Import von Daten beginnen, die dem neuen Schema entsprechen. Weitere Informationen finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Das Schema eines Datensatzes ersetzen () AWS CLI

Um das Schema eines Datensatzes durch den zu ersetzen AWS CLI, verwenden Sie den `update-dataset` Befehl, geben Sie den Amazon-Ressourcennamen (ARN) des zu aktualisierenden Datensatzes und den ARN des neuen Schemas an, das verwendet werden soll. Sie können das Schema eines Datensatzes mit Artikelinteraktionen, eines Datensatzes mit Aktionsinteraktionen oder eines Datensatzes mit Aktionen nicht aktualisieren.

Der folgende Code zeigt, wie Sie das Schema eines Datensatzes mit dem aktualisieren AWS CLI. Um das Schema eines Datensatzes durch ein neues zu ersetzen, verwenden Sie zunächst den `create-schema` Befehl. Verwenden Sie dann den folgenden Code, um das aktuelle Schema durch das neue zu ersetzen. Hinweise zum Erstellen eines Schemas mit dem AWS CLI finden Sie unter [Einen Datensatz und ein Schema erstellen \(AWS CLI\)](#). Hinweise zu Datensets und Schemaanforderungen finden Sie unter [Schemata](#).

```
aws personalize update-dataset \  
--dataset-arn Dataset ARN \  
--schema-arn New schema ARN
```

Wenn der Datensatz aktiv ist, können Sie mit dem Import von Daten beginnen, die dem neuen Schema entsprechen. Weitere Informationen finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#). Für Informationen über die letzte Aktualisierung des Datensatzes können Sie den [DescribeDataset](#) Vorgang verwenden.

Ersetzen des Schemas eines Datensatzes (AWSSDKs)

Um das Schema eines Datensatzes durch die AWS SDKs zu ersetzen, verwenden Sie die `UpdateDataset` API-Operation. Geben Sie den Amazon-Ressourcennamen (ARN) des zu aktualisierenden Datensatzes und das zu verwendende neue Schema an. Sie können das Schema eines Datensatzes mit Artikelinteraktionen, Aktionsinteraktionen oder Aktionsdatensätzen nicht aktualisieren.

Der folgende Code zeigt, wie das Schema eines Datensatzes durch das SDK for Python (Boto3) ersetzt wird. Um das Schema eines Datensatzes durch ein neues zu ersetzen, verwenden Sie zunächst die [CreateSchema](#) Operation. Verwenden Sie dann den folgenden Code, um das aktuelle Schema durch das neue zu ersetzen. Hinweise zum Erstellen eines Schemas mit den AWS SDKs finden Sie unter [Einen Datensatz und ein Schema \(AWSSDKs\) erstellen](#). Informationen zu Datensatz- und Schemaanforderungen finden Sie unter [Schemata](#).

```
import boto3

personalize = boto3.client('personalize')

update_dataset_response = personalize.update_dataset(
    datasetArn = 'dataset_arn',
    schemaArn = 'new_schema_arn'
)

print(update_dataset_response)
```

Wenn der Datensatz aktiv ist, können Sie mit dem Import von Daten beginnen, die dem neuen Schema entsprechen. Weitere Informationen finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#). Für Informationen über die letzte Aktualisierung des Datensatzes können Sie den [DescribeDataset](#) Vorgang verwenden.

Aktualisierung vorhandener Massendaten

Wenn Sie zuvor einen Datensatz-Importauftrag für einen Datensatz erstellt haben, können Sie einen anderen Importjob verwenden, um die vorhandenen Massendaten zu ergänzen oder zu ersetzen. Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Sie können die neuen Datensätze stattdessen an bestehende Daten anhängen, indem Sie den [Importmodus](#) des Jobs ändern.

Um mit einem Datensatz-Importjob Daten an einen Datensatz mit Artikelinteraktionen oder einen Datensatz mit Aktionsinteraktionen anzuhängen, müssen Sie über mindestens 1000 neue Datensätze für Artikelinteraktionen oder Aktionsinteraktionen verfügen.

Wenn Sie bereits einen Empfehlungsgeber erstellt oder eine benutzerdefinierte Lösungsversion mit einer Kampagne bereitgestellt haben, hängt die Art und Weise, wie sich neue Masseneinträge auf Empfehlungen auswirken, vom verwendeten Domain-Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Wie sich neue Daten auf Empfehlungen in Echtzeit auswirken](#).

Filtern Sie Updates nach Bulk-Datensätzen

Innerhalb von 20 Minuten nach Abschluss eines Massenimports aktualisiert Amazon Personalize alle Filter, die Sie in der Datensatzgruppe erstellt haben, mit Ihren neuen Massendaten. Mit diesem Update kann Amazon Personalize die neuesten Daten verwenden, um Empfehlungen für Ihre Benutzer zu filtern.

Themen

- [Modi importieren](#)
- [Aktualisierung von Massendatensätzen \(Konsole\)](#)
- [Massendatensätze werden aktualisiert \(AWS CLI\)](#)
- [Aktualisierung von Massendatensätzen \(AWSSDKs\)](#)

Modi importieren

Um zu konfigurieren, wie Amazon Personalize Ihre neuen Datensätze zu Ihrem Datensatz hinzufügt, geben Sie einen Importmodus für Ihren Datensatz-Importjob an:

- Um alle vorhandenen Massendaten in Ihrem Datensatz zu überschreiben, wählen Sie in der Amazon Personalize Personalize-Konsole die Option **Bestehende Daten ersetzen** oder **FULL** im [CreateDatasetImportJob](#) API-Vorgang angeben. Dadurch werden keine Daten ersetzt, die Sie einzeln importiert haben, einschließlich Ereignisse, die in Echtzeit aufgezeichnet wurden.
- Um die Datensätze an die vorhandenen Daten in Ihrem Datensatz anzuhängen, wählen Sie **Zu vorhandenen Daten hinzufügen** oder **INCREMENTAL** im [CreateDatasetImportJob](#) API-Vorgang angeben aus. Amazon Personalize ersetzt jeden Datensatz mit derselben ID durch den neuen.

Um mit einem Datensatz-Importjob Daten an einen Datensatz mit Artikelinteraktionen oder Aktionsinteraktionen anzuhängen, benötigen Sie mindestens 1000 neue Datensätze für Artikelinteraktionen oder Aktionsinteraktionen.

Wenn Sie keine Massendatensätze importiert haben, ist die Option **Zu vorhandenen Daten hinzufügen** in der Konsole nicht verfügbar, und Sie können sie nur **FULL** im [CreateDatasetImportJob](#) API-Vorgang angeben. Die Standardeinstellung ist ein vollständiger Ersatz.

Aktualisierung von Massendatensätzen (Konsole)

Wichtig

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Sie können dies ändern, indem Sie den [Importmodus](#) des Jobs angeben.

Um Massendaten mit der Amazon Personalize-Konsole zu aktualisieren, erstellen Sie einen Datensatz-Importjob für den Datensatz und geben Sie einen Importmodus an.

Um Massendatensätze zu aktualisieren (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Datensätze aus.
4. Wählen Sie auf der Seite Datensätze den Datensatz aus, den Sie aktualisieren möchten.
5. Wählen Sie unter Jobs zum Importieren von Datensätzen die Option Datensatz-Importjob erstellen aus.
6. Geben Sie unter Jobdetails importieren unter Name des Datensatz-Import-Jobs einen Namen für Ihren Importjob an.
7. Wählen Sie für den Importmodus aus, wie der Datensatz aktualisiert werden soll. Wählen Sie entweder Bestehende Daten ersetzen oder Zu vorhandenen Daten hinzufügen. Weitere Informationen finden Sie unter [Modi importieren](#).
8. Geben Sie unter Eingabequelle für S3 Location an, wo Ihre Datendatei in Amazon S3 gespeichert ist. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder path>/<CSV file name>

Wenn sich Ihre CSV-Dateien in einem Ordner in Ihrem S3-Bucket befinden und Sie mit einem Datensatz-Importauftrag mehrere CSV-Dateien in einen Datensatz hochladen möchten, verwenden Sie diese Syntax ohne den CSV-Dateinamen.

9. Wählen Sie in der IAM-Rolle aus, ob Sie entweder eine neue Rolle erstellen oder eine vorhandene verwenden möchten. Wenn Sie die Voraussetzungen erfüllt haben, wählen Sie Bestehende Servicerolle verwenden aus und geben Sie die Rolle an, in [Erstellen einer IAM-Rolle für Amazon Personalize](#) der Sie sie erstellt haben.
10. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#)
11. Wählen Sie Finish (Abschließen). Der Datenimport-Job wird gestartet und die Datensatz-Übersichtsseite wird angezeigt. Der Datensatz-Import ist abgeschlossen, wenn der Status AKTIV ist.

Massendatensätze werden aktualisiert (AWS CLI)

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Sie können dies ändern, indem Sie den [Importmodus](#) des Jobs angeben.

Verwenden Sie den `create-dataset-import-job` Befehl, um Massendaten zu aktualisieren. Geben Sie für die `FULL` `animport-mode`, ob vorhandene Daten ersetzt oder `INCREMENTAL` erweitert werden sollen. Weitere Informationen finden Sie unter [Modi importieren](#).

Der folgende Code zeigt, wie Sie einen Datensatz-Importjob erstellen, der inkrementell Daten in einen Datensatz importiert.

```
aws personalize create-dataset-import-job \  
--job-name dataset import job name \  
--dataset-arn dataset arn \  
--data-source dataLocation=s3://bucketname/filename \  
--role-arn roleArn \  
--import-mode INCREMENTAL
```

Aktualisierung von Massendatensätzen (AWSSDKs)

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Sie können dies ändern, indem Sie den [Importmodus](#) des Jobs angeben.

Um Massendaten zu aktualisieren, erstellen Sie einen Datensatz-Importjob und geben Sie einen Importmodus an. Der folgende Code zeigt, wie Sie Massendaten in Amazon Personalize mit dem SDK for Python (Boto3) oder dem SDK for Java 2.x aktualisieren.

SDK for Python (Boto3)

Verwenden Sie die Methode, um Massendaten zu aktualisieren.

`create_dataset_import_job` Geben Sie für die `FULL` `animport-mode`, ob vorhandene

Daten ersetzt oder INCREMENTAL erweitert werden sollen. Weitere Informationen finden Sie unter [Modi importieren](#).

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'INCREMENTAL'
)
```

SDK for Java 2.x

Verwenden Sie die folgende `createPersonalizeDatasetImportJob` Methode, um Massendaten zu aktualisieren. Geben Sie für die `ImportMode.FULL` an `importImport`, ob vorhandene Daten ersetzt oder `ImportMode.INCREMENTAL` erweitert werden sollen. Weitere Informationen finden Sie unter [Modi importieren](#).

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,

                                                    String jobName,
                                                    String datasetArn,
                                                    String s3BucketPath,
                                                    String roleArn,
                                                    ImportMode importMode) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
        CreateDatasetImportJobRequest.builder()
            .datasetArn(datasetArn)
            .dataSource(importDataSource)
```



```
        .jobName(jobName)
        .roleArn(roleArn)
        .importMode(importMode)
        .build();

    datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
        .datasetImportJobArn();

    DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
        .datasetImportJobArn(datasetImportJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        DatasetImportJob datasetImportJob = personalizeClient
            .describeDatasetImportJob(describeDatasetImportJobRequest)
            .datasetImportJob();

        status = datasetImportJob.status();
        System.out.println("Dataset import job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN',          /* required */
  dataSource: {
    dataLocation: 's3://<name of your S3 bucket>/<folderName>/<CSVfilename>.csv' /*
  required */
  },
  jobName: 'NAME',                    /* required */
  roleArn: 'ROLE_ARN',                /* required */
  importMode: "INCREMENTAL"          /* optional, default is FULL */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
  CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Aktualisieren von Daten mit einzelnen Importvorgängen

Nachdem Sie Daten in einen Amazon-Personalize-Datensatz importiert haben, können Sie sie aktualisieren, indem Sie zusätzliche einzelne Datensätze importieren, einschließlich Elementinteraktionen, Aktionsinteraktionen, Benutzern, Elementen oder Aktionen. Wenn Sie

Daten einzeln importieren, können Sie Ihren Amazon-Personalize-Datensätzen kleine Stapel von Datensätzen hinzufügen, wenn Ihr Katalog wächst.

Wenn Sie Datensätze einzeln importieren, hängt Amazon Personalize die neuen Datensätze an den Datensatz an. Um ein einzelnes Element, einen Benutzer oder eine Aktion zu aktualisieren, können Sie einen Datensatz mit derselben ID, aber mit den geänderten Attributen importieren. Sie können bis zu 10 Datensätze pro individuelm Importvorgang importieren.

Weitere Informationen zum Importieren von Datensätzen einzeln finden Sie unter [Einzelne Datensätze importieren](#).

Analysieren von Daten in Datensätzen

Wichtig

Sie können die Amazon-Personalize-Konsole nicht verwenden, um Daten in einem Aktionsinteraktionen- oder Aktionsdatensatz zu analysieren.

Nachdem Sie Daten in einen Elementinteraktionen-, Benutzer- oder Elementdatensatz importiert haben, können Sie die Amazon-Personalize-Konsole verwenden, um die Daten zu analysieren. Sie können über Datenerkenntnisse und Spalten- und Zeilenstatistiken mehr über Ihre Daten erfahren. Und Sie können erfahren, welche Maßnahmen Sie ergreifen können, um Ihre Daten zu verbessern. Diese Aktionen können Ihnen helfen, die Amazon-Personalize-Ressourcenanforderungen zu erfüllen, z. B. Anforderungen an das Modelltraining, oder sie können zu verbesserten Empfehlungen führen.

Nachdem Sie empfohlene Änderungen vorgenommen haben, können Sie Ihre Daten erneut importieren und prüfen, ob Sie Probleme oder verbesserte Datensatzstatistiken behoben haben. Informationen zum Aktualisieren von Daten finden Sie unter [Aktualisieren von Daten](#).

Wenn Sie keine Erkenntnisse sehen, stimmen Ihre Daten mit den Datenerwartungen von Amazon Personalize überein. Sie können Daten in einer Domain-Datensatzgruppe oder einer benutzerdefinierten Datensatzgruppe analysieren.

Bei der Generierung von Erkenntnissen und der Berechnung von Statistiken berücksichtigt Amazon Personalize alle Massen- und gestreamten Daten von nicht anonymen Benutzern. Ereignisse von anonymen Benutzern werden erst berücksichtigt, wenn Sie sie einem `zuordnenuserid` zuordnen. Weitere Informationen finden Sie unter [Ereignisse für anonyme Benutzer aufzeichnen](#).

Themen

- [Erforderliche Berechtigungen für die Analyse von Daten](#)
- [Dateneinblicke](#)
- [Anzeigen von Erkenntnissen und Statistiken zu Datensätzen](#)

Erforderliche Berechtigungen für die Analyse von Daten

Wenn Sie Benutzern vollen Zugriff auf Amazon Personalize gewähren, sind keine Berechtigungsänderungen erforderlich. Wenn Sie Ihren Benutzern nur die Berechtigungen erteilen, die zum Ausführen einer Aufgabe in Amazon Personalize erforderlich sind, muss Ihre AWS Identity and Access Management (IAM)-Richtlinie die folgenden zusätzlichen Data Insight-Aktionen enthalten.

- Personalisieren:CreateDataInsightsJob
- Personalisieren:ListDataInsightsJob
- Personalisieren:DescribeDataInsightsJob
- Personalisieren:GetDataInsight

Dateneinblicke

Im Folgenden finden Sie die möglichen Dateneinblicke, die Sie in Amazon Personalize generieren können.

Insight	Aktion	Verwandter Datensatz(e)
Der Interaktionsdatensatz enthält nur X-Interaktionen. Das Modelltraining erfordert mindestens 1 000 Interaktionen. Wir empfehlen mindestens 50.000.	Importieren Sie Y zusätzliche eindeutige Interaktionsdatensätze, bevor Sie ein Modell trainieren.	Elementinteraktionen
Der Interaktionsdatensatz hat nur X eindeutige Benutzer mit zwei oder mehr Interaktionen. Das Modelltraining erfordert	Importieren Sie jeweils mindestens 2 Interaktionsdatensätze für Y zusätzliche Benutzer.	Elementinteraktionen

Insight	Aktion	Verwandter Datensatz(e)
mindestens 25 solcher Benutzer. Wir empfehlen mindestens 1 000.		
X % der Elemente im Datensatz Elemente haben keine Interaktionen im Datensatz Interaktionen, daher werden sie möglicherweise nicht empfohlen.	Stellen Sie sicher, dass Sie alle Ihre Interaktionsdaten importieren, und suchen Sie nach nicht übereinstimmenden IDs zwischen Ihren Elementen und Interaktionsdatensätzen. Überprüfen Sie die Datensatzstatistik unten für Ihre Elemente und Interaktionsdatensätze, um sicherzustellen, dass Sie die erwartete Anzahl von Zeilen importiert haben. Wenn Ihr Anwendungsfall oder Rezept Erkundung verwendet, ändern Sie die Konfiguration der Erkundung, um mehr Elemente ohne Interaktionsdaten zu empfehlen.	Elementinteraktionen und Elemente
X % der Benutzer im Benutzerdatensatz haben keine Interaktionen im Interaktionsdatensatz. Diese Benutzer erhalten Empfehlungen für beliebte Artikel.	Stellen Sie sicher, dass Sie alle Ihre Interaktionsdaten importieren, und suchen Sie nach nicht übereinstimmenden IDs zwischen Ihren Benutzern und Interaktionsdatensätzen. Überprüfen Sie die Datensatzstatistik unten für Ihre Benutzer und Interaktionsdatensätze, um sicherzustellen, dass Sie die erwartete Anzahl von Zeilen importiert haben. Importieren Sie alle zusätzlichen Interaktionen, damit mehr Benutzer über Interaktionsdaten verfügen.	Elementinteraktionen und Benutzer

Insight	Aktion	Verwandter Datensatz(e)
<p>Der Datensatz <Benutzer oder Elemente oder Interaktionen> hat X%-Zeilen mit einem fehlenden Wert. Dies kann sich negativ auf Empfehlungen auswirken. Wir empfehlen, dass alle erforderlichen und optionalen Felder zu mindestens 70 % vollständig sind.</p>	<p>Importieren Sie zusätzliche vollständige Datensätze oder importieren Sie Daten ohne unvollständige Zeilen erneut oder importieren Sie Daten erneut, wenn fehlende Werte durch Ersatzdaten ersetzt werden, z. B. den Durchschnitt für numerische Spalten oder den häufigsten Wert für kategoriale Spalten.</p>	<p>Any</p>
<p>Die folgenden Spalte(n) im Datensatz <datasetType > sind zu weniger als 70 % vollständig: <ColumnName, ColumnName...>. Wenn diese Daten im Training enthalten sind, können sie sich negativ auf Empfehlungen auswirken. Es wird empfohlen, dass Spalten, die Nullwerte zulassen, zu mindestens 70 % vollständig sind.</p>	<p>Importieren Sie zusätzliche vollständige Datensätze oder importieren Sie Daten ohne unvollständige Zeilen erneut oder importieren Sie Daten erneut, wenn fehlende Werte durch Ersatzdaten ersetzt werden, z. B. den Durchschnitt für numerische Spalten oder den häufigsten Wert für kategoriale Spalten.</p>	<p>Any</p>
<p>Die folgenden (numerischen) Spalte(n) haben Ausreißer: <ColumnName, ColumnName...>. Ausreißer sind nicht immer ein Problem, sondern wirken sich manchmal negativ auf Empfehlungen aus.</p>	<p>Überprüfen Sie anhand der folgenden Spaltenstatistik, ob die Mindest- und Höchstwerte für diese Spalten Ihren Erwartungen entsprechen. Wenn diese Werte unerwartet sind, überprüfen Sie die Daten in diesen Spalten auf Ungenauigkeiten und überprüfen Sie Ihre Datenerfassung und Datenverarbeitung auf Probleme.</p>	<p>Any</p>

Insight	Aktion	Verwandter Datensatz(e)
Die folgenden Spalten haben mehr als 1000 mögliche Kategorien: <ColumnName, ColumnName...>. Wenn diese Daten im Training enthalten sind, können sie sich negativ auf Empfehlungen auswirken: <ColumnName, ColumnName...>.	Überprüfen Sie Ihre kategorialen Daten auf Probleme, z. B. doppelte Kategorien, die durch Schreibvariationen verursacht werden. Beheben Sie alle Ungenauigkeiten und importieren Sie Daten erneut.	Any
Die folgenden Textmetadaten spalten sind zu weniger als 85 % vollständig und werden nicht für das Modelltraining verwendet: <ColumnName, ColumnName...>.	Importieren Sie zusätzliche Zeilen oder importieren Sie die Zeilen erneut mit Textdaten für diese Spalte(n).	Items
Der Interaktionsdatensatz verfügt über mehr als 10 eindeutige Ereignistypen, was dazu führt, dass das Modelltraining fehlschlägt.	Überprüfen Sie Ihre Spalte mit dem Ereignistyp auf Ungenauigkeiten wie doppelte Ereignistypen, die durch Schreibschwankungen verursacht werden. Entfernen Sie unnötige Ereignistypen und importieren Sie Daten erneut.	Elementinteraktionen
Der Interaktionsdatensatz hat für alle Datensätze denselben Zeitstempel. Wenn Sie ein USER_SEGMENTATION-Rezept verwenden und alle Datensätze denselben Zeitstempel haben, schlägt das Modelltraining fehl.	Überprüfen Sie Ihre Daten auf Zeitstempelprobleme und ersetzen Sie doppelte Zeitstempel durch eindeutige Zeitstempel.	Elementinteraktionen

Anzeigen von Erkenntnissen und Statistiken zu Datensätzen

Um Erkenntnisse und Statistiken zu Ihren Daten in Amazon-Personalize-Datensätzen anzuzeigen, navigieren Sie in der Amazon-Personalize-Konsole zu Ihren Datensätzen und wählen Sie Analyse ausführen aus.

So zeigen Sie Erkenntnisse und Statistiken an

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich unter Datensätze die Option Datenanalyse aus.
4. Wählen Sie oben rechts Analyse ausführen aus. Amazon Personalize beginnt mit der Analyse Ihrer Daten. Dies kann bis zu 15 Minuten dauern. Bei Erfolg werden die Ergebnisse auf dieser Seite angezeigt.
5. Verwenden Sie in Insights Folgendes, um die angezeigten Erkenntnisse zu filtern.
 - Um Erkenntnisse zu finden, die eine bestimmte Sprache enthalten, geben Sie Ihre Kriterien in Erkenntnis suchen ein. Wenn Sie Text eingeben, wird die Liste so aktualisiert, dass nur Erkenntnisse mit der genauen Zeichenfolge in die Erkenntnis oder empfohlene Aktion aufgenommen werden.
 - Um die Erkenntnisse nach Datensatztyp zu filtern, ändern Sie Alle Datensätze in den spezifischen Datensatztyp. Die Liste wird aktualisiert, sodass sie nur Erkenntnisse zu diesem Datensatz enthält.
6. Gehen Sie wie folgt vor, um Datensatzstatistiken für einen Datensatz anzuzeigen.
 - Um allgemeine Details und Statistiken zu einem Datensatz anzuzeigen, z. B. die Anzahl der Zeilen, eindeutige Benutzer und eindeutige Elemente in einem Interaktionsdatensatz, erweitern Sie den Abschnitt für den Datensatz.
 - Um detaillierte Statistiken für eine Spalte anzuzeigen, erweitern Sie den Datensatzabschnitt, wählen Sie Statistiken auf Spaltenebene und wählen Sie das Optionsfeld für die Spalte aus.
7. Korrigieren Sie alle Probleme in Ihren Daten, importieren Sie sie erneut und führen Sie eine weitere Analyse durch, um sie zu überprüfen. Weitere Informationen zum erneuten Importieren von Daten finden Sie unter [Aktualisieren von Daten](#).

Exportieren eines Datensatzes

Note

Sie können keine Daten in einem Datensatz mit Aktionsinteraktionen oder einem Aktionsdatensatz exportieren.

Nachdem Sie Ihre Daten in einen Amazon Personalize-Datensatz importiert haben, können Sie die Daten in einen Amazon S3 S3-Bucket exportieren. Sie können Daten exportieren, um die Daten zu verifizieren und zu überprüfen, die Amazon Personalize zur Generierung von Empfehlungen verwendet, um die zuvor aufgezeichneten Artikelinteraktionsereignisse in Echtzeit anzuzeigen oder um Offline-Analysen Ihrer Daten durchzuführen.

Sie können wählen, ob Sie nur die Daten exportieren möchten, die Sie in großen Mengen importiert haben (die mit einem Amazon Personalize Personalize-Datensatz-Importauftrag importiert wurden), nur die Daten, die Sie einzeln importiert haben (Datensätze, die mit der Konsole oder den `PutItems` Operationen `PutEvents`, `PutUsers`, oder importiert wurden), oder beides.

Für Datensätze, die für alle Felder exakt übereinstimmen, exportiert Amazon Personalize nur einen Datensatz. Wenn zwei Datensätze dieselbe ID haben, aber ein oder mehrere Felder unterschiedlich sind, schließt Amazon Personalize die Datensätze ein oder entfernt sie, je nachdem, welche Daten Sie exportieren möchten:

- Wenn Sie sowohl Massen- als auch inkrementelle Daten exportieren, exportiert Amazon Personalize nur die neuesten Artikel mit derselben ID (in Artikel-datensatzexporten) und nur Benutzer mit derselben ID (in Datensatzexporten für Benutzer). Für Datensätze zu Artikelinteraktionen exportiert Amazon Personalize alle Artikelinteraktionsdaten.
- Wenn Sie nur inkrementelle Daten exportieren, exportiert Amazon Personalize alle Artikel-, Benutzer- oder Artikelinteraktionsdaten, die Sie einzeln importiert haben, einschließlich Artikel oder Benutzer mit denselben IDs. Nur Datensätze, die für alle Felder exakt übereinstimmen, werden ausgeschlossen.
- Wenn Sie nur Massendaten exportieren, schließt Amazon Personalize alle Artikel-, Benutzer- oder Artikelinteraktionsdaten ein, die Sie in großen Mengen importiert haben, einschließlich Artikel oder Benutzer mit denselben IDs. Nur Datensätze, die für alle Felder exakt übereinstimmen, werden ausgeschlossen.

Um einen Datensatz zu exportieren, erstellen Sie einen Datensatz-Exportjob. Ein Datensatz-Exportjob ist ein Tool zum Exportieren von Datensätzen, das die Datensätze in einem Datensatz in eine oder mehrere CSV-Dateien in einem Amazon S3 S3-Bucket ausgibt. Die CSV-Ausgabedatei enthält eine Kopfzeile mit Spaltennamen, die den Feldern im Schema des Datensatzes entsprechen.

Sie können einen Datensatz-Exportauftrag mit der Amazon Personalize Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs erstellen.

Themen

- [Anforderungen an die Berechtigungen für den Export von Datensätzen](#)
- [Erstellen eines Datensatz-Exportauftrags \(Konsole\)](#)
- [Einen Datensatz-Exportauftrag erstellen \(AWS CLI\)](#)
- [Einen Datensatz-Exportauftrag \(AWSSDKs\) erstellen](#)

Anforderungen an die Berechtigungen für den Export von Datensätzen

Um einen Datensatz zu exportieren, benötigt Amazon Personalize die Erlaubnis, Dateien zu Ihrem Amazon S3 S3-Bucket hinzuzufügen. Um Berechtigungen zu erteilen, fügen Sie Ihrer Amazon Personalize Personalize-Servicerolle eine neue AWS Identity and Access Management (IAM) - Richtlinie hinzu, die der Rolle die Berechtigung erteilt, die `ListBucket` Aktionen `PutObject` und in Ihrem Bucket zu verwenden, und fügen Sie Ihrem Amazon S3 S3-Ausgabe-Bucket eine Bucket-Richtlinie hinzu, die dem Amazon Personalize Personalize-Prinzip die Berechtigung zur Verwendung der `PutObject` Aktionen und gewährt. `ListBucket`

Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Servicerollenrichtlinie für den Export eines Datensatzes

Die folgende Beispielrichtlinie erteilt Ihrer Amazon Personalize Personalize-Servicerolle die Erlaubnis, die `ListBucket` Aktionen `PutObject` und zu verwenden. `bucket-name` Ersetzen Sie es durch den Namen Ihres Ausgabe-Buckets. Informationen zum Anhängen von Richtlinien an eine IAM-Dienstrolle finden Sie unter [Ihrer Amazon Personalize-Servicerolle eine Amazon S3-Richtlinie zuordnen](#)

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
```

```

        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ]
}

```

Amazon S3 S3-Bucket-Richtlinie für den Export eines Datensatzes

Die folgende Beispielrichtlinie erteilt Amazon Personalize die Erlaubnis, die ListBucket Aktionen PutObject und in einem Amazon S3 S3-Bucket zu verwenden. Ersetzen Sie bucket-name durch den Namen von Ihrem Bucket. Informationen zum Hinzufügen einer Amazon S3 S3-Bucket-Richtlinie zu einem Bucket finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#) im Amazon Simple Storage Service-Benutzerhandbuch.

```

{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

Erstellen eines Datensatz-Exportauftrags (Konsole)

Nachdem Sie Ihre Daten in einen Datensatz importiert und einen Amazon S3 S3-Ausgabe-Bucket erstellt haben, können Sie die Daten zur Analyse in den Bucket exportieren. Um einen Datensatz

mit der Amazon Personalize-Konsole zu exportieren, erstellen Sie einen Datensatz-Exportauftrag. Informationen zum Erstellen eines Amazon S3 S3-Buckets finden Sie unter [Erstellen eines Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Bevor Sie einen Datensatz exportieren, stellen Sie sicher, dass Ihre Amazon Personalize-Servicerolle auf Ihren Amazon S3-Ausgabe-Bucket zugreifen und in diesen schreiben kann. Siehe [Anforderungen an die Berechtigungen für den Export von Datensätzen](#).

So erstellen Sie einen Auftrag zum Exportieren von Datensätzen (Konsole)

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home>.
2. Wählen Sie im Navigationsbereich Datensatzgruppen aus.
3. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
4. Wählen Sie im Navigationsbereich Datensätze aus.
5. Wählen Sie den Datensatz aus, den Sie in einen Amazon S3 S3-Bucket exportieren möchten.
6. Wählen Sie unter Datensatz-Exportaufträge die Option Datensatz-Exportjob erstellen aus.
7. Geben Sie im Feld Details zum Datensatz-Exportauftrag für den Namen des Datensatz-Exportjobs einen Namen für den Exportauftrag ein.
8. Wählen Sie für die IAM-Servicerolle die Amazon Personalize-Servicerolle aus, in der Sie sie erstellt haben. [Erstellen einer IAM-Rolle für Amazon Personalize](#)
9. Geben Sie für den Amazon S3 S3-Datenausgabepfad den Amazon S3-Ziel-Bucket ein. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder path>

10. Wenn Sie AWS KMS für die Verschlüsselung den KMS-Schlüssel ARN verwenden, geben Sie den Amazon-Ressourcennamen (ARN) für den AWS KMS Schlüssel ein.
11. Wählen Sie unter Datentyp exportieren den zu exportierenden Datentyp aus, je nachdem, wie Sie die Daten ursprünglich importiert haben.
 - Wählen Sie Bulk aus, um nur Daten zu exportieren, die Sie mithilfe eines Datensatz-Importjobs in großen Mengen importiert haben.
 - Wählen Sie Inkrementell, um nur Daten zu exportieren, die Sie einzeln mit der Konsole oder den PutItems Operationen PutEventsPutUsers, oder importiert haben.
 - Wählen Sie Beide, um alle Daten im Datensatz zu exportieren.

12. Fügen Sie für Tags optional beliebige Tags hinzu. Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#)
13. Wählen Sie „Datensatz-Exportauftrag erstellen“.

Auf der Datensatz-Übersichtsseite unter Datensatz-Exportjobs wird der Job mit dem Status Exportjob aufgeführt. Der Datensatz-Exportjob ist abgeschlossen, wenn der Status AKTIV ist. Anschließend können Sie die Daten aus dem Amazon S3 S3-Ausgabe-Bucket herunterladen. Informationen zum Herunterladen von Objekten aus einem Amazon S3 S3-Bucket finden Sie unter [Objekt herunterladen](#) im Amazon Simple Storage Service-Benutzerhandbuch. .

Einen Datensatz-Exportauftrag erstellen (AWS CLI)

Nachdem Sie Ihre Daten in den Datensatz importiert und einen Amazon S3 S3-Ausgabe-Bucket erstellt haben, können Sie den Datensatz zur Analyse in den Bucket exportieren. Um einen Datensatz mit dem zu exportieren AWS CLI, erstellen Sie mit dem `create-dataset-export-job` AWS CLI Befehl einen Datensatz-Exportauftrag. Informationen zum Erstellen eines Amazon S3 S3-Buckets finden Sie unter [Erstellen eines Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Bevor Sie einen Datensatz exportieren, stellen Sie sicher, dass die Amazon Personalize-Servicerolle auf Ihren Amazon S3-Ausgabe-Bucket zugreifen und in diesen schreiben kann. Siehe [Anforderungen an die Berechtigungen für den Export von Datensätzen](#).

Das Folgende ist ein Beispiel für den `create-dataset-export-job` AWS CLI Befehl. Geben Sie dem Job einen Namen, `dataset_arn` ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN) des Datensatzes, den Sie exportieren möchten, und `role_arn` ersetzen Sie ihn durch den ARN der Amazon Personalize-Servicerolle, in [Erstellen einer IAM-Rolle für Amazon Personalize](#) der Sie erstellt haben. Geben Sie `ins3DataDestination`, für `kmsKeyArn`, optional den ARN für Ihren AWS KMS Schlüssel und für den `path` Pfad zu Ihrem Amazon S3 S3-Ausgabe-Bucket an.

Geben Sie für die `DataIngestion-mode`, die exportiert werden sollen, aus den folgenden Optionen an:

- Geben Sie `BULK` an, dass nur Daten exportiert werden sollen, die Sie mithilfe eines Datensatz-Importjobs in großen Mengen importiert haben.
- Geben Sie `PUT` an, dass nur Daten exportiert werden sollen, die Sie einzeln mit der Konsole oder den `PutItems` Operationen `PutEvents` `PutUsers`, oder importiert haben.
- Geben Sie `ALL` an, dass alle Daten im Datensatz exportiert werden sollen.

Weitere Informationen finden Sie unter [CreateDatasetExportJob](#).

```
aws personalize create-dataset-export-job \  
  --job-name job name \  
  --dataset-arn dataset ARN \  
  --job-output '{"s3DataDestination\":{\"kmsKeyArn\":\"kms key ARN\",\"path\":  
  \"s3://bucket-name/folder-name/\"}}' \  
  --role-arn role ARN \  
  --ingestion-mode PUT
```

Der ARN des Datensatz-Exportjobs wird angezeigt.

```
{  
  "datasetExportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-export-job/  
  DatasetExportJobName"  
}
```

Verwenden Sie den DescribeDatasetExportJob Vorgang, um den Status zu überprüfen.

```
aws personalize describe-dataset-export-job \  
  --dataset-export-job-arn dataset export job ARN
```

Einen Datensatz-Exportauftrag (AWSSDKs) erstellen

Nachdem Sie Ihre Daten in den Datensatz importiert und einen Amazon S3 S3-Ausgabe-Bucket erstellt haben, können Sie den Datensatz zur Analyse in den Bucket exportieren. Um einen Datensatz mithilfe der AWS SDKs zu exportieren, erstellen Sie mithilfe der [CreateDatasetExportJob](#) Operation einen Datensatz-Exportauftrag. Informationen zum Erstellen eines Amazon S3 S3-Buckets finden Sie unter [Erstellen eines Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Der folgende Code zeigt, wie Sie einen Datensatz-Exportauftrag mit dem SDK for Python (Boto3) oder dem SDK for Java 2.x SDK erstellen.

Bevor Sie einen Datensatz exportieren, stellen Sie sicher, dass die Amazon Personalize-Servicerolle auf Ihren Amazon S3-Ausgabe-Bucket zugreifen und in diesen schreiben kann. Siehe [Anforderungen an die Berechtigungen für den Export von Datensätzen](#).

SDK for Python (Boto3)

Gehen Sie wie folgt vorcreate_dataset_export_job, um die Daten in einem Datensatz in einen Amazon S3 S3-Bucket zu exportieren. Geben Sie dem Job einen Namen, dataset

`arn` ersetzen Sie ihn durch den Amazon-Ressourcennamen (ARN) des Datensatzes, den Sie exportieren möchten, und `role` ARN ersetzen Sie ihn durch den ARN der Amazon Personalize-Servicerolle, in [Erstellen einer IAM-Rolle für Amazon Personalize](#) der Sie erstellt haben. Geben Sie `ins3DataDestination`, `fürkmsKeyArn`, optional den ARN für Ihren AWS KMS Schlüssel und für den `path` Pfad zu Ihrem Amazon S3 S3-Ausgabe-Bucket an.

Geben Sie für die `DateningestionMode`, die exportiert werden sollen, aus den folgenden Optionen an:

- Geben Sie `BULK` an, dass nur Daten exportiert werden sollen, die Sie mithilfe eines Datensatz-Importjobs in großen Mengen importiert haben.
- Geben Sie `PUT` an, dass nur Daten exportiert werden sollen, die Sie einzeln mit der Konsole oder den `PutItems` Operationen `PutEvents` `PutUsers`, oder importiert haben.
- Geben Sie `ALL` an, dass alle Daten im Datensatz exportiert werden sollen.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_export_job(
    jobName = 'job name',
    datasetArn = 'dataset ARN',
    jobOutput = {
        "s3DataDestination": {
            "kmsKeyArn": "kms key ARN",
            "path": "s3://bucket-name/folder-name/"
        }
    },
    roleArn = 'role ARN',
    ingestionMode = 'PUT'
)

dsej_arn = response['datasetExportJobArn']

print ('Dataset Export Job arn: ' + dsej_arn)

description = personalize.describe_dataset_export_job(
    datasetExportJobArn = dsej_arn)['datasetExportJob']

print('Name: ' + description['jobName'])
```

```
print('ARN: ' + description['datasetExportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

Verwenden Sie die folgende `createDatasetExportJob` Methode, um einen Datensatz-Exportauftrag zu erstellen. Übergeben Sie Folgendes als Parameter: a `PersonalizeClient`, den Namen für Ihren Exportauftrag, den ARN des Datensatzes, den Sie exportieren möchten, den Aufnahmemodus, den Pfad für den Amazon S3 S3-Ausgabe-Bucket und den ARN für Ihren AWS KMS Schlüssel.

Das `ingestionMode` kann eine der folgenden Optionen sein:

- Wird verwendet `IngestionMode.BULK`, um nur Daten zu exportieren, die Sie mithilfe eines Datensatz-Importjobs in großen Mengen importiert haben.
- Wird verwendet `IngestionMode.PUT`, um nur Daten zu exportieren, die Sie einzeln mit der Konsole oder den `PutItems` Operationen `PutEvents` `PutUsers`, oder importiert haben.
- `IngestionMode.ALL` dient zum Exportieren aller Daten im Datensatz.

```
public static void createDatasetExportJob(PersonalizeClient personalizeClient,
                                          String jobName,
                                          String datasetArn,
                                          IngestionMode ingestionMode,
                                          String roleArn,
                                          String s3BucketPath,
                                          String kmsKeyArn) {

    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String status = null;

    try {
        S3DataConfig exportS3DataConfig = S3DataConfig.builder()
            .path(s3BucketPath)
            .kmsKeyArn(kmsKeyArn)
            .build();

        DatasetExportJobOutput jobOutput = DatasetExportJobOutput.builder()
            .s3DataDestination(exportS3DataConfig)
            .build();
```



```
        CreateDatasetExportJobRequest createRequest =
CreateDatasetExportJobRequest.builder()
    .jobName(jobName)
    .datasetArn(datasetArn)
    .ingestionMode(ingestionMode)
    .jobOutput(jobOutput)
    .roleArn(roleArn)
    .build();

        String datasetExportJobArn =
personalizeClient.createDatasetExportJob(createRequest).datasetExportJobArn();

        DescribeDatasetExportJobRequest describeDatasetExportJobRequest =
DescribeDatasetExportJobRequest.builder()
    .datasetExportJobArn(datasetExportJobArn)
    .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetExportJob datasetExportJob =
personalizeClient.describeDatasetExportJob(describeDatasetExportJobRequest)
                .datasetExportJob();

            status = datasetExportJob.status();
            System.out.println("Export job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Löschen von Daten

Um Daten in Amazon Personalize zu löschen, löschen Sie den Datensatz. Amazon Personalize speichert Ihre Daten in Datensätzen, bis Sie die Datensätze löschen. Sie können einen Datensatz nicht löschen, wenn sich die Version eines Auftrags oder einer Lösung für den Import von Datensätzen im CREATE PENDING Status oder befindet. IN PROGRESS Wenn Sie die Rezepte „Benutzerpersonalisierung“ oder „Nächstbeste Aktion“ oder „Top-Tipps“ für Sie und „Für Ihre Anwendungsfälle empfohlen“ verwenden, werden beim Löschen eines Datensatzes automatische Updates für alle zugehörigen Lösungsversionen oder Empfehlungen gestoppt.

Sie können einen Datensatz mit der Amazon Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs löschen.

Themen

- [Löschen eines Datensatzes \(Konsole\)](#)
- [Löschen eines Datensatzes \(AWS CLI\)](#)
- [Löschen eines Datensatzes \(AWSSDKs\)](#)

Löschen eines Datensatzes (Konsole)

Um einen Datensatz mit der Amazon Personalize-Konsole zu löschen, navigieren Sie zur Datensatz-Detailseite und wählen Sie Löschen.

So löschen Sie einen Datensatz

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home>.
2. Wählen Sie im Navigationsbereich Datensatzgruppen aus.
3. Wählen Sie auf der Seite Datensatzgruppen Ihre Datensatzgruppe aus.
4. Wählen Sie im Navigationsbereich Datensätze aus.
5. Wählen Sie den Datensatz aus, den Sie löschen möchten.
6. Wählen Sie Löschen und bestätigen Sie das Löschen des Datensatzes.

Löschen eines Datensatzes (AWS CLI)

Der folgende Code zeigt, wie ein Datensatz mit der Operation AWS CLI und der [DeleteDataset](#) Operation gelöscht wird.

```
aws personalize delete-dataset --dataset-arn dataset-arn
```

Löschen eines Datensatzes (AWSSDKs)

Der folgende Code zeigt, wie Sie einen Datensatz mit den AWS SDKs und den [DeleteDataset](#) Vorgang löschen.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.delete_dataset(
    datasetArn = 'dataset ARN'
)
```

SDK for Java 2.x

```
public static void deleteDataset(PersonalizeClient personalizeClient,
                                String datasetArn) {

    try {
        DeleteDatasetRequest deleteRequest = DeleteDatasetRequest.builder()
            .datasetArn(datasetArn)
            .build();

        int responseCode =
personalizeClient.deleteDataset(deleteRequest).sdkHttpResponse().statusCode();
        System.out.println(responseCode);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

Filtern von Empfehlungen und Benutzersegmenten

Wenn Sie Empfehlungen mit einem Domain-Empfehlungsgeber oder einer benutzerdefinierten Kampagne erhalten, können Sie Ergebnisse basierend auf benutzerdefinierten Kriterien filtern. Beispielsweise möchten Sie möglicherweise keine Produkte empfehlen, die ein Benutzer bereits gekauft hat, oder nur Artikel für eine bestimmte Altersgruppe empfehlen.

Ebenso möchten Sie bei USER_SEGMENTATION-Rezepten möglicherweise bestimmte Benutzertypen nicht in Benutzersegmente aufnehmen. Durch das Filtern Ihrer Ergebnisse können Sie steuern, welche Elemente Benutzern oder Benutzern empfohlen werden, die in Benutzersegmente aufgenommen werden.

Sie können Filter mithilfe der Amazon-Personalize-Konsole, der AWS Command Line Interface (AWS CLI) und der AWS SDKs erstellen, bearbeiten, löschen und anwenden.

- Für Echtzeitempfehlungen wenden Sie einen Filter an und geben alle Filterparameterwerte an, wenn Sie die GetPersonalizedRanking Operationen GetRecommendations, GetActionRecommendations, oder aufrufen. Sie können auch einen Filter anwenden, wenn Sie Empfehlungen von einer Kampagne oder einem Empfehler in der Konsole erhalten.

Wenn Sie Empfehlungen zu Artikeln in Echtzeit mit personalisierten oder verwandten Artikelrezepten oder Anwendungsfällen erhalten, können Sie in Ihrer Anfrage eine Hochstufung angeben. Bei einer Hochstufung wird ein Filter verwendet, um zusätzliche Geschäftsregeln zu definieren, die für eine konfigurierbare Teilmenge der empfohlenen Elemente gelten. Weitere Informationen finden Sie unter [Werbung für Artikel in Empfehlungen](#).

- Bei Batch-Workflows fügen Sie alle Filterparameterwerte in Ihr Eingabe-JSON ein. Anschließend geben Sie den Amazon-Ressourcennamen (ARN) des Filters an, wenn Sie einen Batch-Inferenzantrag oder Batch-Segmentauftrag erstellen. Weitere Informationen finden Sie unter [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#).

Filtern von Aktualisierungen für neue Datensätze

Für Daten, die Sie mit den PutActionInteractions Operationen PutEvents oder importieren, aktualisiert Amazon Personalize alle Filter in der Datensatzgruppe innerhalb von Sekunden nach dem Import mit den neuen Daten. Wenn Ihr Filter beispielsweise gekaufte Artikel aus Empfehlungen entfernt und Sie ein Kaufereignis für einen Benutzer mit der - PutEvents Operation aufzeichnen, wird dieses Element

innerhalb von Sekunden nach der Aufzeichnung des Ereignisses aus zukünftigen Empfehlungen für diesen Benutzer entfernt.

Für alle anderen Daten, die in großen Mengen oder einzeln importiert werden, aktualisiert Amazon Personalize alle Filter in der Datensatzgruppe innerhalb von 20 Minuten nach dem letzten Import mit den neuen Daten.

Themen

- [Filterausdrücke](#)
- [Filtern von Empfehlungen in Echtzeit](#)
- [Filtern von Batch-Empfehlungen und Benutzersegmenten \(benutzerdefinierte Ressourcen\)](#)

Filterausdrücke

Um Filter zu konfigurieren, müssen Sie einen ordnungsgemäß formatierten Filterausdruck verwenden. Filterausdrücke bestehen aus Datensatz- und Feldkennungen im Format `dataset.field` sowie aus logischen Operatoren, Schlüsselwörtern und Werten. Für Werte können Sie feste Werte angeben oder Platzhalterparameter hinzufügen und die Filterkriterien festlegen, wenn Sie Empfehlungen erhalten.

Sie können Filterausdrücke verwenden, um Elemente, Benutzer oder Aktionen aus Empfehlungen basierend auf Daten aus den folgenden Datensätzen zu filtern:

- **Elementinteraktionen:** Sie können Filterausdrücke verwenden, um Elemente oder Benutzer basierend auf Interaktionsdaten einzuschließen oder auszuschließen. Sie können beispielsweise Elemente ausschließen, auf die ein Benutzer geklickt hat (für Elementempfehlungen), oder nur Benutzer einbeziehen, die Elemente bewertet haben (für das Element-Affinität-Rezept). Für alle Rezepttypen können Sie nur basierend auf dem Ereignistyp filtern. Sie können nicht auf der Grundlage anderer Interaktionsmetadaten filtern, z. B. kontextbezogene Metadaten. Sie können keine Filter für Elementinteraktionen mit der verwenden [Item-Attribute-Affinity-Rezept](#).

Amazon Personalize berücksichtigt bis zu 100 der letzten Interaktionen pro Benutzer und Ereignistyp. Dies ist ein anpassbares Kontingent. Sie können eine Kontingenterhöhung über die [Service Quotas-Konsole](#) beantragen.

- **Aktionsinteraktionen:** Verwenden Sie Filterausdrücke, um Aktionen, mit denen ein Benutzer interagiert hat, je nach Ereignistyp einzu- oder auszuschließen. Sie können beispielsweise

Aktionen ausschließen, die ein Benutzer bereits ausgeführt hat. Sie können nicht auf der Grundlage anderer Aktionsinteraktionsmetadaten filtern.

Amazon Personalize berücksichtigt bis zu 300 der letzten Aktionsinteraktionen pro Benutzer und Ereignistyp. Dies ist ein anpassbares Kontingent. Sie können eine Kontingenterhöhung über die [Service Quotas-Konsole](#) beantragen.

- **Elemente** : Verwenden Sie Filterausdrücke, um Elemente basierend auf bestimmten Elementbedingungen ein- oder auszuschließen. Sie können keine Filter verwenden, um Elemente basierend auf Metadaten unstrukturierter Textelemente wie Produktbeschreibungen einzuschließen oder auszuschließen. Wenn Ihr Domain-Anwendungsfall oder benutzerdefiniertes Rezept Empfehlungen für verwandte Elemente generiert, z. B. das Rezept Ähnliche Elemente oder den Anwendungsfall Mehr Like X Domain, können Sie Filterausdrücke verwenden, um Elemente basierend auf den Eigenschaften des Elements, das Sie in Ihrer Empfehlungsanfrage angeben, einzu- oder auszuschließen.
- **Benutzer**: Wenn Sie für Element- und Aktionsempfehlungen einen Benutzerdatensatz haben, können Sie Elemente oder Aktionen basierend auf einem ausschließen oder einschließen `CurrentUser`. Für personalisierte Empfehlungen, beliebte Elemente und Aktionsempfehlungen ist dies der Benutzer, für den Sie Empfehlungen erhalten. Für verwandte Elemente ist dies ein optionaler Benutzer, den Sie in Ihrer Empfehlungsanfrage angeben können.

Für Benutzersegmente können Sie Filterausdrücke verwenden, um Benutzersegmente basierend auf Attributen wie einzuschließen oder von ihnen auszuschließen `Users.MEMBERSHIP_STATUS`.

- **Aktionen**: Verwenden Sie Filterausdrücke, um Aktionen basierend auf bestimmten Aktionsbedingungen einzu- oder auszuschließen. Amazon Personalize schließt automatisch Aktionen aus, die auf `-Action expiration timestamp` und `-Repeat frequency` Daten basieren. Sie können keine zusätzlichen benutzerdefinierten Filter erstellen, die basierend auf diesen Daten filtern.

Eine vollständige Liste der Filterausdruckselemente finden Sie unter [Filterausdruckselemente](#).

Beispiele für Filterausdrücke finden Sie unter [Beispiele für Filterausdrücke](#).

Themen

- [Richtlinien und Anforderungen](#)
- [Filterausdrucksstruktur und -elemente](#)
- [Beispiele für Filterausdrücke](#)

Richtlinien und Anforderungen

Beachten Sie beim Erstellen eines Filterausdrucks die folgenden Richtlinien und Anforderungen:

- Sie können keine Filter verwenden, um Elemente basierend auf Metadaten unstrukturierter Textelemente wie Produktbeschreibungen einzuschließen oder auszuschließen.
- Wenn Sie auf der Grundlage von Daten zu Element- oder Aktionsinteraktionen filtern, können Sie nur nach Ereignistyp filtern. Sie können nicht auf der Grundlage anderer Interaktionsmetadaten filtern, z. B. kontextbezogene Metadaten.
- Amazon Personalize ignoriert Groß- und Kleinschreibung nur beim Abgleich von Ereignistypen.
- Sie können keine Elementinteraktions- und Elementdatensätze in einem Ausdruck verwenden. Um einen Filter zu erstellen, der nach Interaktion und dann Elementdatensätzen filtert (oder umgekehrt), müssen Sie zwei oder mehr Ausdrücke miteinander verketteten. Weitere Informationen finden Sie unter [Kombinieren mehrerer Ausdrücke](#).
- Sie können keine Elementinteraktions- und Aktionsdatensätze in einem Ausdruck verwenden. Um einen Filter zu erstellen, der nach Elementinteraktion und dann Aktionsdatensätzen (oder umgekehrt) filtert, müssen Sie zwei oder mehr Ausdrücke miteinander verketteten. Weitere Informationen finden Sie unter [Kombinieren mehrerer Ausdrücke](#).
- Sie können keine Filter für Elementinteraktionen mit dem verwenden [Item-Attribute-Affinity-Rezept](#).
- Sie können keine Filterausdrücke erstellen, die mithilfe von Werten mit einem booleschen Typ in Ihrem Schema filtern. Um basierend auf booleschen Werten zu filtern, verwenden Sie ein Schema mit einem Feld vom Typ Zeichenfolge und verwenden Sie die Werte "True" und "False" in Ihren Daten. Oder Sie können den Typ int oder long und die Werte 0 und verwenden1.
- Die maximale Anzahl unterschiedlicher Datensatzfelder für einen Filter, entweder in einem Ausdruck oder über mehrere miteinander verkettete Ausdrücke, beträgt 5. Die maximale Anzahl unterschiedlicher Datensatzfelder für alle Filter in einer Datensatzgruppe beträgt 10.
- Sie können einen Filter mit dem - CurrentItem Element nur anwenden, wenn Ihr Domain-Anwendungsfall oder Ihr benutzerdefiniertes Rezept Empfehlungen für verwandte Elemente generiert, z. B. das Rezept für ähnliche Elemente oder den Anwendungsfall für eine Domain mit mehr Like X.
- Sie können Platzhalterparameter nicht in einem Filterausdruck verwenden, der den Operator NOT_IN verwendet. Verwenden Sie stattdessen den IN-Operator und die entgegengesetzte Aktion. Verwenden Sie beispielsweise Include anstelle von Exclude (oder umgekehrt).

- Sie können keine Filter erstellen, die basierend auf `- Action expiration timestamp` und `-Repeat frequency`Daten filtern. Amazon Personalize filtert automatisch Aktionsempfehlungen basierend auf diesen Daten.

Filterausdrucksstruktur und -elemente

Dieser Abschnitt enthält Informationen über die Struktur von Filterausdrücken und deren Elementen.

Themen

- [Filterausdrucksstruktur](#)
- [Filterausdruckselemente](#)

Filterausdrucksstruktur

Die allgemeine Struktur eines Filterausdrucks lautet wie folgt:

```
EXCLUDE/INCLUDE ItemID/ActionID/UserID WHERE dataset type.field IN/NOT IN (value/parameter)
```

Sie können entweder Filterausdrücke manuell erstellen oder mithilfe des [Ausdrucksgenerators](#) in der Konsole Hilfe zur Syntax und Struktur von Ausdrücken erhalten.

Filterausdruckselemente

Verwenden Sie die folgenden Elemente, um Filterausdrücke zu erstellen:

INCLUDE oder EXCLUDE

Verwenden Sie `, INCLUDE` um Empfehlungen auf Elemente zu beschränken, die die Filterkriterien erfüllen, ODER verwenden Sie `, EXCLUDE` um alle Elemente zu entfernen, die die Filterkriterien erfüllen.

ItemID/ActionID/UserID

Verwenden Sie eines dieser Elemente nach dem `- INCLUDE` oder `-EXCLUDE`Element. Welches Element Sie verwenden, hängt davon ab, ob Sie Elemente (für Elementempfehlungen), Aktionen (für Aktionsempfehlungen) oder Benutzer (für Benutzersegmente) filtern.

WHERE

Verwenden Sie `WHERE` um die Bedingungen für Elemente, Aktionen oder Benutzer zu überprüfen. Sie müssen das `-WHERE`Element nach `ItemID`, `ActionID` oder `userId` verwenden.

UND/ODER

Um mehrere Bedingungen innerhalb desselben Filterausdrucks miteinander zu verketten, verwenden Sie `AND` oder `OR`. Bedingungen, die mit `AND` oder miteinander verkettet sind, `OR` können sich nur auf Felder des Datensatzes auswirken, die in der ersten Bedingung verwendet werden.

Datensatz.Feld

Geben Sie den Datensatz und das Metadatenfeld, nach dem Sie Empfehlungen filtern möchten, im `datasetFormat` an `field`. Um beispielsweise Elementempfehlungen basierend auf dem Genres-Feld in Ihrem Elementdatensatz zu filtern, würden Sie `Items.genres` in Ihrem Filterausdruck verwenden.

IF-Bedingung

Verwenden Sie eine `-IF`Bedingung nur, um die Bedingungen für `currentUser` nur einmal am Ende eines Ausdrucks zu überprüfen. Sie können jedoch eine `IF` Bedingung mit `AND` erweitern.

currentUser.attribute

Um Elementempfehlungen basierend auf dem Benutzer zu filtern, für den Sie Empfehlungen erhalten, verwenden Sie in einer `IF`-Bedingung `currentUser` und geben Sie das Benutzerfeld an. Beispiel: `currentUser.AGE`

CurrentItem.attribute

Verwenden Sie nur verwandte Elementrezepte und Anwendungsfälle, `CurrentItem`attribute um Elemente basierend auf einem Attribut des Elements zu filtern, das Sie in Ihrer Anforderung für Empfehlungen zu verwandten Elementen angeben. Zum Beispiel `CurrentItem.GENRE` oder `CurrentItem.PRICE`.

Sie können einen Filter mit dem `-CurrentItem` Element nur anwenden, wenn Ihr Domain-Anwendungsfall oder Ihr benutzerdefiniertes Rezept Empfehlungen für verwandte Elemente generiert, z. B. das Rezept für ähnliche Elemente oder den Anwendungsfall für eine Domain mit mehr Like X. Wenn Sie zum ersten Mal einen Filter mit einem `-CurrentItem`Element erstellen,

kann die Filtererstellung einige Minuten dauern. Wenn Sie AWS KMS für die Verschlüsselung verwenden, kann die Filtererstellung bis zu 15 Minuten dauern.

IN/NOT IN

Verwenden Sie `IN` oder `NOT IN` als Vergleichsoperatoren, um nach einem oder mehreren Zeichenfolgenwerten zu filtern (oder nicht). Amazon Personalize filtert nur nach exakten Zeichenfolgen.

Vergleichsoperatoren

Verwenden Sie die Operatoren `=`, `<`, `<=`, `>`, `>=`, um numerische Daten, einschließlich Daten, die in einem Platzhalterparameter übergeben werden, auf Gleichheit zu testen.

Sternchen (*)

Verwenden Sie `*`, um Interaktionen aller Typen einzuschließen oder auszuschließen. Verwenden Sie `*` nur für Filterausdrücke, die das `-EVENT_TYPE`Feld eines `-Interactions`Datensatzes verwenden.

Pipe-Trennzeichen

Verwenden Sie das Pipe-Trennzeichen (`|`), um mehrere Ausdrücke miteinander zu verketteten. Weitere Informationen finden Sie unter [Kombinieren mehrerer Ausdrücke](#).

Parameter

Verwenden Sie für Ausdrücke, die Vergleichsoperatoren oder den `-IN`Operator verwenden, das Dollarzeichen (`$`) und einen Parameternamen, um einen Platzhalterparameter als Wert hinzuzufügen. Beispiel: `$GENRES` Wenn Sie in diesem Beispiel Empfehlungen erhalten, geben Sie die Genre oder Genre an, nach denen gefiltert werden soll.

Note

Sie definieren einen Parameternamen, wenn Sie ihn einem Ausdruck hinzufügen. Der Parameternamen muss nicht mit dem Feldnamen übereinstimmen. Wir empfehlen Ihnen, einen Parameternamen zu verwenden, der dem Feldnamen ähnelt und leicht zu merken ist. Sie verwenden den Parameternamen (wobei die Groß- und Kleinschreibung beachtet wird), wenn Sie den Filter auf Empfehlungsanforderungen anwenden. Ein Beispiel, das zeigt, wie ein Filter mit Platzhalterparametern angewendet wird, wenn die AWS SDKs verwendet werden, finden Sie unter [Anwenden eines Filters \(AWS SDKs\)](#).

Beispiele für Filterausdrücke

Verwenden Sie die Filterausdrücke in den folgenden Abschnitten, um zu erfahren, wie Sie Ihre eigenen Filterausdrücke erstellen.

Themen

- [Beispiele für Filterausdrücke für Elementempfehlungen](#)
- [Filterausdrücke für Benutzersegmente](#)
- [Beispiele für Filterausdrücke für Aktionsempfehlungen](#)
- [Kombinieren mehrerer Ausdrücke](#)

Beispiele für Filterausdrücke für Elementempfehlungen

Die folgenden Filterausdrücke zeigen, wie Elementempfehlungen basierend auf Elementinteraktionen, Elementmetadaten und Benutzermetadaten gefiltert werden. Sie sind nach Datentyp organisiert.

Themen

- [Daten zur Elementinteraktion](#)
- [Elementdaten](#)
- [Benutzerdaten](#)

Daten zur Elementinteraktion

Der folgende Ausdruck schließt Elemente basierend auf einem Ereignistyp (z. B. Klick) oder Ereignistypen aus, die Sie angeben, wenn Sie Empfehlungen mit dem `$EVENT_TYPE` Parameter erhalten.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

Der folgende Ausdruck schließt Elemente aus, auf die ein Benutzer geklickt oder gestreamt hat.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click", "stream")
```

Der folgende Ausdruck enthält nur Elemente, auf die der Benutzer geklickt hat.

```
INCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click")
```

Elementdaten

Der folgende Ausdruck schließt Elemente basierend auf einer Kategorie oder Kategorien aus, die Sie angeben, wenn Sie Empfehlungen mit dem \$CATEGORY Parameter erhalten.

```
EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)
```

Der folgende Ausdruck enthält nur Elemente, die günstiger sind als das aktuelle Element (das Element, das Sie in der Anforderung für Empfehlungen für verwandte Elemente angeben), und die von demselben Studio wie das aktuelle Element erstellt wurden. Sie können einen Filter mit dem - CurrentItem Element nur anwenden, wenn Ihr Domain-Anwendungsfall oder Ihr benutzerdefiniertes Rezept Empfehlungen für verwandte Elemente generiert.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.GENRE IN  
CurrentItem.GENRE
```

Der folgende Ausdruck schließt Elemente aus, die auf mehreren Ebenen kategorischer Felder basieren. Es schließt Elemente mit einem CATEGORY_L1-Wert von ausshoe, die keinen CATEGORY_L2-Wert von habenboot.

```
EXCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("shoe") AND Items.CATEGORY_L2 NOT IN  
("boot")
```

Der folgende Ausdruck enthält nur Elemente mit einem Preis kleiner oder gleich dem Preis, den Sie angeben, wenn Sie Empfehlungen mit dem \$PRICE Parameter erhalten.

```
INCLUDE ItemID WHERE Items.PRICE <= $PRICE
```

Der folgende Ausdruck enthält nur Elemente, die vor einem Zeitstempel (in Unix-Epochenzeit) erstellt wurden, den Sie beim Abrufen von Empfehlungen angeben.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

Der folgende Ausdruck enthält nur Elemente mit einer oder mehreren Genen, die Sie angeben, wenn Sie Empfehlungen mit dem \$GENRE Parameter erhalten.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE)
```

Der folgende Ausdruck enthält nur Elemente, die teurer sind als das aktuelle Element und die neuer als ein von Ihnen angegebener Zeitstempel (in Unix-Epochezeit) erstellt wurden. Sie können diesen Filter verwenden, wenn Sie verwandte Artikelempfehlungen erhalten und bestimmte Geschäftsregeln basierend auf dem Preis und einem unterschiedlichen Erstellungsdatum anwenden möchten.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.CREATION_TIMESTAMP > $DATE
```

Benutzerdaten

Der folgende Ausdruck schließt Elemente mit einer oder mehreren Genren aus, die Sie angeben, wenn Sie Empfehlungen mit dem \$GENRE Parameter erhalten, aber nur, wenn das Alter des aktuellen Benutzers dem Wert entspricht, den Sie angeben, wenn Sie Empfehlungen mit dem \$AGE Parameter erhalten.

```
EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE) IF CurrentUser.AGE = $AGE
```

Der folgende Ausdruck enthält nur Elemente mit watch für CATEGORY_L1 und luxury für CATEGORY_L2, wenn das Alter des aktuellen Benutzers überschreitet 18.

```
INCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("watch") AND Items.CATEGORY_L2 IN ("luxury")  
IF CurrentUser.AGE > 18
```

Filterausdrücke für Benutzersegmente

Die folgenden Filterausdrücke zeigen, wie Benutzersegmente basierend auf Elementinteraktionsdaten und Benutzermetadaten gefiltert werden. Sie sind nach Datentyp organisiert.

Benutzerdaten

Der folgende Filterausdruck enthält nur Benutzer mit einem Mitgliedschaftsstatus, der dem Wert entspricht, den Sie beim Abrufen von Benutzersegmenten angeben.

```
INCLUDE UserID WHERE Users.MEMBERSHIP_STATUS IN ($MEMBERSHIP)
```

Der folgende Filterausdruck schließt Benutzer mit einem Wert aus, der AGE kleiner als ein Wert ist, den Sie beim Abrufen von Benutzersegmenten angeben.

```
EXCLUDE UserID WHERE Users.AGE < $AGE
```

Daten zur Elementinteraktion

Der folgende Filterausdruck enthält nur Benutzer, die auf Elemente geklickt oder diese aufgeführt haben.

```
INCLUDE UserID WHERE Interactions.EVENT_TYPE IN ("click", "rating")
```

Der folgende Filterausdruck schließt Benutzer aus Benutzersegmenten aus, die Elementinteraktionen mit einem Ereignistyp haben, den Sie beim Abrufen von Benutzersegmenten angeben.

```
EXCLUDE UserID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

Beispiele für Filterausdrücke für Aktionsempfehlungen

Die folgenden Beispiele für Filterausdrücke zeigen, wie Aktionen basierend auf Aktionsinteraktionsdaten, Aktionsdaten und Benutzerdaten gefiltert werden. Sie sind nach Datentyp organisiert.

Themen

- [Aktionsinteraktionsdaten](#)
- [Aktionsdaten](#)
- [Benutzerdaten](#)

Aktionsinteraktionsdaten

Der folgende Filterausdruck enthält nur Aktionen in Empfehlungen, mit denen der Benutzer interagiert hat, wenn diese Interaktionen einen Ereignistyp haben, den Sie angeben, wenn Sie Empfehlungen erhalten.

