
Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker

Implementation Guide



Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker: Implementation Guide

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Home	1
Overview	2
Cost	2
Architecture	3
Components	5
Amazon SageMaker	5
Dataset	5
Design considerations	6
Customization	6
Regional Deployment	6
AWS CloudFormation template	7
Automated deployment	8
Prerequisites	8
Updating the stack from version 1.0.1 or earlier	8
Deploy the Digital User Engagement Events Database solution	8
Deployment overview	9
Step 1. Launch the stack	9
Step 2. (Optional) Upload the sample Pinpoint segment	11
Step 3. Run the notebook	11
Step 4. Run the state machine	12
Step 5. Verify the segment exists and is receiving records	12
Security	13
IAM roles	13
Amazon Simple Storage Service (Amazon S3)	13
Additional resources	14
Using your own dataset	15
Uninstall the solution	16
Using the AWS Management Console	16
Using AWS Command Line Interface	16
Deleting the Amazon S3 buckets	16
Source code	17
Contributors	18
Revisions	19
Notices	20

Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker

AWS Implementation Guide

Publication date: *November 2019 (last update (p. 19): December 2020)*

This implementation guide discusses architectural considerations and configuration steps for deploying Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker in the Amazon Web Services (AWS) Cloud. It includes links to an [AWS CloudFormation](#) template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for developers and data scientists who have practical experience with machine learning and architecting on the AWS Cloud.

Overview

In a very competitive marketplace, where brands are competing for customers' attention, maximizing your customers' engagement levels with your brand is vital to your success. One method to improve customers' engagement is to deliver personalized messages to your users according to their attributes, activity, and stages along the customer journey. Personalized messaging can strengthen the relationship between your brand and your customers, increasing customers loyalty and conversion rates. [Amazon Pinpoint](#) supports customers' engagement by sending personalized emails, text and voice messages, and push notifications. With Amazon Pinpoint, you can segment your customers based on demographics, behaviors, or other key business important performance indicators, and send personalized, timely, and relevant messages.

With Machine Learning (ML), you can increase customers' engagement by analyzing your customers' activity, identify patterns, and make recommendations to increase conversions. However, it can be a challenge to integrate ML models with messaging tools without in-house ML expertise.

To help you combine segmentation in Amazon Pinpoint and machine learning, AWS provides the Predictive Segmentation Using Amazon Pinpoint and [Amazon SageMaker](#) solution. Amazon SageMaker is an ML service that enables you to build, train, and deploy machine learning models.

Using this solution, you can build a reference architecture that automates the collection of customer data, predicts customer turnover, and maintains tailored audience segmentation for messaging.

This solution includes a sample dataset that you can use as a reference. To develop your own custom ML models with your own data, customize the solution. For more information, refer to [Using your own dataset \(p. 15\)](#).

