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

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.

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.

Features

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

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&Subject_ID2=0.

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.

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

In [ ]:
```
# Predicting FLight Delays
```

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

In [ ]:
```
# Business Problem:
# A travel booking website aims to improve the customer experience by predic
# weather-related flight delays. When a customer books a flight to or from c
# of the busiest domestic airports in the US, the system should notify them
# the flight is likely to be delayed due to weather. This feature would allo
# customers to make informed decisions, potentially choosing alternative fli
# or planning accordingly, leading to increased customer satisfaction and re
```

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

In [ ]:
```
# This is a binary classification problem.
# The task is to predict whether a flight will be "Delayed" or "On-Time" due
# which requires a supervised learning approach with labeled data (historica
# Based on the data, we can train a model to classify each flight booking in
# "Delayed" or "On-Time."
```

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

In [ ]:
```
# Data Appropriateness Analysis:
# 1. Relevance: The dataset includes historical on-time performance data for
#    which is directly relevant to predicting delays. If it includes weather
#    precipitation, wind speed) and specific flight details (e.g., departure
#    this data is well-suited for identifying patterns that lead to weather-
# 2. Completeness: For accurate predictions, the dataset should contain a su
#    covering a variety of weather conditions and seasons. Missing or incomp
#    could impact model performance.
# 3. Data Quality: The dataset should be free from significant errors, incon
#    could lead to incorrect predictions. It's essential to validate the acc
# 4. Representativeness: The data should represent flights across various bu
#    to generalize well for future bookings. If certain airports, times, or
#    the model may struggle with those scenarios.
# 5. Label Availability: For a supervised learning approach, each record sho
#    based on actual flight outcomes. Accurate labels are essential for effe
#
# Overall, the dataset seems appropriate if it meets these conditions, but c
# preprocessing may be needed to ensure it is fully suitable for building a
```

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.

In [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.
```

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.

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.

In [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
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_11.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2014_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_1.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_12.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2014_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_4.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_3.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_6.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_5.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_2.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_10.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2014_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_8.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_9.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2014_7.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2014_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_1.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_10.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2015_10.zip
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_2.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_11.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2015_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_3.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_5.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_4.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_6.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_12.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2015_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_1.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_9.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_8.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2015_7.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2015_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_12.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2016_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_2.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_10.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2016_10.zip
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_11.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2016_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_3.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_4.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_5.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_8.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_1.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_7.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_9.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_10.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2017_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2016_6.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2016_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_12.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2017_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_2.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_11.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2017_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_3.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_3.zip
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_4.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_6.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_9.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_8.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_7.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_7.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2017_5.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2017_5.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_1.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_1.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_10.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2018_10.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_12.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2018_12.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_3.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_3.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_2.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_2.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_11.z
ip to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perfor
mance_1987_present_2018_11.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_4.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_4.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_5.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_5.zip
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_6.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_6.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_9.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_9.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_8.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_8.zip
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/data/On_Time_Reporting_Carrier_On_Time_Performance_1987_present_2018_7.zi
p to ../project/data/FlightDelays/On_Time_Reporting_Carrier_On_Time_Perform
ance_1987_present_2018_7.zip
```

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

Out[3]:
```
60
```

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

In [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_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_10.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_8.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_9.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_1.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_12.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_6.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_3.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_4.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_4.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_6.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_8.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_10.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_2.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_6.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_5.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_4.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_6.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_2.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_11.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_5.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_6.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_12.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_10.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_7.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_3.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_1.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_10.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_7.zip
```

```
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_8.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_5.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_10.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_12.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_4.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_11.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_9.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_8.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_7.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_12.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_3.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_7.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_11.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_8.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_12.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_11.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_1.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_2.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_5.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_7.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_3.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_2.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_9.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_3.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2016_11.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_4.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_1.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_9.zip
```

```
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2018_2.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2014_9.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2015_5.zip
Extracting /home/ec2-user/SageMaker/project/data/FlightDelays/On_Time_Repor
ting_Carrier_On_Time_Performance_1987_present_2017_1.zip
Files Extracted
```

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

Out[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.

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

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

Out[6]:

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.cs` file first. You can use the built-in `read_csv` function in Python (pandas.read_csv documentation).

In [7]:
```
df_temp = pd.read_csv(f"{csv_base_path}On_Time_Reporting_Carrier_On_Time_Per
```

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.

In [8]:
```
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.

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

Out[9]:

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	DOT_ID_R
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	

10 rows × 110 columns

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

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

In [15]:
```
# Enter your code here
print(f'Column names that contain "Del" are: ')
print('#########')
print([x for x in df_temp.columns if "Del" in x])
```

```
Column names that contain "Del" are:
#########
['DepDelay', 'DepDelayMinutes', 'DepDel15', 'DepartureDelayGroups', 'ArrDel
ay', 'ArrDelayMinutes', 'ArrDel15', 'ArrivalDelayGroups', 'CarrierDelay',
'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay', 'DivArrDe
lay']
```

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()`).

