BadziochAndrew_Flight_Delay-Student

October 31, 2024

1 Problem: Predicting Airplane Delays

The goals of this notebook are: - Process and create a dataset from downloaded .zip files - Perform exploratory data analysis (EDA) - Establish a baseline model - Move from a simple model to an ensemble model - Perform hyperparameter optimization - Check feature importance

1.1 Introduction to business scenario

You work for a travel booking website that wants to improve the customer experience for flights that were delayed. The company wants to create a feature to let customers know if the flight will be delayed because of weather when they book a flight to or from the busiest airports for domestic travel in the US.

You are tasked with solving part of this problem by using machine learning (ML) to identify whether the flight will be delayed because of weather. You have been given access to the a dataset about the on-time performance of domestic flights that were operated by large air carriers. You can use this data to train an ML model to predict if the flight is going to be delayed for the busiest airports.

1.2 About this dataset

This dataset contains scheduled and actual departure and arrival times reported by certified US air carriers that account for at least 1 percent of domestic scheduled passenger revenues. The data was collected by the U.S. Office of Airline Information, Bureau of Transportation Statistics (BTS). The dataset contains date, time, origin, destination, airline, distance, and delay status of flights for flights between 2013 and 2018.

1.2.1 Features

For more information about features in the dataset, see On-time delay dataset features.

1.2.2 Dataset attributions

Website: https://www.transtats.bts.gov/

Dataset(s) used in this lab were compiled by the U.S. Office of Airline Information, Bureau of Transportation Statistics (BTS), Airline On-Time Performance Data, available at https://www.transtats.bts.gov/DatabaseInfo.asp?DB_ID=120&DB_URL=Mode_ID=1&Mode_Desc=Aviation&

2 Step 1: Problem formulation and data collection

Start this project by writing a few sentences that summarize the business problem and the business goal that you want to achieve in this scenario. You can write down your ideas in the following sections. Include a business metric that you would like your team to aspire toward. After you define that information, write the ML problem statement. Finally, add a comment or two about the type of ML this activity represents.

Project presentation: Include a summary of these details in your project presentation.

2.0.1 1. Determine if and why ML is an appropriate solution to deploy for this scenario.

Write your answer here - Yes, ML is appropriate for this as deploying a ML model could better inform guests about dalays while improving operational efficieny.

2.0.2 2. Formulate the business problem, success metrics, and desired ML output.

Write your answer here - Business Problem - The travel company aims to increase customer satisfaction by predicting delays due to current weather conditions. By providing this information, the company seeks to reduce customer turnover and improve the travel experience. - Success Matrix 1. Prediction: The percentage of correct predictions made by the model in regards to weather related flight delays, with an accuracy of at least 85%. 2. Customer Satisfaction: Increase satisfaction scores within a one year time frame from model deployment. 3. Reduction in Complaints: Decrease complaints within a one year time frame from model deployment. - Desired ML Output: - 0: Flight is not likely to be delayed - 1: Flight is likely to be delayed

2.0.3 3. Identify the type of ML problem that you're working with.

Write your answer here - This task would be considered a binary classification problem, with the goal of categorizing flights into one of two classes (0: not a possible delay) and (1: possible delay)

2.0.4 4. Analyze the appropriateness of the data that you're working with.

Write your answer here - Data Appropriateness Analysis: The dataset for predicting flight delays contains scheduled and actual departure and arrival times reported by certified US air carriers. It includes data on various factors such as date, time, origin, destination, airline, distance, and delay status for flights between 2013 and 2018. Here's an analysis of the appropriateness of this data: - Strengths 1. Comprehensive Coverage: The dataset covers multiple years (2013-2018) and includes flights from all major US airlines, providing a broad and diverse range of data points for training the ML model. 2. Relevant Features: The dataset contains essential features that are likely to impactflight delays, such as date, time, origin, destination, airline, and distance. Thesefeatures can help in identifying patterns and correlations with weather-relateddelays. 3. Official Source: The data is collected and compiled by the U.S. Office of Airline Information, Bureau of Transportation Statistics (BTS), ensuring its reliability and accuracy. 4. Large Volume: With data spanning multiple years, the dataset provides a substantial volume of records, which is beneficial for training robust ML models. - Potential Challenges 1. Missing Values: The dataset might contain missing or incomplete records, which could affect the model's performance. Proper data cleaning and

preprocessing will be necessary to address this issue. 2. Imbalance in Delay Status: If the proportion of delayed flights is significantly smaller than on-time flights, the dataset may be imbalanced. This could lead to amodel that is biased towards predicting on-time flights. Techniques like resampling or using specialized algorithms for imbalanced data might be required. 3. External Factors: The dataset does not include direct weather information, which is crucial for predicting weather-related delays. Integrating external weather data from reliable sources will be necessary to improve the model's accuracy. 4. Temporal Changes: Flight delay patterns and airline operations may have changedover the years. Ensuring the model accounts for temporal changes and trends is important for maintaining its relevance and accuracy. - Conclusion: Overall, the dataset is appropriate for the task of predicting weather-related flight delays, given its comprehensive coverage, relevant features, and large volume. However,addressing potential challenges such as missing values, data imbalance, and the integration of weather data will be crucial for building an accurate and reliable ML model.

2.0.5 Setup

Now that you have decided where you want to focus your attention, you will set up this lab so that you can start solving the problem.

Note: This notebook was created and tested on an `ml.m4.xlarge` notebook instance with 25 GB storage.

[1]:
```
import os
from pathlib2 import Path
from zipfile import ZipFile
import time

import pandas as pd
import numpy as np
import subprocess

import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
instance_type='ml.m4.xlarge'

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

```
Matplotlib is building the font cache; this may take a moment.
```

3 Step 2: Data preprocessing and visualization

In this data preprocessing phase, you explore and visualize your data to better understand it. First, import the necessary libraries and read the data into a pandas DataFrame. After you import the data, explore the dataset. Look for the shape of the dataset and explore your columns and the types of columns that you will work with (numerical, categorical). Consider performing basic statistics

on the features to get a sense of feature means and ranges. Examine your target column closely, and determine its distribution.

3.0.1 Specific questions to consider

Throughout this section of the lab, consider the following questions:

1. What can you deduce from the basic statistics that you ran on the features?
2. What can you deduce from the distributions of the target classes?
3. Is there anything else you can deduce by exploring the data?

Project presentation: Include a summary of your answers to these questions (and other similar questions) in your project presentation. Start by bringing in the dataset from a public Amazon Simple Storage Service (Amazon S3) bucket to this notebook environment.

[2]:
```
# download the files

zip_path = '/home/ec2-user/SageMaker/project/data/FlightDelays/'
base_path = '/home/ec2-user/SageMaker/project/data/FlightDelays/'
csv_base_path = '/home/ec2-user/SageMaker/project/data/csvFlightDelays/'

!mkdir -p {zip_path}
!mkdir -p {csv_base_path}
!aws s3 cp s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_project/
 ↪data/ {zip_path} --recursive
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_10.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2014_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_1.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_11.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2014_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_4.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_5.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_7.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
```

```
ime_Performance_1987_present_2014_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_6.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_3.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_9.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_12.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2014_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_1.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_2.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_12.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2015_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_11.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2015_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_2.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_4.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_3.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_5.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
```

```
ime_Performance_1987_present_2015_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2014_8.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2014_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_10.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2015_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_6.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_1.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_7.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_10.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2016_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_3.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_9.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_11.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2016_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2015_8.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2015_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_12.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2016_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_2.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
```

```
ime_Performance_1987_present_2016_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_4.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_7.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_6.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_9.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_5.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2016_8.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2016_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_1.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_12.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2017_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_2.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_11.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2017_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_4.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_10.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
```

```
Time_Performance_1987_present_2017_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_3.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_5.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_6.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_9.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_7.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_1.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2017_8.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2017_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_11.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2018_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_10.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2018_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_12.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_
Time_Performance_1987_present_2018_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_2.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_4.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
```

```
ime_Performance_1987_present_2018_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_6.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_3.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_7.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_5.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_8.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-
1/flight_delay_project/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_p
resent_2018_9.zip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_T
ime_Performance_1987_present_2018_9.zip
```

[3]:
```
zip_files = [str(file) for file in list(Path(base_path).iterdir()) if '.zip' in␣
 ↪str(file)]
len(zip_files)
```

[3]:
```
60
```

Extract comma-separated values (CSV) files from the .zip files.

[4]:
```
def zip2csv(zipFile_name , file_path):
    """
    Extract csv from zip files
    zipFile_name: name of the zip file
    file_path : name of the folder to store csv
    """

    try:
        with ZipFile(zipFile_name, 'r') as z:
            print(f'Extracting {zipFile_name} ')
            z.extractall(path=file_path)
    except:
        print(f'zip2csv failed for {zipFile_name}')
```

```
for file in zip_files:
    zip2csv(file, csv_base_path)

print("Files Extracted")
```

```
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_12.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_5.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_4.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_11.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_5.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_4.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_8.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_1.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_3.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_9.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_11.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_10.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_1.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_12.zip
Extracting /home/ec2-
```

```
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_4.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_3.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_2.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_12.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_6.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_10.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_8.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_10.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_7.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_11.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_7.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_2.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_3.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_10.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_8.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_6.zip
Extracting /home/ec2-
```

```
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_8.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_7.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_5.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_4.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_5.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_11.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_4.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_2.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_12.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_1.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_9.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_5.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_9.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_6.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_7.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_6.zip
Extracting /home/ec2-
```

```
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_2.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_2.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_9.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_3.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_7.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_11.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2015_8.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_12.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_6.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2017_10.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_1.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2018_1.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2014_9.zip
Extracting /home/ec2-
user/SageMaker/project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfo
rmance_1987_present_2016_3.zip
Files Extracted
```

[5]:
```
csv_files = [str(file) for file in list(Path(csv_base_path).iterdir()) if '.
  ↪csv' in str(file)]
len(csv_files)
```

[5]:
```
60
```

Before you load the CSV file, read the HTML file from the extracted folder. This HTML file includes the background and more information about the features that are included in the dataset.

[6]:
```
from IPython.display import IFrame

IFrame(src=os.path.relpath(f"{csv_base_path}readme.html"), width=1000,␣
↪height=600)
```

[6]:
```
<IPython.lib.display.IFrame at 0x7f771cc0f940>
```

Load sample CSV file Before you combine all the CSV files, examine the data from a single CSV file. By using pandas, read the `On_Time_Reporting_Carrier_On_Time_Performance_(1987_present)_2018_9.csv` file first. You can use the built-in `read_csv` function in Python (pandas.read_csv documentation).

[7]:
```
df_temp = pd.
↪read_csv(f"{csv_base_path}On_Time_Reporting_Carrier_On_Time_Performance_(1987_present)_2018
↪csv")
```

Question: Print the row and column length in the dataset, and print the column names.

Hint: To view the rows and columns of a DataFrame, use the `<DataFrame>.shape` function. To view the column names, use the `<DataFrame>.columns` function.

[8]:
```
# **ENTER YOUR CODE HERE**
df_shape = df_temp.shape
print(f'Rows and columns in one CSV file is {df_shape}')
```

```
Rows and columns in one CSV file is (585749, 110)
```

Question: Print the first 10 rows of the dataset.

Hint: To print `x` number of rows, use the built-in `head(x)` function in pandas.

[9]:
```
# Enter your code here
df_temp.head(10)
```

[9]:

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	\
0	2018	3	9	3	1	2018-09-03	9E	
1	2018	3	9	9	7	2018-09-09	9E	
2	2018	3	9	10	1	2018-09-10	9E	
3	2018	3	9	13	4	2018-09-13	9E	
4	2018	3	9	14	5	2018-09-14	9E	
5	2018	3	9	16	7	2018-09-16	9E	
6	2018	3	9	17	1	2018-09-17	9E	
7	2018	3	9	20	4	2018-09-20	9E	
8	2018	3	9	21	5	2018-09-21	9E	
9	2018	3	9	23	7	2018-09-23	9E	

```
   DOT_ID_Reporting_Airline IATA_CODE_Reporting_Airline Tail_Number  …  \
0                     20363                         9E      N908XJ  …
1                     20363                         9E      N315PQ  …
2                     20363                         9E      N582CA  …
3                     20363                         9E      N292PQ  …
4                     20363                         9E      N600LR  …
5                     20363                         9E      N316PQ  …
6                     20363                         9E      N916XJ  …
7                     20363                         9E      N371CA  …
8                     20363                         9E      N601LR  …
9                     20363                         9E      N906XJ  …

   Div4TailNum  Div5Airport  Div5AirportID  Div5AirportSeqID Div5WheelsOn  \
0          NaN          NaN            NaN               NaN          NaN
1          NaN          NaN            NaN               NaN          NaN
2          NaN          NaN            NaN               NaN          NaN
3          NaN          NaN            NaN               NaN          NaN
4          NaN          NaN            NaN               NaN          NaN
5          NaN          NaN            NaN               NaN          NaN
6          NaN          NaN            NaN               NaN          NaN
7          NaN          NaN            NaN               NaN          NaN
8          NaN          NaN            NaN               NaN          NaN
9          NaN          NaN            NaN               NaN          NaN

  Div5TotalGTime Div5LongestGTime  Div5WheelsOff Div5TailNum  Unnamed: 109
0            NaN              NaN            NaN         NaN           NaN
1            NaN              NaN            NaN         NaN           NaN
2            NaN              NaN            NaN         NaN           NaN
3            NaN              NaN            NaN         NaN           NaN
4            NaN              NaN            NaN         NaN           NaN
5            NaN              NaN            NaN         NaN           NaN
6            NaN              NaN            NaN         NaN           NaN
7            NaN              NaN            NaN         NaN           NaN
8            NaN              NaN            NaN         NaN           NaN
9            NaN              NaN            NaN         NaN           NaN

[10 rows x 110 columns]
```

Question: Print all the columns in the dataset. To view the column names, use `<DataFrame>.columns`.

[10]:
```
# **ENTER YOUR CODE HERE**
print(f'The column names are :')
print('##########')
for col in df_temp.columns:
    print(col)
```

```
The column names are :
#########
Year
Quarter
Month
DayofMonth
DayOfWeek
FlightDate
Reporting_Airline
DOT_ID_Reporting_Airline
IATA_CODE_Reporting_Airline
Tail_Number
Flight_Number_Reporting_Airline
OriginAirportID
OriginAirportSeqID
OriginCityMarketID
Origin
OriginCityName
OriginState
OriginStateFips
OriginStateName
OriginWac
DestAirportID
DestAirportSeqID
DestCityMarketID
Dest
DestCityName
DestState
DestStateFips
DestStateName
DestWac
CRSDepTime
DepTime
DepDelay
DepDelayMinutes
DepDel15
DepartureDelayGroups
DepTimeBlk
TaxiOut
WheelsOff
WheelsOn
TaxiIn
CRSArrTime
ArrTime
ArrDelay
ArrDelayMinutes
ArrDel15
ArrivalDelayGroups
```

ArrTimeBlk
Cancelled
CancellationCode
Diverted
CRSElapsedTime
ActualElapsedTime
AirTime
Flights
Distance
DistanceGroup
CarrierDelay
WeatherDelay
NASDelay
SecurityDelay
LateAircraftDelay
FirstDepTime
TotalAddGTime
LongestAddGTime
DivAirportLandings
DivReachedDest
DivActualElapsedTime
DivArrDelay
DivDistance
Div1Airport
Div1AirportID
Div1AirportSeqID
Div1WheelsOn
Div1TotalGTime
Div1LongestGTime
Div1WheelsOff
Div1TailNum
Div2Airport
Div2AirportID
Div2AirportSeqID
Div2WheelsOn
Div2TotalGTime
Div2LongestGTime
Div2WheelsOff
Div2TailNum
Div3Airport
Div3AirportID
Div3AirportSeqID
Div3WheelsOn
Div3TotalGTime
Div3LongestGTime
Div3WheelsOff
Div3TailNum
Div4Airport

```
Div4AirportID
Div4AirportSeqID
Div4WheelsOn
Div4TotalGTime
Div4LongestGTime
Div4WheelsOff
Div4TailNum
Div5Airport
Div5AirportID
Div5AirportSeqID
Div5WheelsOn
Div5TotalGTime
Div5LongestGTime
Div5WheelsOff
Div5TailNum
Unnamed: 109
```

Question: Print all the columns in the dataset that contain the word *Del*. This will help you see how many columns have *delay data* in them.

Hint: To include values that pass certain `if` statement criteria, you can use a Python list comprehension.

For example: `[x for x in [1,2,3,4,5] if x > 2]`

Hint: To check if the value is in a list, you can use the `in` keyword (Python in Keyword documentation).

For example: `5 in [1,2,3,4,5]`

[11]:
```python
# Enter your code here
delay = [col for col in df_temp.columns if 'Del' in col]

print(f'The columns containing "Del" are:')
print('##############')
for col in delay:
    print(col)
```

```
The columns containing "Del" are:
##############
DepDelay
DepDelayMinutes
DepDel15
DepartureDelayGroups
ArrDelay
ArrDelayMinutes
ArrDel15
ArrivalDelayGroups
CarrierDelay
WeatherDelay
NASDelay
```

SecurityDelay
LateAircraftDelay
DivArrDelay

Here are some more questions to help you learn more about your dataset.

Questions

1. How many rows and columns does the dataset have?
2. How many years are included in the dataset?
3. What is the date range for the dataset?
4. Which airlines are included in the dataset?
5. Which origin and destination airports are covered?

Hints - To show the dimensions of the DataFrame, use `df_temp.shape`. - To refer to a specific column, use `df_temp.columnName` (for example, `df_temp.CarrierDelay`). - To get unique values for a column, use `df_temp.column.unique()` (for, example `df_temp.Year.unique()`).

[12]:
```
print("The #rows and #columns are ", df_temp.shape[0], " and ", df_temp.
 ↪shape[1])
print("The years in this dataset are: ", df_temp.Year.unique())
print("The months covered in this dataset are: ", df_temp.Month.unique())
print("The date range for data is :" , min(df_temp.FlightDate.unique()), " to␣
 ↪", max(df_temp.FlightDate.unique()))
print("The airlines covered in this dataset are: ", list(df_temp.
 ↪Reporting_Airline.unique()))
print("The Origin airports covered are: ", list(df_temp.OriginAirportID.
 ↪unique()))
print("The Destination airports covered are: ", list(df_temp.DestAirportID.
 ↪unique()))
```

```
The #rows and #columns are  585749  and  110
The years in this dataset are:  [2018]
The months covered in this dataset are:  [9]
The date range for data is : 2018-09-01  to  2018-09-30
The airlines covered in this dataset are:  ['9E', 'B6', 'WN', 'YV', 'YX', 'EV',
'AA', 'AS', 'DL', 'HA', 'UA', 'F9', 'G4', 'MQ', 'NK', 'OH', 'OO']
The Origin airports covered are:  [11298, 12953, 13485, 13487, 10397, 10529,
15607, 12478, 14492, 10994, 11433, 11977, 14307, 14814, 11721, 14122, 14524,
14633, 14635, 11193, 12992, 13930, 12451, 15323, 10721, 11203, 11278, 10990,
10434, 12339, 11986, 10781, 13244, 15370, 11042, 15016, 10785, 13871, 13277,
15380, 14685, 11996, 11618, 13795, 10693, 13198, 15249, 14576, 12945, 14321,
10792, 10208, 11057, 11995, 10821, 14683, 14100, 15412, 10154, 11423, 11953,
10431, 10581, 13296, 12323, 13433, 12266, 14696, 15096, 13931, 13342, 15919,
13495, 14027, 10135, 12197, 11612, 10257, 13029, 10423, 14098, 10980, 12007,
10685, 10731, 11003, 10874, 10408, 10146, 10868, 14986, 13367, 10599, 12264,
```

11150, 11066, 13204, 13184, 11697, 14730, 15304, 13541, 12889, 12954, 14771,
14679, 12892, 14570, 14057, 10299, 10140, 14869, 11292, 14107, 13796, 14893,
14843, 14747, 12191, 15027, 10800, 15070, 14831, 11252, 10732, 14254, 13933,
12250, 15024, 13891, 12206, 12278, 12391, 12896, 13158, 13232, 13851, 14193,
14908, 15376, 10279, 10713, 11140, 11259, 11481, 11540, 11884, 12951, 13256,
13230, 12448, 11109, 13422, 15624, 13502, 11413, 11973, 10561, 13476, 14689,
14262, 11775, 10747, 14457, 11111, 15023, 14108, 11638, 14698, 11778, 12217,
10627, 11267, 10849, 13303, 11624, 13577, 12124, 11921, 11637, 14783, 12177,
11049, 13061, 10185, 11577, 13377, 12915, 14574, 12891, 13290, 11982, 14674,
11980, 12511, 14588, 11823, 11146, 12884, 13830, 12173, 12758, 11503, 12982,
13360, 12441, 11630, 14489, 10170, 10551, 10754, 14709, 12819, 15991, 10926,
12523, 14828, 14256, 15841, 13873, 13970, 10165, 11648, 11641, 14252, 10620,
13486, 12402, 14222, 13264, 11603, 12016, 14955, 11337, 15356, 10643, 14761,
14112, 14082, 10466, 14905, 14512, 14716, 13983, 12223, 10676, 12265, 12544,
11898, 10666, 11537, 14025, 12917, 12003, 13829, 12280, 14314, 15295, 14259,
11027, 12119, 14952, 15008, 10155, 15411, 10136, 10469, 11067, 10728, 11867,
13459, 10268, 15401, 14960, 15074, 11274, 15048, 14842, 11905, 13076, 12898,
10158, 13139, 14092, 12244, 11617, 11308, 14113, 12335, 15897, 11122, 14711,
12156, 15041, 12389, 11076, 11471, 13121, 14802, 12899, 12255, 14877, 11468,
15582, 11092, 11865, 14543, 14288, 12902, 13344, 14006, 10967, 15454, 10558,
11447, 12519, 12888, 14794, 14237, 10372, 14487, 10157, 13964, 11097, 13127,
10141, 10333, 11587, 14150, 10631, 10739, 10779, 10918, 11013, 11525, 15389,
12129, 10577, 14520, 12397, 12343, 11695, 16218, 13241, 14109, 12094]
The Destination airports covered are: [11193, 14321, 14492, 13487, 13485,
14814, 11057, 14122, 14524, 12266, 10397, 12478, 11278, 11433, 12953, 15412,
14307, 11721, 12992, 10792, 13930, 15323, 12339, 10581, 10434, 10821, 12945,
10529, 11986, 11203, 15370, 13244, 10208, 11618, 13277, 14100, 15096, 13871,
15016, 15380, 13931, 11042, 10146, 10721, 13795, 15249, 10781, 14683, 12451,
10693, 10990, 15607, 14576, 11298, 11953, 10154, 14027, 10994, 11977, 13433,
13342, 11423, 12323, 11995, 13198, 14696, 10785, 13541, 15919, 14633, 11612,
12197, 14635, 13230, 14574, 11996, 13204, 11150, 14685, 14098, 10257, 10980,
10135, 10685, 13495, 12264, 12007, 11003, 10874, 10408, 10423, 10731, 13367,
10868, 11066, 10431, 13184, 11697, 14730, 15304, 13029, 14986, 13296, 10599,
12889, 14771, 14679, 14570, 12954, 10299, 14057, 14843, 10140, 14869, 11292,
12892, 14107, 13796, 14893, 14747, 15027, 10800, 11252, 14831, 15070, 12191,
10732, 14254, 13933, 12250, 15024, 13891, 11259, 11481, 11540, 12206, 13158,
13232, 13851, 14193, 14908, 10279, 10713, 11884, 12278, 12896, 15376, 12391,
11140, 13256, 12951, 15624, 12448, 11109, 13422, 11413, 11973, 10561, 11111,
14698, 13502, 14689, 14262, 11775, 11778, 10747, 14108, 15023, 11638, 14457,
13476, 12217, 10627, 11267, 10849, 13303, 11624, 13577, 12124, 11921, 11637,
13377, 13061, 11049, 12915, 11823, 11982, 14783, 12177, 12891, 13290, 14674,
12511, 14588, 11980, 10185, 11146, 12884, 11577, 12173, 12758, 13830, 11503,
12982, 12441, 13360, 14489, 10551, 10170, 10754, 14709, 11630, 12523, 10926,
15991, 14828, 12819, 15841, 14256, 13873, 13970, 10165, 11648, 10620, 14252,
11641, 13486, 12402, 14222, 13264, 11337, 11603, 12016, 14955, 15356, 10643,
10466, 14761, 12917, 10666, 14716, 14112, 14512, 14314, 14025, 10676, 14082,
14952, 12544, 15295, 12280, 11537, 12223, 12119, 14905, 13829, 11898, 15008,
12003, 12265, 11027, 13983, 14259, 10136, 15411, 10268, 15048, 10469, 10155,

```
11067, 10728, 15401, 15074, 14960, 11274, 14842, 11905, 13076, 13459, 11867,
12898, 10158, 13139, 14092, 12244, 11617, 11308, 14113, 15897, 14711, 12335,
12156, 10372, 15041, 12389, 11468, 14794, 15582, 14802, 13121, 13344, 14877,
12255, 10558, 14288, 12902, 11076, 11471, 14006, 15454, 14543, 10967, 11092,
12519, 11447, 12888, 11865, 12899, 14237, 14487, 10157, 13964, 11097, 13127,
10141, 10333, 14150, 10631, 11122, 10739, 10779, 10918, 11013, 11587, 11525,
12397, 12129, 10577, 15389, 14520, 12343, 11695, 16218, 13241, 14109, 12094]
```

Question: What is the count of all the origin and destination airports?

Hint: To find the values for each airport by using the **Origin** and **Dest** columns, you can use the `values_count` function in pandas (pandas.Series.value_counts documentation).

[13]:
```
counts = pd.DataFrame({'Origin':df_temp['Origin'].value_counts(), 'Destination':
 ↪df_temp['Dest'].value_counts()})
counts
```

[13]:

	Origin	Destination
ABE	303	303
ABI	169	169
ABQ	2077	2076
ABR	60	60
ABY	79	79
..	…	…
WRG	60	60
WYS	52	52
XNA	1004	1004
YAK	60	60
YUM	96	96

[346 rows x 2 columns]

Question: Print the top 15 origin and destination airports based on number of flights in the dataset.

Hint: You can use the `sort_values` function in pandas (pandas.DataFrame.sort_values documentation).

[14]:
```
# Enter your code here
counts.sort_values(by='Origin', ascending=False).head(15)
```

[14]:

	Origin	Destination
ATL	31525	31521
ORD	28257	28250
DFW	22802	22795
DEN	19807	19807
CLT	19655	19654
LAX	17875	17873
SFO	14332	14348
IAH	14210	14203

LGA	13850	13850
MSP	13349	13347
LAS	13318	13322
PHX	13126	13128
DTW	12725	12724
BOS	12223	12227
SEA	11872	11877

Given all the information about a flight trip, can you predict if it would be delayed?

The **ArrDel15** column is an indicator variable that takes the value *1* when the delay is more than 15 minutes. Otherwise, it takes a value of *0*.

You could use this as a target column for the classification problem.

Now, assume that you are traveling from San Francisco to Los Angeles on a work trip. You want to better manage your reservations in Los Angeles. Thus, want to have an idea of whether your flight will be delayed, given a set of features. How many features from this dataset would you need to know before your flight?

Columns such as `DepDelay`, `ArrDelay`, `CarrierDelay`, `WeatherDelay`, `NASDelay`, `SecurityDelay`, `LateAircraftDelay`, and `DivArrDelay` contain information about a delay. But this delay could have occured at the origin or the destination. If there were a sudden weather delay 10 minutes before landing, this data wouldn't be helpful to managing your Los Angeles reservations.

So to simplify the problem statement, consider the following columns to predict an arrival delay:

`Year`, `Quarter`, `Month`, `DayofMonth`, `DayOfWeek`, `FlightDate`, `Reporting_Airline`, `Origin`, `OriginState`, `Dest`, `DestState`, `CRSDepTime`, `DepDelayMinutes`, `DepartureDelayGroups`, `Cancelled`, `Diverted`, `Distance`, `DistanceGroup`, `ArrDelay`, `ArrDelayMinutes`, `ArrDel15`, `AirTime`

You will also filter the source and destination airports to be: - Top airports: ATL, ORD, DFW, DEN, CLT, LAX, IAH, PHX, SFO - Top five airlines: UA, OO, WN, AA, DL

This information should help reduce the size of data across the CSV files that will be combined.

Combine all CSV files First, create an empy DataFrame that you will use to copy your individual DataFrames from each file. Then, for each file in the `csv_files` list:

1. Read the CSV file into a dataframe
2. Filter the columns based on the `filter_cols` variable

   ```
   columns = ['col1', 'col2']
   df_filter = df[columns]
   ```

3. Keep only the `subset_vals` in each of the `subset_cols`. To check if the `val` is in the DataFrame column, use the `isin` function in pandas (pandas.DataFram.isin documentation). Then, choose the rows that include it.

   ```
   df_eg[df_eg['col1'].isin('5')]
   ```

4. Concatenate the DataFrame with the empty DataFrame

[15]:
```
def combine_csv(csv_files, filter_cols, subset_cols, subset_vals, file_name):

    """
    Combine csv files into one Data Frame
    csv_files: list of csv file paths
    filter_cols: list of columns to filter
    subset_cols: list of columns to subset rows
    subset_vals: list of list of values to subset rows
    """

    df = pd.DataFrame()

    for file in csv_files:
        df_temp = pd.read_csv(file)
        df_temp = df_temp[filter_cols]
        for col, val in zip(subset_cols,subset_vals):
            df_temp = df_temp[df_temp[col].isin(val)]

        df = pd.concat([df, df_temp], axis=0)

    df.to_csv(file_name, index=False)
    print(f'Combined csv stored at {file_name}')
```

[16]:
```
#cols is the list of columns to predict Arrival Delay
cols = ['Year','Quarter','Month','DayofMonth','DayOfWeek','FlightDate',
        'Reporting_Airline','Origin','OriginState','Dest','DestState',
        'CRSDepTime','Cancelled','Diverted','Distance','DistanceGroup',
        'ArrDelay','ArrDelayMinutes','ArrDel15','AirTime']

subset_cols = ['Origin', 'Dest', 'Reporting_Airline']

# subset_vals is a list collection of the top origin and destination airports␣
 ↪and top 5 airlines
subset_vals = [['ATL', 'ORD', 'DFW', 'DEN', 'CLT', 'LAX', 'IAH', 'PHX', 'SFO'],
               ['ATL', 'ORD', 'DFW', 'DEN', 'CLT', 'LAX', 'IAH', 'PHX', 'SFO'],
               ['UA', 'OO', 'WN', 'AA', 'DL']]
```

Use the previous function to merge all the different files into a single file that you can read easily.

Note: This process will take 5-7 minutes to complete.

[17]:
```
start = time.time()
combined_csv_filename = f"{base_path}combined_files.csv"
combine_csv(csv_files, cols, subset_cols, subset_vals, combined_csv_filename)
print(f'CSVs merged in {round((time.time() - start)/60,2)} minutes')
```

```
Combined csv stored at
/home/ec2-user/SageMaker/project/data/FlightDelays/combined_files.csv
CSVs merged in 4.69 minutes
```

Load the dataset Load the combined dataset.

[18]:
```
data = pd.read_csv(combined_csv_filename)
```

Print the first five records.

[19]:
```
# Enter your code here
data.head()
```

[19]:

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline
0	2015	4	11	2	1	2015-11-02	AA
1	2015	4	11	3	2	2015-11-03	AA
2	2015	4	11	4	3	2015-11-04	AA
3	2015	4	11	1	7	2015-11-01	AA
4	2015	4	11	2	1	2015-11-02	AA

	Origin	OriginState	Dest	DestState	CRSDepTime	Cancelled	Diverted
0	CLT	NC	DFW	TX	600	0.0	0.0
1	CLT	NC	DFW	TX	600	1.0	0.0
2	CLT	NC	DFW	TX	600	0.0	0.0
3	DFW	TX	ORD	IL	515	0.0	0.0
4	DFW	TX	ORD	IL	515	0.0	0.0

	Distance	DistanceGroup	ArrDelay	ArrDelayMinutes	ArrDel15	AirTime
0	936.0	4	19.0	19.0	1.0	162.0
1	936.0	4	NaN	NaN	NaN	NaN
2	936.0	4	7.0	7.0	0.0	152.0
3	802.0	4	-18.0	0.0	0.0	96.0
4	802.0	4	19.0	19.0	1.0	105.0

Here are some more questions to help you learn more about your dataset.

Questions

1. How many rows and columns does the dataset have?
2. How many years are included in the dataset?
3. What is the date range for the dataset?
4. Which airlines are included in the dataset?
5. Which origin and destination airports are covered?

[20]:
```
print("The #rows and #columns are ", data.shape[0], " and ", data.shape[1])
print("The years in this dataset are: ", list(data.Year.unique()))
print("The months covered in this dataset are: ", sorted(list(data.Month.
 ↪unique())))
print("The date range for data is :" , min(data.FlightDate.unique()), " to ",␣
 ↪max(data.FlightDate.unique()))
```

```
print("The airlines covered in this dataset are: ", list(data.Reporting_Airline.
  ↪unique()))
print("The Origin airports covered are: ", list(data.Origin.unique()))
print("The Destination airports covered are: ", list(data.Dest.unique()))
```

```
The #rows and #columns are  1658130  and  20
The years in this dataset are:  [2015, 2018, 2017, 2014, 2016]
The months covered in this dataset are:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
The date range for data is : 2014-01-01  to  2018-12-31
The airlines covered in this dataset are:  ['AA', 'DL', 'UA', 'WN', 'OO']
The Origin airports covered are:  ['CLT', 'DFW', 'ORD', 'LAX', 'SFO', 'PHX',
'IAH', 'DEN', 'ATL']
The Destination airports covered are:  ['DFW', 'ORD', 'ATL', 'PHX', 'SFO',
'LAX', 'IAH', 'DEN', 'CLT']
```

Define your target column: **is_delay** (*1* means that the arrival time delayed more than 15 minutes, and *0* means all other cases). To rename the column from **ArrDel15** to *is_delay*, use the `rename` method .

Hint: You can use the `rename` function in pandas (pandas.DataFrame.rename documentation).

For example:

```
data.rename(columns={'col1':'column1'}, inplace=True)
```

[21]:
```
# Enter your code here
data.rename(columns={'ArrDel15': 'is_delay'}, inplace=True)
```

Look for nulls across columns. You can use the `isnull()` function (pandas.isnull documentation).

Hint: `isnull()` detects whether the particular value is null or not. It returns a boolean (*True* or *False*) in its place. To sum the number of columns, use the `sum(axis=0)` function (for example, `df.isnull().sum(axis=0)`).

[22]:
```
# Enter your code here
data.isnull().sum(axis=0)
```

[22]:
```
Year                  0
Quarter               0
Month                 0
DayofMonth            0
DayOfWeek             0
FlightDate            0
Reporting_Airline     0
Origin                0
OriginState           0
Dest                  0
DestState             0
CRSDepTime            0
Cancelled             0
Diverted              0
```