```
INCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

Der folgende Filterausdruck schließt Aktionen aus, die der Benutzer basierend auf dem Ereignistyp nicht ausgeführt hat.

```
EXCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ("NOT_TAKEN")
```

Aktionsdaten

Der folgende Ausdruck schließt Aktionen basierend auf einer Kategorie oder Kategorien aus, die Sie angeben, wenn Sie Empfehlungen mit dem \$CATEGORY Parameter erhalten.

```
EXCLUDE ActionID WHERE Actions.CATEGORY IN ($CATEGORY)
```

Der folgende Ausdruck enthält nur Aktionen mit einem Wert, der größer als ein Wert ist, den Sie angeben, wenn Sie Empfehlungen erhalten.

```
INCLUDE ActionID WHERE Actions.VALUE > ($VALUE)
```

Benutzerdaten

Der folgende Ausdruck enthält nur Aktionen für Premium-Mitglieder, wenn der aktuelle Benutzer über eine Premium-Mitgliedschaft verfügt.

```
INCLUDE ActionID WHERE Action.MEMBERSHIP_LEVEL IN ("Premium") IF CurrentUser.MEMBERSHIP = $PREMIUM
```

Der folgende Ausdruck schließt Aktionen mit einem Wert aus, der VALUE kleiner als ein Wert ist, den Sie angeben, wenn Sie Empfehlungen erhalten, wenn der aktuelle Benutzer ein Premium-Mitglied ist.

```
EXCLUDE ActionID WHERE Actions.VALUE < ($VALUE) IF CurrentUser.MEMBERSHIP = $PREMIUM
```

Kombinieren mehrerer Ausdrücke

Um mehrere Ausdrücke miteinander zu kombinieren, verwenden Sie ein Pipe-Trennzeichen (|). Verwenden Sie eine Kombination von Ausdrücken, wenn Sie einen einzelnen Filter verwenden und nach Datensätzen für Element- und Elementinteraktionen oder Datensätzen für Aktions- und Aktionsinteraktionen filtern möchten. Jeder Ausdruck wird zuerst unabhängig ausgewertet und das Ergebnis ist entweder die Vereinigung oder die Schnittmenge der beiden Ergebnisse. Die folgenden Beispiele zeigen, wie Ausdrücke für Datensätze für Elemente und Elementinteraktionen erstellt werden, aber dieselben Regeln gelten für die Arbeit mit Aktionen und Aktionsinteraktionen.

Beispiel für übereinstimmende Ausdrücke

Wenn beide Ausdrücke EXCLUDE oder beide Ausdrücke verwenden INCLUDE, ist das Ergebnis die Vereinigung der beiden Ergebnisse wie folgt (A und B sind unterschiedliche Ausdrücke):

- Exclude A | Exclude B ist gleich Exclude result from A or result from B
- Include A | Include B ist gleich Include result from A or result from B

Das folgende Beispiel zeigt, wie zwei Ausdrücke kombiniert werden, die verwenden INCLUDE. Der erste Ausdruck enthält nur Elemente mit einer Kategorie oder Kategorien, die Sie angeben, wenn Sie Empfehlungen mit dem \$CATEGORY Parameter erhalten. Der zweite Ausdruck enthält Elemente, die der Benutzer als markiert hat favorite. Empfehlungen enthalten nur Elemente mit der von Ihnen angegebenen Kategorie sowie Elemente, die der Benutzer als Favorit markiert hat.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY) | INCLUDE ItemID WHERE  
Interactions.EVENT_TYPE IN ("favorite")
```

Beispiel für INCLUDE und EXCLUDE

Wenn ein oder mehrere Ausdrücke INCLUDE und ein oder mehrere Ausdrücke verwenden EXCLUDE, ist das Ergebnis die Subtraktion des EXCLUDE Ausdrucksergebnisses vom INCLUDE Ausdrucksergebnis wie folgt (A, B, C und D sind unterschiedliche Ausdrücke).

- Include A | Exclude B ist gleich Include result from A - result from B
- Include A | Include B | Exclude C | Exclude D ist gleich Include (A or B) - (C or D)

Ausdrucksreihenfolge spielt keine Rolle: Wenn der EXCLUDE-Ausdruck vor dem INCLUDE-Ausdruck steht, ist das Ergebnis dasselbe.

Das folgende Beispiel zeigt, wie ein INCLUDE Ausdruck und ein EXCLUDE Ausdruck kombiniert werden. Der erste Ausdruck enthält nur Elemente mit einem Genre oder Genres, die Sie angeben, wenn Sie Empfehlungen mit dem \$GENRE Parameter erhalten. Der zweite Ausdruck schließt Elemente aus, auf die der Benutzer geklickt oder gestreamt hat. Empfehlungen enthalten nur Elemente mit einem von Ihnen angegebenen Genre, auf die nicht geklickt oder gestreamt wurde.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE) | EXCLUDE ItemID WHERE  
Interactions.EVENT_TYPE IN ("click", "stream")
```


Filtern von Empfehlungen in Echtzeit

Sie können Empfehlungen in Echtzeit mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder den AWS SDKs filtern.

Wenn Sie personalisierte Artikelempfehlungen oder ähnliche Artikel erhalten, können Sie in Ihrer Anfrage eine Hochstufung angeben. Bei einer Hochstufung wird ein Filter verwendet, um zusätzliche Geschäftsregeln zu definieren, die für eine konfigurierbare Teilmenge empfohlener Elemente gelten. Weitere Informationen finden Sie unter [Werbung für Artikel in Empfehlungen](#).

Themen

- [Filtern von Empfehlungen in Echtzeit \(Konsole\)](#)
- [Filtern von Empfehlungen in Echtzeit \(AWS CLI\)](#)
- [Filtern von Echtzeitempfehlungen \(AWS SDKs\)](#)

Filtern von Empfehlungen in Echtzeit (Konsole)

Um Empfehlungen in Echtzeit mithilfe der Konsole zu filtern, erstellen Sie einen Filter und wenden Sie ihn dann auf eine Empfehlungsanfrage an.

Note

Um Empfehlungen mithilfe eines Filters mit Parametern und einer Kampagne zu filtern, die vor dem 10. November 2020 bereitgestellt wurde, müssen Sie die Kampagne mithilfe der [-UpdateCampaign](#) Operation erneut bereitstellen oder eine neue Kampagne erstellen.

Erstellen eines Filters (Konsole)

Um einen Filter in der Konsole zu erstellen, wählen Sie die Datensatzgruppe aus, die die Kampagne oder den Empfehlungsgeber enthält, mit dem Sie gefilterte Empfehlungen erhalten möchten. Geben Sie dann einen Filternamen und einen Filterausdruck an.

So erstellen Sie einen Filter (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.

- Wählen Sie die Datensatzgruppe aus, die die Kampagne oder den Empfehler enthält, die bzw. den Sie verwenden möchten, um gefilterte Empfehlungen zu erhalten.
- Wählen Sie im Navigationsbereich Filter und dann Neuen Filter erstellen aus. Die Seite Filter erstellen wird angezeigt.

Create filter Info

Use filters to include or exclude items from Amazon Personalize recommendations. To create a filter, provide a filter name and filter expression.

Filter configuration

Filter name
The filter name that you enter here can help you distinguish this filter from others.

The filter name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Expression
The expressions specify what to include or exclude from your recommendations.

Build expression
Select this option to build an expression using the expression builder tool.

Input expression
Select this option if you have an existing expression or prefer to input text.

Build expression Info
Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$*" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

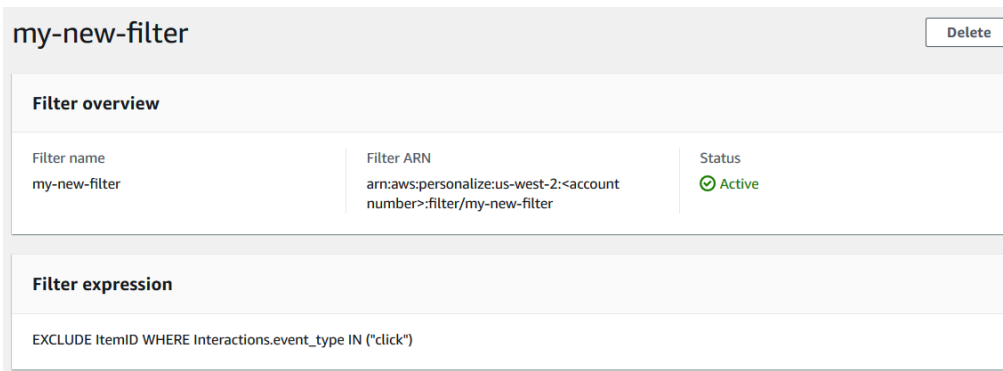
Action	ID	Property	Operator	Value
Exclude ▾	ItemID ▾	WHERE	Interactions.event_type ▾	IN ▾ Value or \$PARAMETER
				+

► **Tags - optional (0) Info**

A tag is an administrative label that you assign to AWS resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your AWS costs.

- Geben Sie unter Filtername einen Namen für Ihren Filter ein. Sie wählen den Filter nach diesem Namen aus, wenn Sie ihn auf eine Empfehlungsanfrage anwenden.
- Wählen Sie für Ausdruck entweder Ausdruck erstellen oder Ausdruck manuell hinzufügen aus und erstellen oder fügen Sie Ihren Ausdruck ein:
 - Um den expression Builder zu verwenden, wählen Sie Ausdruck erstellen aus. Der expression Builder bietet Struktur, Felder und Richtlinien für die Erstellung korrekt formatierter Filterausdrücke. Weitere Informationen finden Sie unter [Verwenden des Filter expression Builders](#).

- Um Ihren eigenen Ausdruck einzugeben, wählen Sie Ausdruck manuell hinzufügen aus. Weitere Informationen finden Sie unter [Filterausdruckselemente](#).
6. Wählen Sie Finish (Abschließen). Auf der Übersichtsseite des Filters werden der Amazon-Ressourcenname (ARN), der Status und der vollständige Filterausdruck des Filters angezeigt. Um den Filter zu löschen, wählen Sie Löschen aus. Informationen zum Suchen und Löschen von Filtern, nachdem Sie die Übersichtsseite verlassen haben, finden Sie unter [Löschen eines Filters \(Konsole\)](#).



Anwenden eines Filters (Konsole)

Um einen Filter anzuwenden, wählen Sie unter Testempfehlung (für Empfehler) oder Testkampagnenergebnisse (für benutzerdefinierte Kampagnen) den Filter aus und geben Sie beliebige Filterparameterwerte ein. Holen Sie sich dann Empfehlungen für einen Benutzer.

Important

Für Filterausdrücke, die ein `-INCLUDE` Element verwenden, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein `-EXCLUDE` Element verwenden, können Sie weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

So wenden Sie einen Filter an (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie die Datensatzgruppe aus, die die Kampagne oder den Empfehler enthält, die bzw. den Sie verwenden möchten, um gefilterte Empfehlungen zu erhalten.

3. Führen Sie je nach Datensatzgruppentyp oder Ressourcentyp einen der folgenden Schritte aus:
 - a. Wählen Sie für eine Domain-Datensatzgruppe im Navigationsbereich Empfehlung aus.
 - b. Wählen Sie für eine benutzerdefinierte Datensatzgruppe oder benutzerdefinierte Ressourcen im Navigationsbereich Benutzerdefinierte Ressourcen und dann Kampagnen aus.
4. Wählen Sie auf der Seite Empfehlungen oder Kampagnen den Zielempfehlunggeber oder die Kampagne aus.
5. Zu Vergleichszwecken erhalten Sie zunächst Empfehlungen, ohne einen Filter anzuwenden. Geben Sie unter Ergebnisse der Testempfehlung/Testkampagne die ID eines Benutzers ein, für den Sie Empfehlungen erhalten möchten, oder die ID des Elements für verwandte Elemente und wählen Sie Empfehlungen abrufen aus. Eine Tabelle mit den wichtigsten Empfehlungen wird angezeigt.

Test campaign results

User ID [Info](#)

This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your user-interactions or user dataset.

Filter name

Select the filter you want to apply to this recommendation.

To find a filter, [go to the filter page](#).

Get recommendations

Item ID	Score
3948	0.0107270
1676	0.0069995
2657	0.0064348
2985	0.0055178
2081	0.0054022

6. Wählen Sie im Menü Filtername den Filter aus, den Sie erstellt haben. Wenn Ihr Filter Platzhalterparameter hat, werden die zugehörigen Felder für jeden Parameter angezeigt.
7. Wenn Sie einen Filter mit Platzhalterparametern verwenden, geben Sie für jeden Parameter den Wert ein, um die Filterkriterien festzulegen. Um mehrere Werte für einen Parameter zu verwenden, trennen Sie jeden Wert durch ein Komma.
8. Wählen Sie mit der gleichen User ID oder Item ID wie im vorherigen Schritt Empfehlungen abrufen aus. Die Tabelle mit Empfehlungen wird angezeigt.

Wenn der Benutzer beispielsweise bereits ein empfohlenes Element gekauft hat, entfernt der Filter es aus der Empfehlungsliste. In diesem Beispiel wurden die Elemente 2657, 2985 durch die am besten geeigneten Elemente ersetzt, die der Benutzer nicht gekauft hat (Elemente 2641 und 1573).

Verwenden des Filter expression Builders

Der Ausdrucks-BUILDER auf der Seite Filter erstellen bietet Struktur, Felder und Richtlinien für die Erstellung eines korrekt formatierten Filters

Select this option to build an expression using the expression builder tool.

Select this option if you have an existing expression or prefer to input text.

Build expression Info
Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

Action	ID	Property	Operator	Value	
Exclude ▼	ItemID ▼	WHERE	Interactions.event_type ▼	IN ▼	Value or \$PARAMETER
		AND ▼	Interactions.event_type ▼	IN ▼	Value or \$PARAMETER

+

Add expression

So erstellen Sie einen Filterausdruck:

- Verwenden Sie die Felder Typ , Aktion , Eigenschaft , Operator und Wert, um einen Ausdruck zu erstellen.

Geben Sie für Wert einen festen Wert ein oder geben Sie \$ + einen Parameternamen ein, um Filterkriterien festzulegen, wenn Sie Empfehlungen erhalten. Beispiel: \$GENRES Wenn Sie Empfehlungen erhalten, geben Sie den Wert oder die Werte an, nach denen gefiltert werden soll. In diesem Beispiel würden Sie ein Genre oder eine Liste von Genres angeben, wenn Sie Empfehlungen erhalten.

Trennen Sie mehrere Nicht-Parameterwerte durch ein Komma. Sie können einem Filter keine kommasetrennten Parameter hinzufügen.

Note

Nachdem Sie eine Eigenschaft (im `dataset.field` Format) ausgewählt haben, muss der Eigenschaftswert für alle nachfolgenden Zeilen, die durch - AND oder -OR Bedingungen verkettet sind, denselben verwendendataset.

- Verwenden Sie die Schaltflächen + und X, um eine Zeile zu Ihrem Ausdruck hinzuzufügen oder daraus zu löschen. Sie können die erste Zeile nicht löschen.
- Verwenden Sie für neue Zeilen die OR Operatoren AND, oder im Menü ANDIF, um eine Kette von Bedingungen zu erstellen.

Für IF Bedingungen:

- Jeder Ausdruck darf nur ein IF Element enthalten. Wenn Sie eine IF-Bedingung entfernen, entfernt der Ausdrucksgenerator alle darauf folgenden AND Bedingungen.
- Sie können IF Bedingungen nur für Ausdrücke verwenden, die nach dem `filterCurrentUser`.
- Wählen Sie die Schaltfläche Ausdruck hinzufügen, um einen zusätzlichen Filterausdruck für eine genauere Filterung hinzuzufügen. Jeder Ausdruck wird zuerst unabhängig ausgewertet und das Ergebnis ist eine Vereinigung der beiden Ergebnisse.

Note

Um einen Filter zu erstellen, der sowohl Element- als auch Elementinteraktionsdatensätze oder Aktions- und Aktionsinteraktionsdatensätze verwendet, müssen Sie mehrere Ausdrücke verwenden.

Beispiel für Expression Builder

Das folgende Beispiel zeigt, wie Sie einen Filter erstellen, der Elemente mit einem Genre ausschließt, das Sie angeben, wenn Sie Empfehlungen erhalten (beachten Sie den Platzhalterparameter `$GENRES`). Der Filter schließt auch Elemente mit einem `DOWNLOAD_COUNT` von mehr als `aus200`, jedoch nur, wenn das Alter des aktuellen Benutzers größer als `ist17`.

Build expression Info

Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

Action	Property	Operator	Value
Exclude ▼	ItemID WHERE	Items.GENRES ▼	IN ▼ \$GENRES
AND ▼	Items.DOWNLOAD_COUNT ▼	> ▼	200
IF ▼	currentUser.AGE ▼	> ▼	17

Add expression

Löschen eines Filters (Konsole)

Durch das Löschen eines Filters wird der Filter aus der Liste der Filter für eine Datensatzgruppe entfernt.

Important

Sie können einen Filter nicht löschen, während ein Batch-Inferenzanfrage ausgeführt wird.

So löschen Sie einen Filter (Konsole)

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie in der Liste Datensatzgruppen die Datensatzgruppe aus, die den Filter enthält, den Sie löschen möchten.
3. Wählen Sie im Navigationsbereich Filter aus.
4. Wählen Sie aus der Filterliste den Filter aus, den Sie löschen möchten, und wählen Sie Details anzeigen aus. Die Seite mit den Filterdetails wird angezeigt.
5. Wählen Sie Löschen und bestätigen Sie den Löschvorgang im Bestätigungsdiaologfeld.

Filtern von Empfehlungen in Echtzeit (AWS CLI)

Um Empfehlungen mit der zu filtern AWS CLI, erstellen Sie einen Filter und wenden ihn dann an, indem Sie den Filter-ARN in einer - [GetRecommendations](#) oder - [GetPersonalizedRanking](#)Anforderung angeben.

⚠ Important

Um Empfehlungen mithilfe eines Filters mit Parametern und einer Kampagne zu filtern, die Sie vor dem 10. November 2020 bereitgestellt haben, müssen Sie die Kampagne mithilfe des [-UpdateCampaign](#)Aufrufs erneut bereitstellen oder eine neue Kampagne erstellen.

Erstellen eines Filters (AWS CLI)

Verwenden Sie die folgende `create-filter` Operation, um einen Filter zu erstellen und den Filterausdruck anzugeben.

Ersetzen Sie durch `Filter name` den Namen des Filters und durch `Dataset group ARN` den Amazon-Ressourcennamen (ARN) der Datensatzgruppe. Ersetzen Sie das Beispiel `filter-expression` durch Ihren eigenen Filterausdruck.

```
aws personalize create-filter \  
  --name Filter name \  
  --dataset-group-arn dataset group arn \  
  --filter-expression "EXCLUDE ItemID WHERE Items.CATEGORY IN (\'$CATEGORY\')"
```

Bei Erfolg wird der Filter-ARN angezeigt. Notieren Sie es zur späteren Verwendung. Um zu überprüfen, ob der Filter aktiv ist, verwenden Sie die [-DescribeFilter](#) Operation, bevor Sie den Filter verwenden.

Weitere Informationen zur API finden Sie unter [CreateFilter](#). Weitere Informationen zu Filterausdrücken, einschließlich Beispielen, finden Sie unter [Filterausdrucksstruktur und -elemente](#).

Anwenden eines Filters (AWS CLI)

Wenn Sie die `get-personalized-ranking` Operationen `get-recommendations`, `get-action-recommendations` oder verwenden, wenden Sie einen Filter an, indem Sie die Filterwerte `filter-arn` und als Parameter übergeben.

Im Folgenden finden Sie ein Beispiel für die `-get-recommendations` Operation. Ersetzen Sie durch `Campaign ARN` den Amazon-Ressourcennamen (ARN) Ihrer Kampagne `User ID` durch die ID des Benutzers, für den Sie Empfehlungen erhalten, und `Filter ARN` durch den ARN Ihres Filters. Wenn Sie Empfehlungen von einem Empfehler anstelle einer Kampagne erhalten, verwenden Sie `recommender-arn` anstelle von `--campaign-arn` und geben Sie den ARN für den Empfehler an.

Wenn Ihr Ausdruck Parameter enthält, schließen Sie das `-filter-values` Objekt ein. Geben Sie für jeden Parameter in Ihrem Filterausdruck den Parameternamen (wobei die Groß- und Kleinschreibung beachtet werden muss) und die Werte an. Wenn Ihr Filterausdruck beispielsweise einen `$GENRE` Parameter hat, geben Sie „GENRE“ als Schlüssel und ein Genre oder Genre, z. B. "Comedy", als Wert an. Trennen Sie mehrere Werte durch ein Komma. Beispiel: `"\"comedy\"","\"drama\"","\"horror\""`

Important

Für Filterausdrücke, die ein `-INCLUDE` Element verwenden, um Elemente einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein `-EXCLUDE` Element zum Ausschließen von Elementen verwenden, können Sie weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

```
aws personalize-runtime get-recommendations \
  --campaign-arn Campaign ARN \
  --user-id User ID \
  --filter-arn Filter ARN \
  --filter-values '{
    "Parameter name": "\"value\"",
    "Parameter name": "\"value1\",\"value2\",\"value3\""
  }'
```

Löschen eines Filters (AWS CLI)

Verwenden Sie die folgende `delete-filter` Operation, um einen Filter zu löschen. Ersetzen Sie `filter ARN` durch den ARN des Filters.

```
aws personalize delete-filter --filter-arn Filter ARN
```

Filtern von Echtzeitempfehlungen (AWS SDKs)

Um Empfehlungen mithilfe der - AWS SDKs zu filtern, erstellen Sie einen Filter und wenden ihn dann an, indem Sie den Filter-ARN in einer - [GetRecommendations](#) oder - [GetPersonalizedRanking](#) Anforderung angeben.

⚠ Important

Um Empfehlungen mithilfe eines Filters mit Parametern und einer Kampagne zu filtern, die Sie vor dem 10. November 2020 bereitgestellt haben, müssen Sie die Kampagne mithilfe des [-UpdateCampaign](#)-Aufrufs erneut bereitstellen oder eine neue Kampagne erstellen.

Erstellen eines Filters (AWS SDKs)

Erstellen Sie einen neuen Filter mit der [-CreateFilter](#)-Operation. Der folgende Code zeigt, wie Sie einen Filter erstellen. Geben Sie den Filternamen, den Amazon-Ressourcennamen (ARN) Ihrer Datensatzgruppe und Ihren Filterausdruck an.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_filter(
    name = 'Filter Name',
    datasetGroupArn = 'Dataset Group ARN',
    filterExpression = 'EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)'
)
filter_arn = response["filterArn"]
print("Filter ARN: " + filter_arn)
```

SDK for Java 2.x

```
public static String createFilter(PersonalizeClient personalizeClient,
                                String filterName,
                                String datasetGroupArn,
                                String filterExpression) {
    try {
        CreateFilterRequest request = CreateFilterRequest.builder()
            .name(filterName)
            .datasetGroupArn(datasetGroupArn)
            .filterExpression(filterExpression)
            .build();

        return personalizeClient.createFilter(request).filterArn();
    }
}
```

```
    }  
    catch(PersonalizeException e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
    return "";  
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.  
import { CreateFilterCommand } from  
    "@aws-sdk/client-personalize";  
import { personalizeClient } from "../libs/personalizeClients.js";  
// Or, create the client here.  
// const personalizeClient = new PersonalizeClient({ region: "REGION"});  
  
// Set the filter's parameters.  
export const createFilterParam = {  
    datasetGroupArn: 'DATASET_GROUP_ARN', /* required */  
    name: 'NAME', /* required */  
    filterExpression: 'FILTER_EXPRESSION' /*required */  
}  
  
export const run = async () => {  
    try {  
        const response = await personalizeClient.send(new  
CreateFilterCommand(createFilterParam));  
        console.log("Success", response);  
        return response; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

Notieren Sie den Filter-ARN zur späteren Verwendung. Um zu überprüfen, ob der Filter aktiv ist, verwenden Sie die [-DescribeFilter](#) Operation, bevor Sie den Filter verwenden. Weitere Informationen zur API finden Sie unter [CreateFilter](#). Weitere Informationen zu Filterausdrücken, einschließlich Beispielen, finden Sie unter [Filterausdrucksstruktur und -elemente](#).

Anwenden eines Filters (AWS SDKs)

Wenn Sie die `GetPersonalizedRanking` Operationen `GetRecommendations`, `GetActionRecommendations`, oder verwenden, wenden Sie einen Filter an, indem Sie einen `filterArn` und alle Filterwerte als Parameter übergeben.

Der folgende Code zeigt, wie Sie gefilterte Amazon-Personalize-Elementempfehlungen für einen Benutzer abrufen. Geben Sie die ID des Benutzers an, für den Sie Empfehlungen erhalten möchten, den Amazon-Ressourcennamen (ARN) Ihrer Kampagne und den ARN Ihres Filters. Wenn Sie Empfehlungen von einem Empfehler anstelle einer Kampagne erhalten, verwenden Sie `recommenderArn` anstelle von `campaignArn` und geben Sie den ARN für den Empfehler an.

Geben Sie `filterValues` für jeden optionalen Parameter in Ihrem Filterausdruck den Parameternamen (wobei die Groß- und Kleinschreibung beachtet werden muss) und den Wert oder die Werte an. Wenn Ihr Filterausdruck beispielsweise einen `$GENRES` Parameter hat, geben Sie „GENRES“ als Schlüssel und ein Genre oder Genre, z. B. `"\"Comedy\""`, als Wert an. Trennen Sie bei mehreren Werten jeden Wert durch ein Komma. Beispiel: `"\"comedy\"\", \"drama\"\", \"horror\""`

Important

Für Filterausdrücke, die ein `-INCLUDEElement` verwenden, um Elemente einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein `-EXCLUDEElement` zum Ausschließen von Elementen verwenden, können Sie weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

SDK for Python (Boto3)

```
import boto3

personalize_runtime = boto3.client("personalize-runtime")

response = personalize_runtime.get_recommendations(
    campaignArn = "Campaign ARN",
    userId = "User ID",
    filterArn = "Filter ARN",
    filterValues = {
        "Parameter name": "\"value1\"",
    }
```

```

        "Parameter name": "\"value1\", \"value2\", \"value3\""
        ....
    }
)

```

SDK for Java 2.x

Im folgenden Beispiel werden zwei Parameter verwendet, einer mit zwei Werten und einer mit einem Wert. Ändern Sie je nach Filterausdruck den Code, um die Felder `parameterName` und `parameterValue` hinzuzufügen oder zu entfernen.

```

public static void getFilteredRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
                                String campaignArn,
                                String userId,
                                String filterArn,
                                String parameter1Name,
                                String parameter1Value1,
                                String parameter1Value2,
                                String parameter2Name,
                                String parameter2Value){

    try {

        Map<String, String> filterValues = new HashMap<>();

        filterValues.put(parameter1Name, String.format("\"%1$s\", \"%2$s\"",
parameter1Value1, parameter1Value2));
        filterValues.put(parameter2Name, String.format("\"%1$s\"",
parameter2Value));

        GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
            .campaignArn(campaignArn)
            .numResults(20)
            .userId(userId)
            .filterArn(filterArn)
            .filterValues(filterValues)
            .build();

        GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient.getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();
    }
}

```

```

        for (PredictedItem item: items) {
            System.out.println("Item Id is : "+item.itemId());
            System.out.println("Item score is : "+item.score());
        }
    } catch (PersonalizeRuntimeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
    "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here:
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
    "REGION"});

// Set recommendation request parameters.
export const getRecommendationsParam = {
    campaignArn: 'CAMPAIGN_ARN', /* required */
    userId: 'USER_ID', /* required */
    numResults: 15, /* optional */
    filterArn: 'FILTER_ARN', /* required to filter recommendations */
    filterValues: {
        "PROPERTY": "\"VALUE\"" /* Only required if your filter has a placeholder
parameter */
    }
}

export const run = async () => {
    try {
        const response = await personalizeRuntimeClient.send(new
GetRecommendationsCommand(getRecommendationsParam));
        console.log("Success!", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};

```

```
run();
```

Löschen eines Filters (AWS Python SDK)

Verwenden Sie die folgende `delete_filter` Methode, um einen Filter zu löschen. Ersetzen Sie `filter Arn` durch den ARN des Filters.

```
import boto3
personalize = boto3.client("personalize")

response = personalize.delete_filter(
    filterArn = "filter ARN"
)
```

Filtern von Batch-Empfehlungen und Benutzersegmenten (benutzerdefinierte Ressourcen)

Das Filtern von Batch-Empfehlungen und Benutzersegmenten funktioniert fast genauso wie das Filtern von Echtzeitempfehlungen. Es folgt demselben Workflow, der unter [beschrieben wird](#) [Batch-Empfehlungen und Benutzersegmente \(benutzerdefinierte Ressourcen\)](#). Um Batch-Empfehlungen oder Benutzersegmente zu filtern, gehen Sie wie folgt vor:

1. Erstellen Sie einen Filter wie bei Empfehlungen in Echtzeit. Weitere Informationen finden Sie unter [Filtern von Empfehlungen in Echtzeit](#).
2. Bereiten Sie Ihre Eingabedaten vor und laden Sie sie auf Amazon S3 hoch, wie unter [Vorbereiten von Eingabedaten für Batch-Empfehlungen](#) oder [beschrieben](#) [Vorbereiten von Eingabedaten für Benutzersegmente](#). Wenn Ihr Filter Platzhalterparameter verwendet, müssen Sie ein zusätzliches `filterValues` Objekt hinzufügen. Weitere Informationen finden Sie unter [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#). Wenn Ihr Filter keine Platzhalterparameter verwendet, können Ihre Eingabedaten den Beispielen in folgen. [Beispiele für Batch-Inferenzanfrageingaben und -ausgabe-JSON](#) [Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON](#)
3. Erstellen Sie einen separaten Speicherort für Ihre Ausgabedaten, entweder einen Ordner oder einen anderen Amazon S3-Bucket.
4. Erstellen Sie einen [Batch-Inferenzanfrage](#) oder einen [Batch-Segmentauftrag](#). Geben Sie beim Erstellen des Auftrags den Amazon-Ressourcennamen (ARN) Ihres Filters an.

5. Wenn die Batch-Inferenz oder der Batch-Segment-Auftrag abgeschlossen ist, rufen Sie die Empfehlungen oder Benutzersegmente von Ihrem Ausgabespeicherort in Amazon S3 ab.

Themen

- [Bereitstellen von Filterwerten in Ihrem Eingabe-JSON](#)
- [Filtern von Batch-Workflows \(Konsole\)](#)
- [Filtern von Batch-Workflows \(AWS -SDKs\)](#)

Bereitstellen von Filterwerten in Ihrem Eingabe-JSON

Für Filter mit Platzhalterparametern, z. B. \$GENRE, müssen Sie die Werte für die Parameter in einem `filterValues` Objekt in Ihrem Eingabe-JSON angeben. Für ein `filterValues` Objekt ist jeder Schlüssel ein Parametername. Jeder Wert ist das Kriterium, das Sie als Parameter übergeben. Umgeben Sie jeden Wert mit Escape-Anführungszeichen: `"filterValues": {"GENRES": "\"drama\""}.` Trennen Sie bei mehreren Werten jeden Wert durch ein Komma: `"filterValues": {"GENRES": "\"horror\"", "\"comedy\"", "\"drama\""}.`

JSON-Beispiel für die Eingabe von Batch-Inferenzaufträgen

Im Folgenden finden Sie ein Beispiel für die ersten Zeilen einer JSON-Eingabedatei für einen Batch-Inferenzauftrag. Das Beispiel enthält das `-filterValues` Objekt. Der `GENRES` Schlüssel entspricht einem `$GENRES` Platzhalter im Filterausdruck. Der Auftrag in diesem Beispiel verwendet das User-Personalization-Rezept. Geben Sie für `RELATED_ITEMS`-Rezepte eine `itemId` anstelle der `userId` an. Geben Sie für `PERSONALIZED_RANKING`-Rezepte die `userId` und eine `itemList` an.

```
{"userId": "5", "filterValues": {"GENRES": "\"horror\"", "\"comedy\"", "\"drama\""}}
{"userId": "3", "filterValues": {"GENRES": "\"horror\"", "\"comedy\""}}
{"userId": "34", "filterValues": {"GENRES": "\"drama\""}}}
```

Weitere Beispiele für Eingabedaten für Batch-Inferenzaufträge nach Rezept finden Sie unter [Beispiele für Batch-Inferenzauftragseingaben und -ausgabe-JSON](#). Sie können diese Beispiele als Ausgangspunkt verwenden und das `filterValues` Objekt aus dem obigen Beispiel hinzufügen.

JSON-Beispiel für die Eingabe von Batchsegmentaufträgen

Im Folgenden finden Sie ein Beispiel für die ersten Zeilen einer JSON-Eingabedatei mit Filterwerten für einen Batch-Segmentauftrag. Der GENRES Schlüssel entspricht einem \$GENRES Platzhalter im Filterausdruck.

```
{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\n\n\", \"filterValues\": {\"COUNTRY\": \"Japan\"}}\n{\"itemAttributes\": \"ITEMS.genres = \"Horror\n\n\", \"filterValues\": {\"COUNTRY\": \"United\nStates\"}}\n{\"itemAttributes\": \"ITEMS.genres = \"Action\" AND ITEMS.genres = \"Adventure\n\n\", \"filterValues\": {\"COUNTRY\": \"England\"}}
```

Weitere Beispiele für Eingabedaten für Batch-Inferenzanfragen nach Rezept finden Sie unter [Beispiele für Batch-Segment-Auftragseingaben und -ausgabe-JSON](#). Sie können diese Beispiele als Ausgangspunkt verwenden und das `filterValues` Objekt aus dem obigen Beispiel hinzufügen.

Filtern von Batch-Workflows (Konsole)

Um Batch-Workflows mit der Amazon-Personalize-Konsole zu filtern, erstellen Sie einen Filter und dann einen Batch-Inferenzanfrage oder Batch-Segmentauftrag und wählen den Filter aus. Eine vollständige Schritt-für-Schritt-Anleitung finden Sie unter [Erstellen eines Batch-Inferenzanfrags \(Konsole\)](#) und [Erstellen eines Batch-Segmentauftrags \(Konsole\)](#).

Filtern von Batch-Workflows (AWS -SDKs)

Um Batch-Empfehlungen mit den - AWS SDKs zu filtern, erstellen Sie einen Filter und fügen Sie den `FilterArn` Parameter in die - [CreateBatchInferenceJob](#) oder - [CreateBatchSegmentJob](#) Anforderung ein.

Der folgende Code zeigt, wie Sie einen Batch-Inferenzanfrage mit einem Filter mithilfe der erstellen AWS SDK for Python (Boto3). Wir empfehlen, einen anderen Speicherort für Ihre Ausgabedaten zu verwenden (entweder einen Ordner oder einen anderen Amazon S3-Bucket). Eine vollständige Erläuterung aller Felder finden Sie unter [Erstellen eines Batch-Inferenzanfrags \(AWS SDKs\)](#).

```
import boto3\n\npersonalize = boto3.client(\"personalize\")\n\npersonalize_rec.create_batch_inference_job (
```

```
solutionVersionArn = "Solution version ARN",
jobName = "Batch job name",
roleArn = "IAM role ARN",
filterArn = "Filter ARN",
jobInput =
    {"s3DataSource": {"path": "S3 input path"}},
jobOutput = e
    {"S3DataDestination": {"path": "S3 output path"}}
)
```

Messung der Wirkung von Empfehlungen

Wenn Ihre Kunden mit Empfehlungen interagieren, können Sie messen, wie die Empfehlungen Ihnen helfen, Ihre Ziele zu erreichen. Sie können herausfinden, welche Kampagnen und Empfehlungen die größte Wirkung haben, z. B. welche Ressource die meisten angesehenen Minuten oder die meisten Klicks generiert. Und Sie können die Leistung von Amazon Personalize mit denen von Drittanbieterdiensten vergleichen.

Folgendes kann Ihnen helfen, die Wirkung von Empfehlungen zu messen:

- [Metrik-Zuordnung](#): Eine Amazon Personalize Personalize-Metrik-Zuordnung erstellt Berichte, die auf von Ihnen angegebenen Metriken und den Artikelinteraktionen und Artikeldaten, die Sie importieren, basieren. Zum Beispiel die Gesamtlänge der von Benutzern angesehenen Filme oder die Gesamtzahl der Klickereignisse.
- [A/B-Tests](#): Die Durchführung eines A/B-Tests besteht darin, ein Experiment mit mehreren Varianten durchzuführen und die Ergebnisse zu vergleichen. Mithilfe von A/B-Tests können Sie verschiedene Empfehlungsstrategien vergleichen und bewerten und die Wirkung der Empfehlungen messen.

Themen

- [Messung der Wirkung von Empfehlungen mit einer metrischen Zuordnung](#)
- [Wie Sie den Einfluss von Empfehlungen mit A/B-Tests messen](#)

Messung der Wirkung von Empfehlungen mit einer metrischen Zuordnung

Um die Wirkung von Artikelempfehlungen zu messen, können Sie eine metrische Zuordnung erstellen. Mit einer metrischen Zuordnung werden Berichte erstellt, die auf den Artikelinteraktionen und Artikeldaten, die Sie importieren, sowie auf den von Ihnen angegebenen Metriken basieren. Zum Beispiel die Gesamtlänge der von Benutzern angesehenen Filme oder die Gesamtzahl der Klickereignisse. Amazon Personalize aggregiert Berechnungen über einen Zeitraum von 15 Minuten. Für PutEvents inkrementelle Massendaten sendet Amazon Personalize automatisch Metrikberichte an Amazon. CloudWatch Für Massendaten können Sie wählen, ob Sie Berichte in einem Amazon S3 S3-Bucket veröffentlichen möchten.

Fügen Sie für jede Interaktion, die Sie importieren, Quelldaten hinzu, um verschiedene Kampagnen, Empfehlungsgeber und Drittanbieter zu vergleichen. Sie können die Empfehlungs-ID der Empfehlungen, die Sie dem Benutzer gezeigt haben, oder die Ereignisquelle, z. B. von Drittanbietern, angeben.

Möglicherweise haben Sie eine Video-Streaming-App, die Filmempfehlungen von zwei verschiedenen Amazon Personalize Personalize-Empfehlungsgebern anzeigt. Wenn Sie herausfinden möchten, welcher Empfehlungsgeber die meisten Wiedergabeereignisse generiert, könnten Sie eine metrische Zuordnung erstellen, die die Gesamtzahl der Wiedergabeereignisse erfasst. Dann könnten Sie die Wiedergabeereignisse aufzeichnen, wenn Nutzer mit Empfehlungen interagieren, und diese `recommendationId` in jedes Ereignis einbeziehen. Amazon Personalize verwendet die `recommendationId`, um jeden Empfehlungsgeber zu identifizieren. Während Sie Ereignisse aufzeichnen, können Sie sich die Gesamtsumme der Ereignisse ansehen, die alle 15 Minuten für beide Empfehlungsgeber zusammengefasst wurden. CloudWatch Codebeispiele, die zeigen, wie Sie ein `recommendationId` oder ein `eventAttributionSource` für ein Ereignis angeben, finden Sie unter [Ereignismetriken und Attributionsberichte](#)

Themen

- [Richtlinien und Anforderungen](#)
- [Eine metrische Zuordnung erstellen](#)
- [Verwaltung einer metrischen Zuordnung](#)
- [Ergebnisse veröffentlichen und anzeigen](#)

Richtlinien und Anforderungen

Amazon Personalize beginnt erst, die Auswirkungen von Empfehlungen zu berechnen und zu melden, nachdem Sie eine Metrik-Attribution erstellt haben. Um einen möglichst vollständigen Verlauf zu erstellen, empfehlen wir, vor dem Import Ihrer Interaktionsdaten eine metrische Zuordnung zu erstellen. Wenn Sie mit der Amazon Personalize Personalize-Konsole einen Datensatz-Importjob für einen Datensatz mit Artikelinteraktionen erstellen, haben Sie die Möglichkeit, auf einer neuen Registerkarte eine Metrik-Zuordnung zu erstellen. Anschließend können Sie zum Importauftrag zurückkehren, um ihn abzuschließen.

Nachdem Sie eine Metrikzuweisung erstellt und Ereignisse aufgezeichnet oder inkrementelle Massendaten importiert haben, fallen einige monatliche CloudWatch Kosten pro Metrik an. Informationen zu den CloudWatch Preisen finden Sie auf der Seite mit den [CloudWatch Amazon-](#)

[Preisen](#). Um das Senden von Metriken an zu beenden CloudWatch, [löschen Sie die Metrik-Zuordnung](#).

Um zu sehen, wie sich Empfehlungen im Laufe der Zeit auswirken, sollten Sie weiterhin Daten importieren, während Kunden mit Empfehlungen interagieren. Wenn Sie bereits Daten importiert haben, können Sie trotzdem eine metrische Zuordnung erstellen und damit beginnen, die Wirkung von Empfehlungen zu messen. Amazon Personalize berichtet jedoch nicht über Daten, die Sie vor der Erstellung importiert haben.

Im Folgenden finden Sie Richtlinien und Anforderungen für die Erstellung von Berichten mit einer metrischen Zuordnung:

- Sie müssen Amazon Personalize die Erlaubnis erteilen, auf Daten zuzugreifen und sie einzugeben. CloudWatch Richtlinienbeispiele finden Sie unter [Amazon Personalize Zugriff gewähren auf CloudWatch](#).
- Um Metriken in Amazon S3 zu veröffentlichen, erteilen Sie Amazon Personalize die Erlaubnis, in Ihren Bucket zu schreiben. Sie müssen auch den Bucket-Pfad in Ihrer Metrik-Zuordnung angeben. Richtlinienbeispiele finden Sie unter [Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).
- Um Metriken veröffentlichen zu können CloudWatch, müssen die Datensätze weniger als 14 Tage alt sein. Wenn Ihre Daten älter sind, werden diese Datensätze nicht in Berechnungen oder Berichten berücksichtigt.
- Das Importieren doppelter Ereignisse (Ereignisse, die exakt mit allen Attributen übereinstimmen) kann zu unerwartetem Verhalten, einschließlich ungenauer Messwerte, führen. Wir empfehlen, doppelte Datensätze vor dem Import aus allen Massendaten zu entfernen und zu vermeiden, dass doppelte Ereignisse während des PutEvents Vorgangs importiert werden.
- Ihr Datensatz mit Artikelinteraktionen muss eine EVENT_TYPE Spalte enthalten.
- Sie können keine Metrikberichte für Daten in einem Datensatz mit Aktionsinteraktionen erstellen.
- Sie können höchstens eine Metrik-Attribution pro Datensatzgruppe erstellen. Jede Metrik-Attribution kann maximal 10 Metriken enthalten.

Um Quellen vergleichen zu können, muss jedes Interaktionsereignis ein `recommendationId` oder `eventAttributionSource` enthalten. Sie können maximal 100 eindeutige Quellen für die Zuordnung von Ereignissen angeben. PutEvents-Codebeispiele finden Sie unter [Ereignismetriken und Attributionsberichte](#).

- Wenn Sie eine `recommendationId` angeben, bestimmt Amazon Personalize automatisch die Quellkampagne oder den Empfehlungsgeber und identifiziert ihn in Berichten in einer `EVENT_ATTRIBUTION_SOURCE`-Spalte.
- Wenn Sie beide Attribute angeben, verwendet Amazon Personalize nur `eventAttributionSource`.
- Wenn Sie keine Quelle angeben, kennzeichnet Amazon Personalize die Quelle `SOURCE_NAME_UNDEFINED` in Berichten.

Themen

- [Amazon Personalize Zugriff gewähren auf CloudWatch](#)
- [Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#)

Amazon Personalize Zugriff gewähren auf CloudWatch

Wichtig

Wenn Sie Berechtigungen erteilen, platziert und validiert Amazon Personalize eine kleine Datenmenge. CloudWatch Dabei fallen einmalige Kosten von weniger als 0,30\$ an. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie auf der Seite mit den [CloudWatch Amazon-Preisen](#).

Um Amazon Personalize Zugriff darauf zu gewähren CloudWatch, fügen Sie Ihrer Amazon Personalize Personalize-Servicerolle eine neue AWS Identity and Access Management (IAM-) Richtlinie hinzu, die der Rolle die Berechtigung erteilt, die `PutMetricData` Aktion für zu verwenden. CloudWatch Das folgende Richtlinienbeispiel gewährt Berechtigungen. `PutMetricData`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren

So gewähren Sie Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket:

- Fügen Sie Ihrer Amazon Personalize Personalize-Servicerolle eine IAM-Richtlinie hinzu, die der Rolle die Erlaubnis erteilt, die PutObject Aktion in Ihrem Bucket zu verwenden.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

- Fügen Sie Ihrem Amazon S3 S3-Ausgabe-Bucket eine Bucket-Richtlinie hinzu, die dem Amazon Personalize Personalize-Prinzip die Erlaubnis erteilt, die PutObject Aktionen zu verwenden.

Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": "personalize.amazonaws.com"
    },
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"
    ]
  }
]
```

Eine metrische Zuordnung erstellen

Important

Nachdem Sie eine Metrik-Zuordnung erstellt und Ereignisse aufgezeichnet oder inkrementelle Massendaten importiert haben, fallen einige monatliche CloudWatch Kosten pro Metrik an. Informationen zu den CloudWatch Preisen finden Sie auf der Seite mit den [CloudWatch Amazon-Preisen](#). Um das Senden von Metriken an zu beenden CloudWatch, [löschen Sie die Metrik-Zuordnung](#).

Um mit der Generierung von Metrikberichten zu beginnen, erstellen Sie eine Metrikzuweisung und importieren Interaktionsdaten. Wenn Sie eine Metrik-Attribution erstellen, geben Sie eine Liste von Ereignistypen an, über die berichtet werden soll. Für jeden Ereignistyp geben Sie eine Funktion an, die Amazon Personalize bei der Erfassung der Daten anwendet. Zu den verfügbaren Funktionen gehören `SUM(DatasetType.COLUMN_NAME)` und `SAMPLECOUNT()`.

Angenommen, Sie haben eine Online-Videostreaming-App und möchten zwei Messwerte verfolgen: die Klickrate für Empfehlungen und die Gesamtlänge der angesehenen Filme, wobei jedes Video im Artikeldatensatz ein `LENGTH` Attribut enthält. Sie würden eine Metrikzuweisung erstellen und zwei Metriken hinzufügen, die jeweils einen Ereignistyp und eine Funktion haben. Die erste könnte für den `Click` Ereignistyp mit einer `SAMPLECOUNT()` Funktion gelten. Die zweite könnte für den `Watch` Ereignistyp mit einer `SUM(Items.LENGTH)` Funktion sein.

Sie können SUM() Funktionen nur auf numerische Spalten von Datensätzen vom Typ Items und Item Interactions anwenden. Um eine SUM() Funktion auf eine Spalte in einem Artikel-Datensatz anzuwenden, müssen Sie zuerst Elementmetadaten importieren.

Sie können eine Metrik-Attribution mit der Amazon Personalize Personalize-Konsole oder AWS Command Line Interface AWS SDKs erstellen.

Themen

- [Eine metrische Zuordnung erstellen \(Konsole\)](#)
- [Eine metrische Zuordnung erstellen \(\) AWS CLI](#)
- [Eine metrische Attribution \(AWSSDKs\) erstellen](#)

Eine metrische Zuordnung erstellen (Konsole)

Um eine Metrik-Attribution mit der Amazon Personalize Personalize-Konsole zu erstellen, navigieren Sie zur Seite Metrik-Zuordnung und wählen Metrik-Attribution erstellen. Wenn Sie eine Metrik-Attribution erstellen, geben Sie einen optionalen Amazon S3 S3-Bucket-Pfad, Ihre Amazon Personalize IAM-Service-Rolle und eine Liste von Metriken an, über die Sie Berichte erstellen möchten.

Wenn Sie mit der Amazon Personalize Personalize-Konsole einen Job zum Importieren von Datensätzen mit Artikelinteraktionen erstellen, haben Sie die Möglichkeit, auf einer neuen Registerkarte eine Metrik-Attribution zu erstellen. Anschließend können Sie zum Importjob zurückkehren, um ihn abzuschließen. Wenn Sie sich bereits auf der Seite Metrik-Attribution konfigurieren befinden, können Sie mit Schritt 4 fortfahren.

Um eine Metrik-Attribution zu erstellen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich unter Benutzerdefinierte Ressourcen die Option Metrische Zuordnung aus.
4. Wählen Sie unter Details zur metrischen Zuordnung die Option Metrik-Attribution erstellen aus.
5. Geben Sie auf der Seite Metrik-Attribution konfigurieren der Metrik-Attribution einen Namen.
6. Wenn Sie Metriken für den Amazon S3-Datenausgabepfad in Amazon S3 veröffentlichen möchten, geben Sie den Amazon S3-Ziel-Bucket ein. Dies ermöglicht die Option, Metriken jedes

Mal zu veröffentlichen, wenn Sie einen Datensatz-Importauftrag erstellen. Verwenden Sie die folgende Syntax:

s3://<name of your S3 bucket>/<folder> path>

7. Wenn Sie AWS KMS für die Verschlüsselung den KMS-Schlüssel ARN verwenden, geben Sie den Amazon-Ressourcennamen (ARN) für den AWS KMS Schlüssel ein. Sie müssen Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).
8. Wählen Sie in der IAM-Rolle, ob Sie eine neue Servicerolle erstellen oder eine bestehende verwenden möchten. Die Rolle, für CloudWatch die Sie sich entscheiden, muss über PutMetricData Berechtigungen verfügen. Wenn Sie auf Amazon S3 veröffentlichen möchten, muss die Rolle über PutObject Berechtigungen für Ihren Amazon S3 S3-Bucket verfügen.

Um die Rolle zu verwenden, in der Sie erstellt haben [Erstellen einer IAM-Rolle für Amazon Personalize](#), müssen Sie möglicherweise Richtlinien für CloudWatch Amazon S3 hinzufügen.

Beispiele für Richtlinien finden Sie unter [Amazon Personalize Zugriff gewähren auf CloudWatch](#) und [Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).

9. Wählen Sie Weiter.
10. Wählen Sie auf der Seite Metrikattribute definieren aus, wie Metriken definiert werden sollen. Wählen Sie Metrikattribute erstellen, um das Builder-Tool zu verwenden. Wählen Sie Metrikattribute eingeben, um Metriken im JSON-Format einzugeben.
 - Wenn Sie Metrikattribute erstellen wählen, geben Sie für jede Metrik einen Namen und einen Ereignistyp an und wählen Sie eine Funktion aus. Wählen Sie für SUM() Funktionen den Spaltennamen aus. Wählen Sie Metrikattribut hinzufügen, um weitere Metriken hinzuzufügen.
 - Wenn Sie Metrikattribute eingeben wählen, geben Sie jede Metrik im JSON-Format ein. Im Folgenden wird gezeigt, wie eine Metrik formatiert wird.

```
{
  "EventType": "watch",
  "MetricName": "MinutesWatchedTracker",
  "MetricMathExpression": "SUM(Items.LENGTH)"
}
```

11. Wählen Sie Weiter.

12. Überprüfen Sie auf der Seite Überprüfen und erstellen die Details für die neue Metrik-Attribution. Wählen Sie Previous (Zurück) aus, um Änderungen vorzunehmen. Um die Metrik-Attribution zu erstellen, wählen Sie Erstellen aus. Wenn die Metrik-Attribution aktiv ist, können Sie mit dem Import von Daten beginnen und die Ergebnisse anzeigen. Informationen zum Anzeigen von Ergebnissen finden Sie unter [Ergebnisse veröffentlichen und anzeigen](#).

Eine metrische Zuordnung erstellen () AWS CLI

Der folgende Code zeigt, wie Sie eine metrische Zuordnung mit dem erstellen. AWS Command Line Interface Die von Ihnen angegebene Rolle muss über PutMetricData Berechtigungen für CloudWatch und, falls Sie auf Amazon S3 veröffentlichen, PutObject über Berechtigungen für Ihren Amazon S3 S3-Bucket verfügen. Um die Rolle zu verwenden, in der Sie erstellt haben [Erstellen einer IAM-Rolle für Amazon Personalize](#), müssen Sie möglicherweise Richtlinien für CloudWatch Amazon S3 hinzufügen. Beispiele für Richtlinien finden Sie unter [Amazon Personalize Zugriff gewähren auf CloudWatch](#) und [Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).

Geben Sie für jede Metrik einen Namen, einen Ereignistyp und einen Ausdruck (eine Funktion) an. Zu den verfügbaren Funktionen gehören SUM(DatasetType.COLUMN_NAME) und SAMPLECOUNT(). Geben Sie für SUM() -Funktionen den Datensatztyp und den Spaltennamen an. Zum Beispiel SUM(Items.LENGTH). Informationen zu den einzelnen Parametern finden Sie unter [CreateMetricAttribution](#).

```
aws personalize create-metric-attribution \
--name metric attribution name \
--dataset-group-arn dataset group arn \
--metrics-output-config "{\"roleArn\": \"Amazon Personalize service role ARN\",
  \"s3DataDestination\": {\"kmsKeyArn\": \"kms key ARN\", \"path\": \"s3://bucket-
  name/folder-name/\"}}\" \
--metrics "[{
  \"eventType\": \"event type\",
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",
  \"metricName\": \"metric name\"
}]\"
```

Eine metrische Attribution (AWSSDKs) erstellen

Der folgende Code zeigt, wie Sie mit dem SDK for Python (Boto3) eine metrische Zuordnung erstellen. Die von Ihnen angegebene Rolle muss über PutMetricData Berechtigungen für CloudWatch und, falls Sie auf Amazon S3 veröffentlichen, PutObject über Berechtigungen für Ihren

Amazon S3 S3-Bucket verfügen. Um die Rolle zu verwenden, in der Sie erstellt haben [Erstellen einer IAM-Rolle für Amazon Personalize](#), müssen Sie möglicherweise Richtlinien für CloudWatch Amazon S3 hinzufügen. Beispiele für Richtlinien finden Sie unter [Amazon Personalize Zugriff gewähren auf CloudWatch](#) und [Amazon Personalize Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).

Geben Sie für jede Metrik einen Namen, einen Ereignistyp und einen Ausdruck (eine Funktion) an. Zu den verfügbaren Funktionen gehören `SUM(DatasetType.COLUMN_NAME)` und `SAMPLECOUNT()`. Geben Sie für `SUM()`-Funktionen den Datensatztyp und den Spaltennamen an. Zum Beispiel `SUM(Items.LENGTH)`. Informationen zu den einzelnen Parametern finden Sie unter [CreateMetricAttribution](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsList = [{
    "eventType": "event type",
    "expression": "SUM(DatasetType.COLUMN_NAME)",
    "metricName": "metric name"
}]

outputConfig = {
    "roleArn": "Amazon Personalize service role ARN",
    "s3DataDestination": {
        "kmsKeyArn": "key ARN",
        "path": "s3://<name of your S3 bucket>/<folder>"
    }
}

response = personalize.create_metric_attribution(
    name = 'metric attribution name',
    datasetGroupArn = 'dataset group arn',
    metricsOutputConfig = outputConfig,
    metrics = metricsList
)

metric_attribution_arn = response['metricAttributionArn']

print ('Metric attribution ARN: ' + metric_attribution_arn)

description = personalize.describe_metric_attribution(
```

```
metricAttributionArn = metric_attribution_arn)['metricAttribution']

print('Name: ' + description['name'])
print('ARN: ' + description['metricAttributionArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createMetricAttribution(PersonalizeClient personalizeClient,
                                             String eventType,
                                             String expression,
                                             String metricName,
                                             String metricAttributionName,
                                             String roleArn,
                                             String s3Path,
                                             String kmsKeyArn,
                                             String datasetGroupArn) {

    String metricAttributionArn = "";

    try {

        MetricAttribute attribute = MetricAttribute.builder()
            .eventType(eventType)
            .expression(expression)
            .metricName(metricName)
            .build();

        ArrayList<MetricAttribute> metricAttributes = new ArrayList<>();
        metricAttributes.add(attribute);

        S3DataConfig s3DataDestination = S3DataConfig.builder()
            .kmsKeyArn(kmsKeyArn)
            .path(s3Path)
            .build();

        MetricAttributionOutput outputConfig = MetricAttributionOutput.builder()
            .roleArn(roleArn)
            .s3DataDestination(s3DataDestination)
            .build();

        CreateMetricAttributionRequest createMetricAttributionRequest =
        CreateMetricAttributionRequest.builder()
            .name(metricAttributionName)
```

```

        .datasetGroupArn(datasetGroupArn)
        .metrics(metricAttributes)
        .metricsOutputConfig(outputConfig)
        .build();
    CreateMetricAttributionResponse createMetricAttributionResponse =
personalizeClient.createMetricAttribution(createMetricAttributionRequest);

    metricAttributionArn =
createMetricAttributionResponse.metricAttributionArn();
    System.out.println("Metric attribution ARN: " + metricAttributionArn);
    return metricAttributionArn;
} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateMetricAttributionCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// set the metric attribution param
export const createMetricAttributionParam = {
    name: "METRIC_ATTRIBUTION_NAME",          /* required */
    datasetGroupArn: "DATASET_GROUP_ARN",    /* required */
    metricsOutputConfig: {
        roleArn: "ROLE_ARN",                 /* required */
        s3DataDestination: {
            kmsKeyArn: "KEY_ARN",             /*
optional */
            path: "s3://<name of your output S3 bucket>/<folderName>/", /* optional */
        },
    },
    metrics: [
        {
            eventType: "EVENT_TYPE",         /* required for each metric */

```

```
        expression: "SUM(DatasetType.COLUMN_NAME)", /* required for each metric */
        metricName: "METRIC_NAME", /* required for each metric */
    }
]
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new CreateMetricAttributionCommand(createMetricAttributionParam)
        );
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

Verwaltung einer metrischen Zuordnung

Nachdem Sie eine Metrik-Attribution erstellt haben, können Sie sie aktualisieren oder löschen. Wenn Sie eine Metrikzuweisung löschen, sendet Amazon Personalize keine Berichte mehr zu PutEvents und inkrementelle Importe an. CloudWatch

Themen

- [Aktualisierung einer metrischen Zuordnung](#)
- [Löschen einer metrischen Zuordnung](#)

Aktualisierung einer metrischen Zuordnung

Wenn Sie eine Metrik-Attribution aktualisieren, können Sie Metriken hinzufügen und entfernen und die Ausgabekonfiguration ändern. Sie können eine Metrikzuweisung mit der Amazon Personalize Personalize-Konsole oder AWS dem AWS Command Line Interface SDKS aktualisieren.

Themen

- [Aktualisierung einer Metrik-Zuordnung \(Konsole\)](#)

- [Aktualisierung einer Metrik-Zuordnung \(\) AWS CLI](#)
- [Aktualisierung einer metrischen Zuordnung \(AWSSDK\)](#)

Aktualisierung einer Metrik-Zuordnung (Konsole)

Um eine Metrik-Zuordnung mit der Amazon Personalize-Konsole zu aktualisieren, nehmen Sie Ihre Änderungen auf der Seite Metrik-Zuordnung vor.

Um eine Metrik-Zuordnung zu aktualisieren

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Metrische Zuordnung aus.
4. Wählen Sie im unteren Bereich die Registerkarte Metrikattribute oder die Registerkarte Konfiguration der metrischen Zuordnung aus, um mit den Änderungen zu beginnen.
 - Um Metriken hinzuzufügen oder zu entfernen, wählen Sie den Tab Metrikattribute und dann Attribute bearbeiten aus. Nehmen Sie Ihre Änderungen auf der Seite Metrikattribute bearbeiten vor und wählen Sie Aktualisieren, um Ihre Änderungen zu speichern.
 - Um Änderungen am Amazon S3 S3-Ausgabe-Bucket oder der IAM-Servicerolle vorzunehmen, wählen Sie die Registerkarte Konfiguration der Metrikzuweisung bearbeiten und nehmen Sie die Änderungen auf der Seite Attributionskonfiguration bearbeiten vor. Wählen Sie Update (Aktualisieren), um Ihre Änderungen zu speichern.

Aktualisierung einer Metrik-Zuordnung () AWS CLI

Nachdem Sie eine Metrik-Attribution erstellt haben, können Sie die AWS Command Line Interface (AWS CLI) verwenden, um Metriken hinzuzufügen und zu entfernen und die Ausgabekonfiguration zu ändern. Der folgende Code zeigt, wie Sie Metriken mit dem `update-metric-attribution` Befehl entfernen:

```
aws personalize update-metric-attribution \  
--metric-attribution-arn metric attribution arn \  
--remove-metrics metricName1 metricName2
```


Der folgende Code zeigt, wie Sie eine zusätzliche Metrik hinzufügen und eine neue Ausgabekonfiguration angeben:

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--metrics-output-config "{\"roleArn\": \"new role ARN\", \"s3DataDestination\":  
{\"kmsKeyArn\": \"kms key ARN\", \"path\": \"s3://new-bucket-name/new-folder-name/\"}}\" \
--add-metrics "[{  
  \"eventType\": \"event type\",  
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",  
  \"metricName\": \"metric name\"  
}]"
```

Bei Erfolg gibt Amazon Personalize den ARN der Metrik-Attribution zurück, die Sie aktualisiert haben. Eine vollständige Liste aller Parameter finden Sie unter [UpdateMetricAttribution](#)

Aktualisierung einer metrischen Zuordnung (AWSSDK)

Nachdem Sie eine Metrik-Attribution erstellt haben, können Sie Metriken hinzufügen oder entfernen und deren Ausgabekonfiguration ändern. Der folgende Code zeigt, wie Sie Metriken aus einer Metrik-Attribution entfernen.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsToRemove = ["metricName1", "metricName2"]

response = personalize.update_metric_attribution(
    metricAttributionArn = "metric attribution ARN",
    removeMetrics = metricsToRemove
)
```

SDK for Java 2.x

```
public static void removeMetrics(PersonalizeClient client,
                                String metricAttributionArn,
                                String metric1Name,
                                String metric2Name) {
```

```

    ArrayList<String> metricsToRemove = new ArrayList<>(Arrays.asList(metric1Name,
metric2Name));

    try {

        UpdateMetricAttributionRequest request =
UpdateMetricAttributionRequest.builder()
            .metricAttributionArn(metricAttributionArn)
            .removeMetrics(metricsToRemove)
            .build();

        UpdateMetricAttributionResponse response =
client.updateMetricAttribution(request);
        System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// set the update request param
export const updateMetricAttributionParam = {
    metricAttributionArn: "METRIC_ATTRIBUTION_ARN",    /* required */
    removeMetrics: ["METRIC_NAME_1", "METRIC_NAME_2"] /* specify list of names of
metrics to delete */
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new UpdateMetricAttributionCommand(updateMetricAttributionParam)
        );
        console.log("Success", response);
    }
}

```

```

    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

Der folgende Code zeigt, wie Sie eine zusätzliche Metrik hinzufügen und eine neue Ausgabekonfiguration angeben:

SDK for Python (Boto3)

```

import boto3

personalize = boto3.client('personalize')

newMetrics = [{
    "eventType": "event type",
    "expression": "SUM(DatasetType.COLUMN_NAME)",
    "metricName": "metric name"
}]

newOutputConfig = {
    "roleArn": "Amazon Personalize service role ARN",
    "s3DataDestination": {
        "kmsKeyArn": "key ARN",
        "path": "s3://<name of your S3 bucket>/<folder>"
    }
}

response = personalize.update_metric_attribution(
    metricAttributionArn = "metric attribution arn",
    metricsOutputConfig = newOutputConfig,
    addMetrics = newMetrics
)

```

SDK for Java 2.x

```

public static void addMetricsAndUpdateOutputConfig(PersonalizeClient
personalizeClient,
                                                    String metricAttributionArn,
                                                    String newMetric1EventType,

```

```
String newMetric1Expression,  
String newMetric1Name,  
String newMetric2EventType,  
String newMetric2Expression,  
String newMetric2Name,  
String roleArn,  
String s3Path,  
String kmsKeyArn) {  
  
try {  
  
    MetricAttribute newAttribute = MetricAttribute.builder()  
        .eventType(newMetric1EventType)  
        .expression(newMetric1Expression)  
        .metricName(newMetric1Name)  
        .build();  
  
    MetricAttribute newAttribute2 = MetricAttribute.builder()  
        .eventType(newMetric2EventType)  
        .expression(newMetric2Expression)  
        .metricName(newMetric2Name)  
        .build();  
  
    ArrayList<MetricAttribute> newAttributes = new  
ArrayList<>(Arrays.asList(newAttribute, newAttribute2));  
  
    S3DataConfig newDataDestination = S3DataConfig.builder()  
        .kmsKeyArn(kmsKeyArn)  
        .path(s3Path)  
        .build();  
  
    MetricAttributionOutput newOutputConfig = MetricAttributionOutput.builder()  
        .roleArn(roleArn)  
        .s3DataDestination(newDataDestination)  
        .build();  
  
    UpdateMetricAttributionRequest request =  
UpdateMetricAttributionRequest.builder()  
        .metricAttributionArn(metricAttributionArn)  
        .metricsOutputConfig(newOutputConfig)  
        .addMetrics(newAttributes)  
        .build();  
  
    UpdateMetricAttributionResponse response =  
personalizeClient.updateMetricAttribution(request);
```

```

        System.out.println("New metrics added!");
        System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

export const updateMetricAttributionParam = {
    metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
    addMetrics: [
        {
            eventType: "EVENT_TYPE",           /* required for each metric */
            expression: "SUM(DatasetType.COLUMN_NAME)", /* required for each metric */
            metricName: "METRIC_NAME",         /* required for each metric */
        }
    ],
    metricsOutputConfig: {
        roleArn: "ROLE_ARN",                   /* required */
        s3DataDestination: {
            kmsKeyArn: "KEY_ARN",              /*
optional */
            path: "s3://<name of your output S3 bucket>/<folderName>/", /* optional */
        },
    }
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new UpdateMetricAttributionCommand(updateMetricAttributionParam)
        );
    }
}

```

```
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Bei Erfolg gibt Amazon Personalize den ARN der Metrik-Attribution zurück, die Sie aktualisiert haben. Eine vollständige Liste aller Parameter finden Sie unter [UpdateMetricAttribution](#)

Löschen einer metrischen Zuordnung

Wenn Sie keine Berichte mehr erstellen möchten, können Sie eine metrische Zuordnung löschen. Durch das Löschen einer Metrikzuweisung werden alle zugehörigen Metriken und die Ausgabekonfiguration gelöscht.

Wenn Sie eine Metrik-Zuordnung löschen, sendet Amazon Personalize nicht mehr automatisch Berichte zu PutEvents und inkrementellen Massendaten an. CloudWatch Daten, die bereits an Amazon S3 gesendet CloudWatch oder dort veröffentlicht wurden, sind nicht betroffen. Sie können eine Metrik-Attribution mit der Amazon Personalize Personalize-Konsole oder AWS Command Line Interface AWS SDKs löschen.

Themen

- [Löschen einer metrischen Zuordnung \(Konsole\)](#)
- [Löschen einer metrischen Zuordnung \(\) AWS CLI](#)
- [Löschen einer metrischen Zuordnung \(SDKs\) AWS](#)

Löschen einer metrischen Zuordnung (Konsole)

Sie löschen eine Metrik-Attribution auf der Übersichtsseite für Ihre Metrik-Attribution.

Um eine Metrik-Attribution zu löschen

1. Öffnen Sie die Amazon Personalize Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem Konto an.
2. Wählen Sie Ihre Datensatzgruppe aus.
3. Wählen Sie im Navigationsbereich Metrische Zuordnung aus.

4. Wählen Sie Löschen und bestätigen Sie dann den Löschvorgang.

Löschen einer metrischen Zuordnung () AWS CLI

Um eine Metrik-Attribution mit dem zu löschenAWS CLI, verwenden Sie den `delete-metric-attribution` Befehl wie folgt.