In [23]:
```
print("The #rows and #columns are ", df_temp.shape[0] , " and ", df_temp.sha
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()), " t
print("The airlines covered in this dataset are: ", list(df_temp.Reporting_A
print("The Orgin airports covered are: ", list(df_temp.Origin.unique()))
print("The Destination airports covered are: ", list(df_temp.Dest.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 Orgin airports covered are:  ['DFW', 'LGA', 'MSN', 'MSP', 'ATL', 'BDL',
'VLD', 'JFK', 'RDU', 'CHS', 'DTW', 'GRB', 'PVD', 'SHV', 'FNT', 'PIT', 'RI
C', 'RST', 'RSW', 'CVG', 'LIT', 'ORD', 'JAX', 'TRI', 'BOS', 'CWA', 'DCA',
'CHO', 'AVP', 'IND', 'GRR', 'BTR', 'MEM', 'TUL', 'CLE', 'STL', 'BTV', 'OM
A', 'MGM', 'TVC', 'SAV', 'GSP', 'EWR', 'OAJ', 'BNA', 'MCI', 'TLH', 'ROC',
'LEX', 'PWM', 'BUF', 'AGS', 'CLT', 'GSO', 'BWI', 'SAT', 'PHL', 'TYS', 'AC
K', 'DSM', 'GNV', 'AVL', 'BGR', 'MHT', 'ILM', 'MOT', 'IAH', 'SBN', 'SYR',
'ORF', 'MKE', 'XNA', 'MSY', 'PBI', 'ABE', 'HPN', 'EVV', 'ALB', 'LNK', 'AU
S', 'PHF', 'CHA', 'GTR', 'BMI', 'BQK', 'CID', 'CAK', 'ATW', 'ABY', 'CAE',
'SRQ', 'MLI', 'BHM', 'IAD', 'CSG', 'CMH', 'MCO', 'MBS', 'FLL', 'SDF', 'TP
A', 'MVY', 'LAS', 'LGB', 'SFO', 'SAN', 'LAX', 'RNO', 'PDX', 'ANC', 'ABQ',
'SLC', 'DEN', 'PHX', 'OAK', 'SMF', 'SJU', 'SEA', 'HOU', 'STX', 'BUR', 'SW
F', 'SJC', 'DAB', 'BQN', 'PSE', 'ORH', 'HYA', 'STT', 'ONT', 'HRL', 'ICT',
'ISP', 'LBB', 'MAF', 'MDW', 'OKC', 'PNS', 'SNA', 'TUS', 'AMA', 'BOI', 'CR
P', 'DAL', 'ECP', 'ELP', 'GEG', 'LFT', 'MFE', 'MDT', 'JAN', 'COS', 'MOB',
'VPS', 'MTJ', 'DRO', 'GPT', 'BFL', 'MRY', 'SBA', 'PSP', 'FSD', 'BRO', 'RA
P', 'COU', 'STS', 'PIA', 'FAT', 'SBP', 'FSM', 'HSV', 'BIS', 'DAY', 'BZN',
'MIA', 'EYW', 'MYR', 'HHH', 'GJT', 'FAR', 'SGF', 'HOB', 'CLL', 'LRD', 'AE
X', 'ERI', 'MLU', 'LCH', 'ROA', 'LAW', 'MHK', 'GRK', 'SAF', 'GRI', 'JLN',
'ROW', 'FWA', 'CRW', 'LAN', 'OGG', 'HNL', 'KOA', 'EGE', 'LIH', 'MLB', 'JA
C', 'FAI', 'RDM', 'ADQ', 'BET', 'BRW', 'SCC', 'KTN', 'YAK', 'CDV', 'JNU',
'SIT', 'PSG', 'WRG', 'OME', 'OTZ', 'ADK', 'FCA', 'FAY', 'PSC', 'BIL', 'MS
O', 'ITO', 'PPG', 'MFR', 'EUG', 'GUM', 'SPN', 'DLH', 'TTN', 'BKG', 'SFB',
'PIE', 'PGD', 'AZA', 'SMX', 'RFD', 'SCK', 'OWB', 'HTS', 'BLV', 'IAG', 'US
A', 'GFK', 'BLI', 'ELM', 'PBG', 'LCK', 'GTF', 'OGD', 'IDA', 'PVU', 'TOL',
'PSM', 'CKB', 'HGR', 'SPI', 'STC', 'ACT', 'TYR', 'ABI', 'AZO', 'CMI', 'BP
T', 'GCK', 'MQT', 'ALO', 'TXK', 'SPS', 'SWO', 'DBQ', 'SUX', 'SJT', 'GGG',
'LSE', 'LBE', 'ACY', 'LYH', 'PGV', 'HVN', 'EWN', 'DHN', 'PIH', 'IMT', 'WY
S', 'CPR', 'SCE', 'HLN', 'SUN', 'ISN', 'CMX', 'EAU', 'LWB', 'SHD', 'LBF',
'HYS', 'SLN', 'EAR', 'VEL', 'CNY', 'GCC', 'RKS', 'PUB', 'LBL', 'MKG', 'PA
H', 'CGI', 'UIN', 'BFF', 'DVL', 'JMS', 'LAR', 'SGU', 'PRC', 'ASE', 'RDD',
'ACV', 'OTH', 'COD', 'LWS', 'ABR', 'APN', 'ESC', 'PLN', 'BJI', 'BRD', 'BT
M', 'CDC', 'CIU', 'EKO', 'TWF', 'HIB', 'BGM', 'RHI', 'ITH', 'INL', 'FLG',
'YUM', 'MEI', 'PIB', 'HDN']
The Destination airports covered are:  ['CVG', 'PWM', 'RDU', 'MSP', 'MSN',
'SHV', 'CLT', 'PIT', 'RIC', 'IAH', 'ATL', 'JFK', 'DCA', 'DTW', 'LGA', 'TY
S', 'PVD', 'FNT', 'LIT', 'BUF', 'ORD', 'TRI', 'IND', 'BGR', 'AVP', 'BWI',
'LEX', 'BDL', 'GRR', 'CWA', 'TUL', 'MEM', 'AGS', 'EWR', 'MGM', 'PHL', 'SY
R', 'OMA', 'STL', 'TVC', 'ORF', 'CLE', 'ABY', 'BOS', 'OAJ', 'TLH', 'BTR',
'SAT', 'JAX', 'BNA', 'CHO', 'VLD', 'ROC', 'DFW', 'GNV', 'ACK', 'PBI', 'CH
S', 'GRB', 'MOT', 'MKE', 'DSM', 'ILM', 'GSO', 'MCI', 'SBN', 'BTV', 'MVY',
'XNA', 'RST', 'EVV', 'HPN', 'RSW', 'MDT', 'ROA', 'GSP', 'MCO', 'CSG', 'SA
V', 'PHF', 'ALB', 'CHA', 'ABE', 'BMI', 'MSY', 'IAD', 'GTR', 'CID', 'CAK',
'ATW', 'AUS', 'BQK', 'MLI', 'CAE', 'CMH', 'AVL', 'MBS', 'FLL', 'SDF', 'TP
A', 'LNK', 'SRQ', 'MHT', 'BHM', 'LAS', 'SFO', 'SAN', 'RNO', 'LGB', 'ANC',
'PDX', 'SJU', 'ABQ', 'SLC', 'DEN', 'LAX', 'PHX', 'OAK', 'SMF', 'SEA', 'ST
X', 'BUR', 'DAB', 'SJC', 'SWF', 'HOU', 'BQN', 'PSE', 'ORH', 'HYA', 'STT',
'ONT', 'DAL', 'ECP', 'ELP', 'HRL', 'MAF', 'MDW', 'OKC', 'PNS', 'SNA', 'AM
A', 'BOI', 'GEG', 'ICT', 'LBB', 'TUS', 'ISP', 'CRP', 'MFE', 'LFT', 'VPS',
'JAN', 'COS', 'MOB', 'DRO', 'GPT', 'BFL', 'COU', 'SBP', 'MTJ', 'SBA', 'PS
```

```
P', 'FSD', 'FSM', 'BRO', 'PIA', 'STS', 'FAT', 'RAP', 'MRY', 'HSV', 'BIS',
'DAY', 'BZN', 'MIA', 'EYW', 'MYR', 'HHH', 'GJT', 'FAR', 'MLU', 'LRD', 'CL
L', 'LCH', 'FWA', 'GRK', 'SGF', 'HOB', 'LAW', 'MHK', 'SAF', 'JLN', 'ROW',
'GRI', 'AEX', 'CRW', 'LAN', 'ERI', 'HNL', 'KOA', 'OGG', 'EGE', 'LIH', 'JA
C', 'MLB', 'RDM', 'BET', 'ADQ', 'BRW', 'SCC', 'FAI', 'JNU', 'CDV', 'YAK',
'SIT', 'KTN', 'WRG', 'PSG', 'OME', 'OTZ', 'ADK', 'FCA', 'BIL', 'PSC', 'FA
Y', 'MSO', 'ITO', 'PPG', 'MFR', 'DLH', 'EUG', 'GUM', 'SPN', 'TTN', 'BKG',
'AZA', 'SFB', 'LCK', 'BLI', 'SCK', 'PIE', 'RFD', 'PVU', 'PBG', 'BLV', 'PG
D', 'SPI', 'USA', 'TOL', 'IDA', 'ELM', 'HTS', 'HGR', 'SMX', 'OGD', 'GFK',
'STC', 'GTF', 'IAG', 'CKB', 'OWB', 'PSM', 'ABI', 'TYR', 'ALO', 'SUX', 'AZ
O', 'ACT', 'CMI', 'BPT', 'TXK', 'SWO', 'SPS', 'DBQ', 'SJT', 'GGG', 'LSE',
'MQT', 'GCK', 'LBE', 'ACY', 'LYH', 'PGV', 'HVN', 'EWN', 'DHN', 'PIH', 'WY
S', 'SCE', 'IMT', 'HLN', 'ASE', 'SUN', 'ISN', 'EAR', 'SGU', 'VEL', 'SHD',
'LWB', 'MKG', 'SLN', 'HYS', 'BFF', 'PUB', 'LBL', 'CMX', 'EAU', 'PAH', 'UI
N', 'RKS', 'CGI', 'CNY', 'JMS', 'DVL', 'LAR', 'GCC', 'LBF', 'PRC', 'RDD',
'ACV', 'OTH', 'COD', 'LWS', 'ABR', 'APN', 'PLN', 'BJI', 'CPR', 'BRD', 'BT
M', 'CDC', 'CIU', 'ESC', 'EKO', 'ITH', 'HIB', 'BGM', 'TWF', 'RHI', 'INL',
'FLG', 'YUM', 'MEI', 'PIB', 'HDN']
```