```
Distance                0
DistanceGroup           0
ArrDelay            22540
ArrDelayMinutes     22540
is_delay            22540
AirTime             22540
dtype: int64
```

The arrival delay details and airtime are missing for 22,540 out of 1,658,130 rows, which is 1.3 percent. You can either remove or impute these rows. The documentation doesn't mention any information about missing rows.

[23]:
```
### Remove null columns
data = data[~data.is_delay.isnull()]
data.isnull().sum(axis = 0)
```

[23]:
```
Year                0
Quarter             0
Month               0
DayofMonth          0
DayOfWeek           0
FlightDate          0
Reporting_Airline   0
Origin              0
OriginState         0
Dest                0
DestState           0
CRSDepTime          0
Cancelled           0
Diverted            0
Distance            0
DistanceGroup       0
ArrDelay            0
ArrDelayMinutes     0
is_delay            0
AirTime             0
dtype: int64
```

Get the hour of the day in 24-hour-time format from CRSDepTime.

[24]:
```
data['DepHourofDay'] = (data['CRSDepTime']//100)
```

3.1 The ML problem statement

- Given a set of features, can you predict if a flight is going to be delayed more than 15 minutes?
- Because the target variable takes only a value of *0* or *1*, you could use a classification algorithm.

Before you start modeling, it's a good practice to look at feature distribution, correlations, and others. - This will give you an idea of any non-linearity or patterns in the data - Linear mod-

els: Add power, exponential, or interaction features - Try a non-linear model - Data imbalance - Choose metrics that won't give biased model performance (accuracy versus the area under the curve, or AUC) - Use weighted or custom loss functions - Missing data - Do imputation based on simple statistics – mean, median, mode (numerical variables), frequent class (categorical variables) - Clustering-based imputation (k-nearest neighbors, or KNNs, to predict column value) - Drop column

3.1.1 Data exploration

Check the classes *delay* versus *no delay*.

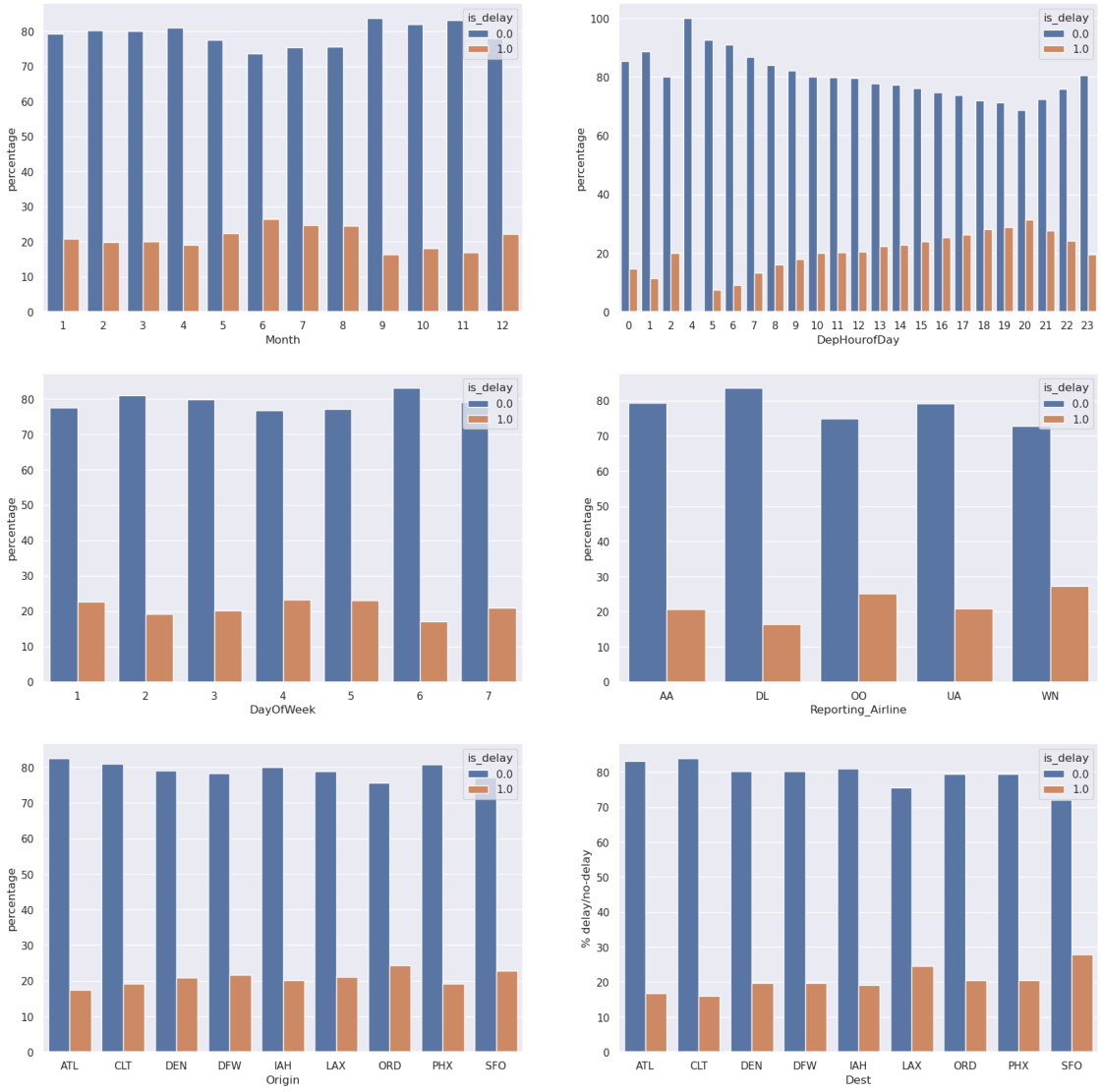
[25]:
```
# Enter your code here
(data.groupby('is_delay').size()/len(data) ).plot(kind='bar')
plt.ylabel('Frequency')
plt.title('Distribution of classes')
plt.show()
```

Question: What can you deduce from the bar plot about the ratio of *delay* versus *no delay*?

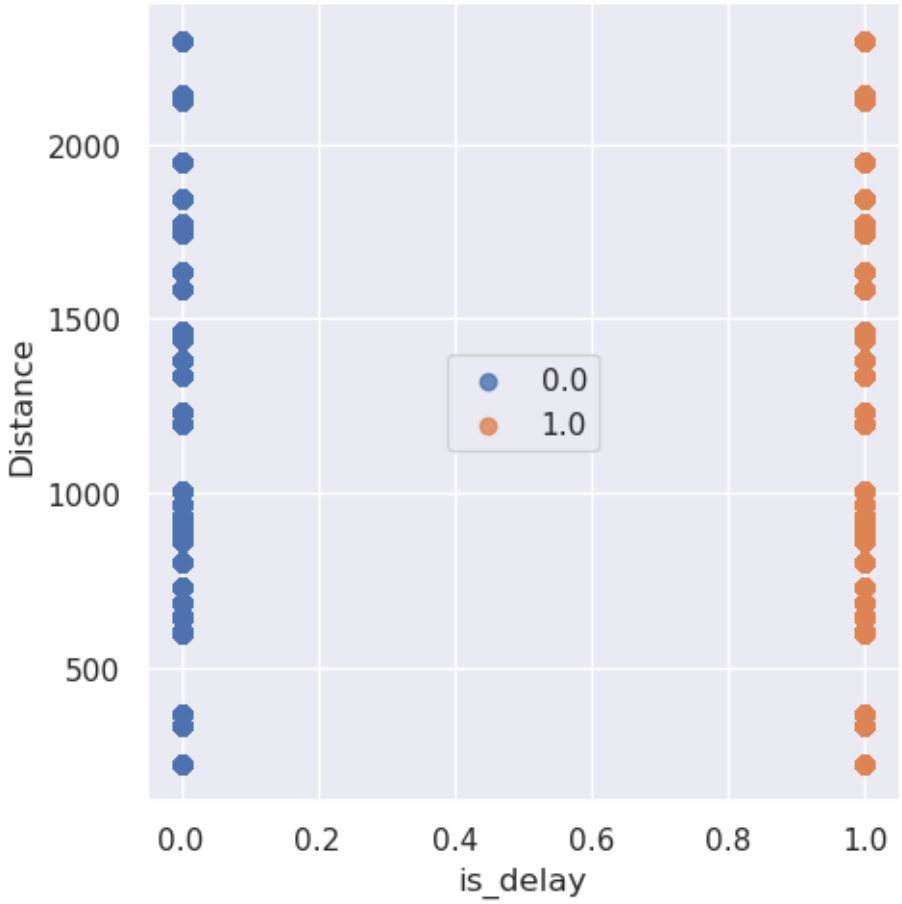
Enter your answer here - It seems to be about 80/20

Run the following two cells and answer the questions.

[26]:
```
viz_columns = ['Month', 'DepHourofDay', 'DayOfWeek', 'Reporting_Airline',␣
 ↪'Origin', 'Dest']
fig, axes = plt.subplots(3, 2, figsize=(20,20), squeeze=False)
# fig.autofmt_xdate(rotation=90)

for idx, column in enumerate(viz_columns):
    ax = axes[idx//2, idx%2]
    temp = data.groupby(column)['is_delay'].value_counts(normalize=True).
 ↪rename('percentage').\
    mul(100).reset_index().sort_values(column)
    sns.barplot(x=column, y="percentage", hue="is_delay", data=temp, ax=ax)
    plt.ylabel('% delay/no-delay')


plt.show()
```



[27]:
```
sns.lmplot( x="is_delay", y="Distance", data=data, fit_reg=False,␣
 ↪hue='is_delay', legend=False)
plt.legend(loc='center')
plt.xlabel('is_delay')
plt.ylabel('Distance')
plt.show()
```



Questions

Using the data from the previous charts, answer these questions:

- Which months have the most delays?
  - June and August
- What time of the day has the most delays?
  - between 4PM and 9PM
- What day of the week has the most delays?
  - Monday, Thursday, and Friday
- Which airline has the most delays?
  - Southwest Airlines
- Which origin and destination airports have the most delays?
  - Origin: Chicago O'Hare
  - Destination: San Fransisco
- Is flight distance a factor in the delays?
  - No, there doesn't seem to be any correlation with distance and delays

3.1.2 Features

Look at all the columns and what their specific types are.

[28]:
```
data.columns
```

[28]:
```
Index(['Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek', 'FlightDate',
       'Reporting_Airline', 'Origin', 'OriginState', 'Dest', 'DestState',
       'CRSDepTime', 'Cancelled', 'Diverted', 'Distance', 'DistanceGroup',
       'ArrDelay', 'ArrDelayMinutes', 'is_delay', 'AirTime', 'DepHourofDay'],
      dtype='object')
```

[29]:
```
data.dtypes
```

[29]:
```
Year                 int64
Quarter              int64
Month                int64
DayofMonth           int64
DayOfWeek            int64
FlightDate          object
Reporting_Airline   object
Origin              object
OriginState         object
Dest                object
DestState           object
CRSDepTime           int64
Cancelled          float64
Diverted           float64
Distance           float64
DistanceGroup        int64
ArrDelay           float64
ArrDelayMinutes    float64
is_delay           float64
AirTime            float64
DepHourofDay         int64
dtype: object
```

Filtering the required columns: - *Date* is redundant, because you have *Year*, *Quarter*, *Month*, *DayofMonth*, and *DayOfWeek* to describe the date. - Use *Origin* and *Dest* codes instead of *OriginState* and *DestState*. - Because you are only classifying whether the flight is delayed or not, you don't need *TotalDelayMinutes*, *DepDelayMinutes*, and *ArrDelayMinutes*.

Treat *DepHourofDay* as a categorical variable because it doesn't have any quantitative relation with the target. - If you needed to do a one-hot encoding of this variable, it would result in 23 more columns. - Other alternatives to handling categorical variables include hash encoding, regularized mean encoding, and bucketizing the values, among others. - In this case, you only need to split into buckets.

To change a column type to category, use the `astype` function (pandas.DataFrame.astype documentation).

[30]:
```
data_orig = data.copy()
data = data[[ 'is_delay', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek',
       'Reporting_Airline', 'Origin', 'Dest','Distance','DepHourofDay']]
categorical_columns  = ['Quarter', 'Month', 'DayofMonth', 'DayOfWeek',
       'Reporting_Airline', 'Origin', 'Dest', 'DepHourofDay']
for c in categorical_columns:
    data[c] = data[c].astype('category')
```

To use one-hot encoding, use the `get_dummies` function in pandas for the categorical columns that you selected. Then, you can concatenate those generated features to your original dataset by using the `concat` function in pandas. For encoding categorical variables, you can also use *dummy encoding* by using a keyword `drop_first=True`. For more information about dummy encoding, see Dummy variable (statistics).

For example:

`pd.get_dummies(df[['column1','columns2']], drop_first=True)`

[31]:
```
# Enter your code here
data_dummies = pd.get_dummies(data[categorical_columns], drop_first=True)
data_dummies = data_dummies.replace({True: 1, False: 0})
data = pd.concat([data, data_dummies], axis = 1)
data.drop(categorical_columns,axis=1, inplace=True)
```

Check the length of the dataset and the new columns.

Hint: Use the `shape` and `columns` properties.

[32]:
```
# Enter your code here
data.shape
```

[32]:
```
(1635590, 94)
```

[33]:
```
# Enter your code here
data.columns
```

[33]:
```
Index(['is_delay', 'Distance', 'Quarter_2', 'Quarter_3', 'Quarter_4',
       'Month_2', 'Month_3', 'Month_4', 'Month_5', 'Month_6', 'Month_7',
       'Month_8', 'Month_9', 'Month_10', 'Month_11', 'Month_12',
       'DayofMonth_2', 'DayofMonth_3', 'DayofMonth_4', 'DayofMonth_5',
       'DayofMonth_6', 'DayofMonth_7', 'DayofMonth_8', 'DayofMonth_9',
       'DayofMonth_10', 'DayofMonth_11', 'DayofMonth_12', 'DayofMonth_13',
       'DayofMonth_14', 'DayofMonth_15', 'DayofMonth_16', 'DayofMonth_17',
       'DayofMonth_18', 'DayofMonth_19', 'DayofMonth_20', 'DayofMonth_21',
       'DayofMonth_22', 'DayofMonth_23', 'DayofMonth_24', 'DayofMonth_25',
       'DayofMonth_26', 'DayofMonth_27', 'DayofMonth_28', 'DayofMonth_29',
       'DayofMonth_30', 'DayofMonth_31', 'DayOfWeek_2', 'DayOfWeek_3',
       'DayOfWeek_4', 'DayOfWeek_5', 'DayOfWeek_6', 'DayOfWeek_7',
       'Reporting_Airline_DL', 'Reporting_Airline_OO', 'Reporting_Airline_UA',
       'Reporting_Airline_WN', 'Origin_CLT', 'Origin_DEN', 'Origin_DFW',
```

```
       'Origin_IAH', 'Origin_LAX', 'Origin_ORD', 'Origin_PHX', 'Origin_SFO',
       'Dest_CLT', 'Dest_DEN', 'Dest_DFW', 'Dest_IAH', 'Dest_LAX', 'Dest_ORD',
       'Dest_PHX', 'Dest_SFO', 'DepHourofDay_1', 'DepHourofDay_2',
       'DepHourofDay_4', 'DepHourofDay_5', 'DepHourofDay_6', 'DepHourofDay_7',
       'DepHourofDay_8', 'DepHourofDay_9', 'DepHourofDay_10',
       'DepHourofDay_11', 'DepHourofDay_12', 'DepHourofDay_13',
       'DepHourofDay_14', 'DepHourofDay_15', 'DepHourofDay_16',
       'DepHourofDay_17', 'DepHourofDay_18', 'DepHourofDay_19',
       'DepHourofDay_20', 'DepHourofDay_21', 'DepHourofDay_22',
       'DepHourofDay_23'],
      dtype='object')
```

You are now ready to train the model. Before you split the data, rename the **is_delay** column to *target*.

Hint: You can use the `rename` function in pandas (pandas.DataFrame.rename documentation).

[34]:
```
# Enter your code here
data.rename(columns={'is_delay': 'target'}, inplace=True)
```

3.2 End of Step 2

Save the project file to your local computer. Follow these steps:

1. In the file explorer on the left, right-click the notebook that you're working on.
2. Choose **Download**, and save the file locally.

This action downloads the current notebook to the default download folder on your computer.

4 Step 3: Model training and evaluation

You must include some preliminary steps when you convert the dataset from a DataFrame to a format that a machine learning algorithm can use. For Amazon SageMaker, you must perform these steps:

1. Split the data into `train_data`, `validation_data`, and `test_data` by using `sklearn.model_selection.train_test_split`.
2. Convert the dataset to an appropriate file format that the Amazon SageMaker training job can use. This can be either a CSV file or record protobuf. For more information, see Common Data Formats for Training.
3. Upload the data to your S3 bucket. If you haven't created one before, see Create a Bucket.

Use the following cells to complete these steps. Insert and delete cells where needed.

Project presentation: In your project presentation, write down the key decisions that you made in this phase.

4.0.1 Train-test split

[35]:
```
from sklearn.model_selection import train_test_split
def split_data(data):
    train, test_and_validate = train_test_split(data, test_size=0.2,␣
 ↪random_state=42, stratify=data['target'])
    test, validate = train_test_split(test_and_validate, test_size=0.5,␣
 ↪random_state=42, stratify=test_and_validate['target'])
    return train, validate, test
```

[36]:
```
train, validate, test = split_data(data)
print(train['target'].value_counts())
print(test['target'].value_counts())
print(validate['target'].value_counts())
```

```
0.0    1033806
1.0     274666
Name: target, dtype: int64
0.0    129226
1.0     34333
Name: target, dtype: int64
0.0    129226
1.0     34333
Name: target, dtype: int64
```

Sample answer

```
0.0    1033570
1.0     274902
Name: target, dtype: int64
0.0    129076
1.0     34483
Name: target, dtype: int64
0.0    129612
1.0     33947
Name: target, dtype: int64
```

4.0.2 Baseline classification model

[37]:
```
import sagemaker
from sagemaker.serializers import CSVSerializer
from sagemaker.amazon.amazon_estimator import RecordSet
import boto3

# Instantiate the LinearLearner estimator object with 1 ml.m4.xlarge
classifier_estimator = sagemaker.LinearLearner(role=sagemaker.
 ↪get_execution_role(),
                                               instance_count=1,
                                               instance_type='ml.m4.xlarge',
```

```
                                                    ␣
 ↪predictor_type='binary_classifier',
                                                    ␣
 ↪binary_classifier_model_selection_criteria='cross_entropy_loss')
```

```
sagemaker.config INFO - Not applying SDK defaults from location:
/etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location:
/home/ec2-user/.config/sagemaker/config.yaml
```

4.0.3 Sample code

```
num_classes = len(pd.unique(train_labels))
classifier_estimator = sagemaker.LinearLearner(role=sagemaker.get_execution_role(),
                                               instance_count=1,
                                               instance_type='ml.m4.xlarge',
                                               predictor_type='binary_classifier',
                                               binary_classifier_model_selection_criteria = 'cr
```

Linear learner accepts training data in protobuf or CSV content types. It also accepts inference requests in protobuf, CSV, or JavaScript Object Notation (JSON) content types. Training data has features and ground-truth labels, but the data in an inference request has only features.

In a production pipeline, AWS recommends converting the data to the Amazon SageMaker protobuf format and storing it in Amazon S3. To get up and running quickly, AWS provides the `record_set` operation for converting and uploading the dataset when it's small enough to fit in local memory. It accepts NumPy arrays like the ones you already have, so you will use it for this step. The `RecordSet` object will track the temporary Amazon S3 location of your data. Create train, validation, and test records by using the `estimator.record_set` function. Then, start your training job by using the `estimator.fit` function.

[38]:
```
### Create train, validate, and test records
train_records = classifier_estimator.record_set(train.values[:, 1:].astype(np.
 ↪float32), train.values[:, 0].astype(np.float32), channel='train')
val_records = classifier_estimator.record_set(validate.values[:, 1:].astype(np.
 ↪float32), validate.values[:, 0].astype(np.float32), channel='validation')
test_records = classifier_estimator.record_set(test.values[:, 1:].astype(np.
 ↪float32), test.values[:, 0].astype(np.float32), channel='test')
```

Now, train your model on the dataset that you just uploaded.

4.0.4 Sample code

```
linear.fit([train_records,val_records,test_records])
```

[39]:
```
# Fit the classifier
# Enter your code here
classifier_estimator.fit([train_records, val_records, test_records])
```

```
INFO:sagemaker.image_uris:Same images used for training and inference.
Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating training-job with name: linear-
learner-2024-10-24-19-57-55-060

2024-10-24 19:57:56 Starting - Starting the training job…
2024-10-24 19:58:22 Starting - Preparing the instances for training…
2024-10-24 19:59:10 Downloading - Downloading input data…
2024-10-24 19:59:55 Downloading - Downloading the training image…
2024-10-24 20:01:06 Training - Training image download completed. Training in
progress.Docker entrypoint called with argument(s): train
Running default environment configuration script
[10/24/2024 20:01:21 INFO 140356604819264] Reading default configuration
from /opt/amazon/lib/python3.8/site-packages/algorithm/resources/default-
input.json: {'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': 'auto',
'use_bias': 'true', 'binary_classifier_model_selection_criteria': 'accuracy',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false'}
[10/24/2024 20:01:21 INFO 140356604819264] Merging with provided
configuration from /opt/ml/input/config/hyperparameters.json:
{'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'feature_dim': '93', 'mini_batch_size': '1000', 'predictor_type':
'binary_classifier'}
```

```
[10/24/2024 20:01:21 INFO 140356604819264] Final configuration:
{'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': '93', 'use_bias':
'true', 'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false',
'predictor_type': 'binary_classifier'}
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:01:25 WARNING 140356604819264] Loggers have already been
setup.
```

```
[10/24/2024 20:01:25 INFO 140356604819264] Final configuration:
{'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': '93', 'use_bias':
'true', 'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false',
'predictor_type': 'binary_classifier'}
[10/24/2024 20:01:25 WARNING 140356604819264] Loggers have already been
setup.
Process 7 is a worker.
[10/24/2024 20:01:25 INFO 140356604819264] Using default worker.
[10/24/2024 20:01:25 INFO 140356604819264] Checkpoint loading and saving
are disabled.
[2024-10-24 20:01:25.142] [tensorio] [warning] TensorIO is already
initialized; ignoring the initialization routine.
[2024-10-24 20:01:25.147] [tensorio] [warning] TensorIO is already
initialized; ignoring the initialization routine.
[2024-10-24 20:01:25.192] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 0, "duration": 55, "num_examples": 1,
"num_bytes": 420000}
[10/24/2024 20:01:25 INFO 140356604819264] Create Store: local
```

```
[2024-10-24 20:01:25.375] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 1, "duration": 181, "num_examples": 11,
"num_bytes": 4620000}
[10/24/2024 20:01:25 INFO 140356604819264] Scaler algorithm parameters
 <algorithm.scaler.ScalerAlgorithmStable object at 0x7fa6c140dd90>
[10/24/2024 20:01:25 INFO 140356604819264] Scaling model computed with
parameters:
 {'stdev_label': None, 'stdev_weight':
[5.3676862e+02 4.3185183e-01 4.4104236e-01 4.3405735e-01 2.5499377e-01
 2.7697837e-01 2.6860031e-01 2.7711493e-01 2.8022981e-01 2.8747982e-01
 2.8500608e-01 2.7793241e-01 2.8434965e-01 2.7185535e-01 2.7546957e-01
 1.8719329e-01 1.7257783e-01 1.7911108e-01 1.7887399e-01 1.8285699e-01
 1.7887397e-01 1.7744364e-01 1.8239361e-01 1.7911111e-01 1.7768298e-01
 1.8308821e-01 1.6907685e-01 1.8029131e-01 1.8122916e-01 1.7863652e-01
 1.7624156e-01 1.8076093e-01 1.8331905e-01 1.8377975e-01 1.7429848e-01
 1.7599998e-01 1.7768294e-01 1.7839867e-01 1.6678227e-01 1.8076095e-01
 1.8262547e-01 1.7958426e-01 1.7033654e-01 1.6549201e-01 1.3491301e-01
 3.5182565e-01 3.4942272e-01 3.5455844e-01 3.6190468e-01 3.2739079e-01
 3.4960875e-01 3.7354308e-01 2.3815936e-01 4.5729917e-01 3.1300303e-01
 2.4668624e-01 3.2000676e-01 3.2801935e-01 2.6774195e-01 3.5902092e-01
 3.3630344e-01 3.0024225e-01 3.2410914e-01 2.4684641e-01 3.1232232e-01
 3.2978889e-01 2.8022978e-01 3.5519007e-01 3.3789679e-01 3.0084661e-01
 3.2357445e-01 5.7114072e-02 1.6512204e-02 1.0000000e+00 1.6128524e-01
 2.2868866e-01 2.5127733e-01 2.5236887e-01 2.3849465e-01 2.4796402e-01
 2.4588311e-01 2.4377874e-01 2.3681191e-01 2.4032813e-01 2.3949701e-01
 2.4247177e-01 2.4442877e-01 2.1719252e-01 2.1869417e-01 1.9467381e-01
 1.5191342e-01 1.6419035e-01 1.1405288e-01]
<NDArray 93 @cpu(0)>, 'mean_label': None, 'mean_weight':
```

```
[1.01142212e+03 2.48000026e-01 2.64454544e-01 2.51818180e-01
 6.99091032e-02 8.37272853e-02 7.82727376e-02 8.38181973e-02
 8.59091058e-02 9.09090936e-02 8.91818330e-02 8.43636468e-02
 8.87272656e-02 8.03636387e-02 8.27272832e-02 3.63636389e-02
 3.07272747e-02 3.31818201e-02 3.30909118e-02 3.46363671e-02
 3.30909118e-02 3.25454548e-02 3.44545469e-02 3.31818238e-02
 3.26363668e-02 3.47272791e-02 2.94545498e-02 3.36363688e-02
 3.39999981e-02 3.30000035e-02 3.20909098e-02 3.38181816e-02
 3.48181799e-02 3.50000001e-02 3.13636400e-02 3.20000015e-02
 3.26363668e-02 3.29090916e-02 2.86363643e-02 3.38181816e-02
 3.45454589e-02 3.33636366e-02 2.99090929e-02 2.81818211e-02
 1.85454562e-02 1.44727305e-01 1.42363638e-01 1.47454545e-01
 1.55000016e-01 1.22090913e-01 1.42545491e-01 1.67636365e-01
 6.03636391e-02 2.97818184e-01 1.10090919e-01 6.50909096e-02
 1.15818188e-01 1.22636378e-01 7.77272731e-02 1.52000010e-01
 1.30000010e-01 1.00181833e-01 1.19272724e-01 6.51818290e-02
 1.09545454e-01 1.24181822e-01 8.59091058e-02 1.48090929e-01
 1.31454557e-01 1.00636370e-01 1.18818179e-01 3.27272760e-03
 2.72727280e-04 0.00000000e+00 2.67272722e-02 5.53636402e-02
 6.77272677e-02 6.83636367e-02 6.05454594e-02 6.58181757e-02
 6.46363646e-02 6.34545535e-02 5.96363731e-02 6.15454577e-02
 6.10909164e-02 6.27272800e-02 6.38181940e-02 4.96363677e-02
 5.03636375e-02 3.94545458e-02 2.36363672e-02 2.77272761e-02
 1.31818196e-02]
<NDArray 93 @cpu(0)>}
/opt/amazon/python3.8/lib/python3.8/subprocess.py:848: RuntimeWarning: line
buffering (buffering=1) isn't supported in binary mode, the default buffer size
will be used
  self.stdout = io.open(c2pread, 'rb', bufsize)
[10/24/2024 20:01:25 INFO 140356604819264] nvidia-smi: took 0.037 seconds
to run.
[10/24/2024 20:01:25 INFO 140356604819264] nvidia-smi identified 0
GPUs.
```

[10/24/2024 20:01:25 INFO 140356604819264] Number of GPUs being used: 0
#metrics {"StartTime": 1729800085.4938643, "EndTime": 1729800085.4939065,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "Meta": "init_train_data_iter"}, "Metrics": {"Total Records Seen":
{"sum": 12000.0, "count": 1, "min": 12000, "max": 12000}, "Total Batches Seen":
{"sum": 12.0, "count": 1, "min": 12, "max": 12}, "Max Records Seen Between
Resets": {"sum": 11000.0, "count": 1, "min": 11000, "max": 11000}, "Max Batches
Seen Between Resets": {"sum": 11.0, "count": 1, "min": 11, "max": 11}, "Reset
Count": {"sum": 2.0, "count": 1, "min": 2, "max": 2}, "Number of Records Since
Last Reset": {"sum": 0.0, "count": 1, "min": 0, "max": 0}, "Number of Batches
Since Last Reset": {"sum": 0.0, "count": 1, "min": 0, "max": 0}}}
[2024-10-24 20:02:07.286] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 3, "duration": 41792, "num_examples": 1309,
"num_bytes": 549558240}
#metrics {"StartTime": 1729800127.286857, "EndTime": 1729800127.2869453,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49445161762295875, "count": 1, "min": 0.49445161762295875, "max":
0.49445161762295875}}}
#metrics {"StartTime": 1729800127.2870529, "EndTime": 1729800127.2870698,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49505131423436904, "count": 1, "min": 0.49505131423436904, "max":
0.49505131423436904}}}
#metrics {"StartTime": 1729800127.2871022, "EndTime": 1729800127.2871108,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49453589338483434, "count": 1, "min": 0.49453589338483434, "max":
0.49453589338483434}}}

#metrics {"StartTime": 1729800127.287153, "EndTime": 1729800127.2871668,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4950743450433107, "count": 1, "min": 0.4950743450433107, "max":
0.4950743450433107}}}
#metrics {"StartTime": 1729800127.2872212, "EndTime": 1729800127.2872345,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507805377155269, "count": 1, "min": 0.507805377155269, "max":
0.507805377155269}}}
#metrics {"StartTime": 1729800127.2872784, "EndTime": 1729800127.2872925,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5186418846666995, "count": 1, "min": 0.5186418846666995, "max":
0.5186418846666995}}}
#metrics {"StartTime": 1729800127.2873454, "EndTime": 1729800127.2873602,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078239459349839, "count": 1, "min": 0.5078239459349839, "max":
0.5078239459349839}}}
#metrics {"StartTime": 1729800127.287417, "EndTime": 1729800127.287432,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5186900423615716, "count": 1, "min": 0.5186900423615716, "max":
0.5186900423615716}}}

#metrics {"StartTime": 1729800127.287481, "EndTime": 1729800127.287497,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4945912940582369, "count": 1, "min": 0.4945912940582369, "max":
0.4945912940582369}}}
#metrics {"StartTime": 1729800127.287547, "EndTime": 1729800127.287562,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49506217963542415, "count": 1, "min": 0.49506217963542415, "max":
0.49506217963542415}}}
#metrics {"StartTime": 1729800127.2876194, "EndTime": 1729800127.2876358,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.494642915591552, "count": 1, "min": 0.494642915591552, "max":
0.494642915591552}}}
#metrics {"StartTime": 1729800127.2876883, "EndTime": 1729800127.2877028,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4950726842384455, "count": 1, "min": 0.4950726842384455, "max":
0.4950726842384455}}}
#metrics {"StartTime": 1729800127.2877605, "EndTime": 1729800127.2877767,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5069791751826574, "count": 1, "min": 0.5069791751826574, "max":
0.5069791751826574}}}

#metrics {"StartTime": 1729800127.2878287, "EndTime": 1729800127.2878454,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5169523872118842, "count": 1, "min": 0.5169523872118842, "max":
0.5169523872118842}}}
#metrics {"StartTime": 1729800127.2878962, "EndTime": 1729800127.2879105,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5070459675453489, "count": 1, "min": 0.5070459675453489, "max":
0.5070459675453489}}}
#metrics {"StartTime": 1729800127.2879648, "EndTime": 1729800127.2879798,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5169103758692377, "count": 1, "min": 0.5169103758692377, "max":
0.5169103758692377}}}
#metrics {"StartTime": 1729800127.2880352, "EndTime": 1729800127.2880502,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5092376513699873, "count": 1, "min": 0.5092376513699873, "max":
0.5092376513699873}}}
#metrics {"StartTime": 1729800127.2881012, "EndTime": 1729800127.2881157,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5093272020299136, "count": 1, "min": 0.5093272020299136, "max":
0.5093272020299136}}}

#metrics {"StartTime": 1729800127.288162, "EndTime": 1729800127.28818,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5092102044715064, "count": 1, "min": 0.5092102044715064, "max":
0.5092102044715064}}}
#metrics {"StartTime": 1729800127.2882333, "EndTime": 1729800127.288249,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5093262071755319, "count": 1, "min": 0.5093262071755319, "max":
0.5093262071755319}}}
#metrics {"StartTime": 1729800127.2883005, "EndTime": 1729800127.2883155,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5106375551836206, "count": 1, "min": 0.5106375551836206, "max":
0.5106375551836206}}}
#metrics {"StartTime": 1729800127.2883692, "EndTime": 1729800127.288385,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5124191874932805, "count": 1, "min": 0.5124191874932805, "max":
0.5124191874932805}}}
#metrics {"StartTime": 1729800127.288444, "EndTime": 1729800127.2884593,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5106607389537566, "count": 1, "min": 0.5106607389537566, "max":
0.5106607389537566}}}

#metrics {"StartTime": 1729800127.2885103, "EndTime": 1729800127.2885263,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5124128902563626, "count": 1, "min": 0.5124128902563626, "max":
0.5124128902563626}}}
#metrics {"StartTime": 1729800127.2885766, "EndTime": 1729800127.2885919,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150533366538698, "count": 1, "min": 0.5150533366538698, "max":
0.5150533366538698}}}
#metrics {"StartTime": 1729800127.2886493, "EndTime": 1729800127.2886646,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5151092435970948, "count": 1, "min": 0.5151092435970948, "max":
0.5151092435970948}}}
#metrics {"StartTime": 1729800127.2887197, "EndTime": 1729800127.2887347,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150678829694743, "count": 1, "min": 0.5150678829694743, "max":
0.5150678829694743}}}
#metrics {"StartTime": 1729800127.2887826, "EndTime": 1729800127.2887976,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150988247839318, "count": 1, "min": 0.5150988247839318, "max":
0.5150988247839318}}}

#metrics {"StartTime": 1729800127.2888424, "EndTime": 1729800127.2888563,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150801241580135, "count": 1, "min": 0.5150801241580135, "max":
0.5150801241580135}}}
#metrics {"StartTime": 1729800127.2889006, "EndTime": 1729800127.2889154,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5158264294078955, "count": 1, "min": 0.5158264294078955, "max":
0.5158264294078955}}}
#metrics {"StartTime": 1729800127.2889602, "EndTime": 1729800127.2889745,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150880644212076, "count": 1, "min": 0.5150880644212076, "max":
0.5150880644212076}}}
#metrics {"StartTime": 1729800127.2890198, "EndTime": 1729800127.2890346,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.515788094173513, "count": 1, "min": 0.515788094173513, "max":
0.515788094173513}}}
[10/24/2024 20:02:07 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=0, train binary_classification_cross_entropy_objective
<loss>=0.49445161762295875
[2024-10-24 20:02:07.315] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 0, "duration": 42172, "num_examples":
1, "num_bytes": 420000}
[2024-10-24 20:02:11.574] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 2, "duration": 4259, "num_examples":
164, "num_bytes": 68694780}

#metrics {"StartTime": 1729800131.5835762, "EndTime": 1729800131.5836358,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49160091893266056, "count": 1, "min": 0.49160091893266056, "max":
0.49160091893266056}}}
#metrics {"StartTime": 1729800131.5837255, "EndTime": 1729800131.5837455,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49282310316561034, "count": 1, "min": 0.49282310316561034, "max":
0.49282310316561034}}}
#metrics {"StartTime": 1729800131.583803, "EndTime": 1729800131.5838203,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.491597603325673, "count": 1, "min": 0.491597603325673, "max":
0.491597603325673}}}
#metrics {"StartTime": 1729800131.5838647, "EndTime": 1729800131.583879,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4928150548421771, "count": 1, "min": 0.4928150548421771, "max":
0.4928150548421771}}}
#metrics {"StartTime": 1729800131.583923, "EndTime": 1729800131.583939,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5004385824096613, "count": 1, "min": 0.5004385824096613, "max":
0.5004385824096613}}}

#metrics {"StartTime": 1729800131.5839899, "EndTime": 1729800131.5840046,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5120320966907154, "count": 1, "min": 0.5120320966907154, "max":
0.5120320966907154}}}
#metrics {"StartTime": 1729800131.5840533, "EndTime": 1729800131.584069,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5004145083414806, "count": 1, "min": 0.5004145083414806, "max":
0.5004145083414806}}}
#metrics {"StartTime": 1729800131.58412, "EndTime": 1729800131.584136,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5111137957680194, "count": 1, "min": 0.5111137957680194, "max":
0.5111137957680194}}}
#metrics {"StartTime": 1729800131.584183, "EndTime": 1729800131.584198,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49182105303307744, "count": 1, "min": 0.49182105303307744, "max":
0.49182105303307744}}}
#metrics {"StartTime": 1729800131.5842419, "EndTime": 1729800131.584256,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49297706319882323, "count": 1, "min": 0.49297706319882323, "max":
0.49297706319882323}}}

#metrics {"StartTime": 1729800131.5843, "EndTime": 1729800131.584314,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49182052220010725, "count": 1, "min": 0.49182052220010725, "max":
0.49182052220010725}}}
#metrics {"StartTime": 1729800131.5843623, "EndTime": 1729800131.5843775,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4929734748052619, "count": 1, "min": 0.4929734748052619, "max":
0.4929734748052619}}}
#metrics {"StartTime": 1729800131.5844316, "EndTime": 1729800131.584446,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5003362391191071, "count": 1, "min": 0.5003362391191071, "max":
0.5003362391191071}}}
#metrics {"StartTime": 1729800131.5844922, "EndTime": 1729800131.584506,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5092739031313815, "count": 1, "min": 0.5092739031313815, "max":
0.5092739031313815}}}
#metrics {"StartTime": 1729800131.584551, "EndTime": 1729800131.5845659,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5003268207055157, "count": 1, "min": 0.5003268207055157, "max":
0.5003268207055157}}}

#metrics {"StartTime": 1729800131.5846145, "EndTime": 1729800131.5846295,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5092742748077529, "count": 1, "min": 0.5092742748077529, "max":
0.5092742748077529}}}
#metrics {"StartTime": 1729800131.5846848, "EndTime": 1729800131.5847008,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077122462695276, "count": 1, "min": 0.5077122462695276, "max":
0.5077122462695276}}}
#metrics {"StartTime": 1729800131.584747, "EndTime": 1729800131.5847614,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5084983128743751, "count": 1, "min": 0.5084983128743751, "max":
0.5084983128743751}}}
#metrics {"StartTime": 1729800131.5848057, "EndTime": 1729800131.58482,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077076040465759, "count": 1, "min": 0.5077076040465759, "max":
0.5077076040465759}}}
#metrics {"StartTime": 1729800131.584876, "EndTime": 1729800131.5848858,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5085101440124435, "count": 1, "min": 0.5085101440124435, "max":
0.5085101440124435}}}

#metrics {"StartTime": 1729800131.5849369, "EndTime": 1729800131.5849524,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5111900362568659, "count": 1, "min": 0.5111900362568659, "max":
0.5111900362568659}}}
#metrics {"StartTime": 1729800131.5850022, "EndTime": 1729800131.5850172,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5099147620098576, "count": 1, "min": 0.5099147620098576, "max":
0.5099147620098576}}}
#metrics {"StartTime": 1729800131.585066, "EndTime": 1729800131.585081,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5112012697647046, "count": 1, "min": 0.5112012697647046, "max":
0.5112012697647046}}}
#metrics {"StartTime": 1729800131.5851378, "EndTime": 1729800131.5851533,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5099128560489473, "count": 1, "min": 0.5099128560489473, "max":
0.5099128560489473}}}
#metrics {"StartTime": 1729800131.5852103, "EndTime": 1729800131.585226,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137645875437749, "count": 1, "min": 0.5137645875437749, "max":
0.5137645875437749}}}

#metrics {"StartTime": 1729800131.5852723, "EndTime": 1729800131.5852869,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144468769560029, "count": 1, "min": 0.5144468769560029, "max":
0.5144468769560029}}}
#metrics {"StartTime": 1729800131.5853415, "EndTime": 1729800131.5853565,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137677428746564, "count": 1, "min": 0.5137677428746564, "max":
0.5137677428746564}}}
#metrics {"StartTime": 1729800131.585406, "EndTime": 1729800131.5854223,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144375811284436, "count": 1, "min": 0.5144375811284436, "max":
0.5144375811284436}}}
#metrics {"StartTime": 1729800131.5854702, "EndTime": 1729800131.5854847,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5153195159811167, "count": 1, "min": 0.5153195159811167, "max":
0.5153195159811167}}}
#metrics {"StartTime": 1729800131.5855289, "EndTime": 1729800131.5855434,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5138188315831128, "count": 1, "min": 0.5138188315831128, "max":
0.5138188315831128}}}