```
aws personalize delete-metric-attribution --metric-attribution-arn metric attribution ARN
```

Löschen einer metrischen Zuordnung (SDKs) AWS

Der folgende Code zeigt, wie eine metrische Zuordnung mit dem SDK for Python (Boto3) gelöscht wird:

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.delete_metric_attribution(
    metricAttributionArn = 'metric attribution ARN'
)
```

SDK for Java 2.x

```
public static void deleteMetricAttribution(PersonalizeClient client, String
metricAttributionArn) {

    try {

        DeleteMetricAttributionRequest request =
DeleteMetricAttributionRequest.builder()
                .metricAttributionArn(metricAttributionArn)
                .build();

        DeleteMetricAttributionResponse response =
client.deleteMetricAttribution(request);
        if (response.sdkHttpResponse().statusCode() == 200) {
            System.out.println("Metric attribution deleted!");
        }
    }
}
```

```
    }

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { DeleteMetricAttributionCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

export const deleteMetricAttributionParam = {
    metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
};
export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new DeleteMetricAttributionCommand(deleteMetricAttributionParam)
        );
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

Ergebnisse veröffentlichen und anzeigen

Amazon Personalize sendet die Berichte für jede Metrik an CloudWatch oder Amazon S3:

- Für PutEvents Daten und inkrementelle Massendaten sendet Amazon Personalize automatisch Metriken an. CloudWatch Informationen zum Anzeigen und Identifizieren von Berichten in finden Sie CloudWatch unter. [Metriken anzeigen in CloudWatch](#)

- Wenn Sie bei der Erstellung Ihrer Metrik-Attribution einen Amazon S3 S3-Bucket angeben, können Sie für alle Massendaten wählen, dass jedes Mal, wenn Sie einen Datensatz-Importjob für Interaktionsdaten erstellen, Metrikberichte in Ihrem Amazon S3 S3-Bucket veröffentlicht werden.

Informationen zum Veröffentlichen von Metrikberichten in Amazon S3 finden Sie unter [Veröffentlichen von Metriken auf Amazon S3](#).

Themen

- [Metriken anzeigen in CloudWatch](#)
- [Veröffentlichen von Metriken auf Amazon S3](#)

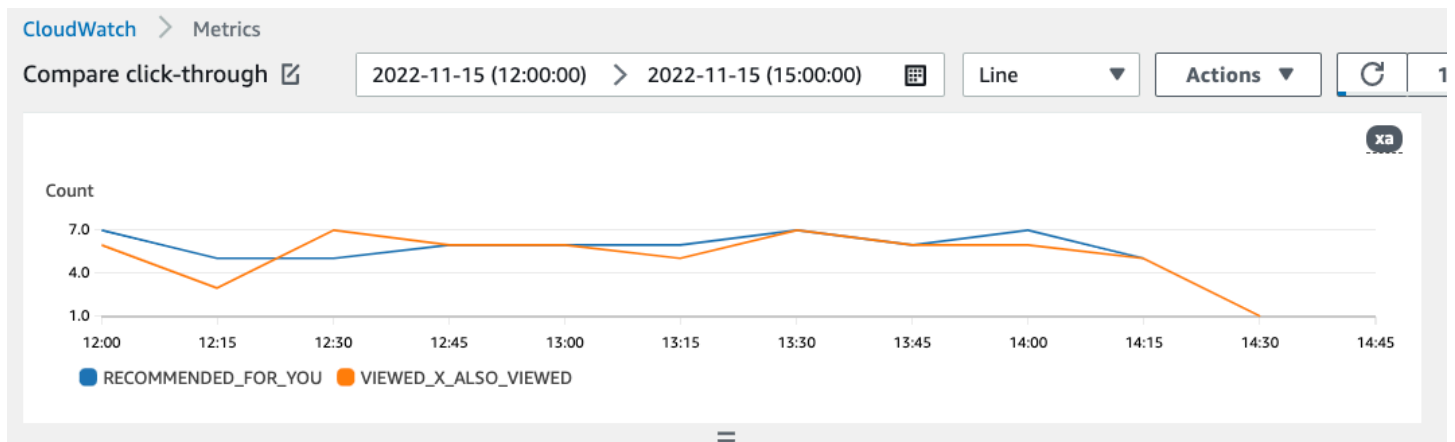
Metriken anzeigen in CloudWatch

Important

Nachdem Sie eine Metrik-Zuordnung erstellt und Ereignisse aufgezeichnet oder inkrementelle Massendaten importiert haben, fallen einige monatliche CloudWatch Kosten pro Metrik an. Informationen zu den CloudWatch Preisen finden Sie auf der Seite mit den [CloudWatch Amazon-Preisen](#). Um das Senden von Metriken an zu beenden CloudWatch, [löschen Sie die Metrik-Zuordnung](#).

Um Metriken in anzuzeigen CloudWatch, gehen Sie wie unter Metrik [grafisch darstellen beschrieben vor](#). Der Mindestzeitraum, den Sie grafisch darstellen können, beträgt 15 Minuten. Geben Sie für den Suchbegriff den Namen ein, den Sie der Metrik bei der Erstellung der Metrik-Attribution gegeben haben.

Im Folgenden finden Sie ein Beispiel dafür, wie eine Metrik in CloudWatch erscheinen könnte. Die Metrik zeigt die Klickrate für zwei verschiedene Empfehlungen alle 15 Minuten.



Veröffentlichen von Metriken auf Amazon S3

Um Metriken in Amazon S3 zu veröffentlichen, geben Sie in Ihrer Metrik-Zuordnung einen Pfad zu Ihrem Amazon S3 S3-Bucket an. Anschließend veröffentlichen Sie Berichte in Amazon S3, wenn Sie einen Datensatz-Importauftrag erstellen.

Wenn der Job abgeschlossen ist, finden Sie die Metriken in Ihrem Amazon S3 S3-Bucket. Jedes Mal, wenn Sie Metriken veröffentlichen, erstellt Amazon Personalize eine neue Datei in Ihrem Amazon S3 S3-Bucket. Der Dateiname beinhaltet die Importmethode und das Datum wie folgt:

`AggregatedAttributionMetrics - ImportMethod - Timestamp.csv`

Im Folgenden finden Sie ein Beispiel dafür, wie die ersten Zeilen einer CSV-Datei mit einem Metrikbericht aussehen könnten. Die Metrik in diesem Beispiel gibt Auskunft über die Gesamtzahl der Klicks von zwei verschiedenen Empfehlungen in Intervallen von 15 Minuten. Jeder Empfehlungsgeber wird durch seinen Amazon-Ressourcennamen (ARN) in der Spalte `EVENT_ATTRIBUTION_SOURCE` identifiziert.

```
METRIC_NAME,EVENT_TYPE,VALUE,MATH_FUNCTION,EVENT_ATTRIBUTION_SOURCE,TIMESTAMP
COUNTWATCHES,WATCH,12.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender1Name,1666925124
COUNTWATCHES,WATCH,112.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender2Name,1666924224
COUNTWATCHES,WATCH,10.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender1Name,1666924224
COUNTWATCHES,WATCH,254.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender2Name,1666922424
COUNTWATCHES,WATCH,112.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender1Name,1666922424
```

```
COUNTWATCHES, WATCH, 100.0, samplecount, arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name, 1666922424
.....
.....
```

Veröffentlichen von Metriken für Massendaten in Amazon S3 (Konsole)

Um Metriken mit der Amazon Personalize-Konsole in einem Amazon S3-Bucket zu veröffentlichen, erstellen Sie einen Datensatz-Import-Job und wählen Sie unter Event-Metriken in S3 veröffentlichen die Option Metriken für diesen Importjob veröffentlichen aus.

Publish event metrics to S3 - optional

When you create a metric attribution, reports related to this import job can be published to S3 for analysis with your tool of choice.

Publish metrics from this import job

S3 output destination

s3://bucketName/folderName/ [Edit metric attribution configuration](#)

step-by-step Eine Anleitung finden Sie unter [Massendatensätze importieren \(Konsole\)](#).

Veröffentlichen von Metriken für Massendaten in Amazon S3 (AWS CLI)

Um Metriken mit dem AWS Command Line Interface (AWS CLI) in einem Amazon S3 S3-Bucket zu veröffentlichen, verwenden Sie den folgenden Code, um einen Datensatz-Importjob zu erstellen und das `publishAttributionMetricsToS3` Flag bereitzustellen. Wenn Sie keine Metriken für einen bestimmten Job veröffentlichen möchten, lassen Sie das Kennzeichen weg. Informationen zu den einzelnen Parametern finden Sie unter [CreateDatasetImportJob](#).

```
aws personalize create-dataset-import-job \
--job-name dataset import job name \
--dataset-arn dataset arn \
--data-source dataLocation=s3://bucketname/filename \
--role-arn roleArn \
--import-mode INCREMENTAL \
--publish-attribution-metrics-to-s3
```

Veröffentlichen von Metriken für Massendaten in Amazon S3 (AWSSDKs)

Um Metriken mit den AWS SDKs in einem Amazon S3 S3-Bucket zu veröffentlichen, erstellen Sie einen Job zum Importieren von Datensätzen und setzen Sie ihn `publishAttributionMetricsToS3` auf `true`. Informationen zu den einzelnen Parametern finden Sie unter [CreateDatasetImportJob](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'INCREMENTAL',
    publishAttributionMetricsToS3 = True
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,

                                                    String jobName,
                                                    String datasetArn,
                                                    String s3BucketPath,
                                                    String roleArn,
                                                    ImportMode importMode,
                                                    boolean publishToS3) {
```

```
long waitInMilliseconds = 60 * 1000;
String status;
String datasetImportJobArn;

try {
    DataSource importDataSource = DataSource.builder()
        .dataLocation(s3BucketPath)
        .build();

    CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
    .datasetArn(datasetArn)
    .dataSource(importDataSource)
    .jobName(jobName)
    .roleArn(roleArn)
    .importMode(importMode)
    .publishAttributionMetricsToS3(publishToS3)
    .build();

    datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
    .datasetImportJobArn();

    DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
    .datasetImportJobArn(datasetImportJobArn)
    .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        DatasetImportJob datasetImportJob = personalizeClient
            .describeDatasetImportJob(describeDatasetImportJobRequest)
            .datasetImportJob();

        status = datasetImportJob.status();
        System.out.println("Dataset import job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
    }
}
```

```

        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
    datasetArn: 'DATASET_ARN', /* required */
    dataSource: {
        dataLocation: 's3://<name of your S3 bucket>/<folderName>/<CSVfilename>.csv' /*
required */
    },
    jobName: 'NAME', /* required */
    roleArn: 'ROLE_ARN', /* required */
    importMode: "FULL", /* optional, default is FULL */
    publishAttributionMetricsToS3: true /* set to true to publish metrics to
Amazon S3 bucket */
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
CreateDatasetImportJobCommand(datasetImportJobParam));
        console.log("Success", response);
    }
}

```

```
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Wie Sie den Einfluss von Empfehlungen mit A/B-Tests messen

Die Durchführung eines A/B-Tests besteht darin, ein Experiment mit mehreren Varianten durchzuführen und die Ergebnisse zu vergleichen. Bei der Durchführung von A/B-Tests mit Amazon Personalize Recommendations werden verschiedenen Benutzergruppen verschiedene Arten von Empfehlungen angezeigt und anschließend die Ergebnisse verglichen. Sie können A/B-Tests verwenden, um verschiedene Empfehlungsstrategien zu vergleichen und zu bewerten und die Wirkung der Empfehlungen zu messen.

Sie können beispielsweise A/B-Tests verwenden, um festzustellen, ob die Empfehlungen von Amazon Personalize die Klickrate erhöhen. Um dieses Szenario zu testen, können Sie einer Gruppe von Benutzern Empfehlungen zeigen, die nicht personalisiert sind, z. B. empfohlene Produkte. Und vielleicht zeigen Sie einer anderen Gruppe personalisierte Empfehlungen, die von Amazon Personalize generiert wurden. Wenn Ihre Kunden mit Artikeln interagieren, können Sie die Ergebnisse aufzeichnen und sehen, welche Strategie zu der höchsten Klickrate führt.

Der Arbeitsablauf für die Durchführung von A/B-Tests mit den Empfehlungen von Amazon Personalize sieht wie folgt aus:

1. Planen Sie Ihr Experiment — Definieren Sie eine quantifizierbare Hypothese, identifizieren Sie Geschäftsziele, definieren Sie Versuchsvarianten und legen Sie Ihren Versuchszeitraum fest.
2. Teilen Sie Ihre Benutzer auf — Teilen Sie die Benutzer in zwei oder mehr Gruppen mit einer Kontrollgruppe und einer oder mehreren Experimentgruppen auf.
3. Führen Sie Ihr Experiment durch — Zeigen Sie den Benutzern in der Experimentgruppe geänderte Empfehlungen. Zeigt den Benutzern in der Kontrollgruppe Empfehlungen ohne Änderungen an. Notieren Sie ihre Interaktionen mit Empfehlungen, um die Ergebnisse zu verfolgen.
4. Ergebnisse auswerten — Analysieren Sie die Versuchsergebnisse, um festzustellen, ob die Änderung für die Versuchsgruppe einen statistisch signifikanten Unterschied gemacht hat.

Sie können Amazon CloudWatch Evidently verwenden, um A/B-Tests mit Amazon Personalize Personalize-Empfehlungen durchzuführen. Mit CloudWatch Evidently können Sie Ihr Experiment definieren, wichtige Leistungsindikatoren (KPIs) verfolgen, den Traffic von Empfehlungsanfragen an die entsprechende Amazon Personalize-Ressource weiterleiten und die Ergebnisse der Experimente auswerten. Weitere Informationen finden Sie unter [A/B-Tests mit CloudWatch Evidently](#).

Themen

- [Bewährte Methoden für A/B-Tests](#)
- [A/B-Tests mit CloudWatch Evidently](#)

Bewährte Methoden für A/B-Tests

Verwenden Sie die folgenden bewährten Methoden, um A/B-Tests für Amazon-Personalisierungsempfehlungen zu entwerfen und zu verwalten.

- Identifizieren Sie ein quantifizierbares Geschäftsziel. Stellen Sie sicher, dass die verschiedenen Empfehlungen, die Sie vergleichen möchten, beide mit diesem Geschäftsziel übereinstimmen und sich nicht auf unterschiedliche oder nicht quantifizierbare Ziele beziehen.
- Definieren Sie eine quantifizierbare Hypothese, die mit Ihrem Geschäftsziel übereinstimmt. Sie könnten beispielsweise prognostizieren, dass eine Werbung für Ihre eigenen, maßgeschneiderten Inhalte zu 20% mehr Klicks auf diese Artikel führen wird. Ihre Hypothese bestimmt die Änderung, die Sie für Ihre Experimentgruppe vornehmen.
- Definieren Sie relevante Leistungsindikatoren (KPIs) in Bezug auf Ihre Hypothese. Sie verwenden KPIs, um das Ergebnis Ihrer Experimente zu messen. Dies könnten die folgenden sein:
 - Klickrate
 - Uhrzeit ansehen
 - Gesamtpreis
- Stellen Sie sicher, dass die Gesamtzahl der Benutzer im Experiment groß genug ist, um je nach Ihrer Hypothese ein statistisch signifikantes Ergebnis zu erzielen.
- Definieren Sie Ihre Strategie zur Aufteilung des Traffics, bevor Sie mit Ihrem Experiment beginnen. Vermeiden Sie es, die Verkehrsaufteilung zu ändern, während das Experiment läuft.
- Sorgen Sie dafür, dass die Benutzererfahrung Ihrer Anwendung oder Website sowohl für Ihre Experimentgruppe als auch für die Kontrollgruppe gleich bleibt, mit Ausnahme von Änderungen, die sich auf Ihr Experiment beziehen (z. B. Modell). Abweichungen in der Benutzererfahrung, wie z. B. die Benutzeroberfläche oder die Latenz, können zu irreführenden Ergebnissen führen.

- Kontrollieren Sie externe Faktoren wie Feiertage, laufende Marketingkampagnen und Browserbeschränkungen. Diese externen Faktoren können zu irreführenden Ergebnissen führen.
- Vermeiden Sie es, die Empfehlungen von Amazon Personalize zu ändern, es sei denn, dies steht in direktem Zusammenhang mit Ihrer Hypothese oder Ihren Geschäftsanforderungen. Änderungen wie das Anwenden eines Filters oder das manuelle Ändern der Reihenfolge können zu irreführenden Ergebnissen führen.
- Stellen Sie bei der Auswertung der Ergebnisse sicher, dass die Ergebnisse statistisch signifikant sind, bevor Sie Schlüsse ziehen. Der Industriestandard ist ein Signifikanzniveau von 5%. Weitere Informationen zur statistischen Signifikanz finden Sie unter [Eine Auffrischung zur statistischen Signifikanz](#).

A/B-Tests mit CloudWatch Evidently-

Nachdem Sie einen Empfehlungsgeber erstellt oder eine benutzerdefinierte Lösungsversion mit einer Kampagne bereitgestellt haben, können Sie A/B-Tests mit Amazon Personalize Recommendations und Amazon Evidently durchführen. Das folgende Video beschreibt den Prozess der Verwendung von CloudWatch Evidently zur Durchführung von A/B-Tests mit Amazon Personalize Personalize-Empfehlungen. step-by-stepAnweisungen finden Sie unter [Durchführung eines A/B-Tests mit Evidently CloudWatch](#).

[Führen Sie AB-Tests mit Amazon Personalize und CloudWatch Evidently durch](#)

Themen

- [Durchführung eines A/B-Tests mit Evidently CloudWatch](#)
- [Beispielimplementierungen](#)

Durchführung eines A/B-Tests mit Evidently CloudWatch

Um einen A/B-Test mit Amazon Personalize und Amazon CloudWatch Evidently durchzuführen, erstellen Sie ein Evidently-Projekt, definieren Sie eine CloudWatch Funktion und ihre Varianten, aktualisieren Sie Ihre Anwendung, um Ihr Experiment zu unterstützen, und erstellen Sie das Experiment und führen Sie es aus. Während das Experiment läuft, können Sie sich die Ergebnisse in CloudWatch Evidently ansehen.

Um einen A/B-Test mit Amazon Personalize und Evidently durchzuführen CloudWatch

1. Erstellen Sie ein CloudWatch Evidently-Projekt. Ein Projekt ist eine logische Gruppierung von CloudWatch Ressourcen. Innerhalb des Projekts erstellen Sie verschiedene Funktionen, die Sie dann testen oder launchen. step-by-stepAnweisungen finden Sie unter [Neues Projekt erstellen](#) im CloudWatchAmazon-Benutzerhandbuch.
2. Fügen Sie Ihrem Projekt ein Feature hinzu und definieren Sie dessen Variationen. Für dieses Experiment sollte Ihre Funktion das Empfehlungsszenario darstellen, das Sie testen möchten, z. B. die Klickrate.

Wenn Sie eine Funktion hinzufügen, geben Sie Kennungen an, um die verschiedenen Varianten Ihres Szenarios Amazon Personalize Personalize-Empfehlungen oder benutzerdefinierten Kampagnen zuzuordnen. Geben Sie für jede Variante den Variationstyp an, z. B. Zeichenfolge, geben Sie der Variante einen Namen und geben Sie ihr einen Wert.

Wenn Ihr Experiment ausgeführt wird, bestimmt Ihre Anwendung anhand des Variationswerts, welche Amazon Personalize Personalize-Ressource für Empfehlungen verwendet werden soll. Wenn Sie beispielsweise zwei VIDEO_ON_DEMAND-Empfehlungen testen, von denen einer für die Top-Picks für Ihren Anwendungsfall und einer für den Anwendungsfall Jetzt im Trend liegt, erstellt wurde, können Sie den folgenden JSON als Wert für jede Variante festlegen.

```
{"type":"top-picks-recommendations","arn":"arn:aws:personalize:us-west-2:<acct-id>:recommender/top-picks-recommender"}
```

```
{"type":"trending-recommendations","arn":"arn:aws:personalize:us-west-2:<acct-id>:recommender/trending-now-recommender"}
```

Sie können einen beliebigen Bezeichner angeben, sofern Ihre Anwendung ihn verwenden kann, um die entsprechende Ressource zu identifizieren. Sie können beispielsweise nur den Namen des Empfehlungsgebers oder der Kampagne angeben und den Amazon-Ressourcennamen (ARN) der Ressource in Ihrer Anwendung erstellen.

step-by-stepAnweisungen zum Hinzufügen einer Funktion finden [Sie unter Hinzufügen einer Funktion zu einem Projekt](#) im CloudWatchAmazon-Benutzerhandbuch.

3. Aktualisieren Sie Ihre Anwendung, um Ihr Experiment zu unterstützen:
 - Evaluierung von Funktionen — Verwenden Sie den EvaluateFeature API-Vorgang CloudWatch Evidently, um jeder Benutzersitzung Varianten zuzuweisen. Die

`EvaluateFeature` Antwort enthält den Variationswert, den Sie im vorherigen Schritt angegeben haben. In diesem Fall handelt es sich um ein JSON-Objekt mit dem Typ des Empfehlungsgebers und es ist der ARN des Empfehlungsgebers. Aktualisieren Sie Ihren Empfehlungsanforderungscode, um Empfehlungen von dieser Ressource zu erhalten.

Informationen zur Evaluierung einer Funktion finden Sie [unter Verwenden EvaluateFeature](#) im CloudWatchAmazon-Benutzerhandbuch.

- Ergebnisse aufzeichnen — Fügen Sie Ihrer Anwendung Code hinzu, um die Ergebnisse von Benutzerinteraktionen mit Empfehlungen zu verfolgen.

Um Kennzahlen für Ihre Experimente in CloudWatch Evidently zu verfolgen, verwenden Sie den CloudWatch `PutProjectEvents` Evidently-API-Vorgang, um die Ergebnisse für jeden Benutzer aufzuzeichnen. Wenn ein Benutzer in einem Experiment beispielsweise auf ein empfohlenes Element klickt, würden Sie Details zu diesem Ereignis an CloudWatch Evidently senden.

Informationen zum Senden von Ereignissen an CloudWatch Evidently finden Sie [unter Verwenden PutProjectEvents](#) im CloudWatchAmazon-Benutzerhandbuch.

Um die Relevanz der Empfehlungen von Amazon Personalize zu verbessern, können Sie Ergebnisereignisse mit der Amazon Personalize `PutEvents Personalize`-API-Operation aufzeichnen. Wenn Ihr Domain-Anwendungsfall oder Ihr benutzerdefiniertes Rezept Aktualisierungen von Empfehlungen in Echtzeit unterstützt, kann Amazon Personalize aus den letzten Aktivitäten Ihres Benutzers lernen und Empfehlungen aktualisieren, während dieser Ihre Anwendung verwendet. Wenn Amazon Personalize keine Updates unterstützt, verwendet Amazon Personalize diese Daten bei der nächsten vollständigen Umschulung Ihres Modells und wirkt sich dann auf die Empfehlungen aus.

Informationen zum Streamen von Ereignissen zu Amazon Personalize finden Sie [unter Ereignisse aufzeichnen](#).

4. Erstellen Sie ein Experiment und starten Sie es. Beim Erstellen eines Experiments geben Sie Folgendes an:
 - Funktion — Wählen Sie die Funktion aus, die im Experiment getestet werden soll.
 - Zielgruppe — Konfigurieren Sie, wie viele Ihrer Benutzer teilnehmen werden, und konfigurieren Sie, wie der Traffic zwischen den Funktionsvarianten aufgeteilt wird.
 - Metriken — Geben Sie die Metriken an, die den Erfolg des Experiments bestimmen. Zum Beispiel die Anzahl der Klicks.

Nachdem Sie die Erstellung des Experiments abgeschlossen haben, geben Sie dessen Dauer an und starten Sie das Experiment. [step-by-stepAnweisungen zum Erstellen und Starten von Experimenten in CloudWatch Evidently](#) finden Sie unter [Erstellen eines Experiments](#) im CloudWatchAmazon-Benutzerhandbuch.

5. Während Sie Ihr Experiment ausführen, können Sie die Ergebnisse im CloudWatch Evidently-Experiment-Dashboard einsehen. Informationen zum Anzeigen von Testergebnissen finden Sie unter [Testergebnisse im Dashboard anzeigen](#) im CloudWatchAmazon-Benutzerhandbuch.

Beispielimplementierungen

Die folgenden Beispielimplementierungen zeigen, wie Sie A/B-Tests mit CloudWatch Evidently implementieren.

- Ein vollständiges Beispiel für Echtzeit-APIs, die Quellcode für die Implementierung von A/B-Tests enthalten, finden Sie unter [Real-Time Personalization APIs](#) im AWS GitHub Beispiel-Repository.
- Eine Beispiel-Webanwendung für den Einzelhandel, die einen Workshop zu Personalisierung und A/B-Tests beinhaltet, finden Sie im [Retail Demo Store](#) im AWS GitHub Beispiel-Repository. Ein Notizbuch, in dem beschrieben wird, wie ein A/B-Experiment mit CloudWatch Evidently und dem Retail Demo Store durchgeführt wird, finden Sie unter [Retail Demo Store Experimentation Workshop](#) — Evidently. CloudWatch
- Ein Tutorial, das die Verwendung von A/B-Tests mit CloudWatch Evidently und eine Beispielanwendung für React beschreibt, finden Sie unter [Tutorial: A/B-Tests mit der Evidently-Beispielanwendung im Amazon-Benutzerhandbuch](#). CloudWatch

Personalisierung von Suchergebnissen von OpenSearch

Sie können Amazon Personalize verwenden, um Ergebnisse aus Open Source OpenSearch oder Amazon OpenSearch Service für Ihre Benutzer zu personalisieren.

[OpenSearch](#) ist ein selbstverwalteter Open-Source-Suchdienst, der auf der Apache 2.0-Lizenz basiert. [Amazon OpenSearch Service](#) ist ein verwalteter Service, der Sie bei der Bereitstellung, dem Betrieb und der Skalierung von OpenSearch Ressourcen in der AWS Cloud unterstützt. Wenn Sie Amazon OpenSearch Service verwenden, ruft OpenSearch Ergebnisse ab und stuft sie ein.

OpenSearch verwendet bei der Rangfolge von Abfrageergebnissen ein probabilistisches Ranking-Framework namens [BM-25](#), um Relevanzwerte zu berechnen. Wenn ein bestimmtes Schlüsselwort in einem Dokument häufiger vorkommt, weist BM-25 diesem Dokument eine höhere Relevanzbewertung zu. OpenSearch Das Ranking berücksichtigt nicht das Nutzerverhalten wie Click-Through-Daten.

Wenn Sie Amazon Personalize mit verwenden OpenSearch, ordnet Amazon Personalize die OpenSearch Ergebnisse auf der Grundlage des Verhaltens eines Benutzers in der Vergangenheit, aller Metadaten zu den Artikeln und aller Metadaten über den Benutzer neu. OpenSearch berücksichtigt dann die Neurangierung, bevor die Suchantwort an Ihre Anwendung zurückgesendet wird. Sie kontrollieren, wie viel Gewicht OpenSearch dem Ranking von Amazon Personalize beigemessen wird, wenn Sie es auf OpenSearch Ergebnisse anwenden.

Durch diese Neurangierung können die Ergebnisse ansprechender und relevanter für die Interessen eines Benutzers sein. Dies kann zu einer Erhöhung der Klickrate und der Konversionsrate für Ihre Anwendung führen. Ein Anwendungsbeispiel, das beschreibt, wie die personalisierte Suche die Ergebnisse einer E-Commerce-Anwendung verbessern kann, finden Sie unter [Beispiel für einen Anwendungsfall](#)

Bevor Sie mit der Personalisierung der OpenSearch Ergebnisse beginnen, sollten Sie die unter aufgeführten Anforderungen überprüfen. [Richtlinien und Anforderungen](#)

Themen

- [Beispiel für einen Anwendungsfall](#)
- [Personalisierter Such-Workflow](#)
- [So funktioniert das Amazon Personalize Search Ranking-Plugin](#)
- [Zusätzliche Informationen](#)

- [Richtlinien und Anforderungen](#)
- [Einrichtung OpenSearch und Installation des Plug-ins](#)
- [Konfiguration des Plug-ins](#)
- [Anwenden des Plugins auf OpenSearch Abfragen](#)
- [OpenSearch Ergebnisse mit Ergebnissen aus dem Plugin vergleichen](#)
- [Überwachung des Plugins](#)

Beispiel für einen Anwendungsfall

Wenn Sie Amazon Personalize verwenden, um OpenSearch Ergebnisse neu zu ordnen, können die Suchergebnisse für Ihre Nutzer relevanter sein. Zum Beispiel könnten Sie eine E-Commerce-Anwendung haben, die Autos verkauft. Wenn Ihr Benutzer eine Suchanfrage nach Toyota-Fahrzeugen eingibt und Sie die Ergebnisse nicht personalisieren, OpenSearch wird eine Liste von Autos angezeigt, die von Toyota auf der Grundlage von Schlüsselwörtern in Ihren Daten hergestellt wurden. Diese Liste würde für alle Benutzer in derselben Reihenfolge geordnet werden.

Wenn Sie jedoch Amazon Personalize verwenden, um Ergebnisse zu personalisieren, ordnen Sie diese OpenSearch Autos anhand ihres Verhaltens, z. B. ihrer Klicks, in der Reihenfolge ihrer Relevanz für den jeweiligen Nutzer neu ein. Das Auto, auf das der Nutzer am wahrscheinlichsten klickt, wird an erster Stelle eingestuft.

Wenn Sie OpenSearch Ergebnisse personalisieren, kontrollieren Sie, wie viel Gewicht (Betonung) OpenSearch dem Ranking von Amazon Personalize beigemessen wird. Um mit diesem Beispiel fortzufahren: Wenn ein Benutzer nach einem bestimmten Fahrzeugtyp aus einem bestimmten Jahr sucht (z. B. einem Toyota Prius von 2008), möchten Sie vielleicht mehr Wert auf das ursprüngliche Ranking von legen. OpenSearch

Bei allgemeineren Suchanfragen, die zu einer Vielzahl von Ergebnissen führen (z. B. eine Suche nach allen Toyota-Fahrzeugen), können Sie jedoch großen Wert auf Personalisierung legen. Auf diese Weise sind die Fahrzeuge, die ganz oben auf der Liste stehen, für den jeweiligen Benutzer relevanter.

Personalisierter Such-Workflow

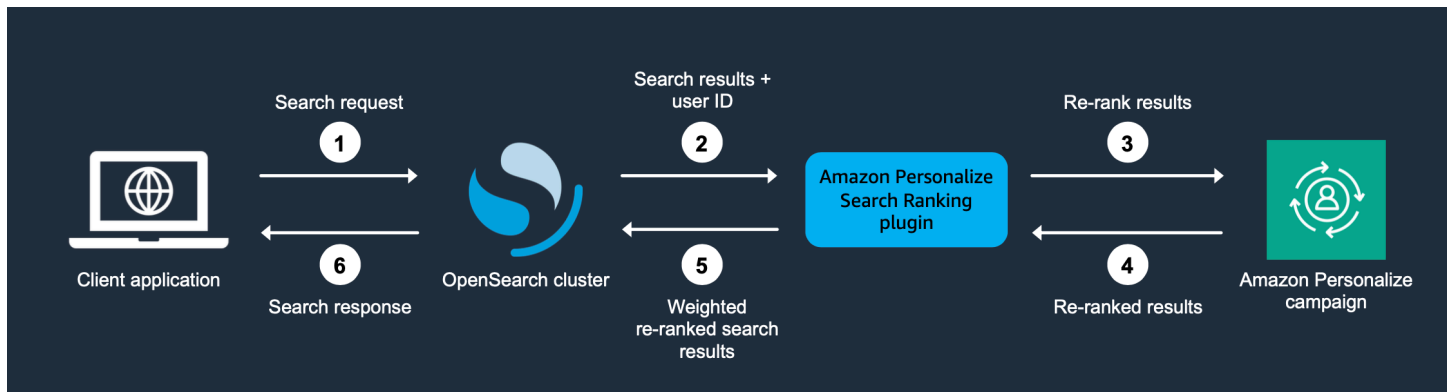
Gehen Sie wie folgt vor, um OpenSearch Ergebnisse zu personalisieren:

1. Amazon Personalize einrichten — Falls Sie dies noch nicht getan haben, führen Sie die Schritte unter [Einrichten von Amazon Personalize](#) zum Einrichten Ihrer Anmeldeinformationen und zum Einrichten von Berechtigungen für Amazon Personalize aus. Sie müssen die AWS SDKs nicht einrichten, um die Ergebnisse zu personalisieren. OpenSearch
2. Den Amazon Personalize-Workflow abschließen — Schließen Sie den Amazon Personalize Personalize-Workflow ab, um Daten zu importieren, eine Lösung mit dem Rezept für personalisiertes Ranking zu erstellen, eine benutzerdefinierte Lösungsversion zu trainieren und sie in einer Kampagne einzusetzen. Sie können nur das Rezept für personalisiertes Ranking verwenden. Sie müssen einen Datensatz mit Artikelinteraktionen erstellen. Ein Benutzerdatensatz und ein Artikeldatensatz sind optional. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Arbeitsablauf](#).
3. Richten Sie das Amazon Personalize Search Ranking-Plugin ein OpenSearch und installieren Sie es — falls Sie es noch nicht getan haben, richten Sie Ihre OpenSearch Service-Domain oder Ihren OpenSearch Open-Source-Cluster ein. Installieren Sie dann das Amazon Personalize Search Ranking-Plugin. Dieses Plugin kümmert sich um die Kommunikation mit Amazon Personalize und um die Neurangierung der Ergebnisse. Weitere Informationen finden Sie unter [Einrichtung OpenSearch und Installation des Plug-ins](#).
4. Das Amazon Personalize Search Ranking-Plugin konfigurieren — Um das Plugin zu konfigurieren, erstellen Sie Such-Pipelines. Such-Pipelines sind Gruppen von Anforderungs- und Antwortprozessoren. Wenn Sie eine Pipeline für das Plugin erstellen, geben Sie Ihre Amazon Personalize in einem `personalized_search_ranking` Antwortprozessor an. Sie konfigurieren auch, wie viel Gewicht das Plugin den Ergebnissen von Amazon Personalize beimisst, wenn es die Ergebnisse neu bewertet. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).
5. Wenden Sie das Amazon Personalize Search Ranking-Plugin auf OpenSearch Abfragen an — [Sie können das Amazon Personalize Search Ranking-Plugin auf alle Abfragen und Antworten für einen OpenSearch Index anwenden](#). Sie können das Plugin auch auf einzelne OpenSearch Abfragen anwenden. Weitere Informationen finden Sie unter [Anwenden des Plugins auf OpenSearch Abfragen](#).
6. Ergebnisse vergleichen — Das Amazon Personalize Search Ranking-Plugin ordnet die Suchergebnisse in der OpenSearch Abfrageantwort neu an. Es berücksichtigt sowohl das Ranking von Amazon Personalize als auch das Ranking von OpenSearch. Um zu verstehen, wie Ergebnisse neu eingestuft werden, können Sie Ergebnisse von Abfragen vergleichen, die Personalisierung verwenden, und solche, bei denen dies nicht der Fall ist. Weitere Informationen finden Sie unter [OpenSearch Ergebnisse mit Ergebnissen aus dem Plugin vergleichen](#).

- Überwachen Sie das Amazon Personalize Search Ranking-Plugin — Wenn Sie das Amazon Personalize Search Ranking-Plugin auf Suchanfragen anwenden, können Sie das Plugin überwachen, indem Sie Metriken für Ihre Such-Pipelines abrufen. Weitere Informationen finden Sie unter [Überwachung des Plugins](#).

So funktioniert das Amazon Personalize Search Ranking-Plugin

Das folgende Diagramm zeigt, wie das Amazon Personalize Search Ranking-Plugin funktioniert.



- Sie senden die Anfrage Ihres Kunden an Ihre OpenSearch Service-Domain oder Ihren OpenSearch Open-Source-Cluster.
- OpenSearch sendet die Antwort auf die Anfrage (Liste der Artikel, die für die Anfrage relevant sind) und die Benutzer-ID an das Amazon Personalize Search Ranking-Plugin.
- Das Plugin sendet die Artikel und den Benutzer als Antwort auf Ihre Amazon Personalize Personalize-Kampagne zum Ranking. Es verwendet die Rezeptur- und Kampagnenwerte des Amazon Resource Name (ARN) in Ihrer Suchpipeline, um ein personalisiertes Ranking für den Benutzer zu erhalten. Es verwendet den `GetPersonalizedRanking` API-Vorgang für Empfehlungen. In der Anfrage werden die `userId` des Benutzers, der die Anfrage gestellt hat, und die von der Abfrage zurückgegebenen Elemente in der OpenSearch übergeben. `inputList`
- Amazon Personalize gibt die neu eingestuft Ergebnisse an das Plugin zurück.
- Das Plugin ordnet die Suchergebnisse neu an und gibt sie an Ihre OpenSearch Service-Domain oder Ihren Open-Source-Cluster zurück. OpenSearch Die Ergebnisse werden auf der Grundlage der Reaktion auf Ihre Amazon Personalize Personalize-Kampagne und der Betonung der Personalisierung, die Sie bei der Einrichtung angegeben haben, neu eingestuft.
- Ihr OpenSearch Open-Source-Cluster oder Ihre OpenSearch Service-Domain gibt die Endergebnisse an Ihre Anwendung zurück.

Zusätzliche Informationen

Die folgenden Ressourcen bieten zusätzliche Informationen zur Verwendung von OpenSearch.

- Informationen zu den ersten Schritten mit Open Source finden Sie OpenSearch unter [Schnellstart](#).
- Informationen zu den ersten Schritten mit OpenSearch Service finden Sie unter [Erste Schritte mit Amazon OpenSearch Service](#) im Amazon OpenSearch Service Developer Guide.
- Informationen zum Rezept für personalisiertes Ranking in Amazon Personalize finden Sie unter [Personalisiertes Ranking-Rezept](#)

Richtlinien und Anforderungen

Dieser Abschnitt enthält Anforderungen für die Verwendung des Amazon Personalize Search Ranking-Plug-ins. Außerdem wird beschrieben, wie Berechtigungen für Amazon OpenSearch Service oder Open Source eingerichtet OpenSearch werden.

Themen

- [Anforderungen an das Plugin](#)
- [Amazon OpenSearch Service-Berechtigungen einrichten](#)
- [Open-Source-Berechtigungen einrichten OpenSearch](#)

Anforderungen an das Plugin

Bevor Sie mit der Personalisierung von Ergebnissen beginnen OpenSearch, beachten Sie die folgenden Richtlinien und Anforderungen für das Amazon Personalize Search Ranking-Plugin:

- Sie müssen OpenSearch Version 2.9.0 oder höher verwenden. Wenn Sie Amazon OpenSearch Service verwenden, muss Ihre Domain Version 2.9 oder höher verwenden.
- Falls Sie dies noch nicht getan haben, folgen Sie den Anweisungen unter [Einrichten von Berechtigungen](#) So gewähren Sie Ihren Benutzern die Erlaubnis, auf Amazon Personalize zuzugreifen, und geben Sie Amazon Personalize die Erlaubnis, auf Ihre Ressourcen in Amazon Personalize zuzugreifen.
- Sie müssen von Ihrer OpenSearch Service-Domain oder Ihrem OpenSearch Open-Source-Cluster aus auf Ihre Amazon Personalize Personalize-Ressourcen zugreifen können.

- Informationen zur Gewährung des Zugriffs für eine OpenSearch Service-Domain finden Sie unter [Amazon OpenSearch Service-Berechtigungen einrichten](#).
- Hinweise zur Gewährung von Zugriff für einen OpenSearch Cluster finden Sie unter [Open-Source-Berechtigungen einrichten OpenSearch](#).
- Sie können nur benutzerdefinierte Amazon Personalize verwenden. Wenn Sie eine Domain-Datensatzgruppe erstellt haben, können Sie trotzdem benutzerdefinierte Ressourcen hinzufügen.
- Sie können nur das benutzerdefinierte Rezept Personalized-Ranking verwenden. Weitere Informationen zu diesem Rezept finden Sie unter [Personalisiertes Ranking-Rezept](#)
- Sie müssen in Amazon Personalize einen Datensatz mit Artikelinteraktionen erstellen. Artikel- und Benutzerdatensätze sind optional.
- Sie können Amazon Personalize Personalize-Filter nicht anwenden, wenn Sie das Amazon Personalize Search Ranking-Plugin verwenden.
- Standardmäßig geht das Plugin davon aus, dass `_id` für ein indiziertes Dokument in der ItemID OpenSearch in Ihren Amazon Personalize-Daten angegeben ist. Wenn Ihre OpenSearch Daten ein anderes Feld verwenden, das Ihren Amazon Personalize Personalize-Artikelnummern entspricht, müssen Sie bei der Konfiguration des Plug-ins den Namen des Felds angeben.
- Die `userId`, die Sie für einen Benutzer verwenden, der eine Anfrage stellt, muss mit seiner `userId` in den Daten übereinstimmen, die Sie in Amazon Personalize importieren.
- Das Plugin ordnet nur die 500 besten Suchergebnisse neu. OpenSearch Die verbleibenden Artikel werden nicht neu eingestuft und landen am Ende der Liste.

Amazon OpenSearch Service-Berechtigungen einrichten

Wenn Sie Amazon OpenSearch Service verwenden, müssen Sie von Ihrer OpenSearch Service-Domain aus auf Ihre Amazon Personalize Personalize-Ressourcen zugreifen können.

So richten Sie Berechtigungen ein

1. Je nachdem, ob sich Ihre Ressourcen in denselben oder unterschiedlichen Konten befinden, erstellen Sie eine oder mehrere IAM-Servicerollen mit Zugriffsberechtigungen für Ihre Ressourcen.
 - Wenn sich Ihre OpenSearch Service- und Amazon Personalize Personalize-Ressourcen auf demselben Konto befinden, erstellen Sie eine IAM-Servicerolle für OpenSearch Service und erteilen ihr die Erlaubnis, ein personalisiertes Ranking aus Ihrer Amazon Personalize-Kampagne zu erhalten. Weitere Informationen finden Sie unter [Berechtigungen konfigurieren, wenn sich Ressourcen im selben Konto befinden](#).

- Wenn sich Ihre OpenSearch Service- und Amazon Personalize Personalize-Ressourcen in separaten Konten befinden, erstellen Sie zwei IAM-Servicerollen. Sie erstellen eine in dem Konto mit Ihren OpenSearch Servicere Ressourcen und gewähren ihm Zugriff auf Ihre OpenSearch Servicere Ressourcen. Und Sie erstellen eines in dem Konto mit Ihren Amazon Personalize Personalize-Ressourcen und erteilen ihm die Erlaubnis, ein personalisiertes Ranking aus Ihrer Amazon Personalize-Kampagne zu erhalten. Weitere Informationen finden Sie unter [Konfiguration von Berechtigungen, wenn sich Ressourcen in unterschiedlichen Konten befinden](#).
2. Erteilen Sie dem Benutzer oder der Rolle, der auf Ihre OpenSearch Service-Domain zugreift, PassRole Berechtigungen für die IAM-Servicerolle, die Sie für Service erstellt haben. OpenSearch Weitere Informationen finden Sie unter [Konfiguration der Amazon OpenSearch Service-Domain-Sicherheit](#).

Themen

- [Berechtigungen konfigurieren, wenn sich Ressourcen im selben Konto befinden](#)
- [Konfiguration von Berechtigungen, wenn sich Ressourcen in unterschiedlichen Konten befinden](#)
- [Konfiguration der Amazon OpenSearch Service-Domain-Sicherheit](#)

Berechtigungen konfigurieren, wenn sich Ressourcen im selben Konto befinden

Wenn sich Ihre OpenSearch Service- und Amazon Personalize Personalize-Ressourcen auf demselben Konto befinden, müssen Sie eine IAM-Servicerolle für OpenSearch Service erstellen. Diese Rolle muss über die Berechtigung verfügen, ein personalisiertes Ranking für Ihre Amazon Personalize-Kampagne zu erhalten. Folgendes ist erforderlich, um Ihrer OpenSearch Service-Rolle die Erlaubnis zu erteilen, ein personalisiertes Ranking aus Ihrer Amazon Personalize-Kampagne zu erhalten:

- Die Vertrauensrichtlinie der Rolle muss AssumeRole Berechtigungen für OpenSearch Service gewähren. Ein Beispiel für eine Vertrauensrichtlinie finden Sie unter [Beispiel für eine Vertrauensrichtlinie](#).
- Die Rolle muss über die Berechtigung verfügen, ein personalisiertes Ranking für Ihre Amazon Personalize-Kampagne zu erhalten. Eine Beispielrichtlinie finden Sie in [Beispiel für eine Berechtigungsrichtlinie](#).

Informationen zum Erstellen einer IAM-Rolle finden Sie unter [Erstellen von IAM-Rollen im IAM-Benutzerhandbuch](#). Informationen zum Anhängen einer IAM-Richtlinie an eine Rolle finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch](#).

Nachdem Sie eine IAM-Service-Rolle für OpenSearch Service erstellt haben, müssen Sie dem Benutzer oder der Rolle, die auf Ihre OpenSearch Service-Domain zugreift, `PassRole` Berechtigungen für die Service-Rolle gewähren. OpenSearch Weitere Informationen finden Sie unter [Konfiguration der Amazon OpenSearch Service-Domain-Sicherheit](#).

Themen

- [Beispiel für eine Vertrauensrichtlinie](#)
- [Beispiel für eine Berechtigungsrichtlinie](#)

Beispiel für eine Vertrauensrichtlinie

Das folgende Beispiel für eine Vertrauensrichtlinie gewährt `AssumeRole` Berechtigungen für den OpenSearch Dienst.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "",
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Principal": {
      "Service": [
        "es.amazonaws.com"
      ]
    }
  }]
}
```

Beispiel für eine Berechtigungsrichtlinie

Das folgende Richtlinienbeispiel gewährt der Rolle die Mindestberechtigungen, um ein personalisiertes Ranking für Ihre Amazon Personalize-Kampagne zu erhalten. Geben Sie für Campaign ARN den Amazon-Ressourcennamen (ARN) Ihrer Amazon Personalize-Kampagne an.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "personalize:GetPersonalizedRanking"
    ],
    "Resource": "Campaign ARN"
  }
]
```

Konfiguration von Berechtigungen, wenn sich Ressourcen in unterschiedlichen Konten befinden

Wenn sich Ihre OpenSearch Service- und Amazon Personalize Personalize-Ressourcen in separaten Konten befinden, erstellen Sie in jedem Konto eine IAM-Rolle und gewähren der Rolle Zugriff auf die Ressourcen im Konto.

Um Berechtigungen für mehrere Konten einzurichten

1. Erstellen Sie in dem Konto, in dem Ihre Amazon Personalize Personalize-Kampagne existiert, eine IAM-Rolle, die berechtigt ist, ein personalisiertes Ranking für Ihre Amazon Personalize-Kampagne zu erhalten. Wenn Sie das Plugin konfigurieren, geben Sie den ARN für diese Rolle im `external_account_iam_role_arn` Parameter des `personalized_search_ranking` Antwortprozessors an. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).

Eine Beispielrichtlinie finden Sie in [Beispiel für eine Berechtigungsrichtlinie](#).

2. Erstellen Sie in dem Konto, in dem Ihre OpenSearch Dienstdomäne existiert, eine Rolle mit einer Vertrauensrichtlinie, die OpenSearch AssumeRole Dienstberechtigungen gewährt. Wenn Sie das Plugin konfigurieren, geben Sie den ARN für diese Rolle im `iam_role_arn` Parameter des `personalized_search_ranking` Antwortprozessors an. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).

Ein Beispiel für eine Vertrauensrichtlinie finden Sie unter [Beispiel für eine Vertrauensrichtlinie](#).

3. Ändern Sie jede Rolle, um den anderen Rollen AssumeRole Berechtigungen zu gewähren. Für die Rolle, die Zugriff auf Ihre Amazon Personalize-Ressourcen hat, würde ihre IAM-Richtlinie der Rolle im Konto mit der OpenSearch Service-Domain beispielsweise Rollenberechtigungen wie folgt zuweisen:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "",
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::<Account number for role with access to
OpenSearch Service domain>:role/roleName"
  }]
}
```

4. Erteilen Sie in dem Konto, in dem Ihre OpenSearch Service-Domain existiert, dem Benutzer oder der Rolle, die auf Ihre OpenSearch Service-Domain zugreift, PassRole Berechtigungen für die OpenSearch Service-Service-Rolle, die Sie gerade erstellt haben. Weitere Informationen finden Sie unter [Konfiguration der Amazon OpenSearch Service-Domain-Sicherheit](#).

Konfiguration der Amazon OpenSearch Service-Domain-Sicherheit

Um das Plugin mit OpenSearch Service verwenden zu können, muss der Benutzer oder die Rolle, der auf Ihre Domain zugreift, über PassRole Berechtigungen für die [IAM-Dienstrolle für OpenSearch Service](#) verfügen, die Sie gerade erstellt haben. Außerdem muss der Benutzer oder die Rolle über die Berechtigung verfügen, die es :ESHttpPut Aktionen es :ESHttpGet und auszuführen.

Informationen zur Konfiguration des Zugriffs auf OpenSearch Service finden Sie unter [Sicherheit in Amazon OpenSearch Service](#) im Amazon OpenSearch Service Developer Guide. Richtlinienbeispiele finden Sie unter [Richtlinienbeispiele für OpenSearch Servicebenutzer oder -rollen](#).

Richtlinienbeispiele für OpenSearch Servicebenutzer oder -rollen

Das folgende Beispiel für eine IAM-Richtlinie gewährt einem Benutzer oder einer Rolle PassRole Berechtigungen für die IAM-Service-Rolle, die Sie für OpenSearch Service in erstellt haben. [Berechtigungen konfigurieren, wenn sich Ressourcen im selben Konto befinden](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
```

```

    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "OpenSearch Service role ARN"
  }
]
}

```

Die folgende IAM-Richtlinie gewährt die Mindestberechtigungen zum Erstellen von Pipelines und Suchabfragen mit Service. OpenSearch

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:ResourceTag/environment": [
            "production"
          ]
        }
      }
    }
  ]
}

```

Open-Source-Berechtigungen einrichten OpenSearch

Wenn Sie Open Source verwenden OpenSearch, müssen Sie von Ihrem Open-Search-Cluster aus auf Ihre Amazon Personalize Personalize-Ressourcen zugreifen können. Gehen Sie wie folgt vor, um Zugriff zu gewähren:

- Wenn Sie OpenSearch von Grund auf neu einrichten, können Sie ein [Schnellstart-Bash-Skript](#) verwenden, um einen OpenSearch Cluster in einem Docker-Container auszuführen. Das Skript verwendet die Standardanmeldedaten in Ihrem AWS Profil. Sie können ein alternatives Profil angeben, wenn Sie das Skript ausführen.

Diese Anmeldeinformationen müssen einem Benutzer oder einer Rolle zugeordnet sein, der bzw. die berechtigt ist, die `GetPersonalizedRanking` Aktion für Ihre Amazon Personalize durchzuführen. Ein Beispiel für eine IAM-Richtlinie finden Sie unter [Beispiele für IAM-Richtlinien](#). Alternativ müssen die Anmeldeinformationen berechtigt sein, eine Rolle anzunehmen, die über diese Berechtigungen verfügt. Sie können den Amazon-Ressourcennamen (ARN) für diese Rolle angeben, wenn Sie eine Pipeline für das Amazon Personalize Search Ranking-Plugin erstellen.

- Wenn Sie das [Schnellstart-Bash-Skript](#) nicht verwenden, können Sie Ihre Anmeldeinformationen manuell zu Ihrem OpenSearch Keystore hinzufügen. Diese Anmeldeinformationen müssen einem Benutzer oder einer Rolle entsprechen, der bzw. der berechtigt ist, die `GetPersonalizedRanking` Aktion für Ihre Amazon Personalize durchzuführen.

Um Ihre AWS Anmeldeinformationen manuell zu Ihrem OpenSearch Keystore hinzuzufügen, führen Sie den folgenden Befehl an der Stelle aus, an der Ihr OpenSearch Cluster ausgeführt wird (z. B. in einem Docker-Container). Geben Sie dann die einzelnen Anmeldeinformationen ein. Wenn Sie kein Sitzungstoken verwenden, können Sie die letzte Zeile im Befehl weglassen.

```
opensearch-keystore add \  
personalized_search_ranking.aws.access_key \  
personalized_search_ranking.aws.secret_key \  
personalized_search_ranking.aws.session_token
```

- Wenn Sie Ihren OpenSearch Cluster auf einer Amazon EC2 EC2-Instance ausführen, können Sie Berechtigungen mit einem IAM-Instance-Profil gewähren. Die der Rolle beigefügte Richtlinie muss ihr die Erlaubnis erteilen, die `GetPersonalizedRanking` Aktion für Ihre Amazon Personalize durchzuführen. Es muss auch Amazon EC2 EC2-Berechtigungen erteilen, um die Rolle übernehmen zu können.

Informationen zu Amazon EC2 EC2-Instance-Profilen finden Sie unter [Instanzprofile verwenden](#). Eine Beispielrichtlinie finden Sie in [Beispiele für IAM-Richtlinien](#).

Beispiele für IAM-Richtlinien

Das folgende Richtlinienbeispiel gewährt einem Benutzer oder einer Rolle die Mindestberechtigungen, um aus Ihrer Amazon Personalize-Kampagne ein personalisiertes Ranking zu erhalten. Geben Sie für Campaign ARN den Amazon-Ressourcennamen (ARN) Ihrer Amazon Personalize-Kampagne an.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:GetPersonalizedRanking"
      ],
      "Resource": "Campaign ARN"
    }
  ]
}
```

Wenn Sie Ihren OpenSearch Cluster auf einer Amazon EC2 EC2-Instance ausführen und Berechtigungen mit einem IAM-Instance-Profil gewähren, muss die Vertrauensrichtlinie für die Rolle außerdem Amazon EC2 AssumeRole EC2-Berechtigungen wie folgt gewähren. Informationen zu Amazon EC2 EC2-Instance-Profilen finden Sie unter [Instanzprofile verwenden](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Einrichtung OpenSearch und Installation des Plug-ins

Das Amazon Personalize Search Ranking-Plugin verarbeitet die Kommunikation mit Amazon Personalize von Ihrer OpenSearch Service-Domain oder Ihrem OpenSearch Open-Source-Cluster aus. Es kümmert sich auch um die Neurangierung von Ergebnissen. Je nachdem, wie Sie darauf zugreifen OpenSearch, richten Sie das Plugin wie folgt ein OpenSearch und installieren es:

- Wenn Sie Amazon OpenSearch Service verwenden, müssen Sie bei der OpenSearch Einrichtung eine Domain in OpenSearch Service erstellen, Daten aufnehmen und das Plugin installieren.

- Wenn Sie Open Source verwenden OpenSearch, erstellen Sie einen OpenSearch Cluster, nehmen Daten auf und installieren das Plugin.

Themen

- [Amazon OpenSearch Service einrichten](#)
- [Open Source einrichten OpenSearch](#)

Amazon OpenSearch Service einrichten

Nachdem Sie den Amazon Personalize Personalize-Workflow abgeschlossen und die unter aufgeführten Anforderungen erfüllt haben [Richtlinien und Anforderungen](#), können Sie Amazon OpenSearch Service einrichten und das Amazon Personalize Search Ranking-Plugin installieren.

Um Amazon OpenSearch Service einzurichten, erstellen Sie eine Domain, nehmen Ihre Daten auf und installieren das Plugin. Wenn Sie bereits eine Domain erstellt und Ihre Daten aufgenommen haben, können Sie mit Schritt 3 fortfahren.

Um den Service einzurichten OpenSearch

1. Falls Sie dies noch nicht getan haben, führen Sie die Schritte unter aus, [Amazon OpenSearch Service-Berechtigungen einrichten](#) damit Sie von Ihrer OpenSearch Service-Domain aus auf Ihre Amazon Personalize Personalize-Ressourcen zugreifen können.
2. Falls Sie dies noch nicht getan haben, erstellen Sie eine OpenSearch Service-Domain. Eine OpenSearch Service-Domain ist gleichbedeutend mit einem OpenSearch Open-Source-Cluster. Domains sind Cluster mit den Einstellungen, Instance-Typen, Instance-Anzahl und Speicherressourcen, die Sie angeben.
 - Eine kurze Anleitung zur Konfiguration einer Testdomain finden Sie unter [Schritt 1: Amazon OpenSearch Service-Domain erstellen](#) im Abschnitt „Erste Schritte“ des Amazon OpenSearch Service Developer Guide.
 - Ausführlichere Schritte finden Sie unter [Amazon OpenSearch Service-Domains erstellen und verwalten](#).
3. Falls Sie es noch nicht getan haben, nehmen Sie Ihre Artikel in OpenSearch Service auf.
 - Eine kurze Anleitung zum Hochladen einer kleinen Menge von Testdaten in OpenSearch Service finden Sie unter [Schritt 2: Daten zur Indexierung auf Amazon OpenSearch Service hochladen](#) im Abschnitt „Erste Schritte“ des Amazon OpenSearch Service Developer Guide.

- Vollständige Informationen zur Datenaufnahme finden Sie unter [Indexierung von Daten in Amazon OpenSearch Service](#) im Amazon OpenSearch Service Developer Guide.
4. Verknüpfen Sie das `Amazon_Personalize_Search_Ranking_Plugin` Plugin mit Ihrer Domain. Das Plugin ist vorinstalliert und Sie müssen es nicht aus Amazon S3 importieren. Sie verknüpfen das Plugin auf die gleiche Weise wie ein OpenSearch Servicepaket.

Informationen zum Zuordnen eines OpenSearch Servicepakets finden Sie unter [Benutzerdefinierte Pakete für Amazon OpenSearch Service](#).

Nachdem Sie eine Domain erstellt, Daten aufgenommen und das Amazon Personalize Search Ranking-Plugin installiert haben, können Sie das Plugin konfigurieren. Sie konfigurieren es, indem Sie eine Suchpipeline erstellen und einen `personalized_search_ranking` Antwortprozessor angeben. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).

Open Source einrichten OpenSearch

Nachdem Sie den Amazon Personalize Personalize-Workflow abgeschlossen und die unter aufgeführten Anforderungen erfüllt haben [Richtlinien und Anforderungen](#), sind Sie bereit, Open Source einzurichten und das Amazon Personalize Search Ranking-Plugin zu installieren.

Wenn bei Ihnen bereits ein OpenSearch Cluster läuft, können Sie das Plugin manuell installieren. Wenn kein Cluster läuft, können Sie das Plugin mit einem Bash-Skript von Grund auf neu installieren OpenSearch .

Themen

- [Manuelles Installieren des Plugins auf einem vorhandenen Cluster OpenSearch](#)
- [Richten Sie Ihren Cluster ein und installieren Sie das Plugin mit einem Schnellstart-Skript](#)

Manuelles Installieren des Plugins auf einem vorhandenen Cluster OpenSearch

Wenn Sie bereits einen OpenSearch Cluster haben, können Sie das Plugin manuell direkt aus dem OpenSearch GitHub Repository auf Ihrem Cluster installieren.

Um das Plugin manuell zu installieren

1. Verwenden Sie den folgenden Befehl, um Ihren OpenSearch Cluster zu starten:

```
bin/opensearch
```

2. Laden Sie Ihre Katalogdaten in Ihren OpenSearch Cluster hoch, falls Sie dies noch nicht getan haben. Wenn Sie Ihre Daten hochladen, erstellen Sie einen OpenSearch Index und definieren Ihre Feldzuordnungen. Anschließend laden Sie Ihre Daten in diesen Index hoch. Ein Beispiel finden Sie unter [Erstellen eines Indexes und von Feldzuordnungen mithilfe von Beispieldaten](#).
3. Verwenden Sie den folgenden Befehl, um das Plugin zu installieren:

```
bin/opensearch-plugin install https://github.com/opensearch-project/search-processor/releases/download/2.9.0/opensearch-search-processor-2.9.0.0.zip
```

Weitere Informationen zur Installation von Plugins finden Sie unter [Plugins installieren](#).

Nachdem Sie das Amazon Personalize Search Ranking-Plugin installiert haben, können Sie es konfigurieren. Sie konfigurieren das Plugin, indem Sie eine Suchpipeline erstellen und einen `personalized_search_ranking` Antwortprozessor angeben. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).

Richten Sie Ihren Cluster ein und installieren Sie das Plugin mit einem Schnellstart-Skript

Wenn Sie noch keinen OpenSearch Cluster erstellt haben, können Sie ein Schnellstart-Bash-Skript verwenden, um einen zu erstellen. Dieses Skript richtet einen OpenSearch Cluster in einem Docker-Container ein, richtet Anmeldeinformationen mithilfe Ihres AWS Standardprofils ein und installiert das Amazon Personalize Search Ranking-Plugin.

Informationen zur manuellen Erstellung eines OpenSearch Clusters finden Sie in den [Schnellstartanweisungen](#) in der Dokumentation. OpenSearch

Um das Plugin mit einem Schnellstart-Bash-Skript zu installieren

1. Bevor Sie das Skript ausführen, laden Sie [Docker Desktop](#) für Ihr Betriebssystem herunter und installieren Sie es.
2. Laden Sie das [Schnellstart-Bash-Skript](#) von herunter. GitHub
3. Führen Sie das Skript in Ihrem Arbeitsverzeichnis mit dem folgenden Befehl aus.

```
sh personalized_search_ranking_quickstart.sh
```

Mit diesem Befehl verwendet das Skript die Anmeldeinformationen in Ihrem AWS Standardprofil. Verwenden Sie das `--profile` Argument, um ein alternatives Profil bereitzustellen.