Cost

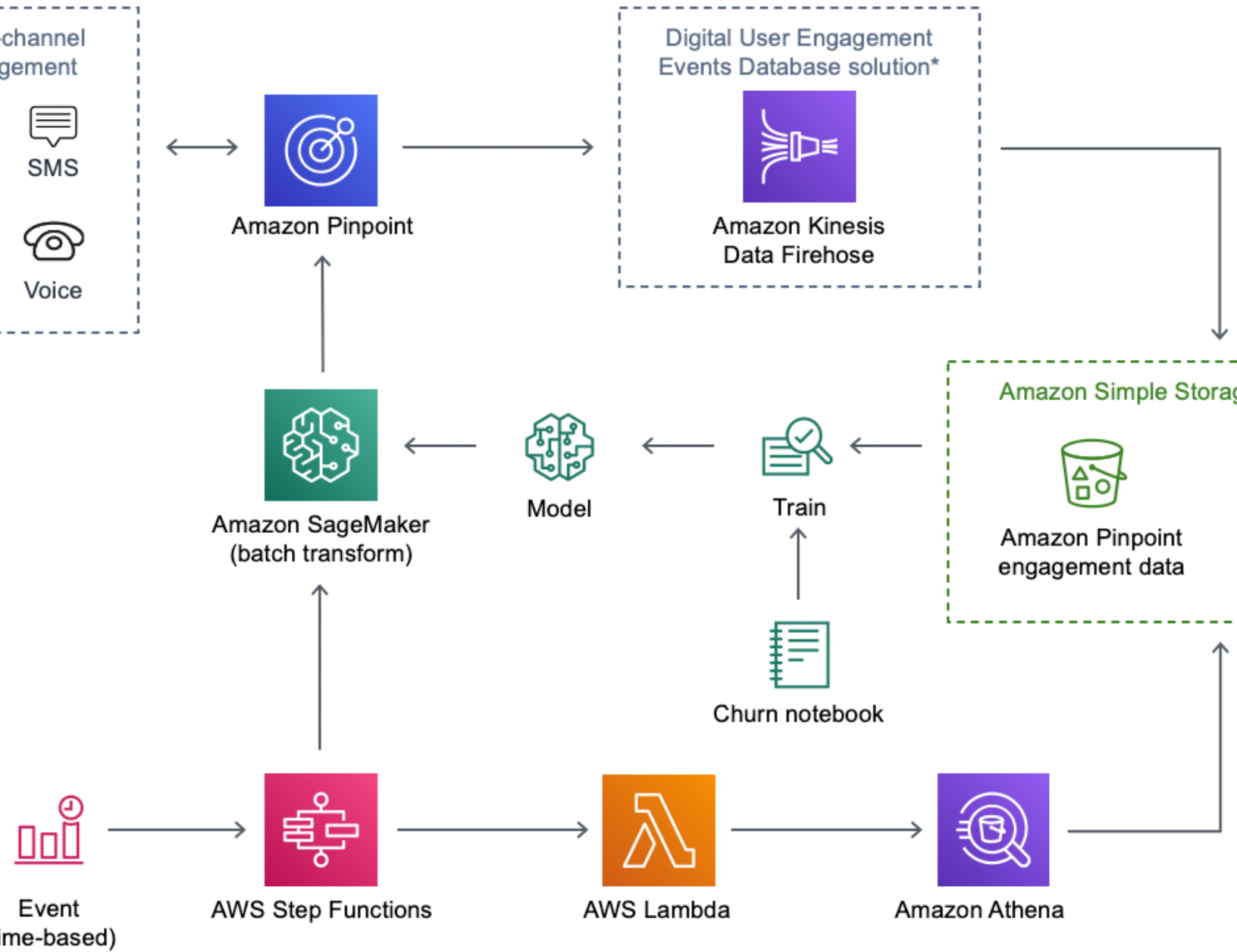
You are responsible for the cost of the AWS services used while running this reference deployment. As of the date of publication, the cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **\$130 per month**. This cost estimate assumes 100,000 customer endpoints stored in Amazon Pinpoint with 1 GB of other customer data in [Amazon Simple Storage Service \(Amazon S3\)](#). Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

The following table provides a cost estimate for deploying the solution.

AWS service	Monthly cost
Amazon Athena	\$0.15
Amazon CloudWatch	\$0.01
Amazon Kinesis Data Firehose	No cost
Amazon Pinpoint	\$120.00
SageMaker	\$8.50
AWS Step Functions	\$0.11
Amazon S3	\$0.05
Total	\$128.82

Architecture overview

Deploying this solution builds the following environment in the AWS Cloud.



*Deploying the Digital User Engagement Events Database solution is a prerequisite to deploying this solution

Figure 1: Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker architecture on AWS

The AWS CloudFormation template deploys a daily batch process orchestrated by [AWS Step Functions](#). The process runs according to the following:

- An [Amazon CloudWatch](#) time-based event triggers a series of [AWS Lambda](#) functions.
- An [Amazon Athena](#) query to query customer data stored in Amazon Simple Storage Service (Amazon S3).

The customer data includes endpoints exported from Amazon Pinpoint and end-user engagement data streamed from Amazon Pinpoint and end-user engagement data streamed from Amazon Pinpoint using the Digital User Engagement Events Database solution. Amazon SageMaker performs batch transform requests to predict customer churn based on a trained machine learning (ML) model.

By default, this solution is configured to process data from the example dataset. To modify the solution by using your own dataset, refer to [Using your own dataset \(p. 15\)](#).

Solution components

Amazon SageMaker

This solution uses an Amazon SageMaker notebook instance, which is a fully managed machine learning (ML) [Amazon Elastic Compute Cloud](#) (Amazon EC2) compute instance that runs the solution's [Jupyter](#) notebook. The notebook is used to train and deploy the solution's ML model. For more information on notebook instances, refer to [Use SageMaker Notebook Instances](#) in the *Amazon SageMaker Developer Guide*.

Important

By default, the solution uses an ml.m4.xlarge instance to train the model, and an ml.m5.large instance to run batch transform requests. After the model is trained, you can terminate the model-training instance to reduce costs.

