Md Ashfaq Ahmed

AI Package Tracker hands-on lab

Lab

Build your own prediction model using the Amazon SageMaker Canvas console with a step-by-step guide.

AWS Console

No AWS Account required!

This lab provides AWS Console access and resources without requiring an AWS account.

Open AWS Console

Overview | Lab guide

Introduction

Welcome to the AI Package Tracker Lab! This lab walks you through the foundations of Amazon SageMaker Canvas. Canvas is a no-code feature that you can use to build machine learning models that are customized to make predictions for your data. In this lab, we focus on building and training a model to improve prediction accuracy with a provided dataset. We also demonstrate how to analyze your model and make predictions.

The use case for this lab is to build a machine learning model in Canvas that can predict whether a delivery will be on time or delayed based on historical data. The goal is to build a model that predicts the delivery status of shipments with an accuracy of greater than 80%.

Time remaining: **07h 02min 01sec**

Exit lab

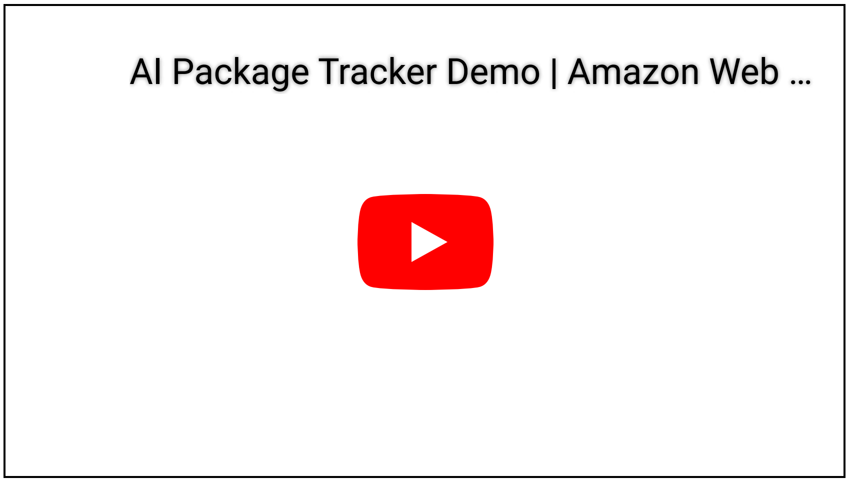
Finish lab

Lab resources

In this lab you'll be using the following dataset files:

On-time-delivery-data.csv

On-time-delivery-data_extended.csv



1. Access the AWS Console

This lab provides AWS Console access and resources without requiring an AWS account. To get started, in the right-hand **AWS Console** section, choose **Open AWS Console**.

(Optional) Create a Domain

To launch the Canvas application, you or your IT administrator needs to set up a domain and create a user profile. An Amazon SageMaker Domain supports SageMaker machine learning (ML) environments and consists of an associated Amazon Elastic File System (Amazon EFS) volume, a list of authorized users, and a variety of security, application, policy, and Amazon Virtual Private Cloud (Amazon VPC) configurations. Users within a Domain can share notebook files and other artifacts with each other.

In this lab, we have already provisioned the necessary resources for you to directly launch the Canvas application, so you can skip directly to the **Launch SageMaker Canvas application** section.

If you're interested in launching the Canvas application from your own AWS account, use the following instructions to create a domain:

1. In the Amazon SageMaker console, on the left-hand navigation panel, choose **Domains**.
2. Choose **Create domain**.
3. Choose **Standard setup**, and then choose **Configure**.
4. For the **Domain name**, enter a name for the Domain.
5. For **Authentication**, select **AWS Identity and Access Management (IAM)**.
6. For **Default execution role**, open the role selector dropdown and choose **Create a new role** to open the

Create an IAM role dialog box. In the dialog box, do the following:
    a. For **S3 buckets you specify**, select **Any S3 bucket**.
    b. Choose **Create role**.
7. In the Network and Storage Section, do the following:
    a. For **VPC**, select the default VPC.
    b. For **Subnet**, select one of the default subnets.
    c. For **Security group(s)**, select the default security group.
    d. Select **Public Internet Only**.
    e. For **Encryption key**, leave the default option, which is **No Custom Encryption**.
8. Choose **Next**.
9. On the **Studio settings** page, leave the default settings and choose **Next**.
10. On the **RStudio settings** page, leave the default settings and choose **Next**.
11. On the Canvas settings page, do the following:
    a. Turn on the toggles for Enable Canvas base permissions.
    b. For **Canvas storage configuration**, select **System managed**. This saves all of the Canvas application data, including copies of your datasets and model artifacts, to a SageMaker-created Amazon S3 bucket that follows the naming pattern `s3://sagemaker-{Region}-{your-account-id}`.
12. Choose **Submit**.