```
sh personalized_search_ranking_quickstart.sh --profile profile-name
```

Nachdem Sie das Skript ausgeführt haben, finden Sie weitere Informationen zum Skript in der README-Datei, die sich in dem eindeutigen Verzeichnis befindet, das durch das Skript erstellt wurde. In diesem Verzeichnis werden die Dateien Dockerfile und docker-compose.yml gespeichert, die das Skript verwendet. Zum Beispiel: `../opensearch-personalize-intelligent-ranking-docker.1234/README`.

4. Laden Sie Ihre Katalogdaten in Ihren Cluster hoch. OpenSearch Wenn Sie Ihre Daten hochladen, erstellen Sie einen OpenSearch Index und definieren Ihre Feldzuordnungen. Anschließend laden Sie Ihre Daten in diesen Index hoch. Ein Beispiel finden Sie unter [Erstellen eines Indexes und von Feldzuordnungen mithilfe von Beispieldaten](#).

Nachdem Sie das Amazon Personalize Search Ranking-Plugin eingerichtet OpenSearch und installiert haben, können Sie es konfigurieren. Sie konfigurieren das Plugin, indem Sie eine Suchpipeline erstellen und einen `personalized_search_ranking` Antwortprozessor angeben. Weitere Informationen finden Sie unter [Konfiguration des Plug-ins](#).

Konfiguration des Plug-ins

Nachdem Sie das Amazon Personalize Search Ranking-Plugin installiert haben, können Sie es konfigurieren, indem Sie eine OpenSearch Suchpipeline erstellen.

Eine Suchpipeline besteht aus einer Reihe von Anforderungs- und Antwortprozessoren, die nacheinander in der Reihenfolge ausgeführt werden, in der Sie sie erstellen. Wenn Sie eine Suchpipeline für das Plugin erstellen, geben Sie einen `personalized_search_ranking` Antwortprozessor an. Informationen zu Such-Pipelines finden Sie unter [Such-Pipelines](#).

Themen

- [Felder für den Antwortprozessor `personalized_search_ranking`](#)
- [Eine Pipeline mit Amazon OpenSearch Service erstellen](#)
- [Eine Pipeline mit Open Source erstellen OpenSearch](#)

Felder für den Antwortprozessor **personalized_search_ranking**

Für den `personalized_search_ranking` Antwortprozessor geben Sie die folgenden Felder an:

- `campaign_arn` (erforderlich) — Geben Sie den Amazon-Ressourcennamen (ARN) der Amazon Personalize-Kampagne an, die zur Personalisierung der Ergebnisse verwendet werden soll.
- `item_id_field` (optional) — Wenn das `_id` Feld für ein indexiertes Dokument in OpenSearch nicht Ihren Amazon Personalize Personalize-Artikel-IDs entspricht, geben Sie den Namen des Felds an, das dies tut. Standardmäßig geht das Plugin davon aus, dass die `_id` Daten mit der `itemId` in Ihren Amazon Personalize übereinstimmen.
- `Rezept` (erforderlich) — Geben Sie den Namen des Amazon Personalize-Rezepts an, das verwendet werden soll. Sie können nur `aws-personalized-ranking` angeben.
- `Gewicht` (erforderlich) — Geben Sie an, welchen Schwerpunkt der Antwortprozessor bei der Neueinteilung der Ergebnisse auf Personalisierung legt. Geben Sie einen Wert innerhalb eines Bereichs von 0,0-1,0 an. Je näher es 1.0 daran liegt, desto wahrscheinlicher ist es, dass Ergebnisse von Amazon Personalize einen höheren Rang einnehmen. Wenn Sie angeben 0.0, findet keine Personalisierung statt und OpenSearch hat Vorrang.
- `tag` (optional) — Geben Sie eine Kennung für den Prozessor an.
- `iam_role_arn` (erforderlich für OpenSearch Service, optional für Open Source OpenSearch) — Geben Sie für OpenSearch Service den Amazon-Ressourcennamen (ARN) für die Rolle an, die Sie beim [Einrichten von Berechtigungen](#) für OpenSearch Service für den Zugriff auf Ihre Amazon Personalize-Ressourcen erstellt haben. Wenn Ihre OpenSearch Service- und Amazon Personalize Personalize-Ressourcen in unterschiedlichen Konten vorhanden sind, geben Sie die Rolle an, die `AssumeRole` Berechtigungen für OpenSearch Service gewährt. Weitere Informationen finden Sie unter [Konfiguration von Berechtigungen, wenn sich Ressourcen in unterschiedlichen Konten befinden](#).

Wenn Sie bei Open Source mehrere Rollen verwenden OpenSearch, um die Berechtigungen für verschiedene Benutzergruppen in Ihrer Organisation einzuschränken, geben Sie den ARN der Rolle an, die berechtigt ist, auf Amazon Personalize zuzugreifen. Wenn Sie nur die AWS Anmeldeinformationen in Ihrem OpenSearch Keystore verwenden, können Sie dieses Feld weglassen.

- `aws_region` (erforderlich) — Die AWS Region, in der Sie Ihre Amazon Personalize-Kampagne erstellt haben.
- `ignore_failure` (optional) — Geben Sie an, ob das Plugin alle Prozessorausfälle ignoriert. Geben Sie für Werte oder an. `true false` Für Ihre Produktionsumgebungen empfehlen wir die Angabe, `true`

um Unterbrechungen bei der Beantwortung von Abfragen zu vermeiden. Für Testumgebungen können Sie angeben `false`, dass alle vom Plugin generierten Fehler angezeigt werden sollen.

- `external_account_iam_role_arn` — Wenn Sie OpenSearch Service verwenden und Ihre Amazon Personalize- und Service-Ressourcen in verschiedenen Konten vorhanden sind, geben Sie den ARN der Rolle an, die berechtigt ist, auf Ihre Amazon Personalize Personalize-Ressourcen zuzugreifen. OpenSearch Diese Rolle muss in demselben Konto wie Ihre Amazon Personalize existieren. Weitere Informationen finden Sie unter [Konfiguration von Berechtigungen, wenn sich Ressourcen in unterschiedlichen Konten befinden](#).

Eine Pipeline mit Amazon OpenSearch Service erstellen

Sie können den folgenden Python-Code verwenden, um eine Suchpipeline mit einem `personalized_search_ranking` Antwortprozessor in einer OpenSearch Service-Domain zu erstellen. Ersetzen Sie es `domain_endpoint` durch die URL Ihres Domain-Endpunkts. Zum Beispiel: `https://<domain name>.<AWS region>.es-staging.amazonaws.com`.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
pipeline_name = 'pipeline name'
url = f'{domain_endpoint}/_search/pipeline/{pipeline_name}'
auth = AWSSigV4('es')

headers = {'Content-Type': 'application/json'}

body = {
    "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
    "response_processors": [
        {
            "personalized_search_ranking" : {
                "campaign_arn" : "Amazon Personalize Campaign ARN",
                "item_id_field" : "productId",
                "recipe" : "aws-personalized-ranking",
                "weight" : "0.3",
                "tag" : "personalize-processor",
                "iam_role_arn": "Role ARN",
                "aws_region": "AWS region",
                "ignore_failure": true
            }
        }
    ]
}
```

```

]
}
try:
    response = requests.put(url, auth=auth, json=body, headers=headers, verify=False)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")

```

Nachdem Sie eine Suchpipeline mit einem `personalized_search_ranking` Antwortprozessor erstellt haben, können Sie damit beginnen, das Plugin auf OpenSearch Abfragen anzuwenden. Sie können es auf einen OpenSearch Index oder eine einzelne OpenSearch Abfrage anwenden. Weitere Informationen finden Sie unter [Anwenden des Plugins auf OpenSearch Abfragen](#).

Eine Pipeline mit Open Source erstellen OpenSearch

Sie können den folgenden curl-Befehl verwenden, um eine Suchpipeline mit einem `personalized_search_ranking` Antwortprozessor auf einem OpenSearch Open-Source-Cluster zu erstellen.

```

curl -X PUT "http://localhost:9200/_search/pipeline/pipeline-name" -ku 'admin:admin' --
insecure -H 'Content-Type: application/json' -d'
{
  "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
  "response_processors" : [
    {
      "personalized_search_ranking" : {
        "campaign_arn" : "Amazon Personalize Campaign ARN",
        "item_id_field" : "productId",
        "recipe" : "aws-personalized-ranking",
        "weight" : "0.3",
        "tag" : "personalize-processor",
        "iam_role_arn": "Role ARN",
        "aws_region": "AWS region",
        "ignore_failure": true
      }
    }
  ]
}'

```

Nachdem Sie eine Suchpipeline mit einem `personalized_search_ranking` Antwortprozessor erstellt haben, können Sie damit beginnen, das Plugin auf OpenSearch Abfragen anzuwenden. Sie

können es auf einen OpenSearch Index oder eine einzelne OpenSearch Abfrage anwenden. Weitere Informationen finden Sie unter [Anwenden des Plugins auf OpenSearch Abfragen](#).

Anwenden des Plugins auf OpenSearch Abfragen

Nachdem Sie eine Suchpipeline mit einem `personalized_search_ranking` Antwortprozessor konfiguriert haben, können Sie das Amazon Personalize Search Ranking-Plugin auf Ihre OpenSearch Anfragen anwenden und die neu eingestufteten Ergebnisse anzeigen.

Wenn Sie das Plugin auf OpenSearch Anfragen anwenden, können Sie das Plugin überwachen, indem Sie Metriken für Ihre Suchpipeline abrufen. Weitere Informationen finden Sie unter [Überwachung des Plugins](#).

Themen

- [Anwenden des Plug-ins auf Amazon OpenSearch Service-Abfragen](#)
- [Anwenden des Plug-ins auf Abfragen in Open Source OpenSearch](#)

Anwenden des Plug-ins auf Amazon OpenSearch Service-Abfragen

Sie können das Amazon Personalize Search Ranking-Plugin auf alle Abfragen und Antworten für einen Index anwenden. Sie können das Plugin auch auf einzelne Anfragen und Antworten anwenden.

- Sie können den folgenden Python-Code verwenden, um eine Suchpipeline auf einen Index anzuwenden. Bei diesem Ansatz verwenden alle Suchanfragen, die diesen Index verwenden, das Plugin, um die Suchergebnisse zu personalisieren.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
index = 'index name'
url = f'{domain_endpoint}/{index}/_settings/'
auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
body = {
    "index.search.default_pipeline": "pipeline name"
}
try:
    response = requests.put(url, auth=auth, json=body, headers=headers)
```

```
print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

- Sie können den folgenden Python-Code verwenden, um eine Suchpipeline auf eine einzelne Abfrage für Fahrzeuge der Marke Toyota anzuwenden.

Aktualisieren Sie den Code, um Ihren Domain-Endpunkt, Ihren OpenSearch Serviceindex, den Namen Ihrer Pipeline und Ihre Abfrage anzugeben. Geben Sie für `user_id` die ID des Benutzers an, für den Sie Suchergebnisse erhalten. Dieser Benutzer muss in den Daten enthalten sein, mit denen Sie Ihre Amazon Personalize-Lösungsversion erstellt haben. Wenn der Benutzer nicht anwesend war, ordnet Amazon Personalize die Artikel nach ihrer Beliebtheit.

Denn `context` wenn Sie kontextuelle Metadaten verwenden, geben Sie die kontextuellen Metadaten des Benutzers an, z. B. seinen Gerätetyp. Das Feld `context` ist optional. Weitere Informationen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#).

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
index = 'index name'
url = f'{domain_endpoint}/{index}/_search/'

auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
params = {"search_pipeline": "pipeline-name"}
body = {
    "query": {
        "multi_match": {
            "query": "Toyota",
            "fields": ["BRAND"]
        }
    },
    "ext": {
        "personalize_request_parameters": {
            "user_id": "USER ID"
            "context": { "DEVICE" : "mobile phone" }
        }
    }
}
```

```
try:
    response = requests.post(url, auth=auth, params=params, json=body,
                              headers=headers, verify=False)
    print(response)
except Exception as e:
    print(f"Error: {e}")
```

Anwenden des Plug-ins auf Abfragen in Open Source OpenSearch

Sie können das Amazon Personalize Search Ranking-Plugin auf alle Abfragen und Antworten für einen OpenSearch Index anwenden. Sie können das Plugin auch auf einzelne OpenSearch Anfragen und Antworten anwenden.

- Der folgende curl-Befehl wendet eine Suchpipeline auf einen OpenSearch Index in einem lokal laufenden OpenSearch Open-Source-Cluster an. Bei diesem Ansatz verwenden alle Suchanfragen in diesem Index das Plugin, um die Suchergebnisse zu personalisieren.

```
curl -XGET "https://localhost:9200/index/_settings" -ku 'admin:admin' --insecure -H
'Content-Type: application/json' -d'
{
  "index.search.default_pipeline" : "pipeline-name"
}
```

- Der folgende curl-Befehl wendet eine Suchpipeline auf eine einzelne Abfrage nach Fahrzeugen der Marke Toyota in einem Index in einem lokal ausgeführten OpenSearch Open-Source-Cluster an.

Geben Sie für `user_id` die ID des Benutzers an, für den Sie Suchergebnisse erhalten. Dieser Benutzer muss in den Daten enthalten sein, mit denen Sie Ihre Amazon Personalize-Lösungsversion erstellt haben. Wenn der Benutzer nicht anwesend war, ordnet Amazon Personalize die Artikel nach ihrer Beliebtheit. Denn `context` wenn Sie kontextuelle Metadaten verwenden, geben Sie die kontextuellen Metadaten des Benutzers an, z. B. seinen Gerätetyp. Das Feld `context` ist optional. Weitere Informationen finden Sie unter [Erhöhung der Relevanz von Empfehlungen mit kontextbezogenen Metadaten](#).

```
curl -XGET "http://localhost:9200/index/_search?search_pipeline=pipeline-name" -ku
'admin:admin' --insecure -H 'Content-Type: application/json' -d'
{
  "query": {
    "multi_match": {
```

```
    "query": "Toyota",
    "fields": ["BRAND"]
  }
},
"ext": {
  "personalize_request_parameters": {
    "user_id": "user ID",
    "context": { "DEVICE" : "mobile phone" }
  }
}
}
```

Um zu verstehen, wie Ergebnisse neu eingestuft werden, können Sie mithilfe von OpenSearch Dashboards Ergebnisse mit OpenSearch Ergebnissen vergleichen, die mit dem Plugin neu eingestuft wurden. Weitere Informationen finden Sie unter [OpenSearch Ergebnisse mit Ergebnissen aus dem Plugin vergleichen](#).

Wenn Sie das Plugin auf OpenSearch Abfragen anwenden, können Sie das Plugin überwachen, indem Sie Metriken für Ihre Pipeline abrufen. OpenSearch Weitere Informationen finden Sie unter [Überwachung des Plugins](#).

OpenSearch Ergebnisse mit Ergebnissen aus dem Plugin vergleichen

Das Amazon Personalize Search Ranking-Plugin ordnet die Suchergebnisse sowohl auf der Grundlage des Rankings von Amazon Personalize als auch des Rankings von neu an. OpenSearch Wie das Plugin die Ergebnisse neu einstuft, hängt davon ab, wie Sie den `personalized_search_ranking` Antwortprozessor in Ihren Pipelines konfiguriert haben.

Um zu verstehen, wie Ergebnisse neu eingestuft werden, können Sie Abfragen mit und ohne Personalisierung ausführen und die Ergebnisse vergleichen.

Themen

- [Ergebnisse mit Amazon OpenSearch Service vergleichen](#)
- [Vergleich der Ergebnisse mit Open Source OpenSearch](#)

Ergebnisse mit Amazon OpenSearch Service vergleichen

Um zu verstehen, wie Ergebnisse eingestuft werden, können Sie Abfragen mit und ohne Personalisierung ausführen und die Ergebnisse vergleichen. Sie können den folgenden Python-Code verwenden, um zwei verschiedene Abfragen auszuführen und die Ergebnisse in zwei JSON-Dateien auszugeben. Die erste Methode führt eine Abfrage aus, die das Plugin verwendet, um die Ergebnisse neu zu ordnen. Die zweite führt eine Methode aus, die Ergebnisse ohne Personalisierung generiert.

```
import json
import requests
from requests_auth_aws_sigv4 import AWSSigV4

# Returns re-ranked OpenSearch results using the Amazon Personalize Search Ranking
# plugin.
def get_personalized_results(pipeline_name):
    url = f'{domain}/{index}/_search/'
    auth = AWSSigV4('es')
    headers = {'Content-Type': 'application/json'}
    params = {"search_pipeline": pipeline_name}
    body = {
        "query": {
            "multi_match": {
                "query": "Toyota",
                "fields": ["BRAND"]
            }
        },
        "ext": {
            "personalize_request_parameters": {
                "user_id": "1"
            }
        }
    }
    try:
        response = requests.post(url, auth=auth, params=params, json=body,
headers=headers, verify=False)
    except Exception as e:
        return f"Error: {e}"
    return response.text

# Returns OpenSearch results without personalization.
def get_opensearch_results():
```

```
url = f'{domain}/{index}/_search/'
auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
body = {
    "query": {
        "multi_match": {
            "query": "Toyota",
            "fields": ["BRAND"]
        }
    }
}
try:
    response = requests.post(url, auth=auth, json=body, headers=headers,
verify=False)
except Exception as e:
    return f"Error: {e}"
return response.text

def print_results(file_name, results):
    results_file = open(file_name, 'w')
    results_file.write(json.dumps(results, indent=4))
    results_file.close()

# specify domain endpoint
domain = "DOMAIN_ENDPOINT"

# specify the region where you created your Amazon Personalize resources and Amazon
OpenSearch domain
aws_region = "REGION"

# specify the name of the pipeline that uses the Amazon Personalize plugin
pipeline_name = "PIPELINE_NAME"

# specify your Amazon OpenSearch index
index = "INDEX"

# specify names for json files for comparison
personalized_results_file = "personalized_results.json"
opensearch_results_file = "opensearch_results.json"

# get personalized results
personalized_results = json.loads(get_personalized_results(pipeline_name))
```

```
# get OpenSearch results without personalization
opensearch_results = json.loads(get_opensearch_results())

# print results to files
print_results(personalized_results_file, personalized_results)
print_results(opensearch_results_file, opensearch_results)
```

Vergleich der Ergebnisse mit Open Source OpenSearch

Um zu verstehen, wie Ergebnisse neu eingestuft werden, können Sie Abfragen mit der [Dev Tools-Konsole](#) in zwei separaten Browserfenstern ausführen. Anschließend können Sie die Ergebnisse für Abfragen mit und ohne Personalisierung vergleichen.

Um Ergebnisse mit der Dev Tools-Konsole zu vergleichen

1. Falls Sie es noch nicht getan haben, folgen Sie den Schritten unter [Einrichtung OpenSearch und Installation des Plug-ins](#) und [Konfiguration des Plug-ins](#).
2. Stellen Sie sicher, dass OpenSearch Dashboards installiert ist. Das Schnellstart-Bash-Skript installiert Dashboards. OpenSearch Wenn Sie das Skript nicht verwenden oder bereits ein Cluster läuft, müssen Sie Dashboards installieren. OpenSearch Weitere Informationen finden Sie unter [OpenSearch Dashboards installieren](#).
3. Starten Sie OpenSearch Dashboards. Öffnen Sie `http://localhost:5601` es in einem Browser und melden Sie sich bei OpenSearch Dashboards an. Die Standardanmeldedaten sind der Benutzername „admin“ und das Passwort „admin“.
4. Wählen Sie auf der Startseite der OpenSearch Dashboards im Verwaltungsmenü die Option Dev Tools aus.
5. Öffnen Sie ein separates Browserfenster und öffnen Sie die Dev Tools-Konsole erneut. Sie können die URL aus dem vorherigen Fenster verwenden.
6. Geben Sie in einem Fenster eine Abfrage ein, die keine Neurangierung für die Personalisierung verwendet. Geben Sie im anderen Fenster einen curl-Befehl ein, der eine Pipeline mit dem `personalized_search_ranking` Antwortprozessor verwendet. Wenn Sie einen curl-Befehl direkt in die Konsole einfügen, wird der Befehl automatisch in das von der Konsole verwendete Format konvertiert. Ein Befehlsbeispiel finden Sie unter [Anwenden des Plugins auf OpenSearch Abfragen](#).
7. Führen Sie beide Abfragen aus und vergleichen Sie die Ergebnisse.

Überwachung des Plugins

Wenn Sie OpenSearch Service verwenden, können Sie das Plugin anhand von Metriken in Amazon überwatchen CloudWatch. Weitere Informationen finden Sie unter [Amazon OpenSearch Service-Domains überwachen](#).

Wenn Sie das Amazon Personalize Search Ranking-Plugin auf OpenSearch Abfragen anwenden, können Sie das Plugin überwachen, indem Sie Metriken für Ihre Such-Pipelines abrufen. Zu den Pipeline-Metriken gehören Statistiken wie die Anzahl der fehlgeschlagenen Anfragen an den `personalized_search_ranking` Antwortprozessor.

Themen

- [Überwachung des Plugins mit Amazon OpenSearch Service](#)
- [Überwachung des Plugins mit Open Source OpenSearch](#)
- [Beispiel für Pipeline-Kennzahlen](#)

Überwachung des Plugins mit Amazon OpenSearch Service

Sie können den folgenden Python-Code verwenden, um Metriken für alle Ihre Pipelines abzurufen. Ein Beispiel für Pipeline-Metriken finden Sie unter [Beispiel für Pipeline-Kennzahlen](#).

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
url = f'{domain_endpoint}/_nodes/stats/search_pipeline'

auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
try:
    response = requests.get(url, auth=auth, headers=headers, verify=False)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```


Überwachung des Plugins mit Open Source OpenSearch

Sie können den folgenden Code verwenden, um Metriken für alle Ihre Pipelines abzurufen. Die Antwort enthält Statistiken für alle Such-Pipelines. Ein Beispiel für Pipeline-Metriken finden Sie unter [Beispiel für Pipeline-Kennzahlen](#).

```
curl -XGET "https://localhost:9200/_nodes/stats/search_pipeline?pretty" -ku  
'admin:admin'
```

Beispiel für Pipeline-Kennzahlen

Der folgende Code zeigt einen Auszug der Pipeline-Metriken, die von OpenSearch zurückgegeben werden. Es wird nur das `pipelines` Objekt angezeigt, das Statistiken für zwei verschiedene Pipelines enthält. Für jede Pipeline finden Sie die Kennzahlen des Amazon Personalize Search Ranking-Plug-ins in der Liste der `personalized_search_ranking` Antwortprozessoren. Ein vollständiges Beispiel für alle Metriken finden Sie unter [Metriken zur Suchpipeline](#).

```
{  
  ....  
  ....  
  "pipelines": {  
    "pipelineA": {  
      "request": {  
        "count": 0,  
        "time_in_millis": 0,  
        "current": 0,  
        "failed": 0  
      },  
      "response": {  
        "count": 6,  
        "time_in_millis": 2246,  
        "current": 0,  
        "failed": 0  
      },  
      "request_processors": [],  
      "response_processors": [  
        {  
          "personalized_search_ranking": {  
            "type": "personalized_search_ranking",  
            "stats": {  
              "count": <number of requests>,  

```

```
        "time_in_millis": <time>,
        "current": 0,
        "failed": <number of failed requests>
    }
}
],
},
"pipelineB": {
  "request": {
    "count": 0,
    "time_in_millis": 0,
    "current": 0,
    "failed": 0
  },
  "response": {
    "count": 8,
    "time_in_millis": 2248,
    "current": 0,
    "failed": 0
  },
  "request_processors": [],
  "response_processors": [
    {
      "personalized_search_ranking": {
        "type": "personalized_search_ranking",
        "stats": {
          "count": <number of requests>,
          "time_in_millis": <time>,
          "current": 0,
          "failed": <number of failed requests>
        }
      }
    }
  ]
}
}
....
....
}
```

Markieren von Amazon-Personalize-Ressourcen

Ein Tag ist eine Bezeichnung, die Sie optional definieren und mit - AWS Ressourcen verknüpfen, einschließlich bestimmter Arten von Amazon-Personalize-Ressourcen. Eine Ressource kann bis zu 50 Tags enthalten.

Tags können Ihnen helfen, Ressourcen auf unterschiedliche Weise zu kategorisieren und zu verwalten, z. B. nach Zweck, Umgebung oder anderen Kriterien. Sie können beispielsweise Tags verwenden, um den Umsatz auf verschiedene Funktionen aufzuteilen oder Entwicklungsumgebungen für verschiedene Ressourcen zu identifizieren.

Um Amazon-Personalizes-Ressourcen nach Tag abzurufen, können Sie die Filter im `GetResources` Betrieb der `Resource-Groups-Tagging-API` verwenden. Weitere Informationen finden Sie unter [GetResources](#) im API-Referenzhandbuch für `Resource Groups Tagging`.

Sie können Tags zu den folgenden Arten von Amazon-Personalize-Ressourcen hinzufügen:

- Batch-Inferenzaufträge
- Batch-Segment-Aufträge
- Kampagnen
- Datensätze
- Datensatzgruppen
- Import- und Exportaufträge für Datensätze
- Ereignis-Tracker
- Filter
- Empfehlungsgeber
- Lösungen
- Lösungsversionen

Themen

- [Richtlinien und Anforderungen](#)
- [Hinzufügen von Tags zu Amazon-Personalize-Ressourcen](#)
- [Entfernen von Tags aus Amazon-Personalize-Ressourcen](#)
- [Verwenden von Tags in IAM-Richtlinien](#)

Richtlinien und Anforderungen

Jedes Tag besteht aus einem Schlüssel und einem optionalen Wert. Beides können Sie definieren. Ein Tag-Schlüssel ist eine allgemeine Markierung, die wie eine Kategorie für spezifischere Tag-Werte fungiert. Ein Tag-Wert dient als Bezeichnung für einen Tag-Schlüssel.

Wenn Sie beispielsweise zwei Versionen einer Amazon-Personalize-Datensatzgruppe haben (eine für interne Tests und eine für die Produktion), können Sie beiden Projekten einen `-EnvironmentTag`-Schlüssel zuweisen. Der Tag-Wert des `Environment` Tags kann `Test` für eine Version der Datensatzgruppe und `Production` für die andere Version sein.

Für Tags gelten die folgenden Einschränkungen:

- Maximale Anzahl von Tags (Markierungen) pro Ressource: 50
- Maximale Schlüssellänge: 128 Unicode-Zeichen in UTF-8
- Maximale Wertlänge: 256 Unicode-Zeichen in UTF-8
- Tag-Schlüssel und -Werte können die folgenden Zeichen enthalten: A-Z, a-z, 0-9, Leerzeichen und `_ . : / = + @ -` (Bindestrich). Dies ist der Standardsatz von Zeichen, die für AWS alle Services verfügbar sind, die Tags unterstützen. Einige -Services unterstützen zusätzliche Symbole.
- Bei Tag-Schlüsseln und Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Für jede zugeordnete Ressource muss jeder Tag-Schlüssel eindeutig sein und kann nur einen Tag-Wert haben.
- Ihre Tag-Schlüssel und Tag-Werte können nicht mit `beginnenaws :` - AWS Services wenden Tags an, die mit `beginnenaws :`, und diese Tags können nicht geändert werden. Sie werden nicht auf Tag-Limits angerechnet.
- Es ist nicht möglich, eine Ressource nur anhand ihrer Tags zu aktualisieren oder zu löschen. Darüber hinaus müssen Sie den Amazon-Ressourcennamen (ARN) oder die Ressourcen-ID angeben – je nach Operation, die Sie verwenden.

Zusätzliche Informationen

Weitere Informationen zum Markieren finden Sie in den folgenden Ressourcen.

- [AWS Markierungsprinzipien](#) in der AWS Allgemeinen Referenz zu
- [AWS Tagging-Strategien](#) (herunterladbares PDF)
- [AWS Zugriffskontrolle](#) im AWS IAM-Benutzerhandbuch

- [AWS Markieren von Richtlinien](#) im AWS Organizations -Benutzerhandbuch

Hinzufügen von Tags zu Amazon-Personalize-Ressourcen

Sie können Tag-Schlüssel und -Werte zu Amazon-Personalize-Ressourcen mit der Amazon-Personalize-Konsole, AWS Command Line Interface (AWS CLI) oder AWS SDKs hinzufügen, anzeigen, aktualisieren und entfernen. Die folgenden Beispiele zeigen, wie Sie ein Tag zur Amazon-Personalize-Datensatzgruppe hinzufügen. Sie können Tags auf die gleiche Weise zu anderen Amazon-Personalize-Ressourcen hinzufügen.

Themen

- [Hinzufügen von Tags \(Konsole\)](#)
- [Hinzufügen von Tags \(AWS CLI\)](#)
- [Hinzufügen von Tags \(AWS SDKs\)](#)

Hinzufügen von Tags (Konsole)

Wenn Sie eine Ressource in Amazon Personalize erstellen, können Sie optionale Tags mit der Amazon-Personalize-Konsole hinzufügen. Im folgenden Beispiel wird einer Datensatzgruppe ein Tag hinzugefügt.

So fügen Sie einer neuen Datensatzgruppe Tags hinzu

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie Create dataset group (Dataset-Gruppe erstellen).
3. Geben Sie unter Name einen Namen ein.
4. Wählen Sie für Domain eine Domain aus.
5. Erweitern Sie den Abschnitt Tags und wählen Sie Neues Tag hinzufügen aus.
6. Geben Sie für Schlüssel und Wert die entsprechenden Werte ein.

Beispiel: **Environment** bzw. **Test**.

7. Um weitere Tags hinzuzufügen, wählen Sie Neues Tag hinzufügen aus.

Sie können einer Ressource bis zu 50 Tags hinzufügen.

8. Wählen Sie Weiter, um mit der Erstellung Ihrer Ressource fortzufahren.

Das Hinzufügen von Tags zu einer vorhandenen Ressource ist ähnlich: Wählen Sie Ihre Ressource aus und verwenden Sie die Felder Tags, um Ihre Tags hinzuzufügen.

Hinzufügen von Tags (AWS CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Tags hinzuzufügen, wenn Sie eine Ressource erstellen oder einer vorhandenen Ressource Tags hinzufügen.

Themen

- [Hinzufügen von Tags beim Erstellen einer Ressource](#)
- [Hinzufügen von Tags zu einer vorhandenen Ressource](#)

Hinzufügen von Tags beim Erstellen einer Ressource

Um eine neue Ressource zu erstellen und ihr mit der ein Tag hinzuzufügen AWS CLI, verwenden Sie den entsprechenden `create` Befehl für die Ressource und fügen Sie den `tags` Parameter und die Werte ein. Der folgende Befehl erstellt beispielsweise eine neue Domänendatensatzgruppe mit dem Namen `myDatasetGroup` für die `ECOMMERCE`-Domäne und fügt die folgenden Tags hinzu: einen `Environment` Tag-Schlüssel mit einem `Test` Tag-Wert sowie einen `Owner` Tag-Schlüssel und einen `xyzCorp` Wert.

```
aws personalize create-dataset-group \  
--name myDatasetGroup \  
--domain ECOMMERCE \  
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

Informationen zu den Befehlen, die Sie zum Erstellen einer Amazon-Personalize-Ressource verwenden können, finden Sie in der [Amazon-Personalize- AWS CLI Befehlsreferenz](#).

Hinzufügen von Tags zu einer vorhandenen Ressource

Um einer vorhandenen Ressource ein Tag hinzuzufügen, verwenden Sie den `tag-resource` Befehl. Geben Sie den ARN der Ressource an und geben Sie den Tag-Schlüssel und den Wert im `tags` Parameter an.

```
aws personalize tag-resource \  
--resource-arn resource ARN \  
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

```
--tags tagKey=key,tagValue=value
```

Hinzufügen von Tags (AWS SDKs)

Sie können die - AWS SDKs verwenden, um Tags hinzuzufügen, wenn Sie eine Ressource erstellen, oder um Tags zu einer vorhandenen Ressource hinzuzufügen.

Themen

- [Hinzufügen von Tags beim Erstellen einer Ressource](#)
- [Hinzufügen von Tags zu einer vorhandenen Ressource](#)

Hinzufügen von Tags beim Erstellen einer Ressource

Verwenden Sie die entsprechende `create` Methode, um eine neue Ressource zu erstellen und ihr mit den - AWS SDKs ein Tag hinzuzufügen. Verwenden Sie den `tags` Parameter, um die Schlüssel-Wert-Paare für jedes Ihrer Tags anzugeben. Der folgende Code erstellt beispielsweise eine neue Domain-Datensatzgruppe mit dem Namen `myDatasetGroup` für die `ECOMMERCE`-Domain und fügt die folgenden Tags hinzu: einen `Environment` Tag-Schlüssel mit einem `Test` Tag-Wert sowie einen `Owner` Tag-Schlüssel und einen `xyzCorp` Wert.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
    name = 'myDatasetGroup'
    domain = 'ECOMMERCE'
    tags = [
        {
            'tagKey': 'Environment',
            'tagValue': 'Test'
        },
        {
            'tagKey': 'Owner',
            'tagValue': 'xyzCorp'
        }
    ]
)
```

```
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                             String datasetGroupName,
                                             String domain) {

    try {

        ArrayList <Tag> tags = new ArrayList<>();

        Tag tag1 = Tag.builder()
            .tagKey("Environment")
            .tagValue("Test")
            .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
            .tagKey("Owner")
            .tagValue("xyzCorp")
            .build();
        tags.add(tag2);

        CreateDatasetGroupRequest createDatasetGroupRequest =
        CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .domain(domain)
            .tags(tags)
            .build();

        return
        personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```


Hinzufügen von Tags zu einer vorhandenen Ressource

Der folgende Code zeigt, wie Sie einer vorhandenen Amazon-Personalize-Ressource ein Tag hinzufügen. Geben Sie den Amazon-Ressourcennamen (ARN) der Ressource an, der Sie Tags hinzufügen möchten, und geben Sie Schlüssel-Wert-Paare für jedes Ihrer Tags an.

SDK for Python (Boto3)

```
import boto3
personalize = boto3.client('personalize')

add_tags_response = personalize.tag_resource(
    resourceArn = "resourceArn",
    tags = [
        {
            'tagKey': 'Environment',
            'tagValue': 'Test'
        },
        {
            'tagKey': 'Owner',
            'tagValue': 'xyzCorp'
        }
    ]
)
```

SDK for Java 2.x

```
public static void tagResource(PersonalizeClient personalizeClient,
                              String resourceArn,
                              String domain) {

    try {

        ArrayList <Tag> tagList = new ArrayList<>();

        Tag tag1 = Tag.builder()
            .tagKey("Environment")
            .tagValue("Test")
            .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
            .tagKey("Owner")
            .tagValue("xyzCorp")
```

```
        .build();
        tags.add(tag2);

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
            .resourceArn(resourceArn)
            .tags(tagList)
            .build();

        personalizeClient.tagResource(tagResourceRequest);
        System.out.println("Tags have been added to "+ resourceArn);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

Entfernen von Tags aus Amazon-Personalize-Ressourcen

Sie können Tags aus Amazon-Personalize-Ressourcen mit der Amazon-Personalize-Konsole oder der [UntagResource](#) -API-Operation mit der AWS Command Line Interface (AWS CLI) oder AWS SDKs entfernen. Die folgenden Beispiele zeigen, wie Sie ein Tag aus einer Amazon-Personalize-Datensatzgruppe entfernen. Sie können Tags auf die gleiche Weise aus anderen Amazon-Personalize-Ressourcen entfernen.

Themen

- [Entfernen von Tags \(Konsole\)](#)
- [Entfernen von Tags \(AWS CLI\)](#)
- [Entfernen von Tags \(AWS SDKs\)](#)

Entfernen von Tags (Konsole)

Nachdem Sie einer Ressource in Amazon Personalize Tags hinzugefügt haben, können Sie die Tags mit der Amazon-Personalize-Konsole entfernen. Im folgenden Beispiel wird ein Tag aus einer Datensatzgruppe entfernt.

So entfernen Sie Tags aus einer Datensatzgruppe

1. Öffnen Sie die Amazon-Personalize-Konsole unter <https://console.aws.amazon.com/personalize/home> und melden Sie sich bei Ihrem -Konto an.
2. Wählen Sie Ihre Datensatzgruppe aus.
3. Wählen Sie unten auf der Seite die Registerkarte Tags und dann Tags verwalten aus.
4. Wählen Sie für jedes Tag, das Sie entfernen möchten, Entfernen aus.
5. Wählen Sie Speichern, um die Tags zu entfernen.

Entfernen von Tags (AWS CLI)

AWS CLIVERwenden Sie den folgenden `untag-resource` Befehl, um Tags aus einer vorhandenen Ressource mit der zu entfernen. `resource-arn`Geben Sie für den Amazon-Ressourcennamen (ARN) der Ressource an. Geben Sie für die Schlüssel der Tags `tag-keys`, die entfernt werden sollen.

```
aws personalize untag-resource \  
--resource-arn resource ARN \  
--tag-keys key1 key2
```

Entfernen von Tags (AWS SDKs)

Um Tags aus einer vorhandenen Amazon-Personalize-Ressource mit den - AWS SDKs zu entfernen, verwenden Sie die [UntagResource](#)-API-Operation. Der folgende Code zeigt, wie Sie mit dem SDK for Python (Boto3) mehrere Tags aus einer Datensatzgruppe entfernen. `resourceArn`Geben Sie für den Amazon-Ressourcennamen (ARN) der Ressource an. Geben Sie für die Schlüssel der Tags `tagKeys`, die entfernt werden sollen.

```
import boto3  
  
personalize = boto3.client('personalize')  
  
response = personalize.untag_resource(  
    resourceArn="Resource ARN",  
    tagKeys=["tag1Key", "tag2Key"]  
)
```

Verwenden von Tags in IAM-Richtlinien

Nachdem Sie mit der Implementierung von Tags begonnen haben, können Sie tagbasierte Berechtigungen auf Ressourcenebene auf AWS Identity and Access Management (IAM)-Richtlinien und API-Vorgänge anwenden. Dies umfasst Vorgänge, die das Hinzufügen von Tags zu Ressourcen beim Erstellen von Ressourcen unterstützen. Durch die Verwendung von Tags auf diese Weise können Sie eine detaillierte Kontrolle darüber implementieren, welche Gruppen und Benutzer in Ihrem AWS Konto über die Berechtigung zum Erstellen und Markieren von Ressourcen verfügen und welche Gruppen und Benutzer über die Berechtigung zum Erstellen, Aktualisieren und Entfernen von Tags verfügen.

Sie können beispielsweise eine Richtlinie erstellen, die einem Benutzer vollen Zugriff auf alle Amazon-Personalize-Ressourcen ermöglicht, wobei sein Name ein Wert im `Owner` Tag für die Ressource ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "personalize:*",
      "Resource": "*",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

Das folgende Beispiel zeigt, wie Sie eine Richtlinie erstellen, um das Erstellen und Löschen eines Datensatzes zu ermöglichen. Diese Operationen sind nur zulässig, wenn der Benutzername lautet `johndoe`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Effect": "Allow",
    "Action": [
      "personalize:CreateDataset",
      "personalize>DeleteDataset"
    ],
    "Resource": "arn:aws:personalize:*:*:dataset/*",
    "Condition": {
      "StringEquals": {"aws:username" : "johndoe"}
    }
  },
  {
    "Effect": "Allow",
    "Action": "personalize:DescribeDataset",
    "Resource": "*"
  }
]
```

Wenn Sie Tag-basierte Berechtigungen auf Ressourcenebene definieren, werden die Berechtigungen sofort wirksam. Dies bedeutet, dass Ihre Ressourcen besser geschützt sind, sobald sie erstellt wurden, und Sie schnell damit beginnen können, die Verwendung von Tags für neue Ressourcen zu erzwingen. Mithilfe von Berechtigungen auf Ressourcenebene können Sie auch steuern, welche Tag-Schlüssel und -Werte können mit neuen und vorhandenen Ressourcen verknüpft werden können. Weitere Informationen finden Sie unter [Zugriffssteuerung mit Tags](#) im AWS -IAM-Benutzerhandbuch.

Fehlerbehebung

Die folgenden Themen enthalten Antworten auf häufig gestellte Fragen und Ratschläge zur Fehlerbehebung für Fehlermeldungen, die möglicherweise bei Amazon Personalize auftreten. Eine kurze Referenz, mit der Sie feststellen können, ob Amazon Personalize Ihrem Anwendungsfall passt, finden Sie in der [Amazon Personalize Spickzettel](#) im [Amazon Personalize Proben](#)-Repository.

Themen

- [Häufig gestellte Fragen](#)
- [Fehlermeldungen](#)

Häufig gestellte Fragen

Im Folgenden finden Sie Antworten auf häufig gestellte Fragen zum Import von Daten, zu Schulungen, zur Modellbereitstellung, zu Empfehlungen und Filtern in Amazon Personalize.

Weitere Fragen und Antworten finden Sie im [Amazon Personalize Cheat Sheet im Amazon Personalize Personalize-Beispiel-Repository](#).

Themen

- [Datenimport und -verwaltung](#)
- [Eine benutzerdefinierte Lösung und Lösungsversion erstellen](#)
- [Modellbereitstellung \(benutzerdefinierte Kampagnen\)](#)
- [Empfehlungen](#)
- [Empfehlungen zum Filtern](#)

Datenimport und -verwaltung

In welchem Format sollten meine Massendaten vorliegen?

Ihre Massendaten müssen im CSV-Format (Comma-Separated Values) vorliegen. Die erste Zeile Ihrer CSV-Datei muss Spaltenüberschriften enthalten. Die Spaltenüberschriften in Ihrer CSV-Datei müssen dem zum Erstellen des Datasets verwendeten Schema entsprechen. Wenn Ihre Daten nicht-ASCII-kodierte Zeichen enthalten, muss Ihre CSV-Datei im UTF-8-Format codiert sein. Setzen Sie

Überschriften nicht in Anführungszeichen („“). `TIMEESTAMP` und die `CREATION_TIMESTAMP` Daten müssen im UNIX-Epochenzeitformat vorliegen. Weitere Hinweise zu Zeitstempeldaten finden Sie unter [Zeitstempeldaten](#). Weitere Informationen zu Schemata finden Sie unter [Schemata](#).

Vollständige Richtlinien für das Datenformat finden Sie unter [Richtlinien für Datenformate](#). Wenn Sie sich nicht sicher sind, wie Sie Ihre Daten formatieren sollen, können Sie Amazon SageMaker Data Wrangler (Data Wrangler) verwenden, um Ihre Daten vorzubereiten. Weitere Informationen finden Sie unter [Vorbereiten und Importieren von Daten mit Amazon SageMaker Data Wrangler](#).

Wie viele Trainingsdaten benötige ich?

Für alle Anwendungsfälle (Domain-Datensatzgruppen) und benutzerdefinierte Rezepte müssen Ihre Interaktionsdaten Folgendes enthalten:

- Mindestens 1000 Datensätze zu Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Für Qualitätsempfehlungen empfehlen wir, mindestens 50.000 Artikelinteraktionen von mindestens 1.000 Benutzern mit jeweils zwei oder mehr Artikelinteraktionen durchzuführen.

Sie können mit einem leeren Datensatz für Artikelinteraktionen beginnen und, wenn Sie genügend Daten aufgezeichnet haben, Ihre Empfehlungsversion (Domain-Datensatzgruppe) oder benutzerdefinierte Lösungsversion erstellen, die nur neue aufgezeichnete Ereignisse verwendet. Für einige Rezepte und Anwendungsfälle sind möglicherweise zusätzliche Datenanforderungen erforderlich. Informationen zu den Anforderungen für Anwendungsfälle finden Sie unter [Auswahl eines Anwendungsfalls](#). Informationen zu Rezeptanforderungen finden Sie unter [Ein Rezept wählen](#).

Wie aktualisiere ich die Attribute eines Artikels oder eines Benutzers?

Verwenden Sie die Amazon Personalize Personalize-Konsole [PutItems](#) oder die [PutUsers](#) Operationen oder, um einen Artikel oder Benutzer mit derselben Artikel-ID, aber mit den geänderten Attributen zu importieren.

Wie lösche ich einen Artikel oder einen Benutzer?

Amazon Personalize unterstützt das Löschen eines bestimmten Artikels oder Benutzers nicht. Um sicherzustellen, dass ein Artikel oder ein Benutzer nicht in den Empfehlungen erscheint, verwenden

Sie einen Filter, um Artikel auszuschließen. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wie lösche ich ein Schema?

Sie können ein Schema nur mit der [DeleteSchema](#) Operation löschen. Sie können die Amazon Personalize Personalize-Konsole nicht verwenden, um ein Schema zu löschen.

Eine benutzerdefinierte Lösung und Lösungsversion erstellen

Welches Rezept sollte ich verwenden?

Welches Amazon Personalize Personalize-Rezept Sie verwenden, hängt von Ihrem Anwendungsfall ab. Informationen zur Zuordnung von Anwendungsfällen zu Rezepten finden Sie unter [Ein Rezept wählen](#). Das [Amazon Personalize Cheat Sheet](#) enthält auch Informationen zu Anwendungsfällen und Rezepten.

Wie oft sollte ich mein Modell umschulen?

Durch Umschulungen können Sie dafür sorgen, dass Ihre Empfehlungen relevant bleiben, wenn Ihr Katalog wächst und Benutzer mit Artikeln interagieren. Die Häufigkeit von Umschulungen hängt von Ihren Geschäftsanforderungen und dem von Ihnen verwendeten Rezept ab. Für die meisten Workloads empfehlen wir, wöchentlich eine neue Lösungsversion mit aktiviertem Trainingsmodus zu erstellen. FULL Dadurch wird eine neue Lösungsversion erstellt, die auf der Gesamtheit der Trainingsdaten aus den Datensätzen in Ihrer Datensatzgruppe basiert.

Weitere Informationen finden Sie unter [Aufrechterhaltung der Relevanz von Empfehlungen](#).

Sollte ich AutoML verwenden?

Nein, stattdessen empfehlen wir Ihnen, Ihren Anwendungsfall verschiedenen Amazon Personalize-Rezepten zuzuordnen und ein Rezept auszuwählen. Informationen zum Abgleichen von Anwendungsfällen mit Rezepten finden Sie unter [Ein Rezept wählen](#).

Modellbereitstellung (benutzerdefinierte Kampagnen)

Wie lege ich einen maximalen Transaktionsdurchsatz für eine Kampagne fest?

Sie können nur den Minstdurchsatz für eine Kampagne festlegen. Wenn Sie eine Amazon Personalize-Kampagne erstellen, geben Sie eine spezielle Transaktionskapazität an, um Empfehlungen für Ihre Anwendungsbenutzer in Echtzeit zu erstellen. Wenn Ihr TPS darüber hinaus

steigt `minProvisionedTPS`, skaliert Amazon Personalize die bereitgestellte Kapazität automatisch nach oben und unten, aber niemals unter den Wert `minProvisionedTPS`. Weitere Informationen finden Sie unter [Minimale Anzahl bereitgestellter Transaktionen pro Sekunde und auto-scaling](#).

Wie überwache ich die Kosten meiner Kampagnen?

Das Amazon Personalize Monitor-Projekt bietet ein CloudWatch Dashboard, benutzerdefinierte Metriken, Nutzungsalarme und Funktionen zur Kostenoptimierung für Amazon Personalize Personalize-Kampagnen. Sehen Sie sich [Amazon Personalize Monitor](#) im Amazon [Personalize Personalize-Beispiel-Repository](#) an.

Empfehlungen

Woran erkenne ich, ob mein Amazon Personalize Personalize-Modell Qualitätsempfehlungen generiert?

Bewerten Sie die Leistung Ihrer Lösungsversion anhand von Offline- und Online-Metriken (siehe [Auswerten einer Lösungsversion mit Metriken](#)) und Online-Tests (z. B. A/B-Tests). Weitere Informationen zu Tests finden Sie unter [Verwenden von A/B-Tests zur Messung der Wirksamkeit von Empfehlungen, die von Amazon Personalize generiert wurden](#)

Wie lösche ich meinen Batch-Inferenz-Job und warum hat er den Status „aktiv“?

Batch-Inferenzjobs können nicht gelöscht werden. Wenn der Status eines Batch-Inferenzjobs aktiv ist, ist der Job abgeschlossen. Sie können im Amazon S3 S3-Ausgabe-Bucket oder -Ordner auf Ihre Empfehlungen zugreifen. Sobald der Job abgeschlossen ist, entstehen Ihnen keine zusätzlichen Kosten durch den Batch-Inferenz-Job. Es können jedoch zusätzliche Gebühren für andere Dienste wie Amazon S3 für die Speicherung von Eingabe- und Ausgabedaten anfallen.

Warum werden in meiner von SIMS unterstützten Kampagne Artikel empfohlen, die sich anhand von Metadaten nicht ähneln?

SIMS verwendet Ihren Datensatz mit Artikelinteraktionen, um Ähnlichkeit zu ermitteln, nicht Artikelmetadaten wie Farbe oder Preis. SIMS identifiziert das gleichzeitige Vorkommen des Elements in den Benutzerhistorien in Ihrem Interaktionsdatensatz, um ähnliche Artikel zu empfehlen. Weitere Informationen finden Sie unter [SIMS-Rezept](#).

Kann ich mit einem einzigen `GetRecommendations` API-Vorgang mehr als 500 Elemente abrufen?

500 ist die maximale Anzahl von Elementen, die Sie in einem einzigen abrufen können [GetRecommendations](#). Dieser Wert kann nicht erhöht werden.

Empfehlungen zum Filtern

Warum werden meine Empfehlungen nicht wie erwartet gefiltert?

Dies kann verschiedene Gründe haben:

- Möglicherweise liegt ein Problem mit dem Format oder der Syntax Ihres Filterausdrucks vor. Beispiele für korrekt formatierte Filterausdrücke finden Sie unter [Beispiele für Filterausdrücke](#).
- Amazon Personalize berücksichtigt bis zu 100 der neuesten Interaktionen pro Benutzer und Ereignistyp. Dies ist ein anpassbares Kontingent. Sie können über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen.

Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Wie kann ich bereits gekaufte Artikel aus den Empfehlungen entfernen?

Wenn Sie für E-COMMERCE-Domain-Datensatzgruppen einen Empfehlungsgeber mit dem [Kunden, die X auch angesehen haben](#) Anwendungsfall [Empfohlen für Sie](#) oder erstellen, filtert Amazon Personalize automatisch Artikel, die der Benutzer gekauft hat, basierend auf der von Ihnen angegebenen `userId` und Ereignissen. `Purchase`

Verwenden Sie für andere Anwendungsfälle von Domain-Datensatzgruppen oder benutzerdefinierte Ressourcen einen Filter, um gekaufte Artikel zu entfernen. Fügen Sie Ihren Daten ein `Purchased` Ereignistypattribut hinzu, zeichnen Sie Einkaufsereignisse mit dem `PutItems` Vorgang auf und erstellen Sie einen Filter, der gekaufte Artikel aus Empfehlungen entfernt. Beispiel:

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("purchased")
```

Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Fehlermeldungen

In den folgenden Abschnitten werden einige der Meldungen aufgeführt und erklärt, die bei der Verwendung von Amazon Personalize auftreten können.

Themen

- [Datenimport und -verwaltung](#)

- [Eine Lösung und eine Lösungsversion erstellen \(benutzerdefinierte Ressourcen\)](#)
- [Modellbereitstellung \(benutzerdefinierte Kampagnen\)](#)
- [Empfehlungsgeber \(Domain-Datensatzgruppen\)](#)
- [Empfehlungen](#)
- [Empfehlungen zum Filtern](#)

Datenimport und -verwaltung

Fehlermeldung: Ungültiger Datenspeicherort.

Stellen Sie sicher, dass Sie die richtige Syntax für Ihren Amazon S3 S3-Bucket-Standort verwendet haben. Verwenden Sie für Datensatz-Importaufträge die folgende Syntax für den Speicherort Ihrer Daten in Amazon S3:

s3://<name of your S3 bucket>/<folder path>/<CSVfilename>

Wenn sich Ihre CSV-Dateien in einem Ordner befinden und Sie mehrere Dateien mit einem Datensatz-Importauftrag hochladen möchten, verwenden Sie diese Syntax ohne den CSV-Dateinamen.

Fehlermeldung: Beim Aufrufen des CreateDatasetImportJob Vorgangs ist ein Fehler aufgetreten (LimitExceededException): Mehr als 5 Ressourcen mit dem Status PENDING oder IN_PROGRESS.

Sie können pro Region insgesamt 5 ausstehende oder in Bearbeitung befindliche Datensatz-Importaufträge haben. Dieses Kontingent ist nicht anpassbar. Eine vollständige Liste der Kontingente für Amazon Personalize finden Sie unter [Endpunkte und Kontingente von Amazon Personalize](#).

Fehlermeldung: Fehler beim Erstellen eines Datenimportauftrags für den Datensatz... Unzureichende Rechte für den Zugriff auf Daten in Amazon S3. <dataset type>

Gewähren Sie Amazon Personalize Zugriff auf Ihre Amazon S3 S3-Ressourcen, indem Sie Ihrem Amazon S3-Bucket und Ihrer Amazon Personalize-Servicerolle Zugriffsrichtlinien zuordnen. Siehe [Amazon Personalize Zugriff auf Amazon S3-Ressourcen gewähren](#).

Wenn Sie AWS Key Management Service (AWS KMS) für die Verschlüsselung verwenden, müssen Sie Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Fehlermeldung: Datenimport-Job-Datensatz konnte nicht erstellt werden... In der CSV-Eingabe fehlen die folgenden Spalten: [COLUMN_NAME, COLUMN_NAME<dataset type>].

Die Daten, die Sie in Amazon Personalize importieren, einschließlich Attributnamen und Datentypen, müssen dem Schema des Ziel-Datensets entsprechen. Weitere Informationen finden Sie unter [Schemata](#).

Fehlermeldung: <COLUMN_NAME>Die Länge für darf nicht mehr als Zeichen betragen<character limit>. Wenn keine Werte die Zeichenbeschränkung überschreiten, stellen Sie sicher, dass Ihre Daten den unter <https://docs.aws.amazon.com/personalize/latest/dg/data-prep-formatting.html> aufgeführten Formatierungsrichtlinien entsprechen.

Stellen Sie sicher, dass alle Werte in dieser Spalte die Zeichenbeschränkung nicht überschreiten. Wenn keine Werte die Zeichenbeschränkung überschreiten, überprüfen Sie alle vorangegangenen Textfelder auf Folgendes:

- Stellen Sie sicher, dass alle Textdaten in doppelte Anführungszeichen gesetzt sind. Verwenden Sie das \ Zeichen, um doppelte Anführungszeichen oder \ Zeichen in Ihren Daten zu maskieren.
- Stellt sicher, dass sich jeder Datensatz in Ihrer CSV-Datei in einer einzigen Zeile befindet.

Eine Lösung und eine Lösungsversion erstellen (benutzerdefinierte Ressourcen)

Fehlermeldung: Die Erstellung ist fehlgeschlagen. Der Datensatz hat weniger als 25 Benutzer mit jeweils mindestens 2 Interaktionen.

Sie müssen mehr Daten importieren, bevor Sie das Modell trainieren können. Die minimalen Datenanforderungen zum Schulen eines Modells sind:

- Mindestens 1000 Datensätze über Artikelinteraktionen von Benutzern, die mit Artikeln in Ihrem Katalog interagieren. Diese Interaktionen können aus Massenimporten, gestreamten Ereignissen oder beidem stammen.
- Mindestens 25 eindeutige Benutzer-IDs mit jeweils mindestens zwei Artikelinteraktionen.

Um Empfehlungen in Echtzeit zu erhalten, importieren Sie mehr Daten mit einem Job zum Importieren von Datensätzen oder zeichnen Sie mit einem Event-Tracker und dem [PutEvents](#) Vorgang mehr [Interaktionsergebnisse](#) für Ihre Benutzer auf. Weitere Informationen zur Aufzeichnung von Ereignissen in Echtzeit finden Sie unter [Ereignisse aufzeichnen](#).

Batch-Empfehlungen erhalten Sie, indem Sie Ihre Daten mit einem Datensatz-Importjob importieren, wenn Sie mehr Daten haben. Weitere Informationen zum Importieren von Massendaten finden Sie unter [Schritt 2: Daten vorbereiten und importieren](#).

Modellbereitstellung (benutzerdefinierte Kampagnen)

Fehler: Es kann keine Kampagne erstellt werden. Mehr als 5 Ressourcen im Status AKTIV. Bitte löschen Sie einige und versuchen Sie es erneut.

Sie können insgesamt 5 aktive Amazon Personalize-Kampagnen pro Datensatzgruppe haben. Dieses Kontingent ist anpassbar und Sie können über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen. Eine vollständige Liste der Limits und Kontingente für Amazon Personalize finden Sie unter [Endpunkte und Kontingente von Amazon Personalize](#).

Empfehlungsgeber (Domain-Datensatzgruppen)

Fehler: Der Datensatz hat nach dem Filtern nach Ereignistyp weniger als 1000 Interaktionen: <event type>

Verschiedene Anwendungsfälle erfordern unterschiedliche Ereignistypen. Ihre Daten müssen mindestens 1000 Ereignisse mit dem für Ihren Anwendungsfall erforderlichen Typ enthalten. Weitere Informationen finden Sie unter [Auswahl eines Anwendungsfalls](#)

Empfehlungen

Fehlermeldung beim Batch-Inferenzjob: Ungültiger S3-Eingabepfad oder Ungültiger S3-Ausgabepfad

Stellen Sie sicher, dass Sie die richtige Syntax für Ihre Amazon S3 S3-Eingabe- oder -Ausgabespeicherorte verwenden. Stellen Sie außerdem sicher, dass sich Ihr Ausgabeort von Ihren Eingabedaten unterscheidet. Es sollte sich um einen Ordner im selben Amazon S3 S3-Bucket oder in einem anderen Bucket handeln.

Verwenden Sie die folgende Syntax für den Speicherort der Eingabedatei in Amazon S3: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>**

Verwenden Sie die folgende Syntax für den Ausgabeordner in Amazon S3: **s3://<name of your S3 bucket>/<output folder name>/**

Empfehlungen zum Filtern

Fehlermeldung: Filter konnte nicht erstellt werden. Ungültiges Eingabesymbol: \$parameterName. Platzhalter sind mit dem NOT_IN-Operator nicht zulässig.

Sie können keine Platzhalterparameter in einem Filterausdruck verwenden, der den NOT_IN-Operator verwendet. Verwenden Sie stattdessen den IN-Operator und die entgegengesetzte Aktion: Verwenden Sie Include statt Exclude (oder umgekehrt).

Wenn Sie beispielsweise verwenden möchten `INCLUDE ItemID WHERE Items.GENRE NOT IN ($GENRE)`, können Sie dieselben Ergebnisse verwenden `EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE)` und erhalten dieselben Ergebnisse.

Weitere Informationen zu Filtern finden Sie unter [Filterausdruckselemente](#).

Fehlermeldung: Filter konnte nicht erstellt werden. Ungültiger Ausdruck... beim Filtern nach Feldern vom Typ Boolean

Sie können in Ihrem Schema keine Filterausdrücke erstellen, die mithilfe von Werten vom Typ Boolean filtern. Um auf der Grundlage boolescher Werte zu filtern, verwenden Sie ein Schema mit einem Feld vom Typ `String` und verwenden Sie die Werte `True` und `False` in Ihren Daten. Oder Sie können den Typ `int` oder `long` und die Werte `0` und verwenden. `1`

Weitere Informationen zu Filtern finden Sie unter [Filterausdruckselemente](#).

Angeben von Ressourcen mit AWS CloudFormation

Amazon Personalize ist integriert mit AWS CloudFormation, ein Service, der Ihnen hilft, Ihre zu modellieren und einzurichten AWS-Ressourcen, damit Sie weniger Zeit für die Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur aufwenden müssen. Sie erstellen eine Vorlage, die alle AWS-Ressourcen, die Sie angeben können (z. B. Amazon Personalize von Datensatzgruppen). AWS CloudFormation stellt Ihnen anschließend diese Ressourcen bereit und konfiguriert sie.

Wenn Sie verwenden AWS CloudFormation verwenden, können Sie Ihre Vorlage wiederverwenden, um Ihre Amazon Personalize -Ressourcen konsistent und wiederholt einzurichten. Sie beschreiben Ihre Ressourcen dann einmal und können die gleichen Ressourcen dann in mehreren AWS-Konten-Konten und -Regionen immer wieder bereitstellen.

Themen

- [Amazon Personalize und AWS CloudFormation Vorlagen](#)
- [Beispiel AWS CloudFormation-Vorlagen für Amazon Personalize Resources](#)
- [Weitere Informationen zu AWS CloudFormation](#)

Amazon Personalize und AWS CloudFormation Vorlagen

Um Ressourcen für Amazon Personalize und zugehörige Dienste bereitzustellen und zu konfigurieren, müssen Sie verstehen [AWS CloudFormation Vorlagen](#) aus. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation-Stacks bereitstellen möchten. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormation Designer verwenden, der den Einstieg in die Arbeit mit AWS CloudFormation-Vorlagen erleichtert. Weitere Informationen finden Sie unter [Was ist AWS CloudFormation-Designer?](#) im AWS CloudFormation-Benutzerhandbuch.

Amazon Personalize unterstützt die Angabe von Datensätzen, Datensatzgruppen, Datensatzimport-Jobs, Schemas und Lösungen in AWS CloudFormation aus. Weitere Informationen finden Sie im [Amazon Personalize resource type reference](#) im AWS CloudFormation-Benutzerhandbuch aus.

BeispielAWS CloudFormation-Vorlagen für Amazon Personalize Resources

FolgendesAWS CloudFormationVorlagenbeispiele zeigen, wie Sie verschiedene Amazon Personalize -Ressourcenangeben.

Themen

- [CreateDatasetGruppe](#)
- [CreateDataset](#)
- [CreateSchema](#)
- [CreateSolution](#)

CreateDatasetGruppe

JSON

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Resources":{
    "MyDatasetGroup": {
      "Type": "AWS::Personalize::DatasetGroup",
      "Properties": {
        "Name": "my-dataset-group-name"
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDatasetGroup:
    Type: 'AWS::Personalize::DatasetGroup'
    Properties:
      Name: my-dataset-group-name
```


CreateDataset

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyDataset": {
      "Type": "AWS::Personalize::Dataset",
      "Properties": {
        "Name": "my-dataset-name",
        "DatasetType": "Interactions",
        "DatasetGroupArn": "arn:aws:personalize:us-west-2:123456789012:dataset-
group/dataset-group-name",
        "SchemaArn": "arn:aws:personalize:us-west-2:123456789012:schema/schema-
name",
        "DatasetImportJob": {
          "JobName": "my-import-job-name",
          "DataSource": {
            "DataLocation": "s3://bucket-name/file-name.csv"
          },
          "RoleArn": "arn:aws:iam::123456789012:role/personalize-role"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDataset:
    Type: 'AWS::Personalize::Dataset'
    Properties:
      Name: my-dataset-name
      DatasetType: Interactions
      DatasetGroupArn: 'arn:aws:personalize:us-west-2:123456789012:dataset-group/
dataset-group-name'
      SchemaArn: 'arn:aws:personalize:us-west-2:123456789012:schema/schema-name'
      DatasetImportJob:
        JobName: my-import-job-name
        DataSource:
```

```
DataLocation: 's3://bucket-name/file-name.csv'
RoleArn: 'arn:aws:iam::123456789012:role/personalize-role'
```

CreateSchema

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MySchema": {
      "Type": "AWS::Personalize::Schema",
      "Properties": {
        "Name": "my-schema-name",
        "Schema": "{\"type\": \"record\", \"name\": \"Interactions\",
  \"namespace\": \"com.amazonaws.personalize.schema\", \"fields\": [ { \"name\":
  \"USER_ID\", \"type\": \"string\" }, { \"name\": \"ITEM_ID\", \"type\": \"string
  \\\" }, { \"name\": \"TIMESTAMP\", \"type\": \"long\"}], \"version\": \"1.0\"}"
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySchema:
    Type: AWS::Personalize::Schema
    Properties:
      Name: "my-schema-name"
      Schema: >-
        {"type": "record", "name": "Interactions", "namespace":
        "com.amazonaws.personalize.schema", "fields": [ { "name": "USER_ID",
        "type": "string" }, { "name": "ITEM_ID", "type": "string" }, { "name":
        "TIMESTAMP", "type": "long"}], "version": "1.0"}
```

CreateSolution

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MySolution": {
      "Type": "AWS::Personalize::Solution",
      "Properties": {
        "Name": "my-solution-name",
        "DatasetGroupArn": "arn:aws:personalize:us-
west-2:123456789012:dataset-group/my-dataset-group-name",
        "RecipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
        "SolutionConfig": {
          "EventValueThreshold" : ".05"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySolution:
    Type: 'AWS::Personalize::Solution'
    Properties:
      Name: my-solution-name
      DatasetGroupArn: >-
        arn:aws:personalize:us-west-2:123456789012:dataset-group/my-dataset-group-
name
      RecipeArn: 'arn:aws:personalize:::recipe/aws-user-personalization'
      SolutionConfig:
        EventValueThreshold: '.05'
```

Weitere Informationen zu AWS CloudFormation

Weitere Informationen zu AWS CloudFormation finden Sie in den folgenden Ressourcen.

- [AWS CloudFormation](#)
- [AWS CloudFormation-Benutzerhandbuch](#)
- [AWS CloudFormation-API-Referenz](#)
- [AWS CloudFormationBenutzerhandbuch für Befehlszeilenschnittstelle](#)

Sicherheit in Amazon Personalize

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Amazon Personalize verwendet Datenverschlüsselung, um Ihre Daten zu schützen. Weitere Informationen finden Sie unter [Datenverschlüsselung](#). Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS Compliance-Programme AWSAWS](#). Weitere Informationen zu den Compliance-Programmen, die für Amazon Personalize gelten, finden Sie unter [AWS Services im Umfang nach Compliance-Programm AWS](#).
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon Personalize anwenden können. In den folgenden Themen erfahren Sie, wie Sie Amazon Personalize konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon Personalize-Ressourcen überwachen und sichern können.

Themen

- [Datenschutz in Amazon Personalize](#)
- [Identity and Access Management für Amazon Personalize](#)
- [Protokollierung und Überwachung in Amazon Personalize](#)
- [Konformitätsprüfung für Amazon Personalize](#)
- [Resilienz in Amazon Personalize](#)
- [Infrastruktursicherheit in Amazon Personalize](#)
- [Amazon VPC-Endpunkte personalisieren und verbinden \(\)AWS PrivateLink](#)

Datenschutz in Amazon Personalize

Das AWS [Modell](#) der gilt für den Datenschutz in Amazon Personalize. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Personalize oder anderen Geräten arbeiten und die Konsole AWS CLI, API oder AWS SDKs AWS-Services verwenden. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine

Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

In den folgenden Informationen wird erklärt, wo Amazon Personalize Datenverschlüsselung zum Schutz Ihrer Daten verwendet.

Verschlüsselung im Ruhezustand

Alle in Amazon Personalize gespeicherten Daten werden im Ruhezustand immer mit verwalteten AWS Key Management Service (AWS KMS) -Schlüsseln von Amazon Personalize verschlüsselt. Wenn Sie bei der Erstellung der Ressource Ihren eigenen AWS KMS Schlüssel angeben, verwendet Amazon Personalize den Schlüssel, um Ihre Daten zu verschlüsseln und zu speichern. Wenn Sie beispielsweise während des [CreateDatasetGroup](#) Vorgangs einen AWS KMS ARN angeben, verwendet Amazon Personalize den Schlüssel, um Daten zu verschlüsseln und zu speichern, die Sie in alle Datensätze importieren, die Sie in dieser Datensatzgruppe erstellen.