```
#metrics {"StartTime": 1729800131.5855997, "EndTime": 1729800131.5856154,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5153097005156587, "count": 1, "min": 0.5153097005156587, "max":
0.5153097005156587}}}
#metrics {"StartTime": 1729800131.5856652, "EndTime": 1729800131.5856795,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139941421102128, "count": 1, "min": 0.5139941421102128, "max":
0.5139941421102128}}}
[10/24/2024 20:02:11 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=0, validation binary_classification_cross_entropy_objective
<loss>=0.49160091893266056
[10/24/2024 20:02:11 INFO 140356604819264] #early_stopping_criteria_metric:
host=algo-1, epoch=0, criteria=binary_classification_cross_entropy_objective,
value=0.491597603325673
[10/24/2024 20:02:11 INFO 140356604819264] Epoch 0: Loss improved. Updating
best model
[10/24/2024 20:02:11 INFO 140356604819264] Saving model for epoch: 0
[10/24/2024 20:02:11 INFO 140356604819264] Saved checkpoint to
"/tmp/tmphp0jvh_l/mx-mod-0000.params"
[10/24/2024 20:02:11 INFO 140356604819264] #progress_metric: host=algo-1,
completed 6.666666666666667 % of epochs
```

#metrics {"StartTime": 1729800085.4942775, "EndTime": 1729800131.593861,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 1320472.0, "count": 1, "min": 1320472, "max": 1320472},
"Total Batches Seen": {"sum": 1321.0, "count": 1, "min": 1321, "max": 1321},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 3.0, "count": 1,
"min": 3, "max": 3}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:02:11 INFO 140356604819264] #throughput_metric: host=algo-1,
train throughput=28383.496663567053 records/second
[2024-10-24 20:02:53.229] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 5, "duration": 41634, "num_examples": 1309,
"num_bytes": 549558240}
#metrics {"StartTime": 1729800173.2291656, "EndTime": 1729800173.2292697,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4923544073309009, "count": 1, "min": 0.4923544073309009, "max":
0.4923544073309009}}}
#metrics {"StartTime": 1729800173.2293813, "EndTime": 1729800173.2294018,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.493343270678039, "count": 1, "min": 0.493343270678039, "max":
0.493343270678039}}}

#metrics {"StartTime": 1729800173.2294538, "EndTime": 1729800173.2294693,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4923566930228417, "count": 1, "min": 0.4923566930228417, "max":
0.4923566930228417}}}
#metrics {"StartTime": 1729800173.2295215, "EndTime": 1729800173.2295363,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49333944151525466, "count": 1, "min": 0.49333944151525466, "max":
0.49333944151525466}}}
#metrics {"StartTime": 1729800173.2295868, "EndTime": 1729800173.2296007,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49640422704387516, "count": 1, "min": 0.49640422704387516, "max":
0.49640422704387516}}}
#metrics {"StartTime": 1729800173.2296486, "EndTime": 1729800173.2296627,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5148287836943927, "count": 1, "min": 0.5148287836943927, "max":
0.5148287836943927}}}
#metrics {"StartTime": 1729800173.2297132, "EndTime": 1729800173.229728,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49640690762114453, "count": 1, "min": 0.49640690762114453, "max":
0.49640690762114453}}}

#metrics {"StartTime": 1729800173.2297864, "EndTime": 1729800173.229802,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5147686998559794, "count": 1, "min": 0.5147686998559794, "max":
0.5147686998559794}}}
#metrics {"StartTime": 1729800173.2298534, "EndTime": 1729800173.22987,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49252507676990753, "count": 1, "min": 0.49252507676990753, "max":
0.49252507676990753}}}
#metrics {"StartTime": 1729800173.2299185, "EndTime": 1729800173.2299602,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4934666142128294, "count": 1, "min": 0.4934666142128294, "max":
0.4934666142128294}}}
#metrics {"StartTime": 1729800173.230022, "EndTime": 1729800173.2300386,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4925244559425097, "count": 1, "min": 0.4925244559425097, "max":
0.4925244559425097}}}
#metrics {"StartTime": 1729800173.23009, "EndTime": 1729800173.2301042,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4934663531102172, "count": 1, "min": 0.4934663531102172, "max":
0.4934663531102172}}}

#metrics {"StartTime": 1729800173.2301607, "EndTime": 1729800173.230177,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4963528660065537, "count": 1, "min": 0.4963528660065537, "max":
0.4963528660065537}}}
#metrics {"StartTime": 1729800173.2302287, "EndTime": 1729800173.2302442,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5134636345026325, "count": 1, "min": 0.5134636345026325, "max":
0.5134636345026325}}}
#metrics {"StartTime": 1729800173.230295, "EndTime": 1729800173.2303104,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4963520320227387, "count": 1, "min": 0.4963520320227387, "max":
0.4963520320227387}}}
#metrics {"StartTime": 1729800173.2303605, "EndTime": 1729800173.230376,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5134636790190997, "count": 1, "min": 0.5134636790190997, "max":
0.5134636790190997}}}
#metrics {"StartTime": 1729800173.2304256, "EndTime": 1729800173.230446,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078131305566257, "count": 1, "min": 0.5078131305566257, "max":
0.5078131305566257}}}

#metrics {"StartTime": 1729800173.2304997, "EndTime": 1729800173.230515,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.508066865810196, "count": 1, "min": 0.508066865810196, "max":
0.508066865810196}}}
#metrics {"StartTime": 1729800173.2305655, "EndTime": 1729800173.2305825,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078131958147801, "count": 1, "min": 0.5078131958147801, "max":
0.5078131958147801}}}
#metrics {"StartTime": 1729800173.2306333, "EndTime": 1729800173.230648,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080669260287504, "count": 1, "min": 0.5080669260287504, "max":
0.5080669260287504}}}
#metrics {"StartTime": 1729800173.2307045, "EndTime": 1729800173.230721,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5086696236053373, "count": 1, "min": 0.5086696236053373, "max":
0.5086696236053373}}}
#metrics {"StartTime": 1729800173.2307708, "EndTime": 1729800173.2307863,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5115897032711484, "count": 1, "min": 0.5115897032711484, "max":
0.5115897032711484}}}

#metrics {"StartTime": 1729800173.2308452, "EndTime": 1729800173.2308621,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5086701415409006, "count": 1, "min": 0.5086701415409006, "max":
0.5086701415409006}}}
#metrics {"StartTime": 1729800173.230913, "EndTime": 1729800173.2309284,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5115892548867322, "count": 1, "min": 0.5115892548867322, "max":
0.5115892548867322}}}
#metrics {"StartTime": 1729800173.2309885, "EndTime": 1729800173.231005,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137811036649464, "count": 1, "min": 0.5137811036649464, "max":
0.5137811036649464}}}
#metrics {"StartTime": 1729800173.2310643, "EndTime": 1729800173.2310815,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138790364629996, "count": 1, "min": 0.5138790364629996, "max":
0.5138790364629996}}}
#metrics {"StartTime": 1729800173.2311394, "EndTime": 1729800173.2311554,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137815257314513, "count": 1, "min": 0.5137815257314513, "max":
0.5137815257314513}}}

#metrics {"StartTime": 1729800173.2312062, "EndTime": 1729800173.2312224,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138804039678078, "count": 1, "min": 0.5138804039678078, "max":
0.5138804039678078}}}
#metrics {"StartTime": 1729800173.2312748, "EndTime": 1729800173.23129,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5141604372776977, "count": 1, "min": 0.5141604372776977, "max":
0.5141604372776977}}}
#metrics {"StartTime": 1729800173.231347, "EndTime": 1729800173.2313633,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5152463864901015, "count": 1, "min": 0.5152463864901015, "max":
0.5152463864901015}}}
#metrics {"StartTime": 1729800173.2314138, "EndTime": 1729800173.231429,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5141615462463566, "count": 1, "min": 0.5141615462463566, "max":
0.5141615462463566}}}
#metrics {"StartTime": 1729800173.231485, "EndTime": 1729800173.2315001,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5152457514304998, "count": 1, "min": 0.5152457514304998, "max":
0.5152457514304998}}}
[10/24/2024 20:02:53 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=1, train binary_classification_cross_entropy_objective
<loss>=0.4923544073309009

[2024-10-24 20:02:56.972] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 5, "duration": 3721, "num_examples":
164, "num_bytes": 68694780}
#metrics {"StartTime": 1729800176.9805307, "EndTime": 1729800176.9806032,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49132780061161035, "count": 1, "min": 0.49132780061161035, "max":
0.49132780061161035}}}
#metrics {"StartTime": 1729800176.9806974, "EndTime": 1729800176.9807167,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4928287626473857, "count": 1, "min": 0.4928287626473857, "max":
0.4928287626473857}}}
#metrics {"StartTime": 1729800176.9807687, "EndTime": 1729800176.9807854,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49132624878812603, "count": 1, "min": 0.49132624878812603, "max":
0.49132624878812603}}}
#metrics {"StartTime": 1729800176.9808345, "EndTime": 1729800176.9808495,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49282696061405495, "count": 1, "min": 0.49282696061405495, "max":
0.49282696061405495}}}
#metrics {"StartTime": 1729800176.9809, "EndTime": 1729800176.9809139,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.493076644478672, "count": 1, "min": 0.493076644478672, "max":
0.493076644478672}}}

#metrics {"StartTime": 1729800176.9809566, "EndTime": 1729800176.980969,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5095907994060773, "count": 1, "min": 0.5095907994060773, "max":
0.5095907994060773}}}
#metrics {"StartTime": 1729800176.9810119, "EndTime": 1729800176.981027,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49307624276219225, "count": 1, "min": 0.49307624276219225, "max":
0.49307624276219225}}}
#metrics {"StartTime": 1729800176.981078, "EndTime": 1729800176.9810927,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5095608107950772, "count": 1, "min": 0.5095608107950772, "max":
0.5095608107950772}}}
#metrics {"StartTime": 1729800176.9811375, "EndTime": 1729800176.9811525,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4915669185678147, "count": 1, "min": 0.4915669185678147, "max":
0.4915669185678147}}}
#metrics {"StartTime": 1729800176.9811933, "EndTime": 1729800176.981207,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4930113113479167, "count": 1, "min": 0.4930113113479167, "max":
0.4930113113479167}}}

#metrics {"StartTime": 1729800176.9812512, "EndTime": 1729800176.9812665,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4915664808405219, "count": 1, "min": 0.4915664808405219, "max":
0.4915664808405219}}}
#metrics {"StartTime": 1729800176.981318, "EndTime": 1729800176.9813328,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49301041630195613, "count": 1, "min": 0.49301041630195613, "max":
0.49301041630195613}}}
#metrics {"StartTime": 1729800176.9813797, "EndTime": 1729800176.9813943,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49319221642537775, "count": 1, "min": 0.49319221642537775, "max":
0.49319221642537775}}}
#metrics {"StartTime": 1729800176.9814398, "EndTime": 1729800176.9814544,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.507777828117453, "count": 1, "min": 0.507777828117453, "max":
0.507777828117453}}}
#metrics {"StartTime": 1729800176.981505, "EndTime": 1729800176.98152,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49319294000016095, "count": 1, "min": 0.49319294000016095, "max":
0.49319294000016095}}}

#metrics {"StartTime": 1729800176.9815738, "EndTime": 1729800176.9815888,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077787681302839, "count": 1, "min": 0.5077787681302839, "max":
0.5077787681302839}}}
#metrics {"StartTime": 1729800176.9816446, "EndTime": 1729800176.9816608,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077417019951277, "count": 1, "min": 0.5077417019951277, "max":
0.5077417019951277}}}
#metrics {"StartTime": 1729800176.9817128, "EndTime": 1729800176.9817276,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5086740499336965, "count": 1, "min": 0.5086740499336965, "max":
0.5086740499336965}}}
#metrics {"StartTime": 1729800176.9817722, "EndTime": 1729800176.9817872,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077412510203337, "count": 1, "min": 0.5077412510203337, "max":
0.5077412510203337}}}
#metrics {"StartTime": 1729800176.981836, "EndTime": 1729800176.981851,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.508675980150595, "count": 1, "min": 0.508675980150595, "max":
0.508675980150595}}}

#metrics {"StartTime": 1729800176.981899, "EndTime": 1729800176.981914,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5080723708198618, "count": 1, "min": 0.5080723708198618, "max":
0.5080723708198618}}}
#metrics {"StartTime": 1729800176.9819846, "EndTime": 1729800176.9819996,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.510345066037368, "count": 1, "min": 0.510345066037368, "max":
0.510345066037368}}}
#metrics {"StartTime": 1729800176.9820495, "EndTime": 1729800176.9820652,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5080705068404693, "count": 1, "min": 0.5080705068404693, "max":
0.5080705068404693}}}
#metrics {"StartTime": 1729800176.9821167, "EndTime": 1729800176.982132,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5103459450370594, "count": 1, "min": 0.5103459450370594, "max":
0.5103459450370594}}}
#metrics {"StartTime": 1729800176.9821813, "EndTime": 1729800176.9821956,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137669095881733, "count": 1, "min": 0.5137669095881733, "max":
0.5137669095881733}}}

#metrics {"StartTime": 1729800176.9822433, "EndTime": 1729800176.982258,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5145788164703037, "count": 1, "min": 0.5145788164703037, "max":
0.5145788164703037}}}
#metrics {"StartTime": 1729800176.9823077, "EndTime": 1729800176.9823196,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137660656663722, "count": 1, "min": 0.5137660656663722, "max":
0.5137660656663722}}}
#metrics {"StartTime": 1729800176.9823632, "EndTime": 1729800176.9823718,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5145432941354974, "count": 1, "min": 0.5145432941354974, "max":
0.5145432941354974}}}
#metrics {"StartTime": 1729800176.9824014, "EndTime": 1729800176.982413,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139341081665174, "count": 1, "min": 0.5139341081665174, "max":
0.5139341081665174}}}
#metrics {"StartTime": 1729800176.9824605, "EndTime": 1729800176.9824748,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139470851201463, "count": 1, "min": 0.5139470851201463, "max":
0.5139470851201463}}}

#metrics {"StartTime": 1729800176.9825242, "EndTime": 1729800176.9825394, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 30}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5139453142464312, "count": 1, "min": 0.5139453142464312, "max": 0.5139453142464312}}}
#metrics {"StartTime": 1729800176.9825737, "EndTime": 1729800176.982587, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 31}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5140024169472579, "count": 1, "min": 0.5140024169472579, "max": 0.5140024169472579}}}
[10/24/2024 20:02:56 INFO 140356604819264] #quality_metric: host=algo-1, epoch=1, validation binary_classification_cross_entropy_objective <loss>=0.49132780061161035
[10/24/2024 20:02:56 INFO 140356604819264] #early_stopping_criteria_metric: host=algo-1, epoch=1, criteria=binary_classification_cross_entropy_objective, value=0.49132624878812603
[10/24/2024 20:02:56 INFO 140356604819264] Saving model for epoch: 1
[10/24/2024 20:02:56 INFO 140356604819264] Saved checkpoint to "/tmp/tmpfun52n1w/mx-mod-0000.params"
[10/24/2024 20:02:56 INFO 140356604819264] #progress_metric: host=algo-1, completed 13.333333333333334 % of epochs
#metrics {"StartTime": 1729800131.5942187, "EndTime": 1729800176.9891036, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "Meta": "training_data_iter"}, "Metrics": {"Total Records Seen": {"sum": 2628944.0, "count": 1, "min": 2628944, "max": 2628944}, "Total Batches Seen": {"sum": 2630.0, "count": 1, "min": 2630, "max": 2630}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 4.0, "count": 1, "min": 4, "max": 4}, "Number of Records Since Last Reset": {"sum": 1308472.0, "count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}

[10/24/2024 20:02:56 INFO 140356604819264] #throughput_metric: host=algo-1,
train throughput=28824.11215180907 records/second
[2024-10-24 20:03:38.380] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 7, "duration": 41390, "num_examples": 1309,
"num_bytes": 549558240}
#metrics {"StartTime": 1729800218.3804111, "EndTime": 1729800218.3805132,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4921841292075061, "count": 1, "min": 0.4921841292075061, "max":
0.4921841292075061}}}
#metrics {"StartTime": 1729800218.3806212, "EndTime": 1729800218.3806453,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49320107833908966, "count": 1, "min": 0.49320107833908966, "max":
0.49320107833908966}}}
#metrics {"StartTime": 1729800218.3806953, "EndTime": 1729800218.3807065,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4921864276850989, "count": 1, "min": 0.4921864276850989, "max":
0.4921864276850989}}}
#metrics {"StartTime": 1729800218.3807514, "EndTime": 1729800218.3807623,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49319977140280813, "count": 1, "min": 0.49319977140280813, "max":
0.49319977140280813}}}

#metrics {"StartTime": 1729800218.380809, "EndTime": 1729800218.380824,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49328891446174833, "count": 1, "min": 0.49328891446174833, "max":
0.49328891446174833}}}
#metrics {"StartTime": 1729800218.3808632, "EndTime": 1729800218.3808722,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5120100802033684, "count": 1, "min": 0.5120100802033684, "max":
0.5120100802033684}}}
#metrics {"StartTime": 1729800218.380898, "EndTime": 1729800218.3809104,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4932889357867226, "count": 1, "min": 0.4932889357867226, "max":
0.4932889357867226}}}
#metrics {"StartTime": 1729800218.3809566, "EndTime": 1729800218.3809707,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5120240621070978, "count": 1, "min": 0.5120240621070978, "max":
0.5120240621070978}}}
#metrics {"StartTime": 1729800218.3810198, "EndTime": 1729800218.3810349,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49237036802426026, "count": 1, "min": 0.49237036802426026, "max":
0.49237036802426026}}}

#metrics {"StartTime": 1729800218.3810856, "EndTime": 1729800218.3811007,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4933414632581425, "count": 1, "min": 0.4933414632581425, "max":
0.4933414632581425}}}
#metrics {"StartTime": 1729800218.3811457, "EndTime": 1729800218.3811615,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4923699998534783, "count": 1, "min": 0.4923699998534783, "max":
0.4923699998534783}}}
#metrics {"StartTime": 1729800218.3812075, "EndTime": 1729800218.381223,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49334138213458045, "count": 1, "min": 0.49334138213458045, "max":
0.49334138213458045}}}
#metrics {"StartTime": 1729800218.381273, "EndTime": 1729800218.3812883,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4934254325843368, "count": 1, "min": 0.4934254325843368, "max":
0.4934254325843368}}}
#metrics {"StartTime": 1729800218.381338, "EndTime": 1729800218.3813524,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5108510734721425, "count": 1, "min": 0.5108510734721425, "max":
0.5108510734721425}}}

#metrics {"StartTime": 1729800218.3813996, "EndTime": 1729800218.3814251,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4934253898410622, "count": 1, "min": 0.4934253898410622, "max":
0.4934253898410622}}}
#metrics {"StartTime": 1729800218.381474, "EndTime": 1729800218.3814898,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5108510219795623, "count": 1, "min": 0.5108510219795623, "max":
0.5108510219795623}}}
#metrics {"StartTime": 1729800218.3815367, "EndTime": 1729800218.3815506,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507786056728538, "count": 1, "min": 0.507786056728538, "max":
0.507786056728538}}}
#metrics {"StartTime": 1729800218.3815944, "EndTime": 1729800218.3816094,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080442258502366, "count": 1, "min": 0.5080442258502366, "max":
0.5080442258502366}}}
#metrics {"StartTime": 1729800218.381651, "EndTime": 1729800218.3816648,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077860342603211, "count": 1, "min": 0.5077860342603211, "max":
0.5077860342603211}}}

#metrics {"StartTime": 1729800218.381714, "EndTime": 1729800218.3817284,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080442374693145, "count": 1, "min": 0.5080442374693145, "max":
0.5080442374693145}}}
#metrics {"StartTime": 1729800218.3817692, "EndTime": 1729800218.3817825,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080313072554562, "count": 1, "min": 0.5080313072554562, "max":
0.5080313072554562}}}
#metrics {"StartTime": 1729800218.3818276, "EndTime": 1729800218.3818429,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5111188523878745, "count": 1, "min": 0.5111188523878745, "max":
0.5111188523878745}}}
#metrics {"StartTime": 1729800218.3818977, "EndTime": 1729800218.3819077,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080313420426955, "count": 1, "min": 0.5080313420426955, "max":
0.5080313420426955}}}
#metrics {"StartTime": 1729800218.3819685, "EndTime": 1729800218.3819845,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5111187683012143, "count": 1, "min": 0.5111187683012143, "max":
0.5111187683012143}}}

#metrics {"StartTime": 1729800218.3820336, "EndTime": 1729800218.3820493,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.513772972760215, "count": 1, "min": 0.513772972760215, "max":
0.513772972760215}}}
#metrics {"StartTime": 1729800218.3820992, "EndTime": 1729800218.3821154,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138817439881305, "count": 1, "min": 0.5138817439881305, "max":
0.5138817439881305}}}
#metrics {"StartTime": 1729800218.3821623, "EndTime": 1729800218.3821757,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.513772815646017, "count": 1, "min": 0.513772815646017, "max":
0.513772815646017}}}
#metrics {"StartTime": 1729800218.382232, "EndTime": 1729800218.3822477,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138813661814465, "count": 1, "min": 0.5138813661814465, "max":
0.5138813661814465}}}
#metrics {"StartTime": 1729800218.3823063, "EndTime": 1729800218.3823235,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138743831856171, "count": 1, "min": 0.5138743831856171, "max":
0.5138743831856171}}}

#metrics {"StartTime": 1729800218.3823783, "EndTime": 1729800218.3823943,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150690107899704, "count": 1, "min": 0.5150690107899704, "max":
0.5150690107899704}}}
#metrics {"StartTime": 1729800218.3824527, "EndTime": 1729800218.3824685,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138742513394137, "count": 1, "min": 0.5138742513394137, "max":
0.5138742513394137}}}
#metrics {"StartTime": 1729800218.3825333, "EndTime": 1729800218.3825488,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150694421424049, "count": 1, "min": 0.5150694421424049, "max":
0.5150694421424049}}}
[10/24/2024 20:03:38 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=2, train binary_classification_cross_entropy_objective
<loss>=0.4921841292075061
[2024-10-24 20:03:41.871] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 8, "duration": 3464, "num_examples":
164, "num_bytes": 68694780}
#metrics {"StartTime": 1729800221.8795063, "EndTime": 1729800221.8795767,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49120593908719246, "count": 1, "min": 0.49120593908719246, "max":
0.49120593908719246}}}

#metrics {"StartTime": 1729800221.8796825, "EndTime": 1729800221.8797033,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4926866624862578, "count": 1, "min": 0.4926866624862578, "max":
0.4926866624862578}}}
#metrics {"StartTime": 1729800221.879758, "EndTime": 1729800221.879774,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4912048020411045, "count": 1, "min": 0.4912048020411045, "max":
0.4912048020411045}}}
#metrics {"StartTime": 1729800221.879826, "EndTime": 1729800221.8798416,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.492686130347196, "count": 1, "min": 0.492686130347196, "max":
0.492686130347196}}}
#metrics {"StartTime": 1729800221.87989, "EndTime": 1729800221.879904,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49188916011594774, "count": 1, "min": 0.49188916011594774, "max":
0.49188916011594774}}}
#metrics {"StartTime": 1729800221.8799467, "EndTime": 1729800221.8799608,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5087600678247581, "count": 1, "min": 0.5087600678247581, "max":
0.5087600678247581}}}

#metrics {"StartTime": 1729800221.8800037, "EndTime": 1729800221.8800182,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49188920564257155, "count": 1, "min": 0.49188920564257155, "max":
0.49188920564257155}}}
#metrics {"StartTime": 1729800221.8800626, "EndTime": 1729800221.880078,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5078685147264687, "count": 1, "min": 0.5078685147264687, "max":
0.5078685147264687}}}
#metrics {"StartTime": 1729800221.8801243, "EndTime": 1729800221.8801396,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4914462299137133, "count": 1, "min": 0.4914462299137133, "max":
0.4914462299137133}}}
#metrics {"StartTime": 1729800221.8801885, "EndTime": 1729800221.880203,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4928836991018914, "count": 1, "min": 0.4928836991018914, "max":
0.4928836991018914}}}
#metrics {"StartTime": 1729800221.8802478, "EndTime": 1729800221.8802624,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49144595656738593, "count": 1, "min": 0.49144595656738593, "max":
0.49144595656738593}}}

#metrics {"StartTime": 1729800221.8803098, "EndTime": 1729800221.8803248,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49288342426288784, "count": 1, "min": 0.49288342426288784, "max":
0.49288342426288784}}}
#metrics {"StartTime": 1729800221.8803706, "EndTime": 1729800221.8803856,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.492117814598643, "count": 1, "min": 0.492117814598643, "max":
0.492117814598643}}}
#metrics {"StartTime": 1729800221.8804336, "EndTime": 1729800221.880449,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5063099090813321, "count": 1, "min": 0.5063099090813321, "max":
0.5063099090813321}}}
#metrics {"StartTime": 1729800221.8804958, "EndTime": 1729800221.88051,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4921177573171942, "count": 1, "min": 0.4921177573171942, "max":
0.4921177573171942}}}
#metrics {"StartTime": 1729800221.880555, "EndTime": 1729800221.8805704,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5063100316673642, "count": 1, "min": 0.5063100316673642, "max":
0.5063100316673642}}}

#metrics {"StartTime": 1729800221.880617, "EndTime": 1729800221.8806322,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077011480354686, "count": 1, "min": 0.5077011480354686, "max":
0.5077011480354686}}}
#metrics {"StartTime": 1729800221.8806806, "EndTime": 1729800221.8806958,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5086371188857356, "count": 1, "min": 0.5086371188857356, "max":
0.5086371188857356}}}
#metrics {"StartTime": 1729800221.8807447, "EndTime": 1729800221.8807578,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077012902128757, "count": 1, "min": 0.5077012902128757, "max":
0.5077012902128757}}}
#metrics {"StartTime": 1729800221.8808014, "EndTime": 1729800221.8808162,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5086374446623141, "count": 1, "min": 0.5086374446623141, "max":
0.5086374446623141}}}
#metrics {"StartTime": 1729800221.8808613, "EndTime": 1729800221.8808768,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5079359909670641, "count": 1, "min": 0.5079359909670641, "max":
0.5079359909670641}}}

#metrics {"StartTime": 1729800221.880927, "EndTime": 1729800221.8809419,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5099848241253682, "count": 1, "min": 0.5099848241253682, "max":
0.5099848241253682}}}
#metrics {"StartTime": 1729800221.8809912, "EndTime": 1729800221.8810065,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5079360904166152, "count": 1, "min": 0.5079360904166152, "max":
0.5079360904166152}}}
#metrics {"StartTime": 1729800221.8810534, "EndTime": 1729800221.881068,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5099848774885419, "count": 1, "min": 0.5099848774885419, "max":
0.5099848774885419}}}
#metrics {"StartTime": 1729800221.8811126, "EndTime": 1729800221.8811274,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137635135632559, "count": 1, "min": 0.5137635135632559, "max":
0.5137635135632559}}}
#metrics {"StartTime": 1729800221.8811755, "EndTime": 1729800221.881191,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5145031969347524, "count": 1, "min": 0.5145031969347524, "max":
0.5145031969347524}}}

#metrics {"StartTime": 1729800221.8812401, "EndTime": 1729800221.8812559,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137593962025697, "count": 1, "min": 0.5137593962025697, "max":
0.5137593962025697}}}
#metrics {"StartTime": 1729800221.8813033, "EndTime": 1729800221.8813174,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5145151879757577, "count": 1, "min": 0.5145151879757577, "max":
0.5145151879757577}}}
#metrics {"StartTime": 1729800221.8813624, "EndTime": 1729800221.8813772,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139562197386771, "count": 1, "min": 0.5139562197386771, "max":
0.5139562197386771}}}
#metrics {"StartTime": 1729800221.8814209, "EndTime": 1729800221.8814359,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139162206813916, "count": 1, "min": 0.5139162206813916, "max":
0.5139162206813916}}}
#metrics {"StartTime": 1729800221.8814843, "EndTime": 1729800221.8815,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139622124604118, "count": 1, "min": 0.5139622124604118, "max":
0.5139622124604118}}}

```
#metrics {"StartTime": 1729800221.8815494, "EndTime": 1729800221.8815646,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139162244130822, "count": 1, "min": 0.5139162244130822, "max":
0.5139162244130822}}}
[10/24/2024 20:03:41 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=2, validation binary_classification_cross_entropy_objective
<loss>=0.49120593908719246
[10/24/2024 20:03:41 INFO 140356604819264] #early_stopping_criteria_metric:
host=algo-1, epoch=2, criteria=binary_classification_cross_entropy_objective,
value=0.4912048020411045
[10/24/2024 20:03:41 INFO 140356604819264] Saving model for epoch: 2
[10/24/2024 20:03:41 INFO 140356604819264] Saved checkpoint to
"/tmp/tmptrx64w_4/mx-mod-0000.params"
[10/24/2024 20:03:41 INFO 140356604819264] #progress_metric: host=algo-1,
completed 20.0 % of epochs
#metrics {"StartTime": 1729800176.9897177, "EndTime": 1729800221.8901048,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 3937416.0, "count": 1, "min": 3937416, "max": 3937416},
"Total Batches Seen": {"sum": 3939.0, "count": 1, "min": 3939, "max": 3939},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 5.0, "count": 1,
"min": 5, "max": 5}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:03:41 INFO 140356604819264] #throughput_metric: host=algo-1,
train throughput=29141.558007694497 records/second
[2024-10-24 20:04:23.743] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 9, "duration": 41852, "num_examples": 1309,
"num_bytes": 549558240}
```

#metrics {"StartTime": 1729800263.7430663, "EndTime": 1729800263.74317,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4921148091820766, "count": 1, "min": 0.4921148091820766, "max":
0.4921148091820766}}}
#metrics {"StartTime": 1729800263.7432775, "EndTime": 1729800263.7433019,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49306685458775323, "count": 1, "min": 0.49306685458775323, "max":
0.49306685458775323}}}
#metrics {"StartTime": 1729800263.7433505, "EndTime": 1729800263.7433634,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4921323065101554, "count": 1, "min": 0.4921323065101554, "max":
0.4921323065101554}}}
#metrics {"StartTime": 1729800263.7433932, "EndTime": 1729800263.7434027,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49306629558878207, "count": 1, "min": 0.49306629558878207, "max":
0.49306629558878207}}}
#metrics {"StartTime": 1729800263.7434494, "EndTime": 1729800263.743464,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49242453889525994, "count": 1, "min": 0.49242453889525994, "max":
0.49242453889525994}}}

#metrics {"StartTime": 1729800263.7435098, "EndTime": 1729800263.7435222,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5095862446122942, "count": 1, "min": 0.5095862446122942, "max":
0.5095862446122942}}}
#metrics {"StartTime": 1729800263.743549, "EndTime": 1729800263.7435565,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4924245387319396, "count": 1, "min": 0.4924245387319396, "max":
0.4924245387319396}}}
#metrics {"StartTime": 1729800263.7435794, "EndTime": 1729800263.743586,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5095839901845389, "count": 1, "min": 0.5095839901845389, "max":
0.5095839901845389}}}
#metrics {"StartTime": 1729800263.7436326, "EndTime": 1729800263.7436476,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49230677692751634, "count": 1, "min": 0.49230677692751634, "max":
0.49230677692751634}}}
#metrics {"StartTime": 1729800263.7436984, "EndTime": 1729800263.7437143,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49321808701261466, "count": 1, "min": 0.49321808701261466, "max":
0.49321808701261466}}}

#metrics {"StartTime": 1729800263.7437637, "EndTime": 1729800263.7437787,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49232062626176654, "count": 1, "min": 0.49232062626176654, "max":
0.49232062626176654}}}
#metrics {"StartTime": 1729800263.743828, "EndTime": 1729800263.743843,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49321806914069966, "count": 1, "min": 0.49321806914069966, "max":
0.49321806914069966}}}
#metrics {"StartTime": 1729800263.743888, "EndTime": 1729800263.743902,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4926141987371882, "count": 1, "min": 0.4926141987371882, "max":
0.4926141987371882}}}
#metrics {"StartTime": 1729800263.7439494, "EndTime": 1729800263.743965,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5085899786992905, "count": 1, "min": 0.5085899786992905, "max":
0.5085899786992905}}}
#metrics {"StartTime": 1729800263.7440164, "EndTime": 1729800263.744032,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4926141959374104, "count": 1, "min": 0.4926141959374104, "max":
0.4926141959374104}}}

#metrics {"StartTime": 1729800263.7440782, "EndTime": 1729800263.7440941,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5085899629738717, "count": 1, "min": 0.5085899629738717, "max":
0.5085899629738717}}}
#metrics {"StartTime": 1729800263.744139, "EndTime": 1729800263.7441552,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077777675197029, "count": 1, "min": 0.5077777675197029, "max":
0.5077777675197029}}}
#metrics {"StartTime": 1729800263.7442014, "EndTime": 1729800263.7442172,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080107293318535, "count": 1, "min": 0.5080107293318535, "max":
0.5080107293318535}}}
#metrics {"StartTime": 1729800263.7442732, "EndTime": 1729800263.7442899,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077823283577548, "count": 1, "min": 0.5077823283577548, "max":
0.5077823283577548}}}
#metrics {"StartTime": 1729800263.7443352, "EndTime": 1729800263.7443507,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5080107348847462, "count": 1, "min": 0.5080107348847462, "max":
0.5080107348847462}}}

#metrics {"StartTime": 1729800263.7443936, "EndTime": 1729800263.7444136,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078313445111663, "count": 1, "min": 0.5078313445111663, "max":
0.5078313445111663}}}
#metrics {"StartTime": 1729800263.7444715, "EndTime": 1729800263.7444873,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5106897554076775, "count": 1, "min": 0.5106897554076775, "max":
0.5106897554076775}}}
#metrics {"StartTime": 1729800263.744529, "EndTime": 1729800263.7445433,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507831343531244, "count": 1, "min": 0.507831343531244, "max":
0.507831343531244}}}
#metrics {"StartTime": 1729800263.7445881, "EndTime": 1729800263.7446022,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5106897431819811, "count": 1, "min": 0.5106897431819811, "max":
0.5106897431819811}}}
#metrics {"StartTime": 1729800263.744652, "EndTime": 1729800263.7446673,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137682971079415, "count": 1, "min": 0.5137682971079415, "max":
0.5137682971079415}}}

#metrics {"StartTime": 1729800263.7447145, "EndTime": 1729800263.7447298,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138697870936962, "count": 1, "min": 0.5138697870936962, "max":
0.5138697870936962}}}
#metrics {"StartTime": 1729800263.7447774, "EndTime": 1729800263.7447925,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137703418556704, "count": 1, "min": 0.5137703418556704, "max":
0.5137703418556704}}}
#metrics {"StartTime": 1729800263.7448454, "EndTime": 1729800263.7448614,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138709132576572, "count": 1, "min": 0.5138709132576572, "max":
0.5138709132576572}}}
#metrics {"StartTime": 1729800263.7449112, "EndTime": 1729800263.744927,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137888718117997, "count": 1, "min": 0.5137888718117997, "max":
0.5137888718117997}}}
#metrics {"StartTime": 1729800263.7449782, "EndTime": 1729800263.7449882,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5148606979650096, "count": 1, "min": 0.5148606979650096, "max":
0.5148606979650096}}}

#metrics {"StartTime": 1729800263.7450254, "EndTime": 1729800263.745038,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.513789148173201, "count": 1, "min": 0.513789148173201, "max":
0.513789148173201}}}
#metrics {"StartTime": 1729800263.7450824, "EndTime": 1729800263.745097,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5148606981283299, "count": 1, "min": 0.5148606981283299, "max":
0.5148606981283299}}}
[10/24/2024 20:04:23 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=3, train binary_classification_cross_entropy_objective
<loss>=0.4921148091820766
[2024-10-24 20:04:27.626] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 11, "duration": 3861, "num_examples":
164, "num_bytes": 68694780}
#metrics {"StartTime": 1729800267.6343775, "EndTime": 1729800267.634439,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4911792199968026, "count": 1, "min": 0.4911792199968026, "max":
0.4911792199968026}}}
#metrics {"StartTime": 1729800267.6345289, "EndTime": 1729800267.6345499,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49250462036270887, "count": 1, "min": 0.49250462036270887, "max":
0.49250462036270887}}}

#metrics {"StartTime": 1729800267.6346145, "EndTime": 1729800267.6346307,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49120362264032974, "count": 1, "min": 0.49120362264032974, "max":
0.49120362264032974}}}
#metrics {"StartTime": 1729800267.6346672, "EndTime": 1729800267.6346805,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4925044404952279, "count": 1, "min": 0.4925044404952279, "max":
0.4925044404952279}}}
#metrics {"StartTime": 1729800267.6347237, "EndTime": 1729800267.634738,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4913612427155611, "count": 1, "min": 0.4913612427155611, "max":
0.4913612427155611}}}
#metrics {"StartTime": 1729800267.63479, "EndTime": 1729800267.634805,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5075574981032862, "count": 1, "min": 0.5075574981032862, "max":
0.5075574981032862}}}
#metrics {"StartTime": 1729800267.6348627, "EndTime": 1729800267.634876,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4913612447679908, "count": 1, "min": 0.4913612447679908, "max":
0.4913612447679908}}}

#metrics {"StartTime": 1729800267.6349115, "EndTime": 1729800267.6349242,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.50663487961471, "count": 1, "min": 0.50663487961471, "max":
0.50663487961471}}}
#metrics {"StartTime": 1729800267.6349738, "EndTime": 1729800267.6349866,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49141838777107477, "count": 1, "min": 0.49141838777107477, "max":
0.49141838777107477}}}
#metrics {"StartTime": 1729800267.6350446, "EndTime": 1729800267.6350594,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49271212437012646, "count": 1, "min": 0.49271212437012646, "max":
0.49271212437012646}}}
#metrics {"StartTime": 1729800267.6351073, "EndTime": 1729800267.6351218,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49144532553852655, "count": 1, "min": 0.49144532553852655, "max":
0.49144532553852655}}}
#metrics {"StartTime": 1729800267.6351595, "EndTime": 1729800267.635173,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4927120454448729, "count": 1, "min": 0.4927120454448729, "max":
0.4927120454448729}}}

#metrics {"StartTime": 1729800267.6352267, "EndTime": 1729800267.6352386,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49161707752558903, "count": 1, "min": 0.49161707752558903, "max":
0.49161707752558903}}}
#metrics {"StartTime": 1729800267.6352704, "EndTime": 1729800267.635283,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5054414814555045, "count": 1, "min": 0.5054414814555045, "max":
0.5054414814555045}}}
#metrics {"StartTime": 1729800267.63533, "EndTime": 1729800267.6353395,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49161707957801876, "count": 1, "min": 0.49161707957801876, "max":
0.49161707957801876}}}
#metrics {"StartTime": 1729800267.6353905, "EndTime": 1729800267.6354063,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5054414553336711, "count": 1, "min": 0.5054414553336711, "max":
0.5054414553336711}}}
#metrics {"StartTime": 1729800267.6354558, "EndTime": 1729800267.6354718,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5076552206281091, "count": 1, "min": 0.5076552206281091, "max":
0.5076552206281091}}}

#metrics {"StartTime": 1729800267.6355143, "EndTime": 1729800267.635529,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5085202940239526, "count": 1, "min": 0.5085202940239526, "max":
0.5085202940239526}}}
#metrics {"StartTime": 1729800267.6355817, "EndTime": 1729800267.635597,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.507700649108452, "count": 1, "min": 0.507700649108452, "max":
0.507700649108452}}}
#metrics {"StartTime": 1729800267.635646, "EndTime": 1729800267.6356618,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5085203393639919, "count": 1, "min": 0.5085203393639919, "max":
0.5085203393639919}}}
#metrics {"StartTime": 1729800267.6357071, "EndTime": 1729800267.6357217,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077501600581752, "count": 1, "min": 0.5077501600581752, "max":
0.5077501600581752}}}
#metrics {"StartTime": 1729800267.6357684, "EndTime": 1729800267.635784,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5089586377920081, "count": 1, "min": 0.5089586377920081, "max":
0.5089586377920081}}}

#metrics {"StartTime": 1729800267.6358318, "EndTime": 1729800267.6358464,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077501544606395, "count": 1, "min": 0.5077501544606395, "max":
0.5077501544606395}}}
#metrics {"StartTime": 1729800267.6358943, "EndTime": 1729800267.635908,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5089586372322545, "count": 1, "min": 0.5089586372322545, "max":
0.5089586372322545}}}
#metrics {"StartTime": 1729800267.6359603, "EndTime": 1729800267.6359746,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137618298245133, "count": 1, "min": 0.5137618298245133, "max":
0.5137618298245133}}}
#metrics {"StartTime": 1729800267.6360297, "EndTime": 1729800267.6360452,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144169155863433, "count": 1, "min": 0.5144169155863433, "max":
0.5144169155863433}}}
#metrics {"StartTime": 1729800267.6360967, "EndTime": 1729800267.636112,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137560311506832, "count": 1, "min": 0.5137560311506832, "max":
0.5137560311506832}}}

#metrics {"StartTime": 1729800267.6361625, "EndTime": 1729800267.6361763,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144298388036294, "count": 1, "min": 0.5144298388036294, "max":
0.5144298388036294}}}
#metrics {"StartTime": 1729800267.6362321, "EndTime": 1729800267.6362476,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137788517440333, "count": 1, "min": 0.5137788517440333, "max":
0.5137788517440333}}}
#metrics {"StartTime": 1729800267.6362944, "EndTime": 1729800267.6363087,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5138853439280584, "count": 1, "min": 0.5138853439280584, "max":
0.5138853439280584}}}
#metrics {"StartTime": 1729800267.6363552, "EndTime": 1729800267.6363642,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137745071233971, "count": 1, "min": 0.5137745071233971, "max":
0.5137745071233971}}}
#metrics {"StartTime": 1729800267.6364014, "EndTime": 1729800267.6364145,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5138853329195715, "count": 1, "min": 0.5138853329195715, "max":
0.5138853329195715}}}
[10/24/2024 20:04:27 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=3, validation binary_classification_cross_entropy_objective
<loss>=0.4911792199968026