Dataset

This solution contains a simple, example dataset that is used to train the solution's ML model. The dataset includes example customer data, engagement data, and endpoint export data. The dataset is designed to train and deploy a simplistic churn model to use to demonstrate the solution's functionality.

You can deploy and test this solution with little or no ML experience. However, to use this solution in your production environment, we recommend consulting with a data scientist to analyze and develop an ML model tailored specifically to your real customer data. For more information, refer to [Customization \(p. 6\)](#).

Design considerations

Customization

By default, Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker uses a simple, example dataset to train the machine learning (ML) model. You can customize the solution to use your own dataset. To train the model on your own dataset, you must modify the included notebook to point the model to your dataset. You must also create your own Amazon Athena query, and modify the solution's AWS Lambda function to point to that query. For more information, refer to [Using your own dataset \(p. 15\)](#).

Regional deployment

This solution uses Amazon Pinpoint, which is currently available in specific AWS Regions only. Therefore, you must launch this solution in a Region where Amazon Pinpoint is available. For the most current service availability by Region, refer to the [AWS Regional Services List](#).

AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker solution in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

[View
Template](#)

predictive-segmentation-using-amazon-pinpoint-and-amazon-sagemaker.template: Use this template to launch the solution and all associated components. The default configuration deploys AWS Step Functions, an Amazon CloudWatch time-based event, AWS Lambda functions, an Amazon Athena query, Amazon Simple Storage Service (Amazon S3) buckets, Amazon Pinpoint, [Amazon Kinesis Data Firehose](#), [AWS Identity and Access Management \(IAM\)](#) roles, and Amazon SageMaker. You can customize the template to meet your specific needs.

Automated deployment

Before you launch the automated deployment, review the considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

Prerequisites

Before you deploy this solution, you must deploy the Digital User Engagement Events Database solution that will create an Amazon Pinpoint project, configure the project's event stream, and a queryable database of engagement data in Amazon Athena.

Updating the stack from version 1.0.1 or earlier

If you have previously deployed version 1.0.1 or an earlier version of this solution, follow this procedure to fully delete the previous Amazon CloudFormation stack before following the steps in this guide.

1. Sign in to the [AWS CloudFormation console](#).
2. On the **Stacks** page, choose the stack created from the earlier version of this solution.
3. Choose **Delete** to open the confirmation dialog and choose **Delete stack**.

Note

The solution is configured to retain all Amazon S3 buckets, Amazon CloudWatch logs, and any previously imported Amazon Pinpoint segments in order to retain all data. The SageMaker model that was previously deployed will also be retained. Deleting the stack will remove all other configured service artifacts. In AWS CloudFormation, choose the **Resources** tab to view all service artifacts.

Deploy the Digital User Engagement Events Database solution

This solution relies on the Digital User Engagement Events Database solution to set up an Amazon Kinesis Data Firehose to route the Amazon Pinpoint events into Amazon S3 where they can be queried with Amazon Athena. The Digital User Engagement Events Database solution also configures an Amazon Pinpoint project, removing the requirement for you to create one separately. Outputs from the Digital User Engagement Events Database solution will be used as inputs in this solution.

1. [Download the template](#) to deploy the Digital User Engagement Events Database solution.

Note

For complete deployment instructions, refer to the separate Digital User Engagement Events Database solution [implementation guide](#).

2. Navigate to the [AWS CloudFormation console](#).
3. On the **Stacks** page, choose the stack created for the Digital User Engagement Events Database solution.
4. On the stack details page, choose the **Outputs** tab and, under the **Key** column, locate `DUES3DataLakeName` and `PinpointProjectId`. These keys identify the Amazon S3 bucket that was

created to store the Amazon Pinpoint engagement data and the Amazon Pinpoint Project ID. Copy the values of these outputs to use as inputs in this solution.

5. Choose the **Parameters** tab and, under the **Key** column, locate the `EventAthenaDatabaseName`. This key identifies the Amazon Athena database name used for querying the engagement events. Copy the value of the output to use as an input in this solution.

Deployment overview

Deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

[Step 1. Launch the stack \(p. 9\)](#)

- Launch the AWS CloudFormation template into your AWS account.
- Review the other template parameters, and adjust if necessary.

[Step 2. \(Optional\) Upload the sample Pinpoint segment \(p. 11\)](#)

- Upload the sample Pinpoint segment data for demonstration of the working solution.

[Step 3. Run the notebook \(p. 11\)](#)

- Run the Jupyter Notebook to train the ML model.

[Step 4. Run the state machine \(p. 12\)](#)

- Execute the AWS Step Functions state machine.