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).

In [24]:
```
counts = pd.DataFrame({'Origin':df_temp.Origin.value_counts(), 'Destination'
counts
```

Out[24]:

	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 × 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).

In [25]:
```
counts.sort_values(by=['Origin', 'Destination'],ascending=False).head(15) #
```

Out[25]:

	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

In [26]:
```
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}')
```

In [27]:
```
#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 airport
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.

In [28]:
```
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/c
ombined_files.csv
CSVs merged in 4.47 minutes
```

Load the dataset

Load the combined dataset.

In [29]:
```
data = pd.read_csv(combined_csv_filename)
```

Print the first five records.

In [30]:
```
# Enter your code here
# Printed the 1st 5 records here.
data.head(5)
```

Out[30]:

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	Origin	Or
0	2018	3	8	31	5	2018-08-31	UA	LAX	
1	2018	3	8	31	5	2018-08-31	UA	SFO	
2	2018	3	8	31	5	2018-08-31	UA	DEN	
3	2018	3	8	31	5	2018-08-31	UA	IAH	
4	2018	3	8	31	5	2018-08-31	UA	IAH	

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?

In [35]:
```
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.uni
print("The date range for data is :" , min(data.FlightDate.unique()), " to "
print("The airlines covered in this dataset are: ", list(data.Reporting_Airl
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:  [2018, 2017, 2015, 2016, 2014]
The months covered in this dataset are:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1
1, 12]
The date range for data is : 2014-01-01  to  2018-12-31
The airlines covered in this dataset are:  ['UA', 'AA', 'OO', 'WN', 'DL']
The Origin airports covered are:  ['LAX', 'SFO', 'DEN', 'IAH', 'PHX', 'OR
D', 'ATL', 'DFW', 'CLT']
The Destination airports covered are:  ['SFO', 'IAH', 'ORD', 'LAX', 'DEN',
'PHX', 'DFW', 'ATL', '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)
```

In [36]:
```
data.rename(columns={'ArrDel15': 'is_delay'}, inplace=True) # Enter your coc
```

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)`).

In [37]:
```
# Enter your code here
data.isnull().sum(axis=0)
```

Out[37]:
```
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.

In [38]:
```
### Remove null columns
data = data[~data.is_delay.isnull()]
data.isnull().sum(axis = 0)
```

Out[38]:
```
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.

In [42]:
```
data['DepHourofDay'] = (data['CRSDepTime']//100)