```
[10/24/2024 20:04:27 INFO 140356604819264] #early_stopping_criteria_metric:
host=algo-1, epoch=3, criteria=binary_classification_cross_entropy_objective,
value=0.4911792199968026
[10/24/2024 20:04:27 INFO 140356604819264] Saving model for epoch: 3
[10/24/2024 20:04:27 INFO 140356604819264] Saved checkpoint to
"/tmp/tmpm_ibkli1/mx-mod-0000.params"
[10/24/2024 20:04:27 INFO 140356604819264] #progress_metric: host=algo-1,
completed 26.666666666666668 % of epochs
#metrics {"StartTime": 1729800221.8904572, "EndTime": 1729800267.6427898,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 5245888.0, "count": 1, "min": 5245888, "max": 5245888},
"Total Batches Seen": {"sum": 5248.0, "count": 1, "min": 5248, "max": 5248},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 6.0, "count": 1,
"min": 6, "max": 6}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:04:27 INFO 140356604819264] #throughput_metric: host=algo-1,
train throughput=28598.92535863174 records/second
[2024-10-24 20:05:10.186] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 11, "duration": 42540, "num_examples":
1309, "num_bytes": 549558240}
#metrics {"StartTime": 1729800310.1865525, "EndTime": 1729800310.1866555,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49208716111956025, "count": 1, "min": 0.49208716111956025, "max":
0.49208716111956025}}}
```

#metrics {"StartTime": 1729800310.1867476, "EndTime": 1729800310.1867707,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49294695545628164, "count": 1, "min": 0.49294695545628164, "max":
0.49294695545628164}}}
#metrics {"StartTime": 1729800310.186825, "EndTime": 1729800310.186839,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4921312772185066, "count": 1, "min": 0.4921312772185066, "max":
0.4921312772185066}}}
#metrics {"StartTime": 1729800310.1868832, "EndTime": 1729800310.186896,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49294667284204324, "count": 1, "min": 0.49294667284204324, "max":
0.49294667284204324}}}
#metrics {"StartTime": 1729800310.1869357, "EndTime": 1729800310.1869483,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.492190994682662, "count": 1, "min": 0.492190994682662, "max":
0.492190994682662}}}
#metrics {"StartTime": 1729800310.186992, "EndTime": 1729800310.1870077,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5075795787379647, "count": 1, "min": 0.5075795787379647, "max":
0.5075795787379647}}}

#metrics {"StartTime": 1729800310.1870637, "EndTime": 1729800310.18708,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.492190995312612, "count": 1, "min": 0.492190995312612, "max":
0.492190995312612}}}
#metrics {"StartTime": 1729800310.187135, "EndTime": 1729800310.1871505,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507566114886456, "count": 1, "min": 0.507566114886456, "max":
0.507566114886456}}}
#metrics {"StartTime": 1729800310.187195, "EndTime": 1729800310.18721,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4922792485555013, "count": 1, "min": 0.4922792485555013, "max":
0.4922792485555013}}}
#metrics {"StartTime": 1729800310.1872602, "EndTime": 1729800310.187275,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4931064445892241, "count": 1, "min": 0.4931064445892241, "max":
0.4931064445892241}}}
#metrics {"StartTime": 1729800310.187322, "EndTime": 1729800310.1873384,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4923204597449813, "count": 1, "min": 0.4923204597449813, "max":
0.4923204597449813}}}

#metrics {"StartTime": 1729800310.187387, "EndTime": 1729800310.1874013,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4931064410661703, "count": 1, "min": 0.4931064410661703, "max":
0.4931064410661703}}}
#metrics {"StartTime": 1729800310.1874518, "EndTime": 1729800310.1874669,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4923970412974693, "count": 1, "min": 0.4923970412974693, "max":
0.4923970412974693}}}
#metrics {"StartTime": 1729800310.187524, "EndTime": 1729800310.1875403,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5067230989415347, "count": 1, "min": 0.5067230989415347, "max":
0.5067230989415347}}}
#metrics {"StartTime": 1729800310.1875935, "EndTime": 1729800310.1876092,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49239704146078966, "count": 1, "min": 0.49239704146078966, "max":
0.49239704146078966}}}
#metrics {"StartTime": 1729800310.187666, "EndTime": 1729800310.18768,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5067230926653661, "count": 1, "min": 0.5067230926653661, "max":
0.5067230926653661}}}

#metrics {"StartTime": 1729800310.1877298, "EndTime": 1729800310.1877441,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077756460580621, "count": 1, "min": 0.5077756460580621, "max":
0.5077756460580621}}}
#metrics {"StartTime": 1729800310.1877935, "EndTime": 1729800310.1878083,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507977475227566, "count": 1, "min": 0.507977475227566, "max":
0.507977475227566}}}
#metrics {"StartTime": 1729800310.1878562, "EndTime": 1729800310.187871,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507782365898109, "count": 1, "min": 0.507782365898109, "max":
0.507782365898109}}}
#metrics {"StartTime": 1729800310.1879263, "EndTime": 1729800310.1879418,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5079774761608252, "count": 1, "min": 0.5079774761608252, "max":
0.5079774761608252}}}
#metrics {"StartTime": 1729800310.1880014, "EndTime": 1729800310.1880174,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077911946941224, "count": 1, "min": 0.5077911946941224, "max":
0.5077911946941224}}}

#metrics {"StartTime": 1729800310.1880758, "EndTime": 1729800310.1880913,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5103509402027189, "count": 1, "min": 0.5103509402027189, "max":
0.5103509402027189}}}
#metrics {"StartTime": 1729800310.188143, "EndTime": 1729800310.1881592,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5077911953940668, "count": 1, "min": 0.5077911953940668, "max":
0.5077911953940668}}}
#metrics {"StartTime": 1729800310.1882102, "EndTime": 1729800310.1882246,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5103509391528022, "count": 1, "min": 0.5103509391528022, "max":
0.5103509391528022}}}
#metrics {"StartTime": 1729800310.1882887, "EndTime": 1729800310.1883047,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137663243378339, "count": 1, "min": 0.5137663243378339, "max":
0.5137663243378339}}}
#metrics {"StartTime": 1729800310.1883562, "EndTime": 1729800310.1883705,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138561439864132, "count": 1, "min": 0.5138561439864132, "max":
0.5138561439864132}}}

#metrics {"StartTime": 1729800310.18843, "EndTime": 1729800310.188447,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137704289754231, "count": 1, "min": 0.5137704289754231, "max":
0.5137704289754231}}}
#metrics {"StartTime": 1729800310.1884966, "EndTime": 1729800310.1885111,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138560668058716, "count": 1, "min": 0.5138560668058716, "max":
0.5138560668058716}}}
#metrics {"StartTime": 1729800310.1885657, "EndTime": 1729800310.1885817,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137758055054085, "count": 1, "min": 0.5137758055054085, "max":
0.5137758055054085}}}
#metrics {"StartTime": 1729800310.1886322, "EndTime": 1729800310.1886466,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5147246576944987, "count": 1, "min": 0.5147246576944987, "max":
0.5147246576944987}}}
#metrics {"StartTime": 1729800310.1886952, "EndTime": 1729800310.1887097,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137757476899967, "count": 1, "min": 0.5137757476899967, "max":
0.5137757476899967}}}

#metrics {"StartTime": 1729800310.1887565, "EndTime": 1729800310.1887708,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5147246556179969, "count": 1, "min": 0.5147246556179969, "max":
0.5147246556179969}}}
[10/24/2024 20:05:10 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=4, train binary_classification_cross_entropy_objective
<loss>=0.49208716111956025
[2024-10-24 20:05:13.841] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 14, "duration": 3635, "num_examples":
164, "num_bytes": 68694780}
#metrics {"StartTime": 1729800313.849184, "EndTime": 1729800313.8492455,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49117606093423044, "count": 1, "min": 0.49117606093423044, "max":
0.49117606093423044}}}
#metrics {"StartTime": 1729800313.8493345, "EndTime": 1729800313.8493502,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49231248662860666, "count": 1, "min": 0.49231248662860666, "max":
0.49231248662860666}}}
#metrics {"StartTime": 1729800313.8494112, "EndTime": 1729800313.849428,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4912033547049536, "count": 1, "min": 0.4912033547049536, "max":
0.4912033547049536}}}

#metrics {"StartTime": 1729800313.8494816, "EndTime": 1729800313.8494966,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49231242542888287, "count": 1, "min": 0.49231242542888287, "max":
0.49231242542888287}}}
#metrics {"StartTime": 1729800313.8495433, "EndTime": 1729800313.8495579,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49121002864678465, "count": 1, "min": 0.49121002864678465, "max":
0.49121002864678465}}}
#metrics {"StartTime": 1729800313.8496013, "EndTime": 1729800313.8496158,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5057216970786703, "count": 1, "min": 0.5057216970786703, "max":
0.5057216970786703}}}
#metrics {"StartTime": 1729800313.8496711, "EndTime": 1729800313.8496866,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49121002864678465, "count": 1, "min": 0.49121002864678465, "max":
0.49121002864678465}}}
#metrics {"StartTime": 1729800313.849731, "EndTime": 1729800313.849745,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5061617801876563, "count": 1, "min": 0.5061617801876563, "max":
0.5061617801876563}}}

#metrics {"StartTime": 1729800313.8497946, "EndTime": 1729800313.8498101,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4914151710538848, "count": 1, "min": 0.4914151710538848, "max":
0.4914151710538848}}}
#metrics {"StartTime": 1729800313.8498595, "EndTime": 1729800313.8498738,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4925279794389945, "count": 1, "min": 0.4925279794389945, "max":
0.4925279794389945}}}
#metrics {"StartTime": 1729800313.8499174, "EndTime": 1729800313.8499532,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49144546696959557, "count": 1, "min": 0.49144546696959557, "max":
0.49144546696959557}}}
#metrics {"StartTime": 1729800313.8500085, "EndTime": 1729800313.8500235,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4925279585415278, "count": 1, "min": 0.4925279585415278, "max":
0.4925279585415278}}}
#metrics {"StartTime": 1729800313.850069, "EndTime": 1729800313.850084,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49146992409580365, "count": 1, "min": 0.49146992409580365, "max":
0.49146992409580365}}}

#metrics {"StartTime": 1729800313.850133, "EndTime": 1729800313.8501484,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5049549854462281, "count": 1, "min": 0.5049549854462281, "max":
0.5049549854462281}}}
#metrics {"StartTime": 1729800313.8501925, "EndTime": 1729800313.8502073,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4914699291335858, "count": 1, "min": 0.4914699291335858, "max":
0.4914699291335858}}}
#metrics {"StartTime": 1729800313.850251, "EndTime": 1729800313.850265,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5049549791023542, "count": 1, "min": 0.5049549791023542, "max":
0.5049549791023542}}}
#metrics {"StartTime": 1729800313.8503096, "EndTime": 1729800313.850324,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5076392848171021, "count": 1, "min": 0.5076392848171021, "max":
0.5076392848171021}}}
#metrics {"StartTime": 1729800313.8503838, "EndTime": 1729800313.8503993,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5083595130203039, "count": 1, "min": 0.5083595130203039, "max":
0.5083595130203039}}}

#metrics {"StartTime": 1729800313.8504853, "EndTime": 1729800313.8505042,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077007362434246, "count": 1, "min": 0.5077007362434246, "max":
0.5077007362434246}}}
#metrics {"StartTime": 1729800313.8505685, "EndTime": 1729800313.8505847,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5083595150727337, "count": 1, "min": 0.5083595150727337, "max":
0.5083595150727337}}}
#metrics {"StartTime": 1729800313.8506436, "EndTime": 1729800313.8506591,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077155984470818, "count": 1, "min": 0.5077155984470818, "max":
0.5077155984470818}}}
#metrics {"StartTime": 1729800313.8507113, "EndTime": 1729800313.8507266,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5089833604279935, "count": 1, "min": 0.5089833604279935, "max":
0.5089833604279935}}}
#metrics {"StartTime": 1729800313.8507764, "EndTime": 1729800313.8507917,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077155997531735, "count": 1, "min": 0.5077155997531735, "max":
0.5077155997531735}}}

#metrics {"StartTime": 1729800313.850849, "EndTime": 1729800313.8508644,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5089833662121137, "count": 1, "min": 0.5089833662121137, "max":
0.5089833662121137}}}
#metrics {"StartTime": 1729800313.8509219, "EndTime": 1729800313.850937,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137606336311313, "count": 1, "min": 0.5137606336311313, "max":
0.5137606336311313}}}
#metrics {"StartTime": 1729800313.8509953, "EndTime": 1729800313.8510113,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5143034104364074, "count": 1, "min": 0.5143034104364074, "max":
0.5143034104364074}}}
#metrics {"StartTime": 1729800313.8510616, "EndTime": 1729800313.8510766,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137575639425461, "count": 1, "min": 0.5137575639425461, "max":
0.5137575639425461}}}
#metrics {"StartTime": 1729800313.8511336, "EndTime": 1729800313.8511493,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5142991354117985, "count": 1, "min": 0.5142991354117985, "max":
0.5142991354117985}}}

```
#metrics {"StartTime": 1729800313.8511992, "EndTime": 1729800313.851215,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137589467204519, "count": 1, "min": 0.5137589467204519, "max":
0.5137589467204519}}}
#metrics {"StartTime": 1729800313.8512611, "EndTime": 1729800313.851276,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.513710332869119, "count": 1, "min": 0.513710332869119, "max":
0.513710332869119}}}
#metrics {"StartTime": 1729800313.8513272, "EndTime": 1729800313.8513434,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.513759571218853, "count": 1, "min": 0.513759571218853, "max":
0.513759571218853}}}
#metrics {"StartTime": 1729800313.8513982, "EndTime": 1729800313.8514135,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137103494751416, "count": 1, "min": 0.5137103494751416, "max":
0.5137103494751416}}}
[10/24/2024 20:05:13 INFO 140356604819264] #quality_metric: host=algo-1,
epoch=4, validation binary_classification_cross_entropy_objective
<loss>=0.49117606093423044
[10/24/2024 20:05:13 INFO 140356604819264] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objective,
value=0.49117606093423044
[10/24/2024 20:05:13 INFO 140356604819264] Saving model for epoch: 4
[10/24/2024 20:05:13 INFO 140356604819264] Saved checkpoint to
"/tmp/tmpdlgv6kuc/mx-mod-0000.params"
```

```
[10/24/2024 20:05:13 INFO 140356604819264] Early stop condition met.
Stopping training.
[10/24/2024 20:05:13 INFO 140356604819264] #progress_metric: host=algo-1,
completed 100 % epochs
#metrics {"StartTime": 1729800267.645199, "EndTime": 1729800313.8582847,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 6554360.0, "count": 1, "min": 6554360, "max": 6554360},
"Total Batches Seen": {"sum": 6557.0, "count": 1, "min": 6557, "max": 6557},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 7.0, "count": 1,
"min": 7, "max": 7}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:05:13 INFO 140356604819264] #throughput_metric: host=algo-1,
train throughput=28313.798391479842 records/second
[10/24/2024 20:05:13 WARNING 140356604819264] wait_for_all_workers will not
sync workers since the kv store is not running distributed
[10/24/2024 20:05:13 WARNING 140356604819264] wait_for_all_workers will not
sync workers since the kv store is not running distributed
[2024-10-24 20:05:17.342] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 17, "duration": 3465, "num_examples":
164, "num_bytes": 68694780}
[10/24/2024 20:05:17 INFO 140356604819264] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objective,
value=0.49117606093423044
[2024-10-24 20:05:17.959] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 19, "duration": 605, "num_examples":
164, "num_bytes": 68694780}
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('binary_classification_cross_entropy_objective', 0.491597603325673)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('binary_classification_accuracy', 0.7902224885209619)
```

```
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('binary_f_1.000', 0.00671626668210636)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('precision', 0.5523809523809524)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('recall', 0.0033786735793551394)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('roc_auc_score', 0.6463752073008947)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('binary_balanced_accuracy', 0.5)
[10/24/2024 20:05:18 INFO 140356604819264] #validation_score (algo-1) :
('binary_log_loss', 0.753166440982043)
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation binary_classification_cross_entropy_objective
<loss>=0.491597603325673
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation binary_classification_accuracy <score>=0.7902224885209619
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation binary_f_1.000 <score>=0.00671626668210636
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation precision <score>=0.5523809523809524
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation recall <score>=0.0033786735793551394
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation roc_auc_score <score>=0.6463752073008947
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation binary_balanced_accuracy <score>=0.5
[10/24/2024 20:05:18 INFO 140356604819264] #quality_metric: host=algo-1,
validation binary_log_loss <score>=0.753166440982043
[10/24/2024 20:05:18 INFO 140356604819264] Best model found for
hyperparameters: {"optimizer": "adam", "learning_rate": 0.005, "wd": 0.0001,
"l1": 0.0, "lr_scheduler_step": 10, "lr_scheduler_factor": 0.99,
"lr_scheduler_minimum_lr": 0.0001}
[10/24/2024 20:05:18 INFO 140356604819264] Saved checkpoint to
"/tmp/tmpvwpy_01_/mx-mod-0000.params"
```

```
[2024-10-24 20:05:18.217] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 0, "duration": 233069, "num_examples": 1,
"num_bytes": 420000}
[2024-10-24 20:05:18.856] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 1, "duration": 639, "num_examples": 164,
"num_bytes": 68694780}
#metrics {"StartTime": 1729800318.213688, "EndTime": 1729800319.102901,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "Meta": "test_data_iter"}, "Metrics": {"Total Records Seen": {"sum":
163559.0, "count": 1, "min": 163559, "max": 163559}, "Total Batches Seen":
{"sum": 164.0, "count": 1, "min": 164, "max": 164}, "Max Records Seen Between
Resets": {"sum": 163559.0, "count": 1, "min": 163559, "max": 163559}, "Max
Batches Seen Between Resets": {"sum": 164.0, "count": 1, "min": 164, "max":
164}, "Reset Count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}, "Number of
Records Since Last Reset": {"sum": 163559.0, "count": 1, "min": 163559, "max":
163559}, "Number of Batches Since Last Reset": {"sum": 164.0, "count": 1, "min":
164, "max": 164}}}
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('binary_classification_cross_entropy_objective', 0.49218737256277023)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('binary_classification_accuracy', 0.7902469445276628)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('binary_f_1.000', 0.007406764459103665)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('precision', 0.5565217391304348)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('recall', 0.003728191535840154)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('roc_auc_score', 0.6436559297319842)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('binary_balanced_accuracy', 0.5)
[10/24/2024 20:05:19 INFO 140356604819264] #test_score (algo-1) :
('binary_log_loss', 0.7533501373409128)
```

[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test binary_classification_cross_entropy_objective
<loss>=0.49218737256277023
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test binary_classification_accuracy <score>=0.7902469445276628
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test binary_f_1.000 <score>=0.007406764459103665
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test precision <score>=0.5565217391304348
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test recall <score>=0.003728191535840154
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test roc_auc_score <score>=0.6436559297319842
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test binary_balanced_accuracy <score>=0.5
[10/24/2024 20:05:19 INFO 140356604819264] #quality_metric: host=algo-1,
test binary_log_loss <score>=0.7533501373409128
#metrics {"StartTime": 1729800085.1371136, "EndTime": 1729800319.111345,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training"}, "Metrics": {"initialize.time": {"sum": 326.69854164123535, "count":
1, "min": 326.69854164123535, "max": 326.69854164123535}, "epochs": {"sum":
15.0, "count": 1, "min": 15, "max": 15}, "check_early_stopping.time": {"sum":
2.4318695068359375, "count": 6, "min": 0.24461746215820312, "max":
0.9083747863769531}, "update.time": {"sum": 228344.65098381042, "count": 5,
"min": 44897.249937057495, "max": 46210.04295349121}, "finalize.time": {"sum":
4351.868629455566, "count": 1, "min": 4351.868629455566, "max":
4351.868629455566}, "setuptime": {"sum": 2.5920867919921875, "count": 1, "min":
2.5920867919921875, "max": 2.5920867919921875}, "totaltime": {"sum":
234094.975233078, "count": 1, "min": 234094.975233078, "max":
234094.975233078}}}

2024-10-24 20:05:35 Uploading - Uploading generated training model
2024-10-24 20:05:35 Completed - Training job completed
Training seconds: 386
Billable seconds: 386

4.1 Model evaluation

In this section, you will evaluate your trained model.

First, examine the metrics for the training job:

[40]:
```
sagemaker.analytics.TrainingJobAnalytics(classifier_estimator._current_job_name,
                                         metric_names = ['test:objective_loss',
                                                         'test:binary_f_beta',
                                                         'test:precision',
                                                         'test:recall']
                                        ).dataframe()
```

[40]:

	timestamp	metric_name	value
0	0.0	test:objective_loss	0.492187
1	0.0	test:binary_f_beta	0.007407
2	0.0	test:precision	0.556522
3	0.0	test:recall	0.003728

Next, set up some functions that will help load the test data into Amazon S3 and perform a prediction by using the batch prediction function. Using batch prediction will help reduce costs because the instances will only run when predictions are performed on the supplied test data.

Note: Replace `<LabBucketName>` with the name of the lab bucket that was created during the lab setup.

[41]:
```
import io
bucket='c134412a340974518003998t1w11504374297-flightbucket-zyuwb8c4zacf'
prefix='flight-linear'
train_file='flight_train.csv'
test_file='flight_test.csv'
validate_file='flight_validate.csv'
whole_file='flight.csv'
s3_resource = boto3.Session().resource('s3')

def upload_s3_csv(filename, folder, dataframe):
    csv_buffer = io.StringIO()
    dataframe.to_csv(csv_buffer, header=False, index=False )
    s3_resource.Bucket(bucket).Object(os.path.join(prefix, folder, filename)).
 ↪put(Body=csv_buffer.getvalue())
```

```
INFO:botocore.credentials:Found credentials from IAM Role:
BaseNotebookInstanceEc2InstanceRole
```

[42]:
```
def batch_linear_predict(test_data, estimator):
    batch_X = test_data.iloc[:,1:];
    batch_X_file='batch-in.csv'
    upload_s3_csv(batch_X_file, 'batch-in', batch_X)

    batch_output = "s3://{}/{}/batch-out/".format(bucket,prefix)
```

```
    batch_input = "s3://{}/{}/batch-in/{}".format(bucket,prefix,batch_X_file)

    classifier_transformer = estimator.transformer(instance_count=1,
                                                   instance_type='ml.m4.xlarge',
                                                   strategy='MultiRecord',
                                                   assemble_with='Line',
                                                   output_path=batch_output)

    classifier_transformer.transform(data=batch_input,
                                     data_type='S3Prefix',
                                     content_type='text/csv',
                                     split_type='Line')

    classifier_transformer.wait()

    s3 = boto3.client('s3')
    obj = s3.get_object(Bucket=bucket, Key="{}/batch-out/{}".
↪format(prefix,'batch-in.csv.out'))
    target_predicted_df = pd.read_json(io.BytesIO(obj['Body'].
↪read()),orient="records",lines=True)
    return test_data.iloc[:,0], target_predicted_df.iloc[:,0]
```

To run the predictions on the test dataset, run the `batch_linear_predict` function (which was defined previously) on your test dataset.

[43]:
```
test_labels, target_predicted = batch_linear_predict(test, classifier_estimator)
```

```
INFO:sagemaker.image_uris:Same images used for training and inference.
Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating model with name: linear-learner-2024-10-24-20-11-47-764
INFO:sagemaker:Creating transform job with name: linear-
learner-2024-10-24-20-11-48-692

…Docker entrypoint called

with argument(s): serve
Running default environment configuration script
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[10/24/2024 20:19:56 INFO 139918576293696] Memory profiler is not enabled

by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:

"is" with a literal. Did you mean "=="?

  if num_device is 1 and 'dist' not in kvstore:
```

```
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:19:56 INFO 139918576293696] Memory profiler is not enabled
by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:20:00 WARNING 139918576293696] Loggers have already been
setup.
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:20:00 INFO 139918576293696] loading entry points
[10/24/2024 20:20:00 WARNING 139918576293696] Loggers have already been
setup.
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:20:00 INFO 139918576293696] loading entry points
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder
application/json
```

```
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class

algorithm:model
[10/24/2024 20:20:00 INFO 139918576293696] Number of server workers: 4
[10/24/2024 20:20:00 INFO 139918576293696] loading model…
[10/24/2024 20:20:00 INFO 139918576293696] …model loaded.
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class

algorithm:model
[10/24/2024 20:20:00 INFO 139918576293696] Number of server workers: 4
[10/24/2024 20:20:00 INFO 139918576293696] loading model…
[10/24/2024 20:20:00 INFO 139918576293696] …model loaded.
[2024-10-24 20:20:01 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:20:01 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080

(1)
[2024-10-24 20:20:01 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:20:01 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:20:01 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:20:01 +0000] [61] [INFO] Booting worker with pid: 61
```

```
[2024-10-24 20:20:01 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801201.9865332,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
[2024-10-24 20:20:01 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:20:01 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-10-24 20:20:01 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:20:01 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:20:01 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:20:01 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:20:01 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801201.9865332,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.6750495,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
134.18173789978027, "count": 1, "min": 134.18173789978027, "max":
134.18173789978027}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.8153608,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
127.93588638305664, "count": 1, "min": 127.93588638305664, "max":
127.93588638305664}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.9654138,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
136.9473934173584, "count": 1, "min": 136.9473934173584, "max":
136.9473934173584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
```

#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.6750495, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 134.18173789978027, "count": 1, "min": 134.18173789978027, "max": 134.18173789978027}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.8153608, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 127.93588638305664, "count": 1, "min": 127.93588638305664, "max": 127.93588638305664}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.9654138, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.9473934173584, "count": 1, "min": 136.9473934173584, "max": 136.9473934173584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
2024-10-24T20:20:01.996:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
#metrics {"StartTime": 1729801201.9867227, "EndTime": 1729801205.0834043, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.42852210998535, "count": 1, "min": 137.42852210998535, "max": 137.42852210998535}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801204.8155324, "EndTime": 1729801205.722294, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 60.41741371154785, "count": 1, "min": 60.41741371154785, "max": 60.41741371154785}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}

```
#metrics {"StartTime": 1729801201.9867227, "EndTime": 1729801205.0834043,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
137.42852210998535, "count": 1, "min": 137.42852210998535, "max":
137.42852210998535}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801204.8155324, "EndTime": 1729801205.722294,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
60.41741371154785, "count": 1, "min": 60.41741371154785, "max":
60.41741371154785}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}

Docker entrypoint called with argument(s): serve
Running default environment configuration script
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[10/24/2024 20:19:56 INFO 139918576293696] Memory profiler is not enabled
by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:19:56 INFO 139918576293696] Memory profiler is not enabled
by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
```

```
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:20:00 WARNING 139918576293696] Loggers have already been
setup.
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:20:00 INFO 139918576293696] loading entry points
[10/24/2024 20:20:00 WARNING 139918576293696] Loggers have already been
setup.
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:20:00 INFO 139918576293696] loading entry points
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder
application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder
application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder
application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder
text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class
algorithm:model
[10/24/2024 20:20:00 INFO 139918576293696] Number of server workers: 4
[10/24/2024 20:20:00 INFO 139918576293696] loading model…
[10/24/2024 20:20:00 INFO 139918576293696] …model loaded.
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator
application/jsonlines
```

```
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded request iterator

text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/json
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/jsonlines
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

application/x-recordio-protobuf
[10/24/2024 20:20:00 INFO 139918576293696] loaded response encoder

text/csv
[10/24/2024 20:20:00 INFO 139918576293696] loaded entry point class

algorithm:model
[10/24/2024 20:20:00 INFO 139918576293696] Number of server workers: 4
[10/24/2024 20:20:00 INFO 139918576293696] loading model…
[10/24/2024 20:20:00 INFO 139918576293696] …model loaded.
[2024-10-24 20:20:01 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:20:01 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080

(1)
[2024-10-24 20:20:01 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:20:01 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:20:01 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:20:01 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:20:01 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801201.9865332,

"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",

"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,

"count": 1, "min": 1, "max": 1}}}
[2024-10-24 20:20:01 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:20:01 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080

(1)
[2024-10-24 20:20:01 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:20:01 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:20:01 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:20:01 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:20:01 +0000] [70] [INFO] Booting worker with pid: 70
```

#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801201.9865332,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.6750495,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
134.18173789978027, "count": 1, "min": 134.18173789978027, "max":
134.18173789978027}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.8153608,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
127.93588638305664, "count": 1, "min": 127.93588638305664, "max":
127.93588638305664}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.9654138,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
136.9473934173584, "count": 1, "min": 136.9473934173584, "max":
136.9473934173584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.6750495,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
134.18173789978027, "count": 1, "min": 134.18173789978027, "max":
134.18173789978027}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}

#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.8153608, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 127.93588638305664, "count": 1, "min": 127.93588638305664, "max": 127.93588638305664}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801200.999997, "EndTime": 1729801204.9654138, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.9473934173584, "count": 1, "min": 136.9473934173584, "max": 136.9473934173584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
2024-10-24T20:20:01.996:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
#metrics {"StartTime": 1729801201.9867227, "EndTime": 1729801205.0834043, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.42852210998535, "count": 1, "min": 137.42852210998535, "max": 137.42852210998535}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801204.8155324, "EndTime": 1729801205.722294, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 60.41741371154785, "count": 1, "min": 60.41741371154785, "max": 60.41741371154785}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729801201.9867227, "EndTime": 1729801205.0834043, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.42852210998535, "count": 1, "min": 137.42852210998535, "max": 137.42852210998535}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}

```
#metrics {"StartTime": 1729801204.8155324, "EndTime": 1729801205.722294,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
60.41741371154785, "count": 1, "min": 60.41741371154785, "max":
60.41741371154785}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
```

To view a plot of the confusion matrix, and various scoring metrics, create a couple of functions:

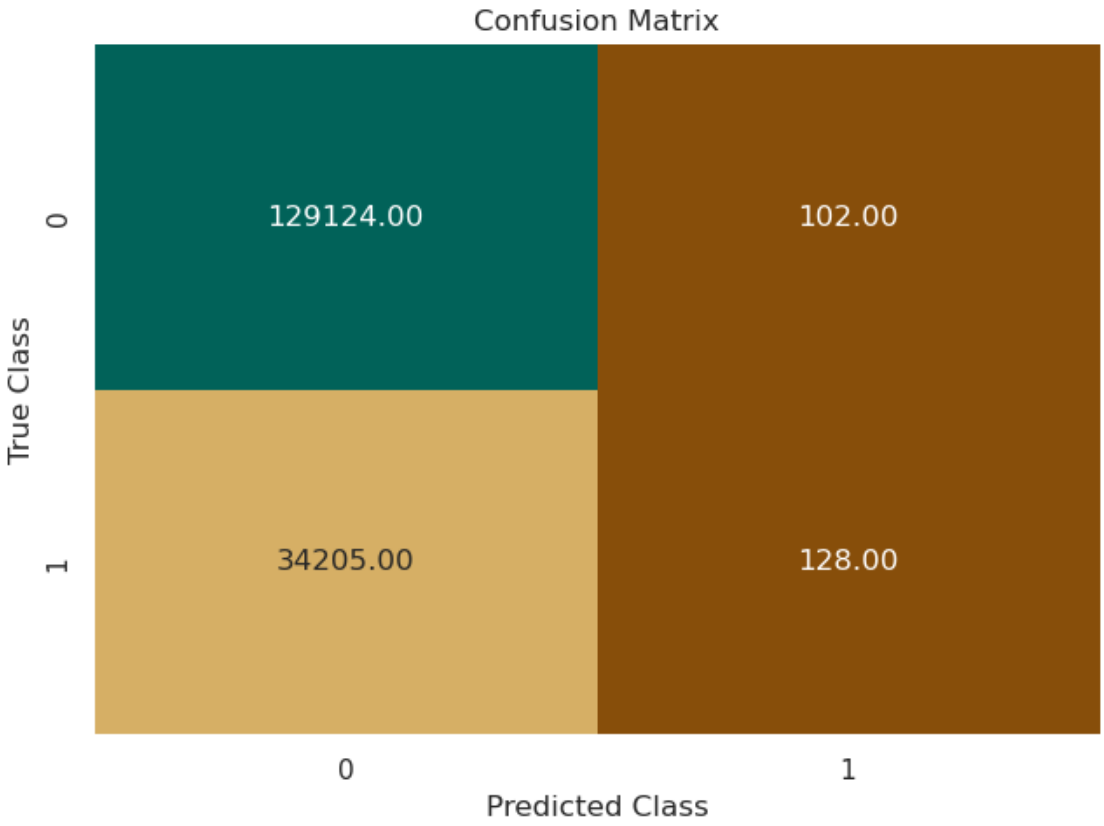
[44]:
```
from sklearn.metrics import confusion_matrix

def plot_confusion_matrix(test_labels, target_predicted):
    matrix = confusion_matrix(test_labels, target_predicted)
    df_confusion = pd.DataFrame(matrix)
    colormap = sns.color_palette("BrBG", 10)
    sns.heatmap(df_confusion, annot=True, fmt='.2f', cbar=None, cmap=colormap)
    plt.title("Confusion Matrix")
    plt.tight_layout()
    plt.ylabel("True Class")
    plt.xlabel("Predicted Class")
    plt.show()
```

[45]:
```
from sklearn import metrics

def plot_roc(test_labels, target_predicted):
    TN, FP, FN, TP = confusion_matrix(test_labels, target_predicted).ravel()
    # Sensitivity, hit rate, recall, or true positive rate
    Sensitivity  = float(TP)/(TP+FN)*100
    # Specificity or true negative rate
    Specificity  = float(TN)/(TN+FP)*100
    # Precision or positive predictive value
    Precision = float(TP)/(TP+FP)*100
    # Negative predictive value
    NPV = float(TN)/(TN+FN)*100
    # Fall out or false positive rate
    FPR = float(FP)/(FP+TN)*100
    # False negative rate
    FNR = float(FN)/(TP+FN)*100
    # False discovery rate
    FDR = float(FP)/(TP+FP)*100
    # Overall accuracy
    ACC = float(TP+TN)/(TP+FP+FN+TN)*100

    print("Sensitivity or TPR: ", Sensitivity, "%")
    print( "Specificity or TNR: ",Specificity, "%")
```

```
    print("Precision: ",Precision, "%")
    print("Negative Predictive Value: ",NPV, "%")
    print( "False Positive Rate: ",FPR,"%")
    print("False Negative Rate: ",FNR, "%")
    print("False Discovery Rate: ",FDR, "%" )
    print("Accuracy: ",ACC, "%")

    test_labels = test.iloc[:,0];
    print("Validation AUC", metrics.roc_auc_score(test_labels,␣
↪target_predicted) )

    fpr, tpr, thresholds = metrics.roc_curve(test_labels, target_predicted)
    roc_auc = metrics.auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % (roc_auc))
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic')
    plt.legend(loc="lower right")

    # create the axis of thresholds (scores)
    ax2 = plt.gca().twinx()
    ax2.plot(fpr, thresholds, markeredgecolor='r',linestyle='dashed', color='r')
    ax2.set_ylabel('Threshold',color='r')
    ax2.set_ylim([thresholds[-1],thresholds[0]])
    ax2.set_xlim([fpr[0],fpr[-1]])

    print(plt.figure())
```

To plot the confusion matrix, call the `plot_confusion_matrix` function on the `test_labels` and the `target_predicted` data from your batch job:

[46]:
```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



4.1.1 Key questions to consider:

1. How does your model's performance on the test set compare to its performance on the training set? What can you deduce from this comparison?
2. Are there obvious differences between the outcomes of metrics like accuracy, precision, and recall? If so, why might you be seeing those differences?
3. Given your business situation and goals, which metric (or metrics) is the most important for you to consider? Why?
4. From a business standpoint, is the outcome for the metric (or metrics) that you consider to be the most important sufficient for what you need? If not, what are some things you might change in your next iteration? (This will happen in the feature engineering section, which is next.)

Use the following cells to answer these (and other) questions. Insert and delete cells where needed.

Project presentation: In your project presentation, write down your answers to these questions – and other similar questions that you might answer – in this section. Record the key details and decisions that you made. **Question**: What can you summarize from the confusion matrix?

Enter your answer here - Model Performance Summary - Confusion Matrix Results - True Negatives (TN): 129124 - False Negatives (FN): 34205 - False Positives (FP): 102 - True Positives

(TP): 128

4.2 End of Step 3

Save the project file to your local computer. Follow these steps:

1. In the file explorer on the left, right-click the notebook that you're working on.
2. Select **Download**, and save the file locally.

This action downloads the current notebook to the default download folder on your computer.

5 Iteration II

6 Step 4: Feature engineering

You have now gone through one iteration of training and evaluating your model. Given that the first outcome that you reached for your model probably wasn't sufficient for solving your business problem, what could you change about your data to possibly improve model performance?

6.0.1 Key questions to consider:

1. How might the balance of your two main classes (*delay* and *no delay*) impact model performance?
2. Do you have any features that are correlated?
3. At this stage, could you perform any feature-reduction techniques that might have a positive impact on model performance?
4. Can you think of adding some more data or datasets?
5. After performing some feature engineering, how does the performance of your model compare to the first iteration?

Use the following cells to perform specific feature-engineering techniques that you think could improve your model performance (use the previous questions as a guide). Insert and delete cells where needed.

Project presentation: In your project presentation, record your key decisions and the methods that you use in this section. Also include any new performance metrics that you obtain after you evaluate your model again. Before you start, think about why the precision and recall are around 80 percent, and the accuracy is at 99 percent.

Add more features:

1. Holidays
2. Weather

Because the list of holidays from 2014 to 2018 is known, you can create an indicator variable **is_holiday** to mark them.

The hypothesis is that airplane delays could be higher during holidays compared to the rest of the days. Add a boolean variable `is_holiday` that includes the holidays for the years 2014-2018.

[47]:
```
# Source: http://www.calendarpedia.com/holidays/federal-holidays-2014.html

holidays_14 = ['2014-01-01',  '2014-01-20', '2014-02-17', '2014-05-26',␣
 ↪'2014-07-04', '2014-09-01', '2014-10-13', '2014-11-11', '2014-11-27',␣
 ↪'2014-12-25' ]
holidays_15 = ['2015-01-01',  '2015-01-19', '2015-02-16', '2015-05-25',␣
 ↪'2015-06-03', '2015-07-04', '2015-09-07', '2015-10-12', '2015-11-11',␣
 ↪'2015-11-26', '2015-12-25']
holidays_16 = ['2016-01-01',  '2016-01-18', '2016-02-15', '2016-05-30',␣
 ↪'2016-07-04', '2016-09-05', '2016-10-10', '2016-11-11', '2016-11-24',␣
 ↪'2016-12-25', '2016-12-26']
holidays_17 = ['2017-01-02', '2017-01-16', '2017-02-20', '2017-05-29' ,␣
 ↪'2017-07-04', '2017-09-04' ,'2017-10-09', '2017-11-10', '2017-11-23',␣
 ↪'2017-12-25']
holidays_18 = ['2018-01-01', '2018-01-15', '2018-02-19', '2018-05-28' ,␣
 ↪'2018-07-04', '2018-09-03' ,'2018-10-08', '2018-11-12','2018-11-22',␣
 ↪'2018-12-25']
holidays = holidays_14+ holidays_15+ holidays_16 + holidays_17+ holidays_18

# Add indicator variable for holidays
# Enter your code here
data_orig['is_holiday'] = data_orig['FlightDate'].isin(holidays).astype(int)
```

Weather data was fetched from https://www.ncei.noaa.gov/access/services/data/v1?dataset=daily-summaries&stations=USW00023174,USW00012960,USW00003017,USW00094846,USW00013874,USW00023234,U
01-01&endDate=2018-12-31.