Sie müssen Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Informationen zur Datenverschlüsselung in Amazon S3 finden Sie unter [Schützen von Daten mithilfe von Verschlüsselung](#) im Amazon Simple Storage Service-Benutzerhandbuch. Informationen zur Verwaltung Ihres eigenen AWS KMS Schlüssels finden Sie unter [Schlüssel verwalten](#) im AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung während der Übertragung

Amazon Personalize verwendet TLS mit AWS Zertifikaten, um alle an andere AWS Dienste gesendeten Daten zu verschlüsseln. Jegliche Kommunikation mit anderen AWS Diensten erfolgt über HTTPS, und Amazon Personalize Personalize-Endgeräte unterstützen nur sichere Verbindungen über HTTPS.

Amazon Personalize kopiert Daten aus Ihrem Konto und verarbeitet sie in einem internen AWS System. Bei der Verarbeitung von Daten verschlüsselt Amazon Personalize Daten entweder mit einem Amazon Personalize AWS KMS Personalize-Schlüssel oder einem beliebigen AWS KMS Schlüssel, den Sie angeben.

Schlüsselverwaltung

AWS verwaltet alle Standardschlüssel. AWS KMS Es liegt in Ihrer Verantwortung, alle AWS KMS Schlüssel, die Sie besitzen, zu verwalten. Sie müssen Amazon Personalize und Ihrer Amazon Personalize IAM-Servicerolle die Erlaubnis zur Verwendung Ihres Schlüssels erteilen. Weitere Informationen finden Sie unter [Amazon Personalize die Erlaubnis geben, Ihren AWS KMS Schlüssel zu verwenden](#).

Informationen zur Verwaltung Ihres eigenen AWS KMS Schlüssels finden Sie unter [Schlüssel verwalten im AWS Key Management Service Entwicklerhandbuch](#).

Identity and Access Management für Amazon Personalize

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon Personalize-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert Amazon Personalize mit IAM](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)
- [Fehlerbehebung bei Amazon Personalize: Identität und Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon Personalize ausführen.

Servicebenutzer — Wenn Sie den Amazon Personalize Personalize-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur

Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon Personalize Personalize-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf eine Funktion in Amazon Personalize nicht zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung bei Amazon Personalize: Identität und Zugriff](#).

Service-Administrator — Wenn Sie in Ihrem Unternehmen für die Amazon Personalize-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Personalize. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon Personalize Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit Amazon Personalize verwenden kann, finden Sie unter [So funktioniert Amazon Personalize mit IAM](#)

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon Personalize zu verwalten. Beispiele für identitätsbasierte Amazon Personalize Personalize-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportale anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff:** Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen:** Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Serviceroles oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

- **Service-Rolle:** Eine Service-Rolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Service-Rolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Service-Rolle, die mit einer Service-Rolle verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon EC2 ausgeführte Anwendungen** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen zu ACLs finden Sie unter [Zugriffssteuerungsliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen:** Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.

- **Sitzungsrichtlinien:** Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

So funktioniert Amazon Personalize mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Personalize zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen mit Amazon Personalize verwendet werden können.

IAM-Funktionen, die Sie mit Amazon Personalize verwenden können

IAM-Feature	Amazon Personalize Personalize-Unters tützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja

IAM-Feature	Amazon Personalize Personalize-Unters tützung
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie Amazon Personalize und andere AWS Services mit den meisten IAM-Funktionen funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für Amazon Personalize

Unterstützt Richtlinien auf Identitätsbasis.	Ja
--	----

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon Personalize

Beispiele für identitätsbasierte Richtlinien von Amazon Personalize finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)

Ressourcenbasierte Richtlinien in Amazon Personalize

Unterstützt ressourcenbasierte Richtlinien	Nein
--	------

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) im IAM-Benutzerhandbuch.

Richtlinienmaßnahmen für Amazon Personalize

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen,

die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Amazon Personalize-Aktionen finden Sie unter [Von Amazon Personalize definierte Aktionen](#) in der Service Authorization Reference.

Richtlinienaktionen in Amazon Personalize verwenden das folgende Präfix vor der Aktion:

```
personalize
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "personalize:action1",  
  "personalize:action2"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "personalize:Describe*"
```

Beispiele für identitätsbasierte Richtlinien von Amazon Personalize finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)

Richtlinienressourcen für Amazon Personalize

Unterstützt Richtlinienressourcen

Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"

```

Eine Liste der Amazon Personalize-Ressourcentypen und ihrer ARNs finden Sie unter [Von Amazon Personalize definierte Ressourcen](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Personalize definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien von Amazon Personalize finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)

Schlüssel für Richtlinienbedingungen für Amazon Personalize

Unterstützt servicespezifische Richtlinienbedingungen	Ja
---	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation

aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Amazon Personalize-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Personalize](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Personalize definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien von Amazon Personalize finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Personalize](#)

ACLs in Amazon Personalize

Unterstützt ACLs	Nein
------------------	------

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon Personalize

Unterstützt ABAC (Tags in Richtlinien)	Ja
--	----

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen

Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen zum Taggen von Amazon Personalize Personalize-Ressourcen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#)

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter [Verwenden von Tags in IAM-Richtlinien](#).

Temporäre Anmeldeinformationen mit Amazon Personalize verwenden

Unterstützt temporäre Anmeldeinformationen	Ja
--	----

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden

und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Serviceübergreifende Hauptberechtigungen für Amazon Personalize

Unterstützt Forward Access Sessions (FAS)	Ja
---	----

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon Personalize

Unterstützt Servicerollen	Ja
---------------------------	----

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon Personalize beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon Personalize Ihnen eine Anleitung dazu bietet.

Servicebezogene Rollen für Amazon Personalize

Unterstützt serviceverknüpfte Rollen Nein

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel [aws:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die Amazon Personalize einem anderen Service für die Ressource erteilt.

Um das Problem der verwirrten Stellvertreter in Rollen zu vermeiden, die von Amazon Personalize übernommen wurden, setzen Sie in der Vertrauensrichtlinie der Rolle den Wert `aws:SourceArn` auf `arn:aws:personalize:region:accountNumber:*`. Der Platzhalter (*) wendet die Bedingung für alle Amazon Personalize an.

Die folgende Vertrauensstellungsrichtlinie gewährt Amazon Personalize Zugriff auf Ihre Ressourcen und verwendet die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel, um das Problem des verwirrten Stellvertreters zu verhindern. Verwenden Sie diese Richtlinie, wenn Sie eine Rolle für Amazon Personalize ([Erstellen einer IAM-Rolle für Amazon Personalize](#)) erstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "personalize.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountNumber"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:personalize:region:accountNumber:*"
        }
      }
    }
  ]
}
```

Beispiele für identitätsbasierte Richtlinien für Amazon Personalize

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon Personalize-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS API, der AWS Management Console, der AWS Command Line Interface (AWS CLI) oder ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon Personalize definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Personalize](#) in der Service Authorization Reference.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [AWS verwaltete Richtlinien](#)
- [Verwenden der Amazon Personalize Personalize-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Vollzugriff auf Amazon Personalize gewähren](#)
- [Zulassen des schreibgeschützten Zugriffs auf Amazon Personalize Personalize-Ressourcen](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon Personalize Personalize-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten: Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt

als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs: Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten: IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinien

AWS verwaltete Richtlinien sind Richtlinien, die von erstellt und verwaltet AWS werden. Im Folgenden finden Sie Beispiele für AWS verwaltete Richtlinien, die Sie bei der Arbeit mit Amazon Personalize verwenden könnten.

AmazonPersonalizeFullAccess Richtlinie

Sie können die AWS verwaltete `AmazonPersonalizeFullAccess` Richtlinie verwenden, um Benutzern die folgenden Berechtigungen zu erteilen:

- Greifen Sie auf alle Amazon Personalize-Ressourcen zu
- Metriken auf Amazon veröffentlichen und auflisten CloudWatch
- Auflisten, Lesen, Schreiben und Löschen aller Objekte in einem Amazon S3 S3-Bucket, die Personalize oder personalize im Bucket-Namen enthalten
- Übergeben Sie eine Rolle an Amazon Personalize

AmazonPersonalizeFullAccess bietet mehr Berechtigungen als nötig. Wir empfehlen, eine neue IAM-Richtlinie zu erstellen, die nur die erforderlichen Berechtigungen gewährt (siehe [Amazon Personalize die Berechtigung für den Zugriff auf Ihre -Ressourcen erteilen](#)).

CloudWatchFullAccess

Um Ihren Benutzern die Erlaubnis zu geben, Amazon Personalize mit zu überwachen CloudWatch, fügen Sie die CloudWatchFullAccess Richtlinie Ihrer Rolle bei. Weitere Informationen finden Sie unter [Überwachung von Amazon Personalize](#).

Die CloudWatchFullAccess Richtlinie ist optional und gewährt die Genehmigung für die folgenden Aktionen:

- Veröffentlichen und listen Sie Amazon Personalize Personalize-Metriken in CloudWatch
- Metriken und Metrikstatistiken anzeigen.
- Stellen Sie metrikbasierte Alarme ein.

Verwenden der Amazon Personalize Personalize-Konsole

Um auf die Amazon Personalize Personalize-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon Personalize Personalize-Ressourcen in Ihrem AWS-Konto aufzulisten und anzuzeigen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Vollzugriff auf Amazon Personalize gewähren

Das folgende Beispiel gewährt einem IAM-Benutzer in Ihrem AWS Konto vollen Zugriff auf alle Amazon Personalize und -Aktionen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Zulassen des schreibgeschützten Zugriffs auf Amazon Personalize Personalize-Ressourcen

In diesem Beispiel gewähren Sie einem IAM-Benutzer in Ihrem AWS Konto schreibgeschützten Zugriff auf Ihre Amazon Personalize Personalize-Ressourcen, einschließlich Amazon Personalize Personalize-Datensätze, Datensatzgruppen, Lösungen und Kampagnen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:DescribeAlgorithm",
        "personalize:DescribeBatchInferenceJob",
        "personalize:DescribeBatchSegmentJob",
        "personalize:DescribeCampaign",
        "personalize:DescribeDataset",
        "personalize:DescribeDatasetExportJob",
        "personalize:DescribeDatasetGroup",
        "personalize:DescribeDatasetImportJob",
        "personalize:DescribeEventTracker",
        "personalize:DescribeFeatureTransformation",
        "personalize:DescribeFilter",

```

```
        "personalize:DescribeRecipe",
        "personalize:DescribeRecommender",
        "personalize:DescribeSchema",
        "personalize:DescribeSolution",
        "personalize:DescribeSolutionVersion",
        "personalize:GetSolutionMetrics",
        "personalize:ListBatchInferenceJobs",
        "personalize:ListBatchSegmentJobs",
        "personalize:ListCampaigns",
        "personalize:ListDatasetExportJobs",
        "personalize:ListDatasetGroups",
        "personalize:ListDatasetImportJobs",
        "personalize:ListDatasets",
        "personalize:ListEventTrackers",
        "personalize:ListFilters",
        "personalize:ListRecipes",
        "personalize:ListRecommenders",
        "personalize:ListSchemas",
        "personalize:ListSolutions",
        "personalize:ListSolutionVersions"
    ],
    "Resource": "*"
}
]
```

Fehlerbehebung bei Amazon Personalize: Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Personalize und IAM auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion in Amazon Personalize durchzuführen](#)
- [Ich bin nicht berechtigt, iam durchzuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon Personalize ermöglichen](#)

Ich bin nicht berechtigt, eine Aktion in Amazon Personalize durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven `my-example-widget`-Ressource anzuzeigen, jedoch nicht über `personalize:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
personalize:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `personalize:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon Personalize übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Personalize auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon Personalize ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffssteuerungslisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon Personalize diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Personalize mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Protokollierung und Überwachung in Amazon Personalize

Dieser Abschnitt enthält Informationen zur Überwachung und Protokollierung von Amazon Personalize mit Amazon CloudWatch und AWS CloudTrail.

Themen

- [Überwachung von Amazon Personalize](#)
- [CloudWatch Metriken für Amazon Personalize](#)

- [Protokollieren Amazon Personalize Personalize-API-Aufrufen mit AWS CloudTrail](#)

Überwachung von Amazon Personalize

Mit Amazon können Sie Kennzahlen abrufen CloudWatch, die mit Amazon Personalize verknüpft sind. Sie können Alarme einrichten, die Sie benachrichtigen, wenn eine oder mehrere dieser Metriken einen definierten Schwellenwert überschreiten. Um Metriken zu sehen, können Sie [Amazon CloudWatch](#) AWS Command Line Interface, [Amazon](#) oder die [CloudWatch API](#) verwenden.

Themen

- [Verwenden von CloudWatch Metriken für Amazon Personalize](#)
- [Zugreifen auf Amazon Personalize Personalize-Metriken](#)
- [Erstellen eines Alarms](#)
- [Beispiel für eine serverlose Überwachungs-App von Amazon Personalize](#)

Verwenden von CloudWatch Metriken für Amazon Personalize

Um Metriken zu verwenden, müssen Sie die folgenden Informationen angeben:

- Der -Metrikname
- Die Metrikdimension. Eine Dimension ist ein Name-Wert-Paar, mit dem Sie eine Metrik eindeutig identifizieren.

Sie können Überwachungsdaten für Amazon Personalize mithilfe der AWS Management Console AWS CLI, der oder der CloudWatch API abrufen. Sie können die CloudWatch API auch über eines der AWS SDKs oder die CloudWatch API-Tools verwenden. Die Konsole zeigt eine Reihe von Diagrammen an, die auf den Rohdaten der CloudWatch API basieren. Je nach Anforderungen können Sie entweder die in der Konsole angezeigten oder die mit der API aufgerufenen Graphen verwenden.

In der folgenden Liste finden Sie einige häufige Verwendungszwecke für die Metriken. Es handelt sich dabei um Vorschläge für den Einstieg und nicht um eine umfassende Liste.

Wie gehe ich vor?	Relevante Metrik
Wie verfolge ich die Anzahl der Ereignisse, die aufgezeichnet wurden?	Überwachen Sie die PutEventsRequests - Metrik.
Wie kann ich die DatasetImportJob Fehler überwachen?	Verwenden Sie die DatasetImportJobError -Metrik.
Wie überwache ich die Latenz der GetRecommendations -Aufrufe?	Verwenden Sie die GetRecommendations Latency -Metrik.

Sie müssen über die entsprechenden CloudWatch Berechtigungen verfügen, um Amazon Personalize überwachen zu CloudWatch können. Weitere Informationen finden Sie unter [Authentifizierung und Zugriffskontrolle für Amazon CloudWatch](#).

Zugreifen auf Amazon Personalize Personalize-Metriken

Die folgenden Beispiele zeigen, wie Sie mithilfe der CloudWatch Konsole, der und der CloudWatch API auf Amazon Personalize-Metriken zugreifen können. AWS CLI

So zeigen Sie Metriken an (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie Metriken, klicken Sie auf die Registerkarte Alle Metriken und wählen Sie dann AWS/Personalize.
3. Wählen Sie die Metrikdimension.
4. Wählen Sie die gewünschte Metrik aus der Liste und einen Zeitraum für das Diagramm aus.

So zeigen Sie Metriken für Ereignisse an, die über einen bestimmten Zeitraum empfangen wurden (CLI)

- Öffnen Sie den AWS CLI und geben Sie den folgenden Befehl ein:

```
aws cloudwatch get-metric-statistics \
  --metric-name PutEventsRequests \
  --start-time 2019-03-15T00:00:20Z \
```

```
--period 3600 \  
--end-time 2019-03-16T00:00:00Z \  
--namespace AWS/Personalize \  
--dimensions Name=EventTrackerArn,Value=EventTrackerArn \  
--statistics Sum
```

Dieses Beispiel zeigt die für den angegebenen Ereignis-Tracker ARN empfangenen Ereignisse über einen bestimmten Zeitraum an. Weitere Informationen finden Sie unter [get-metric-statistics](#).

So greifen Sie auf Metriken zu (CloudWatch API)

- Rufen Sie die folgende Seite auf [GetMetricStatistics](#). Weitere Informationen finden Sie in der [Amazon CloudWatch API-Referenz](#).

Erstellen eines Alarms


Sie können einen CloudWatch Alarm erstellen, der eine Amazon Simple Notification Service (Amazon SNS) -Meldung sendet, wenn sich der Status des Alarms ändert. Ein Alarm überwacht eine Metrik über einen bestimmten, von Ihnen festgelegten Zeitraum. Der Alarm führt eine oder mehrere Aktionen durch, die vom Wert der Metrik im Vergleich zu einem gegebenen Schwellenwert in einer Reihe von Zeiträumen abhängt. Die Aktion ist eine Benachrichtigung, die an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Richtlinie gesendet wird.

Bei Alarmen werden nur Aktionen für anhaltende Statusänderungen ausgelöst. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Zustand befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

So richten Sie einen Alarm ein (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und dann Create Alarm aus. Dadurch wird der Assistent zum Erstellen von Alarmen gestartet.
3. Wählen Sie Select metric (Metrik auswählen) aus.
4. Wählen Sie auf der Registerkarte Alle Metriken die OptionAWS/Personalize.
5. Wählen Sie EventTrackerArnund wählen Sie dann PutEventsRequestsMetriken aus.
6. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.

7. Wählen Sie für **Statistic (Statistik)** die Option **Sum (Summe)** aus.
8. Wählen Sie **Select metric (Metrik auswählen)** aus.
9. Geben Sie **Name** und **Description** an. Für **Whenever (Jederzeit)** wählen Sie **>** aus und geben einen maximalen Wert Ihrer Wahl an.
10. Wenn Sie Ihnen eine E-Mail senden CloudWatch möchten, wenn der Alarmstatus erreicht ist, wählen Sie für **Wann immer dieser Alarm:** die Option **Status ist ALARM**. Um Alarme für ein bestehendes Amazon-S3-Thema zu senden, wählen Sie für **Benachrichtigung senden an:** ein bestehendes SNS-Thema aus. Um den Namen und die E-Mail-Adressen für eine neue E-Mail-Abonnementliste festzulegen, wählen Sie **Neue Liste**. CloudWatch speichert die Liste und zeigt sie im Feld an, sodass Sie sie verwenden können, um future Alarme einzustellen.

 **Note**

Wenn Sie **Neue Liste** verwenden, um ein neues Amazon SNS SNS-Thema zu erstellen, müssen die E-Mail-Adressen verifiziert werden, bevor die vorgesehenen Empfänger Benachrichtigungen erhalten. Amazon SNS sendet nur dann eine E-Mail, wenn der Alarm einen Alarmzustand auslöst. Wenn es zu dieser Änderung des Alarmzustands kommt, bevor die E-Mail-Adressen überprüft wurden, erhalten die vorgesehenen Empfänger keine Benachrichtigung.

11. Wählen Sie **Alarm erstellen** aus.

So richten Sie einen Alarm ein (AWS CLI)

- Öffnen Sie den AWS CLI und geben Sie dann den folgenden Befehl ein. Ändern Sie den Wert des `alarm-actions` Parameters so, dass er auf ein Amazon SNS SNS-Thema verweist, das Sie zuvor erstellt haben.

```
aws cloudwatch put-metric-alarm \  
  --alarm-name PersonalizeCLI \  
  --alarm-description "Alarm when more than 10 events occur" \  
  --metric-name PutEventsRequests \  
  --namespace AWS/Personalize \  
  --statistic Sum \  
  --period 300 \  
  --threshold 10 \  
  --comparison-operator GreaterThanThreshold \  
  --evaluation-periods 1 \  
  --actions sns:SendMessage --action-arn arn:aws:sns:us-east-1:123456789012:my-topic:SendMessage
```

```
--unit Count \
--dimensions Name=EventTrackerArn,Value=EventTrackerArn \
--alarm-actions SNSTopicArn
```

Dieses Beispiel zeigt, wie man einen Alarm erstellt, wenn mehr als 10 Ereignisse für den gegebenen Ereignis-Tracker ARN innerhalb von 5 Minuten auftreten. Weitere Informationen finden Sie unter [put-metric-alarm](#).

So richten Sie einen Alarm ein (CloudWatch API)

- Rufen Sie die folgende Seite auf [PutMetricAlarm](#). Weitere Informationen finden Sie unter [Amazon CloudWatch API-Referenz](#).

Beispiel für eine serverlose Überwachungs-App von Amazon Personalize

Eine Beispiel-App, die Überwachungs-, Warnungs- und Optimierungsfunktionen für Amazon Personalize hinzufügt, finden Sie unter Amazon Personalize [Monitor im Amazon Personalize Personalize-Beispiel-Repository](#).

CloudWatch Metriken für Amazon Personalize

Dieser Abschnitt enthält Informationen zu den für Amazon Personalize verfügbaren CloudWatch Amazon-Metriken. Weitere Informationen finden Sie unter [Überwachung von Amazon Personalize](#).

In der folgenden Tabelle sind die Amazon Personalize-Metriken aufgeführt. Alle Metriken außer diesen Statistiken GetRecommendations und GetPersonalizedRanking unterstützen diese: Average, Minimum, Maximum, Sum. GetRecommendations und Sum nur GetPersonalizedRanking Support.

Metrik	Beschreibung
DatasetImportJobRequests	Die Anzahl der erfolgreichen CreateDatasetImportJob API-Aufrufe. Maße: DatasetGroupArn, DatasetArn, DatasetImportJobArn
DatasetImportJobError	Die Anzahl der CreateDatasetImportJob API-Aufrufe, die zu einem Fehler führten.

Metrik	Beschreibung
	Maße: DatasetGroupArn, DatasetArn, DatasetImportJobArn
DatasetImportJobExecutionTime	<p>Die Zeit zwischen dem Aufruf der API <code>CreateDatasetImportJob</code> und dem Abschluss (oder Misserfolg) der Operation.</p> <p>Maße: DatasetGroupArn, DatasetArn, DatasetImportJobArn</p> <p>Einheit: Sekunden</p>
DatasetSize	<p>Die Größe der Daten, die vom Importauftrag für Datensätze importiert werden.</p> <p>Maße: DatasetGroupArn, DatasetArn, DatasetImportJobArn</p> <p>Einheit: Byte</p>
SolutionTrainingJobRequests	<p>Die Anzahl der erfolgreichen CreateSolutionVersion API-Aufrufe.</p> <p>Maße: SolutionArn, SolutionVersionArn</p>
SolutionTrainingJobError	<p>Die Anzahl der <code>CreateSolutionVersion</code> API-Aufrufe, die zu einem Fehler führten.</p> <p>Maße: SolutionArn, SolutionVersionArn</p>
SolutionTrainingJobExecutionTime	<p>Die Zeit zwischen dem Aufruf der API <code>CreateSolutionVersion</code> und dem Abschluss (oder Misserfolg) der Operation.</p> <p>Maße: SolutionArn, SolutionVersionArn</p> <p>Einheit: Sekunden</p>

Metrik	Beschreibung
GetPersonalizedRanking	<p>Ob ein GetPersonalizedRanking API-Aufruf erfolgreich ist. Verwenden Sie die sum Statistik, um die Gesamtzahl der erfolgreichen GetPersonalizedRanking API-Aufrufe anzuzeigen. Diese Metrik unterstützt keine anderen Statistiken.</p> <p>Dimension: CampaignArn</p>
GetPersonalizedRanking4xxErrors	<p>Die Anzahl der GetPersonalizedRanking API-Aufrufe, die einen 4xx-HTTP-Antwortcode zurückgeben.</p> <p>Dimension: CampaignArn</p>
GetPersonalizedRanking5xxErrors	<p>Die Anzahl der GetPersonalizedRanking API-Aufrufe, die einen 5xx-HTTP-Antwortcode zurückgeben.</p> <p>Dimension: CampaignArn</p>
GetPersonalizedRankingLatency	<p>Der Zeitraum zwischen dem Eingang des GetPersonalizedRanking API-Aufrufs und dem Versenden von Empfehlungen (ohne 4xx und 5xx-Fehler).</p> <p>Dimension: CampaignArn</p> <p>Einheit: Millisekunden</p>
GetRecommendations	<p>Ob ein GetRecommendations API-Aufruf erfolgreich ist. Verwenden Sie die sum Statistik, um die Gesamtzahl der erfolgreichen GetRecommendations API-Aufrufe anzuzeigen. Diese Metrik unterstützt keine anderen Statistiken.</p> <p>Dimension: CampaignArn</p>

Metrik	Beschreibung
GetRecommendations 4xxErrors	Die Anzahl der GetRecommendations API-Aufrufe, die einen 4xx-HTTP-Antwortcode zurückgeben. Dimension: CampaignArn
GetRecommendations5XX-Fehler	Die Anzahl der GetRecommendations API-Aufrufe, die einen 5xx-HTTP-Antwortcode zurückgeben. Abmessung: CampaignArn
GetRecommendationsLatency	Der Zeitraum zwischen dem Eingang des GetRecommendations API-Aufrufs und dem Versenden von Empfehlungen (ohne 4xx und 5xx-Fehler). Dimension: CampaignArn Einheit: Millisekunden
PutEventsRequests	Die Anzahl der erfolgreichen PutEvents API-Aufrufe. Dimension: DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents4xxErrors	Die Anzahl der PutEvents API-Aufrufe, die einen 4xx-HTTP-Antwortcode zurückgeben. Dimension: DatasetGroupArn, DatasetArn, EventTrackerArn
PutEvents5xxErrors	Die Anzahl der PutEvents API-Aufrufe, die einen 5xx-HTTP-Antwortcode zurückgeben. Dimension: DatasetGroupArn, DatasetArn, EventTrackerArn

Metrik	Beschreibung
PutEventLatency	<p>Die erforderliche Zeit für den Abschluss des PutEvents API-Aufrufs (ohne 4xx und 5xx-Fehler).</p> <p>Dimension: DatasetGroupArn, DatasetArn, EventTrackerArn</p> <p>Einheit: Millisekunden</p>
PutItemsRequests	<p>Die Anzahl der erfolgreichen PutItems API-Aufrufe.</p> <p>Dimension: DatasetGroupArn, DatasetArn</p>
PutItems4xxErrors	<p>Die Anzahl der PutItems API-Aufrufe, die einen 4xx-HTTP-Antwortcode zurückgeben.</p> <p>Dimension: DatasetGroupArn, DatasetArn</p>
PutItems5xxErrors	<p>Die Anzahl der PutItems API-Aufrufe, die einen 5xx-HTTP-Antwortcode zurückgeben.</p> <p>Dimension: DatasetGroupArn, DatasetArn</p>
PutItemsLatency	<p>Die erforderliche Zeit für den Abschluss des PutItems API-Aufrufs (ohne 4xx und 5xx-Fehler).</p> <p>Dimension: DatasetGroupArn, DatasetArn</p> <p>Einheit: Millisekunden</p>
PutUsersRequests	<p>Die Anzahl der erfolgreichen PutUsers API-Aufrufe.</p> <p>Dimension: DatasetGroupArn, DatasetArn</p>
PutUsers4xxErrors	<p>Die Anzahl der PutUsers API-Aufrufe, die einen 4xx-HTTP-Antwortcode zurückgeben.</p> <p>Dimension: DatasetGroupArn, DatasetArn</p>

Metrik	Beschreibung
PutUsers5xxErrors	Die Anzahl der PutUsers API-Aufrufe, die einen 5xx-HTTP-Antwortcode zurückgeben. Dimension: DatasetGroupArn, DatasetArn
PutUsersLatency	Die erforderliche Zeit für den Abschluss des PutUsers API-Aufrufs (ohne 4xx und 5xx-Fehler). Dimension: DatasetGroupArn, DatasetArn Einheit: Millisekunden

Protokollieren Amazon Personalize Personalize-API-Aufrufen mit AWS CloudTrail

Amazon Personalize ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon Personalize ausgeführt wurden. CloudTrail erfasst eine Teilmenge von API-Aufrufen für Amazon Personalize als Ereignisse, einschließlich Aufrufe von der Amazon Personalize Personalize-Konsole und von Codeaufrufen an die Amazon Personalize Personalize-APIs. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Personalize. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon Personalize gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Amazon Personalize Sie Informationen in CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn unterstützte Ereignisaktivitäten in Amazon Personalize auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können

aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Amazon Personalize, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Amazon Personalize unterstützt die Protokollierung jeder Aktion (API-Operation) als Ereignis in CloudTrail Protokolldateien. Weitere Informationen finden Sie unter [Aktionen](#).

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Service gestellt wurde.

Weitere Informationen finden Sie unter dem [CloudTrail UserIdentity-Element](#).

Beispiel: Amazon Personalize Personalize-Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion,

Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag mit Aktionen für den ListDatasetGroups API-Vorgang. Beachten Sie, dass die responseElements Antwort Null ist, da es sich bei der ListDatasetGroups API-Operation um eine Aktion handelt, die den Status nicht ändert. Weitere Informationen zum Hauptteil von CloudTrail Datensätzen finden Sie unter [CloudTrail Datensatzinhalt](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "principal-id",
    "arn": "arn:aws:iam:user-arn",
    "accountId": "account-id",
    "accessKeyId": "access-key",
    "userName": "user-name"
  },
  "eventTime": "2018-11-22T02:18:03Z",
  "eventSource": "personalize.amazonaws.com",
  "eventName": "ListDatasetGroups",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "source-ip-address",
  "userAgent": "aws-cli/1.11.16 Python/2.7.11 Darwin/15.6.0 boto3/1.4.73",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "request-id",
  "eventID": "event-id",
  "eventType": "AwsApiCall",
  "recipientAccountId": "recipient-account-id"
}
```

Konformitätsprüfung für Amazon Personalize

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon Personalize im Rahmen mehrerer AWS Compliance-Programme. Zu diesen Programmen gehören SOC, PCI, HIPAA und andere.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) . Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter heruntergeladenen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen unter](#) .

Ihre Compliance-Verantwortung bei der Verwendung von Amazon Personalize richtet sich nach der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- Schnellstartanleitungen zu [Sicherheit und Compliance Schnellstartanleitungen](#) zu — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben. AWS
- [Whitepaper „Architecting for HIPAA Security and Compliance“](#) — Erfahren Sie, wie Sie damit sensible Workloads ausführen können AWS , die unter den US-amerikanischen Health Insurance Portability and Accountability Act (HIPAA) fallen.
- [AWS Ressourcen zur Einhaltung](#) von — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Developer Guide — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Resilienz in Amazon Personalize

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Amazon Personalize nutzt die AWS globale Infrastruktur für Datenstabilität. Wenn Sie eine Amazon Personalize-Ressource in einer AWS Region erstellen, verwaltet Amazon Personalize die

Widerstandsfähigkeit und Datenredundanz der Ressource in mehreren Availability Zones. Eine Liste der AWS Regionen, in denen Sie Amazon Personalize-Ressourcen erstellen können, finden Sie unter [AWS Regionen und Endpunkte](#) in der Allgemeinen Referenz zu Amazon Web Services. Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Infrastruktursicherheit in Amazon Personalize

Als verwalteter Service ist Amazon Personalize durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Personalize zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Amazon VPC-Endpunkte personalisieren und verbinden ()AWS PrivateLink

Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten Ihrer AWS Ressourcen verwenden, können Sie eine private Verbindung zwischen Ihrer VPC und Amazon Personalize herstellen. Diese Verbindung ermöglicht Amazon Personalize, mit Ihren Ressourcen auf Ihrer VPC zu kommunizieren, ohne das öffentliche Internet nutzen zu müssen.

Amazon VPC ist eine AWS-Service , die Sie verwenden, um AWS Ressourcen in einer von Ihnen definierten Virtual Private Cloud (VPC) oder einem virtuellen Netzwerk zu starten. Mit einer VPC

haben Sie die Kontrolle über Ihre Netzwerkeinstellungen, wie IP-Adressbereich, Subnetze, Routing-Tabellen und Netzwerk-Gateways. Bei VPC-Endpunkten übernimmt das AWS Netzwerk das Routing zwischen Ihrer VPC und AWS-Services

Um Ihre VPC mit Amazon Personalize zu verbinden, definieren Sie einen VPC-Schnittstellen-Endpunkt für Amazon Personalize. Ein Schnittstellenendpunkt ist eine elastic network interface mit einer privaten IP-Adresse, die als Einstiegspunkt für Datenverkehr dient, der an einen unterstützten AWS-Service Empfänger gerichtet ist. Der Endpunkt bietet zuverlässige, skalierbare Konnektivität zu Amazon Personalize. Es erfordert kein Internet-Gateway, keine Network Address Translation (NAT) - Instance oder eine VPN-Verbindung. Weitere Informationen finden Sie unter [Was ist Amazon VPC](#) im Benutzerhandbuch zu Amazon VPC.

Schnittstellen-VPC-Endpunkte werden aktiviert durch AWS PrivateLink. Diese AWS Technologie ermöglicht die private Kommunikation zwischen Personen AWS-Services mithilfe einer elastic network interface mit privaten IP-Adressen.

Note

Alle Amazon Personalize Federal Information Processing Standard (FIPS) -Endgeräte werden unterstützt von AWS PrivateLink

Überlegungen zu Amazon Personalize VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon Personalize einrichten, sollten Sie die [Eigenschaften und Einschränkungen der Schnittstellen-Endpunkte](#) im Amazon VPC-Benutzerhandbuch lesen.

Amazon Personalize unterstützt Aufrufe all seiner API-Aktionen von Ihrer VPC aus.

Erstellen eines VPC-Schnittstellen-Endpunkts für Amazon Personalize

Sie können einen VPC-Endpunkt für den Amazon Personalize Personalize-Service entweder mit der Amazon VPC-Konsole oder mit () erstellen. AWS Command Line Interface AWS CLI. Weitere Informationen finden Sie unter [Erstellung eines Schnittstellenendpunkts](#) im Benutzerhandbuch für Amazon VPC.

Um einen VPC-Endpunkt für Amazon Personalize zu erstellen, wählen Sie eine der folgenden Optionen für den Service:

- `com.amazonaws.region .personalisieren`
- `com.amazonaws.region .personalize-events`
- `com.amazonaws.region .personalize-runtime`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an Amazon Personalize stellen, indem Sie den Standard-DNS-Namen für die Region verwenden, `personalize.us-east-1.amazonaws.com` zum Beispiel.

Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunktrichtlinie für Amazon Personalize

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf Amazon Personalize steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie, die alle Amazon Personalize Personalize-Aktionen und PassRole-Aktionen zulässt

Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie Zugriff auf alle Amazon Personalize Personalize-Aktionen und PassRole-Aktionen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "personalize:*",
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    }  
  ]  
}
```

Beispiel: VPC-Endpunktrichtlinie, die Amazon Personalize Personalize-Aktionen ListDatasets ermöglicht

Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie Zugriff auf die aufgelisteten Amazon Personalize ListDatasets Personalize-Aktionen.

```
{  
  "Statement": [  
    {  
      "Principal": "*",  
      "Effect": "Allow",  
      "Action": [  
        "personalize:ListDatasets"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Endpunkte und Kontingente von Amazon Personalize

Die folgenden Abschnitte enthalten Informationen zu den Richtlinien, Kontingenten und Endpunkten von Amazon Personalize. Für anpassbare Kontingente können Sie über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen. Weitere Informationen finden Sie unter [Beantragen einer Kontingenterhöhung](#).

Themen

- [Amazon Personalize Endpunkte und Regionen](#)
- [-Compliance](#)
- [Service Quotas](#)
- [Beantragen einer Kontingenterhöhung](#)

Amazon Personalize Endpunkte und Regionen

Eine Liste der Amazon Personalize Personalize-Endpunkte nach Regionen finden Sie unter [AWSRegionen und Endpunkte](#) in der Amazon Web Services General Reference.

-Compliance

Informationen zu den Compliance-Programmen von Amazon Personalize finden Sie unter [AWSAWSCompliance, Compliance-Programme](#) und [AWSServices nach Compliance-Programmen](#).

Service Quotas

Ihr AWS Konto hat die folgenden Kontingente für Amazon Personalize.

Ressource	Kontingent
Item interactions	
Minimum number of unique item interactions required to create a solution version or recommender. For a custom solution, you must have this many records after any filtering by event type or event value before training.	1000

Ressource	Kontingent
Maximum number of item interactions that are considered by a model during training.	500 million (adjustable)
Maximum number of distinct event types combined with total number of optional metadata columns in an Item interactions dataset.	10
Maximum number of metadata columns, excluding reserved fields, in an Item interactions dataset.	5
Maximum number of characters for categorical data and impression values.	1000
Maximum amount of bulk item interactions data per dataset import job with FULL import mode.	100 GB (increases to 1TB with any increase to Interaktionen zwischen Artikeln, die von einem Modell berücksichtigt wurden)
Maximum amount of bulk item interactions data per dataset import job with INCREMENTAL import mode.	1 GB
Minimum number of item interactions records per dataset import job with FULL or INCREMENTAL import mode.	1000
Users	
Minimum number of unique users in item interactions data, with at minimum 2 item interactions each, required to create a domain recommender or custom solution version.	25

Ressource	Kontingent
Minimum percentage of total users that must have at minimum 2 item interactions or more before you can create a domain recommender or custom solution version.	1 percent
Maximum number of metadata fields for a Users dataset.	25
Maximum number of characters for USER_ID data values.	256
Maximum number of characters for categorical data values.	1000 characters
Maximum amount of bulk user data per dataset import job with FULL import mode.	100 GB
Maximum amount of bulk user data per dataset import job with INCREMENTAL import mode.	1 GB
Items	
Maximum number of items that are considered by a model during training and generating recommendations.	750,000
Maximum number of metadata fields for an Items dataset.	100
Maximum number of characters for ITEM_ID data values.	256
Maximum number of characters for categorical data values.	1000 characters
Maximum number of textual fields for an Items dataset.	1

Ressource	Kontingent
Maximum number of characters for textual data values for Chinese and Japanese languages.	7,000 characters
Maximum number of characters for textual data values for all other languages.	20,000 characters
Maximum amount of bulk items data per dataset import job with BULK import mode.	100 GB
Maximum amount of bulk item data per dataset import job with INCREMENTAL import mode.	1 GB
Actions	
Maximum number of actions that are considered by a model during training and generating recommendations.	1000
Maximum number of metadata fields for an Actions dataset.	10
Maximum number of characters for ACTION_ID data values.	256
Maximum number of characters for categorical data values.	1000 characters
Maximum amount of bulk actions data per dataset import job with BULK import mode.	100 GB
Maximum amount of bulk actions data per dataset import job with INCREMENTAL import mode.	1 GB
Action interactions	
Maximum number of action interactions that are considered by a model during training.	500 million

Ressource	Kontingent
Maximum number of metadata columns, excluding reserved fields, in a Action interactions dataset.	5
Maximum amount of bulk interactions data per dataset import job with FULL import mode.	100 GB (increases to 1TB with any increase to Von einem Modell berücksichtigte Interaktionen zwischen Aktionselementen)
Maximum amount of bulk interactions data per dataset import job with INCREMENTAL import mode.	1 GB
Individual record import APIs	
Maximum rate of PutEvents requests per dataset group.	1000/second
Maximum number of events in a PutEvents call.	10
Maximum size of an event.	10 KB
Maximum rate of PutActionInteractions requests per dataset group.	1000/second
Maximum number of action interaction events in a PutActionInteractions call.	10
Maximum size of an action interaction event.	10 KB
Maximum rate of PutItems requests per dataset group.	10/second
Maximum number of items in a PutItems call.	10
Maximum rate of PutUsers requests per dataset group.	10/second

Ressource	Kontingent
Maximum number of users in a PutUsers call.	10
Maximum rate of PutActions requests per dataset group.	10/second
Maximum number of users in a PutActions call.	10
Legacy recipes	
Maximum amount of combined data for Users and Items datasets for HRNN-metadata and HRNN-Coldstart recipes.	5 GB
Maximum number of cold start items the HRNN-Coldstart recipe supports to train a model (create a solution version).	80000
Minimum number of cold start items the HRNN-Coldstart recipe requires to train a model (create a solution version).	100
Filters	
Total number of filters per dataset group.	10
Maximum number of distinct dataset fields for a filter.	5
Total number of distinct dataset fields across all filters in a dataset group.	10
Maximum number of item interactions per user per event type considered by a filter.	100 interactions (adjustable)
Maximum number of action interactions per user per event type considered by a filter.	300 action interactions (adjustable)

Ressource	Kontingent
GetRecommendations / GetPersonalizedRanking / GetActionRecommendations requests	
Maximum transaction rate for GetRecommendations , GetActionRecommendations and GetPersonalizedRanking requests.	2500/sec
Maximum number of GetRecommendations requests per second per campaign.	500/sec
Maximum number of GetActionRecommendations requests per second per campaign.	500/sec
Maximum number of GetPersonalizedRanking requests per second per campaign.	500/sec.
Maximum number of metadata columns per GetRecommendations or GetPersonalizedRanking request.	10
Maximum number of recommendation results for a GetRecommendation request without metadata.	500
Maximum number of recommendation results for a GetRecommendation request with metadata.	50
Maximum number of items for ranking in a GetPersonalizedRanking request without metadata.	500
Maximum number of items for ranking in a GetPersonalizedRanking request with metadata.	50
Metric attribution quotas	

Ressource	Kontingent
Maximum number of metrics for a metric attribution	10
Maximum number of unique event attribution sources	100
Batch inference jobs	
Maximum number of input files for a batch inference job.	1000
Maximum size of batch inference job input.	1 GB
Maximum number of records per input file for a batch inference job without themes.	50 million
Maximum number of records per input file for a batch inference job with themes.	100
Batch segment jobs	
Maximum number of queries per input file for Item-Affinity recipe.	500
Maximum number of queries per input file for Item-Attribute-Affinity recipe.	10
Maximum number of users per segment	5 million

Ihr AWS Konto hat die folgenden Kontingente für jede Region.

Ressource	Kontingent
Gesamtzahl der aktiven Schemata.	500
Gesamtzahl aktiver Datasets-Gruppen.	5 (einstellbar)
Gesamtzahl der ausstehenden oder laufenden Datenimportaufträge.	5

Ressource	Kontingent
Gesamtzahl der ausstehenden oder laufenden Batch-Inferenzanfragen.	5 (einstellbar)
Gesamtzahl der ausstehenden oder in Bearbeitung befindlichen Batchsegmentanfragen.	5
Gesamtzahl der ausstehenden oder in Bearbeitung befindlichen Lösungsvarianten.	20 (einstellbar)

Jede Datensatzgruppe hat die folgenden Kontingente.

Ressource	Kontingent
Gesamtzahl der aktiven Lösungen.	10 (einstellbar)
Gesamtzahl der aktiven Kampagnen.	5 (einstellbar)
Gesamtzahl der Empfehlungen.	5
Gesamtzahl der Filter.	10 (einstellbar)
Gesamtzahl der unterschiedlichen Datensatzfelder in allen Filtern.	10

Beantragen einer Kontingenterhöhung

Für anpassbare Kontingente können Sie über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen. Die folgenden Amazon Personalize Personalize-Kontingente sind einstellbar:

- Maximale Anzahl von Artikelinteraktionen, die von einem Modell während des Trainings berücksichtigt werden.
- Aktive Kampagnen pro Datensatzgruppe
- Aktive Datensatzgruppen
- Aktive Filter pro Datensatzgruppe
- Aktive Lösungen pro Datensatzgruppe

- Datenmenge pro inkrementellem Import
- Maximale Anzahl von Artikelinteraktionen pro Benutzer und Ereignistyp, die von einem Filter berücksichtigt werden
- Gesamtzahl der ausstehenden oder laufenden Batch-Inferenzanfragen
- Gesamtzahl der ausstehenden oder in Bearbeitung befindlichen Lösungsversionen
- Maximale Anzahl von PutEvents PutActionInteraction Anfragen

Um eine Kontingenterhöhung zu beantragen, verwenden Sie die [Service Quotas Quota-Konsole](#) und folgen Sie den Schritten im Abschnitt [Eine Kontingenterhöhung beantragen](#) im Service Quotas Quota-Benutzerhandbuch.

API-Referenz

Dieser Abschnitt enthält die Dokumentation für die Amazon Personalize Personalize-API-Operationen. Eine Liste der Amazon Personalize Personalize-Endpunkte nach Regionen finden Sie in der [AWS Allgemeinen Referenz unter Regionen und Endpunkte](#).AWS

Themen

- [Aktionen](#)
- [Datentypen](#)
- [Häufige Fehler](#)
- [Geläufige Parameter](#)

Aktionen

Die folgenden Aktionen werden von Amazon Personalize unterstützt:

- [CreateBatchInferenceJob](#)
- [CreateBatchSegmentJob](#)
- [CreateCampaign](#)
- [CreateDataset](#)
- [CreateDatasetExportJob](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateEventTracker](#)
- [CreateFilter](#)
- [CreateMetricAttribution](#)
- [CreateRecommender](#)
- [CreateSchema](#)
- [CreateSolution](#)
- [CreateSolutionVersion](#)
- [DeleteCampaign](#)
- [DeleteDataset](#)

- [DeleteDatasetGroup](#)
- [DeleteEventTracker](#)
- [DeleteFilter](#)
- [DeleteMetricAttribution](#)
- [DeleteRecommender](#)
- [DeleteSchema](#)
- [DeleteSolution](#)
- [DescribeAlgorithm](#)
- [DescribeBatchInferenceJob](#)
- [DescribeBatchSegmentJob](#)
- [DescribeCampaign](#)
- [DescribeDataset](#)
- [DescribeDatasetExportJob](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeEventTracker](#)
- [DescribeFeatureTransformation](#)
- [DescribeFilter](#)
- [DescribeMetricAttribution](#)
- [DescribeRecipe](#)
- [DescribeRecommender](#)
- [DescribeSchema](#)
- [DescribeSolution](#)
- [DescribeSolutionVersion](#)
- [GetSolutionMetrics](#)
- [ListBatchInferenceJobs](#)
- [ListBatchSegmentJobs](#)
- [ListCampaigns](#)
- [ListDatasetExportJobs](#)
- [ListDatasetGroups](#)

- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListEventTrackers](#)
- [ListFilters](#)
- [ListMetricAttributionMetrics](#)
- [ListMetricAttributions](#)
- [ListRecipes](#)
- [ListRecommenders](#)
- [ListSchemas](#)
- [ListSolutions](#)
- [ListSolutionVersions](#)
- [ListTagsForResource](#)
- [StartRecommender](#)
- [StopRecommender](#)
- [StopSolutionVersionCreation](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCampaign](#)
- [UpdateDataset](#)
- [UpdateMetricAttribution](#)
- [UpdateRecommender](#)

Die folgenden Aktionen werden von Amazon Personalize Events unterstützt:

- [PutActionInteractions](#)
- [PutActions](#)
- [PutEvents](#)
- [PutItems](#)
- [PutUsers](#)

Die folgenden Aktionen werden von Amazon Personalize Runtime unterstützt:

- [GetActionRecommendations](#)
- [GetPersonalizedRanking](#)
- [GetRecommendations](#)

Amazon Personalize

Die folgenden Aktionen werden von Amazon Personalize unterstützt:

- [CreateBatchInferenceJob](#)
- [CreateBatchSegmentJob](#)
- [CreateCampaign](#)
- [CreateDataset](#)
- [CreateDatasetExportJob](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateEventTracker](#)
- [CreateFilter](#)
- [CreateMetricAttribution](#)
- [CreateRecommender](#)
- [CreateSchema](#)
- [CreateSolution](#)
- [CreateSolutionVersion](#)
- [DeleteCampaign](#)
- [DeleteDataset](#)
- [DeleteDatasetGroup](#)
- [DeleteEventTracker](#)
- [DeleteFilter](#)
- [DeleteMetricAttribution](#)
- [DeleteRecommender](#)
- [DeleteSchema](#)
- [DeleteSolution](#)
- [DescribeAlgorithm](#)

- [DescribeBatchInferenceJob](#)
- [DescribeBatchSegmentJob](#)
- [DescribeCampaign](#)
- [DescribeDataset](#)
- [DescribeDatasetExportJob](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeEventTracker](#)
- [DescribeFeatureTransformation](#)
- [DescribeFilter](#)
- [DescribeMetricAttribution](#)
- [DescribeRecipe](#)
- [DescribeRecommender](#)
- [DescribeSchema](#)
- [DescribeSolution](#)
- [DescribeSolutionVersion](#)
- [GetSolutionMetrics](#)
- [ListBatchInferenceJobs](#)
- [ListBatchSegmentJobs](#)
- [ListCampaigns](#)
- [ListDatasetExportJobs](#)
- [ListDatasetGroups](#)
- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListEventTrackers](#)
- [ListFilters](#)
- [ListMetricAttributionMetrics](#)
- [ListMetricAttributions](#)
- [ListRecipes](#)
- [ListRecommenders](#)

- [ListSchemas](#)
- [ListSolutions](#)
- [ListSolutionVersions](#)
- [ListTagsForResource](#)
- [StartRecommender](#)
- [StopRecommender](#)
- [StopSolutionVersionCreation](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCampaign](#)
- [UpdateDataset](#)
- [UpdateMetricAttribution](#)
- [UpdateRecommender](#)

CreateBatchInferenceJob

Service: Amazon Personalize

Generiert Batch-Empfehlungen auf der Grundlage einer Liste von Artikeln oder Benutzern, die in Amazon S3 gespeichert sind, und exportiert die Empfehlungen in einen Amazon S3 S3-Bucket.

Um Batch-Empfehlungen zu generieren, geben Sie den ARN einer Lösungsversion und einen Amazon S3 S3-URI für die Eingabe- und Ausgabedaten an. Für Benutzerpersonalisierung, beliebte Artikel und personalisierte Rankinglösungen generiert der Batch-Inferenz-Job eine Liste mit empfohlenen Elementen für jede Benutzer-ID in der Eingabedatei. Bei Lösungen für verwandte Artikel generiert der Job eine Liste mit empfohlenen Elementen für jede Element-ID in der Eingabedatei.

Weitere Informationen finden Sie unter [Einen Batch-Inferenzjob erstellen](#).

Wenn Sie das Rezept für ähnliche Artikel verwenden, kann Amazon Personalize den Batch-Empfehlungen beschreibende Themen hinzufügen. Um Themen zu generieren, setzen Sie den Auftragsmodus auf THEME_GENERATION und geben Sie den Namen des Felds an, das Artikelnamen in den Eingabedaten enthält.

Weitere Informationen zum Generieren von Themes finden Sie unter [Batch-Empfehlungen mit Themes aus dem Content Generator](#).

Mit den Rezepten „Trending-Now“ oder „Next-Best-Action“ können Sie keine Batch-Empfehlungen erhalten.

Anforderungssyntax

```
{
  "batchInferenceJobConfig": {
    "itemExplorationConfig": {
      "string" : "string"
    }
  },
  "batchInferenceJobMode": "string",
  "filterArn": "string",
  "jobInput": {
    "s3DataSource": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "jobName": "string",
```

```
"jobOutput": {
  "s3DataDestination": {
    "kmsKeyArn": "string",
    "path": "string"
  }
},
"numResults": number,
"roleArn": "string",
"solutionVersionArn": "string",
"tags": [
  {
    "tagKey": "string",
    "tagValue": "string"
  }
],
"themeGenerationConfig": {
  "fieldsForThemeGeneration": {
    "itemName": "string"
  }
}
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[batchInferenceJobConfig](#)

Die Konfigurationsdetails eines Batch-Inferenzjobs.

Typ: [BatchInferenceJobConfig](#) Objekt

Erforderlich: Nein

[batchInferenceJobMode](#)

Der Modus des Batch-Inferenzjobs. Um beschreibende Themen für Gruppen ähnlicher Elemente zu generieren, stellen Sie den Jobmodus auf ein. `THEME_GENERATION` Wenn Sie keine Themen generieren möchten, verwenden Sie die Standardeinstellung `BATCH_INFERENCE`.

Wenn Sie Batch-Empfehlungen mit Themes erhalten, fallen zusätzliche Kosten an. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Preise](#).

Typ: Zeichenfolge

Zulässige Werte: BATCH_INFERENCE | THEME_GENERATION

Erforderlich: Nein

filterArn

Der ARN des Filters, der auf den Batch-Inferenzjob angewendet werden soll. Weitere Informationen zur Verwendung von Filtern finden Sie unter [Empfehlungen zum Filtern von Batches](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

jobInput

Der Amazon S3 S3-Pfad, der zur Eingabedatei führt, auf der Ihre Empfehlungen basieren sollen. Das Eingabematerial muss im JSON-Format vorliegen.

Typ: [BatchInferenceJobInput](#) Objekt

Erforderlich: Ja

jobName

Der Name des zu erstellenden Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

jobOutput

Der Pfad zum Amazon S3 S3-Bucket, in dem die Ausgabe des Jobs gespeichert wird.

Typ: [BatchInferenceJobOutput](#) Objekt

Erforderlich: Ja

numResults

Die Anzahl der abzurufenden Empfehlungen.

Typ: Ganzzahl

Erforderlich: Nein

roleArn

Der ARN der Amazon Identity and Access Management-Rolle, die über Lese- und Schreibberechtigungen für Ihre Eingabe- bzw. Ausgabe-Amazon S3-Buckets verfügt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\]+`

Erforderlich: Ja

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion, die zur Generierung der Batch-Inferenzempfehlungen verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Batch-Inferenzjob angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

[themeGenerationConfig](#)

Geben Sie für Jobs zur Themengenerierung den Namen der Spalte in Ihrem Artikeldatensatz an, die die Namen der einzelnen Elemente enthält.

Typ: [ThemeGenerationConfig](#) Objekt

Erforderlich: Nein

Antwortsyntax

```
{  
  "batchInferenceJobArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[batchInferenceJobArn](#)

Der ARN des Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:(\[a-z\d-\]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateBatchSegmentJob

Service: Amazon Personalize

Erstellt einen Batch-Segment-Job. Der Vorgang kann bis zu 50 Millionen Datensätze verarbeiten und die Eingabedatei muss im JSON-Format vorliegen. Weitere Informationen finden Sie unter [Batch-Empfehlungen und Benutzersegmente abrufen](#).

Anforderungssyntax

```
{
  "filterArn": "string",
  "jobInput": {
    "s3DataSource": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "jobName": "string",
  "jobOutput": {
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "numResults": number,
  "roleArn": "string",
  "solutionVersionArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

filterArn

Der ARN des Filters, der auf den Batch-Segment-Job angewendet werden soll. Weitere Informationen zur Verwendung von Filtern finden Sie unter [Empfehlungen zum Filtern von Batches](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

jobInput

Der Amazon S3 S3-Pfad für die Eingabedaten, die zur Generierung des Batch-Segment-Jobs verwendet wurden.

Typ: [BatchSegmentJobInput](#) Objekt

Erforderlich: Ja

jobName

Der Name des zu erstellenden Batch-Segment-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

jobOutput

Der Amazon S3 S3-Pfad für den Bucket, in dem die Ausgabe des Jobs gespeichert wird.

Typ: [BatchSegmentJobOutput](#) Objekt

Erforderlich: Ja

numResults

Die Anzahl der prognostizierten Benutzer, die durch den Batch-Segment-Job für jede Zeile von Eingabedaten generiert wurden. Die maximale Anzahl von Benutzern pro Segment beträgt 5 Millionen.

Typ: Ganzzahl

Erforderlich: Nein

roleArn

Der ARN der Amazon Identity and Access Management-Rolle, die über Lese- und Schreibberechtigungen für Ihre Eingabe- bzw. Ausgabe-Amazon S3-Buckets verfügt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Erforderlich: Ja

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion, die der Batch-Segment-Job zur Generierung von Batchsegmenten verwenden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Batch-Segment-Job angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
```

```
"batchSegmentJobArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

batchSegmentJobArn

Der ARN des Batch-Segment-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateCampaign

Service: Amazon Personalize

Erstellt eine Kampagne, die eine Lösungsversion bereitstellt. Wenn ein Kunde die [GetPersonalizedRanking](#) APIs [GetRecommendations](#) und aufruft, wird in der Anfrage eine Kampagne angegeben.

Mindestens bereitgestelltes TPS und automatische Skalierung

Important

Ein Hoch erhöht `minProvisionedTPS` Ihre Kosten. Wir empfehlen, mit 1 für zu beginnen `minProvisionedTPS` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minProvisionedTPS` diese bei Bedarf.

Wenn Sie eine Amazon Personalize Personalize-Kampagne erstellen, können Sie die Mindestanzahl an bereitgestellten Transaktionen pro Sekunde (`minProvisionedTPS`) für die Kampagne angeben. Dies ist der grundlegende Transaktionsdurchsatz für die von Amazon Personalize bereitgestellte Kampagne. Es legt die Mindestabrechnungsgebühr für die Kampagne fest, solange sie aktiv ist. Eine Transaktion ist eine einzelne `GetRecommendations` oder eine `GetPersonalizedRanking` Anfrage. Die Standardeinstellung `minProvisionedTPS` ist 1.

Steigt Ihr TPS über den Wert hinaus `minProvisionedTPS`, skaliert Amazon Personalize die bereitgestellte Kapazität automatisch nach oben und unten, aber niemals darunter. `minProvisionedTPS` Bei der Erhöhung der Kapazität kommt es zu einer kurzen Zeitverzögerung, die zum Verlust von Transaktionen führen kann. Wenn Ihr Traffic abnimmt, kehrt die Kapazität auf die zurück `minProvisionedTPS`.

Ihnen wird das bereitgestellte Mindest-TPS oder, falls Ihre Anfragen das TPS überschreiten `minProvisionedTPS`, das tatsächliche TPS in Rechnung gestellt. Der tatsächliche TPS ist die Gesamtzahl der von Ihnen gestellten Empfehlungsanfragen. Wir empfehlen, mit einem niedrigen Wert zu beginnen `minProvisionedTPS`, Ihre Nutzung anhand von CloudWatch Amazon-Metriken zu verfolgen und den Wert dann bei `minProvisionedTPS` Bedarf zu erhöhen.


Weitere Informationen zu den Kampagnenkosten finden Sie unter [Amazon Personalize Personalize-Preise](#).

Status

Eine Kampagne kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Rufen Sie an, um den Kampagnenstatus zu erfahren [DescribeCampaign](#).

 Note

Warten Sie, bis status die Kampagne gestartet ist, ACTIVE bevor Sie die Kampagne um Empfehlungen bitten.

Verbundene APIs

- [ListCampaigns](#)
- [DescribeCampaign](#)
- [UpdateCampaign](#)
- [DeleteCampaign](#)

Anforderungssyntax

```
{
  "campaignConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    }
  },
  "minProvisionedTPS": number,
  "name": "string",
  "solutionVersionArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```


Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[campaignConfig](#)

Die Konfigurationsdetails einer Kampagne.

Typ: [CampaignConfig](#) Objekt

Erforderlich: Nein

[minProvisionedTPS](#)

Gibt die angeforderte Mindestanzahl bereitgestellter Transaktionen (Empfehlungen) pro Sekunde an, die Amazon Personalize unterstützt. Ein Hoch erhöht `minProvisionedTPS` Ihre Rechnung. Wir empfehlen, mit 1 für zu beginnen `minProvisionedTPS` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minProvisionedTPS` diese bei Bedarf.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Nein

[name](#)

Ein Name für die neue Kampagne. Der Kampagnenname muss in Ihrem Konto eindeutig sein.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

[solutionVersionArn](#)

Den Amazon-Ressourcennamen (ARN) der Lösungsversion, die bereitgestellt werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

[tags](#)

Eine Liste von [Tags](#), die auf die Kampagne angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{  
  "campaignArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[campaignArn](#)

Der Amazon-Ressourcennamen (ARN) der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)

- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateDataset

Service: Amazon Personalize

Erstellt einen leeren Datensatz und fügt sie der angegebenen Datensatz-Gruppe hinzu. Wird verwendet [CreateDatasetImportJob](#), um Ihre Trainingsdaten in einen Datensatz zu importieren.

Es gibt 5 Arten von Datensätzen:

- Interaktionen mit Artikeln
- Items
- Benutzer
- Interaktionen mit Aktionen
- Aktionen

Jeder Datensatz-Typ besitzt ein zugehöriges Schema mit erforderlichen Feldtypen. Nur der `Item interactions`-Datensatz wird benötigt, um ein Modell zu trainieren (auch als Erstellen einer Lösung bezeichnet).

Ein Datensatz kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Rufen Sie an, um den Status des Datensatzes abzurufen [DescribeDataset](#).

Verbundene APIs

- [CreateDatasetGroup](#)
- [ListDatasets](#)
- [DescribeDataset](#)
- [DeleteDataset](#)

Anforderungssyntax

```
{  
  "datasetGroupArn": "string",  
  "datasetType": "string",
```

```
"name": "string",
"schemaArn": "string",
"tags": [
  {
    "tagKey": "string",
    "tagValue": "string"
  }
]
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatzgruppe, zu der der Datensatz hinzugefügt werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

datasetType

Der Typ des Datensatzes.

Einer der folgenden Werte (ohne Berücksichtigung von Groß- und Kleinschreibung):

- Interaktionen
- Items
- Benutzer
- Aktionen
- Action_Interactions

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

name

Der Name für den Datensatz.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

schemaArn

Der ARN des Schemas, das dem Datensatz zugeordnet werden soll. Das Schema definiert die Datensatzfelder.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Datensatz angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

datasetArn

Der ARN des Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateDatasetExportJob

Service: Amazon Personalize

Erstellt einen Job, der Daten aus Ihrem Datensatz in einen Amazon S3 S3-Bucket exportiert. Damit Amazon Personalize die Trainingsdaten exportieren kann, müssen Sie eine serviceverknüpfte IAM-Rolle angeben, die Amazon Personalize PutObject Personalize-Berechtigungen für Ihren Amazon S3 S3-Bucket erteilt. Weitere Informationen finden Sie unter [Exportieren eines Datensatzes](#) im Amazon Personalize Personalize-Entwicklerhandbuch.

Status

Ein Datensatz-Exportauftrag kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Um den Status des Exportjobs abzurufen [DescribeDatasetExportJob](#), rufen Sie den Amazon-Ressourcennamen (ARN) des Datensatz-Exportjobs auf und geben Sie ihn an. Der Datensatz-Export ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Wenn der Status CREATE FAILED lautet, enthält die Antwort einen `failureReason` Schlüssel, der beschreibt, warum der Job fehlgeschlagen ist.

Anforderungssyntax

```
{
  "datasetArn": "string",
  "ingestionMode": "string",
  "jobName": "string",
  "jobOutput": {
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Datensatzes, der die zu exportierenden Daten enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

[ingestionMode](#)

Die zu exportierenden Daten, basierend darauf, wie Sie die Daten importiert haben. Sie können wählen, ob Sie nur BULK Daten exportieren möchten, die Sie mithilfe eines Datensatz-Importjobs importiert haben, nur PUT Daten, die Sie inkrementell importiert haben (mithilfe der Konsole PutUsers und der PutItems Operationen), oder ALL für beide Typen. PutEvents Der Standardwert ist PUT.

Typ: Zeichenfolge

Zulässige Werte: BULK | PUT | ALL

Erforderlich: Nein

[jobName](#)

Der Name für den Datensatz-Exportauftrag.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

[jobOutput](#)

Der Pfad zum Amazon S3 S3-Bucket, in dem die Ausgabe des Jobs gespeichert ist.

Typ: [DatasetExportJobOutput](#) Objekt

Erforderlich: Ja

[roleArn](#)

Der Amazon-Ressourcenname (ARN) der IAM-Servicerolle, die berechtigt ist, Daten zu Ihrem Amazon S3-Ausgabe-Bucket hinzuzufügen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]):iam::\d{12}:role/?[a-zA-Z_0-9+=, .@\-_/]+`

Erforderlich: Ja

[tags](#)

Eine Liste von [Tags](#), die auf den Datensatz-Exportauftrag angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetExportJobArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetExportJobArn](#)

Der Amazon-Ressourcenname (ARN) des Datensatz-Exportjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateDatasetGroup

Service: Amazon Personalize

Erzeugt eine leere Datensatzgruppe. Eine Datensatzgruppe ist ein Container für Amazon Personalize Personalize-Ressourcen. Eine Datensatzgruppe kann höchstens drei Datensätze enthalten, einen für jeden Datensatztyp:

- Interaktionen zwischen Elementen
- Items
- Benutzer
- Aktionen
- Interaktionen mit Aktionen

Bei einer Datensatzgruppe kann es sich um eine Domain-Datensatzgruppe handeln, in der Sie eine Domain angeben und vorkonfigurierte Ressourcen wie Empfehlungen verwenden, oder um eine benutzerdefinierte Datensatzgruppe, in der Sie benutzerdefinierte Ressourcen verwenden, z. B. eine Lösung mit einer Lösungsversion, die Sie mit einer Kampagne bereitstellen. Wenn Sie mit einer Domain-Datensatzgruppe beginnen, können Sie trotzdem benutzerdefinierte Ressourcen wie Lösungen und Lösungsversionen hinzufügen, die mit Rezepten für benutzerdefinierte Anwendungsfälle trainiert und zusammen mit Kampagnen bereitgestellt werden.

Eine Datensatzgruppe kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHEND LÖSCHEN

Rufen Sie an, um den Status der Datensatzgruppe abzurufen [DescribeDatasetGroup](#). Wenn der Status CREATE FAILED lautet, enthält die Antwort einen `failureReason` Schlüssel, der beschreibt, warum die Erstellung fehlgeschlagen ist.

Note

Sie müssen warten, bis der `status` Wert der Datensatzgruppe erreicht ist, `ACTIVE` bevor Sie der Gruppe einen Datensatz hinzufügen.

Sie können einen AWS Key Management Service (KMS-)Schlüssel angeben, um die Datensätze in der Gruppe zu verschlüsseln. Wenn Sie einen KMS-Schlüssel angeben, müssen Sie auch eine AWS Identity and Access Management (IAM-) Rolle angeben, die berechtigt ist, auf den Schlüssel zuzugreifen.