[Step 5. Verify the segment exists and is receiving records \(p. 12\)](#)

- Verify that the solution created the Amazon Pinpoint segment, and that the churn segment shows records.

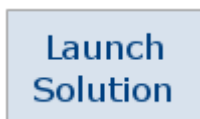
Step 1. Launch the stack

This automated AWS CloudFormation template deploys this solution in the AWS Cloud. Make sure that you have deployed the Digital User Engagement Events Database solution before launching the stack.

Note

You are responsible for the cost of the AWS services used while running this solution. Refer to the [Cost \(p. 2\)](#) section for more details. For full details, refer to the pricing webpage for each AWS service you will be using in this solution. Refer to [Additional resources \(p. 14\)](#) for links to the webpages for all services used in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `predictive-segmentation-using-amazon-pinpoint-and-amazon-sagemaker` AWS CloudFormation template.



You can also [download the template](#) as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.

Note

This solution uses Amazon Pinpoint, which is currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where Amazon Pinpoint is available. For the most current service availability by Region, refer to the [AWS Regional Services List](#).

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack, for example, `predictive-segmentation`.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Amazon Pinpoint Project ID	<i><Requires input></i>	Specifies the Amazon Pinpoint project ID that contains the endpoints to be exported and where the churn segment will be created.
Database Name	due_eventdb	Displays the name of the Amazon Athena database that was created from the Digital User Engagement Events Database solution.
Amazon S3	<i><Requires input></i>	Displays the Amazon S3 bucket created by the Digital User Engagement Events Database solution where the Amazon Pinpoint engagement events are stored.
Load Sample Events	<i><Requires selection></i>	Indicates whether the sample customer data should be loaded into the Amazon S3 bucket for demonstration purposes.
Model Training Instance Type	m1.m4.xlarge	Displays the SageMaker instance type used for model training.
Transform Instance Type	m1.m5.large	Displays the SageMaker instance type used for batch transform requests.
VPC Id	<i><Requires selection></i>	Specifies the identifier of Amazon Virtual Private Cloud where the SageMaker notebook instance is deployed.

Parameter	Default	Description
Subnet Id	<Requires selection>	Specifies the identifier of Amazon Virtual Private Cloud Subnet where the SageMaker notebook instance is deployed.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately 10 minutes.

Note

In addition to the primary AWS Lambda functions, this solution includes the `CustomResourceHelper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.

When running this solution, the `CustomResourceHelper` Lambda function is inactive. However, do not delete the function because it is necessary to manage associated resources.

Step 2. (Optional) Upload the sample Pinpoint segment

You can use a provided sample set of Amazon Pinpoint endpoint data to test the deployed solution end-to-end. The following steps outline how to import the sample data into Amazon Pinpoint for the solution to use. This is intended for demonstration purposes only.

1. Download the [SampleImport.csv](#) sample segment file.
2. Navigate to the [Amazon Pinpoint console](#).
3. In the **All projects** section, select the project to configure.
4. In the navigation pane, choose **Segments**.
5. Choose **Create a Segment**.
6. Choose **Import a segment** and upload the `SampleImport.csv` file and choose **Create segment**.

Step 3. Run the notebook

For the solution to work, you must train and deploy an SageMaker model. The following steps outline how to run the supplied SageMaker notebook using the sample training data to deploy a model.

1. Navigate to the [Amazon SageMaker console](#).
2. In the navigation pane, select **Notebook instances**.
3. Select **PredictiveChurnNotebookInstance**.

The notebook instance will already be running.

4. Select **Open Jupyter**.
5. In the Jupyter notebook interface, open the `xboost_customer_churn.ipynb` file.

6. In the **Cell** dropdown menu, select **Run All** to run the file.

Note

After you run the notebook, you can terminate the notebook instance to reduce costs.

Step 4. Run the state machine

The solution is configured to run an AWS Step Functions state machine daily to update Amazon Pinpoint endpoints with a churn score from the SageMaker model. The following steps outline how to run the AWS Step Functions state machine manually to ensure that it runs successfully.

1. Navigate to the [AWS Step Functions console](#).
2. In the navigation pane, select **State machines**.
3. Select the the solution's state machine. The name starts with `DailyChurnListStateMachine`.
4. Choose **Start execution**.

Wait for the state machine to finish executing. It can take up to 10 minutes to complete all of the steps.