This dataset has information on wind speed, precipitation, snow, and temperature for cities by their airport codes.

Question: Could bad weather because of rain, heavy winds, or snow lead to airplane delays? You will now check.

[48]:
```
!aws s3 cp s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_project/
 ↪data2/daily-summaries.csv /home/ec2-user/SageMaker/project/data/
#!wget 'https://www.ncei.noaa.gov/access/services/data/v1?
 ↪dataset=daily-summaries&stations=USW00023174,USW00012960,USW00003017,USW00094846,USW0001387
 ↪-O /home/ec2-user/SageMaker/project/data/daily-summaries.csv
```

```
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMLFO-1/flight_delay_project/data2/daily-summaries.csv to
../project/data/daily-summaries.csv
```

Import the weather data that was prepared for the airport codes in the dataset. Use the following stations and airports for the analysis. Create a new column called *airport* that maps the weather station to the airport name.

[49]:
```
weather = pd.read_csv('/home/ec2-user/SageMaker/project/data/daily-summaries.
 ↪csv')
```

```
station =␣
 ↪['USW00023174','USW00012960','USW00003017','USW00094846','USW00013874','USW00023234','USW00
airports = ['LAX', 'IAH', 'DEN', 'ORD', 'ATL', 'SFO', 'DFW', 'PHX', 'CLT']

### Map weather stations to airport code
station_map = {s:a for s,a in zip(station, airports)}
weather['airport'] = weather['STATION'].map(station_map)
```

From the **DATE** column, create another column called *MONTH*.

[50]:
```
weather['MONTH'] = weather['DATE'].apply(lambda x: x.split('-')[1])
weather.head()
```

[50]:
```
       STATION        DATE  AWND  PRCP  SNOW  SNWD   TAVG   TMAX   TMIN  \
0  USW00023174  2014-01-01    16     0   NaN   NaN  131.0  178.0   78.0
1  USW00023174  2014-01-02    22     0   NaN   NaN  159.0  256.0  100.0
2  USW00023174  2014-01-03    17     0   NaN   NaN  140.0  178.0   83.0
3  USW00023174  2014-01-04    18     0   NaN   NaN  136.0  183.0  100.0
4  USW00023174  2014-01-05    18     0   NaN   NaN  151.0  244.0   83.0

  airport MONTH
0     LAX    01
1     LAX    01
2     LAX    01
3     LAX    01
4     LAX    01
```

6.0.2 Sample output

```
  STATION     DATE       AWND PRCP SNOW SNWD TAVG TMAX  TMIN airport MONTH
0 USW00023174 2014-01-01 16   0    NaN  NaN 131.0 178.0 78.0  LAX    01
1 USW00023174 2014-01-02 22   0    NaN  NaN 159.0 256.0 100.0 LAX    01
2 USW00023174 2014-01-03 17   0    NaN  NaN 140.0 178.0 83.0  LAX    01
3 USW00023174 2014-01-04 18   0    NaN  NaN 136.0 183.0 100.0 LAX    01
4 USW00023174 2014-01-05 18   0    NaN  NaN 151.0 244.0 83.0  LAX    01
```

Analyze and handle the **SNOW** and **SNWD** columns for missing values by using `fillna()`. To check the missing values for all the columns, use the `isna()` function.

[51]:
```
weather.SNOW.fillna(0, inplace=True)
weather.SNWD.fillna(0, inplace=True)
weather.isna().sum()
```

[51]:
```
STATION    0
DATE       0
AWND       0
PRCP       0
SNOW       0
SNWD       0
```

```
TAVG       62
TMAX       20
TMIN       20
airport     0
MONTH       0
dtype: int64
```

Question: Print the index of the rows that have missing values for *TAVG*, *TMAX*, *TMIN*.

Hint: To find the rows that are missing, use the `isna()` function. Then, to get the index, use the list on the *idx* variable.

[52]:
```
idx = np.array([i for i in range(len(weather))])
TAVG_idx = idx[weather.TAVG.isna()]
TMAX_idx = idx[weather.TMAX.isna()] # Enter your code here
TMIN_idx = idx[weather.TMIN.isna()] # Enter your code here
TAVG_idx
```

[52]:
```
array([ 3956,  3957,  3958,  3959,  3960,  3961,  3962,  3963,  3964,
        3965,  3966,  3967,  3968,  3969,  3970,  3971,  3972,  3973,
        3974,  3975,  3976,  3977,  3978,  3979,  3980,  3981,  3982,
        3983,  3984,  3985,  4017,  4018,  4019,  4020,  4021,  4022,
        4023,  4024,  4025,  4026,  4027,  4028,  4029,  4030,  4031,
        4032,  4033,  4034,  4035,  4036,  4037,  4038,  4039,  4040,
        4041,  4042,  4043,  4044,  4045,  4046,  4047, 13420])
```

6.0.3 Sample output

```
array([ 3956,  3957,  3958,  3959,  3960,  3961,  3962,  3963,  3964,
        3965,  3966,  3967,  3968,  3969,  3970,  3971,  3972,  3973,
        3974,  3975,  3976,  3977,  3978,  3979,  3980,  3981,  3982,
        3983,  3984,  3985,  4017,  4018,  4019,  4020,  4021,  4022,
        4023,  4024,  4025,  4026,  4027,  4028,  4029,  4030,  4031,
        4032,  4033,  4034,  4035,  4036,  4037,  4038,  4039,  4040,
        4041,  4042,  4043,  4044,  4045,  4046,  4047, 13420])
```

You can replace the missing *TAVG*, *TMAX*, and *TMIN* values with the average value for a particular station or airport. Because consecutive rows of *TAVG_idx* are missing, replacing them with a previous value would not be possible. Instead, replace them with the mean. Use the `groupby` function to aggregate the variables with a mean value.

Hint: Group by `MONTH` and `STATION`.

[53]:
```
# Enter your code here
weather_impute = weather.groupby(['MONTH', 'STATION']).agg({'TAVG':
  ↪'mean','TMAX':'mean', 'TMIN':'mean' }).reset_index()
weather_impute.head(2)
```

[53]:

	MONTH	STATION	TAVG	TMAX	TMIN
0	01	USW00003017	-2.741935	74.000000	-69.858065
1	01	USW00003927	79.529032	143.767742	20.696774

Merge the mean data with the weather data.

[54]:
```
weather = pd.merge(weather, weather_impute,  how='left',␣
 ↪left_on=['MONTH','STATION'], right_on = ['MONTH','STATION'])\
.rename(columns = {'TAVG_y':'TAVG_AVG',
                   'TMAX_y':'TMAX_AVG',
                   'TMIN_y':'TMIN_AVG',
                   'TAVG_x':'TAVG',
                   'TMAX_x':'TMAX',
                   'TMIN_x':'TMIN'})
```

Check for missing values again.

[55]:
```
weather.TAVG[TAVG_idx] = weather.TAVG_AVG[TAVG_idx]
weather.TMAX[TMAX_idx] = weather.TMAX_AVG[TMAX_idx]
weather.TMIN[TMIN_idx] = weather.TMIN_AVG[TMIN_idx]
weather.isna().sum()
```

[55]:
```
STATION     0
DATE        0
AWND        0
PRCP        0
SNOW        0
SNWD        0
TAVG        0
TMAX        0
TMIN        0
airport     0
MONTH       0
TAVG_AVG    0
TMAX_AVG    0
TMIN_AVG    0
dtype: int64
```

Drop `STATION,MONTH,TAVG_AVG,TMAX_AVG,TMIN_AVG,TMAX,TMIN,SNWD` from the dataset.

[56]:
```
weather.drop(columns=['STATION','MONTH','TAVG_AVG', 'TMAX_AVG', 'TMIN_AVG',␣
 ↪'TMAX' ,'TMIN', 'SNWD'],inplace=True)
```

Add the origin and destination weather conditions to the dataset.

[57]:
```
# Add origin weather conditions
data_orig = pd.merge(data_orig, weather,  how='left',␣
 ↪left_on=['FlightDate','Origin'], right_on = ['DATE','airport'])\
```

```
.rename(columns = {'AWND':'AWND_O','PRCP':'PRCP_O', 'TAVG':'TAVG_O', 'SNOW':␣
 ↪'SNOW_O'})\
.drop(columns=['DATE','airport'])

# Add destination weather conditions
data_orig = pd.merge(data_orig, weather,  how='left',␣
 ↪left_on=['FlightDate','Dest'], right_on = ['DATE','airport'])\
.rename(columns = {'AWND':'AWND_D','PRCP':'PRCP_D', 'TAVG':'TAVG_D', 'SNOW':␣
 ↪'SNOW_D'})\
.drop(columns=['DATE','airport'])
```

Note: It's always a good practice to check for nulls or NAs after joins.

[58]:
```
sum(data.isna().any())
```

[58]:
```
0
```

[59]:
```
data_orig.columns
```

[59]:
```
Index(['Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek', 'FlightDate',
       'Reporting_Airline', 'Origin', 'OriginState', 'Dest', 'DestState',
       'CRSDepTime', 'Cancelled', 'Diverted', 'Distance', 'DistanceGroup',
       'ArrDelay', 'ArrDelayMinutes', 'is_delay', 'AirTime', 'DepHourofDay',
       'is_holiday', 'AWND_O', 'PRCP_O', 'SNOW_O', 'TAVG_O', 'AWND_D',
       'PRCP_D', 'SNOW_D', 'TAVG_D'],
      dtype='object')
```

Convert the categorical data into numerical data by using one-hot encoding.

[60]:
```
data = data_orig.copy()
data = data[['is_delay', 'Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek',
       'Reporting_Airline', 'Origin',␣
 ↪'Dest','Distance','DepHourofDay','is_holiday', 'AWND_O', 'PRCP_O',
       'TAVG_O', 'AWND_D', 'PRCP_D', 'TAVG_D', 'SNOW_O', 'SNOW_D']]


categorical_columns  = ['Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek',
       'Reporting_Airline', 'Origin', 'Dest', 'is_holiday']
for c in categorical_columns:
    data[c] = data[c].astype('category')
```

[61]:
```
data_dummies = pd.get_dummies(data[['Year', 'Quarter', 'Month', 'DayofMonth',␣
 ↪'DayOfWeek', 'Reporting_Airline', 'Origin', 'Dest', 'is_holiday']],␣
 ↪drop_first=True)
data_dummies = data_dummies.replace({True: 1, False: 0})
data = pd.concat([data, data_dummies], axis = 1)
data.drop(categorical_columns,axis=1, inplace=True)
```

Check the new columns.

[62]:
```
data.shape
```

[62]:
```
(1635590, 86)
```

[63]:
```
data.columns
```

[63]:
```
Index(['is_delay', 'Distance', 'DepHourofDay', 'AWND_O', 'PRCP_O', 'TAVG_O',
       'AWND_D', 'PRCP_D', 'TAVG_D', 'SNOW_O', 'SNOW_D', 'Year_2015',
       'Year_2016', 'Year_2017', 'Year_2018', 'Quarter_2', 'Quarter_3',
       'Quarter_4', 'Month_2', 'Month_3', 'Month_4', 'Month_5', 'Month_6',
       'Month_7', 'Month_8', 'Month_9', 'Month_10', 'Month_11', 'Month_12',
       'DayofMonth_2', 'DayofMonth_3', 'DayofMonth_4', 'DayofMonth_5',
       'DayofMonth_6', 'DayofMonth_7', 'DayofMonth_8', 'DayofMonth_9',
       'DayofMonth_10', 'DayofMonth_11', 'DayofMonth_12', 'DayofMonth_13',
       'DayofMonth_14', 'DayofMonth_15', 'DayofMonth_16', 'DayofMonth_17',
       'DayofMonth_18', 'DayofMonth_19', 'DayofMonth_20', 'DayofMonth_21',
       'DayofMonth_22', 'DayofMonth_23', 'DayofMonth_24', 'DayofMonth_25',
       'DayofMonth_26', 'DayofMonth_27', 'DayofMonth_28', 'DayofMonth_29',
       'DayofMonth_30', 'DayofMonth_31', 'DayOfWeek_2', 'DayOfWeek_3',
       'DayOfWeek_4', 'DayOfWeek_5', 'DayOfWeek_6', 'DayOfWeek_7',
       'Reporting_Airline_DL', 'Reporting_Airline_OO', 'Reporting_Airline_UA',
       'Reporting_Airline_WN', 'Origin_CLT', 'Origin_DEN', 'Origin_DFW',
       'Origin_IAH', 'Origin_LAX', 'Origin_ORD', 'Origin_PHX', 'Origin_SFO',
       'Dest_CLT', 'Dest_DEN', 'Dest_DFW', 'Dest_IAH', 'Dest_LAX', 'Dest_ORD',
       'Dest_PHX', 'Dest_SFO', 'is_holiday_1'],
      dtype='object')
```

6.0.4 Sample output

```
Index(['Distance', 'DepHourofDay', 'is_delay', 'AWND_O', 'PRCP_O', 'TAVG_O',
       'AWND_D', 'PRCP_D', 'TAVG_D', 'SNOW_O', 'SNOW_D', 'Year_2015',
       'Year_2016', 'Year_2017', 'Year_2018', 'Quarter_2', 'Quarter_3',
       'Quarter_4', 'Month_2', 'Month_3', 'Month_4', 'Month_5', 'Month_6',
       'Month_7', 'Month_8', 'Month_9', 'Month_10', 'Month_11', 'Month_12',
       'DayofMonth_2', 'DayofMonth_3', 'DayofMonth_4', 'DayofMonth_5',
       'DayofMonth_6', 'DayofMonth_7', 'DayofMonth_8', 'DayofMonth_9',
       'DayofMonth_10', 'DayofMonth_11', 'DayofMonth_12', 'DayofMonth_13',
       'DayofMonth_14', 'DayofMonth_15', 'DayofMonth_16', 'DayofMonth_17',
       'DayofMonth_18', 'DayofMonth_19', 'DayofMonth_20', 'DayofMonth_21',
       'DayofMonth_22', 'DayofMonth_23', 'DayofMonth_24', 'DayofMonth_25',
       'DayofMonth_26', 'DayofMonth_27', 'DayofMonth_28', 'DayofMonth_29',
       'DayofMonth_30', 'DayofMonth_31', 'DayOfWeek_2', 'DayOfWeek_3',
       'DayOfWeek_4', 'DayOfWeek_5', 'DayOfWeek_6', 'DayOfWeek_7',
       'Reporting_Airline_DL', 'Reporting_Airline_OO', 'Reporting_Airline_UA',
       'Reporting_Airline_WN', 'Origin_CLT', 'Origin_DEN', 'Origin_DFW',
       'Origin_IAH', 'Origin_LAX', 'Origin_ORD', 'Origin_PHX', 'Origin_SFO',
       'Dest_CLT', 'Dest_DEN', 'Dest_DFW', 'Dest_IAH', 'Dest_LAX', 'Dest_ORD',
       'Dest_PHX', 'Dest_SFO', 'is_holiday_1'],
```

```
      dtype='object')
```

Rename the **is_delay** column to *target* again. Use the same code that you used previously.

[64]:
```
# Enter your code here
data.rename(columns={'is_delay': 'target'}, inplace=True)
```

Create the training sets again.

Hint: Use the `split_data` function that you defined (and used) earlier.

[65]:
```
# Enter your code here
train, validate, test = split_data(data)
print(train['target'].value_counts())
print(validate['target'].value_counts())
print(test['target'].value_counts())
```

```
0.0    1033806
1.0     274666
Name: target, dtype: int64
0.0    129226
1.0     34333
Name: target, dtype: int64
0.0    129226
1.0     34333
Name: target, dtype: int64
```

6.0.5 New baseline classifier

Now, see if these new features add any predictive power to the model.

[66]:
```
# Instantiate the LinearLearner estimator object
# Enter your code here
classifier_estimator2 = sagemaker.LinearLearner(role=sagemaker.
 ↪get_execution_role(),
                                                 instance_count=1,
                                                 instance_type='ml.m4.xlarge',
                                                ␣
 ↪predictor_type='binary_classifier',
                                                ␣
 ↪binary_classifier_model_selection_criteria='cross_entropy_loss')
```

6.0.6 Sample code

```
num_classes = len(pd.unique(train_labels))
classifier_estimator2 = sagemaker.LinearLearner(role=sagemaker.get_execution_role(),
                                                instance_count=1,
                                                instance_type='ml.m4.xlarge',
                                                predictor_type='binary_classifier',
                                                binary_classifier_model_selection_criteria = 'c
```

[67]:
```
train_records = classifier_estimator2.record_set(train.values[:, 1:].astype(np.
↪float32), train.values[:, 0].astype(np.float32), channel='train')
val_records = classifier_estimator2.record_set(validate.values[:, 1:].astype(np.
↪float32), validate.values[:, 0].astype(np.float32), channel='validation')
test_records = classifier_estimator2.record_set(test.values[:, 1:].astype(np.
↪float32), test.values[:, 0].astype(np.float32), channel='test')
```

Train your model by using the three datasets that you just created.

[68]:
```
# Enter your code here
classifier_estimator2.fit([train_records, val_records, test_records])
```

```
INFO:sagemaker.image_uris:Same images used for training and inference.
Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker.image_uris:Same images used for training and inference.
Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating training-job with name: linear-
learner-2024-10-24-20-36-00-223

2024-10-24 20:36:02 Starting - Starting the training job…
2024-10-24 20:36:17 Starting - Preparing the instances for training…
2024-10-24 20:36:43 Downloading - Downloading input data…
2024-10-24 20:37:33 Downloading - Downloading the training image…
2024-10-24 20:38:34 Training - Training image download completed. Training in
progress…Docker entrypoint called with argument(s): train
Running default environment configuration script
```

```
[10/24/2024 20:38:53 INFO 139943714142016] Reading default configuration
from /opt/amazon/lib/python3.8/site-packages/algorithm/resources/default-
input.json: {'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': 'auto',
'use_bias': 'true', 'binary_classifier_model_selection_criteria': 'accuracy',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false'}
[10/24/2024 20:38:53 INFO 139943714142016] Merging with provided
configuration from /opt/ml/input/config/hyperparameters.json:
{'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'feature_dim': '85', 'mini_batch_size': '1000', 'predictor_type':
'binary_classifier'}
```

```
[10/24/2024 20:38:53 INFO 139943714142016] Final configuration:
{'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': '85', 'use_bias':
'true', 'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false',
'predictor_type': 'binary_classifier'}
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:38:56 WARNING 139943714142016] Loggers have already been
setup.
```

```
[10/24/2024 20:38:56 INFO 139943714142016] Final configuration:
{'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': '85', 'use_bias':
'true', 'binary_classifier_model_selection_criteria': 'cross_entropy_loss',
'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8',
'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method':
'uniform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0',
'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5',
'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1',
'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto',
'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult':
'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step':
'auto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto',
'positive_example_weight_mult': '1.0', 'balance_multiclass_weights': 'false',
'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto',
'unbias_label': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto',
'_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3',
'early_stopping_tolerance': '0.001', '_enable_profiler': 'false',
'predictor_type': 'binary_classifier'}
[10/24/2024 20:38:56 WARNING 139943714142016] Loggers have already been
setup.
Process 7 is a worker.
[10/24/2024 20:38:56 INFO 139943714142016] Using default worker.
[10/24/2024 20:38:56 INFO 139943714142016] Checkpoint loading and saving
are disabled.
[2024-10-24 20:38:56.554] [tensorio] [warning] TensorIO is already
initialized; ignoring the initialization routine.
[2024-10-24 20:38:56.558] [tensorio] [warning] TensorIO is already
initialized; ignoring the initialization routine.
[2024-10-24 20:38:56.613] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 0, "duration": 62, "num_examples": 1,
"num_bytes": 388000}
[10/24/2024 20:38:56 INFO 139943714142016] Create Store: local
```

```
[2024-10-24 20:38:56.746] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 1, "duration": 132, "num_examples": 11,
"num_bytes": 4268000}
[10/24/2024 20:38:56 INFO 139943714142016] Scaler algorithm parameters
 <algorithm.scaler.ScalerAlgorithmStable object at 0x7f469ea12220>
[10/24/2024 20:38:56 INFO 139943714142016] Scaling model computed with
parameters:
 {'stdev_label': None, 'stdev_weight':
[5.3676862e+02 4.9751568e+00 1.6331228e+01 8.2915504e+01 9.0630386e+01
 1.6398325e+01 7.7946526e+01 8.9692650e+01 7.1496153e+00 8.0103464e+00
 3.9394417e-01 4.0942764e-01 4.1387710e-01 4.0390077e-01 4.3185183e-01
 4.4104236e-01 4.3405735e-01 2.5499377e-01 2.7697837e-01 2.6860031e-01
 2.7711493e-01 2.8022981e-01 2.8747982e-01 2.8500608e-01 2.7793241e-01
 2.8434965e-01 2.7185535e-01 2.7546957e-01 1.8719329e-01 1.7257783e-01
 1.7911108e-01 1.7887399e-01 1.8285699e-01 1.7887397e-01 1.7744364e-01
 1.8239361e-01 1.7911111e-01 1.7768298e-01 1.8308821e-01 1.6907685e-01
 1.8029131e-01 1.8122916e-01 1.7863652e-01 1.7624156e-01 1.8076093e-01
 1.8331905e-01 1.8377975e-01 1.7429848e-01 1.7599998e-01 1.7768294e-01
 1.7839867e-01 1.6678227e-01 1.8076095e-01 1.8262547e-01 1.7958426e-01
 1.7033654e-01 1.6549201e-01 1.3491301e-01 3.5182565e-01 3.4942272e-01
 3.5455844e-01 3.6190468e-01 3.2739079e-01 3.4960875e-01 3.7354308e-01
 2.3815936e-01 4.5729917e-01 3.1300303e-01 2.4668624e-01 3.2000676e-01
 3.2801935e-01 2.6774195e-01 3.5902092e-01 3.3630344e-01 3.0024225e-01
 3.2410914e-01 2.4684641e-01 3.1232232e-01 3.2978889e-01 2.8022978e-01
 3.5519007e-01 3.3789679e-01 3.0084661e-01 3.2357445e-01 1.5473518e-01]
<NDArray 85 @cpu(0)>, 'mean_label': None, 'mean_weight':
```

```
[1.01142212e+03 1.28570013e+01 3.85490913e+01 2.14453659e+01
 1.72294220e+02 3.85935516e+01 2.06123657e+01 1.72392700e+02
 6.31818175e-01 6.93545461e-01 1.92090929e-01 2.13000000e-01
 2.19454557e-01 2.05272734e-01 2.48000026e-01 2.64454544e-01
 2.51818180e-01 6.99091032e-02 8.37272853e-02 7.82727376e-02
 8.38181973e-02 8.59091058e-02 9.09090936e-02 8.91818330e-02
 8.43636468e-02 8.87272656e-02 8.03636387e-02 8.27272832e-02
 3.63636389e-02 3.07272747e-02 3.31818201e-02 3.30909118e-02
 3.46363671e-02 3.30909118e-02 3.25454548e-02 3.44545469e-02
 3.31818238e-02 3.26363668e-02 3.47272791e-02 2.94545498e-02
 3.36363688e-02 3.39999981e-02 3.30000035e-02 3.20909098e-02
 3.38181816e-02 3.48181799e-02 3.50000001e-02 3.13636400e-02
 3.20000015e-02 3.26363668e-02 3.29090916e-02 2.86363643e-02
 3.38181816e-02 3.45454589e-02 3.33636366e-02 2.99090929e-02
 2.81818211e-02 1.85454562e-02 1.44727305e-01 1.42363638e-01
 1.47454545e-01 1.55000016e-01 1.22090913e-01 1.42545491e-01
 1.67636365e-01 6.03636391e-02 2.97818184e-01 1.10090919e-01
 6.50909096e-02 1.15818188e-01 1.22636378e-01 7.77272731e-02
 1.52000010e-01 1.30000010e-01 1.00181833e-01 1.19272724e-01
 6.51818290e-02 1.09545454e-01 1.24181822e-01 8.59091058e-02
 1.48090929e-01 1.31454557e-01 1.00636370e-01 1.18818179e-01
 2.45454554e-02]
<NDArray 85 @cpu(0)>}
/opt/amazon/python3.8/lib/python3.8/subprocess.py:848: RuntimeWarning: line
buffering (buffering=1) isn't supported in binary mode, the default buffer size
will be used
  self.stdout = io.open(c2pread, 'rb', bufsize)
[10/24/2024 20:38:56 INFO 139943714142016] nvidia-smi: took 0.041 seconds
to run.
[10/24/2024 20:38:56 INFO 139943714142016] nvidia-smi identified 0
GPUs.
[10/24/2024 20:38:56 INFO 139943714142016] Number of GPUs being used: 0
```

#metrics {"StartTime": 1729802336.8610756, "EndTime": 1729802336.8611166, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "Meta": "init_train_data_iter"}, "Metrics": {"Total Records Seen": {"sum": 12000.0, "count": 1, "min": 12000, "max": 12000}, "Total Batches Seen": {"sum": 12.0, "count": 1, "min": 12, "max": 12}, "Max Records Seen Between Resets": {"sum": 11000.0, "count": 1, "min": 11000, "max": 11000}, "Max Batches Seen Between Resets": {"sum": 11.0, "count": 1, "min": 11, "max": 11}, "Reset Count": {"sum": 2.0, "count": 1, "min": 2, "max": 2}, "Number of Records Since Last Reset": {"sum": 0.0, "count": 1, "min": 0, "max": 0}, "Number of Batches Since Last Reset": {"sum": 0.0, "count": 1, "min": 0, "max": 0}}}
[2024-10-24 20:39:28.834] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 3, "duration": 31972, "num_examples": 1309, "num_bytes": 507687136}
#metrics {"StartTime": 1729802368.8341124, "EndTime": 1729802368.834191, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 0}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4830578342169432, "count": 1, "min": 0.4830578342169432, "max": 0.4830578342169432}}}
#metrics {"StartTime": 1729802368.8342733, "EndTime": 1729802368.834287, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 1}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4836235268502425, "count": 1, "min": 0.4836235268502425, "max": 0.4836235268502425}}}
#metrics {"StartTime": 1729802368.834322, "EndTime": 1729802368.8343308, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 2}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4831616775792673, "count": 1, "min": 0.4831616775792673, "max": 0.4831616775792673}}}

#metrics {"StartTime": 1729802368.8343596, "EndTime": 1729802368.8343678,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4836716299727787, "count": 1, "min": 0.4836716299727787, "max":
0.4836716299727787}}}
#metrics {"StartTime": 1729802368.834395, "EndTime": 1729802368.8344033,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49523429968597693, "count": 1, "min": 0.49523429968597693, "max":
0.49523429968597693}}}
#metrics {"StartTime": 1729802368.8344333, "EndTime": 1729802368.834443,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5049271954877661, "count": 1, "min": 0.5049271954877661, "max":
0.5049271954877661}}}
#metrics {"StartTime": 1729802368.8344736, "EndTime": 1729802368.8344839,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4952537602649187, "count": 1, "min": 0.4952537602649187, "max":
0.4952537602649187}}}
#metrics {"StartTime": 1729802368.8345122, "EndTime": 1729802368.834522,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.504929616595627, "count": 1, "min": 0.504929616595627, "max":
0.504929616595627}}}

#metrics {"StartTime": 1729802368.834554, "EndTime": 1729802368.8345637,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48334211159335727, "count": 1, "min": 0.48334211159335727, "max":
0.48334211159335727}}}
#metrics {"StartTime": 1729802368.8345978, "EndTime": 1729802368.8346076,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48383210112778785, "count": 1, "min": 0.48383210112778785, "max":
0.48383210112778785}}}
#metrics {"StartTime": 1729802368.8346379, "EndTime": 1729802368.8346465,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4833001008573293, "count": 1, "min": 0.4833001008573293, "max":
0.4833001008573293}}}
#metrics {"StartTime": 1729802368.8346765, "EndTime": 1729802368.834685,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48378568550133194, "count": 1, "min": 0.48378568550133194, "max":
0.48378568550133194}}}
#metrics {"StartTime": 1729802368.8347185, "EndTime": 1729802368.8347292,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49451852281669595, "count": 1, "min": 0.49451852281669595, "max":
0.49451852281669595}}}

#metrics {"StartTime": 1729802368.8347588, "EndTime": 1729802368.834769,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5034419816043398, "count": 1, "min": 0.5034419816043398, "max":
0.5034419816043398}}}
#metrics {"StartTime": 1729802368.8348024, "EndTime": 1729802368.834812,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49449119045245904, "count": 1, "min": 0.49449119045245904, "max":
0.49449119045245904}}}
#metrics {"StartTime": 1729802368.8348458, "EndTime": 1729802368.8348565,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.50346705384794, "count": 1, "min": 0.50346705384794, "max":
0.50346705384794}}}
#metrics {"StartTime": 1729802368.8348906, "EndTime": 1729802368.8349013,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5059612143980253, "count": 1, "min": 0.5059612143980253, "max":
0.5059612143980253}}}
#metrics {"StartTime": 1729802368.8349364, "EndTime": 1729802368.8349466,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5060933309665878, "count": 1, "min": 0.5060933309665878, "max":
0.5060933309665878}}}

#metrics {"StartTime": 1729802368.8349795, "EndTime": 1729802368.8349905,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5059554905089398, "count": 1, "min": 0.5059554905089398, "max":
0.5059554905089398}}}
#metrics {"StartTime": 1729802368.835025, "EndTime": 1729802368.8350356,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5060575725310439, "count": 1, "min": 0.5060575725310439, "max":
0.5060575725310439}}}
#metrics {"StartTime": 1729802368.8350718, "EndTime": 1729802368.835082,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507342192460273, "count": 1, "min": 0.507342192460273, "max":
0.507342192460273}}}
#metrics {"StartTime": 1729802368.8351092, "EndTime": 1729802368.8351178,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5091846610486325, "count": 1, "min": 0.5091846610486325, "max":
0.5091846610486325}}}
#metrics {"StartTime": 1729802368.835147, "EndTime": 1729802368.8351557,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5073184235132433, "count": 1, "min": 0.5073184235132433, "max":
0.5073184235132433}}}

#metrics {"StartTime": 1729802368.8351827, "EndTime": 1729802368.8351915,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.509192109320871, "count": 1, "min": 0.509192109320871, "max":
0.509192109320871}}}
#metrics {"StartTime": 1729802368.835221, "EndTime": 1729802368.83523,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150309810813414, "count": 1, "min": 0.5150309810813414, "max":
0.5150309810813414}}}
#metrics {"StartTime": 1729802368.835261, "EndTime": 1729802368.8352711,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150545352620816, "count": 1, "min": 0.5150545352620816, "max":
0.5150545352620816}}}
#metrics {"StartTime": 1729802368.8353038, "EndTime": 1729802368.8353126,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150275845308916, "count": 1, "min": 0.5150275845308916, "max":
0.5150275845308916}}}
#metrics {"StartTime": 1729802368.8353465, "EndTime": 1729802368.835355,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150666991534218, "count": 1, "min": 0.5150666991534218, "max":
0.5150666991534218}}}

#metrics {"StartTime": 1729802368.8353837, "EndTime": 1729802368.8353927,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150316406623304, "count": 1, "min": 0.5150316406623304, "max":
0.5150316406623304}}}
#metrics {"StartTime": 1729802368.8354218, "EndTime": 1729802368.8354301,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.515751163797641, "count": 1, "min": 0.515751163797641, "max":
0.515751163797641}}}
#metrics {"StartTime": 1729802368.8354595, "EndTime": 1729802368.8354695,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150188810482667, "count": 1, "min": 0.5150188810482667, "max":
0.5150188810482667}}}
#metrics {"StartTime": 1729802368.835501, "EndTime": 1729802368.83551,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5157561756798981, "count": 1, "min": 0.5157561756798981, "max":
0.5157561756798981}}}
[10/24/2024 20:39:28 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=0, train binary_classification_cross_entropy_objective
<loss>=0.4830578342169432
[2024-10-24 20:39:28.855] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 0, "duration": 32300, "num_examples":
1, "num_bytes": 388000}
[2024-10-24 20:39:31.947] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 2, "duration": 3091, "num_examples":
164, "num_bytes": 63460892}

#metrics {"StartTime": 1729802371.9537594, "EndTime": 1729802371.953825,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48069846838764985, "count": 1, "min": 0.48069846838764985, "max":
0.48069846838764985}}}
#metrics {"StartTime": 1729802371.953902, "EndTime": 1729802371.9539135,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48179004121493907, "count": 1, "min": 0.48179004121493907, "max":
0.48179004121493907}}}
#metrics {"StartTime": 1729802371.9539466, "EndTime": 1729802371.9539564,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48069966495420086, "count": 1, "min": 0.48069966495420086, "max":
0.48069966495420086}}}
#metrics {"StartTime": 1729802371.9539926, "EndTime": 1729802371.954003,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48179020596907357, "count": 1, "min": 0.48179020596907357, "max":
0.48179020596907357}}}
#metrics {"StartTime": 1729802371.954033, "EndTime": 1729802371.9540398,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48900497722277636, "count": 1, "min": 0.48900497722277636, "max":
0.48900497722277636}}}

#metrics {"StartTime": 1729802371.9540696, "EndTime": 1729802371.954079,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5004189986846299, "count": 1, "min": 0.5004189986846299, "max":
0.5004189986846299}}}
#metrics {"StartTime": 1729802371.9541008, "EndTime": 1729802371.9541066,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4890071626873034, "count": 1, "min": 0.4890071626873034, "max":
0.4890071626873034}}}
#metrics {"StartTime": 1729802371.9541254, "EndTime": 1729802371.9541314,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5004109503611965, "count": 1, "min": 0.5004109503611965, "max":
0.5004109503611965}}}
#metrics {"StartTime": 1729802371.95416, "EndTime": 1729802371.954166,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4808837680903995, "count": 1, "min": 0.4808837680903995, "max":
0.4808837680903995}}}
#metrics {"StartTime": 1729802371.9541893, "EndTime": 1729802371.9541986,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48193697335547925, "count": 1, "min": 0.48193697335547925, "max":
0.48193697335547925}}}

#metrics {"StartTime": 1729802371.95423, "EndTime": 1729802371.9542365,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4808832208379913, "count": 1, "min": 0.4808832208379913, "max":
0.4808832208379913}}}
#metrics {"StartTime": 1729802371.954255, "EndTime": 1729802371.9542608,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48193520341468665, "count": 1, "min": 0.48193520341468665, "max":
0.48193520341468665}}}
#metrics {"StartTime": 1729802371.9542804, "EndTime": 1729802371.9542885,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48914809165737544, "count": 1, "min": 0.48914809165737544, "max":
0.48914809165737544}}}
#metrics {"StartTime": 1729802371.9543214, "EndTime": 1729802371.954331,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4994068054442152, "count": 1, "min": 0.4994068054442152, "max":
0.4994068054442152}}}
#metrics {"StartTime": 1729802371.9543557, "EndTime": 1729802371.9543624,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48915316731617586, "count": 1, "min": 0.48915316731617586, "max":
0.48915316731617586}}}

#metrics {"StartTime": 1729802371.9543903, "EndTime": 1729802371.9543982,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49939660206953224, "count": 1, "min": 0.49939660206953224, "max":
0.49939660206953224}}}
#metrics {"StartTime": 1729802371.954432, "EndTime": 1729802371.9544415,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5045263397819651, "count": 1, "min": 0.5045263397819651, "max":
0.5045263397819651}}}
#metrics {"StartTime": 1729802371.9544792, "EndTime": 1729802371.9544864,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5051114710875548, "count": 1, "min": 0.5051114710875548, "max":
0.5051114710875548}}}
#metrics {"StartTime": 1729802371.954518, "EndTime": 1729802371.9545271,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5045341834221758, "count": 1, "min": 0.5045341834221758, "max":
0.5045341834221758}}}
#metrics {"StartTime": 1729802371.9545557, "EndTime": 1729802371.9545636,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5051156559918387, "count": 1, "min": 0.5051156559918387, "max":
0.5051156559918387}}}

#metrics {"StartTime": 1729802371.9545925, "EndTime": 1729802371.954602,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5091314018135276, "count": 1, "min": 0.5091314018135276, "max":
0.5091314018135276}}}
#metrics {"StartTime": 1729802371.9546325, "EndTime": 1729802371.9546418,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5053033848385035, "count": 1, "min": 0.5053033848385035, "max":
0.5053033848385035}}}
#metrics {"StartTime": 1729802371.9546678, "EndTime": 1729802371.9546762,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5091349224769067, "count": 1, "min": 0.5091349224769067, "max":
0.5091349224769067}}}
#metrics {"StartTime": 1729802371.954707, "EndTime": 1729802371.9547164,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5053070321927743, "count": 1, "min": 0.5053070321927743, "max":
0.5053070321927743}}}
#metrics {"StartTime": 1729802371.9547493, "EndTime": 1729802371.954759,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.51371927735802, "count": 1, "min": 0.51371927735802, "max":
0.51371927735802}}}

#metrics {"StartTime": 1729802371.954792, "EndTime": 1729802371.954801,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5143974682545986, "count": 1, "min": 0.5143974682545986, "max":
0.5143974682545986}}}
#metrics {"StartTime": 1729802371.9548316, "EndTime": 1729802371.954841,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137282546857972, "count": 1, "min": 0.5137282546857972, "max":
0.5137282546857972}}}
#metrics {"StartTime": 1729802371.9548717, "EndTime": 1729802371.954881,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144111620659679, "count": 1, "min": 0.5144111620659679, "max":
0.5144111620659679}}}
#metrics {"StartTime": 1729802371.954911, "EndTime": 1729802371.9549217,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5152479845124814, "count": 1, "min": 0.5152479845124814, "max":
0.5152479845124814}}}
#metrics {"StartTime": 1729802371.9549475, "EndTime": 1729802371.9549572,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5141612143708344, "count": 1, "min": 0.5141612143708344, "max":
0.5141612143708344}}}

```
#metrics {"StartTime": 1729802371.954988, "EndTime": 1729802371.9549973,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.51525550965291, "count": 1, "min": 0.51525550965291, "max":
0.51525550965291}}}
#metrics {"StartTime": 1729802371.9550335, "EndTime": 1729802371.9550416,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139069177636205, "count": 1, "min": 0.5139069177636205, "max":
0.5139069177636205}}}
[10/24/2024 20:39:31 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=0, validation binary_classification_cross_entropy_objective
<loss>=0.48069846838764985
[10/24/2024 20:39:31 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=0, criteria=binary_classification_cross_entropy_objective,
value=0.48069846838764985
[10/24/2024 20:39:31 INFO 139943714142016] Epoch 0: Loss improved. Updating
best model
[10/24/2024 20:39:31 INFO 139943714142016] Saving model for epoch: 0
[10/24/2024 20:39:31 INFO 139943714142016] Saved checkpoint to
"/tmp/tmps191lg1n/mx-mod-0000.params"
[10/24/2024 20:39:31 INFO 139943714142016] #progress_metric: host=algo-1,
completed 6.666666666666667 % of epochs
```