APIs, für die in der Anfrage ein Dataset-Gruppen-ARN erforderlich ist

- [CreateDataset](#)
- [CreateEventTracker](#)
- [CreateSolution](#)

Verbundene APIs

- [ListDatasetGroups](#)
- [DescribeDatasetGroup](#)
- [DeleteDatasetGroup](#)

Anforderungssyntax

```
{
  "domain": "string",
  "kmsKeyArn": "string",
  "name": "string",
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

domain

Die Domäne der Datensatzgruppe. Geben Sie eine Domäne an, um eine Domain-Datensatzgruppe zu erstellen. Die von Ihnen angegebene Domäne bestimmt die

Standardschemas für Datensätze und die Anwendungsfälle, die für Empfehlungen verfügbar sind. Wenn Sie keine Domain angeben, erstellen Sie eine benutzerdefinierte Datensatzgruppe mit Lösungsversionen, die Sie mit einer Kampagne bereitstellen.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Erforderlich: Nein

[kmsKeyArn](#)

Der Amazon-Ressourcenname (ARN) eines AWS Key Management Service (KMS) -Schlüssels, der zur Verschlüsselung der Datensätze verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 2 048.

Pattern: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Erforderlich: Nein

[name](#)

Der Name für die neue Datensatzgruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

[roleArn](#)

Der ARN der AWS Identity and Access Management (IAM) -Rolle, die über Berechtigungen für den Zugriff auf den AWS Key Management Service (KMS-) Schlüssel verfügt. Die Angabe einer IAM-Rolle ist nur gültig, wenn auch ein KMS-Schlüssel angegeben wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=, .@-_/\]+`

Erforderlich: Nein

[tags](#)

Eine Liste von [Tags](#), die auf die Datensatzgruppe angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetGroupArn": "string",
  "domain": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der neuen Datensatzgruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

[domain](#)

Die Domain für die neue Domain-Datensatzgruppe.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)

- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateDatasetImportJob

Service: Amazon Personalize

Erstellt einen Job, der Trainingsdaten aus Ihrer Datenquelle (einem Amazon S3 S3-Bucket) in einen Amazon Personalize importiert. Damit Amazon Personalize die Trainingsdaten importieren kann, müssen Sie eine IAM-Servicerolle angeben, die berechtigt ist, aus der Datenquelle zu lesen, da Amazon Personalize eine Kopie Ihrer Daten erstellt und diese intern verarbeitet. Informationen zur Gewährung des Zugriffs auf Ihren Amazon S3 S3-Bucket finden Sie unter [Amazon Personalize Access to Amazon S3 S3-Ressourcen gewähren](#).

Wenn Sie bereits einen Empfehlungsgeber erstellt oder eine benutzerdefinierte Lösungsversion mit einer Kampagne bereitgestellt haben, hängt die Art und Weise, wie sich neue Sammeleinträge auf Empfehlungen auswirken, vom verwendeten Domain-Anwendungsfall oder Rezept ab. Weitere Informationen finden Sie unter [Wie neue Daten Empfehlungen in Echtzeit beeinflussen](#).

Important

Standardmäßig ersetzt ein Datensatz-Importjob alle vorhandenen Daten in dem Datensatz, den Sie in großen Mengen importiert haben. Um neue Datensätze hinzuzufügen, ohne vorhandene Daten zu ersetzen, geben Sie INCREMENTAL für den Importmodus im CreateDatasetImportJob Vorgang an.

Status

Ein Import-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Rufen Sie an, um den Status des Importjobs abzurufen [DescribeDatasetImportJob](#), und geben Sie dabei den Amazon-Ressourcennamen (ARN) des Datensatz-Importjobs an. Der Dataset-Importvorgang ist abgeschlossen, wenn der Status als AKTIV angezeigt wird. Wenn der Status CREATE FAILED lautet, enthält die Antwort einen `failureReason` Schlüssel, der beschreibt, warum der Job fehlgeschlagen ist.

Note

Das Importieren ist zeitaufwändig. Sie müssen warten, bis der Status als ACTIVE angezeigt wird, bevor Sie ein Modell mit dem Datensatz trainieren.

Verbundene APIs

- [ListDatasetImportJobs](#)
- [DescribeDatasetImportJob](#)

Anforderungssyntax

```
{
  "datasetArn": "string",
  "dataSource": {
    "dataLocation": "string"
  },
  "importMode": "string",
  "jobName": "string",
  "publishAttributionMetricsToS3": boolean,
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der ARN des Datensatzes, der die importierten Daten empfängt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

[dataSource](#)

Der Amazon-S3-Bucket mit den zu importierenden Trainingsdaten.

Typ: [DataSource](#) Objekt

Erforderlich: Ja

[importMode](#)

Geben Sie an, wie die neuen Datensätze zu einem vorhandenen Datensatz hinzugefügt werden sollen. Der Standard-Importmodus ist FULL. Wenn Sie zuvor keine Massendatensätze in den Datensatz importiert haben, können Sie nur Folgendes angeben FULL.

- Geben Sie FULL an, dass alle vorhandenen Massendaten in Ihrem Datensatz überschrieben werden sollen. Daten, die Sie einzeln importiert haben, werden nicht ersetzt.
- Geben Sie INCREMENTAL an, dass die neuen Datensätze an die vorhandenen Daten in Ihrem Datensatz angehängt werden sollen. Amazon Personalize ersetzt jeden Datensatz mit derselben ID durch den neuen.

Typ: Zeichenfolge

Zulässige Werte: FULL | INCREMENTAL

Erforderlich: Nein

[jobName](#)

Der Name für den Datensatz-Importjob.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

[publishAttributionMetricsToS3](#)

Wenn Sie eine Metrik-Attribution erstellt haben, geben Sie an, ob Metriken für diesen Importauftrag in Amazon S3 veröffentlicht werden sollen

Typ: Boolesch

Erforderlich: Nein

roleArn

Der ARN der IAM-Rolle, die für das Lesen aus der Amazon-S3-Datenquelle berechtigt ist.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\]+`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Datensatz-Importjob angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetImportJobArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

datasetImportJobArn

Der ARN des Datensatz-Import-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateEventTracker

Service: Amazon Personalize

Erstellt einen Event-Tracker, den Sie verwenden, wenn Sie mithilfe der [PutEvents](#)API Ereignisdaten zu einer angegebenen Datensatzgruppe hinzufügen.

Note

Nur ein Ereignis-Tracker kann einer Dataset-Gruppe zugeordnet werden. Sie erhalten eine Fehlermeldung, wenn Sie `CreateEventTracker` mit der gleichen Dataset-Gruppe wie die eines vorhandenen Ereignis-Trackers aufrufen.

Wenn Sie einen Event-Tracker erstellen, enthält die Antwort eine Tracking-ID, die Sie als Parameter übergeben, wenn Sie die [PutEvents](#)Operation verwenden. Amazon Personalize hängt dann die Ereignisdaten an den Datensatz Artikelinteraktionen der Datensatzgruppe an, die Sie in Ihrem Event-Tracker angeben.

Der Event-Tracker kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Rufen Sie an, um den Status des Event-Trackers zu erfahren [DescribeEventTracker](#).

Note

Der Event-Tracker muss sich im Status AKTIV befinden, bevor er die Tracking-ID verwenden kann.

Verbundene APIs

- [ListEventTrackers](#)
- [DescribeEventTracker](#)
- [DeleteEventTracker](#)

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "name": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatzgruppe, die die Ereignisdaten empfängt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

name

Der Name für den Event-Tracker.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Event-Tracker angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "eventTrackerArn": "string",
  "trackingId": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[eventTrackerArn](#)

Der ARN des Event-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

[trackingId](#)

Die ID des Event-Trackers. Nehmen Sie diese ID in Anfragen an die [PutEventsAPI](#) auf.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)

- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateFilter

Service: Amazon Personalize

Erstellt einen Empfehlungsfiler. Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "filterExpression": "string",
  "name": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der ARN der Datensatzgruppe, zu der der Filter gehören wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

[filterExpression](#)

Der Filterausdruck definiert, welche Elemente in Empfehlungen aufgenommen oder ausgeschlossen werden. Der Filterausdruck muss bestimmten Formatregeln entsprechen. Informationen zur Struktur und Syntax von Filterausdrücken finden Sie unter [Filterausdrücke](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 2500.

Erforderlich: Ja

name

Der Name des zu erstellenden Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf den Filter angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "filterArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

filterArn

Der ARN des neuen Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateMetricAttribution

Service: Amazon Personalize

Erstellt eine metrische Zuordnung. Eine Metrikzuweisung erstellt Berichte über die Daten, die Sie in Amazon Personalize importieren. Je nachdem, wie Sie die Daten importiert haben, können Sie Berichte in Amazon CloudWatch oder Amazon S3 anzeigen. Weitere Informationen finden Sie unter [Messung der Wirkung von Empfehlungen](#).

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "metrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "metricsOutputConfig": {
    "roleArn": "string",
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "name": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der Zieldatensatzgruppe für die Metrikzuweisung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

metrics

Eine Liste von Metrikattributen für die metrische Zuordnung. Jedes Metrikattribut spezifiziert einen Ereignistyp, der verfolgt werden soll, und eine Funktion. Verfügbare Funktionen sind `SUM()` oder `SAMPLECOUNT()`. Geben Sie für `SUM()`-Funktionen den Datensatztyp (entweder Interaktionen oder Elemente) und die Spalte, die summiert werden soll, als Parameter an. Zum Beispiel `SUM (Items.Price)`.

Typ: Array von [MetricAttribute](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 10 Elementen.

Erforderlich: Ja

metricsOutputConfig

Die Details der Ausgabekonfiguration für die metrische Zuordnung.

Typ: [MetricAttributionOutput](#) Objekt

Erforderlich: Ja

name

Ein Name für die Metrik-Attribution.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

Antwortsyntax

```
{
  "metricAttributionArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[metricAttributionArn](#)

Der Amazon-Ressourcenname (ARN) für die neue Metrikzuweisung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateRecommender

Service: Amazon Personalize

Erstellt eine Empfehlung mit dem von Ihnen angegebenen Rezept (ein Anwendungsfall für Domain-Datensatzgruppen). Sie erstellen Empfehlungen für eine Domain-Datensatzgruppe und geben den Amazon-Ressourcennamen (ARN) des Empfehlungsgebers an, wenn Sie eine [GetRecommendations](#)Anfrage stellen.

Minimale Anzahl von Empfehlungsanfragen pro Sekunde

Important

Ein Hoch `minRecommendationRequestsPerSecond` erhöht Ihre Rechnung. Wir empfehlen, mit 1 für zu beginnen `minRecommendationRequestsPerSecond` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minRecommendationRequestsPerSecond` diese bei Bedarf.

Wenn Sie einen Empfehlungsgeber erstellen, können Sie die Mindestanzahl der Empfehlungsanfragen pro Sekunde für den Empfehlungsgeber konfigurieren. Die Mindestanzahl von Empfehlungsanfragen pro Sekunde (`minRecommendationRequestsPerSecond`) gibt den von Amazon Personalize bereitgestellten Basisdurchsatz für Empfehlungsanfragen an. Die Standardeinstellung ist `minRecommendationRequestsPerSecond . 1` Eine Empfehlungsanfrage ist ein einziger `GetRecommendations` Vorgang. Der Anforderungsdurchsatz wird in Anfragen pro Sekunde gemessen und Amazon Personalize verwendet Ihre Anfragen pro Sekunde, um Ihre Anfragen pro Stunde und den Preis für Ihre empfohlene Nutzung abzuleiten.

Wenn Ihre Anfragen pro Sekunde darüber hinaus `steigenminRecommendationRequestsPerSecond`, skaliert Amazon Personalize die bereitgestellte Kapazität automatisch nach oben und unten, aber niemals darunter. `minRecommendationRequestsPerSecond` Bei der Erhöhung der Kapazität kommt es zu einer kurzen Zeitverzögerung, die zum Verlust von Anfragen führen kann.

Ihre Rechnung ist entweder die Mindestanzahl der Anfragen pro Stunde (basierend auf `minRecommendationRequestsPerSecond`) oder die tatsächliche Anzahl der Anfragen, je nachdem, welcher Betrag höher ist. Der tatsächlich verwendete Anforderungsdurchsatz wird als durchschnittliche Anzahl von Anfragen pro Sekunde innerhalb eines Zeitfensters von einer Stunde berechnet. Wir empfehlen, mit der Standardeinstellung zu

beginnen `minRecommendationRequestsPerSecond`, Ihre Nutzung anhand von CloudWatch Amazon-Metriken zu verfolgen und diese dann bei `minRecommendationRequestsPerSecond` Bedarf zu erhöhen.

Status

Ein Empfehlungsgeber kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INAKTIV > START AUSSTEHEND > START IN_PROGRESS > AKTIV
- DELETE PENDING > DELETE IN_PROGRESS

Rufen Sie an, um den Empfehlungsstatus zu erhalten. [DescribeRecommender](#)

Note

Warten Sie, bis der `status` Status des Empfehlungsgebers erreicht ist, `ACTIVE` bevor Sie den Empfehlungsgeber um Empfehlungen bitten.

Verbundene APIs

- [ListRecommenders](#)
- [DescribeRecommender](#)
- [UpdateRecommender](#)
- [DeleteRecommender](#)

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "name": "string",
  "recipeArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
```

```
    "string" : "string"
  },
  "minRecommendationRequestsPerSecond": number,
  "trainingDataConfig": {
    "excludedDatasetColumns": {
      "string" : [ "string" ]
    }
  }
},
"tags": [
  {
    "tagKey": "string",
    "tagValue": "string"
  }
]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Zieldomänen-Datensatzgruppe für den Empfehlungsgeber.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

name

Der Name des Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

[recipeArn](#)

Der Amazon-Ressourcenname (ARN) des Rezepts, das der Empfehlungsgeber verwenden wird. Für einen Empfehlungsgeber ist ein Rezept ein Anwendungsfall für Domain-Datensatzgruppen. Nur Anwendungsfälle für Domain-Datensatzgruppen können verwendet werden, um eine Empfehlung zu erstellen. Informationen zu Anwendungsfällen finden Sie unter Anwendungsfälle für [Empfehlungen auswählen](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

[recommenderConfig](#)

Die Konfigurationsdetails des Empfehlungsgebers.

Typ: [RecommenderConfig](#) Objekt

Erforderlich: Nein

[tags](#)

Eine Liste von [Tags](#), die auf den Empfehlungsgeber angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{
  "recommenderArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateSchema

Service: Amazon Personalize

Erstellt ein Amazon-Personalize-Schema aus der angegebenen Schemazeichenfolge. Das Schema, das Sie erstellen, muss im Avro-JSON-Format vorliegen.

Amazon Personalize erkennt drei Schemavarianten. Jedes Schema ist einem Datensatztyp zugeordnet und hat eine Reihe von erforderlichen Feldern und Schlüsselwörtern. Wenn Sie ein Schema für einen Datensatz in einer Domäne-Datensatz-Gruppe erstellen, geben Sie die Domäne der Domäne-Datensatz-Gruppe an. Sie geben ein Schema an, wenn Sie aufrufen [CreateDataset](#).

Weitere Informationen zu Schemata finden Sie unter [Datensätze und Schemata](#).

Verbundene APIs

- [ListSchemas](#)
- [DescribeSchema](#)
- [DeleteSchema](#)

Anforderungssyntax

```
{
  "domain": "string",
  "name": "string",
  "schema": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[domain](#)

Die Domäne für das Schema. Wenn Sie ein Schema für einen Datensatz in einer Domain-Datensatzgruppe erstellen, geben Sie die Domain an, die Sie bei der Erstellung der Domain-Datensatzgruppe ausgewählt haben.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Erforderlich: Nein

name

Der Name des Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

schema

Ein Schema im Avro-JSON-Format.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 20000.

Erforderlich: Ja

Antwortsyntax

```
{  
  "schemaArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

schemaArn

Der Amazon-Ressourcenname (ARN) des erstellten Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: arn:([a-z\d-]+):personalize:.*:.*:.*+

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateSolution

Service: Amazon Personalize

Erzeugt die Konfiguration für das Training eines Modells. Ein trainiertes Modell wird als Lösungsversion bezeichnet. Nachdem die Konfiguration erstellt wurde, trainieren Sie das Modell (erstellen Sie eine Lösungsversion), indem Sie die [CreateSolutionVersion](#) Operation aufrufen. Bei jedem Aufruf `CreateSolutionVersion` wird eine neue Version der Lösung erstellt.

Nachdem Sie eine Lösungsversion erstellt haben, überprüfen Sie deren Richtigkeit, indem Sie anrufen [GetSolutionMetrics](#). Wenn Sie mit der Version zufrieden sind, stellen Sie sie mithilfe von bereit [CreateCampaign](#). Die Kampagne bietet einem Kunden über die [GetRecommendationsAPI](#) Empfehlungen.

Um ein Modell zu trainieren, benötigt Amazon Personalize Trainingsdaten und ein Rezept. Die Trainingsdaten stammen aus der Datensatzgruppe, die Sie in der Anfrage angeben. Ein Rezept spezifiziert den Trainingsalgorithmus und eine Merkmalstransformation. Sie können eines der von Amazon Personalize bereitgestellten vordefinierten Rezepte angeben.

Note

Amazon Personalize unterstützt derzeit nicht die Konfiguration der Hyperparameter-Optimierung `hpoObjective` für die Lösung.

Status

Eine Lösung kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Rufen Sie an, um den Status der Lösung zu erfahren [DescribeSolution](#). Warten Sie, bis der Status als AKTIV angezeigt wird, bevor Sie anrufen `CreateSolutionVersion`.

Verbundene APIs

- [ListSolutions](#)
- [CreateSolutionVersion](#)
- [DescribeSolution](#)

- [DeleteSolution](#)
- [ListSolutionVersions](#)
- [DescribeSolutionVersion](#)

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "eventType": "string",
  "name": "string",
  "performAutoML": boolean,
  "performHPO": boolean,
  "recipeArn": "string",
  "solutionConfig": {
    "algorithmHyperParameters": {
      "string" : "string"
    },
    "autoMLConfig": {
      "metricName": "string",
      "recipeList": [ "string" ]
    },
    "eventValueThreshold": "string",
    "featureTransformationParameters": {
      "string" : "string"
    },
    "hpoConfig": {
      "algorithmHyperParameterRanges": {
        "categoricalHyperParameterRanges": [
          {
            "name": "string",
            "values": [ "string" ]
          }
        ],
        "continuousHyperParameterRanges": [
          {
            "maxValue": number,
            "minValue": number,
            "name": "string"
          }
        ],
        "integerHyperParameterRanges": [
```

```

        {
            "maxValue": number,
            "minValue": number,
            "name": "string"
        }
    ]
},
"hpoObjective": {
    "metricName": "string",
    "metricRegex": "string",
    "type": "string"
},
"hpoResourceConfig": {
    "maxNumberOfTrainingJobs": "string",
    "maxParallelTrainingJobs": "string"
}
},
"optimizationObjective": {
    "itemAttribute": "string",
    "objectiveSensitivity": "string"
},
"trainingDataConfig": {
    "excludedDatasetColumns": {
        "string" : [ "string" ]
    }
}
},
"tags": [
    {
        "tagKey": "string",
        "tagValue": "string"
    }
]
}

```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe, die die Trainingsdaten bereitstellt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

eventType

Wenn Sie mehrere Ereignistypen haben (unter Verwendung eines `EVENT_TYPE` Schemafelds), gibt dieser Parameter an, welcher Ereignistyp (z. B. „Klicken“ oder „Gefällt mir“) für das Training des Modells verwendet wird.

Wenn Sie keine angebeneventType, verwendet Amazon Personalize alle Interaktionen für das Training mit gleichem Gewicht, unabhängig vom Typ.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

name

Der Name für die Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Ja

performAutoML

Important

Wir raten von der Verwendung von automatisiertem Machine Learning ab. Ordnen Sie Ihren Anwendungsfall stattdessen den verfügbaren Amazon-Personalize-Rezepten zu. Weitere Informationen finden Sie unter [Ein Rezept auswählen](#).

Ob automatisiertes maschinelles Lernen (AutoML) durchgeführt werden soll. Der Standardwert ist `false`. In diesem Fall müssen Sie Folgendes angeben `recipeArn`.

Wenn diese Option aktiviert ist `true`, analysiert Amazon Personalize Ihre Trainingsdaten und wählt das optimale `USER_PERSONALIZATION`-Rezept und die optimalen Hyperparameter aus. In diesem Fall müssen Sie es weglassen. `recipeArn` Amazon Personalize ermittelt das optimale Rezept, indem Tests mit unterschiedlichen Werten für die Hyperparameter ausgeführt werden. AutoML verlängert den Trainingsprozess im Vergleich zur Auswahl eines bestimmten Rezepts.

Typ: Boolesch

Erforderlich: Nein

[performHPO](#)

Ob eine Hyperparameter-Optimierung (HPO) für das angegebene oder ausgewählte Rezept durchgeführt werden soll. Der Standardwert ist `false`.

Wenn Sie AutoML ausführen, ist dieser Parameter immer gültig `true` und Sie sollten ihn nicht auf `false` setzen.

Typ: Boolesch

Erforderlich: Nein

[recipeArn](#)

Der Amazon-Ressourcenname (ARN) des Rezepts, das für das Modelltraining verwendet werden soll. Dies ist erforderlich, wenn der Wert falsch `performAutoML` ist. Informationen zu den verschiedenen Amazon Personalize Personalize-Rezepten und ihren ARNs finden Sie unter [Ein Rezept auswählen](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

[solutionConfig](#)

Die Konfiguration, die mit der Lösung verwendet werden soll. Wenn auf `true` gesetzt `performAutoML` ist, wertet Amazon Personalize nur den `autoMLConfig` Abschnitt der Lösungskonfiguration aus.

Note

Amazon Personalize unterstützt derzeit nicht die Konfiguration von `hpoObjective`.

Typ: [SolutionConfig](#) Objekt

Erforderlich: Nein

tags

Eine Liste von [Tags, die auf](#) die Lösung angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

Antwortsyntax

```
{  
  "solutionArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[solutionArn](#)

Der ARN der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)

- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

CreateSolutionVersion

Service: Amazon Personalize

Trainiert oder trainiert eine aktive Lösung in einer benutzerdefinierten Datensatzgruppe.

Eine Lösung wird mithilfe der [CreateSolution](#) Operation erstellt und muss sich vor dem Aufruf `CreateSolutionVersion` im Status `ACTIVE` befinden. Bei jedem Aufruf dieser Operation wird eine neue Version der Lösung erstellt.

Status

Eine Lösungsversion kann sich in einem der folgenden Zustände befinden:

- `AUSSTEHEND ERSTELLEN`
- `IN_PROGRESS ERSTELLEN`
- `ACTIVE`
- `ERSTELLEN IST FEHLGESCHLAGEN`
- `STOPPEN ERSTELLEN`
- `CREATE HAT AUFGEHÖRT`

Rufen Sie an, um den Status der Version abzurufen [DescribeSolutionVersion](#). Warten Sie, bis der Status als `AKTIV` angezeigt wird, bevor Sie `anrufenCreateCampaign` aufrufen.

Wenn der Status `CREATE FAILED` lautet, enthält die Antwort einen `failureReason` Schlüssel, der beschreibt, warum der Job fehlgeschlagen ist.

Verbundene APIs

- [ListSolutionVersions](#)
- [DescribeSolutionVersion](#)
- [ListSolutions](#)
- [CreateSolution](#)
- [DescribeSolution](#)
- [DeleteSolution](#)

Anforderungssyntax

```
{
```

```
"name": "string",  
"solutionArn": "string",  
"tags": [  
  {  
    "tagKey": "string",  
    "tagValue": "string"  
  }  
],  
"trainingMode": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

name

Der Name der Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

solutionArn

Der Amazon-Ressourcenname (ARN) der Lösung, die die Informationen zur Trainingskonfiguration enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

tags

Eine Liste von [Tags](#), die auf die Lösungsversion angewendet werden sollen.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Nein

[trainingMode](#)

Der Umfang der Schulung, die bei der Erstellung der Lösungsversion durchgeführt werden muss. Der Standardwert ist FULL. Dadurch wird ein völlig neues Modell erstellt, das auf der Gesamtheit der Trainingsdaten aus den Datensätzen in Ihrer Datensatzgruppe basiert.

Wenn Sie die [Benutzerpersonalisierung](#) verwenden, können Sie einen Trainingsmodus von angeben. UPDATE Dadurch wird das Modell aktualisiert, sodass es neue Elemente für Empfehlungen berücksichtigt. Es handelt sich nicht um eine vollständige Umschulung. Sie sollten trotzdem wöchentlich eine vollständige Umschulung absolvieren. Wenn Sie dies angebenUPDATE, stoppt Amazon Personalize automatische Updates für die Lösungsversion. Um die Updates wieder aufzunehmen, erstellen Sie eine neue Lösung, bei der der Trainingsmodus aktiviert ist, FULL und stellen Sie sie in einer Kampagne bereit. Weitere Informationen zu automatischen Updates finden Sie unter [Automatische Updates](#).

Die UPDATE Option kann nur verwendet werden, wenn Sie bereits eine aktive Lösungsversion aus der Eingabelösung mit der FULL Option erstellt haben und die Eingabelösung mit dem [Benutzerpersonalisierungsrezept oder dem älteren HRNN-ColdStart-Rezept](#) trainiert wurde.

Typ: Zeichenfolge

Zulässige Werte: FULL | UPDATE

Erforderlich: Nein

Antwortsyntax

```
{
  "solutionVersionArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[solutionVersionArn](#)

Der ARN der neuen Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anfragen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteCampaign

Service: Amazon Personalize

Entfernt eine Kampagne, indem die Lösungsbereitstellung gelöscht wird. Die Lösung, auf der die Kampagne basiert, wird nicht gelöscht und kann bei Bedarf erneut bereitgestellt werden. Eine gelöschte Kampagne kann nicht mehr in einer [GetRecommendations](#)Anfrage angegeben werden. Informationen zum Erstellen von Kampagnen finden Sie unter [CreateCampaign](#).

Anforderungssyntax

```
{  
  "campaignArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[campaignArn](#)

Der Amazon-Ressourcenname (ARN) der zu löschenden Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteDataset

Service: Amazon Personalize

Löscht einen Datensatz. Sie können einen Datensatz nicht löschen, wenn er verknüpft `SolutionVersion` ist `DatasetImportJob` oder sich im Status `CREATE PENDING` oder `IN PROGRESS` befindet. Weitere Informationen zu Datensätzen finden Sie unter [CreateDataset](#).

Anforderungssyntax

```
{
  "datasetArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des zu löschenden Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

`InvalidInputException`

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteDatasetGroup

Service: Amazon Personalize

Löscht eine Datensatzgruppe. Bevor Sie eine Datensatzgruppe löschen, müssen Sie Folgendes löschen:

- Alle zugehörigen Ereignis-Tracker.
- Alle zugehörigen Lösungen.
- Alle Datasets in der Dataset-Gruppe.

Anforderungssyntax

```
{  
  "datasetGroupArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der ARN der zu löschenden Datensatzgruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteEventTracker

Service: Amazon Personalize

Löscht den Event-Tracker. Löscht den Datensatz nicht aus der Datensatzgruppe. Weitere Informationen zu Event-Trackern finden Sie unter [CreateEventTracker](#).

Anforderungssyntax

```
{  
  "eventTrackerArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[eventTrackerArn](#)

Der Amazon-Ressourcenname (ARN) des zu löschenden Event-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteFilter

Service: Amazon Personalize

Löscht einen Filter.

Anforderungssyntax

```
{  
  "filterArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

filterArn

Der ARN des zu löschenden Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteMetricAttribution

Service: Amazon Personalize

Löscht eine metrische Zuordnung.

Anforderungssyntax

```
{  
  "metricAttributionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

metricAttributionArn

Der Amazon-Ressourcenname (ARN) der Metrikzuweisung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteRecommender

Service: Amazon Personalize

Deaktiviert und entfernt einen Empfehlungsgeber. Ein gelöschter Empfehlungsgeber kann nicht mehr in einer Anfrage angegeben werden. [GetRecommendations](#)

Anforderungssyntax

```
{  
  "recommenderArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

recommenderArn

Der Amazon-Ressourcenname (ARN) des zu löschenden Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteSchema

Service: Amazon Personalize

Löscht ein Schema. Bevor Sie ein Schema löschen, müssen Sie alle Datensätze löschen, die auf das Schema verweisen. Weitere Informationen zu Schemata finden Sie unter [CreateSchema](#).

Anforderungssyntax

```
{
  "schemaArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[schemaArn](#)

Der Amazon-Ressourcenname (ARN) des zu löschenden Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DeleteSolution

Service: Amazon Personalize

Löscht alle Versionen einer Lösung und das `Solution` Objekt selbst. Bevor Sie eine Lösung löschen, müssen Sie alle Kampagnen löschen, die auf der Lösung basieren. Um festzustellen, welche Kampagnen die Lösung verwenden, rufen Sie an [ListCampaigns](#) und geben Sie den Amazon-Ressourcennamen (ARN) der Lösung an. Sie können eine Lösung nicht löschen, wenn sich eine verknüpfte Lösung im Status `CREATE PENDING` oder `IN PROGRESS` `SolutionVersion` befindet. Weitere Informationen zu Lösungen finden Sie unter [CreateSolution](#).

Anforderungssyntax

```
{
  "solutionArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[solutionArn](#)

Der ARN der zu löschenden Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

`InvalidInputException`

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeAlgorithm

Service: Amazon Personalize

Beschreibt den angegebenen Algorithmus.

Anforderungssyntax

```
{  
  "algorithmArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

algorithmArn

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Algorithmus.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "algorithm": {  
    "algorithmArn": "string",  
    "algorithmImage": {  
      "dockerURI": "string",  
      "name": "string"  
    },  
    "creationDateTime": number,  
    "defaultHyperParameterRanges": {  
      "categoricalHyperParameterRanges": [  
        {  
          "isTunable": boolean,  
          "name": "string",
```



```

        "values": [ "string" ]
    }
],
"continuousHyperParameterRanges": [
    {
        "isTunable": boolean,
        "maxValue": number,
        "minValue": number,
        "name": "string"
    }
],
"integerHyperParameterRanges": [
    {
        "isTunable": boolean,
        "maxValue": number,
        "minValue": number,
        "name": "string"
    }
]
},
"defaultHyperParameters": {
    "string" : "string"
},
"defaultResourceConfig": {
    "string" : "string"
},
"lastUpdatedDateTime": number,
"name": "string",
"roleArn": "string",
"trainingInputMode": "string"
}
}

```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

algorithm

Eine Liste der Eigenschaften des Algorithmus.

Typ: [Algorithm](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeBatchInferenceJob

Service: Amazon Personalize

Ruft die Eigenschaften eines Batch-Inferenzjobs ab, einschließlich Name, Amazon-Ressourcenname (ARN), Status, Eingabe- und Ausgabekonfigurationen sowie den ARN der Lösungsversion, die zur Generierung der Empfehlungen verwendet wurde.

Anforderungssyntax

```
{  
  "batchInferenceJobArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[batchInferenceJobArn](#)

Der ARN des zu beschreibenden Batch-Inferenz-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "batchInferenceJob": {  
    "batchInferenceJobArn": "string",  
    "batchInferenceJobConfig": {  
      "itemExplorationConfig": {  
        "string" : "string"  
      }  
    },  
    "batchInferenceJobMode": "string",  
    "creationDateTime": number,  
    "failureReason": "string",  
    "filterArn": "string",  
  }  
}
```

```

    "jobInput": {
      "s3DataSource": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    },
    "jobName": "string",
    "jobOutput": {
      "s3DataDestination": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    },
    "lastUpdatedDateTime": number,
    "numResults": number,
    "roleArn": "string",
    "solutionVersionArn": "string",
    "status": "string",
    "themeGenerationConfig": {
      "fieldsForThemeGeneration": {
        "itemName": "string"
      }
    }
  }
}

```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[batchInferenceJob](#)

Informationen zum angegebenen Batch-Inferenzjob.

Typ: [BatchInferenceJob](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeBatchSegmentJob

Service: Amazon Personalize

Ruft die Eigenschaften eines Batch-Segmentjobs ab, einschließlich Name, Amazon-Ressourcenname (ARN), Status, Eingabe- und Ausgabekonfigurationen und den ARN der Lösungsversion, die zum Generieren von Segmenten verwendet wurde.

Anforderungssyntax

```
{  
  "batchSegmentJobArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[batchSegmentJobArn](#)

Der ARN des zu beschreibenden Batch-Segment-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortsyntax

```
{  
  "batchSegmentJob": {  
    "batchSegmentJobArn": "string",  
    "creationDateTime": number,  
    "failureReason": "string",  
    "filterArn": "string",  
    "jobInput": {  
      "s3DataSource": {  
        "kmsKeyArn": "string",  

```

```
    "path": "string"
  }
},
"jobName": "string",
"jobOutput": {
  "s3DataDestination": {
    "kmsKeyArn": "string",
    "path": "string"
  }
},
"lastUpdatedDateTime": number,
"numResults": number,
"roleArn": "string",
"solutionVersionArn": "string",
"status": "string"
}
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[batchSegmentJob](#)

Informationen zum angegebenen Batch-Segment-Job.

Typ: [BatchSegmentJob](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeCampaign

Service: Amazon Personalize

Beschreibt die angegebene Kampagne, einschließlich ihres Status.

Eine Kampagne kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Wenn ja `statusCREATE_FAILED`, beinhaltet die Antwort den `failureReason` Schlüssel, der beschreibt, warum.

Weitere Informationen zu Kampagnen finden Sie unter [CreateCampaign](#).

Anforderungssyntax

```
{  
  "campaignArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[campaignArn](#)

Der Amazon-Ressourcennamen (ARN) der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
```

```

"campaign": {
  "campaignArn": "string",
  "campaignConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    }
  },
  "creationDateTime": number,
  "failureReason": "string",
  "lastUpdatedDateTime": number,
  "latestCampaignUpdate": {
    "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
        "string" : "string"
      }
    },
    "creationDateTime": number,
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "minProvisionedTPS": number,
    "solutionVersionArn": "string",
    "status": "string"
  },
  "minProvisionedTPS": number,
  "name": "string",
  "solutionVersionArn": "string",
  "status": "string"
}
}

```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

campaign

Die Eigenschaften der Kampagne.

Typ: [Campaign](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeDataset

Service: Amazon Personalize

Beschreibt den angegebenen Datensatz. Weitere Hinweise zu Datensätzen finden Sie unter [CreateDataset](#).

Anforderungssyntax

```
{  
  "datasetArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "dataset": {  
    "creationDateTime": number,  
    "datasetArn": "string",  
    "datasetGroupArn": "string",  
    "datasetType": "string",  
    "lastUpdatedDateTime": number,  
    "latestDatasetUpdate": {  
      "creationDateTime": number,  
      "failureReason": "string",  
      "lastUpdatedDateTime": number,  
      "schemaArn": "string",  
    }  
  }  
}
```

```
    "status": "string"  
  },  
  "name": "string",  
  "schemaArn": "string",  
  "status": "string",  
  "trackingId": "string"  
}  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

dataset

Eine Liste der Eigenschaften des Datensatzes.

Typ: [Dataset](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeDatasetExportJob

Service: Amazon Personalize

Beschreibt den von erstellten Datensatz-Exportauftrag [CreateDatasetExportJob](#), einschließlich des Exportauftragsstatus.

Anforderungssyntax

```
{  
  "datasetExportJobArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetExportJobArn](#)

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Datensatz-Exportjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "datasetExportJob": {  
    "creationDateTime": number,  
    "datasetArn": "string",  
    "datasetExportJobArn": "string",  
    "failureReason": "string",  
    "ingestionMode": "string",  
    "jobName": "string",  
    "jobOutput": {  
      "s3DataDestination": {  
        "kmsKeyArn": "string",  

```

```
        "path": "string"
    }
},
"lastUpdatedDateTime": number,
"roleArn": "string",
"status": "string"
}
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetExportJob](#)

Informationen über den Datensatz-Exportauftrag, einschließlich des Status.

Der Status ist einer der folgenden Werte:

- AUSSTEHEND ERSTELLEN
- IN_PROGRESS ERSTELLEN
- ACTIVE
- ERSTELLEN IST FEHLGESCHLAGEN

Typ: [DatasetExportJob](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeDatasetGroup

Service: Amazon Personalize

Beschreibt die angegebene Datensatzgruppe. Weitere Hinweise zu Datensatzgruppen finden Sie unter [CreateDatasetGroup](#).

Anforderungssyntax

```
{  
  "datasetGroupArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der zu beschreibenden Datensatzgruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "datasetGroup": {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "domain": "string",  
    "failureReason": "string",  
    "kmsKeyArn": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "roleArn": "string",  
    "status": "string"  
  }  
}
```

```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetGroup](#)

Eine Liste der Eigenschaften der Datensatzgruppe.

Typ: [DatasetGroup](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)

- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeDatasetImportJob

Service: Amazon Personalize

Beschreibt den von erstellten Datensatz-Importjob [CreateDatasetImportJob](#), einschließlich des Importauftragsstatus.

Anforderungssyntax

```
{  
  "datasetImportJobArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetImportJobArn](#)

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Datensatz-Import-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "datasetImportJob": {  
    "creationDateTime": number,  
    "datasetArn": "string",  
    "datasetImportJobArn": "string",  
    "dataSource": {  
      "dataLocation": "string"  
    },  
    "failureReason": "string",  
    "importMode": "string",  
    "jobName": "string",  
    "lastUpdatedDateTime": number,  
  }
```

```
"publishAttributionMetricsToS3": boolean,  
"roleArn": "string",  
"status": "string"  
}  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetImportJob](#)

Informationen über den Datensatz-Importauftrag, einschließlich des Status.

Der Status ist einer der folgenden Werte:

- AUSSTEHEND ERSTELLEN
- IN_PROGRESS ERSTELLEN
- ACTIVE
- ERSTELLEN IST FEHLGESCHLAGEN

Typ: [DatasetImportJob](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeEventTracker

Service: Amazon Personalize

Beschreibt einen Event-Tracker. Die Antwort beinhaltet das `trackingId` und `status` des Event-Trackers. Weitere Informationen zu Event-Trackern finden Sie unter [CreateEventTracker](#).

Anforderungssyntax

```
{
  "eventTrackerArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[eventTrackerArn](#)

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Event-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-])*:personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
  "eventTracker": {
    "accountId": "string",
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "eventTrackerArn": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string",
    "trackingId": "string"
  }
}
```



```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[eventTracker](#)

Ein Objekt, das den Event-Tracker beschreibt.

Typ: [EventTracker](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)

- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeFeatureTransformation

Service: Amazon Personalize

Beschreibt die angegebene Merkmalstransformation.

Anforderungssyntax

```
{  
  "featureTransformationArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

featureTransformationArn

Der Amazon-Ressourcenname (ARN) der zu beschreibenden Feature-Transformation.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "featureTransformation": {  
    "creationDateTime": number,  
    "defaultParameters": {  
      "string" : "string"  
    },  
    "featureTransformationArn": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "status": "string"  
  }  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[featureTransformation](#)

Eine Auflistung der FeatureTransformation Eigenschaften.

Typ: [FeatureTransformation](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)

- [AWS-SDK für Ruby V3](#)

DescribeFilter

Service: Amazon Personalize

Beschreibt die Eigenschaften eines Filters.

Anforderungssyntax

```
{
  "filterArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

filterArn

Der ARN des zu beschreibenden Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
  "filter": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "failureReason": "string",
    "filterArn": "string",
    "filterExpression": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

filter

Die Details des Filters.

Typ: [Filter](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)

- [AWS-SDK für Ruby V3](#)

DescribeMetricAttribution

Service: Amazon Personalize

Beschreibt eine metrische Zuordnung.

Anforderungssyntax

```
{  
  "metricAttributionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

metricAttributionArn

Der Amazon-Ressourcenname (ARN) der Metrikzuweisung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "metricAttribution": {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "failureReason": "string",  
    "lastUpdatedDateTime": number,  
    "metricAttributionArn": "string",  
    "metricsOutputConfig": {  
      "roleArn": "string",  
      "s3DataDestination": {  
        "kmsKeyArn": "string",  
        "path": "string"  
      }  
    }  
  }  
}
```

```
    },  
    "name": "string",  
    "status": "string"  
  }  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[metricAttribution](#)

Die Details der metrischen Zuordnung.

Typ: [MetricAttribution](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)

- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeRecipe

Service: Amazon Personalize

Beschreibt ein Rezept.

Ein Rezept enthält drei Elemente:

- Ein Algorithmus, der ein Modell trainiert.
- Hyperparameter, die das Training bestimmen.
- Informationen zur Merkmalstransformation zur Änderung der Eingabedaten vor dem Training.

Amazon Personalize bietet eine Reihe vordefinierter Rezepte. Sie geben ein Rezept an, wenn Sie eine Lösung mit der [CreateSolution](#)API erstellen. `CreateSolution` trainiert ein Modell mithilfe des Algorithmus im angegebenen Rezept und eines Trainingsdatensatzes. Wenn die Lösung als Kampagne bereitgestellt wird, kann sie mithilfe der [GetRecommendations](#)API Empfehlungen geben.

Anforderungssyntax

```
{  
  "recipeArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[recipeArn](#)

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
```

```
"recipe": {
  "algorithmArn": "string",
  "creationDateTime": number,
  "description": "string",
  "featureTransformationArn": "string",
  "lastUpdatedDateTime": number,
  "name": "string",
  "recipeArn": "string",
  "recipeType": "string",
  "status": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[recipe](#)

Ein Objekt, das das Rezept beschreibt.

Typ: [Recipe](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeRecommender

Service: Amazon Personalize

Beschreibt den angegebenen Empfehlungsgeber, einschließlich seines Status.

Ein Empfehlungsgeber kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INAKTIV > START AUSSTEHEND > START IN_PROGRESS > AKTIV
- DELETE PENDING > DELETE IN_PROGRESS

Wenn jaCREATE_FAILED, enthält die Antwort den failureReason Schlüssel, der beschreibt, warum. status

Der modelMetrics Schlüssel ist Null, wenn der Empfehlungsgeber erstellt oder gelöscht wird.

Weitere Informationen zu Empfehlungsgebern finden Sie unter [CreateRecommender](#)

Anforderungssyntax

```
{  
  "recommenderArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

recommenderArn

Der Amazon-Ressourcenname (ARN) des zu beschreibenden Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: arn:([a-z\d-]+):personalize:.*:.*:.*+

Erforderlich: Ja

Antwortsyntax

```

{
  "recommender": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "latestRecommenderUpdate": {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "recommenderConfig": {
        "enableMetadataWithRecommendations": boolean,
        "itemExplorationConfig": {
          "string" : "string"
        },
        "minRecommendationRequestsPerSecond": number,
        "trainingDataConfig": {
          "excludedDatasetColumns": {
            "string" : [ "string" ]
          }
        }
      }
    },
    "status": "string"
  },
  "modelMetrics": {
    "string" : number
  },
  "name": "string",
  "recipeArn": "string",
  "recommenderArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "minRecommendationRequestsPerSecond": number,
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "string" : [ "string" ]
      }
    }
  }
},

```



```
    "status": "string"  
  }  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

recommender

Die Eigenschaften des Empfehlungsgebers.

Typ: [Recommender](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeSchema

Service: Amazon Personalize

Beschreibt ein Schema. Weitere Informationen zu Schemata finden Sie unter [CreateSchema](#).

Anforderungssyntax

```
{
  "schemaArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[schemaArn](#)

Der Amazon-Ressourcenname (ARN) des abzurufenden Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
  "schema": {
    "creationDateTime": number,
    "domain": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "schema": "string",
    "schemaArn": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

schema

Das angeforderte Schema.

Typ: [DatasetSchema](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeSolution

Service: Amazon Personalize

Beschreibt eine Lösung. Weitere Informationen zu Lösungen finden Sie unter [CreateSolution](#).

Anforderungssyntax

```
{  
  "solutionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[solutionArn](#)

Der Amazon-Ressourcenname (ARN) der zu beschreibenden Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "solution": {  
    "autoMLResult": {  
      "bestRecipeArn": "string"  
    },  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "eventType": "string",  
    "lastUpdatedDateTime": number,  
    "latestSolutionVersion": {  
      "creationDateTime": number,  
      "failureReason": "string",  
      "lastUpdatedDateTime": number,  
      "solutionVersionArn": "string",  
    }  
  }  
}
```

```

    "status": "string"
  },
  "name": "string",
  "performAutoML": boolean,
  "performHPO": boolean,
  "recipeArn": "string",
  "solutionArn": "string",
  "solutionConfig": {
    "algorithmHyperParameters": {
      "string" : "string"
    },
    "autoMLConfig": {
      "metricName": "string",
      "recipeList": [ "string" ]
    },
    "eventValueThreshold": "string",
    "featureTransformationParameters": {
      "string" : "string"
    },
    "hpoConfig": {
      "algorithmHyperParameterRanges": {
        "categoricalHyperParameterRanges": [
          {
            "name": "string",
            "values": [ "string" ]
          }
        ],
        "continuousHyperParameterRanges": [
          {
            "maxValue": number,
            "minValue": number,
            "name": "string"
          }
        ],
        "integerHyperParameterRanges": [
          {
            "maxValue": number,
            "minValue": number,
            "name": "string"
          }
        ]
      },
      "hpoObjective": {
        "metricName": "string",

```

```
        "metricRegex": "string",
        "type": "string"
    },
    "hpoResourceConfig": {
        "maxNumberOfTrainingJobs": "string",
        "maxParallelTrainingJobs": "string"
    }
},
"optimizationObjective": {
    "itemAttribute": "string",
    "objectiveSensitivity": "string"
},
"trainingDataConfig": {
    "excludedDatasetColumns": {
        "string" : [ "string" ]
    }
}
},
"status": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

solution

Ein Objekt, das die Lösung beschreibt.

Typ: [Solution](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

DescribeSolutionVersion

Service: Amazon Personalize

Beschreibt eine bestimmte Version einer Lösung. Weitere Informationen zu Lösungen finden Sie unter [CreateSolution](#)

Anforderungssyntax

```
{  
  "solutionVersionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[solutionVersionArn](#)

Der Amazon-Ressourcenname (ARN) der Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "solutionVersion": {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "eventType": "string",  
    "failureReason": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "performAutoML": boolean,  
    "performHPO": boolean,  
    "recipeArn": "string",  
    "solutionArn": "string",  
    "solutionConfig": {
```

```
"algorithmHyperParameters": {
  "string" : "string"
},
"autoMLConfig": {
  "metricName": "string",
  "recipeList": [ "string" ]
},
"eventValueThreshold": "string",
"featureTransformationParameters": {
  "string" : "string"
},
"hpoConfig": {
  "algorithmHyperParameterRanges": {
    "categoricalHyperParameterRanges": [
      {
        "name": "string",
        "values": [ "string" ]
      }
    ],
    "continuousHyperParameterRanges": [
      {
        "maxValue": number,
        "minValue": number,
        "name": "string"
      }
    ],
    "integerHyperParameterRanges": [
      {
        "maxValue": number,
        "minValue": number,
        "name": "string"
      }
    ]
  },
  "hpoObjective": {
    "metricName": "string",
    "metricRegex": "string",
    "type": "string"
  },
  "hpoResourceConfig": {
    "maxNumberOfTrainingJobs": "string",
    "maxParallelTrainingJobs": "string"
  }
},
```

```
"optimizationObjective": {
  "itemAttribute": "string",
  "objectiveSensitivity": "string"
},
"trainingDataConfig": {
  "excludedDatasetColumns": {
    "string" : [ "string" ]
  }
},
"solutionVersionArn": "string",
"status": "string",
"trainingHours": number,
"trainingMode": "string",
"tunedHPOParams": {
  "algorithmHyperParameters": {
    "string" : "string"
  }
}
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[solutionVersion](#)

Die Lösungsversion.

Typ: [SolutionVersion](#) Objekt

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

GetSolutionMetrics

Service: Amazon Personalize

Ruft die Metriken für die angegebene Lösungsversion ab.

Anforderungssyntax

```
{  
  "solutionVersionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[solutionVersionArn](#)

Der Amazon-Ressourcenname (ARN) der Lösungsversion, für die Metriken abgerufen werden sollen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "metrics": {  
    "string" : number  
  },  
  "solutionVersionArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

metrics

Die Metriken für die Lösungsversion. Weitere Informationen finden Sie unter [Evaluieren einer Lösungsversion anhand von Metriken](#).

Typ: Zeichenfolge zu doppeltem Mapping

Karteneinträge: Maximale Anzahl von 100 Elementen.

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 256.

solutionVersionArn

Dieselbe Lösungsversion ARN wie in der Anfrage angegeben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListBatchInferenceJobs

Service: Amazon Personalize

Ruft eine Liste der Batch-Inferenzjobs ab, die anhand einer Lösungsversion ausgeführt wurden.

Anforderungssyntax

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionVersionArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Ergebnissen von Batch-Inferenzaufträgen, die auf jeder Seite zurückgegeben werden sollen. Der Standardwert lautet 100.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Das Token zum Anfordern der nächsten Ergebnisseite.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

[solutionVersionArn](#)

Der Amazon-Ressourcenname (ARN) der Lösungsversion, aus der die Batch-Inferenzjobs erstellt wurden.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

Antwortsyntax

```
{
  "batchInferenceJobs": [
    {
      "batchInferenceJobArn": "string",
      "batchInferenceJobMode": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[batchInferenceJobs](#)

Eine Liste mit Informationen zu jedem Auftrag, der zurückgegeben wird.

Typ: Array von [BatchInferenceJobSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

[nextToken](#)

Das Token, das zum Abrufen der nächsten Ergebnisseite verwendet werden soll. Der Wert gibt `annull`, wenn keine weiteren Ergebnisse zurückgegeben werden können.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListBatchSegmentJobs

Service: Amazon Personalize

Ruft eine Liste der Batchsegmentjobs ab, die mit einer von Ihnen angegebenen Lösungsversion ausgeführt wurden.

Anforderungssyntax

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionVersionArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Ergebnissen von Batchsegmentaufträgen, die auf jeder Seite zurückgegeben werden sollen. Der Standardwert lautet 100.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Das Token zum Anfordern der nächsten Ergebnisseite.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

[solutionVersionArn](#)

Der Amazon-Ressourcenname (ARN) der Lösungsversion, die die Batch-Segment-Jobs zur Generierung von Batchsegmenten verwendet haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

Antwortsyntax

```
{
  "batchSegmentJobs": [
    {
      "batchSegmentJobArn": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[batchSegmentJobs](#)

Eine Liste mit Informationen zu jedem Auftrag, der zurückgegeben wird.

Typ: Array von [BatchSegmentJobSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

[nextToken](#)

Das Token, das zum Abrufen der nächsten Ergebnisseite verwendet werden soll. Der Wert gibt `null`, wenn keine weiteren Ergebnisse zurückgegeben werden können.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListCampaigns

Service: Amazon Personalize

Gibt eine Liste von Kampagnen zurück, die die angegebene Lösung verwenden. Wenn keine Lösung angegeben ist, werden alle Kampagnen aufgelistet, die dem Konto zugeordnet sind. Die Antwort enthält die Eigenschaften für jede Kampagne, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Kampagnen finden Sie unter [CreateCampaign](#).

Anforderungssyntax

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Kampagnen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein Token, das vom vorherigen Aufruf zurückgegeben wurde, [ListCampaigns](#) um die nächsten Kampagnen abzurufen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

solutionArn

Der Amazon-Ressourcenname (ARN) der Lösung, für die die Kampagnen aufgelistet werden sollen. Wenn keine Lösung angegeben ist, werden alle mit dem Konto verknüpften Kampagnen aufgelistet.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

Antwortsyntax

```
{
  "campaigns": [
    {
      "campaignArn": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

campaigns

Eine Liste der Kampagnen.

Typ: Array von [CampaignSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen der nächsten Kampagnengruppe (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListDatasetExportJobs

Service: Amazon Personalize

Gibt eine Liste von Datensatz-Exportjobs zurück, die den angegebenen Datensatz verwenden. Wenn kein Datensatz angegeben ist, werden alle mit dem Konto verknüpften Datensatz-Exportaufträge aufgelistet. Die Antwort enthält die Eigenschaften für jeden Datensatz-Exportauftrag, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Datensatz-Exportaufträgen finden Sie unter [CreateDatasetExportJob](#). Weitere Informationen zu Datensätzen finden Sie unter [CreateDataset](#).

Anforderungssyntax

```
{  
  "datasetArn": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Datensatzes, für den die Datensatz-Exportjobs aufgelistet werden sollen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Datensatz-Exportaufträgen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein vom vorherigen Aufruf von zurückgegebenes Token `ListDatasetExportJobs` zum Abrufen der nächsten Gruppe von Datensatz-Exportaufträgen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetExportJobs": [
    {
      "creationDateTime": number,
      "datasetExportJobArn": "string",
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetExportJobs](#)

Die Liste der Datenexportaufträge.

Typ: Array von [DatasetExportJobSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen der nächsten Gruppe von Datensatz-Exportaufträgen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListDatasetGroups

Service: Amazon Personalize

Gibt eine Liste von Datensatzgruppen zurück. Die Antwort enthält die Eigenschaften für jede Datensatzgruppe, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Datensatzgruppen finden Sie unter [CreateDatasetGroup](#).

Anforderungssyntax

```
{
  "maxResults": number,
  "nextToken": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Datensatzgruppen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein vom vorherigen Aufruf von zurückgegebenes Token ListDatasetGroups zum Abrufen der nächsten Gruppe von Datensatzgruppen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
```

```
"datasetGroups": [  
  {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "domain": "string",  
    "failureReason": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "status": "string"  
  }  
],  
"nextToken": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetGroups](#)

Die Liste Ihrer Datensatzgruppen.

Typ: Array von [DatasetGroupSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

[nextToken](#)

Ein Token zum Abrufen der nächsten Gruppe von Datensatzgruppen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListDatasetImportJobs

Service: Amazon Personalize

Gibt eine Liste von Datensatz-Importjobs zurück, die den angegebenen Datensatz verwenden. Wenn kein Datensatz angegeben ist, werden alle mit dem Konto verknüpften Datenmengen-Importaufträge aufgelistet. Die Antwort enthält die Eigenschaften für jeden Datensatz-Import-Job, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Datensatz-Importaufträgen finden Sie unter [CreateDatasetImportJob](#). Weitere Informationen zu Datensätzen finden Sie unter [CreateDataset](#).

Anforderungssyntax

```
{
  "datasetArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Datensatzes, für den die Datensatz-Import-Jobs aufgelistet werden sollen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Datensatz-Import-Jobs, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein vom vorherigen Aufruf von zurückgegebenes Token `ListDatasetImportJobs` zum Abrufen der nächsten Gruppe von Datensatz-Importaufträgen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasetImportJobs": [
    {
      "creationDateTime": number,
      "datasetImportJobArn": "string",
      "failureReason": "string",
      "importMode": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[datasetImportJobs](#)

Die Liste der Jobs für den Import von Datensätzen.

Typ: Array von [DatasetImportJobSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen der nächsten Gruppe von Datensatz-Importaufträgen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListDatasets

Service: Amazon Personalize

Gibt die Liste der Datensätze zurück, die in der angegebenen Datensatzgruppe enthalten sind. Die Antwort enthält die Eigenschaften für jeden Datensatz, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Datensätzen finden Sie unter [CreateDataset](#).

Anforderungssyntax

```
{  
  "datasetGroupArn": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der Datensatzgruppe, die die aufzulistenden Datensätze enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Datensätzen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Ein vom vorherigen Aufruf von zurückgegebenes Token `ListDatasets` zum Abrufen der nächsten Gruppe von Datensatz-Importaufträgen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "datasets": [
    {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetType": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

datasets

Ein Array von Dataset-Objekten. Jedes Objekt stellt Metadateninformationen bereit.

Typ: Array von [DatasetSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen des nächsten Datensatzes (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListEventTrackers

Service: Amazon Personalize

Gibt die Liste der mit dem Konto verknüpften Event-Tracker zurück. Die Antwort enthält die Eigenschaften für jeden Event-Tracker, einschließlich des Amazon-Ressourcennamens (ARN) und der Tracking-ID. Weitere Informationen zu Event-Trackern finden Sie unter [CreateEventTracker](#).

Anforderungssyntax

```
{  
  "datasetGroupArn": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der ARN einer Datensatzgruppe, die zum Filtern der Antwort verwendet wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Event-Trackern, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein vom vorherigen Aufruf zurückgegebenes Token `ListEventTrackers` zum Abrufen der nächsten Gruppe von Event-Trackern (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "eventTrackers": [
    {
      "creationDateTime": number,
      "eventTrackerArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

eventTrackers

Eine Liste von Event-Trackern.

Typ: Array von [EventTrackerSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen der nächsten Gruppe von Event-Trackern (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListFilters

Service: Amazon Personalize

Listet alle Filter auf, die zu einer bestimmten Datensatzgruppe gehören.

Anforderungssyntax

```
{  
  "datasetGroupArn": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetGroupArn

Der ARN der Datensatzgruppe, die die Filter enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

maxResults

Die maximale Anzahl von Filtern, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Ein vom vorherigen Aufruf zurückgegebenes Token `ListFilters` zum Abrufen des nächsten Filtersatzes (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "Filters": [
    {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "filterArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

Filters

Eine Liste der zurückgegebenen Filter.

Typ: Array von [FilterSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Ein Token zum Abrufen des nächsten Filtersatzes (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListMetricAttributionMetrics

Service: Amazon Personalize

Listet die Metriken für die Metrikzuweisung auf.

Anforderungssyntax

```
{
  "maxResults": number,
  "metricAttributionArn": "string",
  "nextToken": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Metriken, die auf einer Ergebnisseite zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[metricAttributionArn](#)

Der Amazon-Ressourcenname (ARN) der Metrikzuweisung, für die Attribute abgerufen werden sollen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[nextToken](#)

Geben Sie das Paginierungstoken aus einer vorherigen Anfrage an, um die nächste Ergebnisseite abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "metrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

metrics

Die Metriken für die angegebene Metrik-Attribution.

Typ: Array von [MetricAttribute](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 10 Elementen.

nextToken

Geben Sie das Paginierungstoken aus einer vorherigen `ListMetricAttributionMetricsResponse` Anfrage an, um die nächste Ergebnisseite abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListMetricAttributions

Service: Amazon Personalize

Listet metrische Attributionen auf.

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetGroupArn

Die Datensatzgruppe Amazon Resource Name (ARN) der metrischen Attributionen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

maxResults

Die maximale Anzahl von metrischen Attributen, die auf einer Ergebnisseite zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Geben Sie das Paginierungstoken aus einer vorherigen Anfrage an, um die nächste Ergebnisseite abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "metricAttributions": [
    {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "metricAttributionArn": "string",
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

metricAttributions

Die Liste der metrischen Attributionen.

Typ: Array von [MetricAttributionSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

nextToken

Geben Sie das Paginierungstoken aus einer vorherigen Anfrage an, um die nächste Ergebnisseite abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListRecipes

Service: Amazon Personalize

Gibt eine Liste der verfügbaren Rezepte zurück. Die Antwort enthält die Eigenschaften für jedes Rezept, einschließlich des Amazon-Ressourcennamens (ARN) des Rezepts.

Anforderungssyntax

```
{
  "domain": "string",
  "maxResults": number,
  "nextToken": "string",
  "recipeProvider": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

domain

Filtert zurückgegebene Rezepte nach Domain für eine Domain-Datensatzgruppe. Nur Rezepte (Anwendungsfälle für Domain-Datensatzgruppen) für diese Domäne sind in der Antwort enthalten. Wenn Sie keine Domäne angeben, werden alle Rezepte zurückgegeben.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Erforderlich: Nein

maxResults

Die maximale Anzahl von Rezepten, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Ein Token, das vom vorherigen Aufruf zurückgegeben wurde, ListRecipes um den nächsten Satz von Rezepten abzurufen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

[recipeProvider](#)

Der Standardwert ist SERVICE.

Typ: Zeichenfolge

Zulässige Werte: SERVICE

Erforderlich: Nein

Antwortsyntax

```
{
  "nextToken": "string",
  "recipes": [
    {
      "creationDateTime": number,
      "domain": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "status": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[nextToken](#)

Ein Token, um die nächsten Rezepte zu erhalten.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

[recipes](#)

Die Liste der verfügbaren Rezepte.

Typ: Array von [RecipeSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListRecommenders

Service: Amazon Personalize

Gibt eine Liste von Empfehlungen in einer bestimmten Domain-Datensatzgruppe zurück. Wenn keine Domain-Datensatzgruppe angegeben ist, werden alle mit dem Konto verknüpften Empfehlungen aufgelistet. Die Antwort enthält die Eigenschaften für jeden Empfehlungsgeber, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Empfehlungsgebern finden Sie unter [CreateRecommender](#)

Anforderungssyntax

```
{  
  "datasetGroupArn": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der Domain-Datensatzgruppe, für die die Empfehlungen aufgelistet werden sollen. Wenn keine Domain-Datensatzgruppe angegeben ist, werden alle mit dem Konto verknüpften Empfehlungsgeber aufgelistet.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Empfehlungsgebern, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Ein vom vorherigen Aufruf zurückgegebenes Token ListRecommenders zum Abrufen der nächsten Gruppe von Empfehlungen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "nextToken": "string",
  "recommenders": [
    {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "recommenderArn": "string",
      "recommenderConfig": {
        "enableMetadataWithRecommendations": boolean,
        "itemExplorationConfig": {
          "string" : "string"
        },
        "minRecommendationRequestsPerSecond": number,
        "trainingDataConfig": {
          "excludedDatasetColumns": {
            "string" : [ "string" ]
          }
        }
      },
      "status": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

nextToken

Ein Token zum Abrufen der nächsten Gruppe von Empfehlungen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

recommenders

Eine Liste der Empfehlungsgeber.

Typ: Array von [RecommenderSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListSchemas

Service: Amazon Personalize

Gibt die Liste der Schemas zurück, die dem Konto zugeordnet sind. Die Antwort enthält die Eigenschaften für jedes Schema, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Schemata finden Sie unter [CreateSchema](#).

Anforderungssyntax

```
{  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Schemas, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein Token, das vom vorherigen Aufruf zurückgegeben wurde, ListSchemas um den nächsten Satz von Schemas abzurufen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{  
  "nextToken": "string",  
  "schemas": [  
    ...  
  ]  
}
```



```
{
  "creationDateTime": number,
  "domain": "string",
  "lastUpdatedDateTime": number,
  "name": "string",
  "schemaArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[nextToken](#)

Ein Token, das verwendet wird, um den nächsten Satz von Schemas abzurufen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

[schemas](#)

Eine Liste von Schemas.

Typ: Array von [DatasetSchemaSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Fehler

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListSolutions

Service: Amazon Personalize

Gibt eine Liste von Lösungen zurück, die die angegebene Datensatzgruppe verwenden. Wenn keine Datensatzgruppe angegeben ist, werden alle mit dem Konto verknüpften Lösungen aufgelistet. Die Antwort enthält die Eigenschaften für jede Lösung, einschließlich des Amazon-Ressourcennamens (ARN). Weitere Informationen zu Lösungen finden Sie unter [CreateSolution](#).

Anforderungssyntax

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetGroupArn](#)

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[maxResults](#)

Die maximale Anzahl von Lösungen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

nextToken

Ein vom vorherigen Aufruf zurückgegebenes Token ListSolutions zum Abrufen der nächsten Lösungsgruppe (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

Antwortsyntax

```
{
  "nextToken": "string",
  "solutions": [
    {
      "creationDateTime": number,
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "solutionArn": "string",
      "status": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

nextToken

Ein Token zum Abrufen der nächsten Reihe von Lösungen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

solutions

Eine Liste der aktuellen Lösungen.

Typ: Array von [SolutionSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListSolutionVersions

Service: Amazon Personalize

Gibt eine Liste von Lösungsversionen für die angegebene Lösung zurück. Wenn keine Lösung angegeben ist, werden alle mit dem Konto verknüpften Lösungsversionen aufgelistet. Die Antwort enthält die Eigenschaften für jede Lösungsversion, einschließlich des Amazon-Ressourcennamens (ARN).

Anforderungssyntax

```
{  
  "maxResults": number,  
  "nextToken": "string",  
  "solutionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[maxResults](#)

Die maximale Anzahl von Lösungsversionen, die zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

[nextToken](#)

Ein vom vorherigen Aufruf zurückgegebenes Token ListSolutionVersions zum Abrufen der nächsten Lösungsversionen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

Erforderlich: Nein

[solutionArn](#)

Den Amazon-Ressourcenname (ARN) der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

Antwortsyntax

```
{
  "nextToken": "string",
  "solutionVersions": [
    {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[nextToken](#)

Ein Token zum Abrufen der nächsten Lösungsversionen (falls vorhanden).

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 1500.

[solutionVersions](#)

Eine Liste von Lösungsversionen, in der die Versionseigenschaften beschrieben werden.

Typ: Array von [SolutionVersionSummary](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

InvalidNextTokenException

Das Token ist nicht gültig.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListTagsForResource

Service: Amazon Personalize

Ruft eine Liste von [Tags](#) ab, die einer Ressource zugeordnet sind.

Anforderungssyntax

```
{
  "resourceArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[resourceArn](#)

Der Amazon-Ressourcenname (ARN) der Ressource.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

tags

Die Tags der Ressource.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)

- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

StartRecommender

Service: Amazon Personalize

Startet eine Empfehlung, die INAKTIV ist. Durch das Starten eines Empfehlungsgebers werden keine neuen Modelle erstellt, sondern die Abrechnung und automatische Umschulung für den Empfehlungsgeber wieder aufgenommen.

Anforderungssyntax

```
{  
  "recommenderArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

recommenderArn

Der Amazon-Ressourcenname (ARN) des zu startenden Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
{  
  "recommenderArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungsgebers, den Sie gestartet haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)

- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

StopRecommender

Service: Amazon Personalize

Stoppt eine Empfehlung, die AKTIV ist. Wenn ein Empfehlungsgeber beendet wird, werden die Abrechnung und die automatische Umschulung für den Empfehlungsgeber gestoppt.

