Train a model with Amazon SageMaker

**SPL-TF-300-MLMLMD-1 - Version 1.0.0**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab Overview**

AnyCompany Consulting has received data from the nonprofit advocacy group and has prepared the dataset for training. As the data scientist for AnyCompany Consulting, you now need to train the machine learning (ML) model and evaluate the results. Because the label feature that you are trying to predict on is a binary classification (less than $50,000 or not), you need to choose an ML algorithm that can work on binary classification problems. To this end, you will configure and train a model based on SageMaker’s built-in XGBoost, then you will evaluate the prediction efficiency of the model.

You also need to explore a more custom scenario. To this end, you will configure a custom model built on the XGBoost framework. By running XGBoost as a framework inside SageMaker, you have more flexibility and access to more advanced scenarios, such as k-fold cross-validation, because you can customize your own training scripts.

OBJECTIVES

By the end of this lab, you should be able to do the following:

* Train a model using built-in SageMaker Algorithms.
* Understand how to write custom training and inference code while still using common ML frameworks maintained by AWS.
* Understand how to Import custom libraries and dependencies to train your model.
* Understand how to setup a Hyperparameter Tuning Job in SageMaker.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Learn more:** Where to find more information.
* **Note:** A hint, tip, or important guidance.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

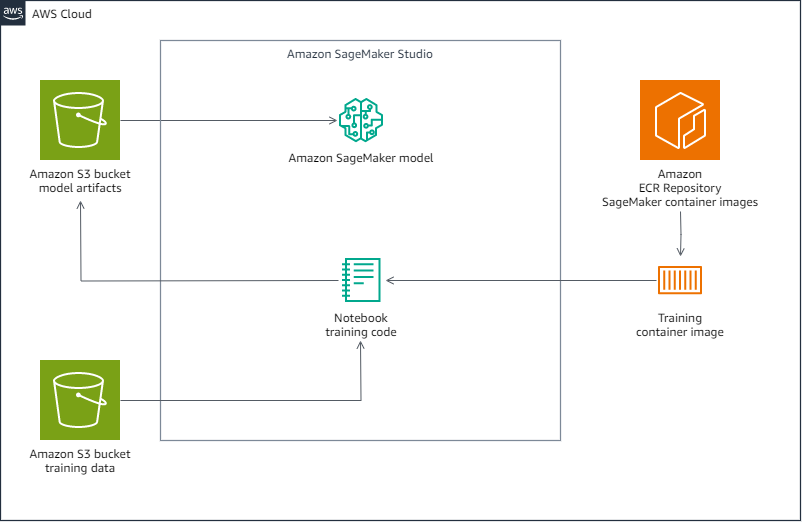
**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

LAB ENVIRONMENT

The following diagram shows the basic architecture of the lab environment:



*Image description: The preceding diagram depicts an Amazon Simple Storage Service (Amazon S3) bucket with the processed training data. A Jupyter notebook with training code uses a training container image from an Amazon ECR Repository of SageMaker container images. The notebook code creates a model and stores the model in another Amazon S3 bucket.*

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Task 1: Train a model using a built-in algorithm**

In this task, you first launch a SageMaker Studio application. Then you configure an estimator object and hyperparameters, run a SageMaker training job, and evaluate the model.

TASK 1.1: SET UP THE ENVIRONMENT

1. Copy the **SageMakerStudioUrl** value that is listed to the left of these instructions.
2. Open a new browser tab, and then paste the **SageMakerStudioUrl** into the address bar.
3. Press **Enter**.
4. Wait for SageMaker Studio to launch.

**Note:** If you see a pop-up window asking if you want to take a tour of Sagemaker Studio, choose to **Skip Tour for now**.

1. In the left navigation section, locate the **Applications** pane.
2. From the Applications pane, choose **Studio Classic**.

The browser displays the **SageMaker Studio Classic** page.

1. For the listed **Studio Classic** application, in the **Action** column, choose **Open**.

A new browser tab opens, taking you to the **Amazon SageMaker Studio Classic** interface.

**Note:** It might take 1–2 minutes for the SageMaker Studio Classic interface to load for the first time.

**Note:** If you see a *The loading screen is taking a long time. Would you like to clear the workspace or keep waiting?* message, choose Keep waiting.

The SageMaker Studio application opens with the Home tab displayed.

Next, clone a Git repository and open the lab repository folder.

1. In the left menu bar, choose the **Git** icon.
2. Choose Clone a Repository.

The SageMaker Studio environment displays the **Clone Git Repository** window.

1. For **Git repository URL (.git):**, copy and paste the **CloneUrlForRepo** value that is listed to the left of these instructions, and select the suggested URL from the dropdown menu.
2. Clear the **Project directory to clone into** field.

**Caution:** Make sure that all text is cleared from the **Project directory to clone into** field before proceeding.

1. Choose Clone.
2. Wait for the repository cloning to complete.

The SageMaker Studio environment opens the **MLMLMD\_Repository** folder when the clone completes.

**Task complete:** You have successfully launched SageMaker Studio and cloned a Git repository into the environment.

TASK 1.2: TRAIN THE MODEL

1. In the left navigation pane, open the **train\_built\_in.ipynb** notebook.

The SageMaker Studio environment displays the **Set up notebook environment** window.

1. In the **Set up notebook environment** window, select the following:

* **Image**: Data Science 3.0
* **Kernel**: Python 3

1. Choose Select.

**Note:** It might take 1–2 minutes for the kernel to load.

1. Carefully advance through the **train\_built\_in.ipynb** notebook. Run each code cell and review the completed output. To run a cell, select within the cell and press **Shift + Enter** or, at the top of the page, choose the **Run** button. An asterisk appears next to the code block while it runs.

**Task complete:** When you have finished running the notebook, you can return here to move to the next task.

**Task 2: Train a model using a custom script in script-mode**

In this task, you once again train a model, but this time using a customized script and the SageMaker script-mode.

1. In the left navigation pane of the SageMaker Studio environment, open the **train\_script\_mode.ipynb** notebook.

The SageMaker Studio environment displays the **Set up notebook environment** window.

1. In the **Set up notebook environment** window, select the following:

* **Image**: Data Science 3.0
* **Kernel**: Python 3

1. Choose Select.

**Note:** It might take 1–2 minutes for the kernel to load.

1. Carefully advance through the **train\_script\_mode.ipynb** notebook. Run each code cell and review the completed output. To run a cell, select within the cell and press **Shift + Enter** or, at the top of the page, choose the **Run** button. An asterisk appears next to the code block while it runs.

**Task complete:** When you have finished running the notebook, you can return here to finish the lab.

**Conclusion**

You have successfully done the following:

* Trained a model using built-in SageMaker Algorithms.
* Understood how to write custom training and inference code while still using common ML frameworks maintained by AWS.
* Understood how to Import custom libraries and dependencies to train your model.
* Understood how to setup a Hyperparameter Tuning Job in SageMaker.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Amazon SageMaker Documentation](https://docs.aws.amazon.com/sagemaker/)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).