#Adding the data head to see more

data.head()
```

Out[42]:

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	Origin	Or
0	2018	3	8	31	5	2018-08-31	UA	LAX	
1	2018	3	8	31	5	2018-08-31	UA	SFO	
2	2018	3	8	31	5	2018-08-31	UA	DEN	
3	2018	3	8	31	5	2018-08-31	UA	IAH	
4	2018	3	8	31	5	2018-08-31	UA	IAH	

5 rows × 21 columns

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 models: 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

Data exploration

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

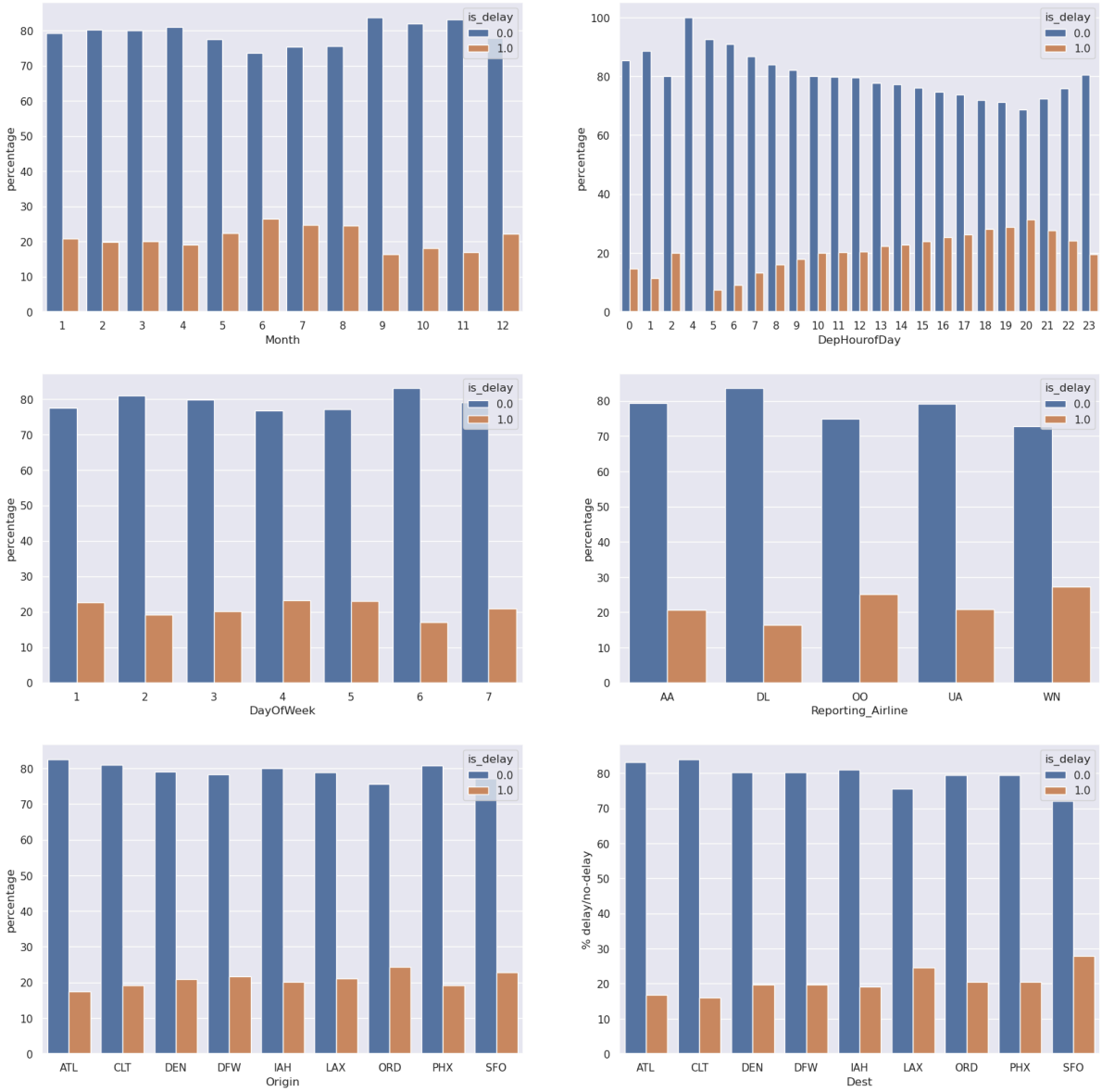
In [43]:
```
(data.groupby('is_delay').size()/len(data) ).plot(kind='bar')# Enter your co
plt.ylabel('Frequency')
plt.title('Distribution of classes')
plt.show()
```

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

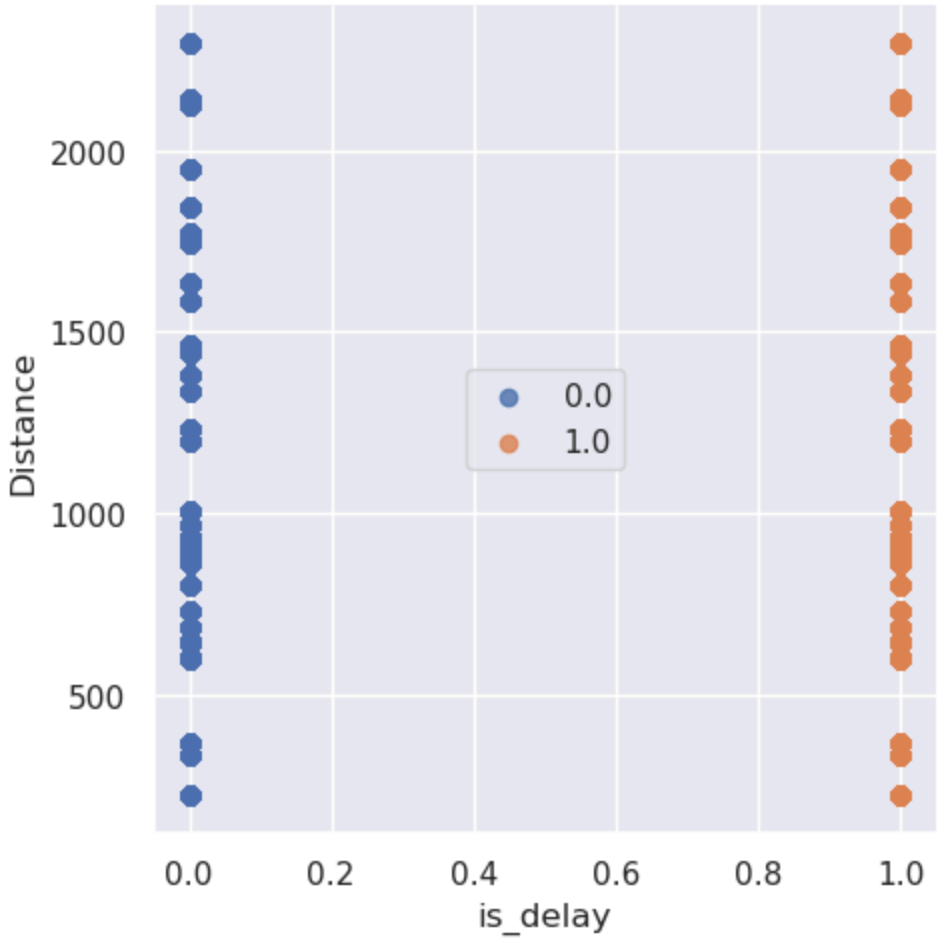
In [ ]:
```
# Enter your answer here
```

Run the following two cells and answer the questions.

In [44]:
```
viz_columns = ['Month', 'DepHourofDay', 'DayOfWeek', 'Reporting_Airline', 'O
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).rer
    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()
```



In [45]:
```
sns.lmplot( x="is_delay", y="Distance", data=data, fit_reg=False, hue='is_de
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?
- What time of the day has the most delays?
- What day of the week has the most delays?
- Which airline has the most delays?
- Which origin and destination airports have the most delays?
- Is flight distance a factor in the delays?