Anforderungssyntax

```
{  
  "recommenderArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungsgebers, der beendet werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortsyntax

```
{  
  "recommenderArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungsgebers, den Sie gestoppt haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

StopSolutionVersionCreation

Service: Amazon Personalize

Beendet die Erstellung einer Lösungsversion, die sich im Status CREATE_PENDING oder CREATE_IN_PROGRESS befindet.

Je nach aktuellem Status der Lösungsversion ändert sich der Status der Lösungsversion wie folgt:

- CREATE_PENDING > CREATE_STOPPED

or

- CREATE_IN_PROGRESS > CREATE_STOPS > CREATE_STOPPED

Ihnen werden alle abgeschlossenen Schulungen in Rechnung gestellt, bis Sie die Erstellung der Lösungsversion beenden. Sie können die Erstellung einer Lösungsversion nicht fortsetzen, nachdem sie beendet wurde.

Anforderungssyntax

```
{  
  "solutionVersionArn": "string"  
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion, deren Erstellung Sie beenden möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

TagResource

Service: Amazon Personalize

Fügen Sie einer Ressource eine Liste von Tags hinzu.

Anforderungssyntax

```
{
  "resourceArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

resourceArn

Der Amazon-Ressourcenname (ARN) der Ressource.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

tags

Tags, die auf die Ressource angewendet werden sollen. Weitere Informationen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

LimitExceededException

Das Limit für die Anzahl der Anforderungen pro Sekunde wurde überschritten.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagsException

Sie haben die maximale Anzahl von Tags überschritten, die Sie auf diese Ressource anwenden können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)

- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UntagResource

Service: Amazon Personalize

Entfernt die angegebenen Tags, die einer Ressource zugeordnet sind. Weitere Informationen finden Sie unter [Entfernen von Tags aus Amazon-Personalize-Ressourcen](#).

Anforderungssyntax

```
{
  "resourceArn": "string",
  "tagKeys": [ "string" ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[resourceArn](#)

Der Amazon-Ressourcenname (ARN) der Ressource.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

[tagKeys](#)

Die Schlüssel der Tags, die entfernt werden sollen.

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 200 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[^\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Erforderlich: Ja

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

TooManyTagKeysException

Die Anforderung enthält mehr Tag-Schlüssel, als einer Ressource zugeordnet werden können (50 Tag-Schlüssel pro Ressource).

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)

- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateCampaign

Service: Amazon Personalize

Aktualisiert eine Kampagne, um eine neu trainierte Lösungsversion mit einer vorhandenen Kampagne bereitzustellen, die Ihrer Kampagne zu ändern oder die Konfiguration Ihrer Kampagne zu ändern, z. B. die Explorationskonfiguration. `minProvisionedTPS`

Um eine Kampagne zu aktualisieren, muss der Kampagnenstatus `AKTIV` oder `CREATE FAILED` lauten. Überprüfen Sie den Kampagnenstatus mithilfe der [DescribeCampaign](#) Operation.

Note

Sie können immer noch Empfehlungen von einer Kampagne erhalten, während ein Update läuft. Die Kampagne verwendet die vorherige Lösungsversion und Kampagnenkonfiguration, um Empfehlungen zu generieren, bis der aktuelle Status der Kampagnenaktualisierung `lauteActive`.

Weitere Informationen zum Aktualisieren einer Kampagne, einschließlich Codebeispielen, finden Sie unter [Kampagne aktualisieren](#). Weitere Informationen zu Kampagnen finden Sie unter [Kampagne erstellen](#).

Anforderungssyntax

```
{
  "campaignArn": "string",
  "campaignConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string": "string"
    }
  },
  "minProvisionedTPS": number,
  "solutionVersionArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[campaignArn](#)

Der Amazon-Ressourcennamen (ARN) der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

[campaignConfig](#)

Die Konfigurationsdetails einer Kampagne.

Typ: [CampaignConfig](#) Objekt

Erforderlich: Nein

[minProvisionedTPS](#)

Gibt die angeforderte Mindestanzahl bereitgestellter Transaktionen (Empfehlungen) pro Sekunde an, die Amazon Personalize unterstützt. Ein Hoch erhöht `minProvisionedTPS` Ihre Rechnung. Wir empfehlen, mit 1 für zu beginnen `minProvisionedTPS` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minProvisionedTPS` diese bei Bedarf.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Nein

[solutionVersionArn](#)

Der ARN einer neuen Lösungsversion, die bereitgestellt werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

Antwortsyntax

```
{  
  "campaignArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

campaignArn

Derselbe Kampagnen-ARN wie in der Anfrage angegeben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

UpdateDataset

Service: Amazon Personalize

Aktualisieren Sie einen Datensatz, um sein Schema durch ein neues oder vorhandenes zu ersetzen. Weitere Informationen finden Sie unter [Ersetzen des Schemas eines Datensatzes](#).

Anforderungssyntax

```
{
  "datasetArn": "string",
  "schemaArn": "string"
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Datensatzes, den Sie aktualisieren möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

[schemaArn](#)

Der Amazon-Ressourcenname (ARN) des neuen Schemas, das Sie verwenden möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

Antwortsyntax

```
{
```

```
"datasetArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

datasetArn

Der Amazon-Ressourcenname (ARN) des Datensatzes, den Sie aktualisiert haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

UpdateMetricAttribution

Service: Amazon Personalize

Aktualisiert eine metrische Zuordnung.

Anforderungssyntax

```
{
  "addMetrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "metricAttributionArn": "string",
  "metricsOutputConfig": {
    "roleArn": "string",
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "removeMetrics": [ "string" ]
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[addMetrics](#)

Fügt der Metrik-Attribution neue Metrikattribute hinzu.

Typ: Array von [MetricAttribute](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 10 Elementen.

Erforderlich: Nein

[metricAttributionArn](#)

Der Amazon-Ressourcenname (ARN) für die zu aktualisierende Metrikzuweisung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

[metricsOutputConfig](#)

Eine Ausgabekonfiguration für die Metrikzuweisung.

Typ: [MetricAttributionOutput](#) Objekt

Erforderlich: Nein

[removeMetrics](#)

Entferne Metrikattribute aus der Metrik-Attribution.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 10 Elementen.

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Antwortsyntax

```
{  
  "metricAttributionArn": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[metricAttributionArn](#)

Der Amazon-Ressourcenname (ARN) für die Metrikzuweisung, die Sie aktualisiert haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceAlreadyExistsException

Die angegebene Ressource ist bereits vorhanden.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

UpdateRecommender

Service: Amazon Personalize

Aktualisiert den Empfehlungsgeber, um die Empfehlungskonfiguration zu ändern. Wenn Sie den Empfehlungsgeber aktualisieren, um die in der Schulung verwendeten Spalten zu ändern, startet Amazon Personalize automatisch eine vollständige Neuschulung der Modelle, die Ihren Empfehlungsgeber unterstützen. Während das Update abgeschlossen ist, können Sie weiterhin Empfehlungen vom Empfehlungsgeber erhalten. Der Empfehlungsgeber verwendet die vorherige Konfiguration, bis das Update abgeschlossen ist. Um den Status dieses Updates zu verfolgen, verwenden Sie den im [DescribeRecommender](#) Vorgang `latestRecommenderUpdate` zurückgegebenen Wert.

Anforderungssyntax

```
{
  "recommenderArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "minRecommendationRequestsPerSecond": number,
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "string" : [ "string" ]
      }
    }
  }
}
```

Anforderungsparameter

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[recommenderArn](#)

Der Amazon-Ressourcenname (ARN) des zu ändernden Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

[recommenderConfig](#)

Die Konfigurationsdetails des Empfehlungsgebers.

Typ: [RecommenderConfig](#) Objekt

Erforderlich: Ja

Antwortsyntax

```
{
  "recommenderArn": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[recommenderArn](#)

Derselbe empfohlene Amazon Resource Name (ARN) wie in der Anfrage angegeben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

Amazon Personalize Events

Die folgenden Aktionen werden von Amazon Personalize Events unterstützt:

- [PutActionInteractions](#)
- [PutActions](#)
- [PutEvents](#)
- [PutItems](#)
- [PutUsers](#)

PutActionInteractions

Service: Amazon Personalize Events

Zeichnet Daten zu Aktionsinteraktionsereignissen auf. Ein Aktionsereignis ist eine Interaktion zwischen einem Benutzer und einer Aktion. Zum Beispiel ein Benutzer, der eine Aktion ausführt, z. B. sich für ein Mitgliedschaftsprogramm anmeldet oder Ihre App herunterlädt.

Weitere Informationen zum Aufzeichnen von Aktionsinteraktionen finden Sie unter [Aufzeichnen von Aktionsinteraktionsereignissen](#). Weitere Informationen zu Aktionen in einem Aktionsdatensatz finden Sie unter [Aktionsdatensatz](#).

Anforderungssyntax

```
POST /action-interactions HTTP/1.1
Content-type: application/json

{
  "actionInteractions": [
    {
      "actionId": "string",
      "eventId": "string",
      "eventType": "string",
      "impression": [ "string" ],
      "properties": "string",
      "recommendationId": "string",
      "sessionId": "string",
      "timestamp": number,
      "userId": "string"
    }
  ],
  "trackingId": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

actionInteractions

Eine Liste von Aktionsinteraktionsereignissen aus der Sitzung.

Typ: Array von [ActionInteraction](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 10 Elemente.

Erforderlich: Ja

trackingId

Die ID Ihres Aktionsinteraktions-Event-Trackers. Wenn Sie einen Datensatz mit Aktionsinteraktionen erstellen, erstellt Amazon Personalize für Sie einen Aktionsinteraktionen-Event-Tracker. Weitere Informationen finden Sie unter [Aktions-Interaktions-Event-Tracker-ID](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter ein.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP-Statuscode: 409

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

PutActions

Service: Amazon Personalize Events

Fügt einem Actions-Datensatz eine oder mehrere Aktionen hinzu. Weitere Informationen finden Sie unter [Aktionen einzeln importieren](#).

Anforderungssyntax

```
POST /actions HTTP/1.1
Content-type: application/json

{
  "actions": [
    {
      "actionId": "string",
      "properties": "string"
    }
  ],
  "datasetArn": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[actions](#)

Eine Liste von Aktionsdaten.

Typ: Array von [Action](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 10 Elemente.

Erforderlich: Ja

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Actions-Datensatzes, zu dem Sie die Aktion oder Aktionen hinzufügen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP-Statuscode: 409

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)

- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

PutEvents

Service: Amazon Personalize Events

Zeichnet Daten zu Artikelinteraktionsereignissen auf. Weitere Informationen finden Sie unter [Aufzeichnen von Interaktionsereignissen mit Artikeln](#).

Anforderungssyntax

```
POST /events HTTP/1.1
Content-type: application/json

{
  "eventList": [
    {
      "eventId": "string",
      "eventType": "string",
      "eventValue": number,
      "impression": [ "string" ],
      "itemId": "string",
      "metricAttribution": {
        "eventAttributionSource": "string"
      },
      "properties": "string",
      "recommendationId": "string",
      "sentAt": number
    }
  ],
  "sessionId": "string",
  "trackingId": "string",
  "userId": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

eventList

Eine Liste von Ereignisdaten aus der Sitzung.

Typ: Array von [Event](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 10 Elemente.

Erforderlich: Ja

[sessionId](#)

Die Sitzungs-ID, die dem Besuch des Benutzers zugeordnet ist. Ihre Anwendung generiert die sessionId, wenn ein Benutzer Ihre Website zum ersten Mal besucht oder Ihre Anwendung verwendet. Amazon Personalize verwendet die sessionId, um Ereignisse mit dem Benutzer zu verknüpfen, bevor er sich anmeldet. Weitere Informationen finden Sie unter [Aufzeichnen von Interaktionsereignissen mit Artikeln](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

[trackingId](#)

Die Tracking-ID für das Ereignis. Die ID wird durch einen Aufruf der [CreateEventTracker](#)API generiert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

[userId](#)

Der mit dem Ereignis verknüpfte Benutzer.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

PutItems

Service: Amazon Personalize Events

Fügt einem Artikel-Datensatz ein oder mehrere Elemente hinzu. Weitere Informationen finden Sie unter [Einzelne Elemente importieren](#).

Anforderungssyntax

```
POST /items HTTP/1.1
Content-type: application/json

{
  "datasetArn": "string",
  "items": [
    {
      "itemId": "string",
      "properties": "string"
    }
  ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[datasetArn](#)

Der Amazon-Ressourcenname (ARN) des Artikeldatensatzes, zu dem Sie den Artikel oder die Artikel hinzufügen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

items

Eine Liste von Artikeldaten.

Typ: Array von [Item](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 10 Elemente.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP-Statuscode: 409

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

PutUsers

Service: Amazon Personalize Events

Fügt einem Benutzerdatensatz einen oder mehrere Benutzer hinzu. Weitere Informationen finden Sie unter [Benutzer einzeln importieren](#).

Anforderungssyntax

```
POST /users HTTP/1.1
Content-type: application/json

{
  "datasetArn": "string",
  "users": [
    {
      "properties": "string",
      "userId": "string"
    }
  ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

datasetArn

Der Amazon-Ressourcenname (ARN) des Benutzerdatensatzes, zu dem Sie den oder die Benutzer hinzufügen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Ja

users

Eine Liste von Benutzerdaten.

Typ: Array von [User](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 10 Elemente.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceInUseException

Die angegebene Ressource wird verwendet.

HTTP-Statuscode: 409

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

Amazon Personalize Runtime

Die folgenden Aktionen werden von Amazon Personalize Runtime unterstützt:

- [GetActionRecommendations](#)
- [GetPersonalizedRanking](#)
- [GetRecommendations](#)

GetActionRecommendations

Service: Amazon Personalize Runtime

Gibt eine Liste empfohlener Aktionen zurück, sortiert in absteigender Reihenfolge nach dem Prognosewert. Verwenden Sie die GetActionRecommendations API, wenn Sie eine benutzerdefinierte Kampagne haben, die eine Lösungsversion bereitstellt, die nach einem Rezept von PERSONALIZED_ACTIONS trainiert wurde.

[Weitere Informationen über PERSONALIZED_ACTIONS-Rezepte finden Sie unter PERSONALIZED_ACTIONS-Rezepte.](#) [Weitere Informationen zum Abrufen von Handlungsempfehlungen finden Sie unter Abrufen von Handlungsempfehlungen.](#)

Anforderungssyntax

```
POST /action-recommendations HTTP/1.1
Content-type: application/json
```

```
{
  "campaignArn": "string",
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "numResults": number,
  "userId": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

campaignArn

Der Amazon-Ressourcenname (ARN) der Kampagne, die verwendet werden soll, um Handlungsempfehlungen zu erhalten. Diese Kampagne muss eine Lösungsversion bereitstellen, die mit einem PERSONALIZED_ACTIONS-Rezept trainiert wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

filterArn

Der ARN des Filters, der auf die zurückgegebenen Empfehlungen angewendet werden soll. Weitere Informationen finden Sie unter [Filtern von Empfehlungen](#).

Wenn Sie diesen Parameter verwenden, stellen Sie sicher, dass die Filterressource aktiviert ist `ACTIVE`.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

filterValues

Die Werte, die beim Filtern von Empfehlungen verwendet werden sollen. Geben Sie für jeden Platzhalterparameter in Ihrem Filterausdruck den Parameternamen (bei entsprechender Groß- und Kleinschreibung) als Schlüssel und die Filterwerte als entsprechenden Wert an. Trennen Sie mehrere Werte für einen Parameter durch ein Komma.

Für Filterausdrücke, die ein `INCLUDE` Element verwenden, um Aktionen einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein `EXCLUDE` Element verwenden, um Aktionen auszuschließen, können Sie das `filter-values` weglassen. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht zum Filtern von Empfehlungen.

Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 25 Elementen.

Einschränkungen der Schlüssellänge: Maximale Länge von 50.

Schlüssel-Muster: `[A-Za-z0-9_]+`

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

numResults

Die Anzahl der zurückzugebenden Ergebnisse. Der Standardwert ist 5. Das Maximum ist 100.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Erforderlich: Nein

userId

Die Benutzer-ID des Benutzers, für den Handlungsempfehlungen gegeben werden sollen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "actionList": [
    {
      "actionId": "string",
      "score": number
    }
  ],
  "recommendationId": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

actionList

Eine Liste von Handlungsempfehlungen, sortiert in absteigender Reihenfolge nach dem Prognosewert. Die Liste kann maximal 100 Aktionen enthalten. Informationen zu Maßnahmenwerten finden Sie unter [Funktionsweise der Bewertung von Aktionsempfehlungen](#).

Typ: Array von [PredictedAction](#)-Objekten

recommendationId

Die ID der Empfehlung.

Typ: Zeichenfolge

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)

- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

GetPersonalizedRanking

Service: Amazon Personalize Runtime

Sortiert eine Liste empfohlener Artikel für den angegebenen Benutzer neu. Das erste Element in der Liste wird als das Element angesehen, das für den Benutzer am wahrscheinlichsten von Interesse ist.

Note

Die der Kampagne zugrunde liegende Lösung muss mit einem Rezept vom Typ PERSONALIZED_RANKING erstellt worden sein.

Anforderungssyntax

```
POST /personalize-ranking HTTP/1.1
Content-type: application/json
```

```
{
  "campaignArn": "string",
  "context": {
    "string" : "string"
  },
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "inputList": [ "string" ],
  "metadataColumns": {
    "string" : [ "string" ]
  },
  "userId": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

campaignArn

Der Amazon-Ressourcenname (ARN) der Kampagne, die für die Erstellung des personalisierten Rankings verwendet werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]):personalize:.*:.*:.*+`

Erforderlich: Ja

context

Die kontextbezogenen Metadaten, die beim Abrufen von Empfehlungen verwendet werden sollen. Zu den kontextuellen Metadaten gehören alle Interaktionsinformationen, die für das Abrufen von Empfehlungen eines Benutzers relevant sein könnten, z. B. der aktuelle Standort oder der Gerätetyp des Benutzers.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 150 Elementen.

Einschränkungen der Schlüssellänge: Maximale Länge von 150.

Schlüssel-Muster: `[A-Za-z\d_]+`

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

filterArn

Der Amazon-Ressourcenname (ARN) eines Filters, den Sie erstellt haben, um Artikel für einen bestimmten Benutzer einzubeziehen oder Artikel von Empfehlungen auszuschließen. Weitere Informationen finden Sie unter [Filtern von Empfehlungen](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]):personalize:.*:.*:.*+`

Erforderlich: Nein

[filterValues](#)

Die Werte, die beim Filtern von Empfehlungen verwendet werden sollen. Geben Sie für jeden Platzhalterparameter in Ihrem Filterausdruck den Parameternamen (bei entsprechender Groß- und Kleinschreibung) als Schlüssel und die Filterwerte als entsprechenden Wert an. Trennen Sie mehrere Werte für einen Parameter durch ein Komma.

Für Filterausdrücke, die ein INCLUDE Element verwenden, um Elemente einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein EXCLUDE Element verwenden, um Artikel auszuschließen, können Sie das weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

Weitere Informationen finden Sie unter [Filtern von Empfehlungen](#).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 25 Elementen.

Einschränkungen der Schlüssellänge: Maximale Länge von 50.

Schlüssel-Muster: `[A-Za-z0-9_]+`

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

[inputList](#)

Eine Liste von Einträgen (`nextItemId`), denen eine Rangfolge zugewiesen werden soll. Wenn ein Element nicht im Trainingsdatensatz enthalten war, wird das Element an das Ende der neu eingestufteten Liste angehängt. Wenn Sie Metadaten in Empfehlungen aufnehmen, ist der Höchstwert 50. Andernfalls ist das Maximum 500.

Typ: Zeichenfolge-Array

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

[metadataColumns](#)

Wenn Sie bei der Erstellung oder Aktualisierung der Kampagne Metadaten in Empfehlungen aktiviert haben, geben Sie Metadaten-spalten aus Ihrem Artikeldatensatz an, die in das

personalisierte Ranking aufgenommen werden sollen. Der Map-Schlüssel ist ITEMS und der Wert ist eine Liste von Spaltennamen aus Ihrem Artikel-Datensatz. Die maximale Anzahl von Spalten, die Sie angeben können, ist 10.

Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Aktivieren von Metadaten in Empfehlungen für eine Kampagne](#).

Typ: Zeichenfolge zu Array von Zeichenfolgen-Mapping

Karteneinträge: Maximale Anzahl von 1 Element.

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 256.

Array-Mitglieder: Maximale Anzahl von 99 Elementen.

Längenbeschränkungen: Die maximale Länge beträgt 150.

Erforderlich: Nein

userId

Der Benutzer, für den die Kampagne ein personalisiertes Ranking bereitstellen soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "personalizedRanking": [
    {
      "itemId": "string",
      "metadata": {
        "string" : "string"
      },
      "promotionName": "string",
      "score": number
    }
  ]
}
```

```
    }  
  ],  
  "recommendationId": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

personalizedRanking

Eine Liste von Elementen in der Reihenfolge, in der sie für den Benutzer am wahrscheinlichsten von Interesse sind. Das Maximum ist 500.

Typ: Array von PredictedItem-Objekten

recommendationId

Die ID der Empfehlung.

Typ: Zeichenfolge

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

GetRecommendations

Service: Amazon Personalize Runtime

Gibt eine Liste empfohlener Artikel zurück. Für Kampagnen ist der Amazon-Ressourcenname (ARN) der Kampagne erforderlich, und die erforderlichen Benutzer- und Artikeleingaben hängen von dem Rezepttyp ab, mit dem die Kampagne unterstützt wird, wie folgt erstellt wurde:

- `USER_PERSONALIZATION` — `userId` erforderlich, wird nicht verwendet `itemId`
- `RELATED_ITEMS` — erforderlich, wird nicht verwendet `itemId` `userId`

Note

Kampagnen, die auf einer Lösung basieren, die mit einem Rezept vom Typ `PERSONALIZED_RANKING` erstellt wurde, verwenden die API. [GetPersonalizedRanking](#)

Für Empfehlungsgeber ist der ARN des Empfehlungsgebers erforderlich, und die erforderlichen Artikel- und Benutzereingaben hängen vom Anwendungsfall (domänenbasiertes Rezept) ab, der den Empfehlungsgeber unterstützt. Informationen zu den Anforderungen für Anwendungsfälle finden Sie unter Anwendungsfälle für Empfehlungsgeber [auswählen](#).

Anforderungssyntax

```
POST /recommendations HTTP/1.1
Content-type: application/json
```

```
{
  "campaignArn": "string",
  "context": {
    "string" : "string"
  },
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "itemId": "string",
  "metadataColumns": {
    "string" : [ "string" ]
  },
  "numResults": number,
```

```
"promotions": [  
  {  
    "filterArn": "string",  
    "filterValues": {  
      "string": "string"  
    },  
    "name": "string",  
    "percentPromotedItems": number  
  }  
],  
"recommenderArn": "string",  
"userId": "string"  
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

campaignArn

Der Amazon-Ressourcenname (ARN) der Kampagne, die zum Abrufen von Empfehlungen verwendet werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

context

Die kontextuellen Metadaten, die beim Abrufen von Empfehlungen verwendet werden sollen. Zu den kontextuellen Metadaten gehören alle Interaktionsinformationen, die für das Abrufen von Empfehlungen eines Benutzers relevant sein könnten, z. B. der aktuelle Standort oder der Gerätetyp des Benutzers.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 150 Elementen.

Einschränkungen der Schlüssellänge: Maximale Länge von 150.

Schlüssel-Muster: `[A-Za-z\d_]+`

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

[filterArn](#)

Der ARN des Filters, der auf die zurückgegebenen Empfehlungen angewendet werden soll. Weitere Informationen finden Sie unter [Filtern von Empfehlungen](#).

Wenn Sie diesen Parameter verwenden, stellen Sie sicher, dass die Filterressource ACTIVE

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

[filterValues](#)

Die Werte, die beim Filtern von Empfehlungen verwendet werden sollen. Geben Sie für jeden Platzhalterparameter in Ihrem Filterausdruck den Parameternamen (bei entsprechender Groß- und Kleinschreibung) als Schlüssel und die Filterwerte als entsprechenden Wert an. Trennen Sie mehrere Werte für einen Parameter durch ein Komma.

Für Filterausdrücke, die ein INCLUDE Element verwenden, um Elemente einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Bei Filtern mit Ausdrücken, die ein EXCLUDE Element verwenden, um Artikel auszuschließen, können Sie das weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

Weitere Informationen finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 25 Elementen.

Einschränkungen der Schlüssellänge: Maximale Länge von 50.

Schlüssel-Muster: [A-Za-z0-9_]+

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

itemId

Die Artikel-ID, für die Empfehlungen abgegeben werden sollen.

Erforderlich für den RELATED_ITEMS Rezepttyp.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

metadataColumns

Wenn Sie bei der Erstellung oder Aktualisierung der Kampagne oder Empfehlung Metadaten in Empfehlungen aktiviert haben, geben Sie die Metadaten­spalten aus Ihrem Artikel-Datensatz an, die in die Artikel­empfehlungen aufgenommen werden sollen. Der Map-Schlüssel ist ITEMS und der Wert ist eine Liste von Spalten­namen aus Ihrem Artikel-Datensatz. Die maximale Anzahl von Spalten, die Sie angeben können, ist 10.

Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Aktivieren von Metadaten in Empfehlungen für eine Kampagne](#). Informationen zum Aktivieren von Metadaten für einen Empfehlungsgeber finden Sie unter [Aktivieren von Metadaten in Empfehlungen für einen Empfehlungsgeber](#).

Typ: Zeichenfolge zu Array von Zeichenfolgen-Mapping

Karteneinträge: Maximale Anzahl von 1 Element.

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 256.

Array-Mitglieder: Maximale Anzahl von 99 Elementen.

Längenbeschränkungen: Die maximale Länge beträgt 150.

Erforderlich: Nein

numResults

Die Anzahl der zurückzugebenden Ergebnisse. Der Standardwert ist 25. Wenn Sie Metadaten in Empfehlungen einbeziehen, liegt der Höchstwert bei 50. Andernfalls ist der Höchstwert 500.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Erforderlich: Nein

promotions

Die Werbeaktionen, die für die Empfehlungsanfrage gelten sollen. Eine Werbeaktion definiert zusätzliche Geschäftsregeln, die für eine konfigurierbare Teilmenge empfohlener Artikel gelten.

Typ: Array von [Promotion](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 1 Element.

Erforderlich: Nein

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungsgebers, der zum Abrufen von Empfehlungen verwendet werden soll. Geben Sie einen Empfehlungs-ARN an, wenn Sie eine Domain-Datensatzgruppe mit einem Empfehlungsgeber für einen Domain-Anwendungsfall erstellt haben.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

userId

Die Benutzer-ID, für die Empfehlungen gegeben werden sollen.

Erforderlich für den USER_PERSONALIZATION Rezepttyp.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "itemList": [
    {
      "itemId": "string",
      "metadata": {
        "string" : "string"
      },
      "promotionName": "string",
      "score": number
    }
  ],
  "recommendationId": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

itemList

Eine Liste von Empfehlungen, sortiert in absteigender Reihenfolge nach dem Prognosewert. Die Liste kann maximal 500 Elemente enthalten.

Typ: Array von [PredictedItem](#)-Objekten

recommendationId

Die ID der Empfehlung.

Typ: Zeichenfolge

Fehler

InvalidInputException

Geben Sie einen gültigen Wert für das Feld oder den Parameter an.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

Datentypen

Die folgenden Datentypen werden von Amazon Personalize unterstützt:

- [Algorithm](#)
- [AlgorithmImage](#)
- [AutoMLConfig](#)
- [AutoMLResult](#)

- [BatchInferenceJob](#)
- [BatchInferenceJobConfig](#)
- [BatchInferenceJobInput](#)
- [BatchInferenceJobOutput](#)
- [BatchInferenceJobSummary](#)
- [BatchSegmentJob](#)
- [BatchSegmentJobInput](#)
- [BatchSegmentJobOutput](#)
- [BatchSegmentJobSummary](#)
- [Campaign](#)
- [CampaignConfig](#)
- [CampaignSummary](#)
- [CampaignUpdateSummary](#)
- [CategoricalHyperParameterRange](#)
- [ContinuousHyperParameterRange](#)
- [Dataset](#)
- [DatasetExportJob](#)
- [DatasetExportJobOutput](#)
- [DatasetExportJobSummary](#)
- [DatasetGroup](#)
- [DatasetGroupSummary](#)
- [DatasetImportJob](#)
- [DatasetImportJobSummary](#)
- [DatasetSchema](#)
- [DatasetSchemaSummary](#)
- [DatasetSummary](#)
- [DatasetUpdateSummary](#)
- [DataSource](#)
- [DefaultCategoricalHyperParameterRange](#)
- [DefaultContinuousHyperParameterRange](#)

- [DefaultHyperParameterRanges](#)
- [DefaultIntegerHyperParameterRange](#)
- [EventTracker](#)
- [EventTrackerSummary](#)
- [FeatureTransformation](#)
- [FieldsForThemeGeneration](#)
- [Filter](#)
- [FilterSummary](#)
- [HPOConfig](#)
- [HPOObjective](#)
- [HPOResourceConfig](#)
- [HyperParameterRanges](#)
- [IntegerHyperParameterRange](#)
- [MetricAttribute](#)
- [MetricAttribution](#)
- [MetricAttributionOutput](#)
- [MetricAttributionSummary](#)
- [OptimizationObjective](#)
- [Recipe](#)
- [RecipeSummary](#)
- [Recommender](#)
- [RecommenderConfig](#)
- [RecommenderSummary](#)
- [RecommenderUpdateSummary](#)
- [S3DataConfig](#)
- [Solution](#)
- [SolutionConfig](#)
- [SolutionSummary](#)
- [SolutionVersion](#)
- [SolutionVersionSummary](#)

- [Tag](#)
- [ThemeGenerationConfig](#)
- [TrainingDataConfig](#)
- [TunedHPOParams](#)

Die folgenden Datentypen werden von Amazon Personalize Events unterstützt:

- [Action](#)
- [ActionInteraction](#)
- [Event](#)
- [Item](#)
- [MetricAttribution](#)
- [User](#)

Die folgenden Datentypen werden von Amazon Personalize Runtime unterstützt:

- [PredictedAction](#)
- [PredictedItem](#)
- [Promotion](#)

Amazon Personalize

Die folgenden Datentypen werden von Amazon Personalize unterstützt:

- [Algorithm](#)
- [AlgorithmImage](#)
- [AutoMLConfig](#)
- [AutoMLResult](#)
- [BatchInferenceJob](#)
- [BatchInferenceJobConfig](#)
- [BatchInferenceJobInput](#)
- [BatchInferenceJobOutput](#)
- [BatchInferenceJobSummary](#)

- [BatchSegmentJob](#)
- [BatchSegmentJobInput](#)
- [BatchSegmentJobOutput](#)
- [BatchSegmentJobSummary](#)
- [Campaign](#)
- [CampaignConfig](#)
- [CampaignSummary](#)
- [CampaignUpdateSummary](#)
- [CategoricalHyperParameterRange](#)
- [ContinuousHyperParameterRange](#)
- [Dataset](#)
- [DatasetExportJob](#)
- [DatasetExportJobOutput](#)
- [DatasetExportJobSummary](#)
- [DatasetGroup](#)
- [DatasetGroupSummary](#)
- [DatasetImportJob](#)
- [DatasetImportJobSummary](#)
- [DatasetSchema](#)
- [DatasetSchemaSummary](#)
- [DatasetSummary](#)
- [DatasetUpdateSummary](#)
- [DataSource](#)
- [DefaultCategoricalHyperParameterRange](#)
- [DefaultContinuousHyperParameterRange](#)
- [DefaultHyperParameterRanges](#)
- [DefaultIntegerHyperParameterRange](#)
- [EventTracker](#)
- [EventTrackerSummary](#)
- [FeatureTransformation](#)

- [FieldsForThemeGeneration](#)
- [Filter](#)
- [FilterSummary](#)
- [HPOConfig](#)
- [HPOObjective](#)
- [HPOResourceConfig](#)
- [HyperParameterRanges](#)
- [IntegerHyperParameterRange](#)
- [MetricAttribute](#)
- [MetricAttribution](#)
- [MetricAttributionOutput](#)
- [MetricAttributionSummary](#)
- [OptimizationObjective](#)
- [Recipe](#)
- [RecipeSummary](#)
- [Recommender](#)
- [RecommenderConfig](#)
- [RecommenderSummary](#)
- [RecommenderUpdateSummary](#)
- [S3DataConfig](#)
- [Solution](#)
- [SolutionConfig](#)
- [SolutionSummary](#)
- [SolutionVersion](#)
- [SolutionVersionSummary](#)
- [Tag](#)
- [ThemeGenerationConfig](#)
- [TrainingDataConfig](#)
- [TunedHPOParams](#)

Algorithm

Bedienung: Amazon Personalize

Beschreibt einen benutzerdefinierten Algorithmus.

Inhalt

algorithmArn

Der Amazon-Ressourcenname (ARN) des Algorithmus.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

algorithmImage

Die URI des Docker-Containers für das Algorithmusbild.

Typ: [AlgorithmImage](#) Objekt

Required: No

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der der Algorithmus erstellt wurde.

Typ: Zeitstempel

Required: No

defaultHyperParameterRanges

Gibt die Standard-Hyperparameter und ihre Bereiche an und gibt an, ob sie einstellbar sind. Der Wert eines einstellbaren Hyperparameters kann während der Hyperparameter-Optimierung (HPO) bestimmt werden.

Typ: [DefaultHyperParameterRanges](#) Objekt

Required: No

defaultHyperParameters

Gibt die standardmäßigen Hyperparameter an.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuordnungseinträge: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellängenbeschränkungen: Maximale Länge von 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

defaultResourceConfig

Gibt die standardmäßige maximale Anzahl von Trainingsjobs und parallel Schulungsjobs an.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuordnungseinträge: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellängenbeschränkungen: Maximale Länge von 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeitinformationen), zu dem der Algorithmus zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Algorithmus.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

Der Amazon-Ressourcenname (ARN) der Rolle.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

trainingInputMode

Der Trainingseingabemodus.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

AlgorithmImage

Bedienung: Amazon Personalize

Beschreibt ein Algorithmus-Image.

Inhalt

dockerURI

Die URI des Docker-Containers für das Algorithmusbild.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

name

Der Name des Algorithmus-Image.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

AutoMLConfig

Bedienung: Amazon Personalize

Wenn die Lösung AutoML ausführt (`performAutoML` wahr in [CreateSolution](#)), ermittelt Amazon Personalize anhand der angegebenen Liste, welches Rezept die angegebene Metrik optimiert. Amazon Personalize verwendet dann dieses Rezept für die Lösung.

Inhalt

`metricName`

Die zu optimierende Metrik.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

`recipeList`

Die Liste der Kandidatenrezepte.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

AutoMLResult

Bedienung: Amazon Personalize

Wenn die Lösung AutoML (`performAutoML` true in [CreateSolution](#)) ausführt, gibt sie das Rezept an, das die angegebene Metrik am besten optimiert hat.

Inhalt

bestRecipeArn

Der Amazon-Ressourcenname (ARN) des besten Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchInferenceJob

Service: Amazon Personalize

Enthält Informationen zu einem Batch-Inferenzjob.

Inhalt

batchInferenceJobArn

Der Amazon-Ressourcenname (ARN) des Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

batchInferenceJobConfig

Eine String-zu-String-Zuordnung der Konfigurationsdetails eines Batch-Inferenzjobs.

Typ: [BatchInferenceJobConfig](#) Objekt

Erforderlich: Nein

batchInferenceJobMode

Der Modus des Jobs.

Typ: Zeichenfolge

Zulässige Werte: BATCH_INFERENCE | THEME_GENERATION

Erforderlich: Nein

creationDateTime

Der Zeitpunkt, zu dem der Batch-Inferenzjob erstellt wurde.

Typ: Zeitstempel

Erforderlich: Nein

failureReason

Wenn der Batch-Inferenzjob fehlgeschlagen ist, der Grund für den Fehler.

Typ: Zeichenfolge

Erforderlich: Nein

filterArn

Der ARN des Filters, der für den Batch-Inferenzjob verwendet wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

jobInput

Der Amazon S3 S3-Pfad, der zu den Eingabedaten führt, die zur Generierung des Batch-Inferenzjobs verwendet wurden.

Typ: [BatchInferenceJobInput](#) Objekt

Erforderlich: Nein

jobName

Der Name des Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

jobOutput

Der Amazon S3 S3-Bucket, der die vom Batch-Inferenzjob generierten Ausgabedaten enthält.

Typ: [BatchInferenceJobOutput](#) Objekt

Erforderlich: Nein

lastUpdatedDateTime

Der Zeitpunkt, zu dem der Batch-Inferenzjob zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

numResults

Die Anzahl der Empfehlungen, die durch den Batch-Inferenzjob generiert wurden. Diese Zahl beinhaltet die Fehlermeldungen, die für fehlgeschlagene Eingabedatensätze generiert wurden.

Typ: Ganzzahl

Erforderlich: Nein

roleArn

Der ARN der Amazon Identity and Access Management (IAM) -Rolle, die den Batch-Inferenzjob angefordert hat.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Erforderlich: Nein

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion, aus der der Batch-Inferenzjob erstellt wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status des Batch-Inferenzjobs. Der Status ist einer der folgenden Werte:

- PENDING
- IN BEARBEITUNG
- ACTIVE
- ERSTELLEN IST FEHLGESCHLAGEN

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

themeGenerationConfig

Die Einstellungen für die Theme-Generierung des Jobs.

Typ: [ThemeGenerationConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

BatchInferenceJobConfig

Bedienung: Amazon Personalize

Die Konfigurationsdetails einer Stapel-Inferenz-Aufgabe.

Inhalt

itemExplorationConfig

Eine Zeichenfolge zu einer Zeichenfolge, die die Hyperparameter der Explorationskonfiguration angibt `explorationItemAgeCutOff`, einschließlich `explorationWeight` und, die Sie verwenden möchten, um den Umfang der Artikelsuche zu konfigurieren, die Amazon Personalize bei der Empfehlung von Artikeln verwendet. Siehe [Benutzerpersonalisierung](#).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuweisungseinträge: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellänge: Maximale Länge von 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchInferenceJobInput

Bedienung: Amazon Personalize

Die Eingabekonfiguration eines Stapel-Inferenz-Jobs.

Inhalt

s3DataSource

Die URI des Amazon-S3-Speicherorts, der Ihre Eingabedaten enthält. Der Amazon S3 S3-Bucket muss sich im gleichen Region als API-Endpunkt, den Sie aufrufen.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchInferenceJobOutput

Bedienung: Amazon Personalize

Die Ausgabekonfigurationsparameter eines Batch-Inferenzjobs.

Inhalt

s3DataDestination

Informationen zum Amazon S3 S3-Bucket, in dem die Ausgabe des Batch-Inferenzjobs gespeichert ist.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchInferenceJobSummary

Service: Amazon Personalize

Eine gekürzte Version von [BatchInferenceJob](#). Der [ListBatchInferenceJobs](#) Vorgang gibt eine Liste von Zusammenfassungen von Batch-Inferenz-Jobs zurück.

Inhalt

batchInferenceJobArn

Der Amazon-Ressourcenname (ARN) des Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

batchInferenceJobMode

Der Modus des Jobs.

Typ: Zeichenfolge

Zulässige Werte: BATCH_INFERENCE | THEME_GENERATION

Erforderlich: Nein

creationDateTime

Der Zeitpunkt, zu dem der Batch-Inferenzjob erstellt wurde.

Typ: Zeitstempel

Erforderlich: Nein

failureReason

Wenn der Batch-Inferenzjob fehlgeschlagen ist, der Grund für den Fehler.

Typ: Zeichenfolge

Erforderlich: Nein

jobName

Der Name des Batch-Inferenzjobs.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

lastUpdatedDateTime

Der Zeitpunkt, zu dem der Batch-Inferenzjob zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

solutionVersionArn

Der ARN der Lösungsversion, die vom Batch-Inferenzjob verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status des Batch-Inferenzjobs. Der Status ist einer der folgenden Werte:

- PENDING
- IN BEARBEITUNG
- ACTIVE
- ERSTELLEN IST FEHLGESCHLAGEN

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

BatchSegmentJob

Bedienung: Amazon Personalize

Enthält Informationen zu einem Batch-Segment-Job.

Inhalt

batchSegmentJobArn

Der Amazon-Ressourcenname (ARN) des Batch-Segment-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

creationDateTime

Der Zeitpunkt, zu dem der Batch-Segment-Auftrag erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Der Grund für den Fehlschlag, wenn der Batch-Segment-Job fehlgeschlagen ist.

Typ: Zeichenfolge

Required: No

filterArn

Der ARN des Filters, der für den Batch-Segment-Job verwendet wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

jobInput

Der Amazon S3-Pfad, der zu den Eingabedaten führt, die zur Generierung des Batch-Segment-Jobs verwendet wurden.

Typ: [BatchSegmentJobInput](#) Objekt

Required: No

jobName

Der Name des Batch-Segment-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

jobOutput

Der Amazon S3-Bucket mit den durch den Batch-Segment-Job generierten Ausgabedaten.

Typ: [BatchSegmentJobOutput](#) Objekt

Required: No

lastUpdatedDateTime

Der Zeitpunkt, zu dem der Batch-Segment-Job zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

numResults

Die Anzahl der prognostizierten Benutzer, die durch den Batch-Segment-Job für jede Zeile von Eingabedaten generiert wurden. Die maximale Anzahl von Benutzern pro Segment beträgt 5 Millionen.

Typ: Ganzzahl

Required: No

roleArn

Der ARN der Amazon Identity and Access Management (IAM) -Rolle, die den Batch-Segment-Job angefordert hat.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\]+`

Erforderlich: Nein

solutionVersionArn

Der Amazon-Ressourcename (ARN) der Lösungsversion, die vom Batch-Segment-Job zum Generieren von Batch-Segmenten verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status des Batch-Segment-Auftrags. Der Status ist einer der folgenden Werte:

- PENDING
- IN ARBEIT
- ACTIVE
- ERSTELLEN IST FEHLGESCHLAGEN

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchSegmentJobInput

Bedienung: Amazon Personalize

Die Eingabekonfiguration eines Batch-Segment-Auftrags.

Inhalt

s3DataSource

Die Konfigurationsdetails eines Amazon S3 S3-Eingangs- oder Ausgabe-Buckets.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchSegmentJobOutput

Bedienung: Amazon Personalize

Die Ausgabekonfigurationsparameter eines Batch-Segment-Jobs.

Inhalt

s3DataDestination

Die Konfigurationsdetails eines Amazon S3 S3-Eingangs- oder Ausgabe-Buckets.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

BatchSegmentJobSummary

Bedienung: Amazon Personalize

Eine gekürzte Version des [BatchSegmentJob](#) Datentyps. [ListBatchSegmentJobs](#) Der Vorgang gibt eine Liste von Auftragszusammenfassungen für Batchsegmente zurück.

Inhalt

batchSegmentJobArn

Der Amazon-Ressourcenname (ARN) des Batch-Segment-Jobs.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

creationDateTime

Der Zeitpunkt, zu dem die Batch-Segment-Aufgabe erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn der Batch-Segment-Job ausgefallen ist, der Grund für den Fehlschlag.

Typ: Zeichenfolge

Required: No

jobName

The name

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

Der Zeitpunkt, zu dem die Batch-Segment-Aufgabe zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion, die vom Batch-Segment-Job zur Generierung von Batch-Segmenten verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

Der Status der Batch-Segmentaufgabe. Der Status ist einer der folgenden Werte:

- PENDING
- IN ARBEIT
- ACTIVE
- DAS ERSTELLEN IST AUSGEFALLEN

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Campaign

Bedienung: Amazon Personalize

Ein Objekt, das die Bereitstellung einer Lösungs-Version beschreibt. Weitere Informationen zu Kampagnen finden Sie unter [CreateCampaign](#).

Inhalt

campaignArn

Der Amazon-Ressourcennamen (ARN) der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

campaignConfig

Die Konfigurationsdetails einer Kampagne.

Typ: [CampaignConfig](#) Objekt

Required: No

creationDateTime

Datum und Uhrzeit (im Unix-Format), zu denen die Kampagne erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn eine Kampagne scheitert, der Grund für den Misserfolg.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit der letzten Aktualisierung der Kampagne (im Unix-Format).

Typ: Zeitstempel

Required: No

latestCampaignUpdate

Stellt eine Zusammenfassung der Eigenschaften eines Kampagnenupdates bereit. Rufen Sie die [DescribeCampaign](#) API auf, um eine vollständige Liste zu erhalten.

Typ: [CampaignUpdateSummary](#) Objekt

Required: No

minProvisionedTPS

Gibt die angeforderte Mindestanzahl bereitgestellter Transaktionen (Empfehlungen) pro Sekunde an. Ein HochminProvisionedTPS erhöht Ihre Rechnung. Wir empfehlen, mit 1 fürminProvisionedTPS (die Standardeinstellung) zu beginnen. Verfolgen Sie Ihre Nutzung mithilfe von CloudWatch Amazon-Metriken und erhöhen SieminProvisionedTPS diese bei Bedarf.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Required: No

name

Der Name der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

solutionVersionArn

Der Amazon-Ressourcenname (ARN) einer bestimmten Version der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status der Kampagne.

Eine Kampagne kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

CampaignConfig

Service: Amazon Personalize

Die Konfigurationsdetails einer Kampagne.

Inhalt

`enableMetadataWithRecommendations`

Ob Metadaten mit Empfehlungen für die Kampagne aktiviert sind. Wenn diese Option aktiviert ist, können Sie die Spalten aus Ihrem Artikeldatensatz in Ihrer Empfehlungsanfrage angeben. Amazon Personalize gibt diese Daten für jeden Artikel in der Empfehlungsantwort zurück. Informationen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Aktivieren von Metadaten in Empfehlungen für eine Kampagne](#).

Wenn Sie Metadaten in Empfehlungen aktivieren, entstehen Ihnen zusätzliche Kosten. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Preise](#).

Typ: Boolesch

Erforderlich: Nein

`itemExplorationConfig`

Gibt die Hyperparameter für die Explorationskonfiguration an, einschließlich `explorationWeight` und `explorationItemAgeCutOff`, die Sie verwenden möchten, um den Umfang der Artikelsuche zu konfigurieren, den Amazon Personalize bei der Empfehlung von Artikeln verwendet. [Geben Sie `itemExplorationConfig` Daten nur an, wenn Ihre Lösung das Rezept für Benutzerpersonalisierung verwendet.](#)

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 100 Elementen.

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

CampaignSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften einer Kampagne bereit. Rufen Sie die [DescribeCampaign](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

campaignArn

Der Amazon-Ressourcennamen (ARN) der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem die Kampagne erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn eine Kampagne fehlschlägt, der Grund für das Scheitern.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeiteinheiten), zu dem die Kampagne zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name der Kampagne.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status der Kampagne.

Eine Kampagne kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

CampaignUpdateSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Kampagnen-Updates bereit. Rufen Sie die [DescribeCampaignAPI](#) auf, um eine vollständige Liste zu erhalten.

Inhalt

campaignConfig

Die Konfigurationsdetails einer Kampagne.

Typ: [CampaignConfig](#) Objekt

Required: No

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem das Kampagnen-Update erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn ein Kampagnen-Update fehlschlägt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit) der letzten Aktualisierung des Kampagnen-Updates.

Typ: Zeitstempel

Required: No

minProvisionedTPS

Gibt die angeforderte Mindestanzahl bereitgestellter Transaktionen (Empfehlungen) pro Sekunde an, die Amazon Personalize unterstützt.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Required: No

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der bereitgestellten Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

status

Der Status des Kampagnen-Updates.

Ein Kampagnen-Update kann die folgenden Zustände aufweisen:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

CategoricalHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Bereich eines kategorialen Hyperparameters an.

Inhalt

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

values

Eine Liste der Kategorien für den Hyperparameter.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Längenbeschränkungen: Maximale Länge von 1000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ContinuousHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Bereich eines kontinuierlichen Hyperparameters an.

Inhalt

maxValue

Der maximal zulässige Wert für den Hyperparameter.

Type: Double

Gültiger Bereich: Mindestwert -1000000.

Required: No

minValue

Der zulässige Mindestwert für den Hyperparameter.

Type: Double

Gültiger Bereich: Mindestwert -1000000.

Required: No

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)

- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Dataset

Service: Amazon Personalize

Stellt Metadaten für einen Datensatz bereit.

Inhalt

creationDateTime

Datum und Uhrzeit der Erstellung (in Unix-Zeit) des Datensatzes.

Typ: Zeitstempel

Erforderlich: Nein

datasetArn

Der Amazon-Ressourcenname (ARN) des Datensatzes, für den Sie Metadaten benötigen.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

datasetType

Einer der folgenden Werte:

- Interaktionen
- Items

- Benutzer
- Aktionen
- Action_Interactions

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

lastUpdatedDateTime

Ein Zeitstempel, der anzeigt, wann der Datensatz aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

latestDatasetUpdate

Beschreibt die letzte Aktualisierung des Datensatzes.

Typ: [DatasetUpdateSummary](#) Objekt

Erforderlich: Nein

name

Der Name des Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

schemaArn

Der ARN des zugehörigen Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status des Datensatzes.

Ein Datensatz kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

trackingId

Die ID des Event-Trackers für einen Datensatz mit Aktionsinteraktionen. Sie geben die ID des Trackers im PutActionInteractions API-Vorgang an. Amazon Personalize leitet damit neue Daten an den Datensatz „Aktionsinteraktionen“ in Ihrer Datensatzgruppe weiter.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

DatasetExportJob

Bedienung: Amazon Personalize

Beschreibt einen Auftrag, die einen Auftrag, die einen Amazon-S3-Bucket verarbeitet wird. Weitere Informationen finden Sie unter [CreateDatasetExportJob](#).

Ein Export-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Inhalt

creationDateTime

Das Erstellungsdatum und die Uhrzeit (in Unix-Zeit) des Datensatz-Exportjobs.

Typ: Zeitstempel

Required: No

datasetArn

Der Amazon-Ressourcenname (ARN) des Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

datasetExportJobArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Export-Aufgabe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

failureReason

Wenn ein Dataset-Exportauftrag fehlschlägt, geben Sie den Grund dafür an.

Typ: Zeichenfolge

Required: No

ingestionMode

Die zu exportierenden Daten basieren darauf, wie Sie die Daten importiert haben. Sie können wählen, ob Sie BULK Daten exportieren möchten, die Sie mithilfe eines Datensatz-Importjobs importiert haben, PUT Daten, die Sie inkrementell importiert haben (mithilfe der `KonsolePutEvents`, `PutUsers` und `PutItems` Operationen), oder ALL für beide Typen. Der Standardwert ist PUT.

Typ: Zeichenfolge

Zulässige Werte: BULK | PUT | ALL

Required: No

jobName

Der Name des Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

jobOutput

Der Pfad zum Amazon S3 S3-Bucket, in dem die Auftragsausgabe gespeichert wird. Beispiel:

`s3://bucket-name/folder-name/`

Typ: [DatasetExportJobOutput](#) Objekt

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der der Status des Datensatz-Exportjobs zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

roleArn

Der Amazon-Ressourcenname (ARN) der IAM-Service-Rolle, die über Berechtigungen zum Hinzufügen von Daten zu Ihrem Amazon-S3-Bucket angenommen wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status der Datensatz-Export-Aufgabe.

Ein Export-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

DatasetExportJobOutput

Bedienung: Amazon Personalize

Die Ausgabekonfigurationsparameter eines Datensatz-Exportauftrags.

Inhalt

s3DataDestination

Die Konfigurationsdetails eines Amazon S3 S3-Eingangs- oder Ausgabe-Buckets.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DatasetExportJobSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Datensatz-Exportauftrags bereit. Rufen Sie die [DescribeDatasetExportJob](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der der Auftrag zum Exportieren von Datensätzen erstellt wurde.

Typ: Zeitstempel

Required: No

datasetExportJobArn

Der Amazon-Ressourcenname (ARN) des Import-Auftrag für Datensätze.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

failureReason

Wenn ein Dataset-Exportauftrag fehlschlägt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

jobName

Der Name des Import-Auftrag für Datensätze.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), zu dem der Import-Auftrag für Datensätze zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

status

Der Status des Import-Auftrag für Datensätze.

Ein Import-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DatasetGroup

Service: Amazon Personalize

Eine Datensatzgruppe ist eine Sammlung verwandter Datensätze (Artikelinteraktionen, Benutzer, Elemente, Aktionen, Aktionsinteraktionen). Sie erstellen eine Datensatzgruppe, indem Sie aufrufen [CreateDatasetGroup](#). Anschließend erstellen Sie einen Datensatz und fügen ihn einer Datensatzgruppe hinzu, indem Sie ihn aufrufen [CreateDataset](#). Die Datensatzgruppe wird verwendet, um eine Lösung durch Aufrufen zu erstellen und zu trainieren [CreateSolution](#). Eine Dataset-Gruppe kann von jeder Art von Dataset nur jeweils eine enthalten.

Sie können einen AWS Key Management Service (KMS-)Schlüssel angeben, um die Datensätze in der Gruppe zu verschlüsseln.

Inhalt

creationDateTime

Datum und Uhrzeit der Erstellung (in Unix-Zeit) der Datensatzgruppe.

Typ: Zeitstempel

Erforderlich: Nein

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

domain

Die Domäne einer Domäne-Datensatz-Gruppe.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Erforderlich: Nein

failureReason

Wenn das Erstellen einer Datensatzgruppe fehlschlägt, wird der Grund dafür angegeben.

Typ: Zeichenfolge

Erforderlich: Nein

kmsKeyArn

Der Amazon-Ressourcenname (ARN) des AWS Key Management Service (KMS)-Schlüssels, der zum Verschlüsseln der Datensätze verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 2 048.

Pattern: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Erforderlich: Nein

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung (in Unix-Zeit) der Datensatzgruppe.

Typ: Zeitstempel

Erforderlich: Nein

name

Der Name der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

roleArn

Der ARN der AWS Identity and Access Management (IAM) -Rolle, die über Berechtigungen für den Zugriff auf den AWS Key Management Service (KMS-) Schlüssel verfügt. Die Angabe einer IAM-Rolle ist nur gültig, wenn auch ein KMS-Schlüssel angegeben wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=, .@\-_/]+`

Erforderlich: Nein

status

Der aktuelle Status der Datensatzgruppe.

Eine Datensatzgruppe kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHEND LÖSCHEN

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

DatasetGroupSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften einer Datensatzgruppe bereit. Rufen Sie die [DescribeDatasetGroup](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit) der Erstellung der Datensatz-Gruppe.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der Amazon-Ressourcename (ARN) der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

domain

Die Domäne einer Domäne-Datensatz-Gruppe.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Required: No

failureReason

Wenn das Erstellen einer Datensatzgruppe fehlschlägt, ist dies der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung der Datensatz-Gruppe (in Unix-Zeit).

Typ: Zeitstempel

Required: No

name

Der Name der Datensatz-Gruppe.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

status

Der Status der Datensatz-Gruppe.

Eine Datensatz-Gruppe kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHENDER INFORMATION

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

DatasetImportJob

Bedienung: Amazon Personalize

Beschreibt einen Auftrag, der Trainingsdaten aus einer Datenquelle (Amazon-S3-Bucket) in einen Amazon-Personalize-Datensatz importiert. Weitere Informationen finden Sie unter [CreateDatasetImportJob](#).

Ein Import-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Inhalt

creationDateTime

Das Erstellungsdatum und die Uhrzeit (in Unix-Zeit) des Datensatz-Importjobs.

Typ: Zeitstempel

Required: No

datasetArn

Der Amazon-Ressourcenname (ARN) des Datensatzes , das die importierten Daten empfängt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

datasetImportJobArn

Der ARN des Datensatz-Import-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

dataSource

Der Amazon-S3-Bucket mit den zu importierenden Trainingsdaten.

Typ: [DataSource](#) Objekt

Required: No

failureReason

Wenn ein Dataset-Importjob fehlschlägt, geben Sie den Grund dafür an.

Typ: Zeichenfolge

Required: No

importMode

Der Importmodus, der vom Dataset-Importjob verwendet wird, um neue Datensätze zu importieren.

Typ: Zeichenfolge

Zulässige Werte: FULL | INCREMENTAL

Required: No

jobName

Der Name des Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

lastUpdatedDateTime

Das Datum und die Uhrzeit der letzten Aktualisierung des Datensatzes (in Unix-Zeit).

Typ: Zeitstempel

Required: No

publishAttributionMetricsToS3

Ob der Job Metriken für eine Metrikzuordnung in Amazon S3 veröffentlicht.

Typ: Boolesch

Required: No

roleArn

Der ARN der IAM-Rolle, die für das Lesen aus der Amazon-S3-Datenquelle berechtigt ist.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

status

Der Status des Datensatz-Import-Auftrags.

Ein Import-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DatasetImportJobSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Datensatz-Importauftrags bereit. Rufen Sie die [DescribeDatasetImportJob](#)API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der der Dataset-Importjob erstellt wurde.

Typ: Zeitstempel

Required: No

datasetImportJobArn

Der Amazon-Ressourcenname (ARN) des Datensatz-Import-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

failureReason

Wenn ein Dataset-Importjob fehlschlägt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

importMode

Der Importmodus, mit dem der Datensatz-Importjob die Daten im Datensatz aktualisiert hat. Weitere Informationen finden Sie unter [Aktualisierung vorhandener Massendaten](#).

Typ: Zeichenfolge

Zulässige Werte: FULL | INCREMENTAL

Required: No

jobName

Der Name des Datensatz-Import-Auftrags.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus des Datensatzes (in Unix-Zeit).

Typ: Zeitstempel

Required: No

status

Der Status des Datensatz-Import-Auftrags.

Ein Import-Auftrag für Datensätze kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

DatasetSchema

Service: Amazon Personalize

Beschreibt das Schema für einen Datensatz. Weitere Informationen zu Schemata finden Sie unter [CreateSchema](#).

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem das Schema erstellt wurde.

Typ: Zeitstempel

Erforderlich: Nein

domain

Die Domäne eines Schemas, das Sie für einen Datensatz in einer Domäne-Datensatz-Gruppe erstellt haben.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Erforderlich: Nein

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem das Schema zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

name

Der Name des Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

schema

Das Schema.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 20000.

Erforderlich: Nein

schemaArn

Der Amazon-Ressourcenname (ARN) des Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

DatasetSchemaSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Datensatzschemas bereit. Rufen Sie die [DescribeSchema](#)API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Das Datum und die Uhrzeit, zu dem das Schema erstellt wurde.

Typ: Zeitstempel

Required: No

domain

Die Domäne eines Schemas, das Sie für einen Datensatz in einer Domäne-Datensatz-Gruppe erstellt haben.

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit, zu dem das Schema zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

schemaArn

Der Amazon-Ressourcenname (ARN) des Schemas.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DatasetSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Datensatzes bereit. Rufen Sie die [DescribeDataset](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der der Datensatz erstellt wurde.

Typ: Zeitstempel

Required: No

datasetArn

Der Amazon-Ressourcenname (ARN) des Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

datasetType

Der Datensatztyp. Einer der folgenden Werte:

- Interaktionen
- Elemente
- Benutzer
- Event-Interaktionen

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit (in Unix-Zeiteinheiten) der letzten Aktualisierung des Dataset.

Typ: Zeitstempel

Required: No

name

Der Name des Datensatzes.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status des Dataset.

Ein Datensatz kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

DatasetUpdateSummary

Bedienung: Amazon Personalize

Beschreibt eine Aktualisierung eines Datensatzes.

Inhalt

creationDateTime

Das Erstellungsdatum und die Uhrzeit (in Unix-Zeit) des Datensatz-Updates.

Typ: Zeitstempel

Required: No

failureReason

Wenn die Aktualisierung eines Datensatzes fehlschlägt, geben Sie den Grund dafür an.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung (in Unix-Zeit) des Datensatzes.

Typ: Zeitstempel

Required: No

schemaArn

Der Amazon-Ressourcenname (ARN) des Schemas, das das vorherige Schema des Datensatzes ersetzt hat.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

status

Der Status der Datensatz-Aktualisierung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DataSource

Bedienung: Amazon Personalize

Beschreibt die Datenquelle, die die Daten enthält, die in einen Datensatz hochgeladen werden sollen.

Inhalt

dataLocation

Der Pfad zum Amazon S3 S3-Bucket, in dem die Daten gespeichert sind, die Sie in Ihren Datensatz hochladen möchten. Beispiel:

```
s3://bucket-name/folder-name/
```

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: (s3|http|https)://.+

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DefaultCategoricalHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Standardbereich eines kategorialen Hyperparameters an und gibt an, ob der Hyperparameter tunbar ist. Der Wert eines einstellbaren Hyperparameters kann während der Hyperparameter-Optimierung (HPO) bestimmt werden.

Inhalt

isTunable

Ob der Hyperparameter abstimmbar ist.

Typ: Boolesch

Required: No

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

values

Eine Liste der Kategorien für den Hyperparameter.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Längenbeschränkungen: Maximale Länge von 1000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DefaultContinuousHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Standardbereich eines kontinuierlichen Hyperparameters an und gibt an, ob der Hyperparameter tunbar ist. Der Wert eines einstellbaren Hyperparameters kann während der Hyperparameter-Optimierung (HPO) bestimmt werden.

Inhalt

isTunable

Ob der Hyperparameter abstimmbar ist.

Typ: Boolesch

Required: No

maxValue

Der maximal zulässige Wert für den Hyperparameter.

Type: Double

Gültiger Bereich: Mindestwert -1000000.

Required: No

minValue

Der zulässige Mindestwert für den Hyperparameter.

Type: Double

Gültiger Bereich: Mindestwert -1000000.

Required: No

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DefaultHyperParameterRanges

Bedienung: Amazon Personalize

Gibt die Hyperparameter und ihre Standardbereiche an. Hyperparameter können kategorische, kontinuierliche oder Ganzzahl-Werte sein.

Inhalt

categoryalHyperParameterRanges

Die kategorialen Hyperparameter und ihre Standardbereiche.

Typ: Array von [DefaultCategoricalHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

continuousHyperParameterRanges

Die kontinuierlichen Hyperparameter und ihre Standardbereiche.

Typ: Array von [DefaultContinuousHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

integerHyperParameterRanges

Die Hyperparameter mit ganzzahligen Werten und ihre Standardbereiche.

Typ: Array von [DefaultIntegerHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DefaultIntegerHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Standardbereich eines Integer-Hyperparameters an und gibt an, ob der Hyperparameter tunbar ist. Der Wert eines einstellbaren Hyperparameters kann während der Hyperparameter-Optimierung (HPO) bestimmt werden.

Inhalt

isTunable

Gibt an, ob der Hyperparameter abstimmbar ist.

Typ: Boolesch

Required: No

maxValue

Der maximal zulässige Wert für den Hyperparameter.

Typ: Ganzzahl

Gültiger Bereich: Maximalwert von 1000000.

Required: No

minValue

Der zulässige Mindestwert für den Hyperparameter.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert -1000000.

Required: No

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

EventTracker

Bedienung: Amazon Personalize

Stellt Informationen über einen Ereignis-Tracker bereit.

Inhalt

accountId

Das AWS Konto, dem der Event Tracker gehört.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

creationDateTime

Das Datum und die Uhrzeit (im Unix-Format), an dem der Event-Tracker erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe, die die Ereignis-Daten empfängt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

eventTrackerArn

Der ARN des Ereignis-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem der Ereignis-Tracker das letzte Mal aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Ereignis-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status des Ereignis-Trackers.

Ein Ereignis-Tracker kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

trackingId

Die Ereignis-ID. Geben Sie diese ID in Anfragen an die [PutEvents](#)API ein.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

EventTrackerSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Event-Trackers bereit. Rufen Sie die [DescribeEventTracker](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Das Datum und die Uhrzeit (in Unix-Zeit), zu der der Event-Tracker erstellt wurde.

Typ: Zeitstempel

Required: No

eventTrackerArn

Der Amazon-Ressourcenname (ARN) des Ereignis-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeitwert), an denen der Ereignis-Tracker zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Ereignis-Trackers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status des Ereignis-Trackers.

Ein Ereignis-Tracker kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

FeatureTransformation

Bedienung: Amazon Personalize

Stellt Informationen zur Feature-Transformation bereit. Funktionstransformierung ist der Prozess des Änderns unformatierter Eingabedaten in ein für die Modellschulung besser geeignetes Format.

Inhalt

creationDateTime

Das Erstellungsdatum und die Uhrzeit (in Unix-Zeit) der Feature-Transformation.

Typ: Zeitstempel

Required: No

defaultParameters

Stellt die Standardparameter für die Feature-Transformation bereit.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuweisen: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellängenfunktion: Maximale Länge von 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

featureTransformationArn

Der Amazon-Ressourcenname (ARN) des FeatureTransformation -Objekts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung (in Unix-Zeit) der Feature-Transformation.

Typ: Zeitstempel

Required: No

name

Der Name der Feature-Transformation.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status der Feature-Transformation.

Eine Feature-Transformation kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

FieldsForThemeGeneration

Service: Amazon Personalize

Eine String-zu-String-Zuordnung der Konfigurationsdetails für die Theme-Generierung.

Inhalt

itemName

Der Name der Datensatzspalte „Elemente“, in der die Namen der einzelnen Elemente im Datensatz gespeichert werden.

Typ: Zeichenfolge

Längenbeschränkungen: Die maximale Länge beträgt 150.