The Domain takes a few minutes to set up.

Now, we have to add a user to our Domain. We'll use this user to open our Canvas application.

To add a user, do the following:

1. On the **Domains** page, choose the Domain you created.
2. On the **Domain details** page, choose **Add user**.
3. For the **General settings** page, leave the default settings and choose **Next**.
4. For the **Studio settings** page, leave the default settings and choose **Next**.

5. For the **RStudio settings** page, leave the default settings and choose **Next**.
6. For the **Canvas settings** page, leave the default settings and choose **Submit**.

After the Domain and user are created, you can launch your Canvas application.

2. Launch SageMaker Canvas application

1. In search box on the navigation bar of the AWS Management Console, enter **SageMaker** and choose **Amazon SageMaker**.
2. In the Amazon SageMaker console, on the left-hand navigation panel, choose **Canvas**.
3. In the right hand side, you should see a dropdown with your user name. Choose **Open Canvas**. The application takes a few minutes to load.

3. Preview and import the dataset

The on-time-delivery-data dataset is a real delivery sample. It has 45,000 samples taken in the United States over three months in the winter of 2018. The main feature is the ZIP code. If you look at a single ZIP code, you can see deliveries are delayed. This means that on a specific day, at least 10 items on one truck were delayed getting to a customer. We want to build a model that can make real time predictions so that customers can be notified of a delay. Your goal is to make predictions with an accuracy greater than 80%. To get started, make sure to download the datasets and review the samples.

1. Once the Canvas application has loaded, in the left-hand navigation panel, choose **Datasets**.
2. Select the **Create** dropdown and choose **Tabular**.
3. Enter **on-time-delivery** as the dataset name and choose **Create**.
4. On the **Import** page, open the **Data Source** dropdown menu. Choose Amazon S3 and choose the S3 bucket with the name `canvas-lab-data-us-east-1-<account id>`. Then, choose `On-time-delivery-data.csv` file.
Tip: If you're importing the files locally, you might need permissions to do so. For more information, see Grant Your Users Permissions to Upload Local Files.

5. Choose **Create dataset**. This will show the preview of the first 100 rows of the sample to make sure the data matches the file.
6. Choose **Create dataset** again.
7. Repeat steps 2-6 steps to create the **on-time-delivery-extended** dataset.

When you review the imported data for on-time-delivery, you can see that Canvas has split the dataset into training and test datasets for you. There were originally 45,000 samples, but when you look at the number of rows, there are only 20,000. This is because Canvas randomly selects a percentage of the samples for training and the remaining percentage is for testing. Now that the data is ready, we can move on to creating a model.

4. Build the model

In this section, we create a model and use the quick build function to make a quick prediction.

1. In the **Datasets** page, select the on-time-delivery dataset you created by checking the box and select **Create a model**.
2. In the **Model name** field, enter `On time delivery model` and under **Problem type**, make sure Predictive Analysis is selected. Choose **Create**.
3. Select the **Target column** dropdown, and select `classification_ontime`. The target in the machine learning model is the prediction you want your model to make. Canvas automatically detects that this target is a 2-category prediction model type, also known as a binary classification task. On the left, you can review the distribution of the categories, on time and delayed. It is recommended to have a good balance of values. Note that we have 2 unique values for the target in this dataset: on time or delayed values.
4. Choose **Validate data** to make sure that there are no issues with the dataset.
5. Choose **Preview model**. The preview model expects an accuracy of approximately 66.889%, which is below our business requirement of 80% minimal accuracy.
6. Choose **Quick build** and choose **Start quick build**. With the quick build function, you can build a model and make a quick prediction in approximately 15 minutes. This checks if our preview is accurate.