In [ ]:
```
# September has the most delays
# The 4th hour of the day has most delays
# The 6th day of the week
# DL Airlines
# ATL
# I can't tell
```

Features

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

In [46]:
```
data.columns
```

Out[46]:
```
Index(['Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek', 'FlightDate',
       'Reporting_Airline', 'Origin', 'OriginState', 'Dest', 'DestState',
       'CRSDepTime', 'Cancelled', 'Diverted', 'Distance', 'DistanceGroup',
       'ArrDelay', 'ArrDelayMinutes', 'is_delay', 'AirTime', 'DepHourofDa
y'],
      dtype='object')
```

In [47]:
```
data.dtypes
```

Out[47]:
```
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).

In [48]:
```
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)
```

In [49]:
```
data_dummies = pd.get_dummies(data[['Quarter', 'Month', 'DayofMonth', 'DayOf
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.

In [50]:
```
# Enter your code here
data.head()
```

Out[50]:

	is_delay	Distance	Quarter_2	Quarter_3	Quarter_4	Month_2	Month_3	Month_4	Mont
0	0.0	337.0	0	1	0	0	0	0	
1	0.0	1635.0	0	1	0	0	0	0	
2	0.0	888.0	0	1	0	0	0	0	
3	0.0	1379.0	0	1	0	0	0	0	
4	0.0	862.0	0	1	0	0	0	0	

5 rows × 94 columns

In [51]:
```
# Enter your code here
data.shape
```

Out[51]:
```
(1635590, 94)
```

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).

In [54]:
```
data.rename(columns = {'is_delay':'target'}, inplace=True )# Enter your code
```

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.

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.

Train-test split

In [55]:
```
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_
```

```
    test, validate = train_test_split(test_and_validate, test_size=0.5, rand
    return train, validate, test
```

In [56]:
```
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
```

Baseline classification model

In [58]:
```
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_
                                               instance_count=1,
                                               instance_type= 'ml.m4.xlarge'
                                               predictor_type='binary_classi
                                               binary_classifier_model_selec
```

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

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 =
'cross_entropy_loss')
```

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.

In [59]:
```
### Create train, validate, and test records
train_records = classifier_estimator.record_set(train.values[:, 1:].astype(r
val_records = classifier_estimator.record_set(validate.values[:, 1:].astype(
test_records = classifier_estimator.record_set(test.values[:, 1:].astype(np.
```

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

Sample code

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

In [60]:
```
### 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. Defa
ulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating training-job with name: linear-learner-2024-11-04-0
0-35-08-187
```

```
2024-11-04 00:35:09 Starting - Starting the training job...
2024-11-04 00:35:36 Starting - Preparing the instances for training......
2024-11-04 00:36:22 Downloading - Downloading input data...
2024-11-04 00:37:08 Downloading - Downloading the training image.........
2024-11-04 00:38:19 Training - Training image download completed. Training
in progress.Docker entrypoint called with argument(s): train
Running default environment configuration script
[11/04/2024 00:38:35 INFO 140618102482752] Reading default configuration fr
om /opt/amazon/lib/python3.8/site-packages/algorithm/resources/default-inpu
t.json: {'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': 'auto',
'use_bias': 'true', 'binary_classifier_model_selection_criteria': 'accurac
y', 'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8', 'nu
m_models': 'auto', 'num_calibration_samples': '10000000', 'init_method': 'u
niform', 'init_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0', 'o
ptimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5', 'los
s_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1', 'accura
cy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto', 'learning_r
ate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult': 'auto',
'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step': 'a
uto', 'lr_scheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto', 'po
sitive_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': 'aut
o', '_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info',
'_tuning_objective_metric': '', 'early_stopping_patience': '3', 'early_stop
ping_tolerance': '0.001', '_enable_profiler': 'false'}
[11/04/2024 00:38:35 INFO 140618102482752] Merging with provided configurat
ion from /opt/ml/input/config/hyperparameters.json: {'binary_classifier_mod
el_selection_criteria': 'cross_entropy_loss', 'feature_dim': '93', 'mini_ba
tch_size': '1000', 'predictor_type': 'binary_classifier'}
[11/04/2024 00:38:35 INFO 140618102482752] Final configuration: {'mini_batc
h_size': '1000', 'epochs': '15', 'feature_dim': '93', 'use_bias': 'true',
'binary_classifier_model_selection_criteria': 'cross_entropy_loss', 'f_bet
a': '1.0', 'target_recall': '0.8', 'target_precision': '0.8', 'num_models':
'auto', 'num_calibration_samples': '10000000', 'init_method': 'uniform', 'i
nit_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0', 'optimizer':
'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5', 'loss_insensiti
vity': '0.01', 'huber_delta': '1.0', 'num_classes': '1', 'accuracy_top_k':
'3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto', 'learning_rate': 'aut
o', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult': 'auto', 'bias_wd_mu
lt': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step': 'auto', 'lr_s
cheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto', 'positive_exam
ple_weight_mult': '1.0', 'balance_multiclass_weights': 'false', 'normalize_
data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto', 'unbias_la
bel': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto', '_num_gp
us': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info', '_tuning_obje
ctive_metric': '', 'early_stopping_patience': '3', 'early_stopping_toleranc
e': '0.001', '_enable_profiler': 'false', 'predictor_type': 'binary_classif
ier'}
/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: Syntax
Warning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743: Syntax
```

