## Lab 2-02: Evaluation and Optimization

### Lab Prerequisites

* Familiarity with basic AWS Cloud Computing concepts and terminology
* An AWS account with an active subscription

### Service Introduction

AWS SageMaker

Amazon SageMaker is a fully managed service that allows any developer or data scientist to quickly create, train, and deploy Machine Learning (ML) models. SageMaker makes it easy to create high-quality models by removing the heavy lifting from each phase of the machine learning process.

AWS Simple Storage Service (S3)

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve an infinite quantity of data from any place and at any time. You may quickly create projects that integrate cloud-native storage using this service. Because Amazon S3 is easily customizable and you only pay for what you use, you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. You can build a simple FTP system or a complex web application like the Amazon.com retail website, read the same piece of data a million times or only for emergency disaster recovery, and store whatever type and amount of data you desire.

AWS CloudWatch

Amazon CloudWatch is a tracking service for Amazon Web Services (AWS) cloud services and software. Amazon CloudWatch may be used to collect and monitor data, monitor log files, and trigger alarms. Amazon CloudWatch can monitor AWS resources, such as Amazon EC2 instances, Amazon DynamoDB tables, Amazon RDS DB instances, and custom metrics and log files created by your applications and services. You can watch your system's resource use, application performance, and operational health using Amazon CloudWatch. These insights might help you react and keep your app working smoothly.

### Case Study Enterprise Retail – LexisNexis Legal & Professional

Background

A leading global source of legal, regulatory, and business information and analytics, LexisNexis Legal & Professional supports customers' efforts to strengthen the rule of law globally, boost productivity, and improve decision-making and outcomes.

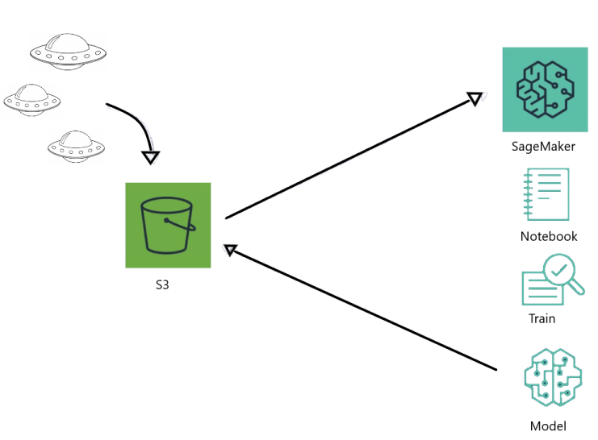
Challenge

You work as a Machine Learning developer in LexisNexis Legal & Professional. Your organization gives you a task to train the customer classification model. They also give you the task tune model to find the most optimized model for the problem. Determine if the model is less accurate or about the same. What objective metric would you want to monitor to ensure this? How do you plan on measuring success? Which hyperparameters need to be tuned? What combination do hyperparameters need to be used? How much faster was training time improved?

Proposed Solution

The solution is for you to use AWS services to automate all the tasks. You create an AWS SageMaker hyperparameter tuning job with different ranges of values for the hyperparameter to find the best configuration, which minimizes the validation: objective\_loss metric. The reason that you use this metric is that it is used in multi-classification problems. This metric measures the performance of the classification model, and what it does is repeatably calculate the difference between the values that the model predicts and the actual values of a label. Hence, every time it passes over data and makes predictions, it recalculates the objective loss and tries to minimize this value overall. Hence, that is the task here. AWS recommends that you minimize this value when using it as our objective metric.

Lab Diagram



*Figure 8-38: Lab Diagram*

Implementation Steps

1. Create an S3 Bucket.
2. Create SageMaker Hyperparameter Training Job.
3. Analyze CloudWatch Training Logs.
4. Create SageMaker Notebook.

### Solution

|  |
| --- |
| **Step 1: Create S3 Bucket**   1. Log in to the **AWS Console.** 2. Click on the **Services.**      1. Select **S3** from the **Storage.**      1. Click on the **Create bucket** button.      1. Give a bucket name **ips‐s3‐bucket.**      1. Scroll down. Click on the **Create bucket** button.      1. Download the **ufo\_sightings\_train\_recordIO\_protobuf.data** and **ufo\_sightings\_validatioin\_recordIO\_protobuf.data** files from the following Github link:   <https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20(1).zip>     1. Click on **ips‐s3‐bucket.**      1. Click on the **Upload** button.      1. Click on the **Add files** button.      1. Select the **ufo\_sightings\_train\_recordIO\_protobuf.data** and **ufo\_sightings\_validatioin\_recordIO\_protobuf.data** files. 2. Click on the **Open** button.      1. Scroll down. Click on the **Upload** button.     Hence, the files have been uploaded. |