After your quick build is finished, you can see that the accuracy has remained below the 80% threshold in the Analyze page. This page also shows you the impact of the ZIP code feature on the prediction. Since the model accuracy is below 80%, we need to think about the dataset and how we can improve it. There is only one feature in the dataset. Do you think that adding more features would help? If so, what are some features that you think might impact whether deliveries are delayed? In the next section, we cover which features we need to add to our dataset to improve the accuracy.

5. Analyze accuracy and create a new model version

In this section, we add some more features and see how this impacts our estimated accuracy. One feature you might have thought about adding is weather. Bad weather is a feature that might cause delayed packages. Since our current data set has no data related to the weather, we need to add it as a feature. In this new dataset we've provided, `on-time-delivery-data-extended`, we've added 13 additional features including temperature, barometric pressure, relative humidity, and the total number of items on the delivery truck to each sample in your existing data set. Now we need to use this updated dataset to create a new model version and see how it impacts the prediction accuracy.

1. In the **Analyze page**, choose **Add version**.
2. Select **on-time-delivery-extended** and choose **Select dataset**.
3. On the **Build** page, you can see that the target column and model type has been picked. Scroll down to review the features.
4. Choose **precipitation_rate**, and in the right side panel, review the distribution of values. Since all of the samples have the same value (0) for this column, it won't have any impact on the model. We'll remove it from the dataset to train our model more efficiently. Uncheck the **precipitation-rate** box.
5. Choose **Validate data** to make sure that there are no issues with the dataset.
6. Choose **Preview model**. The estimated accuracy is approximately 83%.

7. In the **Quick build** dropdown menu, select and choose **Standard build**. Choose Start standard build.

Standard builds takes around 2-4 hours and should give you the best predictions. You can now see that the accuracy is still approximately 83%. It is expected that the preview accuracy and final model accuracy are not always the same. In the next section, we make predictions.

6. Make predictions

In this section, we make a prediction with our updated dataset. We make a single prediction, which is when you only need to classify one data point.

1. Choose the **Predict** tab.
2. In the **Predict** page, choose **Single prediction**. This loads the model and you can change the values.
3. Update **total_items** value to 525, then temperature to −`5.5` and **max_temperature** to −2. Scroll down to see the rest of the features and update the **temperature_at_1500m** value to −7, **min_temperature** to −`6.5`, and **dew_point_temperature** to −8.
4. In the right side panel, choose **Update**. You can see that the prediction value has changed to `Delayed`, with a high percentage chance that the prediction is correct. When you scroll to the top of the list, you can see that the feature importance has changed. **Pressure** and **temperature** actually have the largest impact on this prediction.

Congratulations, you have successfully built a model that predicts the delivery status of shipments with an accuracy greater than 80%!

You can also try this lab in your own AWS account, but don't forget to delete your models and clean up your resources as a best practice. Choose **Finish lab** to discover more content and next steps.

Did you know that you can extend your Canvas capabilities by trying out the following features? To test these features, create your own AWS account:

- Ready-to-use models — Get predictions for your data with pre-built models for a variety of use cases ranging from language detection to personal information detection.

- Explore more use cases — Predictive analysis (classification, regression, forecasting use-cases on tabular data), image analysis (image classification), and text analysis (text classification).
- Import data through 40+ sources — Import your data from SaaS platforms such as Salesforce or Google Analytics.
- Collaboration — Share your models with SageMaker Studio users to get feedback and improve your models. Alternatively, a SageMaker Studio user can share their model with you in Canvas, and you can use their model to make predictions.
- QuickSight Integration — Send the results of your batch predictions to Amazon QuickSight and build predictive visualizations and dashboards.
- MLOps – Register your Canvas models to the SageMaker model registry, which data scientists can use to integrate your model with MLOps processes in your organization and put it into production.
- Automatic dataset updates and automatic batch predictions — If you regularly receive batches of new data that you want to import into Canvas, you can configure automatic updates for your datasets. You can also configure automatic batch prediction jobs to initiate whenever your dataset updates.

Did this page help you?

Yes No

Provide feedback

Do you have questions or an idea about a new lab?

Make a post on our forum and we will work with you to answer questions and bring your ideas to life.