#metrics {"StartTime": 1729802336.8613813, "EndTime": 1729802371.962157, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "Meta": "training_data_iter"}, "Metrics": {"Total Records Seen": {"sum": 1320472.0, "count": 1, "min": 1320472, "max": 1320472}, "Total Batches Seen": {"sum": 1321.0, "count": 1, "min": 1321, "max": 1321}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 3.0, "count": 1, "min": 3, "max": 3}, "Number of Records Since Last Reset": {"sum": 1308472.0, "count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:39:31 INFO 139943714142016] #throughput_metric: host=algo-1, train throughput=37277.472090308685 records/second
[2024-10-24 20:40:03.787] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 5, "duration": 31825, "num_examples": 1309, "num_bytes": 507687136}
#metrics {"StartTime": 1729802403.7878292, "EndTime": 1729802403.7882063, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 0}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4808765957333626, "count": 1, "min": 0.4808765957333626, "max": 0.4808765957333626}}}
#metrics {"StartTime": 1729802403.7882872, "EndTime": 1729802403.7884362, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 1}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4818171103007932, "count": 1, "min": 0.4818171103007932, "max": 0.4818171103007932}}}

#metrics {"StartTime": 1729802403.7884836, "EndTime": 1729802403.7884943,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4808772088613714, "count": 1, "min": 0.4808772088613714, "max":
0.4808772088613714}}}
#metrics {"StartTime": 1729802403.7886431, "EndTime": 1729802403.7886546,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48181700885551054, "count": 1, "min": 0.48181700885551054, "max":
0.48181700885551054}}}
#metrics {"StartTime": 1729802403.7889307, "EndTime": 1729802403.7889435,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4846935196856111, "count": 1, "min": 0.4846935196856111, "max":
0.4846935196856111}}}
#metrics {"StartTime": 1729802403.7889686, "EndTime": 1729802403.788976,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.501487706618936, "count": 1, "min": 0.501487706618936, "max":
0.501487706618936}}}
#metrics {"StartTime": 1729802403.7889986, "EndTime": 1729802403.7890055,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4846934336624379, "count": 1, "min": 0.4846934336624379, "max":
0.4846934336624379}}}

#metrics {"StartTime": 1729802403.7890272, "EndTime": 1729802403.789034,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5014880851255644, "count": 1, "min": 0.5014880851255644, "max":
0.5014880851255644}}}
#metrics {"StartTime": 1729802403.7890558, "EndTime": 1729802403.7890618,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4811311261238308, "count": 1, "min": 0.4811311261238308, "max":
0.4811311261238308}}}
#metrics {"StartTime": 1729802403.7890832, "EndTime": 1729802403.7890894,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4820154693323538, "count": 1, "min": 0.4820154693323538, "max":
0.4820154693323538}}}
#metrics {"StartTime": 1729802403.789111, "EndTime": 1729802403.7891169,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4811308270375663, "count": 1, "min": 0.4811308270375663, "max":
0.4811308270375663}}}
#metrics {"StartTime": 1729802403.789138, "EndTime": 1729802403.7891443,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48201559527569227, "count": 1, "min": 0.48201559527569227, "max":
0.48201559527569227}}}

#metrics {"StartTime": 1729802403.7891657, "EndTime": 1729802403.789172,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48470391078097375, "count": 1, "min": 0.48470391078097375, "max":
0.48470391078097375}}}
#metrics {"StartTime": 1729802403.7891932, "EndTime": 1729802403.7891994,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5003036810732033, "count": 1, "min": 0.5003036810732033, "max":
0.5003036810732033}}}
#metrics {"StartTime": 1729802403.7892208, "EndTime": 1729802403.7892272,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48470391115427747, "count": 1, "min": 0.48470391115427747, "max":
0.48470391115427747}}}
#metrics {"StartTime": 1729802403.7892506, "EndTime": 1729802403.789257,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5003023142916712, "count": 1, "min": 0.5003023142916712, "max":
0.5003023142916712}}}
#metrics {"StartTime": 1729802403.7892783, "EndTime": 1729802403.7892845,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045528268609937, "count": 1, "min": 0.5045528268609937, "max":
0.5045528268609937}}}

#metrics {"StartTime": 1729802403.7893057, "EndTime": 1729802403.789312,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047902124854038, "count": 1, "min": 0.5047902124854038, "max":
0.5047902124854038}}}
#metrics {"StartTime": 1729802403.7893333, "EndTime": 1729802403.7893393,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045527979999507, "count": 1, "min": 0.5045527979999507, "max":
0.5045527979999507}}}
#metrics {"StartTime": 1729802403.7893603, "EndTime": 1729802403.7893667,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047902593350192, "count": 1, "min": 0.5047902593350192, "max":
0.5047902593350192}}}
#metrics {"StartTime": 1729802403.7893875, "EndTime": 1729802403.7893937,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5053947990487475, "count": 1, "min": 0.5053947990487475, "max":
0.5053947990487475}}}
#metrics {"StartTime": 1729802403.7894144, "EndTime": 1729802403.7894206,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5083435083091806, "count": 1, "min": 0.5083435083091806, "max":
0.5083435083091806}}}

#metrics {"StartTime": 1729802403.7894416, "EndTime": 1729802403.7894478,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5053953143011904, "count": 1, "min": 0.5053953143011904, "max":
0.5053953143011904}}}
#metrics {"StartTime": 1729802403.7897363, "EndTime": 1729802403.7897437,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5083426234627353, "count": 1, "min": 0.5083426234627353, "max":
0.5083426234627353}}}
#metrics {"StartTime": 1729802403.7897656, "EndTime": 1729802403.789772,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.51373984261609, "count": 1, "min": 0.51373984261609, "max":
0.51373984261609}}}
#metrics {"StartTime": 1729802403.7897935, "EndTime": 1729802403.7898,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.513838270274871, "count": 1, "min": 0.513838270274871, "max":
0.513838270274871}}}
#metrics {"StartTime": 1729802403.7898228, "EndTime": 1729802403.7898288,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137403497025143, "count": 1, "min": 0.5137403497025143, "max":
0.5137403497025143}}}

#metrics {"StartTime": 1729802403.7898505, "EndTime": 1729802403.7898567,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138379149597354, "count": 1, "min": 0.5138379149597354, "max":
0.5138379149597354}}}
#metrics {"StartTime": 1729802403.7898781, "EndTime": 1729802403.7898843,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5141211516820692, "count": 1, "min": 0.5141211516820692, "max":
0.5141211516820692}}}
#metrics {"StartTime": 1729802403.7899055, "EndTime": 1729802403.7899117,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5152099850389207, "count": 1, "min": 0.5152099850389207, "max":
0.5152099850389207}}}
#metrics {"StartTime": 1729802403.789933, "EndTime": 1729802403.7899392,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5141208339306192, "count": 1, "min": 0.5141208339306192, "max":
0.5141208339306192}}}
#metrics {"StartTime": 1729802403.7899606, "EndTime": 1729802403.7899668,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5152026423183056, "count": 1, "min": 0.5152026423183056, "max":
0.5152026423183056}}}
[10/24/2024 20:40:03 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=1, train binary_classification_cross_entropy_objective
<loss>=0.4808765957333626

[2024-10-24 20:40:06.653] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 5, "duration": 2843, "num_examples":
164, "num_bytes": 63460892}
#metrics {"StartTime": 1729802406.6599808, "EndTime": 1729802406.6600485,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.480478867545802, "count": 1, "min": 0.480478867545802, "max":
0.480478867545802}}}
#metrics {"StartTime": 1729802406.6601195, "EndTime": 1729802406.66013,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4818089802904409, "count": 1, "min": 0.4818089802904409, "max":
0.4818089802904409}}}
#metrics {"StartTime": 1729802406.660169, "EndTime": 1729802406.6601782,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4804794317774019, "count": 1, "min": 0.4804794317774019, "max":
0.4804794317774019}}}
#metrics {"StartTime": 1729802406.6602094, "EndTime": 1729802406.660219,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4818089767453349, "count": 1, "min": 0.4818089767453349, "max":
0.4818089767453349}}}
#metrics {"StartTime": 1729802406.6602478, "EndTime": 1729802406.660257,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4821814714764268, "count": 1, "min": 0.4821814714764268, "max":
0.4821814714764268}}}

#metrics {"StartTime": 1729802406.660286, "EndTime": 1729802406.6602924,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4979750612289803, "count": 1, "min": 0.4979750612289803, "max":
0.4979750612289803}}}
#metrics {"StartTime": 1729802406.6603115, "EndTime": 1729802406.6603184,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48218139180483516, "count": 1, "min": 0.48218139180483516, "max":
0.48218139180483516}}}
#metrics {"StartTime": 1729802406.6603475, "EndTime": 1729802406.6603556,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4979748599042791, "count": 1, "min": 0.4979748599042791, "max":
0.4979748599042791}}}
#metrics {"StartTime": 1729802406.660388, "EndTime": 1729802406.6603966,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48071902906924785, "count": 1, "min": 0.48071902906924785, "max":
0.48071902906924785}}}
#metrics {"StartTime": 1729802406.6604304, "EndTime": 1729802406.66044,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4819624684513931, "count": 1, "min": 0.4819624684513931, "max":
0.4819624684513931}}}

#metrics {"StartTime": 1729802406.6604762, "EndTime": 1729802406.6604848,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4807189452927967, "count": 1, "min": 0.4807189452927967, "max":
0.4807189452927967}}}
#metrics {"StartTime": 1729802406.6605191, "EndTime": 1729802406.6605277,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4819620831543515, "count": 1, "min": 0.4819620831543515, "max":
0.4819620831543515}}}
#metrics {"StartTime": 1729802406.6605659, "EndTime": 1729802406.6605754,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48229731844872065, "count": 1, "min": 0.48229731844872065, "max":
0.48229731844872065}}}
#metrics {"StartTime": 1729802406.6606126, "EndTime": 1729802406.6606226,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49707095304467014, "count": 1, "min": 0.49707095304467014, "max":
0.49707095304467014}}}
#metrics {"StartTime": 1729802406.6606605, "EndTime": 1729802406.66067,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48229758918286464, "count": 1, "min": 0.48229758918286464, "max":
0.48229758918286464}}}

#metrics {"StartTime": 1729802406.6607072, "EndTime": 1729802406.6607172,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.497069449359993, "count": 1, "min": 0.497069449359993, "max":
0.497069449359993}}}
#metrics {"StartTime": 1729802406.6607711, "EndTime": 1729802406.6607814,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.504398646744841, "count": 1, "min": 0.504398646744841, "max":
0.504398646744841}}}
#metrics {"StartTime": 1729802406.6608176, "EndTime": 1729802406.6608276,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052969978636699, "count": 1, "min": 0.5052969978636699, "max":
0.5052969978636699}}}
#metrics {"StartTime": 1729802406.6608753, "EndTime": 1729802406.6608856,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5043990411845243, "count": 1, "min": 0.5043990411845243, "max":
0.5043990411845243}}}
#metrics {"StartTime": 1729802406.6609247, "EndTime": 1729802406.6609347,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052976160181972, "count": 1, "min": 0.5052976160181972, "max":
0.5052976160181972}}}

#metrics {"StartTime": 1729802406.6609697, "EndTime": 1729802406.6609788,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5049937916686529, "count": 1, "min": 0.5049937916686529, "max":
0.5049937916686529}}}
#metrics {"StartTime": 1729802406.661015, "EndTime": 1729802406.6610253,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5076101998349506, "count": 1, "min": 0.5076101998349506, "max":
0.5076101998349506}}}
#metrics {"StartTime": 1729802406.661062, "EndTime": 1729802406.6610727,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5049939174266219, "count": 1, "min": 0.5049939174266219, "max":
0.5049939174266219}}}
#metrics {"StartTime": 1729802406.661108, "EndTime": 1729802406.6611185,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5076095768292257, "count": 1, "min": 0.5076095768292257, "max":
0.5076095768292257}}}
#metrics {"StartTime": 1729802406.6611495, "EndTime": 1729802406.66116,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137390862905716, "count": 1, "min": 0.5137390862905716, "max":
0.5137390862905716}}}

#metrics {"StartTime": 1729802406.661197, "EndTime": 1729802406.6612074,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.514550329491551, "count": 1, "min": 0.514550329491551, "max":
0.514550329491551}}}
#metrics {"StartTime": 1729802406.6612473, "EndTime": 1729802406.6612582,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137334169198166, "count": 1, "min": 0.5137334169198166, "max":
0.5137334169198166}}}
#metrics {"StartTime": 1729802406.661297, "EndTime": 1729802406.661307,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5145547668446956, "count": 1, "min": 0.5145547668446956, "max":
0.5145547668446956}}}
#metrics {"StartTime": 1729802406.661358, "EndTime": 1729802406.6613672,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5138886155010983, "count": 1, "min": 0.5138886155010983, "max":
0.5138886155010983}}}
#metrics {"StartTime": 1729802406.6613922, "EndTime": 1729802406.6613982,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.514144346196962, "count": 1, "min": 0.514144346196962, "max":
0.514144346196962}}}

#metrics {"StartTime": 1729802406.6614208, "EndTime": 1729802406.6614268,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5138966235222745, "count": 1, "min": 0.5138966235222745, "max":
0.5138966235222745}}}
#metrics {"StartTime": 1729802406.6614494, "EndTime": 1729802406.6614556,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5141307902715557, "count": 1, "min": 0.5141307902715557, "max":
0.5141307902715557}}}
[10/24/2024 20:40:06 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=1, validation binary_classification_cross_entropy_objective
<loss>=0.480478867545802
[10/24/2024 20:40:06 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=1, criteria=binary_classification_cross_entropy_objective,
value=0.480478867545802
[10/24/2024 20:40:06 INFO 139943714142016] Saving model for epoch: 1
[10/24/2024 20:40:06 INFO 139943714142016] Saved checkpoint to
"/tmp/tmp3f2x4xq3/mx-mod-0000.params"
[10/24/2024 20:40:06 INFO 139943714142016] #progress_metric: host=algo-1,
completed 13.333333333333334 % of epochs
#metrics {"StartTime": 1729802371.962366, "EndTime": 1729802406.6664808,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 2628944.0, "count": 1, "min": 2628944, "max": 2628944},
"Total Batches Seen": {"sum": 2630.0, "count": 1, "min": 2630, "max": 2630},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 4.0, "count": 1,
"min": 4, "max": 4}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}

[10/24/2024 20:40:06 INFO 139943714142016] #throughput_metric: host=algo-1,
train throughput=37703.53447388967 records/second
[2024-10-24 20:40:38.072] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 7, "duration": 31403, "num_examples": 1309,
"num_bytes": 507687136}
#metrics {"StartTime": 1729802438.072735, "EndTime": 1729802438.0728061,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48071617509101144, "count": 1, "min": 0.48071617509101144, "max":
0.48071617509101144}}}
#metrics {"StartTime": 1729802438.0728831, "EndTime": 1729802438.0728996,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4816903024142671, "count": 1, "min": 0.4816903024142671, "max":
0.4816903024142671}}}
#metrics {"StartTime": 1729802438.0729337, "EndTime": 1729802438.0729454,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48071651290753564, "count": 1, "min": 0.48071651290753564, "max":
0.48071651290753564}}}
#metrics {"StartTime": 1729802438.0729747, "EndTime": 1729802438.0729854,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.481690276936289, "count": 1, "min": 0.481690276936289, "max":
0.481690276936289}}}

#metrics {"StartTime": 1729802438.073013, "EndTime": 1729802438.0730233,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48177246420390746, "count": 1, "min": 0.48177246420390746, "max":
0.48177246420390746}}}
#metrics {"StartTime": 1729802438.0730498, "EndTime": 1729802438.0730603,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4988951502808737, "count": 1, "min": 0.4988951502808737, "max":
0.4988951502808737}}}
#metrics {"StartTime": 1729802438.0730872, "EndTime": 1729802438.0730972,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48177246180076483, "count": 1, "min": 0.48177246180076483, "max":
0.48177246180076483}}}
#metrics {"StartTime": 1729802438.0731237, "EndTime": 1729802438.07314,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49889521042943363, "count": 1, "min": 0.49889521042943363, "max":
0.49889521042943363}}}
#metrics {"StartTime": 1729802438.0731695, "EndTime": 1729802438.0731795,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48098145574167234, "count": 1, "min": 0.48098145574167234, "max":
0.48098145574167234}}}

#metrics {"StartTime": 1729802438.0732055, "EndTime": 1729802438.073216,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48189740494194383, "count": 1, "min": 0.48189740494194383, "max":
0.48189740494194383}}}
#metrics {"StartTime": 1729802438.0732424, "EndTime": 1729802438.0732524,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48098133150153205, "count": 1, "min": 0.48098133150153205, "max":
0.48098133150153205}}}
#metrics {"StartTime": 1729802438.07328, "EndTime": 1729802438.0732906,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4818974254736477, "count": 1, "min": 0.4818974254736477, "max":
0.4818974254736477}}}
#metrics {"StartTime": 1729802438.0733168, "EndTime": 1729802438.073327,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4819743229810616, "count": 1, "min": 0.4819743229810616, "max":
0.4819743229810616}}}
#metrics {"StartTime": 1729802438.073353, "EndTime": 1729802438.0733628,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4979167594559696, "count": 1, "min": 0.4979167594559696, "max":
0.4979167594559696}}}

#metrics {"StartTime": 1729802438.0733893, "EndTime": 1729802438.073399,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4819743525887119, "count": 1, "min": 0.4819743525887119, "max":
0.4819743525887119}}}
#metrics {"StartTime": 1729802438.073425, "EndTime": 1729802438.0734353,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4979164761651182, "count": 1, "min": 0.4979164761651182, "max":
0.4979164761651182}}}
#metrics {"StartTime": 1729802438.0734613, "EndTime": 1729802438.0734713,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045281843366244, "count": 1, "min": 0.5045281843366244, "max":
0.5045281843366244}}}
#metrics {"StartTime": 1729802438.0734966, "EndTime": 1729802438.0735066,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047688854112538, "count": 1, "min": 0.5047688854112538, "max":
0.5047688854112538}}}
#metrics {"StartTime": 1729802438.0735328, "EndTime": 1729802438.0735428,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045281889329263, "count": 1, "min": 0.5045281889329263, "max":
0.5045281889329263}}}

#metrics {"StartTime": 1729802438.0735703, "EndTime": 1729802438.0735807,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047688772452351, "count": 1, "min": 0.5047688772452351, "max":
0.5047688772452351}}}
#metrics {"StartTime": 1729802438.0736067, "EndTime": 1729802438.0736167,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047635312094966, "count": 1, "min": 0.5047635312094966, "max":
0.5047635312094966}}}
#metrics {"StartTime": 1729802438.0736425, "EndTime": 1729802438.073653,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078370848489464, "count": 1, "min": 0.5078370848489464, "max":
0.5078370848489464}}}
#metrics {"StartTime": 1729802438.0736787, "EndTime": 1729802438.0736883,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047635578307172, "count": 1, "min": 0.5047635578307172, "max":
0.5047635578307172}}}
#metrics {"StartTime": 1729802438.0737164, "EndTime": 1729802438.0737267,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5078369772908156, "count": 1, "min": 0.5078369772908156, "max":
0.5078369772908156}}}

#metrics {"StartTime": 1729802438.0737529, "EndTime": 1729802438.0737612,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137313388276173, "count": 1, "min": 0.5137313388276173, "max":
0.5137313388276173}}}
#metrics {"StartTime": 1729802438.0737896, "EndTime": 1729802438.0737998,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138400880539453, "count": 1, "min": 0.5138400880539453, "max":
0.5138400880539453}}}
#metrics {"StartTime": 1729802438.0738263, "EndTime": 1729802438.0738356,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137315426514418, "count": 1, "min": 0.5137315426514418, "max":
0.5137315426514418}}}
#metrics {"StartTime": 1729802438.0738738, "EndTime": 1729802438.073885,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.513839881477006, "count": 1, "min": 0.513839881477006, "max":
0.513839881477006}}}
#metrics {"StartTime": 1729802438.0739121, "EndTime": 1729802438.0739226,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138332368809878, "count": 1, "min": 0.5138332368809878, "max":
0.5138332368809878}}}

#metrics {"StartTime": 1729802438.0739481, "EndTime": 1729802438.0739584,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150235185069046, "count": 1, "min": 0.5150235185069046, "max":
0.5150235185069046}}}
#metrics {"StartTime": 1729802438.0739841, "EndTime": 1729802438.0739944,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138333335433166, "count": 1, "min": 0.5138333335433166, "max":
0.5138333335433166}}}
#metrics {"StartTime": 1729802438.07402, "EndTime": 1729802438.0740304,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5150233853774697, "count": 1, "min": 0.5150233853774697, "max":
0.5150233853774697}}}
[10/24/2024 20:40:38 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=2, train binary_classification_cross_entropy_objective
<loss>=0.48071617509101144
[2024-10-24 20:40:40.822] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 8, "duration": 2734, "num_examples":
164, "num_bytes": 63460892}
#metrics {"StartTime": 1729802440.828196, "EndTime": 1729802440.828254,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4803898497487671, "count": 1, "min": 0.4803898497487671, "max":
0.4803898497487671}}}

#metrics {"StartTime": 1729802440.8283238, "EndTime": 1729802440.8283346,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4816682393975688, "count": 1, "min": 0.4816682393975688, "max":
0.4816682393975688}}}
#metrics {"StartTime": 1729802440.8283682, "EndTime": 1729802440.828376,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4803903632293765, "count": 1, "min": 0.4803903632293765, "max":
0.4803903632293765}}}
#metrics {"StartTime": 1729802440.8284109, "EndTime": 1729802440.82842,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4816682259634831, "count": 1, "min": 0.4816682259634831, "max":
0.4816682259634831}}}
#metrics {"StartTime": 1729802440.8284547, "EndTime": 1729802440.828464,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48096961786880515, "count": 1, "min": 0.48096961786880515, "max":
0.48096961786880515}}}
#metrics {"StartTime": 1729802440.8284981, "EndTime": 1729802440.8285086,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4965108179079778, "count": 1, "min": 0.4965108179079778, "max":
0.4965108179079778}}}

#metrics {"StartTime": 1729802440.8285384, "EndTime": 1729802440.8285491,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4809696665673659, "count": 1, "min": 0.4809696665673659, "max":
0.4809696665673659}}}
#metrics {"StartTime": 1729802440.8285797, "EndTime": 1729802440.8285882,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49651092220872656, "count": 1, "min": 0.49651092220872656, "max":
0.49651092220872656}}}
#metrics {"StartTime": 1729802440.828623, "EndTime": 1729802440.828632,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48064476861536776, "count": 1, "min": 0.48064476861536776, "max":
0.48064476861536776}}}
#metrics {"StartTime": 1729802440.8286626, "EndTime": 1729802440.8286703,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48182625577832866, "count": 1, "min": 0.48182625577832866, "max":
0.48182625577832866}}}
#metrics {"StartTime": 1729802440.8287024, "EndTime": 1729802440.828709,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48064476861536776, "count": 1, "min": 0.48064476861536776, "max":
0.48064476861536776}}}

#metrics {"StartTime": 1729802440.8287506, "EndTime": 1729802440.8287601,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48182616248606674, "count": 1, "min": 0.48182616248606674, "max":
0.48182616248606674}}}
#metrics {"StartTime": 1729802440.8287854, "EndTime": 1729802440.8287916,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48113587774041017, "count": 1, "min": 0.48113587774041017, "max":
0.48113587774041017}}}
#metrics {"StartTime": 1729802440.8288171, "EndTime": 1729802440.8288233,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4955489457672886, "count": 1, "min": 0.4955489457672886, "max":
0.4955489457672886}}}
#metrics {"StartTime": 1729802440.828853, "EndTime": 1729802440.8288612,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48113587419530424, "count": 1, "min": 0.48113587419530424, "max":
0.48113587419530424}}}
#metrics {"StartTime": 1729802440.8289032, "EndTime": 1729802440.8289127,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4955489202052088, "count": 1, "min": 0.4955489202052088, "max":
0.4955489202052088}}}

#metrics {"StartTime": 1729802440.8289404, "EndTime": 1729802440.8289485,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044086125974252, "count": 1, "min": 0.5044086125974252, "max":
0.5044086125974252}}}
#metrics {"StartTime": 1729802440.8289828, "EndTime": 1729802440.828992,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052977339396162, "count": 1, "min": 0.5052977339396162, "max":
0.5052977339396162}}}
#metrics {"StartTime": 1729802440.8290248, "EndTime": 1729802440.8290343,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044087909722299, "count": 1, "min": 0.5044087909722299, "max":
0.5044087909722299}}}
#metrics {"StartTime": 1729802440.829064, "EndTime": 1729802440.8290737,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052980089652043, "count": 1, "min": 0.5052980089652043, "max":
0.5052980089652043}}}
#metrics {"StartTime": 1729802440.8291113, "EndTime": 1729802440.829119,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5047589761649824, "count": 1, "min": 0.5047589761649824, "max":
0.5047589761649824}}}

#metrics {"StartTime": 1729802440.829151, "EndTime": 1729802440.8291605,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077958478910062, "count": 1, "min": 0.5077958478910062, "max":
0.5077958478910062}}}
#metrics {"StartTime": 1729802440.8291922, "EndTime": 1729802440.8292017,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.504759023370867, "count": 1, "min": 0.504759023370867, "max":
0.504759023370867}}}
#metrics {"StartTime": 1729802440.8292377, "EndTime": 1729802440.8292463,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5077956794051812, "count": 1, "min": 0.5077956794051812, "max":
0.5077956794051812}}}
#metrics {"StartTime": 1729802440.8292823, "EndTime": 1729802440.8292916,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137193630003164, "count": 1, "min": 0.5137193630003164, "max":
0.5137193630003164}}}
#metrics {"StartTime": 1729802440.829326, "EndTime": 1729802440.8293357,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5144888334715273, "count": 1, "min": 0.5144888334715273, "max":
0.5144888334715273}}}

#metrics {"StartTime": 1729802440.8293688, "EndTime": 1729802440.8293784,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137195837298081, "count": 1, "min": 0.5137195837298081, "max":
0.5137195837298081}}}
#metrics {"StartTime": 1729802440.829411, "EndTime": 1729802440.8294206,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.514490356560995, "count": 1, "min": 0.514490356560995, "max":
0.514490356560995}}}
#metrics {"StartTime": 1729802440.8294542, "EndTime": 1729802440.8294616,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139028313759652, "count": 1, "min": 0.5139028313759652, "max":
0.5139028313759652}}}
#metrics {"StartTime": 1729802440.8294957, "EndTime": 1729802440.8295057,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5141820532499572, "count": 1, "min": 0.5141820532499572, "max":
0.5141820532499572}}}
#metrics {"StartTime": 1729802440.8295393, "EndTime": 1729802440.8295493,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5139201096626208, "count": 1, "min": 0.5139201096626208, "max":
0.5139201096626208}}}

```
#metrics {"StartTime": 1729802440.8295767, "EndTime": 1729802440.829586,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5141820560487251, "count": 1, "min": 0.5141820560487251, "max":
0.5141820560487251}}}
[10/24/2024 20:40:40 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=2, validation binary_classification_cross_entropy_objective
<loss>=0.4803898497487671
[10/24/2024 20:40:40 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=2, criteria=binary_classification_cross_entropy_objective,
value=0.4803898497487671
[10/24/2024 20:40:40 INFO 139943714142016] Saving model for epoch: 2
[10/24/2024 20:40:40 INFO 139943714142016] Saved checkpoint to
"/tmp/tmpeolvxqlr/mx-mod-0000.params"
[10/24/2024 20:40:40 INFO 139943714142016] #progress_metric: host=algo-1,
completed 20.0 % of epochs
#metrics {"StartTime": 1729802406.668646, "EndTime": 1729802440.8353963,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 3937416.0, "count": 1, "min": 3937416, "max": 3937416},
"Total Batches Seen": {"sum": 3939.0, "count": 1, "min": 3939, "max": 3939},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 5.0, "count": 1,
"min": 5, "max": 5}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:40:40 INFO 139943714142016] #throughput_metric: host=algo-1,
train throughput=38296.50460233517 records/second
[2024-10-24 20:41:12.292] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 9, "duration": 31450, "num_examples": 1309,
"num_bytes": 507687136}
```

#metrics {"StartTime": 1729802472.2930436, "EndTime": 1729802472.293113,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4806527144639135, "count": 1, "min": 0.4806527144639135, "max":
0.4806527144639135}}}
#metrics {"StartTime": 1729802472.2931826, "EndTime": 1729802472.293207,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4815649320503258, "count": 1, "min": 0.4815649320503258, "max":
0.4815649320503258}}}
#metrics {"StartTime": 1729802472.2932403, "EndTime": 1729802472.2932498,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4806675682884473, "count": 1, "min": 0.4806675682884473, "max":
0.4806675682884473}}}
#metrics {"StartTime": 1729802472.2932866, "EndTime": 1729802472.2932968,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.481564926684085, "count": 1, "min": 0.481564926684085, "max":
0.481564926684085}}}
#metrics {"StartTime": 1729802472.2933276, "EndTime": 1729802472.2933366,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48096424702402285, "count": 1, "min": 0.48096424702402285, "max":
0.48096424702402285}}}

#metrics {"StartTime": 1729802472.2933645, "EndTime": 1729802472.2933726,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49669678312995746, "count": 1, "min": 0.49669678312995746, "max":
0.49669678312995746}}}
#metrics {"StartTime": 1729802472.2934022, "EndTime": 1729802472.2934115,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.480964247513984, "count": 1, "min": 0.480964247513984, "max":
0.480964247513984}}}
#metrics {"StartTime": 1729802472.2934434, "EndTime": 1729802472.2934525,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49669678606972417, "count": 1, "min": 0.49669678606972417, "max":
0.49669678606972417}}}
#metrics {"StartTime": 1729802472.2934825, "EndTime": 1729802472.2934914,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48092180894857517, "count": 1, "min": 0.48092180894857517, "max":
0.48092180894857517}}}
#metrics {"StartTime": 1729802472.2935228, "EndTime": 1729802472.2935326,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48177960984349616, "count": 1, "min": 0.48177960984349616, "max":
0.48177960984349616}}}

#metrics {"StartTime": 1729802472.2935624, "EndTime": 1729802472.293571,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4809352214241611, "count": 1, "min": 0.4809352214241611, "max":
0.4809352214241611}}}
#metrics {"StartTime": 1729802472.2936008, "EndTime": 1729802472.2936096,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48177961275993136, "count": 1, "min": 0.48177961275993136, "max":
0.48177961275993136}}}
#metrics {"StartTime": 1729802472.2936416, "EndTime": 1729802472.2936509,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48121522717957105, "count": 1, "min": 0.48121522717957105, "max":
0.48121522717957105}}}
#metrics {"StartTime": 1729802472.2936869, "EndTime": 1729802472.2936976,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.495851781069321, "count": 1, "min": 0.495851781069321, "max":
0.495851781069321}}}
#metrics {"StartTime": 1729802472.2937236, "EndTime": 1729802472.2937343,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48121523032932106, "count": 1, "min": 0.48121523032932106, "max":
0.48121523032932106}}}

#metrics {"StartTime": 1729802472.2937605, "EndTime": 1729802472.2937706,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4958517288068019, "count": 1, "min": 0.4958517288068019, "max":
0.4958517288068019}}}
#metrics {"StartTime": 1729802472.2938023, "EndTime": 1729802472.293813,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045148283698873, "count": 1, "min": 0.5045148283698873, "max":
0.5045148283698873}}}
#metrics {"StartTime": 1729802472.293843, "EndTime": 1729802472.2938526,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047366810369929, "count": 1, "min": 0.5047366810369929, "max":
0.5047366810369929}}}
#metrics {"StartTime": 1729802472.2938817, "EndTime": 1729802472.293892,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045160635618624, "count": 1, "min": 0.5045160635618624, "max":
0.5045160635618624}}}
#metrics {"StartTime": 1729802472.2939177, "EndTime": 1729802472.2939265,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047366749941391, "count": 1, "min": 0.5047366749941391, "max":
0.5047366749941391}}}

#metrics {"StartTime": 1729802472.2939608, "EndTime": 1729802472.2939713,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045690293384983, "count": 1, "min": 0.5045690293384983, "max":
0.5045690293384983}}}
#metrics {"StartTime": 1729802472.2940204, "EndTime": 1729802472.2940304,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.507393019766618, "count": 1, "min": 0.507393019766618, "max":
0.507393019766618}}}
#metrics {"StartTime": 1729802472.2940578, "EndTime": 1729802472.2940662,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045690295251503, "count": 1, "min": 0.5045690295251503, "max":
0.5045690295251503}}}
#metrics {"StartTime": 1729802472.2940943, "EndTime": 1729802472.2941022,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5073930113206216, "count": 1, "min": 0.5073930113206216, "max":
0.5073930113206216}}}
#metrics {"StartTime": 1729802472.294132, "EndTime": 1729802472.2941418,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137268196595918, "count": 1, "min": 0.5137268196595918, "max":
0.5137268196595918}}}

#metrics {"StartTime": 1729802472.2941716, "EndTime": 1729802472.294182,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138282685352757, "count": 1, "min": 0.5138282685352757, "max":
0.5138282685352757}}}
#metrics {"StartTime": 1729802472.2942083, "EndTime": 1729802472.2942188,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137284581595605, "count": 1, "min": 0.5137284581595605, "max":
0.5137284581595605}}}
#metrics {"StartTime": 1729802472.2942448, "EndTime": 1729802472.2942533,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138283050257132, "count": 1, "min": 0.5138283050257132, "max":
0.5138283050257132}}}
#metrics {"StartTime": 1729802472.2942855, "EndTime": 1729802472.294296,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137474204013837, "count": 1, "min": 0.5137474204013837, "max":
0.5137474204013837}}}
#metrics {"StartTime": 1729802472.2943273, "EndTime": 1729802472.294337,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5148148075302077, "count": 1, "min": 0.5148148075302077, "max":
0.5148148075302077}}}

#metrics {"StartTime": 1729802472.2943714, "EndTime": 1729802472.2943816,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137475094576495, "count": 1, "min": 0.5137475094576495, "max":
0.5137475094576495}}}
#metrics {"StartTime": 1729802472.2944136, "EndTime": 1729802472.2944233,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5148148043571262, "count": 1, "min": 0.5148148043571262, "max":
0.5148148043571262}}}
[10/24/2024 20:41:12 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=3, train binary_classification_cross_entropy_objective
<loss>=0.4806527144639135
[2024-10-24 20:41:15.153] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 11, "duration": 2844, "num_examples":
164, "num_bytes": 63460892}
#metrics {"StartTime": 1729802475.1587973, "EndTime": 1729802475.1588502,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48036658583058855, "count": 1, "min": 0.48036658583058855, "max":
0.48036658583058855}}}
#metrics {"StartTime": 1729802475.1589196, "EndTime": 1729802475.158933,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4814940879335305, "count": 1, "min": 0.4814940879335305, "max":
0.4814940879335305}}}

#metrics {"StartTime": 1729802475.1589737, "EndTime": 1729802475.1589851,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.480389334962066, "count": 1, "min": 0.480389334962066, "max":
0.480389334962066}}}
#metrics {"StartTime": 1729802475.1590145, "EndTime": 1729802475.1590254,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4814940677824019, "count": 1, "min": 0.4814940677824019, "max":
0.4814940677824019}}}
#metrics {"StartTime": 1729802475.1590528, "EndTime": 1729802475.159063,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48051821150482443, "count": 1, "min": 0.48051821150482443, "max":
0.48051821150482443}}}
#metrics {"StartTime": 1729802475.159093, "EndTime": 1729802475.1591024,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49538685046225434, "count": 1, "min": 0.49538685046225434, "max":
0.49538685046225434}}}
#metrics {"StartTime": 1729802475.1591406, "EndTime": 1729802475.1591494,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4805182169157756, "count": 1, "min": 0.4805182169157756, "max":
0.4805182169157756}}}

#metrics {"StartTime": 1729802475.1591775, "EndTime": 1729802475.1591864,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4953869058778579, "count": 1, "min": 0.4953869058778579, "max":
0.4953869058778579}}}
#metrics {"StartTime": 1729802475.1592164, "EndTime": 1729802475.1592252,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4806245070287614, "count": 1, "min": 0.4806245070287614, "max":
0.4806245070287614}}}
#metrics {"StartTime": 1729802475.1592562, "EndTime": 1729802475.1592662,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48165574961613145, "count": 1, "min": 0.48165574961613145, "max":
0.48165574961613145}}}
#metrics {"StartTime": 1729802475.1592944, "EndTime": 1729802475.159303,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48064387487549887, "count": 1, "min": 0.48064387487549887, "max":
0.48064387487549887}}}
#metrics {"StartTime": 1729802475.1593335, "EndTime": 1729802475.1593423,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48165572050894573, "count": 1, "min": 0.48165572050894573, "max":
0.48165572050894573}}}

#metrics {"StartTime": 1729802475.159377, "EndTime": 1729802475.159387,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48074344719929046, "count": 1, "min": 0.48074344719929046, "max":
0.48074344719929046}}}
#metrics {"StartTime": 1729802475.1594167, "EndTime": 1729802475.1594253,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49457289581497543, "count": 1, "min": 0.49457289581497543, "max":
0.49457289581497543}}}
#metrics {"StartTime": 1729802475.1594553, "EndTime": 1729802475.1594644,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.480743447385875, "count": 1, "min": 0.480743447385875, "max":
0.480743447385875}}}
#metrics {"StartTime": 1729802475.1594968, "EndTime": 1729802475.1595058,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4945729090624766, "count": 1, "min": 0.4945729090624766, "max":
0.4945729090624766}}}
#metrics {"StartTime": 1729802475.1595428, "EndTime": 1729802475.1595533,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044436882488816, "count": 1, "min": 0.5044436882488816, "max":
0.5044436882488816}}}

#metrics {"StartTime": 1729802475.159585, "EndTime": 1729802475.1595945,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052271702849058, "count": 1, "min": 0.5052271702849058, "max":
0.5052271702849058}}}
#metrics {"StartTime": 1729802475.1596224, "EndTime": 1729802475.159631,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044104220941368, "count": 1, "min": 0.5044104220941368, "max":
0.5044104220941368}}}
#metrics {"StartTime": 1729802475.1596613, "EndTime": 1729802475.1596718,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5052272857807261, "count": 1, "min": 0.5052272857807261, "max":
0.5052272857807261}}}
#metrics {"StartTime": 1729802475.159706, "EndTime": 1729802475.159716,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044292324263172, "count": 1, "min": 0.5044292324263172, "max":
0.5044292324263172}}}
#metrics {"StartTime": 1729802475.1597483, "EndTime": 1729802475.1597576,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5063130282148162, "count": 1, "min": 0.5063130282148162, "max":
0.5063130282148162}}}

#metrics {"StartTime": 1729802475.159787, "EndTime": 1729802475.159797,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044292451140648, "count": 1, "min": 0.5044292451140648, "max":
0.5044292451140648}}}
#metrics {"StartTime": 1729802475.1598299, "EndTime": 1729802475.1598392,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.506313003585659, "count": 1, "min": 0.506313003585659, "max":
0.506313003585659}}}
#metrics {"StartTime": 1729802475.1598752, "EndTime": 1729802475.1598845,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137189918836986, "count": 1, "min": 0.5137189918836986, "max":
0.5137189918836986}}}
#metrics {"StartTime": 1729802475.1599152, "EndTime": 1729802475.1599238,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5143688237985079, "count": 1, "min": 0.5143688237985079, "max":
0.5143688237985079}}}
#metrics {"StartTime": 1729802475.15995, "EndTime": 1729802475.159959,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137181759495761, "count": 1, "min": 0.5137181759495761, "max":
0.5137181759495761}}}