```
Warning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[11/04/2024 00:38:38 WARNING 140618102482752] Loggers have already been set
up.
[11/04/2024 00:38:38 INFO 140618102482752] Final configuration: {'mini_batc
h_size': '1000', 'epochs': '15', 'feature_dim': '93', 'use_bias': 'true',
'binary_classifier_model_selection_criteria': 'cross_entropy_loss', 'f_bet
a': '1.0', 'target_recall': '0.8', 'target_precision': '0.8', 'num_models':
'auto', 'num_calibration_samples': '10000000', 'init_method': 'uniform', 'i
nit_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0', 'optimizer':
'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5', 'loss_insensiti
vity': '0.01', 'huber_delta': '1.0', 'num_classes': '1', 'accuracy_top_k':
'3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto', 'learning_rate': 'aut
o', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_lr_mult': 'auto', 'bias_wd_mu
lt': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step': 'auto', 'lr_s
cheduler_factor': 'auto', 'lr_scheduler_minimum_lr': 'auto', 'positive_exam
ple_weight_mult': '1.0', 'balance_multiclass_weights': 'false', 'normalize_
data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto', 'unbias_la
bel': 'auto', 'num_point_for_scaler': '10000', '_kvstore': 'auto', '_num_gp
us': 'auto', '_num_kv_servers': 'auto', '_log_level': 'info', '_tuning_obje
ctive_metric': '', 'early_stopping_patience': '3', 'early_stopping_toleranc
e': '0.001', '_enable_profiler': 'false', 'predictor_type': 'binary_classif
ier'}
[11/04/2024 00:38:38 WARNING 140618102482752] Loggers have already been set
up.
Process 7 is a worker.
[11/04/2024 00:38:38 INFO 140618102482752] Using default worker.
[11/04/2024 00:38:38 INFO 140618102482752] Checkpoint loading and saving ar
e disabled.
[2024-11-04 00:38:38.398] [tensorio] [warning] TensorIO is already initiali
zed; ignoring the initialization routine.
[2024-11-04 00:38:38.403] [tensorio] [warning] TensorIO is already initiali
zed; ignoring the initialization routine.
[2024-11-04 00:38:38.467] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 0, "duration": 71, "num_examples": 1,
"num_bytes": 420000}
[11/04/2024 00:38:38 INFO 140618102482752] Create Store: local
[2024-11-04 00:38:38.612] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 1, "duration": 142, "num_examples": 1
1, "num_bytes": 4620000}
[11/04/2024 00:38:38 INFO 140618102482752] Scaler algorithm parameters
 <algorithm.scaler.ScalerAlgorithmStable object at 0x7fe3a3be1b50>
[11/04/2024 00:38:38 INFO 140618102482752] Scaling model computed with para
meters:
 {'stdev_label': None, 'stdev_weight':
[5.3317194e+02 4.3147990e-01 4.4152659e-01 4.3259200e-01 2.5453323e-01
 2.8009540e-01 2.6702368e-01 2.7969170e-01 2.7820411e-01 2.9119530e-01
 2.8618178e-01 2.7408755e-01 2.7546960e-01 2.7588221e-01 2.7670491e-01
 1.7208263e-01 1.7527311e-01 1.8099521e-01 1.7982028e-01 1.7208262e-01
 1.7158580e-01 1.7863649e-01 1.8606351e-01 1.8331908e-01 1.8308823e-01
 1.8005601e-01 1.7911111e-01 1.7744364e-01 1.7600000e-01 1.8192884e-01
 1.8005599e-01 1.7816044e-01 1.8146273e-01 1.8354958e-01 1.7982030e-01
 1.7429848e-01 1.6932964e-01 1.7792189e-01 1.7624155e-01 1.8122913e-01
 1.7356344e-01 1.7575806e-01 1.7551577e-01 1.6907682e-01 1.4062189e-01
 3.5108975e-01 3.5264993e-01 3.5356134e-01 3.6025023e-01 3.2718077e-01
 3.4613615e-01 3.6457047e-01 2.3664276e-01 4.5628756e-01 3.2346731e-01
```