|  |
| --- |
| **Step 2: Create SageMaker Hyperparameter Training Job**   1. Click on **Services.**      1. Select **SageMaker** from **Machine Learning**.      1. Click on the **Hyperparameter training jobs** from the left-hand side.      1. Click on the **Create hyperparameter tuning job** button.      1. Give the job name **IPS-linear-learner-tuning-job.**      1. Select **off** for early stopping.      1. Select the **Bayesian** strategy.      1. Click on the **Next** button.      1. Click on the **Add training job definition** button.      1. Give the training job definition name **IPS-linear-learner-tuning-job**.      1. Select the **AmazonSageMaker-ExecutionRole** IAM role.      1. Scroll down. Select the **Linear Learner** algorithm.      1. Select the **Pipe** input mode.      1. Select the **validation:objective\_loss** objective. Then Select the **minimize** type.      1. Type **22** in the **feature\_dim.**      1. Type **500** – **5000** in the **mini\_batch\_size**      1. Select the **multiclass\_classifier** in the **predictor\_type.**      1. Scroll down. Type .**0001 – 1.0** in the **wd**.      1. Type the .**0001 – 1.0** in the **L1.**      1. Type the **.0001** – **1.0** in the **learning\_rate.**      1. Scroll down. Type the **3** in the **num\_classes.**      1. Click on the **Next** button.      1. Select the **Pipe** input mode for the train channel.      1. Go to the **S3 dashboard**. 2. Click on **ips-s3-bucket.**      1. Click on the **ufo\_sightings\_train\_recordIO\_protobuf.data** file.      1. Copy the **key name.**      1. Go back to the SageMaker dashboard. 2. Give the S3 location path of the training file in the following manner **s3://ips-s3—bucket/<Train\_Key\_Name>.**      1. Click on the **Add channel** button.      1. Give a channel name: **Validation**. 2. Select the **Pipe** input mode for the validation channel.      1. Give the S3 location path of the validation file in the following manner **s3://ips-s3-bucket/<Validation\_Key\_Name>.**      1. Give the **S3 location path** of output data. **s3://ips-s3-bucket-output.**      1. Click on the **Next** button.      1. Select the **ml.m4.xlarge** instance type.      1. Set **5 minutes** in **Maximum duration per training job.**      1. Click on the **Next** button.      1. Type **5** in the **Maximum Parallel Training Jobs** box**.** 2. Type **50** in the **Maximum Training Jobs** box**.**      1. Click on the **Next** button.      1. Click on the **Create hyperparameter tuning job** button.     A successfully created job will take time up to 30 minutes to complete.     1. Click on the **IPS-linear-learner-tuning-job**      1. Check the **Training job status counter.**      1. Hence, the training is completed. 2. Click on **IPS-linear-learner-tuning-job.**      1. Click on the **Best training job** tab.      1. Scroll down you will see the best training job hyperparameters.      1. Copy the name **IPS-linear-learner-tuning-job.** |

|  |
| --- |
| **Step 3: Analyze CloudWatch Training Logs**   1. Click on the **Services.**      1. Select the **CloudWatch** from the **Management & Governance.**      1. Click on the **Logs groups** from the left-hand side.      1. Click on the **/aws/sagemaker/TrainingJobs.**      1. Paste the name **ips-linear-learner-tuning-job.** 2. Click on **ips-linear-learner-tuning-job.**      1. You will see the accuracy and other metrics. |

|  |  |
| --- | --- |
| **Step 4: Create SageMaker Notebook**   1. Click on **Services.**      1. Select **SageMaker** from **Machine Learning.**      1. Click on **Notebook Instance** on the left-hand side menu.      1. Click on the **Create notebook instance** button.      1. Give the name **ips-notebook-instance.**      1. Select the **ml.c4.medium** Notebook instance type.      1. Scroll down under permission and encryption. 2. Click on **Create a new role.**      1. Select **Any S3 Bucket.** 2. Click on the **Create role** button.     Hence, you have successfully created an IAM role.     1. Scroll down. Click on the **Create notebook instance** button. It will take a few minutes.      1. Click on **ips-notebook-instance.**      1. Click on the **Open Jupyter** button.      1. Download the jupyter notebook from the following Github link:   <https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20(1).zip>  Use Chapter8/ufo-evaluation-optimization-lab.ipynb     1. Click on the U**pload** button.      1. Select the **ufo-evaluation-optimization-lab.ipynb** jupyter notebook file.      1. Click on the U**pload** button.      1. Click on the **ufo-evaluation-optimization-lab.ipynb** jupyter notebook.      1. Run the cell to import python libraries. Click on the **Run** button or press **Shift + Enter** to execute the cell.      1. Insert the bucket name **ips-s3-bucket**. Then press **Shift + Enter** to execute the cell.      1. Run the cell to get the train data stored in S3. Press **Shift + Enter** to execute the cell.      1. Run the cell to get the validation data stored in S3.      1. Run the cell to call the Linear Learner algorithm.      1. Run the cell to create a job and use the optimized hyperparameters.      1. Insert the hyperparameters in the red box below.      1. Insert the below hyperparameters.  |  | | --- | | early\_stopping\_patience=3  early\_stopping\_tolerance=0.001  epochs=15  l1=0.0647741539306635  learning\_rate=0.09329042024421902  loss='auto'  mini\_batch\_size=744  normalize\_data='true'  normalize\_label='auto'  num\_models='auto'  optimizer='auto'  unbias\_data='auto'  unbias\_label='auto'  use\_bias='true'  wd=0.000212481391205101 |      1. Run the cell to start the training job. 2. The training has started. It will take a few minutes to complete.      1. Now, the training is completed, and you will see the training accuracy logs.      1. After completing the lab, delete all the AWS services used in this lab so you would not get charged. |