#metrics {"StartTime": 1729802475.1599874, "EndTime": 1729802475.1599965,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5143802711322109, "count": 1, "min": 0.5143802711322109, "max":
0.5143802711322109}}}
#metrics {"StartTime": 1729802475.160031, "EndTime": 1729802475.1600413,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137479915967227, "count": 1, "min": 0.5137479915967227, "max":
0.5137479915967227}}}
#metrics {"StartTime": 1729802475.1600702, "EndTime": 1729802475.1600797,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5140466835657713, "count": 1, "min": 0.5140466835657713, "max":
0.5140466835657713}}}
#metrics {"StartTime": 1729802475.1601114, "EndTime": 1729802475.16012,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137520731331804, "count": 1, "min": 0.5137520731331804, "max":
0.5137520731331804}}}
#metrics {"StartTime": 1729802475.1601503, "EndTime": 1729802475.16016,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.514046681326757, "count": 1, "min": 0.514046681326757, "max":
0.514046681326757}}}
[10/24/2024 20:41:15 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=3, validation binary_classification_cross_entropy_objective
<loss>=0.48036658583058855

```
[10/24/2024 20:41:15 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=3, criteria=binary_classification_cross_entropy_objective,
value=0.48036658583058855
[10/24/2024 20:41:15 INFO 139943714142016] Saving model for epoch: 3
[10/24/2024 20:41:15 INFO 139943714142016] Saved checkpoint to
"/tmp/tmpcceh8bf0/mx-mod-0000.params"
[10/24/2024 20:41:15 INFO 139943714142016] #progress_metric: host=algo-1,
completed 26.666666666666668 % of epochs
#metrics {"StartTime": 1729802440.8422947, "EndTime": 1729802475.165963,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 5245888.0, "count": 1, "min": 5245888, "max": 5245888},
"Total Batches Seen": {"sum": 5248.0, "count": 1, "min": 5248, "max": 5248},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 6.0, "count": 1,
"min": 6, "max": 6}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:41:15 INFO 139943714142016] #throughput_metric: host=algo-1,
train throughput=38121.38877980881 records/second
[2024-10-24 20:41:46.378] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 11, "duration": 31211, "num_examples":
1309, "num_bytes": 507687136}
#metrics {"StartTime": 1729802506.3788419, "EndTime": 1729802506.3789132,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 0}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4806276846136157, "count": 1, "min": 0.4806276846136157, "max":
0.4806276846136157}}}
```

#metrics {"StartTime": 1729802506.3789947, "EndTime": 1729802506.3790078,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 1}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4814516556766055, "count": 1, "min": 0.4814516556766055, "max":
0.4814516556766055}}}
#metrics {"StartTime": 1729802506.3790402, "EndTime": 1729802506.3790476,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 2}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4806671591009204, "count": 1, "min": 0.4806671591009204, "max":
0.4806671591009204}}}
#metrics {"StartTime": 1729802506.3790832, "EndTime": 1729802506.379093,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 3}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48145165346011476, "count": 1, "min": 0.48145165346011476, "max":
0.48145165346011476}}}
#metrics {"StartTime": 1729802506.379128, "EndTime": 1729802506.3791392,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 4}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4807435938983882, "count": 1, "min": 0.4807435938983882, "max":
0.4807435938983882}}}
#metrics {"StartTime": 1729802506.3791704, "EndTime": 1729802506.379181,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 5}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49489287907778307, "count": 1, "min": 0.49489287907778307, "max":
0.49489287907778307}}}

#metrics {"StartTime": 1729802506.37921, "EndTime": 1729802506.3792312,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48074359471499006, "count": 1, "min": 0.48074359471499006, "max":
0.48074359471499006}}}
#metrics {"StartTime": 1729802506.379263, "EndTime": 1729802506.379273,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 7}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49489287917110897, "count": 1, "min": 0.49489287917110897, "max":
0.49489287917110897}}}
#metrics {"StartTime": 1729802506.3793042, "EndTime": 1729802506.3793147,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 8}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48089692466339207, "count": 1, "min": 0.48089692466339207, "max":
0.48089692466339207}}}
#metrics {"StartTime": 1729802506.3793428, "EndTime": 1729802506.3793511,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 9}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.481672977482507, "count": 1, "min": 0.481672977482507, "max":
0.481672977482507}}}
#metrics {"StartTime": 1729802506.379377, "EndTime": 1729802506.3793848,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 10}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4809352801961636, "count": 1, "min": 0.4809352801961636, "max":
0.4809352801961636}}}

#metrics {"StartTime": 1729802506.3794122, "EndTime": 1729802506.3794222,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 11}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48167297689921995, "count": 1, "min": 0.48167297689921995, "max":
0.48167297689921995}}}
#metrics {"StartTime": 1729802506.3794553, "EndTime": 1729802506.3794646,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 12}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.48100722230683773, "count": 1, "min": 0.48100722230683773, "max":
0.48100722230683773}}}
#metrics {"StartTime": 1729802506.3794959, "EndTime": 1729802506.3795044,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 13}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.49414732666482253, "count": 1, "min": 0.49414732666482253, "max":
0.49414732666482253}}}
#metrics {"StartTime": 1729802506.3795311, "EndTime": 1729802506.3795395,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 14}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4810072222368433, "count": 1, "min": 0.4810072222368433, "max":
0.4810072222368433}}}
#metrics {"StartTime": 1729802506.3795688, "EndTime": 1729802506.379578,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 15}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.4941473181488317, "count": 1, "min": 0.4941473181488317, "max":
0.4941473181488317}}}

#metrics {"StartTime": 1729802506.379608, "EndTime": 1729802506.3796177,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 16}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045066372932644, "count": 1, "min": 0.5045066372932644, "max":
0.5045066372932644}}}
#metrics {"StartTime": 1729802506.3796504, "EndTime": 1729802506.3796606,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 17}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047051590619102, "count": 1, "min": 0.5047051590619102, "max":
0.5047051590619102}}}
#metrics {"StartTime": 1729802506.3796906, "EndTime": 1729802506.3797004,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 18}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045163239645302, "count": 1, "min": 0.5045163239645302, "max":
0.5045163239645302}}}
#metrics {"StartTime": 1729802506.3797352, "EndTime": 1729802506.3797445,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 19}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5047051571020658, "count": 1, "min": 0.5047051571020658, "max":
0.5047051571020658}}}
#metrics {"StartTime": 1729802506.379779, "EndTime": 1729802506.379789,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 20}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045334245640932, "count": 1, "min": 0.5045334245640932, "max":
0.5045334245640932}}}

#metrics {"StartTime": 1729802506.3798163, "EndTime": 1729802506.379825,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 21}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5070597958812655, "count": 1, "min": 0.5070597958812655, "max":
0.5070597958812655}}}
#metrics {"StartTime": 1729802506.3798556, "EndTime": 1729802506.379866,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 22}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5045334251940433, "count": 1, "min": 0.5045334251940433, "max":
0.5045334251940433}}}
#metrics {"StartTime": 1729802506.3799, "EndTime": 1729802506.3799102,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 23}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5070597971878285, "count": 1, "min": 0.5070597971878285, "max":
0.5070597971878285}}}
#metrics {"StartTime": 1729802506.3799438, "EndTime": 1729802506.379953,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 24}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137247973567485, "count": 1, "min": 0.5137247973567485, "max":
0.5137247973567485}}}
#metrics {"StartTime": 1729802506.3799846, "EndTime": 1729802506.3799942,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 25}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138142591937237, "count": 1, "min": 0.5138142591937237, "max":
0.5138142591937237}}}

#metrics {"StartTime": 1729802506.380025, "EndTime": 1729802506.3800337,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 26}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137287083897022, "count": 1, "min": 0.5137287083897022, "max":
0.5137287083897022}}}
#metrics {"StartTime": 1729802506.3800657, "EndTime": 1729802506.3800752,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 27}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5138147546377386, "count": 1, "min": 0.5138147546377386, "max":
0.5138147546377386}}}
#metrics {"StartTime": 1729802506.3801236, "EndTime": 1729802506.3801332,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 28}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137340838464393, "count": 1, "min": 0.5137340838464393, "max":
0.5137340838464393}}}
#metrics {"StartTime": 1729802506.380155, "EndTime": 1729802506.3801608,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 29}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5146824915474708, "count": 1, "min": 0.5146824915474708, "max":
0.5146824915474708}}}
#metrics {"StartTime": 1729802506.3801792, "EndTime": 1729802506.3801847,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 30}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5137340802767226, "count": 1, "min": 0.5137340802767226, "max":
0.5137340802767226}}}

#metrics {"StartTime": 1729802506.380204, "EndTime": 1729802506.3802097,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics":
{"train_binary_classification_cross_entropy_objective": {"sum":
0.5146824918041171, "count": 1, "min": 0.5146824918041171, "max":
0.5146824918041171}}}
[10/24/2024 20:41:46 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=4, train binary_classification_cross_entropy_objective
<loss>=0.4806276846136157
[2024-10-24 20:41:49.122] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 14, "duration": 2726, "num_examples":
164, "num_bytes": 63460892}
#metrics {"StartTime": 1729802509.1280966, "EndTime": 1729802509.1281528,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 0}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4803632773138129, "count": 1, "min": 0.4803632773138129, "max":
0.4803632773138129}}}
#metrics {"StartTime": 1729802509.1282213, "EndTime": 1729802509.1282315,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 1}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48132407640690583, "count": 1, "min": 0.48132407640690583, "max":
0.48132407640690583}}}
#metrics {"StartTime": 1729802509.1282692, "EndTime": 1729802509.1282773,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 2}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48038945251031595, "count": 1, "min": 0.48038945251031595, "max":
0.48038945251031595}}}

#metrics {"StartTime": 1729802509.1283123, "EndTime": 1729802509.1283197,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 3}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.481324058308207, "count": 1, "min": 0.481324058308207, "max":
0.481324058308207}}}
#metrics {"StartTime": 1729802509.1283534, "EndTime": 1729802509.1283624,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 4}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.480411312379953, "count": 1, "min": 0.480411312379953, "max":
0.480411312379953}}}
#metrics {"StartTime": 1729802509.1283922, "EndTime": 1729802509.1284006,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 5}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49435266450402165, "count": 1, "min": 0.49435266450402165, "max":
0.49435266450402165}}}
#metrics {"StartTime": 1729802509.1284308, "EndTime": 1729802509.12844,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48041131555188993, "count": 1, "min": 0.48041131555188993, "max":
0.48041131555188993}}}
#metrics {"StartTime": 1729802509.1284602, "EndTime": 1729802509.1284685,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 7}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4943526756990931, "count": 1, "min": 0.4943526756990931, "max":
0.4943526756990931}}}

#metrics {"StartTime": 1729802509.1285043, "EndTime": 1729802509.1285136,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 8}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4806219232062761, "count": 1, "min": 0.4806219232062761, "max":
0.4806219232062761}}}
#metrics {"StartTime": 1729802509.1285458, "EndTime": 1729802509.1285546,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 9}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48148810174225404, "count": 1, "min": 0.48148810174225404, "max":
0.48148810174225404}}}
#metrics {"StartTime": 1729802509.1285903, "EndTime": 1729802509.1285996,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 10}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48064399317008694, "count": 1, "min": 0.48064399317008694, "max":
0.48064399317008694}}}
#metrics {"StartTime": 1729802509.1286392, "EndTime": 1729802509.1286488,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 11}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48148808849475283, "count": 1, "min": 0.48148808849475283, "max":
0.48148808849475283}}}
#metrics {"StartTime": 1729802509.1286857, "EndTime": 1729802509.1286955,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 12}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48066221408175674, "count": 1, "min": 0.48066221408175674, "max":
0.48066221408175674}}}

#metrics {"StartTime": 1729802509.12875, "EndTime": 1729802509.128761,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 13}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.49379579310381927, "count": 1, "min": 0.49379579310381927, "max":
0.49379579310381927}}}
#metrics {"StartTime": 1729802509.1287942, "EndTime": 1729802509.1288013,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 14}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.48066221277566507, "count": 1, "min": 0.48066221277566507, "max":
0.48066221277566507}}}
#metrics {"StartTime": 1729802509.128828, "EndTime": 1729802509.128836,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 15}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.4937957940367419, "count": 1, "min": 0.4937957940367419, "max":
0.4937957940367419}}}
#metrics {"StartTime": 1729802509.1288679, "EndTime": 1729802509.1288857,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 16}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5044518425523243, "count": 1, "min": 0.5044518425523243, "max":
0.5044518425523243}}}
#metrics {"StartTime": 1729802509.1289237, "EndTime": 1729802509.1289332,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 17}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5051117509643405, "count": 1, "min": 0.5051117509643405, "max":
0.5051117509643405}}}

#metrics {"StartTime": 1729802509.1289685, "EndTime": 1729802509.128978,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 18}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.504410447469632, "count": 1, "min": 0.504410447469632, "max":
0.504410447469632}}}
#metrics {"StartTime": 1729802509.129014, "EndTime": 1729802509.129023,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 19}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5051117929458583, "count": 1, "min": 0.5051117929458583, "max":
0.5051117929458583}}}
#metrics {"StartTime": 1729802509.1290581, "EndTime": 1729802509.1290681,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 20}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5043945693132429, "count": 1, "min": 0.5043945693132429, "max":
0.5043945693132429}}}
#metrics {"StartTime": 1729802509.1291075, "EndTime": 1729802509.1291173,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 21}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5058854503737829, "count": 1, "min": 0.5058854503737829, "max":
0.5058854503737829}}}
#metrics {"StartTime": 1729802509.1291518, "EndTime": 1729802509.1291628,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 22}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5043945721120107, "count": 1, "min": 0.5043945721120107, "max":
0.5043945721120107}}}

#metrics {"StartTime": 1729802509.1292026, "EndTime": 1729802509.1292129,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 23}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5058854524262126, "count": 1, "min": 0.5058854524262126, "max":
0.5058854524262126}}}
#metrics {"StartTime": 1729802509.1292527, "EndTime": 1729802509.1292634,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 24}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137181966604583, "count": 1, "min": 0.5137181966604583, "max":
0.5137181966604583}}}
#metrics {"StartTime": 1729802509.1293015, "EndTime": 1729802509.1293128,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 25}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5142359390470096, "count": 1, "min": 0.5142359390470096, "max":
0.5142359390470096}}}
#metrics {"StartTime": 1729802509.1293535, "EndTime": 1729802509.1293633,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 26}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.51371890904017, "count": 1, "min": 0.51371890904017, "max":
0.51371890904017}}}
#metrics {"StartTime": 1729802509.1294003, "EndTime": 1729802509.12941,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 27}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5142383680043402, "count": 1, "min": 0.5142383680043402, "max":
0.5142383680043402}}}

```
#metrics {"StartTime": 1729802509.1294444, "EndTime": 1729802509.1294537,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 28}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137198395371901, "count": 1, "min": 0.5137198395371901, "max":
0.5137198395371901}}}
#metrics {"StartTime": 1729802509.1294894, "EndTime": 1729802509.1294987,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 29}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5140488983240685, "count": 1, "min": 0.5140488983240685, "max":
0.5140488983240685}}}
#metrics {"StartTime": 1729802509.1295373, "EndTime": 1729802509.129548,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 30}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5137203832444924, "count": 1, "min": 0.5137203832444924, "max":
0.5137203832444924}}}
#metrics {"StartTime": 1729802509.1295865, "EndTime": 1729802509.1295967,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics":
{"validation_binary_classification_cross_entropy_objective": {"sum":
0.5140488979508995, "count": 1, "min": 0.5140488979508995, "max":
0.5140488979508995}}}
[10/24/2024 20:41:49 INFO 139943714142016] #quality_metric: host=algo-1,
epoch=4, validation binary_classification_cross_entropy_objective
<loss>=0.4803632773138129
[10/24/2024 20:41:49 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objective,
value=0.4803632773138129
[10/24/2024 20:41:49 INFO 139943714142016] Saving model for epoch: 4
[10/24/2024 20:41:49 INFO 139943714142016] Saved checkpoint to
"/tmp/tmpyrhbi_vd/mx-mod-0000.params"
```

```
[10/24/2024 20:41:49 INFO 139943714142016] Early stop condition met.
Stopping training.
[10/24/2024 20:41:49 INFO 139943714142016] #progress_metric: host=algo-1,
completed 100 % epochs
#metrics {"StartTime": 1729802475.1673908, "EndTime": 1729802509.135232,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "Meta": "training_data_iter"}, "Metrics": {"Total
Records Seen": {"sum": 6554360.0, "count": 1, "min": 6554360, "max": 6554360},
"Total Batches Seen": {"sum": 6557.0, "count": 1, "min": 6557, "max": 6557},
"Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "min":
1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 1309.0,
"count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 7.0, "count": 1,
"min": 7, "max": 7}, "Number of Records Since Last Reset": {"sum": 1308472.0,
"count": 1, "min": 1308472, "max": 1308472}, "Number of Batches Since Last
Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 1309}}}
[10/24/2024 20:41:49 INFO 139943714142016] #throughput_metric: host=algo-1,
train throughput=38520.763314509466 records/second
[10/24/2024 20:41:49 WARNING 139943714142016] wait_for_all_workers will not
sync workers since the kv store is not running distributed
[10/24/2024 20:41:49 WARNING 139943714142016] wait_for_all_workers will not
sync workers since the kv store is not running distributed
[2024-10-24 20:41:51.872] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 17, "duration": 2724, "num_examples":
164, "num_bytes": 63460892}
[10/24/2024 20:41:51 INFO 139943714142016] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objective,
value=0.4803632773138129
[2024-10-24 20:41:52.368] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 19, "duration": 486, "num_examples":
164, "num_bytes": 63460892}
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('binary_classification_cross_entropy_objective', 0.48069846838764985)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('binary_classification_accuracy', 0.7930227012882202)
```

```
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('binary_f_1.000', 0.1058136770648987)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('precision', 0.5680657969370392)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('recall', 0.05834037223662366)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('roc_auc_score', 0.6772462277888586)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('binary_balanced_accuracy', 0.5)
[10/24/2024 20:41:52 INFO 139943714142016] #validation_score (algo-1) :
('binary_log_loss', 0.749005144594601)
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation binary_classification_cross_entropy_objective
<loss>=0.48069846838764985
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation binary_classification_accuracy <score>=0.7930227012882202
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation binary_f_1.000 <score>=0.1058136770648987
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation precision <score>=0.5680657969370392
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation recall <score>=0.05834037223662366
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation roc_auc_score <score>=0.6772462277888586
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation binary_balanced_accuracy <score>=0.5
[10/24/2024 20:41:52 INFO 139943714142016] #quality_metric: host=algo-1,
validation binary_log_loss <score>=0.749005144594601
[10/24/2024 20:41:52 INFO 139943714142016] Best model found for
hyperparameters: {"optimizer": "adam", "learning_rate": 0.005, "l1": 0.0, "wd":
0.0001, "lr_scheduler_step": 10, "lr_scheduler_factor": 0.99,
"lr_scheduler_minimum_lr": 1e-05}
[10/24/2024 20:41:52 INFO 139943714142016] Saved checkpoint to
"/tmp/tmpv19qgb0r/mx-mod-0000.params"
```

```
[2024-10-24 20:41:52.621] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 0, "duration": 176062, "num_examples": 1,
"num_bytes": 388000}
[2024-10-24 20:41:53.164] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 1, "duration": 542, "num_examples": 164,
"num_bytes": 63460892}
#metrics {"StartTime": 1729802512.6184742, "EndTime": 1729802513.4041226,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "Meta": "test_data_iter"}, "Metrics": {"Total Records Seen": {"sum":
163559.0, "count": 1, "min": 163559, "max": 163559}, "Total Batches Seen":
{"sum": 164.0, "count": 1, "min": 164, "max": 164}, "Max Records Seen Between
Resets": {"sum": 163559.0, "count": 1, "min": 163559, "max": 163559}, "Max
Batches Seen Between Resets": {"sum": 164.0, "count": 1, "min": 164, "max":
164}, "Reset Count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}, "Number of
Records Since Last Reset": {"sum": 163559.0, "count": 1, "min": 163559, "max":
163559}, "Number of Batches Since Last Reset": {"sum": 164.0, "count": 1, "min":
164, "max": 164}}}
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('binary_classification_cross_entropy_objective', 0.4802489545504938)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('binary_classification_accuracy', 0.792784255222886)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('binary_f_1.000', 0.10636502663080737)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('precision', 0.561369329251322)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('recall', 0.05874814318585617)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('roc_auc_score', 0.678663792860472)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('binary_balanced_accuracy', 0.5)
[10/24/2024 20:41:53 INFO 139943714142016] #test_score (algo-1) :
('binary_log_loss', 0.7490597798139039)
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test binary_classification_cross_entropy_objective <loss>=0.4802489545504938
```

```
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test binary_classification_accuracy <score>=0.792784255222886
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test binary_f_1.000 <score>=0.10636502663080737
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test precision <score>=0.561369329251322
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test recall <score>=0.05874814318585617
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test roc_auc_score <score>=0.678663792860472
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test binary_balanced_accuracy <score>=0.5
[10/24/2024 20:41:53 INFO 139943714142016] #quality_metric: host=algo-1,
test binary_log_loss <score>=0.7490597798139039
#metrics {"StartTime": 1729802336.551319, "EndTime": 1729802513.4119906,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training"}, "Metrics": {"initialize.time": {"sum": 275.6307125091553, "count":
1, "min": 275.6307125091553, "max": 275.6307125091553}, "epochs": {"sum": 15.0,
"count": 1, "min": 15, "max": 15}, "check_early_stopping.time": {"sum":
1.7695426940917969, "count": 6, "min": 0.1811981201171875, "max":
0.6022453308105469}, "update.time": {"sum": 172249.92084503174, "count": 5,
"min": 33965.08169174194, "max": 35097.784996032715}, "finalize.time": {"sum":
3480.1745414733887, "count": 1, "min": 3480.1745414733887, "max":
3480.1745414733887}, "setuptime": {"sum": 2.5224685668945312, "count": 1, "min":
2.5224685668945312, "max": 2.5224685668945312}, "totaltime": {"sum":
176973.85168075562, "count": 1, "min": 176973.85168075562, "max":
176973.85168075562}}}

2024-10-24 20:42:00 Uploading - Uploading generated training model
2024-10-24 20:42:14 Completed - Training job completed
Training seconds: 330
Billable seconds: 330
```

Perform a batch prediction by using the newly trained model.

[69]:
```
# Enter your code here
test_labels, target_predicted = batch_linear_predict(test,␣
↪classifier_estimator2)
```

```
INFO:sagemaker.image_uris:Same images used for training and inference.
Defaulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating model with name: linear-learner-2024-10-24-20-44-51-564
INFO:sagemaker:Creating transform job with name: linear-
learner-2024-10-24-20-44-52-321

…
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[10/24/2024 20:52:24 INFO 140076527445824] Memory profiler is not enabled

by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:

"is" with a literal. Did you mean "=="?

  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:

SyntaxWarning: "is" with a literal. Did you mean "=="?

  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:

SyntaxWarning: "is not" with a literal. Did you mean "!="?

  if len(self.X_min) is not 0:
[10/24/2024 20:52:28 WARNING 140076527445824] Loggers have already been

setup.
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class

algorithm.serve.server_config:config_api
[10/24/2024 20:52:28 INFO 140076527445824] loading entry points
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator

application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator

application/jsonlines
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator

application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator

text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder

application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder

application/jsonlines
```

```
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class
algorithm:model
[10/24/2024 20:52:28 INFO 140076527445824] Number of server workers: 4
[10/24/2024 20:52:28 INFO 140076527445824] loading model…
[10/24/2024 20:52:28 INFO 140076527445824] …model loaded.
[2024-10-24 20:52:28 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:52:28 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-10-24 20:52:28 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:52:28 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:52:28 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:52:28 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:52:28 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803149.2132652,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803151.9577227,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
113.87348175048828, "count": 1, "min": 113.87348175048828, "max":
113.87348175048828}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729803149.2134151, "EndTime": 1729803152.033821,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
118.5147762298584, "count": 1, "min": 118.5147762298584, "max":
118.5147762298584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
```

#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1297357,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
120.81193923950195, "count": 1, "min": 120.81193923950195, "max":
120.81193923950195}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1386588,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
121.00386619567871, "count": 1, "min": 121.00386619567871, "max":
121.00386619567871}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729803151.9579012, "EndTime": 1729803152.805765,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
63.27247619628906, "count": 1, "min": 63.27247619628906, "max":
63.27247619628906}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
2024-10-24T20:52:29.224:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[10/24/2024 20:52:24 INFO 140076527445824] Memory profiler is not enabled
by the environment variable ENABLE_PROFILER.
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[10/24/2024 20:52:24 INFO 140076527445824] Memory profiler is not enabled
by the environment variable ENABLE_PROFILER.
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:
/opt/amazon/lib/python3.8/site-packages/mxnet/model.py:97: SyntaxWarning:
"is" with a literal. Did you mean "=="?
  if num_device is 1 and 'dist' not in kvstore:

```
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:495:
SyntaxWarning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743:
SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[10/24/2024 20:52:28 WARNING 140076527445824] Loggers have already been
setup.
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:52:28 INFO 140076527445824] loading entry points
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/jsonlines
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/jsonlines
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class
algorithm:model
[10/24/2024 20:52:28 INFO 140076527445824] Number of server workers: 4
[10/24/2024 20:52:28 INFO 140076527445824] loading model…
```

```
[10/24/2024 20:52:28 WARNING 140076527445824] Loggers have already been
setup.
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class
algorithm.serve.server_config:config_api
[10/24/2024 20:52:28 INFO 140076527445824] loading entry points
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/jsonlines
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded request iterator
text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/json
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/jsonlines
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
application/x-recordio-protobuf
[10/24/2024 20:52:28 INFO 140076527445824] loaded response encoder
text/csv
[10/24/2024 20:52:28 INFO 140076527445824] loaded entry point class
algorithm:model
[10/24/2024 20:52:28 INFO 140076527445824] Number of server workers: 4
[10/24/2024 20:52:28 INFO 140076527445824] loading model…
[10/24/2024 20:52:28 INFO 140076527445824] …model loaded.
[2024-10-24 20:52:28 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:52:28 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-10-24 20:52:28 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:52:28 +0000] [43] [INFO] Booting worker with pid: 43
[2024-10-24 20:52:28 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:52:28 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:52:28 +0000] [70] [INFO] Booting worker with pid: 70
[10/24/2024 20:52:28 INFO 140076527445824] …model loaded.
[2024-10-24 20:52:28 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-10-24 20:52:28 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-10-24 20:52:28 +0000] [1] [INFO] Using worker: sync
[2024-10-24 20:52:28 +0000] [43] [INFO] Booting worker with pid: 43
```

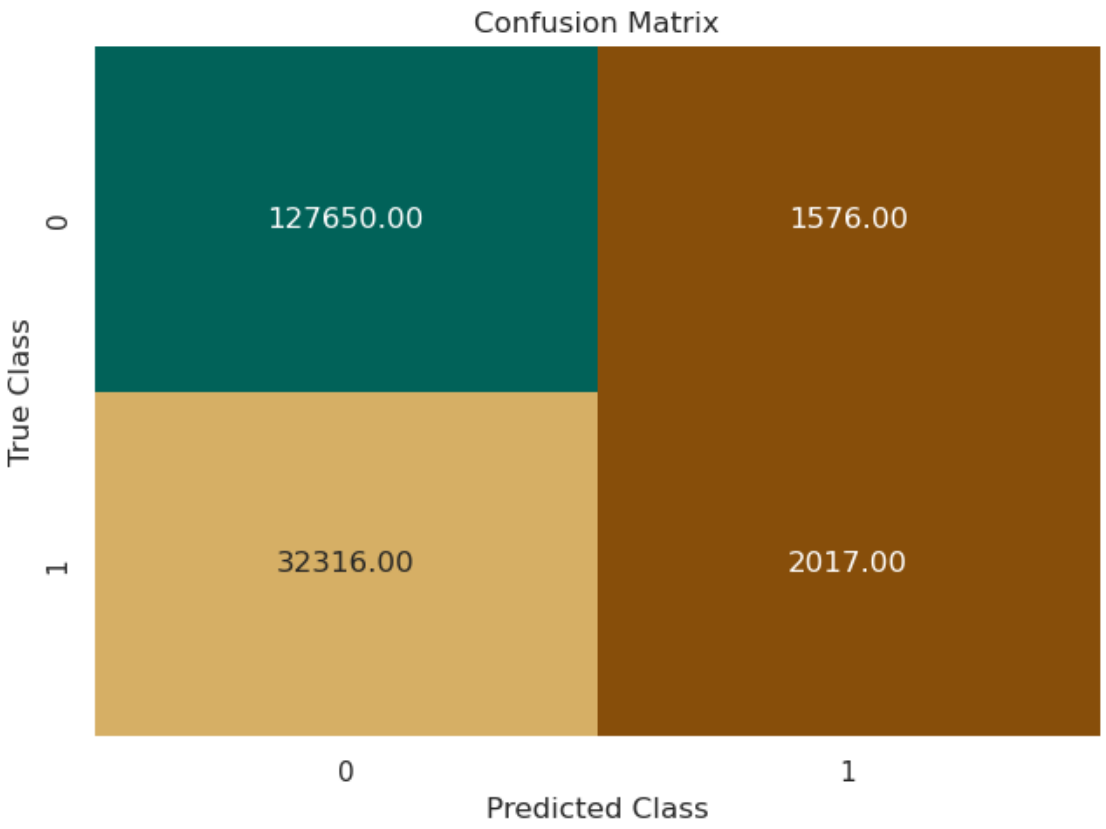
```
[2024-10-24 20:52:28 +0000] [52] [INFO] Booting worker with pid: 52
[2024-10-24 20:52:28 +0000] [61] [INFO] Booting worker with pid: 61
[2024-10-24 20:52:28 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803149.2132652,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803149.2132652,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0,
"count": 1, "min": 1, "max": 1}}}
2024-10-24T20:52:29.224:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803151.9577227,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
113.87348175048828, "count": 1, "min": 113.87348175048828, "max":
113.87348175048828}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729803149.2134151, "EndTime": 1729803152.033821,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
118.5147762298584, "count": 1, "min": 118.5147762298584, "max":
118.5147762298584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1297357,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN",
"Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum":
120.81193923950195, "count": 1, "min": 120.81193923950195, "max":
120.81193923950195}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1,
"max": 1}}}
```

#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1386588, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 121.00386619567871, "count": 1, "min": 121.00386619567871, "max": 121.00386619567871}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803151.9577227, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 113.87348175048828, "count": 1, "min": 113.87348175048828, "max": 113.87348175048828}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803149.2134151, "EndTime": 1729803152.033821, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 118.5147762298584, "count": 1, "min": 118.5147762298584, "max": 118.5147762298584}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1297357, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 120.81193923950195, "count": 1, "min": 120.81193923950195, "max": 120.81193923950195}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803148.6200476, "EndTime": 1729803152.1386588, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 121.00386619567871, "count": 1, "min": 121.00386619567871, "max": 121.00386619567871}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}

#metrics {"StartTime": 1729803151.9579012, "EndTime": 1729803152.805765, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 63.27247619628906, "count": 1, "min": 63.27247619628906, "max": 63.27247619628906}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1729803151.9579012, "EndTime": 1729803152.805765, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 63.27247619628906, "count": 1, "min": 63.27247619628906, "max": 63.27247619628906}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}

Plot a confusion matrix.

[70]:
```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



The linear model shows only a little improvement in performance. Try a tree-based ensemble model, which is called *XGBoost*, with Amazon SageMaker.

6.0.7 Try the XGBoost model

Perform these steps:

1. Use the training set variables and save them as CSV files: train.csv, validation.csv and test.csv.
2. Store the bucket name in the variable. The Amazon S3 bucket name is provided to the left of the lab instructions.

   a. `bucket = <LabBucketName>`

   b. `prefix = 'flight-xgb'`

3. Use the AWS SDK for Python (Boto3) to upload the model to the bucket.

[72]:
```
bucket='c134412a340974518003998t1w11504374297-flightbucket-zyuwb8c4zacf'
prefix='flight-xgb'
train_file='flight_train.csv'
test_file='flight_test.csv'
validate_file='flight_validate.csv'
whole_file='flight.csv'
s3_resource = boto3.Session().resource('s3')

def upload_s3_csv(filename, folder, dataframe):
    csv_buffer = io.StringIO()
    dataframe.to_csv(csv_buffer, header=False, index=False )
    s3_resource.Bucket(bucket).Object(os.path.join(prefix, folder, filename)).
 ↪put(Body=csv_buffer.getvalue())

upload_s3_csv(train_file, 'train', train)
upload_s3_csv(test_file, 'test', test)
upload_s3_csv(validate_file, 'validate', validate)
```

```
INFO:botocore.credentials:Found credentials from IAM Role:
BaseNotebookInstanceEc2InstanceRole
```

Use the `sagemaker.inputs.TrainingInput` function to create a `record_set` for the training and validation datasets.

[73]:
```
train_channel = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/train/".format(bucket,prefix,train_file),
    content_type='text/csv')

validate_channel = sagemaker.inputs.TrainingInput(
```

```
    "s3://{}/{}/validate/".format(bucket,prefix,validate_file),
    content_type='text/csv')

data_channels = {'train': train_channel, 'validation': validate_channel}
```

[74]:
```
from sagemaker.image_uris import retrieve
container = retrieve('xgboost',boto3.Session().region_name,'1.0-1')
```

```
INFO:sagemaker.image_uris:Defaulting to only available Python version: py3
INFO:sagemaker.image_uris:Defaulting to only supported image scope: cpu.
```

[75]:
```
sess = sagemaker.Session()
s3_output_location="s3://{}/{}/output/".format(bucket,prefix)

xgb = sagemaker.estimator.Estimator(container,
                                    role = sagemaker.get_execution_role(),
                                    instance_count=1,
                                    instance_type=instance_type,
                                    output_path=s3_output_location,
                                    sagemaker_session=sess)
xgb.set_hyperparameters(max_depth=5,
                        eta=0.2,
                        gamma=4,
                        min_child_weight=6,
                        subsample=0.8,
                        silent=0,
                        objective='binary:logistic',
                        eval_metric = "auc",
                        num_round=100)

xgb.fit(inputs=data_channels)
```

```
INFO:sagemaker:Creating training-job with name: sagemaker-
xgboost-2024-10-24-20-57-30-475

2024-10-24 20:57:31 Starting - Starting the training job…
2024-10-24 20:57:45 Starting - Preparing the instances for training…
2024-10-24 20:58:17 Downloading - Downloading input data…
2024-10-24 20:58:52 Downloading - Downloading the training image…
2024-10-24 20:59:38 Training - Training image download completed. Training in
progress.[2024-10-24 20:59:54.091 ip-10-2-76-209.ec2.internal:7 INFO

utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
INFO:sagemaker-containers:Imported framework

sagemaker_xgboost_container.training
INFO:sagemaker-containers:Failed to parse hyperparameter eval_metric value

auc to Json.
Returning the value itself
```

```
INFO:sagemaker-containers:Failed to parse hyperparameter objective value
binary:logistic to Json.
Returning the value itself
INFO:sagemaker-containers:No GPUs detected (normal if no gpus
installed)
INFO:sagemaker_xgboost_container.training:Running XGBoost Sagemaker in
algorithm mode
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Single node training.
[21:00:00] 1308472x85 matrix with 111220120 entries loaded from
/opt/ml/input/data/train?format=csv&label_column=0&delimiter=,
[21:00:01] 163559x85 matrix with 13902515 entries loaded from
/opt/ml/input/data/validation?format=csv&label_column=0&delimiter=,
[2024-10-24 21:00:01.750 ip-10-2-76-209.ec2.internal:7 INFO
json_config.py:91] Creating hook from json_config at
/opt/ml/input/config/debughookconfig.json.
[2024-10-24 21:00:01.750 ip-10-2-76-209.ec2.internal:7 INFO hook.py:201]
tensorboard_dir has not been set for the hook. SMDebug will not be exporting
tensorboard summaries.
[2024-10-24 21:00:01.750 ip-10-2-76-209.ec2.internal:7 INFO
profiler_config_parser.py:102] User has disabled profiler.
[2024-10-24 21:00:01.751 ip-10-2-76-209.ec2.internal:7 INFO hook.py:255]
Saving to /opt/ml/output/tensors
[2024-10-24 21:00:01.751 ip-10-2-76-209.ec2.internal:7 INFO
state_store.py:77] The checkpoint config file
/opt/ml/input/config/checkpointconfig.json does not exist.
INFO:root:Debug hook created from config
INFO:root:Train matrix has 1308472 rows
INFO:root:Validation matrix has 163559 rows
[0]#011train-auc:0.65519#011validation-auc:0.65487
[2024-10-24 21:00:07.501 ip-10-2-76-209.ec2.internal:7 INFO hook.py:423]
Monitoring the collections: metrics
[2024-10-24 21:00:07.502 ip-10-2-76-209.ec2.internal:7 INFO hook.py:486]
Hook is writing from the hook with pid: 7
[1]#011train-auc:0.66716#011validation-auc:0.66658
[2]#011train-auc:0.66947#011validation-auc:0.66932
```

[3]#011train-auc:0.67440#011validation-auc:0.67494
[4]#011train-auc:0.67568#011validation-auc:0.67618
[5]#011train-auc:0.67836#011validation-auc:0.67870
[6]#011train-auc:0.67954#011validation-auc:0.67976
[7]#011train-auc:0.68164#011validation-auc:0.68161
[8]#011train-auc:0.68412#011validation-auc:0.68426
[9]#011train-auc:0.68632#011validation-auc:0.68620
[10]#011train-auc:0.68886#011validation-auc:0.68881
[11]#011train-auc:0.69070#011validation-auc:0.69066
[12]#011train-auc:0.69220#011validation-auc:0.69193
[13]#011train-auc:0.69430#011validation-auc:0.69414
[14]#011train-auc:0.69515#011validation-auc:0.69507
[15]#011train-auc:0.69654#011validation-auc:0.69657
[16]#011train-auc:0.69831#011validation-auc:0.69834
[17]#011train-auc:0.69949#011validation-auc:0.69945
[18]#011train-auc:0.70096#011validation-auc:0.70083
[19]#011train-auc:0.70219#011validation-auc:0.70186
[20]#011train-auc:0.70353#011validation-auc:0.70331
[21]#011train-auc:0.70445#011validation-auc:0.70419
[22]#011train-auc:0.70512#011validation-auc:0.70485
[23]#011train-auc:0.70608#011validation-auc:0.70583
[24]#011train-auc:0.70709#011validation-auc:0.70670
[25]#011train-auc:0.70816#011validation-auc:0.70761
[26]#011train-auc:0.70894#011validation-auc:0.70849
[27]#011train-auc:0.70932#011validation-auc:0.70883
[28]#011train-auc:0.71014#011validation-auc:0.70971
[29]#011train-auc:0.71060#011validation-auc:0.71005
[30]#011train-auc:0.71122#011validation-auc:0.71057
[31]#011train-auc:0.71169#011validation-auc:0.71095
[32]#011train-auc:0.71215#011validation-auc:0.71135
[33]#011train-auc:0.71310#011validation-auc:0.71237
[34]#011train-auc:0.71455#011validation-auc:0.71361
[35]#011train-auc:0.71521#011validation-auc:0.71428
[36]#011train-auc:0.71565#011validation-auc:0.71469
[37]#011train-auc:0.71644#011validation-auc:0.71528
[38]#011train-auc:0.71762#011validation-auc:0.71642
[39]#011train-auc:0.71811#011validation-auc:0.71690
[40]#011train-auc:0.71835#011validation-auc:0.71713
[41]#011train-auc:0.71867#011validation-auc:0.71750
[42]#011train-auc:0.71896#011validation-auc:0.71770
[43]#011train-auc:0.71920#011validation-auc:0.71788
[44]#011train-auc:0.71958#011validation-auc:0.71832
[45]#011train-auc:0.71985#011validation-auc:0.71856
[46]#011train-auc:0.72015#011validation-auc:0.71891
[47]#011train-auc:0.72062#011validation-auc:0.71935
[48]#011train-auc:0.72103#011validation-auc:0.71969
[49]#011train-auc:0.72169#011validation-auc:0.72033
[50]#011train-auc:0.72191#011validation-auc:0.72058

[51]#011train-auc:0.72238#011validation-auc:0.72110
[52]#011train-auc:0.72289#011validation-auc:0.72162
[53]#011train-auc:0.72323#011validation-auc:0.72194
[54]#011train-auc:0.72338#011validation-auc:0.72208
[55]#011train-auc:0.72390#011validation-auc:0.72263
[56]#011train-auc:0.72412#011validation-auc:0.72279
[57]#011train-auc:0.72438#011validation-auc:0.72299
[58]#011train-auc:0.72455#011validation-auc:0.72315
[59]#011train-auc:0.72470#011validation-auc:0.72332
[60]#011train-auc:0.72517#011validation-auc:0.72371
[61]#011train-auc:0.72535#011validation-auc:0.72393
[62]#011train-auc:0.72573#011validation-auc:0.72428
[63]#011train-auc:0.72598#011validation-auc:0.72448
[64]#011train-auc:0.72621#011validation-auc:0.72464
[65]#011train-auc:0.72636#011validation-auc:0.72483
[66]#011train-auc:0.72662#011validation-auc:0.72506
[67]#011train-auc:0.72698#011validation-auc:0.72535
[68]#011train-auc:0.72718#011validation-auc:0.72552
[69]#011train-auc:0.72749#011validation-auc:0.72582
[70]#011train-auc:0.72776#011validation-auc:0.72605
[71]#011train-auc:0.72793#011validation-auc:0.72621
[72]#011train-auc:0.72813#011validation-auc:0.72639
[73]#011train-auc:0.72838#011validation-auc:0.72659
[74]#011train-auc:0.72878#011validation-auc:0.72691
[75]#011train-auc:0.72900#011validation-auc:0.72710
[76]#011train-auc:0.72930#011validation-auc:0.72739
[77]#011train-auc:0.72949#011validation-auc:0.72751
[78]#011train-auc:0.72966#011validation-auc:0.72763
[79]#011train-auc:0.72977#011validation-auc:0.72769
[80]#011train-auc:0.72980#011validation-auc:0.72774
[81]#011train-auc:0.72999#011validation-auc:0.72795
[82]#011train-auc:0.73021#011validation-auc:0.72815
[83]#011train-auc:0.73039#011validation-auc:0.72833
[84]#011train-auc:0.73085#011validation-auc:0.72875
[85]#011train-auc:0.73101#011validation-auc:0.72889
[86]#011train-auc:0.73107#011validation-auc:0.72894
[87]#011train-auc:0.73135#011validation-auc:0.72916
[88]#011train-auc:0.73155#011validation-auc:0.72936
[89]#011train-auc:0.73180#011validation-auc:0.72955
[90]#011train-auc:0.73207#011validation-auc:0.72976
[91]#011train-auc:0.73216#011validation-auc:0.72988
[92]#011train-auc:0.73227#011validation-auc:0.72997
[93]#011train-auc:0.73251#011validation-auc:0.73018
[94]#011train-auc:0.73269#011validation-auc:0.73041
[95]#011train-auc:0.73290#011validation-auc:0.73059
[96]#011train-auc:0.73297#011validation-auc:0.73064

2024-10-24 21:04:04 Uploading - Uploading generated training