```
 2.3647349e-01 3.2357442e-01 3.2906225e-01 2.7255571e-01 3.5635710e-01
 3.3809501e-01 3.0048427e-01 3.2591587e-01 2.3748682e-01 3.1814224e-01
 3.2528019e-01 2.6615822e-01 3.6448511e-01 3.3590293e-01 3.0264831e-01
 3.3020279e-01 6.0192529e-02 1.0000000e+00 1.0000000e+00 1.6575097e-01
 2.3062257e-01 2.5437945e-01 2.5329944e-01 2.3460098e-01 2.4732620e-01
 2.4115555e-01 2.5033662e-01 2.4115555e-01 2.4394146e-01 2.3613445e-01
 2.2780272e-01 2.4748586e-01 2.1105410e-01 2.1850717e-01 1.9272673e-01
 1.6234834e-01 1.7356344e-01 1.0929190e-01]
<NDArray 93 @cpu(0)>, 'mean_label': None, 'mean_weight':
[1.00127411e+03 2.47363687e-01 2.65363634e-01 2.49272749e-01
 6.96363747e-02 8.58182013e-02 7.72727355e-02 8.55454504e-02
 8.45454559e-02 9.35454667e-02 9.00000185e-02 8.18181783e-02
 8.27272758e-02 8.29999968e-02 8.35454613e-02 3.05454563e-02
 3.17272730e-02 3.39090899e-02 3.34545486e-02 3.05454563e-02
 3.03636417e-02 3.30000035e-02 3.59090939e-02 3.48181874e-02
 3.47272791e-02 3.35454568e-02 3.31818201e-02 3.25454585e-02
 3.20000015e-02 3.42727304e-02 3.35454568e-02 3.28181870e-02
 3.40909064e-02 3.49090956e-02 3.34545448e-02 3.13636400e-02
 2.95454562e-02 3.27272750e-02 3.20909098e-02 3.40000018e-02
 3.10909152e-02 3.19090895e-02 3.18181850e-02 2.94545460e-02
 2.01818235e-02 1.44000024e-01 1.45545483e-01 1.46454558e-01
 1.53272748e-01 1.21909097e-01 1.39181823e-01 1.57818198e-01
 5.95454574e-02 2.95545459e-01 1.18727282e-01 5.94545491e-02
 1.18818194e-01 1.23545475e-01 8.08181912e-02 1.49272740e-01
 1.31636381e-01 1.00363642e-01 1.20818190e-01 6.00000098e-02
 1.14272736e-01 1.20272733e-01 7.67272860e-02 1.57727271e-01
 1.29636362e-01 1.02000013e-01 1.24545477e-01 3.63636389e-03
 0.00000000e+00 0.00000000e+00 2.82727312e-02 5.63636422e-02
 6.95454627e-02 6.89091012e-02 5.84545508e-02 6.54545575e-02
 6.20000064e-02 6.71818256e-02 6.19999990e-02 6.35454580e-02
 5.92727326e-02 5.49090914e-02 6.55454546e-02 4.67272773e-02
 5.02727292e-02 3.86363640e-02 2.70909127e-02 3.10909115e-02
 1.20909112e-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)
[11/04/2024 00:38:38 INFO 140618102482752] nvidia-smi: took 0.039 seconds t
o run.
[11/04/2024 00:38:38 INFO 140618102482752] nvidia-smi identified 0 GPUs.
[11/04/2024 00:38:38 INFO 140618102482752] Number of GPUs being used: 0
#metrics {"StartTime": 1730680718.7337193, "EndTime": 1730680718.733758, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "Meta": "init_train_data_iter"}, "Metrics": {"Total Records See
n": {"sum": 12000.0, "count": 1, "min": 12000, "max": 12000}, "Total Batche
s 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, "ma
x": 11}, "Reset Count": {"sum": 2.0, "count": 1, "min": 2, "max": 2}, "Numb
er 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}}}
[2024-11-04 00:39:22.307] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 3, "duration": 43572, "num_examples":
1309, "num_bytes": 549558240}
```

#metrics {"StartTime": 1730680762.3072588, "EndTime": 1730680762.3073542, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 0}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4941434306806745, "count": 1, "min": 0.4941434306806745, "max": 0.4941434306806745}}}
#metrics {"StartTime": 1730680762.3074558, "EndTime": 1730680762.307474, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 1}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4947723681481971, "count": 1, "min": 0.4947723681481971, "max": 0.4947723681481971}}}
#metrics {"StartTime": 1730680762.307516, "EndTime": 1730680762.307527, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 2}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4942423890840023, "count": 1, "min": 0.4942423890840023, "max": 0.4942423890840023}}}
#metrics {"StartTime": 1730680762.3075688, "EndTime": 1730680762.3075788, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 3}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.494788185656253, "count": 1, "min": 0.494788185656253, "max": 0.494788185656253}}}
#metrics {"StartTime": 1730680762.3076165, "EndTime": 1730680762.3076262, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 4}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5072997827792386, "count": 1, "min": 0.5072997827792386, "max": 0.5072997827792386}}}
#metrics {"StartTime": 1730680762.3076599, "EndTime": 1730680762.3076692, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 5}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5174161857115019, "count": 1, "min": 0.5174161857115019, "max": 0.5174161857115019}}}
#metrics {"StartTime": 1730680762.3077018, "EndTime": 1730680762.3077111, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 6}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5073093329613362, "count": 1, "min": 0.5073093329613362, "max": 0.5073093329613362}}}
#metrics {"StartTime": 1730680762.3077435, "EndTime": 1730680762.3077524, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 7}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5174124995240378, "count": 1, "min": 0.5174124995240378, "max": 0.5174124995240378}}}
#metrics {"StartTime": 1730680762.3077853, "EndTime": 1730680762.307794, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 8}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4942933858468992, "count": 1, "min": 0.4942933858468992, "max": 0.4942933858468992}}}
#metrics {"StartTime": 1730680762.3078303, "EndTime": 1730680762.3078396, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 9}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49479478099951324, "count": 1, "min": 0.49479478099951324, "max": 0.49479478099951324}}}
#metrics {"StartTime": 1730680762.3078845, "EndTime": 1730680762.3078952, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 10}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4943477173551507, "count": 1, "min": 0.4943477173551507, "max": 0.4943477173551507}}}
#metrics {"StartTime": 1730680762.3079333, "EndTime": 1730680762.3079429,

"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 11}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.49480523096020435, "count": 1,
"min": 0.49480523096020435, "max": 0.49480523096020435}}}
#metrics {"StartTime": 1730680762.3079765, "EndTime": 1730680762.307986, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 12}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5065179613912506, "count": 1, "mi
n": 0.5065179613912506, "max": 0.5065179613912506}}}
#metrics {"StartTime": 1730680762.3080184, "EndTime": 1730680762.3080277,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 13}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5158103942171149, "count": 1,
"min": 0.5158103942171149, "max": 0.5158103942171149}}}
#metrics {"StartTime": 1730680762.3080633, "EndTime": 1730680762.308073, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 14}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5064192143011531, "count": 1, "mi
n": 0.5064192143011531, "max": 0.5064192143011531}}}
#metrics {"StartTime": 1730680762.3081143, "EndTime": 1730680762.3081248,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 15}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5157993875206064, "count": 1,
"min": 0.5157993875206064, "max": 0.5157993875206064}}}
#metrics {"StartTime": 1730680762.308167, "EndTime": 1730680762.3081772, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 16}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5092103479601192, "count": 1, "mi
n": 0.5092103479601192, "max": 0.5092103479601192}}}
#metrics {"StartTime": 1730680762.3082185, "EndTime": 1730680762.3082285,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 17}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5092988991460304, "count": 1,
"min": 0.5092988991460304, "max": 0.5092988991460304}}}
#metrics {"StartTime": 1730680762.3082657, "EndTime": 1730680762.3082755,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 18}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5091715319572239, "count": 1,
"min": 0.5091715319572239, "max": 0.5091715319572239}}}
#metrics {"StartTime": 1730680762.3083687, "EndTime": 1730680762.3083825,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 19}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.509289886894576, "count": 1, "m
in": 0.509289886894576, "max": 0.509289886894576}}}
#metrics {"StartTime": 1730680762.3084269, "EndTime": 1730680762.3084373,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 20}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5105951475324252, "count": 1,
"min": 0.5105951475324252, "max": 0.5105951475324252}}}
#metrics {"StartTime": 1730680762.308475, "EndTime": 1730680762.3084846, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 21}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5123595978623137, "count": 1, "mi
n": 0.5123595978623137, "max": 0.5123595978623137}}}
#metrics {"StartTime": 1730680762.308525, "EndTime": 1730680762.3085353, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":

"training", "epoch": 0, "model": 22}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5105959102385634, "count": 1, "mi
n": 0.5105959102385634, "max": 0.5105959102385634}}}
#metrics {"StartTime": 1730680762.3085768, "EndTime": 1730680762.3085868,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 23}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5123502058209993, "count": 1,
"min": 0.5123502058209993, "max": 0.5123502058209993}}}
#metrics {"StartTime": 1730680762.3086288, "EndTime": 1730680762.3086388,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 24}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5150498709722396, "count": 1,
"min": 0.5150498709722396, "max": 0.5150498709722396}}}
#metrics {"StartTime": 1730680762.3086762, "EndTime": 1730680762.308686, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 25}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5151045057197594, "count": 1, "mi
n": 0.5151045057197594, "max": 0.5151045057197594}}}
#metrics {"StartTime": 1730680762.3087275, "EndTime": 1730680762.3087375,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 26}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5150449723176636, "count": 1,