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

Filter

Bedienung: Amazon Personalize

Enthält Informationen zu einem Empfehlungsfilter, einschließlich seines ARN, Status und Filterausdrucks.

Inhalt

creationDateTime

Die Zeit, zu der der Filter erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der ARN der Datensatz-Gruppe, zu der der Filter gehört.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

failureReason

Wenn der Filter ausgefallen ist, der Grund für seinen Ausfall.

Typ: Zeichenfolge

Required: No

filterArn

Der ARN des Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

filterExpression

Gibt die Art der Artikelinteraktionen an, die aus den Empfehlungsergebnissen herausgefiltert werden sollen. Der Filterausdruck muss bestimmten Formatregeln folgen. Informationen zur Struktur und Syntax von Filterausdrücken finden Sie unter [Filterausdrücke](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 2500 Zeichen.

Required: No

lastUpdatedDateTime

Die Zeit, zu der der Filter zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Filters

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status des Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

FilterSummary

Bedienung: Amazon Personalize

Eine kurze Zusammenfassung der Attribute eines Filters.

Inhalt

creationDateTime

Die Zeit, zu der der Filter erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der ARN der Datensatz-Gruppe, zu der der Filter gehört.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

failureReason

Wenn der Filter ausgefallen ist, der Grund für den Fehlschlag.

Typ: Zeichenfolge

Required: No

filterArn

Der ARN des Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

Die Zeit, zu der der Filter zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Filters

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status des Filters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HPOConfig

Bedienung: Amazon Personalize

Beschreibt die Eigenschaften für die Hyperparameter-Optimierung (HPO).

Inhalt

algorithmHyperParameterRanges

Die Hyperparameter und ihre zulässigen Bereiche.

Typ: [HyperParameterRanges](#) Objekt

Required: No

hpoObjective

Die Metrik zur Optimierung während des HPO.

Note

Amazon Personalize unterstützt die Konfiguration von `hpoObjective` derzeit nicht.

Typ: [HPOObjective](#) Objekt

Required: No

hpoResourceConfig

Beschreibt die Ressourcenkonfiguration für HPO.

Typ: [HPOResourceConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)

- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HPOObjective

Bedienung: Amazon Personalize

Die Metrik, die während der Hyperparameter-Optimierung (HPO) optimiert werden soll.

Note

Amazon Personalize unterstützt die Konfiguration von `hpoObjective` derzeit nicht.

Inhalt

metricName

Name der Metrik.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

metricRegex

Ein regulärer Ausdruck, um die Metrik in den Protokollen der Trainingsaufträge zu finden.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

type

Der Typ der Metrik. Gültige Werte sind `Maximize` und `Minimize`.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HPOResourceConfig

Bedienung: Amazon Personalize

Beschreibt die Ressourcenkonfiguration für die Hyperparameter-Optimierung (HPO).

Inhalt

maxNumberOfTrainingJobs

Die maximale Anzahl von Trainingsjobs, wenn Sie eine Lösungsversion erstellen. Der Höchstwert für `maxNumberOfTrainingJobs` ist 40.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

maxParallelTrainingJobs

Die maximale Anzahl parallel Trainingsjobs, wenn Sie eine Lösungsversion erstellen. Der Höchstwert für `maxParallelTrainingJobs` ist 10.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HyperParameterRanges

Bedienung: Amazon Personalize

Gibt die Hyperparameter und ihre Bereiche an. Hyperparameter können kategorische, kontinuierliche oder Ganzzahl-Werte sein.

Inhalt

categoricalHyperParameterRanges

Die kategorialen Hyperparameter und ihre Bereiche.

Typ: Array von [CategoricalHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

continuousHyperParameterRanges

Die kontinuierlichen Hyperparameter und ihre Bereiche.

Typ: Array von [ContinuousHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

integerHyperParameterRanges

Die ganzzahligen Hyperparameter und ihre Bereiche.

Typ: Array von [IntegerHyperParameterRange](#)-Objekten

Array-Mitglieder: Maximale Anzahl von 100 Elementen.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

IntegerHyperParameterRange

Bedienung: Amazon Personalize

Gibt den Namen und den Bereich eines Integer-Hyperparameters an.

Inhalt

maxValue

Der maximal zulässige Wert für den Hyperparameter.

Typ: Ganzzahl

Gültiger Bereich: Maximalwert von 1000000.

Required: No

minValue

Der zulässige Mindestwert für den Hyperparameter.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert -1000000.

Required: No

name

Der Name des Hyperparameters.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)

- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MetricAttribute

Bedienung: Amazon Personalize

Enthält Informationen zu einer Metrik, über die eine Metrikzuordnung berichtet. Weitere Informationen finden Sie unter [Messen der Wirkung von Empfehlungen](#).

Inhalt

eventType

Der Ereignistyp der Metrik.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

expression

Der Ausdruck des Attributs. Verfügbare Funktionen sind `SUM()` oder `SAMPLECOUNT()`. Geben Sie für `SUM()`-Funktionen den Datensatztyp (entweder `Interactions` oder `Items`) und die zu summierende Spalte als Parameter an. Zum Beispiel `SUM(items.Price)`.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

metricName

Der Name der Metrik. Der Name hilft Ihnen, die Metrik in Amazon CloudWatch oder Amazon S3 zu identifizieren.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MetricAttribution

Bedienung: Amazon Personalize

Enthält Informationen zu einer metrischen Zuordnung. Eine Metrikzuordnung erstellt Berichte über die Daten, die Sie in Amazon Personalize importieren. Je nachdem, wie Sie die Daten importieren, können Sie Berichte in Amazon CloudWatch oder Amazon S3 anzeigen. Weitere Informationen finden Sie unter [Messen der Wirkung von -Empfehlungen](#).

Inhalt

creationDateTime

Datum und Uhrzeit der Erstellung der Metrikzuordnung.

Typ: Zeitstempel

Required: No

datasetGroupArn

Die Datensatz-Gruppe der Metri-Kennung Amazon-Ressourcenname (ARN) zurück.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

failureReason

Der Grund für das Scheitern der Metrikzuordnung.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung der Metrikzuordnung.

Typ: Zeitstempel

Required: No

metricAttributionArn

Der Amazon-Ressourcenname (ARN) der Metri-Kennung zurück.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

metricsOutputConfig

Die Ausgabekonfiguration der metrischen Zuordnung.

Typ: [MetricAttributionOutput](#) Objekt

Required: No

name

Der Name der metrischen Zuordnung.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status der Metrikzuordnung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MetricAttributionOutput

Bedienung: Amazon Personalize

Die Ausgabe-Konfigurationsdetails für eine metrische Zuordnung.

Inhalt

roleArn

Der Amazon-Ressourcenname (ARN) der IAM-Servicerin, die über die Berechtigungen verfügt, Daten zu Ihrem Amazon-S3-Ausgabe-Bucket hinzuzufügen und Metriken zu Amazon hinzuzufügen CloudWatch. Weitere Informationen finden Sie unter [Messen der Wirkung von - Empfehlungen](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=, .@-_/\]+`

Erforderlich: Ja

s3DataDestination

Die Konfigurationsdetails eines Amazon S3 S3-Eingangs- oder Ausgabe-Buckets.

Typ: [S3DataConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MetricAttributionSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften einer metrischen Zuordnung bereit. Für eine vollständige Liste rufen Sie die an [DescribeMetricAttribution](#).

Inhalt

creationDateTime

Datum und Uhrzeit der Erstellung der Metrikzuordnung.

Typ: Zeitstempel

Required: No

failureReason

Der Grund für das Scheitern der Metrikzuordnung.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung der Metrikzuordnung.

Typ: Zeitstempel

Required: No

metricAttributionArn

Der Amazon-Ressourcenname (ARN) der Metrik.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

name

Der Name der Metrik.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

Der Status der Metrikzuordnung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

OptimizationObjective

Bedienung: Amazon Personalize

Beschreibt das zusätzliche Ziel der Lösung, z. B. die Maximierung der Streaming-Minuten oder die Steigerung des Umsatzes. Weitere Informationen finden Sie unter [Optimierung einer Lösung](#).

Inhalt

itemAttribute

Die numerische Metadatenspalte in einem Items-Datensatz, die sich auf das Optimierungsziel bezieht. Zum Beispiel VIDEO_LENGTH (um die Streaming-Minuten zu maximieren) oder PRICE (um den Umsatz zu maximieren).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 100 Zeichen.

Required: No

objectiveSensitivity

Gibt an, wie Amazon Personalize die Bedeutung Ihres Optimierungsziels mit der Relevanz abwägt.

Typ: Zeichenfolge

Zulässige Werte: LOW | MEDIUM | HIGH | OFF

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Recipe

Bedienung: Amazon Personalize

Stellt Informationen zu einem Rezept bereit. Jedes Rezept enthält einen Algorithmus, den Amazon Personalize im Modelltraining verwendet, wenn Sie die [CreateSolution](#) Operation verwenden.

Inhalt

algorithmArn

Der Amazon-Ressourcenname (ARN) des Algorithmus, den Amazon Personalize zum Trainieren des Modells verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

creationDateTime

Datum und Uhrzeit (im Unix-Format), an dem das Rezept erstellt wurde.

Typ: Zeitstempel

Required: No

description

Die Beschreibung des Rezepts.

Typ: Zeichenfolge

Required: No

featureTransformationArn

Der ARN des FeatureTransformation Objekts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit (im Unix-Format) der letzten Aktualisierung des Rezepts.

Typ: Zeitstempel

Required: No

name

Der Name des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

Der Amazon-Ressourcenname (ARN) des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

recipeType

Einer der folgenden Werte:

- PERSONALISIERTE RANGFOLGE
- RELATED_ITEMS
- USER_PERSONALIZATION

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

status

Der Status des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

RecipeSummary

Bedienung: Amazon Personalize

Bietet eine Zusammenfassung der Eigenschaften einer Rezeptur. Rufen Sie die [DescribeRecipeAPI](#) auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der das Rezept erstellt wurde.

Typ: Zeitstempel

Required: No

domain

Die Domäne des Rezepts (wenn es sich bei dem Rezept um einen Anwendungsfall für eine Domänen-Datensatzgruppe handelt).

Typ: Zeichenfolge

Zulässige Werte: ECOMMERCE | VIDEO_ON_DEMAND

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit (in Unix-Zeitstatus).

Typ: Zeitstempel

Required: No

name

Der Name des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

recipeArn

Der Amazon-Ressourcenname (ARN) des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

status

Der Status des Rezepts.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Recommender

Bedienung: Amazon Personalize

Beschreibt einen Empfehlungsgenerator für eine Domäne-Datensatz-Gruppe. Sie erstellen in einer Domänen-Datensatzgruppe einen Empfehlungsgeber für einen bestimmten Domain-Anwendungsfall (Domain-Rezept) und geben den Empfehlenden in einer [GetRecommendations](#)Anfrage an.

Inhalt

creationDateTime

Datum und Uhrzeit (im Unix-Format), an dem der Empfehlungsgeber erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Domäne-Datensatz-Gruppe, die den Recommender enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-] +):personalize:.*:.*:.*+`

Required: No

failureReason

Wenn ein Empfehlungsgeber versagt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (im Unix-Format) der Empfehlungs-Version der letzten Aktualisierung.

Typ: Zeitstempel

Required: No

latestRecommenderUpdate

Stellt dem Empfehlungsgeber eine Zusammenfassung der neuesten Updates zur Verfügung.

Typ: [RecommenderUpdateSummary](#) Objekt

Required: No

modelMetrics

Stellt Bewertungsmetriken bereit, anhand derer Sie die Leistung eines Empfehlungsgebers ermitteln können. Weitere Informationen finden Sie unter [Bewerten eines -Empfehlungs-Kennung](#).

Typ: Zeichenfolge zu doppeltem Mapping

Zuweisungseinträge: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellänge: Maximale Länge von 256.

Required: No

name

Der Name des Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

Der Amazon-Ressourcenname (ARN) des Rezepts (Anwendungsfall für Domäne-Datensatz-Gruppen), für das der Empfehlungsname erstellt wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlungs-Kennung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

recommenderConfig

Die Konfigurationsdetails des -Empfehlungs-Kennung.

Typ: [RecommenderConfig](#) Objekt

Required: No

status

Der Status des Empfehlungsschreibers.

Ein Empfehlungsgeber kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHEND BEENDEN > IN_PROGRESS BEENDEN > INAKTIV > AUSSTEHEND STARTEN > IN_PROGRESS STARTEN > AKTIV
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

RecommenderConfig

Service: Amazon Personalize

Die Konfigurationsdetails des Empfehlungsgebers.

Inhalt

`enableMetadataWithRecommendations`

Ob Metadaten mit Empfehlungen für den Empfehlungsgeber aktiviert sind. Wenn diese Option aktiviert ist, können Sie die Spalten aus Ihrem Artikeldatensatz in Ihrer Empfehlungsanfrage angeben. Amazon Personalize gibt diese Daten für jeden Artikel in der Empfehlungsantwort zurück. Informationen zur Aktivierung von Metadaten für einen Empfehlungsgeber finden Sie unter [Aktivieren von Metadaten in Empfehlungen für einen](#) Empfehlungsgeber.

Wenn Sie Metadaten in Empfehlungen aktivieren, fallen zusätzliche Kosten an. Weitere Informationen finden Sie unter [Amazon Personalize Personalize-Preise](#).

Typ: Boolesch

Erforderlich: Nein

`itemExplorationConfig`

Gibt die Hyperparameter für die Explorationskonfiguration an, einschließlich `explorationWeight` und `explorationItemAgeCutOff`, die Sie verwenden möchten, um den Umfang der Artikelsuche zu konfigurieren, den Amazon Personalize bei der Empfehlung von Artikeln verwendet. Geben Sie nur `itemExplorationConfig` Daten an, wenn Ihre Empfehlungsgeber personalisierte Empfehlungen für einen Benutzer generieren (keine beliebten Artikel oder ähnliche Artikel).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 100 Elementen.

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Erforderlich: Nein

minRecommendationRequestsPerSecond

Gibt die angeforderte Mindestanzahl bereitgestellter Empfehlungsanfragen pro Sekunde an, die Amazon Personalize unterstützt. Ein Hoch erhöht `minRecommendationRequestsPerSecond` Ihre Rechnung. Wir empfehlen, mit 1 für zu beginnen `minRecommendationRequestsPerSecond` (Standard). Verfolgen Sie Ihre Nutzung anhand von CloudWatch Amazon-Metriken und erhöhen Sie `minRecommendationRequestsPerSecond` diese bei Bedarf.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Nein

trainingDataConfig

Gibt die Konfiguration der Trainingsdaten an, die bei der Erstellung eines Domain-Recommenders verwendet werden soll.

Typ: [TrainingDataConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

RecommenderSummary

Bedienung: Amazon Personalize

Bietet eine Zusammenfassung der Eigenschaften des Empfehlungsgebers.

Inhalt

creationDateTime

Datum und Uhrzeit (im Unix-Format), an dem der Empfehlungsgeber erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Domain-Datensatz-Gruppe, die den Empfehlungsgeber enthält.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

Das Datum und die Uhrzeit (im Unix-Format), zu der der Empfehlungsgeber zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

name

Der Name des Empfehlungsgebers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

Der Amazon-Ressourcenname (ARN) des Rezepts (Anwendungsfall für Domain-Datensatz-Gruppen), für das der Empfehlungsgeber erstellt wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

recommenderArn

Der Amazon-Ressourcenname (ARN) des Empfehlers.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

recommenderConfig

Die Konfigurationsdetails des Empfehlers.

Typ: [RecommenderConfig](#) Objekt

Required: No

status

Der Status des Empfehlungsgebers. Ein Empfehlungsgeber kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHEND BEENDEN > IN_PROGRESS BEENDEN > INAKTIV > AUSSTEHEND STARTEN > IN_PROGRESS STARTEN > AKTIV
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

RecommenderUpdateSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften eines Recommender-Updates bereit. Rufen Sie die [DescribeRecommenderAPI](#) auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (im Unix-Format), an dem das Recommender-Update erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn ein Empfehlungs-Update fehlschlägt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), zu dem das Recommender-Update zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

recommenderConfig

Die Konfigurationsdetails des Recommender-Updates.

Typ: [RecommenderConfig](#) Objekt

Required: No

status

Der Status des Empfehlungs-Updates.

Ein Empfehlungsgeber kann die folgenden Zustände aufweisen:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- AUSSTEHEND BEENDEN > IN_PROGRESS BEENDEN > INAKTIV > AUSSTEHEND STARTEN > IN_PROGRESS STARTEN > AKTIV
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

S3DataConfig

Bedienung: Amazon Personalize

Die Konfigurationsdetails eines Amazon S3-Eingabe- oder Ausgabe-Buckets.

Inhalt

path

Der Dateipfad des Amazon S3-Buckets.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: (s3|http|https)://.+

Erforderlich: Ja

kmsKeyArn

Der Amazon-Ressourcenname (ARN) des AWS Key Management Service (KMS) - Schlüssels, den Amazon Personalize zum Verschlüsseln oder Entschlüsseln der Eingabe- und Ausgabedateien verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 2 048.

Pattern: arn:aws.*:kms:.*:[0-9]{12}:key/.*

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Solution

Service: Amazon Personalize

Ein Objekt, das Informationen zu einer Lösung bereitstellt. Eine Lösung umfasst das benutzerdefinierte Rezept, benutzerdefinierte Parameter und trainierte Modelle (Lösungsversionen), die Amazon Personalize zur Generierung von Empfehlungen verwendet.

Inhalt

autoMLResult

Wenn „true“ performAutoML ist, wird das beste gefundene Rezept angegeben.

Typ: [AutoMLResult](#) Objekt

Erforderlich: Nein

creationDateTime

Das Erstellungsdatum und die Erstellungszeit (in Unix-Zeit) der Lösung.

Typ: Zeitstempel

Erforderlich: Nein

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe, die die Trainingsdaten bereitstellt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Erforderlich: Nein

eventType

Der Ereignistyp (z. B. „Klick“ oder „Gefällt mir“), der für das Training des Modells verwendet wird. Wenn eventType nicht bereitgestellt wird, verwendet Amazon Personalize alle Interaktionen für das Training mit gleichem Gewicht, unabhängig vom Typ.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

lastUpdatedDateTime

Das Datum und die Uhrzeit (in Unix-Zeit), zu der die Lösung zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

latestSolutionVersion

Beschreibt die neueste Version der Lösung, einschließlich Status und ARN.

Typ: [SolutionVersionSummary](#) Objekt

Erforderlich: Nein

name

Der Name der Lösung.


Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

performAutoML

 Important

Wir raten von der Verwendung von automatisiertem Machine Learning ab. Ordnen Sie Ihren Anwendungsfall stattdessen den verfügbaren Amazon-Personalize-Rezepten zu. Weitere Informationen finden Sie unter [Bestimmen Ihres Anwendungsfalls](#).

Wenn wahr, führt Amazon Personalize eine Suche nach dem besten USER_PERSONALIZATION-Rezept aus der in der Lösungs-Konfiguration angegebenen Liste durch (`recipeArn` darf nicht angegeben werden). Bei „false“ (Standard-Einstellung) verwendet Amazon Personalize `recipeArn` für das Training.

Typ: Boolesch

Erforderlich: Nein

performHPO

Gibt an, ob Hyperparameter-Optimierung (HPO) für das gewählte Rezept durchgeführt werden soll. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

recipeArn

Die ARN des Rezepts, das zur Erstellung der Lösung verwendet wurde. Dies ist erforderlich, wenn „false“ `performAutoML` ist.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

solutionArn

Der ARN der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

solutionConfig

Beschreibt die Konfigurations-Eigenschaften für die Lösung.

Typ: [SolutionConfig](#) Objekt

Erforderlich: Nein

status

Der Status der Lösung.

Eine Lösung kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

SolutionConfig

Bedienung: Amazon Personalize

Beschreibt die Konfigurations-Eigenschaften für die Lösung.

Inhalt

algorithmHyperParameters

Listet die Hyperparameter des Algorithmus und ihre Werte auf.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 100 Elementen.

Wichtige Längenbeschränkungen: Maximale Länge von 256.

Einschränkungen der Wertelänge: Maximale Länge von 1000.

Required: No

autoMLConfig

Das [AutoMLConfig](#)-Objekt, das eine Liste von Rezepten enthält, nach denen gesucht wird, wenn AutoML ausgeführt wird.

Typ: [AutoMLConfig](#) Objekt

Required: No

eventValueThreshold

Nur Ereignisse mit einem Wert größer als oder gleich dem Schwellenwert für die Schulung des Modells verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

featureTransformationParameters

Listet die Funktions-Transformationsparameter.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 100 Elementen.

Wichtige Längenbeschränkungen: Maximale Länge von 256.

Einschränkungen der Wertelänge: Maximale Länge von 1000.

Required: No

hpoConfig

Beschreibt die Eigenschaften für die Hyperparameter-Optimierung (HPO).

Typ: [HPOConfig](#) Objekt

Required: No

optimizationObjective

Beschreibt das zusätzliche Ziel der Lösung, z. B. die Maximierung der Streaming-Minuten oder die Steigerung des Umsatzes. Weitere Informationen finden Sie unter [Optimierung einer Lösung](#).

Typ: [OptimizationObjective](#) Objekt

Required: No

trainingDataConfig

Gibt die Trainingsdatenkonfiguration an, die beim Erstellen einer benutzerdefinierten Lösungsversion (trainiertes Modell) verwendet werden soll.

Typ: [TrainingDataConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

SolutionSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften einer Lösung bereit. Rufen Sie die [DescribeSolution](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), an dem die Lösung erstellt wurde.

Typ: Zeitstempel

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeitangabe) der letzten Aktualisierung der Lösung.

Typ: Zeitstempel

Required: No

name

Der Name der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

Der Amazon-Ressourcenname (ARN) des Rezepts für die Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

solutionArn

Den Amazon-Ressourcenname (ARN) der Lösung.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

status

Der Status der Lösung.

Eine -Anfrage kann die folgenden Zustände aufweisen:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

SolutionVersion

Bedienung: Amazon Personalize

Ein Objekt, das Informationen über eine bestimmte Version einer [Lösung](#) in einer benutzerdefinierten Datensatzgruppe bereitstellt.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der diese Version der Lösung erstellt wurde.

Typ: Zeitstempel

Required: No

datasetGroupArn

Der Amazon-Ressourcenname (ARN) der Datensatz-Gruppe, die die Trainingsdaten bereitstellt.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

eventType

Der Ereignistyp (z. B. „Klick“ oder „Gefällt mir“), der für das Training des Modells verwendet wird.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

failureReason

Wenn das Training einer Lösungsversion fehlschlägt, der Grund für den Fehlschlag.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit der letzten Aktualisierung des Lösung.

Typ: Zeitstempel

Required: No

name

Der Name der Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

performAutoML

Wenn dies zutrifft, sucht Amazon Personalize entsprechend der Lösungskonfiguration nach dem optimalsten Rezept. Bei „false“ (Standardeinstellung) verwendet `recipeArn` Amazon Personalize.

Typ: Boolesch

Required: No

performHPO

Gibt an, ob Hyperparameter-Optimierung (HPO) für das gewählte Rezept durchgeführt werden soll. Der Standardwert ist `false`.

Typ: Boolesch

Required: No

recipeArn

Der ARN des Rezepts, das in der Lösung verwendet wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

solutionArn

Der ARN der Lösung

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

solutionConfig

Beschreibt die Konfigurations-Eigenschaften für die Lösung.

Typ: [SolutionConfig](#) Objekt

Required: No

solutionVersionArn

Der ARN der Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Erforderlich: Nein

status

Der Status der Lösungsversion.

Eine Lösungsversion kann sich in einem der folgenden Zustände befinden:

- ERSTELLEN AUSSTEHENDER
- IN_PROGRESS ERSTELLEN
- ACTIVE
- DAS ERSTELLEN IST FEHLGESCHLAGEN

- STOPPEN ERSTELLEN
- ERSTELLEN WURDE GESTOPPT

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

trainingHours

Die Zeit, die zum Trainieren des Modells verwendet wurde. Ihnen wird die Zeit in Rechnung gestellt, die für das Trainieren eines Modells benötigt wird. Dieses Feld ist erst sichtbar, nachdem Amazon Personalize ein Modell erfolgreich trainiert hat.


Type: Double

Gültiger Bereich: Mindestwert 0.

Required: No

trainingMode

Der Umfang der Schulung, die bei der Erstellung der Lösungsversion durchgeführt werden muss. Die FULL Option trainiert die Lösungsversion auf der Grundlage der Gesamtheit der Trainingsdaten der Eingabelösung, während die UPDATE Option nur die Daten verarbeitet, die sich im Vergleich zur Eingabelösung geändert haben. Wählen Sie aus, UPDATE wann Sie Ihre Lösungsversion schrittweise aktualisieren möchten, anstatt eine völlig neue zu erstellen.

 Important

Die UPDATE Option kann nur verwendet werden, wenn Sie bereits über eine aktive Lösungsversion verfügen, die mithilfe der FULL Option aus der Eingabelösung erstellt wurde und die Eingabelösung mit dem Rezept für [Benutzerpersonalisierung oder dem HRNN-ColdStart-Rezept](#) trainiert wurde.

Typ: Zeichenfolge

Zulässige Werte: FULL | UPDATE

Required: No

tunedHPOParams

Wenn eine Hyperparameter-Optimierung durchgeführt wurde, enthält die Hyperparameterwerte des Modells mit der besten Leistung.

Typ: [TunedHPOParams](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

SolutionVersionSummary

Bedienung: Amazon Personalize

Stellt eine Zusammenfassung der Eigenschaften einer Lösungsversion bereit. Rufen Sie die [DescribeSolutionVersion](#) API auf, um eine vollständige Liste zu erhalten.

Inhalt

creationDateTime

Datum und Uhrzeit (in Unix-Zeit), zu der diese Version einer Lösung erstellt wurde.

Typ: Zeitstempel

Required: No

failureReason

Wenn eine Lösungsversion ausfällt, der Grund für den Fehler.

Typ: Zeichenfolge

Required: No

lastUpdatedDateTime

Datum und Uhrzeit (in Unix-Zeit), zu dem die Lösungsversion zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

solutionVersionArn

Der Amazon-Ressourcenname (ARN) der Lösungsversion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

Der Status der Lösungsversion.

Eine Lösungsversion kann sich in einem der folgenden Zustände befinden:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -oder- CREATE FAILED

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Tag

Service: Amazon Personalize

Die optionalen Metadaten, die Sie auf Ressourcen anwenden, um sie zu kategorisieren und zu organisieren. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen. Weitere Informationen finden Sie unter [Markieren von Amazon-Personalize-Ressourcen](#).

Inhalt

tagKey

Ein Teil eines Schlüssel-Wert-Paares, aus dem ein Tag besteht. Ein Schlüssel ist eine allgemeine Bezeichnung, die wie eine Kategorie für spezifischere Tag-Werte fungiert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_ .:/=+\-@]*)$`

Erforderlich: Ja

tagValue

Der optionale Teil eines Schlüssel-Wert-Paares, aus dem ein Tag besteht. Ein Wert fungiert als Deskriptor in einer Tag-Kategorie (Schlüssel).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge beträgt 256 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_ .:/=+\-@]*)$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS SDK für C++](#)

- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ThemeGenerationConfig

Service: Amazon Personalize

Die Konfigurationsdetails für die Generierung von Themen mit einem Batch-Inferenzjob.

Inhalt

fieldsForThemeGeneration

Felder, die zur Generierung beschreibender Themen für einen Batch-Inferenzjob verwendet werden.

Typ: [FieldsForThemeGeneration](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

TrainingDataConfig

Bedienung: Amazon Personalize

Die Konfiguration der Trainingsdaten, die bei der Erstellung einer Domainempfehlung oder einer benutzerdefinierten Lösungsversion (trainiertes Modell) verwendet werden soll.

Inhalt

excludedDatasetColumns

Gibt die Spalten an, die vom Training ausgeschlossen werden sollen. Jeder Schlüssel ist ein Datensatztyp, und jeder Wert ist eine Liste von Spalten. Schließen Sie Spalten aus, um zu kontrollieren, welche Daten Amazon Personalize zur Generierung von Empfehlungen verwendet. Sie könnten beispielsweise eine Spalte haben, die Sie nur zum Filtern von Empfehlungen verwenden möchten. Sie können diese Spalte von der Schulung ausschließen und Amazon Personalize berücksichtigt sie nur beim Filtern.

Typ: Zeichenfolge zu Array von Zeichenfolgen-Mapping

Karteneinträge: Maximale Anzahl von 3 Elementen.

Wichtige Längenbeschränkungen: Maximale Länge von 256.

Array-Mitglieder: Maximale Anzahl von 50 Elementen.

Längenbeschränkungen: Maximale Länge von 150.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

TunedHPOParams

Bedienung: Amazon Personalize

Wenn eine Hyperparameter-Optimierung (HPO) durchgeführt wurde, enthält sie die Hyperparameterwerte des Modells mit der besten Leistung.

Inhalt

algorithmHyperParameters

Eine Liste der Hyperparameterwerte des Modells mit der besten Leistung.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuweisen: Maximale Anzahl von 100 Elementen.

Beschränkungen der Schlüssellänge: Maximale Länge von 256.

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Amazon Personalize Events

Die folgenden Datentypen werden von Amazon Personalize Events unterstützt:

- [Action](#)
- [ActionInteraction](#)
- [Event](#)

- [Item](#)
- [MetricAttribution](#)
- [User](#)

Action

Service: Amazon Personalize Events

Stellt Aktionsmetadaten dar, die mithilfe der `PutActions` API zu einem Action-Datensatz hinzugefügt wurden. Weitere Informationen finden Sie unter [Aktionen einzeln importieren](#).

Inhalt

`actionId`

Die der Aktion zugeordnete ID.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

`properties`

Eine String-Map mit aktionsspezifischen Metadaten. Jedes Element in der Map besteht aus einem Schlüssel-Wert-Paar. Zum Beispiel `{"value": "100"}`.

Die Schlüssel verwenden Namen in Kamelform, die den Feldern im Schema für den Actions-Datensatz entsprechen. Im vorherigen Beispiel `value` entspricht das dem Feld 'VALUE', das im Actions-Schema definiert ist. Um bei kategorialen Zeichenkettendaten mehrere Kategorien für eine einzelne Aktion einzubeziehen, trennen Sie jede Kategorie durch ein Trennzeichen (`|`). Beispiel, `\\"DeLuxe|Premium\\"`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 32000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)

- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

ActionInteraction

Service: Amazon Personalize Events

Stellt ein Aktionsinteraktionsereignis dar, das über die PutActionInteractions API gesendet wurde.

Inhalt

actionId

Die ID der Aktion, mit der der Benutzer interagiert hat. Dies entspricht dem ACTION_ID Feld des Aktionsinteraktionsschemas.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

eventType

Der Typ des Aktionsinteraktionsereignisses. Sie können Not Taken Ereignistypen ViewedTaken, und angeben. Weitere Informationen zu Daten vom Typ Aktion, Interaktion und Ereignis finden Sie unter [Ereignistypdaten](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

sessionId

Die mit dem Besuch des Benutzers verknüpfte ID. Ihre Anwendung generiert eine eindeutige KennungsessionId, wenn ein Benutzer Ihre Website zum ersten Mal besucht oder Ihre Anwendung verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

timestamp

Der Zeitstempel, zu dem das Ereignis der Aktionsinteraktion eingetreten ist. Zeitstempel müssen im Zeitformat der Unix-Epoche in Sekunden angegeben werden.

Typ: Zeitstempel

Erforderlich: Ja

eventId

Eine dem Ereignis zugeordnete ID. Wenn keine Ereignis-ID angegeben wird, generiert Amazon Personalize eine eindeutige ID für das Ereignis. Eine Ereignis-ID wird nicht als Eingabe für das Modell verwendet. Amazon Personalize verwendet die Ereignis-ID, um eindeutige Ereignisse zu unterscheiden. Alle nachfolgenden Ereignisse nach dem ersten mit derselben Ereignis-ID werden beim Modelltraining nicht verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

impression

Eine Liste von Aktions-IDs, die die Reihenfolge der Aktionen darstellt, die Sie dem Benutzer gezeigt haben. Zum Beispiel ["actionId1", "actionId2", "actionId3"]. Amazon Personalize verwendet keine Impressionsdaten von Aktionsereignissen. Zeichnen Sie stattdessen mehrere Ereignisse für jede Aktion auf und verwenden Sie den Viewed Ereignistyp.

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Maximale Anzahl von 25 Elementen.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

properties

Eine Stringmap mit ereignisspezifischen Daten, die Sie möglicherweise aufzeichnen möchten. Wenn ein Benutzer beispielsweise eine andere Aktion als die Aktions-ID ausführt, können Sie auch die Anzahl der vom Benutzer ausgeführten Aktionen senden.

Jedes Element in der Map besteht aus einem Schlüssel-Wert-Paar. Zum Beispiel

```
{"numberOfActions": "12"}
```

Die Schlüssel verwenden Kamelnamen, die mit den Feldern im Aktionsinteraktionsschema übereinstimmen. Im obigen Beispiel `numberOfActions` würde das dem Feld `'NUMBER_OF_ACTIONS'` entsprechen, das im Aktionsinteraktionsschema definiert ist.

Folgendes darf nicht als Schlüsselwort für Eigenschaften verwendet werden (Groß- und Kleinschreibung wird nicht beachtet).

- `userId`
- `sessionId`
- `eventType`
- Zeitstempel
- ID der Empfehlung
- Eindruck

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Erforderlich: Nein

`recommendationId`

Die ID der Empfehlungsliste, die die Aktion enthält, mit der der Benutzer interagiert hat.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 40.

Erforderlich: Nein

`userId`

Die ID des Benutzers, der mit der Aktion interagiert hat. Dies entspricht dem `USER_ID` Feld des Aktionsinteraktionsschemas.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

Event

Service: Amazon Personalize Events

Stellt Informationen zu Artikelinteraktionsereignissen dar, die über die PutEvents API gesendet wurden.

Inhalt

eventType

Die Art des Ereignisses, z. B. Klick oder Download. Diese Eigenschaft entspricht dem EVENT_TYPE Feld im Schema Ihres Datensatzes „Artikelinteraktionen“ und hängt von den Ereignistypen ab, die Sie verfolgen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

sentAt

Der Zeitstempel (in Unix-Zeit) auf der Client-Seite, als das Ereignis eingetreten ist.

Typ: Zeitstempel

Erforderlich: Ja

eventId

Eine dem Ereignis zugeordnete ID. Wenn keine Ereignis-ID angegeben wird, generiert Amazon Personalize eine eindeutige ID für das Ereignis. Eine Ereignis-ID wird nicht als Eingabe für das Modell verwendet. Amazon Personalize verwendet die Ereignis-ID, um eindeutige Ereignisse zu unterscheiden. Alle nachfolgenden Ereignisse nach dem ersten mit derselben Ereignis-ID werden beim Modelltraining nicht verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

eventValue

Der Ereigniswert, der dem EVENT_VALUE Feld des Artikelinteraktionsschemas entspricht.

Typ: Float

Erforderlich: Nein

impression

Eine Liste von Artikel-IDs, die die Reihenfolge der Elemente darstellt, die Sie dem Benutzer angezeigt haben. Zum Beispiel ["itemId1", "itemId2", "itemId3"]. Stellen Sie eine Liste mit Elementen bereit, um Impressionsdaten für ein Ereignis manuell aufzuzeichnen.

Weitere Informationen zum Aufzeichnen von Impressionsdaten finden Sie unter [Aufzeichnen von Impressionsdaten](#).

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Maximale Anzahl von 25 Elementen.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

itemId

Der Element-ID-Schlüssel, der dem ITEM_ID Feld des Schemas des Datensatzes „Artikelinteraktionen“ entspricht.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

metricAttribution

Enthält Informationen über die Metrikzuweisung, die einem Ereignis zugeordnet ist. Weitere Informationen zu metrischen Attributionen finden Sie unter [Messung der Wirkung von Empfehlungen](#).

Typ: [MetricAttribution](#) Objekt

Erforderlich: Nein

properties

Eine Stringmap mit ereignisspezifischen Daten, die Sie möglicherweise aufzeichnen möchten. Wenn ein Nutzer beispielsweise einen Film auf Ihrer Website bewertet, bei dem es sich nicht um

die Film-ID (`itemId`) und die Bewertung (`eventValue`) handelt, können Sie auch die Anzahl der vom Nutzer abgegebenen Filmbewertungen angeben.

Jedes Element in der Map besteht aus einem Schlüssel-Wert-Paar. Zum Beispiel

```
{"numberOfRatings": "12"}
```

Die Schlüssel verwenden Kamelnamen, die mit den Feldern im Schema des Datensatzes „Artikelinteraktionen“ übereinstimmen. Im obigen Beispiel `numberOfRatings` würden sie dem Feld 'NUMBER_OF_RATINGS' entsprechen, das im Schema des Datensatzes „Artikelinteraktionen“ definiert ist.

Folgendes darf nicht als Schlüsselwort für Eigenschaften verwendet werden (Groß-/Kleinschreibung wird nicht beachtet).

- `userId`
- `sessionId`
- `eventType`
- Zeitstempel
- ID der Empfehlung
- Eindruck

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Erforderlich: Nein

`recommendationId`

Die ID der Empfehlungsliste, die das Element enthält, mit dem der Benutzer interagiert hat. Geben Sie `recommendationId` an, dass Amazon Personalize die Empfehlungen, die Sie Ihrem Benutzer zeigen, implizit als Impressionsdaten aufzeichnet. Oder geben Sie `anRecommendationId`, wenn Sie eine metrische Zuordnung verwenden, um die Wirkung von Empfehlungen zu messen.

Weitere Informationen zum Aufzeichnen von Impressionsdaten finden Sie unter [Aufzeichnen von Impressionsdaten](#). Weitere Informationen zur Erstellung einer metrischen Zuordnung finden Sie unter [Messung der Wirkung von Empfehlungen](#).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 40.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

Item

Service: Amazon Personalize Events

Stellt Elementmetadaten dar, die mithilfe der PutItems API zu einem Artikel-Datensatz hinzugefügt wurden. Weitere Informationen finden Sie unter [Artikel einzeln importieren](#).

Inhalt

itemId

Die dem Artikel zugeordnete ID.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

properties

Eine String-Map mit elementspezifischen Metadaten. Jedes Element in der Map besteht aus einem Schlüssel-Wert-Paar. Zum Beispiel {"numberOfRatings": "12"}.

Die Schlüssel verwenden Namen in Kamelform, die den Feldern im Schema für den Items-Datensatz entsprechen. Im vorherigen Beispiel numberOfRatings entspricht das dem Feld 'NUMBER_OF RATINGS', das im Artikelschema definiert ist. Um bei kategorialen Zeichenkettendaten mehrere Kategorien für ein einzelnes Element einzubeziehen, trennen Sie jede Kategorie durch ein Trennzeichen (|). | Beispiel, \"Horror|Action\".

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 32000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)

- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

MetricAttribution

Bedienung: Amazon Personalize Events

Enthält Informationen über eine Metrikzuordnung, die einem Ereignis zugeordnet ist. Weitere Informationen zu metrischen Attributen finden Sie unter [Messen der Wirkung von Empfehlungen](#).

Inhalt

eventAttributionSource

Die Quelle des Ereignisses, z. B. ein Dritter.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 1 024.

Pattern: `^[\x20-\x7E]*[\x21-\x7E]+[\x20-\x7E]*$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

User

Service: Amazon Personalize Events

Stellt Benutzermetadaten dar, die mithilfe der PutUsers API zu einem Benutzerdatensatz hinzugefügt wurden. Weitere Informationen finden Sie unter [Benutzer einzeln importieren](#).

Inhalt

userId

Die dem Benutzer zugeordnete ID.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

properties

Eine String-Map mit benutzerspezifischen Metadaten. Jedes Element in der Map besteht aus einem Schlüssel-Wert-Paar. Zum Beispiel {"numberOfVideosWatched": "45"}.

Die Schlüssel verwenden Namen in Kamelform, die den Feldern im Schema für den Benutzerdatensatz entsprechen. Im vorherigen Beispiel numberOfVideosWatched entspricht das dem Feld 'NUMBER_OF_VIDEOS_WATCHED', das im Benutzerschema definiert ist.

Um bei kategorialen Zeichenkettendaten mehrere Kategorien für einen einzelnen Benutzer einzubeziehen, trennen Sie jede Kategorie durch ein Trennzeichen (|). | Beispiel, \"Member | Frequent shopper\".

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 24000.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

Amazon Personalize Runtime

Die folgenden Datentypen werden von Amazon Personalize Runtime unterstützt:

- [PredictedAction](#)
- [PredictedItem](#)
- [Promotion](#)

PredictedAction

Service: Amazon Personalize Runtime

Ein Objekt, das eine Aktion identifiziert.

Die [GetActionRecommendations](#) API gibt eine Liste von PredictedAction s zurück.

Inhalt

actionId

Die ID der empfohlenen Aktion.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

score

Die Punktzahl der empfohlenen Maßnahme. Informationen zu Maßnahmenwerten finden Sie unter [So funktioniert die Bewertung von Aktionsempfehlungen](#).

Type: Double

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

PredictedItem

Service: Amazon Personalize Runtime

Ein Objekt, das einen Artikel identifiziert.

Die [GetPersonalizedRanking](#) APIs [GetRecommendations](#) und geben eine Liste von PredictedItems zurück.

Inhalt

itemId

Die empfohlene Artikel-ID.

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Erforderlich: Nein

metadata

Metadaten über den Artikel aus Ihrem Artikeldatensatz.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Einschränkungen der Schlüssellänge: Die maximale Länge beträgt 150.

Längenbeschränkungen für Werte: Maximale Länge von 20000.

Erforderlich: Nein

promotionName

Der Name der Werbeaktion, die den prognostizierten Artikel enthielt.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Erforderlich: Nein

score

Eine numerische Darstellung der Gewissheit des Modells, dass es sich bei dem Artikel um die nächste Benutzerauswahl handelt. Weitere Informationen zur Bewertungslogik finden Sie unter [Punktzahlen für Empfehlungen](#).

Type: Double

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

Promotion

Bedienung: Amazon Personalize Runtime

Enthält Informationen zu einer Werbeaktion. Eine Werbeaktion definiert zusätzliche Geschäftsregeln, die für eine konfigurierbare Teilmenge empfohlener Artikel gelten.

Inhalt

filterArn

Der Amazon-Ressourcenname (ARN) des Filters für diesen. Dieser Filter definiert die Kriterien für beworbene Artikel. Weitere Informationen finden Sie unter [Werbefilter](#).

Typ: Zeichenfolge

Längenbeschränkungen: Maximale Länge von 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

filterValues

Die Werte, die bei der Werbung für Artikel verwendet werden sollen. Geben Sie für jeden Platzhalterparameter im Filterausdruck Ihrer Werbeaktion den Parameternamen (in passender Groß-/Kleinschreibung) als Schlüssel und die Filterwerte als entsprechenden Wert an. Trennen Sie mehrere Werte für einen Parameter durch ein Komma.

Für Filterausdrücke, die ein `INCLUDE` Element verwenden, um Elemente einzuschließen, müssen Sie Werte für alle Parameter angeben, die im Ausdruck definiert sind. Für Filter mit Ausdrücken, die ein `EXCLUDE` Element verwenden, um Elemente auszuschließen, können Sie das weglassen `filter-values`. In diesem Fall verwendet Amazon Personalize diesen Teil des Ausdrucks nicht, um Empfehlungen zu filtern.

Weitere Informationen zum Erstellen von [Filtern finden Sie unter Empfehlungen und Benutzersegmente](#) filtern.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Zuweisungseinträge: Maximale Anzahl von 25 Elementen.

Längenbeschränkungen für Schlüssel: Maximale Länge von 50.

Schlüssel-Muster: `[A-Za-z0-9_]+`

Längenbeschränkungen für Werte: Maximale Länge von 1000.

Required: No

name

Der Name der Aktion.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 63 Zeichen.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

percentPromotedItems

Der Prozentsatz der empfohlenen Artikel, auf die die Aktion angewendet werden soll.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Häufige Fehler

In diesem Abschnitt sind Fehler aufgeführt, die häufig bei den API-Aktionen aller AWS-Services auftreten. Informationen zu Fehlern, die spezifisch für eine API-Aktion für diesen Service sind, finden Sie unter dem Thema für diese API-Aktion.

AccessDeniedException

Sie haben keinen ausreichenden Zugriff zum Durchführen dieser Aktion.

HTTP Status Code: 400

IncompleteSignature

Die Anforderungssignatur entspricht nicht den AWS-Standards.

HTTP Status Code: 400

InternalFailure

Die Anforderungsverarbeitung ist fehlgeschlagen, da ein unbekannter Fehler, eine Ausnahme oder ein Fehler aufgetreten ist.

HTTP Status Code: 500

InvalidAction

Die angeforderte Aktion oder Operation ist ungültig. Überprüfen Sie, ob die Aktion ordnungsgemäß eingegeben wurde.

HTTP Status Code: 400

InvalidClientTokenId

Das angegebene X.509-Zertifikat oder die AWS-Zugriffsschlüssel-ID ist nicht in unseren Datensätzen vorhanden.

HTTP Status Code: 403

NotAuthorized

Sie haben keine Berechtigung zum Ausführen dieser Aktion.

HTTP Status Code: 400

OptInRequired

Die AWS-Zugriffsschlüssel-ID benötigt ein Abonnement für den Service.

HTTP Status Code: 403

RequestExpired

Die Anforderung hat den Service mehr als 15 Minuten nach dem Datumsstempel oder mehr als 15 Minuten nach dem Ablaufdatum der Anforderung erreicht (z. B. für vorsignierte URLs) oder der Datumsstempel auf der Anforderung liegt mehr als 15 Minuten in der Zukunft.

HTTP Status Code: 400

ServiceUnavailable

Die Anforderung ist aufgrund eines temporären Fehlers des Servers fehlgeschlagen.

HTTP Status Code: 503

ThrottlingException

Die Anforderung wurde aufgrund der Drosselung von Anforderungen abgelehnt.

HTTP Status Code: 400

ValidationError

Die Eingabe erfüllt nicht die von einem AWS-Service definierten Einschränkungen.

HTTP Status Code: 400

Geläufige Parameter

Die folgende Liste enthält die Parameter, die alle Aktionen zum Signieren von Signature-Version-4-Anforderungen mit einer Abfragezeichenfolge verwenden. Alle aktionsspezifischen Parameter werden im Thema für diese Aktion aufgelistet. Weitere Informationen zu Signature Version 4 finden Sie unter [Signieren von AWS API-Anfragen](#) im IAM-Benutzerhandbuch.

Action

Die auszuführende Aktion.

Typ: Zeichenfolge

Erforderlich: Ja

Version

Die API-Version, für die die Anforderung geschrieben wurde, ausgedrückt im Format JJJJ-MM-TT.

Typ: Zeichenfolge

Erforderlich: Ja

X-Amz-Algorithm

Der Hashalgorithmus, den Sie zum Erstellen der Anforderungssignatur verwendet haben.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Zulässige Werte: AWS4-HMAC-SHA256

Required: Conditional

X-Amz-Credential

Der Wert des Anmeldeinformationsumfangs. Dabei handelt es sich um eine Zeichenfolge, die Ihren Zugriffsschlüssel, das Datum, die gewünschte Region und eine Zeichenfolge zur Beendigung („aws4_request“) beinhaltet. Der Wert wird im folgenden Format ausgedrückt: Zugriffsschlüssel/JJJJMMTT/Region/Service/aws4_request.

Weitere Informationen finden Sie unter [Erstellen einer signierten AWS API-Anfrage](#) im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Date

Das Datum, das zum Erstellen der Signatur verwendet wird. Das Format muss das ISO 8601-Basisformat (JJJJMMTT'T'SSMSS'Z') sein. Die folgende Datumszeit ist beispielsweise ein gültiger X-Amz-Date-Wert: 20120325T120000Z.

Bedingung: X-Amz-Date ist bei allen Anforderungen optional. Damit kann das Datum überschrieben werden, das zum Signieren von Anforderungen verwendet wird. Wenn der Date-Header im ISO 8601-Basisformat angegeben ist, ist X-Amz-Date nicht erforderlich. Wenn X-Amz-

Date verwendet wird, überschreibt es immer den Wert des Date-Headers. Weitere Informationen finden Sie unter [Elemente einer AWS API-Anforderungssignatur](#) im IAM-Benutzerhandbuch.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Security-Token

Das temporäre Sicherheitstoken, das durch einen Anruf von AWS Security Token Service (AWS STS) abgerufen wurde. Eine Liste der Services, die temporäre Sicherheitsanmeldeinformationen von unterstützen AWS STS [AWS-Services, finden Sie unter, die mit IAM arbeiten](#) im IAM-Benutzerhandbuch.

Bedingung: Wenn Sie temporäre Sicherheitsanmeldeinformationen von nutzen AWS STS, müssen Sie das Sicherheitstoken einschließen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Signature

Gibt die hex-codierte Signatur an, die aus der zu signierenden Zeichenfolge und dem abgeleiteten Signaturschlüssel berechnet wurde.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-SignedHeaders

Gibt alle HTTP-Header an, die als Teil der kanonischen Anforderung enthalten waren. Weitere Informationen zur Angabe signierter Header finden Sie unter [Erstellen einer signierten AWS API-Anfrage](#) im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

Dokumentenverlauf für Amazon Personalize

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des Amazon Personalize Developer Guide beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Neues Feature	Amazon Personalize unterstützt jetzt die Möglichkeit, einem Benutzer anhand seines Verhaltens die nächstbeste Aktion zu empfehlen. Weitere Informationen finden Sie unter Rezept für die nächste beste Aktion .	26. November 2023
Neues Feature	Amazon Personalize unterstützt jetzt mithilfe generativer KI die Möglichkeit, beschreibende Themen in Batch-Empfehlungen aufzunehmen. Informationen zum Generieren von Batch-Empfehlungen mit Themes finden Sie unter Batch-Empfehlungen mit Themes . Informationen zur Verwendung von Amazon Personalize mit generativer KI finden Sie unter Amazon Personalize und generative KI .	26. November 2023
Neues Feature	Amazon Personalize unterstützt jetzt die Möglichkeit, Metadaten aus einem Artikeldatensatz in Empfehlungen aufzunehmen. Informati	26. November 2023

onen zum Aktivieren von Metadaten für eine Kampagne finden Sie unter [Aktivieren von Metadaten in Empfehlungen für eine Kampagne](#).

Informationen zum Aktivieren von Metadaten für einen Empfehlungsgeber finden Sie unter [Aktivieren von Metadaten in Empfehlungen für einen Empfehlungsgeber](#).

Neues Feature

Amazon Personalize unterstützt jetzt die Möglichkeit, Suchergebnisse von OpenSearch Service zu personalisieren. Weitere Informationen finden Sie unter [Personalisieren von Suchergebnissen von OpenSearch](#)

16. Oktober 2023

Neues Feature

Amazon Personalize unterstützt jetzt den Import von bis zu 100 Metadaten spalten für Artikeldatensätze (bisher 50) und 25 Metadaten spalten für Benutzerdatensätze (bisher 5). Weitere Informationen zu Amazon Personalize-Limits finden Sie unter [Amazon Personalize Personalize-Endpunkte](#) und Kontingente.

5. September 2023

Neues Feature	Amazon Personalize unterstützt jetzt die Möglichkeit, Suchergebnisse von zu personalisieren. OpenSearch Weitere Informationen finden Sie unter Personalisierung von Suchergebnissen von OpenSearch (selbst verwaltet).	25. Juli 2023
Neues Feature	Amazon Personalize unterstützt jetzt die Möglichkeit, das Schema eines Datensatzes durch ein neues oder vorhandenes zu ersetzen. Weitere Informationen finden Sie unter Ersetzen des Schemas eines Datensatzes .	13. Juli 2023
Neues Feature	Wenn Sie die Rezepte „Benutzerpersonalisierung“ oder „Personalisiertes Ranking“ verwenden, können Amazon Personalize Personalize-Batch-Inferenzjobs jetzt Daten verwenden, die Sie inkrementell und ohne Umschulung importieren. Weitere Informationen finden Sie unter Batch-Empfehlungen abrufen.	30. Juni 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Möglichkeit, Artikel nach dem Artikel zu filtern, den Sie in Ihrer Anfrage nach Empfehlungen für verwandte Artikel angegeben haben. Informationen zu Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

21. Juni 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Möglichkeit, Artikel nach dem Artikel zu filtern, den Sie in Ihrer Anfrage nach Empfehlungen für verwandte Artikel angegeben haben. Informationen zu Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

21. Juni 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt private Verbindungen zwischen einer Virtual Private Cloud (VPC) und Amazon Personalize mit einer Amazon VPC-Schnittstelle. Weitere Informationen finden Sie unter [Amazon Personalize and interface VPC-Endpoints](#) ().
AWS PrivateLink

12. Juni 2023

Neues Feature

Amazon Personalize unterstützt jetzt die Konfiguration der Spalten, die beim Training verwendet werden, wenn Sie eine Empfehlung oder eine benutzerdefinierte Lösung erstellen. [Informationen zur Konfiguration von Spalten bei der Erstellung eines Empfehlungsgebers finden Sie unter Empfehlungen erstellen](#). Informationen zur Konfiguration von Spalten bei der Erstellung einer Lösung finden Sie unter [Konfiguration der beim Training verwendeten Spalten](#).

30. Mai 2023

Neue Dokumentationsfunktion

Das Amazon Personalize Personalize-Entwicklerhandbuch enthält jetzt Informationen zur Durchführung von A/B-Tests mit Amazon Personalize Personalize-Empfehlungen. Weitere Informationen finden Sie unter [Messung der Wirkung von Empfehlungen mit A/B-Tests](#).

5. Mai 2023

Neues Feature

Amazon Personalize unterstützt jetzt die Konfiguration, wie die Beliebtheit die Empfehlungen beeinflusst, die nach dem Rezept „Ähnliche Artikel“ generiert werden. Weitere Informationen finden Sie unter Rezept für [ähnliche](#) Artikel.

21. April 2023

Neues Feature

Amazon Personalize unterstützt jetzt die Verwendung von Amazon SageMaker Data Wrangler, um Daten aus über 40 Quellen in Amazon Personalize-Datensätze zu importieren. Weitere Informationen finden Sie unter [Daten mit Amazon SageMaker Data Wrangler importieren](#).

14. April 2023

Neue Dokumentationsfunktion

Das Amazon Personalize Personalize-Entwicklerhandbuch enthält jetzt eine neue Checkliste für die Eignung. Diese Checkliste hilft Ihnen, sich auf die Verwendung von Amazon Personalize mit Ihren eigenen Daten vorzubereiten. Weitere Informationen finden Sie unter Checkliste [zur Eignung](#).

9. Februar 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Generierung von Erkenntnissen und Statistiken für Daten, die Sie in Datensätze importieren. Weitere Informationen finden Sie unter [Analysieren von Daten in Datensätzen](#).

25. Januar 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt ein neues Trending-Now-Rezept für benutzerdefinierte Datensatzgruppen. [Weitere Informationen finden Sie unter Trending-Now-Rezept](#).

06. Januar 2023

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Verwendung von Tags in IAM-Richtlinien. Weitere Informationen finden Sie unter [Verwenden von Tags in IAM-Richtlinien](#).

28. Dezember 2022

Neues Feature

Amazon Personalize unterstützt jetzt Anpassungen an der maximalen Anzahl von Interaktionen, die von einem Modell während der Trainingsquote berücksichtigt werden. Darüber hinaus gilt die Obergrenze für die maximale Anzahl von Benutzern, die von einem Modell während des Trainings berücksichtigt werden, nicht mehr. Weitere Informationen finden Sie unter [Amazon Personalize Endpoints and Quotas](#).

15. Dezember 2022

Neues Feature

Amazon Personalize unterstützt jetzt die Erstellung einer metrischen Zuordnung, um die geschäftlichen Auswirkungen von Empfehlungen zu messen. Weitere Informationen finden Sie unter [Messung der Wirkung von Empfehlungen](#).

17. November 2022

[Neues Feature](#)

Die Amazon Personalize Personalize-Kontingente für die Gesamtzahl der aktiven Lösungen, aktiven Kampagnen, Empfehlungen und Filter wurden jetzt erhöht. Jedes dieser Kontingente gilt jetzt pro Datensatzgruppe und nicht pro Konto. Informationen zu Kontingenten finden Sie unter [Amazon Personalize Endpoints and Quotas](#).

7. September 2022

[Neues Feature](#)

Amazon Personalize Personalize-Filter berücksichtigen jetzt bis zu 100 Interaktionen pro Benutzer und Ereignistyp. Informationen zu Filtern finden Sie unter [Filtern von Empfehlungen und Benutzersegmenten](#).

2p. August 2022

[Neues Feature](#)

Amazon Personalize unterstützt jetzt einen neuen Anwendungsfall „Trending now“ für die Domain VIDEO_ON_DEMAND. [Weitere Informationen finden Sie unter Anwendungsfälle für VIDEO_ON_DEMAND](#).

17. August 2022

[Neues Feature](#)

Amazon Personalize unterstützt jetzt das Bewerben von Artikeln in Empfehlungen mit einem separaten Werbefilter. Informationen zum Bewerben von Artikeln finden Sie unter [Artikel in Empfehlungen bewerben](#).

12. August 2022

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Verwendung von Vergleichsoperatoren in Filterausdrücken mit Platzhalterparametern. Informationen zu Filterausdrücken finden Sie unter [Filterausdrücke](#).

12. August 2022

[Neues Feature](#)

Amazon Personalize unterstützt jetzt inkrementelle Massenaktualisierungen von Datensätzen. Sie können jetzt einen Datensatz-Importjob verwenden, um einen Datensatz zu aktualisieren, ohne die vorhandenen Daten zu ersetzen. Weitere Informationen finden Sie unter [Aktualisieren vorhandener Massendaten](#).

02. August 2022

[Neues Feature](#)

Amazon Personalize kann jetzt unstrukturierte Textmetadaten in verschiedenen Sprachen verwenden. Weitere Informationen finden Sie unter [Unstrukturierte Textmetadaten](#).

6. Juni 2022

[Neues Feature](#)

Amazon Personalize Personalize-Empfehlungen generieren jetzt Offline-Metriken. Sie können diese Kennzahlen verwenden, um die Leistung Ihres Empfehlungsgebers zu bewerten. Weitere Informationen finden Sie unter [Bewertung eines Empfehlungsgebers](#).

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Möglichkeit, eine Empfehlung zu beenden und sie später neu zu starten. Auf diese Weise können Sie die Fakturierung von Empfehlungen pausieren und sie nur dann bezahlen, wenn Sie sie nutzen. Weitere Informationen findest du unter [Einen Empfehlungsgeber beenden und starten](#).

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Verwendung von Tags zur Kategorisierung und Verwaltung von Amazon Personalize Personalize-Ressourcen. Weitere Informationen finden Sie unter [Tagging Amazon Personalize Personalize-Ressourcen](#).

Neues Feature	Amazon Personalize unterstützt jetzt die Angabe von Ressourcen mit AWS CloudFormation. Weitere Informationen finden Sie unter Ressourcen angeben mit AWS CloudFormation .	11. März 2022
Neue Dokumentationsfunktion	Das Amazon Personalize-Entwicklerhandbuch enthält jetzt ein neues Thema zur Fehlerbehebung, das Antworten auf häufig gestellte Fragen und Hinweise zur Fehlerbehebung bei Fehlermeldungen bietet, die bei Amazon Personalize auftreten können. Weitere Informationen finden Sie unter Fehlerbehebung .	15. Februar 2022
Neues Feature	Amazon Personalize unterstützt jetzt die Erstellung einer Domain-Datensatzgruppe mit nutzungsfalloptimierten Ressourcen für Video-on-Demand- oder E-Commerce-Domains. Weitere Informationen finden Sie unter Domain-Datensatzgruppen .	29. November 2021

Neues Feature

Amazon Personalize unterstützt jetzt die Erstellung von Benutzersegmenten mit neuen USER_SEGMENTATION-Rezepten. USER_SEGMENTATION-Rezepte generieren Benutzersegmente auf der Grundlage von Artikelengineeringdaten. [Weitere Informationen finden Sie unter USER_SEGMENTATION-Rezepte.](#)

29. November 2021

Neues Feature

Amazon Personalize unterstützt jetzt ein neues RELATED_ITEMS-Empfehlungsrezept Similar-Items. Verwenden Sie das Rezept „Ähnliche Artikel“, um Empfehlungen für ähnliche Artikel zu generieren, die sowohl auf Interaktionsdaten als auch auf Artikelmetadaten basieren. Weitere Informationen finden Sie unter Rezept für [ähnliche Artikel](#).

5. Oktober 2021

Neue Dokumentationsfunktion

Das Amazon Personalize-Entwicklerhandbuch enthält jetzt ein Tutorial für die ersten Schritte zur Verwendung von Amazon Personalize mit dem SDK for Java 2.x. Weitere Informationen finden Sie unter [Erste Schritte \(SDK for Java 2.x\)](#).

25. August 2021

[Neues Feature](#)

Amazon Personalize kann jetzt aussagekräftige Informationen aus unstrukturierten Textmetadaten in einem Artikeldatensatz extrahieren. Weitere Informationen finden Sie unter [Artikeldatensatz](#).

9. Juni 2021

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Möglichkeit, die Erstellung einer Lösungsversion zu beenden (das Trainieren eines Modells zu beenden). Weitere Informationen finden Sie unter [Stoppen der Erstellung einer Lösungsversion](#).

20. Mai 2021

[Neue Funktion \(Vorschauversion\)](#)

Amazon Personalize kann jetzt eine Lösung für ein bestimmtes Ziel optimieren und gleichzeitig die Relevanz maximieren, z. B. die Umsatzmaximierung. Diese Funktion befindet sich in der Vorschauversion. Weitere Informationen finden Sie unter [Optimieren einer Lösung für ein zusätzliches Ziel](#).

18. Mai 2021

[Neues Feature](#)

Amazon Personalize kann jetzt die Datensätze in einem Amazon Personalize Personalize-Datensatz zur Analyse und Nachverfolgung in einen Amazon S3 S3-Bucket exportieren. Weitere Informationen finden Sie unter [Exportieren eines Datensatzes](#).

26. April 2021

[Neues Feature](#)

Amazon Personalize aktualisiert jetzt automatisch alle zwei Stunden das neueste Modell (Lösungsversion), das Sie mit User-Personalizer trainiert haben, um neue Daten aufzunehmen. [Weitere Informationen finden Sie unter Rezept für die Benutzerpersonalisierung](#).

17. November 2020

[Neues Feature](#)

Amazon Personalize kann Empfehlungen jetzt anhand von Kriterien filtern, die Sie angeben, wenn Sie Empfehlungen erhalten. Weitere Informationen finden Sie unter [Empfehlungen filtern](#).

10. November 2020

[Neues Feature](#)

Amazon Personalize unterstützt jetzt die Möglichkeit, Benutzer und Artikel inkrementell zu importieren. Weitere Informationen finden Sie unter Inkrementelles [Importieren von Datensätzen](#).

2. Oktober 2020

[Neues Feature](#)

Amazon Personalize unterstützt jetzt ein neues USER_PERSONALIZATION-Empfehlungsrezept. Zu den Funktionen von USER_PERSONALIZATION gehören die Modellierung von Impressionsdaten, die automatische Artikelsuche und die automatische Kaltauswahl von Artikeln. [Weitere Informationen finden Sie unter Rezept für Benutzerpersonalisierung](#).

5. August 2020

[Neues Feature](#)

Amazon Personalize kann jetzt Empfehlungen anhand von Artikel- und Benutzermetadaten mithilfe benutzerdefinierter Filterausdrücke filtern. Weitere Informationen finden Sie unter [Empfehlungen zum Filtern](#).

31. Juli 2020

[Neues Feature](#)

Amazon Personalize ermöglicht es Ihnen jetzt, Ergebnisse danach zu filtern, mit welchen Elementen ein Benutzer interagiert hat. Weitere Informationen finden Sie unter [Empfehlungen zum Filtern](#).

3. Juni 2020

Neues Feature

Amazon Personalize zeigt jetzt Bewertungen für empfohlene Artikel an. Die Punktzahlen stehen für die Gewissheit des Amazon Personalize Personalize-Modells, dass ein Benutzer als Nächstes einen bestimmten Artikel auswählt. Weitere Informationen finden Sie unter [Empfehlungen einholen](#).

Neue Region

Amazon Personalize bietet Unterstützung für die Region Asien-Pazifik (Seoul). Eine vollständige Liste der von Amazon Personalize unterstützten AWS Regionen finden Sie in der [AWSRegionentabelle](#) oder [AWSRegionen und Endpunkte](#) in der Amazon Web Services General Reference.

21. Januar 2020

Neues Feature

Amazon Personalize kann jetzt Empfehlungen auf der Grundlage kontextbezogener Metadaten abrufen. [Weitere Informationen finden Sie unter Empfehlungen abrufen](#).

19. Dezember 2019

Neue Regionen	Amazon Personalize bietet Unterstützung für die Regionen Asien-Pazifik (Mumbai), Asien-Pazifik (Sydney) und Kanada (Zentral). Eine vollständige Liste der von Amazon Personalize unterstützten AWS Regionen finden Sie in der AWSRegionentabelle oder AWSRegionen und Endpunkte in der Amazon Web Services General Reference.	18. Dezember 2019
Neues Feature	Amazon Personalize unterstützt jetzt Workflows für Batch-Empfehlungen. Weitere Informationen finden Sie unter Batch-Empfehlungen abrufen .	14. November 2019
Allgemeine Verfügbarkeit von Amazon Personalize	Amazon Personalize ist jetzt für den allgemeinen Gebrauch verfügbar.	10. Juni 2019
Vorabversion von Amazon Personalize	Dies ist die erste Vorschauversion der Dokumentation für Amazon Personalize.	28. November 2018

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.