```
model[97]#011train-auc:0.73309#011validation-auc:0.73072
[98]#011train-auc:0.73324#011validation-auc:0.73076
[99]#011train-auc:0.73327#011validation-auc:0.73078

2024-10-24 21:04:18 Completed - Training job completed
Training seconds: 361
Billable seconds: 361
```

Use the batch transformer for your new model, and evaluate the model on the test dataset.

[76]:
```
batch_X = test.iloc[:,1:];
batch_X_file='batch-in.csv'
upload_s3_csv(batch_X_file, 'batch-in', batch_X)
```

[77]:
```
batch_output = "s3://{}/{}/batch-out/".format(bucket,prefix)
batch_input = "s3://{}/{}/batch-in/{}".format(bucket,prefix,batch_X_file)

xgb_transformer = xgb.transformer(instance_count=1,
                                  instance_type=instance_type,
                                  strategy='MultiRecord',
                                  assemble_with='Line',
                                  output_path=batch_output)

xgb_transformer.transform(data=batch_input,
                          data_type='S3Prefix',
                          content_type='text/csv',
                          split_type='Line')
xgb_transformer.wait()
```

```
INFO:sagemaker:Creating model with name: sagemaker-
xgboost-2024-10-24-21-06-11-203
INFO:sagemaker:Creating transform job with name: sagemaker-
xgboost-2024-10-24-21-06-11-909

…[2024-10-24:21:13:16:INFO] No

GPUs detected (normal if no gpus installed)
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:16:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {

  worker_connections 2048;
}
```

```
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
```

```
}
[2024-10-24 21:13:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 21:13:16 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 21:13:16 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 21:13:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 21:13:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 21:13:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 21:13:16 +0000] [29] [INFO] Booting worker with pid: 29
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 21:13:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 21:13:16 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 21:13:16 +0000] [19] [INFO] Using worker: gevent
```

```
[2024-10-24 21:13:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 21:13:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 21:13:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 21:13:16 +0000] [29] [INFO] Booting worker with pid: 29
[2024-10-24:21:13:21:INFO] No GPUs detected (normal if no gpus

installed)
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /ping HTTP/1.1" 200 0

"-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /execution-parameters

HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:21:13:21:INFO] No GPUs detected (normal if no gpus

installed)
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /ping HTTP/1.1" 200 0

"-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /execution-parameters

HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:23:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:23:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:23:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:21:13:23:INFO] Determined delimiter of CSV input is ','
2024-10-24T21:13:21.950:[sagemaker logs]: MaxConcurrentTransforms=4,

MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations

HTTP/1.1" 200 652570 "-" "Go-http-client/1.1"
```

```
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652651 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652596 "-" "Go-http-client/1.1"
[2024-10-24:21:13:26:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652602 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652570 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652651 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652596 "-" "Go-http-client/1.1"
[2024-10-24:21:13:26:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652602 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:29 +0000] "POST /invocations
HTTP/1.1" 200 602519 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:29 +0000] "POST /invocations
HTTP/1.1" 200 602519 "-" "Go-http-client/1.1"

[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:16:INFO] nginx config:
```

```
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 21:13:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 21:13:16 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 21:13:16 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 21:13:16 +0000] [26] [INFO] Booting worker with pid: 26
```

```
[2024-10-24 21:13:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 21:13:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 21:13:16 +0000] [29] [INFO] Booting worker with pid: 29
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 21:13:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 21:13:16 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 21:13:16 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 21:13:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 21:13:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 21:13:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 21:13:16 +0000] [29] [INFO] Booting worker with pid: 29
[2024-10-24:21:13:21:INFO] No GPUs detected (normal if no gpus
installed)
```

```
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:21:13:21:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:21 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:23:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:23:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:22:INFO] Determined delimiter of CSV input is ','
[2024-10-24:21:13:23:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:21:13:23:INFO] Determined delimiter of CSV input is ','
2024-10-24T21:13:21.950:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652570 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652651 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652596 "-" "Go-http-client/1.1"
[2024-10-24:21:13:26:INFO] Determined delimiter of CSV input is ','
```

```
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652602 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652570 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652651 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652596 "-" "Go-http-client/1.1"
[2024-10-24:21:13:26:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:21:13:26 +0000] "POST /invocations
HTTP/1.1" 200 652602 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:29 +0000] "POST /invocations
HTTP/1.1" 200 602519 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:21:13:29 +0000] "POST /invocations
HTTP/1.1" 200 602519 "-" "Go-http-client/1.1"
```

Get the predicted target and test labels.

[78]:
```
s3 = boto3.client('s3')
obj = s3.get_object(Bucket=bucket, Key="{}/batch-out/{}".
 ↪format(prefix,'batch-in.csv.out'))
target_predicted = pd.read_csv(io.BytesIO(obj['Body'].
 ↪read()),sep=',',names=['target'])
test_labels = test.iloc[:,0]
```

Calculate the predicted values based on the defined threshold.

Note: The predicted target will be a score, which must be converted to a binary class.

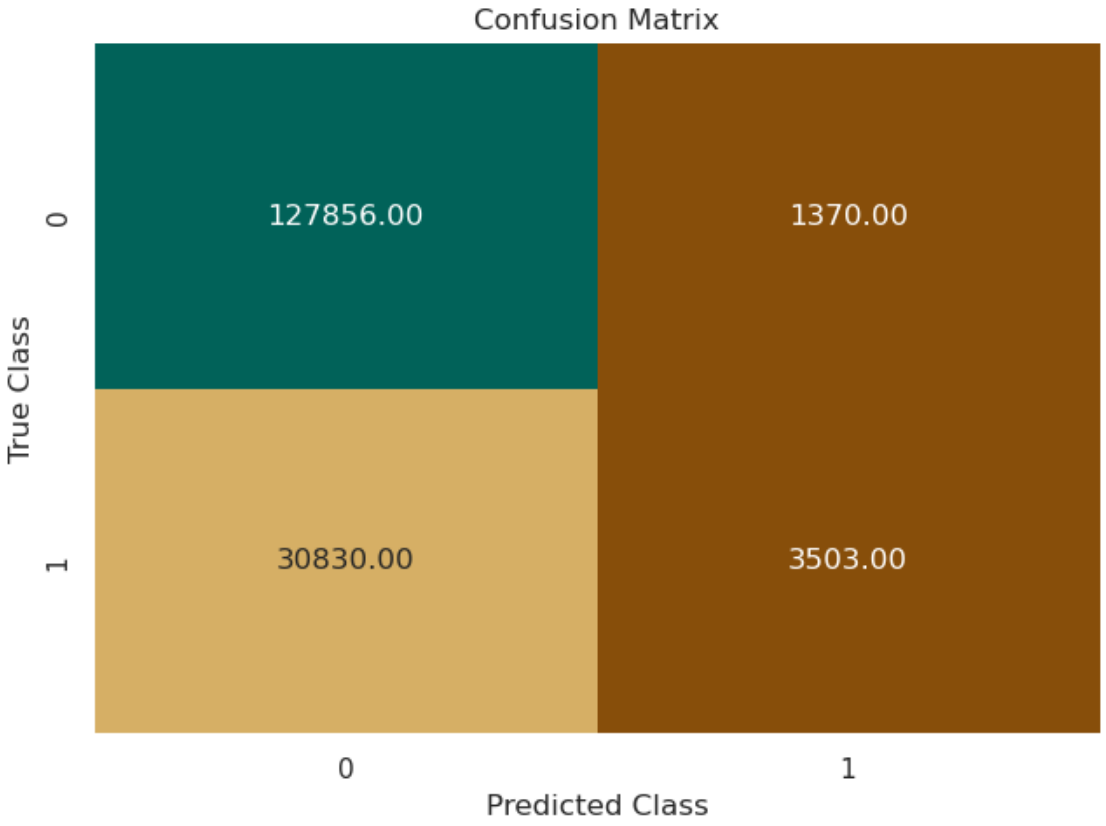
[79]:
```
print(target_predicted.head())

def binary_convert(x):
    threshold = 0.55
    if x > threshold:
        return 1
    else:
        return 0

target_predicted['target'] = target_predicted['target'].apply(binary_convert)

test_labels = test.iloc[:,0]

print(target_predicted.head())
```

```
     target
0  0.268778
```

```
1  0.119989
2  0.281162
3  0.334120
4  0.388860
   target
0       0
1       0
2       0
3       0
4       0
```

Plot a confusion matrix for your `target_predicted` and `test_labels`.

[80]:
```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



6.0.8 Try different thresholds

Question: Based on how well the model handled the test set, what can you conclude?

Enter your answer here - There is a slight improvment in True Positives

6.0.9 Hyperparameter optimization (HPO)

[81]:
```
from sagemaker.tuner import IntegerParameter, CategoricalParameter,␣
↪ContinuousParameter, HyperparameterTuner

### You can spin up multiple instances to do hyperparameter optimization in␣
↪parallel

xgb = sagemaker.estimator.Estimator(container,
                                    role=sagemaker.get_execution_role(),
                                    instance_count= 1, # make sure you have a␣
↪limit set for these instances
                                    instance_type=instance_type,
                                    output_path='s3://{}/{}/output'.
↪format(bucket, prefix),
                                    sagemaker_session=sess)

xgb.set_hyperparameters(eval_metric='auc',
                        objective='binary:logistic',
                        num_round=100,
                        rate_drop=0.3,
                        tweedie_variance_power=1.4)

hyperparameter_ranges = {'alpha': ContinuousParameter(0, 1000,␣
↪scaling_type='Linear'),
                         'eta': ContinuousParameter(0.1, 0.5,␣
↪scaling_type='Linear'),
                         'min_child_weight': ContinuousParameter(3, 10,␣
↪scaling_type='Linear'),
                         'subsample': ContinuousParameter(0.5, 1),
                         'num_round': IntegerParameter(10,150)}

objective_metric_name = 'validation:auc'

tuner = HyperparameterTuner(xgb,
                            objective_metric_name,
                            hyperparameter_ranges,
                            max_jobs=10, # Set this to 10 or above depending␣
↪upon budget and available time.
                            max_parallel_jobs=1)
```

[82]:
```
tuner.fit(inputs=data_channels)
tuner.wait()
```

```
WARNING:sagemaker.estimator:No finished training job found associated with this
estimator. Please make sure this estimator is only used for building workflow
config
```

```
WARNING:sagemaker.estimator:No finished training job found associated with this
estimator. Please make sure this estimator is only used for building workflow
config
INFO:sagemaker:Creating hyperparameter tuning job with name: sagemaker-
xgboost-241024-2116
```

…
…
…
…
…
…
…
…!
!

Wait until the training job is finished. It might take 25-30 minutes.

To monitor hyperparameter optimization jobs:

1. In the AWS Management Console, on the **Services** menu, choose **Amazon SageMaker**.
2. Choose **Training > Hyperparameter tuning jobs**.
3. You can check the status of each hyperparameter tuning job, its objective metric value, and its logs.

Check that the job completed successfully.

[83]:
```
boto3.client('sagemaker').describe_hyper_parameter_tuning_job(
    HyperParameterTuningJobName=tuner.latest_tuning_job.
 ↪job_name)['HyperParameterTuningJobStatus']
```

[83]:
```
'Completed'
```

The hyperparameter tuning job will have a model that worked the best. You can get the information about that model from the tuning job.

[84]:
```
sage_client = boto3.Session().client('sagemaker')
tuning_job_name = tuner.latest_tuning_job.job_name
print(f'tuning job name:{tuning_job_name}')
tuning_job_result = sage_client.
 ↪describe_hyper_parameter_tuning_job(HyperParameterTuningJobName=tuning_job_name)
best_training_job = tuning_job_result['BestTrainingJob']
best_training_job_name = best_training_job['TrainingJobName']
print(f"best training job: {best_training_job_name}")

best_estimator = tuner.best_estimator()

tuner_df = sagemaker.HyperparameterTuningJobAnalytics(tuning_job_name).
 ↪dataframe()
```

```
tuner_df.head()
```

```
INFO:botocore.credentials:Found credentials from IAM Role:
BaseNotebookInstanceEc2InstanceRole

tuning job name:sagemaker-xgboost-241024-2116
best training job: sagemaker-xgboost-241024-2116-009-3afaa09e

2024-10-24 22:00:56 Starting - Found matching resource for reuse
2024-10-24 22:00:56 Downloading - Downloading the training image
2024-10-24 22:00:56 Training - Training image download completed. Training in
progress.
2024-10-24 22:00:56 Uploading - Uploading generated training model
2024-10-24 22:00:56 Completed - Resource reused by training job: sagemaker-
xgboost-241024-2116-010-ffb50ec1
```

[84]:

	alpha	eta	min_child_weight	num_round	subsample	\
0	44.284933	0.491055	7.905340	132.0	0.721190	
1	0.000000	0.444088	7.512628	150.0	0.772652	
2	967.751711	0.415676	9.149434	50.0	0.671650	
3	47.676408	0.119078	7.424666	102.0	0.910618	
4	103.090738	0.258985	4.032673	10.0	0.812495	

	TrainingJobName	TrainingJobStatus	\
0	sagemaker-xgboost-241024-2116-010-ffb50ec1	Completed	
1	sagemaker-xgboost-241024-2116-009-3afaa09e	Completed	
2	sagemaker-xgboost-241024-2116-008-77ebddbe	Completed	
3	sagemaker-xgboost-241024-2116-007-8dcf247a	Completed	
4	sagemaker-xgboost-241024-2116-006-1f1040bd	Completed	

	FinalObjectiveValue	TrainingStartTime	TrainingEndTime	\
0	0.74668	2024-10-24 22:00:58+00:00	2024-10-24 22:07:38+00:00	
1	0.75159	2024-10-24 21:53:19+00:00	2024-10-24 22:00:34+00:00	
2	0.70974	2024-10-24 21:50:10+00:00	2024-10-24 21:52:59+00:00	
3	0.72684	2024-10-24 21:44:30+00:00	2024-10-24 21:49:30+00:00	
4	0.69764	2024-10-24 21:42:54+00:00	2024-10-24 21:44:04+00:00	

	TrainingElapsedTimeSeconds
0	400.0
1	435.0
2	169.0
3	300.0
4	70.0

Use the estimator `best_estimator` and train it by using the data.

Tip: See the previous XGBoost estimator fit function.

[85]:
```
# Enter your code here'
best_estimator.fit(inputs=data_channels)
```

```
INFO:sagemaker:Creating training-job with name: sagemaker-
xgboost-2024-10-24-22-10-44-381

2024-10-24 22:10:46 Starting - Starting the training job…
2024-10-24 22:11:01 Starting - Preparing the instances for training…
2024-10-24 22:11:31 Downloading - Downloading input data…
2024-10-24 22:12:06 Downloading - Downloading the training image…
2024-10-24 22:12:57 Training - Training image download completed. Training in
progress.[2024-10-24 22:13:05.803 ip-10-0-174-228.ec2.internal:7 INFO

utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
INFO:sagemaker-containers:Imported framework

sagemaker_xgboost_container.training
INFO:sagemaker-containers:Failed to parse hyperparameter

_tuning_objective_metric value validation:auc to Json.
Returning the value itself
INFO:sagemaker-containers:Failed to parse hyperparameter eval_metric value

auc to Json.
Returning the value itself
INFO:sagemaker-containers:Failed to parse hyperparameter objective value

binary:logistic to Json.
Returning the value itself
INFO:sagemaker-containers:No GPUs detected (normal if no gpus

installed)
INFO:sagemaker_xgboost_container.training:Running XGBoost Sagemaker in

algorithm mode
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Single node training.
INFO:root:Setting up HPO optimized metric to be : auc
[22:13:12] 1308472x85 matrix with 111220120 entries loaded from

/opt/ml/input/data/train?format=csv&label_column=0&delimiter=,
[22:13:13] 163559x85 matrix with 13902515 entries loaded from

/opt/ml/input/data/validation?format=csv&label_column=0&delimiter=,
[2024-10-24 22:13:13.268 ip-10-0-174-228.ec2.internal:7 INFO

json_config.py:91] Creating hook from json_config at

/opt/ml/input/config/debughookconfig.json.
```

```
[2024-10-24 22:13:13.269 ip-10-0-174-228.ec2.internal:7 INFO hook.py:201]
tensorboard_dir has not been set for the hook. SMDebug will not be exporting
tensorboard summaries.
[2024-10-24 22:13:13.269 ip-10-0-174-228.ec2.internal:7 INFO
profiler_config_parser.py:102] User has disabled profiler.
[2024-10-24 22:13:13.270 ip-10-0-174-228.ec2.internal:7 INFO hook.py:255]
Saving to /opt/ml/output/tensors
[2024-10-24 22:13:13.270 ip-10-0-174-228.ec2.internal:7 INFO
state_store.py:77] The checkpoint config file
/opt/ml/input/config/checkpointconfig.json does not exist.
INFO:root:Debug hook created from config
INFO:root:Train matrix has 1308472 rows
INFO:root:Validation matrix has 163559 rows
[0]#011train-auc:0.66323#011validation-auc:0.66300
[2024-10-24 22:13:18.968 ip-10-0-174-228.ec2.internal:7 INFO hook.py:423]
Monitoring the collections: metrics
[2024-10-24 22:13:18.970 ip-10-0-174-228.ec2.internal:7 INFO hook.py:486]
Hook is writing from the hook with pid: 7
[1]#011train-auc:0.67814#011validation-auc:0.67849
[2]#011train-auc:0.68467#011validation-auc:0.68460
[3]#011train-auc:0.68974#011validation-auc:0.68956
[4]#011train-auc:0.69530#011validation-auc:0.69481
[5]#011train-auc:0.69951#011validation-auc:0.69947
[6]#011train-auc:0.70213#011validation-auc:0.70158
[7]#011train-auc:0.70452#011validation-auc:0.70388
[8]#011train-auc:0.70795#011validation-auc:0.70716
[9]#011train-auc:0.71002#011validation-auc:0.70950
[10]#011train-auc:0.71313#011validation-auc:0.71238
[11]#011train-auc:0.71597#011validation-auc:0.71469
[12]#011train-auc:0.71719#011validation-auc:0.71588
[13]#011train-auc:0.71928#011validation-auc:0.71764
[14]#011train-auc:0.72081#011validation-auc:0.71916
[15]#011train-auc:0.72154#011validation-auc:0.71974
[16]#011train-auc:0.72306#011validation-auc:0.72110
[17]#011train-auc:0.72407#011validation-auc:0.72195
[18]#011train-auc:0.72594#011validation-auc:0.72369
[19]#011train-auc:0.72735#011validation-auc:0.72492
[20]#011train-auc:0.72863#011validation-auc:0.72622
[21]#011train-auc:0.72964#011validation-auc:0.72724
[22]#011train-auc:0.73040#011validation-auc:0.72776
[23]#011train-auc:0.73092#011validation-auc:0.72813
[24]#011train-auc:0.73153#011validation-auc:0.72854
[25]#011train-auc:0.73241#011validation-auc:0.72931
```

[26]#011train-auc:0.73312#011validation-auc:0.72984
[27]#011train-auc:0.73408#011validation-auc:0.73051
[28]#011train-auc:0.73476#011validation-auc:0.73106
[29]#011train-auc:0.73552#011validation-auc:0.73167
[30]#011train-auc:0.73613#011validation-auc:0.73218
[31]#011train-auc:0.73638#011validation-auc:0.73242
[32]#011train-auc:0.73721#011validation-auc:0.73313
[33]#011train-auc:0.73760#011validation-auc:0.73342
[34]#011train-auc:0.73775#011validation-auc:0.73354
[35]#011train-auc:0.73799#011validation-auc:0.73374
[36]#011train-auc:0.73841#011validation-auc:0.73409
[37]#011train-auc:0.73876#011validation-auc:0.73443
[38]#011train-auc:0.73928#011validation-auc:0.73478
[39]#011train-auc:0.73954#011validation-auc:0.73494
[40]#011train-auc:0.74011#011validation-auc:0.73524
[41]#011train-auc:0.74051#011validation-auc:0.73564
[42]#011train-auc:0.74117#011validation-auc:0.73603
[43]#011train-auc:0.74188#011validation-auc:0.73656
[44]#011train-auc:0.74217#011validation-auc:0.73672
[45]#011train-auc:0.74253#011validation-auc:0.73691
[46]#011train-auc:0.74299#011validation-auc:0.73712
[47]#011train-auc:0.74337#011validation-auc:0.73742
[48]#011train-auc:0.74380#011validation-auc:0.73786
[49]#011train-auc:0.74419#011validation-auc:0.73810
[50]#011train-auc:0.74437#011validation-auc:0.73823
[51]#011train-auc:0.74482#011validation-auc:0.73842
[52]#011train-auc:0.74511#011validation-auc:0.73865
[53]#011train-auc:0.74556#011validation-auc:0.73893
[54]#011train-auc:0.74596#011validation-auc:0.73919
[55]#011train-auc:0.74629#011validation-auc:0.73939
[56]#011train-auc:0.74660#011validation-auc:0.73964
[57]#011train-auc:0.74698#011validation-auc:0.73988
[58]#011train-auc:0.74751#011validation-auc:0.74016
[59]#011train-auc:0.74789#011validation-auc:0.74031
[60]#011train-auc:0.74846#011validation-auc:0.74079
[61]#011train-auc:0.74860#011validation-auc:0.74092
[62]#011train-auc:0.74882#011validation-auc:0.74110
[63]#011train-auc:0.74911#011validation-auc:0.74111
[64]#011train-auc:0.74928#011validation-auc:0.74123
[65]#011train-auc:0.74938#011validation-auc:0.74131
[66]#011train-auc:0.74979#011validation-auc:0.74164
[67]#011train-auc:0.75012#011validation-auc:0.74181
[68]#011train-auc:0.75026#011validation-auc:0.74193
[69]#011train-auc:0.75049#011validation-auc:0.74202
[70]#011train-auc:0.75069#011validation-auc:0.74214
[71]#011train-auc:0.75109#011validation-auc:0.74247
[72]#011train-auc:0.75130#011validation-auc:0.74258
[73]#011train-auc:0.75158#011validation-auc:0.74272

[74]#011train-auc:0.75189#011validation-auc:0.74288
[75]#011train-auc:0.75218#011validation-auc:0.74313
[76]#011train-auc:0.75230#011validation-auc:0.74314
[77]#011train-auc:0.75243#011validation-auc:0.74320
[78]#011train-auc:0.75269#011validation-auc:0.74333
[79]#011train-auc:0.75288#011validation-auc:0.74344
[80]#011train-auc:0.75316#011validation-auc:0.74358
[81]#011train-auc:0.75321#011validation-auc:0.74361
[82]#011train-auc:0.75345#011validation-auc:0.74378
[83]#011train-auc:0.75371#011validation-auc:0.74400
[84]#011train-auc:0.75406#011validation-auc:0.74409
[85]#011train-auc:0.75436#011validation-auc:0.74430
[86]#011train-auc:0.75473#011validation-auc:0.74471
[87]#011train-auc:0.75499#011validation-auc:0.74483
[88]#011train-auc:0.75509#011validation-auc:0.74487
[89]#011train-auc:0.75517#011validation-auc:0.74488
[90]#011train-auc:0.75545#011validation-auc:0.74501
[91]#011train-auc:0.75557#011validation-auc:0.74503
[92]#011train-auc:0.75565#011validation-auc:0.74506
[93]#011train-auc:0.75594#011validation-auc:0.74526
[94]#011train-auc:0.75609#011validation-auc:0.74537
[95]#011train-auc:0.75631#011validation-auc:0.74539
[96]#011train-auc:0.75662#011validation-auc:0.74562
[97]#011train-auc:0.75673#011validation-auc:0.74567
[98]#011train-auc:0.75680#011validation-auc:0.74574
[99]#011train-auc:0.75710#011validation-auc:0.74588
[100]#011train-auc:0.75735#011validation-auc:0.74603
[101]#011train-auc:0.75748#011validation-auc:0.74610
[102]#011train-auc:0.75774#011validation-auc:0.74632
[103]#011train-auc:0.75797#011validation-auc:0.74652
[104]#011train-auc:0.75815#011validation-auc:0.74661
[105]#011train-auc:0.75841#011validation-auc:0.74674
[106]#011train-auc:0.75860#011validation-auc:0.74686
[107]#011train-auc:0.75880#011validation-auc:0.74699
[108]#011train-auc:0.75897#011validation-auc:0.74705
[109]#011train-auc:0.75920#011validation-auc:0.74725
[110]#011train-auc:0.75939#011validation-auc:0.74742
[111]#011train-auc:0.75949#011validation-auc:0.74750
[112]#011train-auc:0.75982#011validation-auc:0.74775
[113]#011train-auc:0.76001#011validation-auc:0.74789
[114]#011train-auc:0.76018#011validation-auc:0.74798
[115]#011train-auc:0.76029#011validation-auc:0.74796
[116]#011train-auc:0.76041#011validation-auc:0.74801
[117]#011train-auc:0.76055#011validation-auc:0.74806
[118]#011train-auc:0.76072#011validation-auc:0.74821
[119]#011train-auc:0.76095#011validation-auc:0.74834
[120]#011train-auc:0.76108#011validation-auc:0.74840
[121]#011train-auc:0.76128#011validation-auc:0.74857

```
[122]#011train-auc:0.76158#011validation-auc:0.74871
[123]#011train-auc:0.76188#011validation-auc:0.74878
[124]#011train-auc:0.76192#011validation-auc:0.74881
[125]#011train-auc:0.76202#011validation-auc:0.74888
[126]#011train-auc:0.76219#011validation-auc:0.74903
[127]#011train-auc:0.76240#011validation-auc:0.74916
[128]#011train-auc:0.76263#011validation-auc:0.74934
[129]#011train-auc:0.76299#011validation-auc:0.74967
[130]#011train-auc:0.76310#011validation-auc:0.74974
[131]#011train-auc:0.76329#011validation-auc:0.74979
[132]#011train-auc:0.76349#011validation-auc:0.74995
[133]#011train-auc:0.76374#011validation-auc:0.75016
[134]#011train-auc:0.76392#011validation-auc:0.75031
[135]#011train-auc:0.76399#011validation-auc:0.75034
[136]#011train-auc:0.76419#011validation-auc:0.75044
[137]#011train-auc:0.76453#011validation-auc:0.75071
[138]#011train-auc:0.76462#011validation-auc:0.75073
[139]#011train-auc:0.76487#011validation-auc:0.75086
[140]#011train-auc:0.76501#011validation-auc:0.75091
[141]#011train-auc:0.76516#011validation-auc:0.75096
[142]#011train-auc:0.76522#011validation-auc:0.75099
[143]#011train-auc:0.76528#011validation-auc:0.75100
[144]#011train-auc:0.76548#011validation-auc:0.75110
[145]#011train-auc:0.76579#011validation-auc:0.75130
[146]#011train-auc:0.76599#011validation-auc:0.75133
[147]#011train-auc:0.76618#011validation-auc:0.75146
[148]#011train-auc:0.76634#011validation-auc:0.75148
[149]#011train-auc:0.76649#011validation-auc:0.75159

2024-10-24 22:20:17 Uploading - Uploading generated training model
2024-10-24 22:20:17 Completed - Training job completed
Training seconds: 526
Billable seconds: 526
```

Use the batch transformer for your new model, and evaluate the model on the test dataset.

[86]:
```
batch_output = "s3://{}/{}/batch-out/".format(bucket,prefix)
batch_input = "s3://{}/{}/batch-in/{}".format(bucket,prefix,batch_X_file)

xgb_transformer = best_estimator.transformer(instance_count=1,
                                             instance_type=instance_type,
                                             strategy='MultiRecord',
                                             assemble_with='Line',
                                             output_path=batch_output)

xgb_transformer.transform(data=batch_input,
                          data_type='S3Prefix',
                          content_type='text/csv',
```

```
                           split_type='Line')
xgb_transformer.wait()
```

```
INFO:sagemaker:Creating model with name: sagemaker-
xgboost-2024-10-24-22-20-58-177
INFO:sagemaker:Creating transform job with name: sagemaker-
xgboost-2024-10-24-22-20-58-792
```

```
…[2024-10-24:22:27:30:INFO] No GPUs

detected (normal if no gpus installed)
[2024-10-24:22:27:30:INFO] No GPUs detected (normal if no gpus

installed)
[2024-10-24:22:27:30:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {

  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 22:27:30 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 22:27:30 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 22:27:30 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 22:27:31 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 22:27:31 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 22:27:31 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 22:27:31 +0000] [32] [INFO] Booting worker with pid: 32
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
```

```
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
2024-10-24T22:27:35.972:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
[2024-10-24:22:27:36:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:36:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653307 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653525 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653354 "-" "Go-http-client/1.1"
[2024-10-24:22:27:41:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:41 +0000] "POST /invocations
HTTP/1.1" 200 653267 "-" "Go-http-client/1.1"
```

```
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653307 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653525 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653354 "-" "Go-http-client/1.1"
[2024-10-24:22:27:41:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:41 +0000] "POST /invocations
HTTP/1.1" 200 653267 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:42 +0000] "POST /invocations
HTTP/1.1" 200 603344 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:42 +0000] "POST /invocations
HTTP/1.1" 200 603344 "-" "Go-http-client/1.1"

[2024-10-24:22:27:30:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:30:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:30:INFO] nginx config:
worker_processes auto;
[2024-10-24:22:27:30:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:30:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:30:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 22:27:30 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 22:27:30 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 22:27:30 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 22:27:31 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 22:27:31 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 22:27:31 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 22:27:31 +0000] [32] [INFO] Booting worker with pid: 32
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
```

```
events {
  worker_connections 2048;
}
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2024-10-24 22:27:30 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-10-24 22:27:30 +0000] [19] [INFO] Listening at:
unix:/tmp/gunicorn.sock (19)
[2024-10-24 22:27:30 +0000] [19] [INFO] Using worker: gevent
[2024-10-24 22:27:31 +0000] [26] [INFO] Booting worker with pid: 26
[2024-10-24 22:27:31 +0000] [27] [INFO] Booting worker with pid: 27
[2024-10-24 22:27:31 +0000] [28] [INFO] Booting worker with pid: 28
[2024-10-24 22:27:31 +0000] [32] [INFO] Booting worker with pid: 32
```

```
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-10-24:22:27:35:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [24/Oct/2024:22:27:35 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
2024-10-24T22:27:35.972:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
[2024-10-24:22:27:36:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:36:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] No GPUs detected (normal if no gpus
installed)
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
[2024-10-24:22:27:37:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653307 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653525 "-" "Go-http-client/1.1"
```

```
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653354 "-" "Go-http-client/1.1"
[2024-10-24:22:27:41:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:41 +0000] "POST /invocations
HTTP/1.1" 200 653267 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653307 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653525 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:40 +0000] "POST /invocations
HTTP/1.1" 200 653354 "-" "Go-http-client/1.1"
[2024-10-24:22:27:41:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [24/Oct/2024:22:27:41 +0000] "POST /invocations
HTTP/1.1" 200 653267 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:42 +0000] "POST /invocations
HTTP/1.1" 200 603344 "-" "Go-http-client/1.1"
169.254.255.130 - - [24/Oct/2024:22:27:42 +0000] "POST /invocations
HTTP/1.1" 200 603344 "-" "Go-http-client/1.1"
```

[87]:
```
s3 = boto3.client('s3')
obj = s3.get_object(Bucket=bucket, Key="{}/batch-out/{}".
 ↪format(prefix,'batch-in.csv.out'))
target_predicted = pd.read_csv(io.BytesIO(obj['Body'].
 ↪read()),sep=',',names=['target'])
test_labels = test.iloc[:,0]
```

Get the predicted target and test labels.

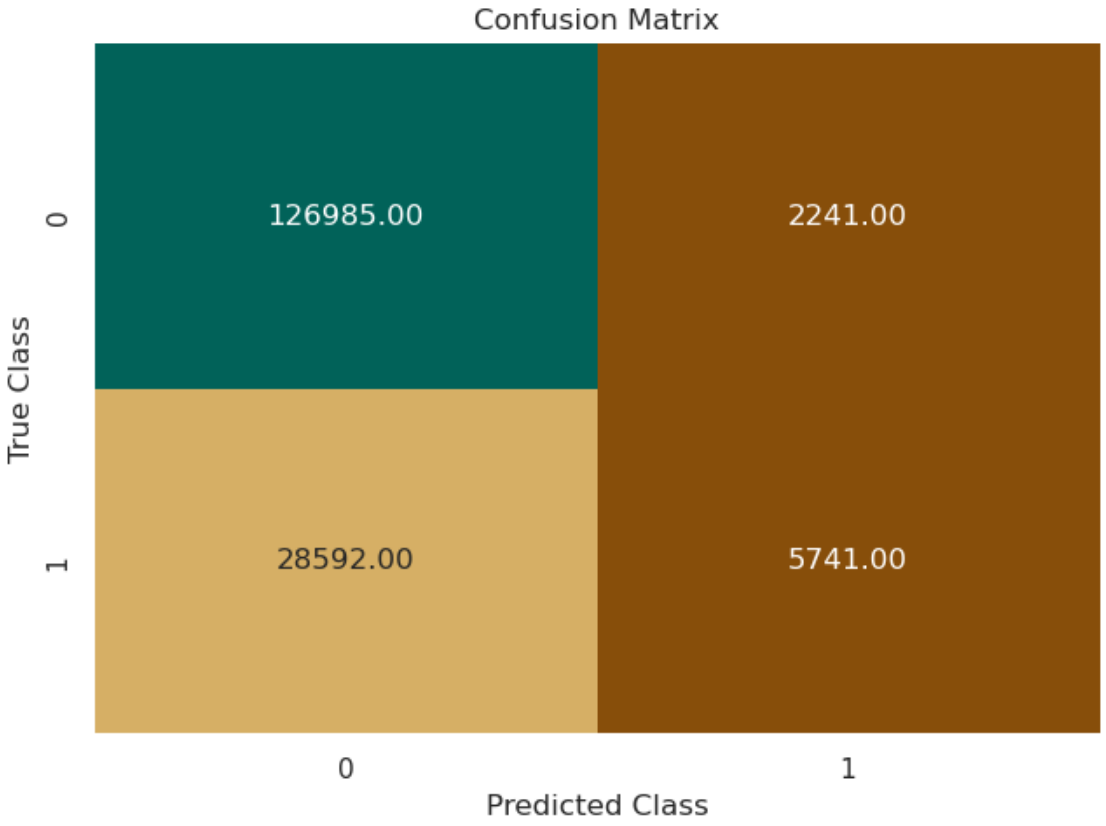
[88]:
```
print(target_predicted.head())

def binary_convert(x):
    threshold = 0.55
    if x > threshold:
        return 1
    else:
        return 0

target_predicted['target'] = target_predicted['target'].apply(binary_convert)

test_labels = test.iloc[:,0]

print(target_predicted.head())
```

```
   target
```

```
0  0.136790
1  0.160483
2  0.318141
3  0.447111
4  0.445784
   target
0       0
1       0
2       0
3       0
4       0
```

Plot a confusion matrix for your `target_predicted` and `test_labels`.

[89]:
```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



Question: Try different hyperparameters and hyperparameter ranges. Do these changes improve the model?

6.1 Conclusion

You have now iterated through training and evaluating your model at least a couple of times. It's time to wrap up this project and reflect on:

- What you learned
- What types of steps you might take moving forward (assuming that you had more time)

Use the following cell to answer some of these questions and other relevant questions:

1. Does your model performance meet your business goal? If not, what are some things you'd like to do differently if you had more time for tuning?
2. How much did your model improve as you made changes to your dataset, features, and hyperparameters? What types of techniques did you employ throughout this project, and which yielded the greatest improvements in your model?
3. What were some of the biggest challenges that you encountered throughout this project?
4. Do you have any unanswered questions about aspects of the pipeline that didn't make sense to you?
5. What were the three most important things that you learned about machine learning while working on this project?

Project presentation: Make sure that you also summarize your answers to these questions in your project presentation. Combine all your notes for your project presentation and prepare to present your findings to the class.