Step 5. Verify the segment exists and is receiving records

To verify successful deployment, check that there is a segment in Amazon Pinpoint that contains all of the endpoints that are predicted to churn by the SageMaker model. The following steps outline how to verify in that the Amazon Pinpoint segment exists and contains endpoints.

1. Navigate to the [Amazon Pinpoint console](#).
2. In the **All Projects** list, select the applicable project.
3. In the navigation pane, choose **Segments**.
4. Select the **Predicted to Churn** segment.
5. Under **Number of Records**, verify there are entries.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the [AWS Security Center](#).

IAM roles

AWS Identity and Access Management (IAM) roles enable customers to assign granular access policies and permissions to services and users on AWS. Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker creates several IAM roles, including roles that grant the solution's AWS Lambda functions access to the other AWS services used in this solution.

Amazon Simple Storage Service (Amazon S3)

All Amazon S3 buckets follow [security best practices for Amazon S3](#) and are encrypted with SSE-S3 managed encryption, configured to block all public access, and enforce encryption of data in transit. Two of the Amazon S3 buckets might contain end-user data including Amazon Pinpoint endpoint exports, results of Amazon Athena queries, and Amazon SageMaker data. A third Amazon S3 bucket is used for [Amazon S3 server access logs](#) for the two buckets that might contain the end-user data.

Note

The Amazon S3 buckets are configured with the retention policy set to **Retain**.

Additional resources

- | | |
|---|---|
| <ul style="list-style-type: none">• Amazon Athena• Amazon CloudWatch• Amazon Kinesis Data Firehose• Amazon Pinpoint• Amazon SageMaker | <ul style="list-style-type: none">• Amazon Simple Storage Service• AWS CloudFormation• AWS Identity and Access Management• AWS Lambda• AWS Step Functions |
|---|---|

Using your own dataset

The Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker solution includes a simple, example dataset that contains example customer data, engagement data, and endpoint export data.

If you want to use this solution using your data, we recommend following AWS best practices for [uploading data in Amazon Simple Storage Service \(Amazon S3\)](#).

Use the following steps to modify the solution to use your dataset.

1. With the help of an experienced data scientist, train the ML model using your dataset and features from Amazon Pinpoint behavioral data.

For a list of Amazon Pinpoint events, refer to [Events](#) in the *Amazon Pinpoint REST API Reference*.

2. Create a new Amazon Athena query that pulls the applicable ML model features from your dataset. For more information, refer to [Getting Started](#) in the *Amazon Athena User Guide*.
3. Update the `QueryAugmentStart` AWS Lambda function's **NAMED_QUERY** environment variable with the identifier for the query you created in the previous step.

Uninstall the solution

To uninstall this solution, use the AWS Management Console or the AWS Command Line Interface (AWS CLI). Note that you must manually delete the Amazon Simple Storage Service (Amazon S3) buckets created by this solution.

Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#).
2. On the **Stacks** page, choose the stack created from the earlier version of this solution.
3. Choose **Delete** to open the confirmation dialog and choose **Delete stack**.

Note

The solution is configured to retain all Amazon S3 buckets, Amazon CloudWatch logs, and any previously imported Amazon Pinpoint segments in order to retain all data. The SageMaker model that was previously deployed will also be retained. Deleting the stack will remove all other configured service artifacts. In AWS CloudFormation, choose the **Resources** tab to view all service artifacts.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Deleting the Amazon S3 buckets

1. Sign in to the [Amazon S3 console](#).
2. Choose **Buckets** from the left navigation pane.
3. Locate the **<stack-name>** S3 buckets.
4. Select one of the S3 buckets and choose **Delete**.

Repeat the steps until you have deleted all of the **<stack-name>** S3 buckets.

Alternatively, you can configure the AWS CloudFormation template to delete the Amazon S3 buckets automatically. Before deleting the stack, change the deletion behavior in the AWS CloudFormation [DeletionPolicy](#) attribute.

Source code

Visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

Contributors

The following individuals contributed to this document:

- Ryan Lowe
- Chelsea Graf

Revisions

Date	Change	
November 2019	Initial release	
January 2020	Bug Fixes	
December 2020	Release v1.1.0: added dependency on the Digital User Engagement Events Database solution; updated SageMaker predictive score to be written as an endpoint metric to all endpoints; deployment of sample data is now optional; CloudFormation manually registers schema with AWS Glue; Amazon SageMaker notebook instance deploys to a VPC. For more information, refer to the CHANGELOG.md file in the GitHub repository.	

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Predictive Segmentation Using Amazon Pinpoint and Amazon SageMaker is licensed under the terms of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>