"min": 0.5150449723176636, "max": 0.5150449723176636}}}
#metrics {"StartTime": 1730680762.3087862, "EndTime": 1730680762.3087962,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 27}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5151026800079812, "count": 1,
"min": 0.5151026800079812, "max": 0.5151026800079812}}}
#metrics {"StartTime": 1730680762.3088443, "EndTime": 1730680762.3088593,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 28}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5150842277923491, "count": 1,
"min": 0.5150842277923491, "max": 0.5150842277923491}}}
#metrics {"StartTime": 1730680762.3089166, "EndTime": 1730680762.3089316,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 29}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5158147742828463, "count": 1,
"min": 0.5158147742828463, "max": 0.5158147742828463}}}
#metrics {"StartTime": 1730680762.3089848, "EndTime": 1730680762.308999, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5150950631657871, "count": 1, "mi
n": 0.5150950631657871, "max": 0.5150950631657871}}}
#metrics {"StartTime": 1730680762.309059, "EndTime": 1730680762.3090746, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5158025581756498, "count": 1, "mi
n": 0.5158025581756498, "max": 0.5158025581756498}}}
[11/04/2024 00:39:22 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=0, train binary_classification_cross_entropy_objective <loss>=0.4941434
306806745
[2024-11-04 00:39:22.331] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 0, "duration": 43932, "num_exampl
es": 1, "num_bytes": 420000}
[2024-11-04 00:39:26.519] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 2, "duration": 4188, "num_example

s": 164, "num_bytes": 68694780}
#metrics {"StartTime": 1730680766.5270057, "EndTime": 1730680766.5270574,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 0}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49259978556004874, "count":
1, "min": 0.49259978556004874, "max": 0.49259978556004874}}}
#metrics {"StartTime": 1730680766.527137, "EndTime": 1730680766.5271568, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 1}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49387791044541124, "count": 1,
"min": 0.49387791044541124, "max": 0.49387791044541124}}}
#metrics {"StartTime": 1730680766.5272171, "EndTime": 1730680766.5272348,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 2}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4926038146662553, "count":
1, "min": 0.4926038146662553, "max": 0.4926038146662553}}}
#metrics {"StartTime": 1730680766.5272806, "EndTime": 1730680766.5272963,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 3}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49387066891345877, "count":
1, "min": 0.49387066891345877, "max": 0.49387066891345877}}}
#metrics {"StartTime": 1730680766.5273461, "EndTime": 1730680766.5273614,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 4}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5003789656689611, "count":
1, "min": 0.5003789656689611, "max": 0.5003789656689611}}}
#metrics {"StartTime": 1730680766.5274153, "EndTime": 1730680766.5274315,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 5}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5154898049589283, "count":
1, "min": 0.5154898049589283, "max": 0.5154898049589283}}}
#metrics {"StartTime": 1730680766.527474, "EndTime": 1730680766.5274875, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 6}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.5003781808944541, "count": 1,
"min": 0.5003781808944541, "max": 0.5003781808944541}}}
#metrics {"StartTime": 1730680766.5275369, "EndTime": 1730680766.5275514,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 7}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5155003803831516, "count":
1, "min": 0.5155003803831516, "max": 0.5155003803831516}}}
#metrics {"StartTime": 1730680766.5276017, "EndTime": 1730680766.5276158,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 8}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4927659734100276, "count":
1, "min": 0.4927659734100276, "max": 0.4927659734100276}}}
#metrics {"StartTime": 1730680766.52767, "EndTime": 1730680766.5276852, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 9}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4940258389473084, "count": 1,
"min": 0.4940258389473084, "max": 0.4940258389473084}}}
#metrics {"StartTime": 1730680766.5277352, "EndTime": 1730680766.5277503,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 10}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.4927652188622134, "count":
1, "min": 0.4927652188622134, "max": 0.4927652188622134}}}

#metrics {"StartTime": 1730680766.5277967, "EndTime": 1730680766.5278118, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 11}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.49402681870264276, "count": 1, "min": 0.49402681870264276, "max": 0.49402681870264276}}}
#metrics {"StartTime": 1730680766.5278533, "EndTime": 1730680766.5278678, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 12}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5004263294039847, "count": 1, "min": 0.5004263294039847, "max": 0.5004263294039847}}}
#metrics {"StartTime": 1730680766.5279086, "EndTime": 1730680766.5279222, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 13}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5140210153194193, "count": 1, "min": 0.5140210153194193, "max": 0.5140210153194193}}}
#metrics {"StartTime": 1730680766.5279737, "EndTime": 1730680766.527989, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 14}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5004321709922552, "count": 1, "min": 0.5004321709922552, "max": 0.5004321709922552}}}
#metrics {"StartTime": 1730680766.5280294, "EndTime": 1730680766.5280433, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 15}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5140275299180671, "count": 1, "min": 0.5140275299180671, "max": 0.5140275299180671}}}
#metrics {"StartTime": 1730680766.5280898, "EndTime": 1730680766.5281034, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 16}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5080263672860166, "count": 1, "min": 0.5080263672860166, "max": 0.5080263672860166}}}
#metrics {"StartTime": 1730680766.528147, "EndTime": 1730680766.5281615, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 17}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.508363059992101, "count": 1, "min": 0.508363059992101, "max": 0.508363059992101}}}
#metrics {"StartTime": 1730680766.5282128, "EndTime": 1730680766.5282278, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 18}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5080193914504261, "count": 1, "min": 0.5080193914504261, "max": 0.5080193914504261}}}
#metrics {"StartTime": 1730680766.5282714, "EndTime": 1730680766.5282853, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 19}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5083601242712038, "count": 1, "min": 0.5083601242712038, "max": 0.5083601242712038}}}
#metrics {"StartTime": 1730680766.5283344, "EndTime": 1730680766.528349, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 20}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5102375767456477, "count": 1, "min": 0.5102375767456477, "max": 0.5102375767456477}}}
#metrics {"StartTime": 1730680766.5283952, "EndTime": 1730680766.5284095, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 21}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.509569687180627, "count": 1, "min": 0.509569687180627, "max": 0.509569687180627}}}
#metrics {"StartTime": 1730680766.5284524, "EndTime": 1730680766.528466, "D

imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 22}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.510237523382474, "count": 1,
"min": 0.510237523382474, "max": 0.510237523382474}}}
#metrics {"StartTime": 1730680766.5285099, "EndTime": 1730680766.528524, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 23}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5095846822324651, "count": 1,
"min": 0.5095846822324651, "max": 0.5095846822324651}}}
#metrics {"StartTime": 1730680766.5285685, "EndTime": 1730680766.5285828,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 24}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5137468590286633, "count":
1, "min": 0.5137468590286633, "max": 0.5137468590286633}}}
#metrics {"StartTime": 1730680766.5286293, "EndTime": 1730680766.5286434,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 25}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5144164743139445, "count":
1, "min": 0.5144164743139445, "max": 0.5144164743139445}}}
#metrics {"StartTime": 1730680766.5286999, "EndTime": 1730680766.5287147,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 26}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5137622281824716, "count":
1, "min": 0.5137622281824716, "max": 0.5137622281824716}}}
#metrics {"StartTime": 1730680766.5287573, "EndTime": 1730680766.5287716,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 27}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.514421754842552, "count":
1, "min": 0.514421754842552, "max": 0.514421754842552}}}
#metrics {"StartTime": 1730680766.5288203, "EndTime": 1730680766.5288353,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 28}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5153283475863808, "count":
1, "min": 0.5153283475863808, "max": 0.5153283475863808}}}
#metrics {"StartTime": 1730680766.5288804, "EndTime": 1730680766.5288947,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "model": 29}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.513802403562132, "count":
1, "min": 0.513802403562132, "max": 0.513802403562132}}}
#metrics {"StartTime": 1730680766.528943, "EndTime": 1730680766.5289571, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 30}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5153278033193249, "count": 1,
"min": 0.5153278033193249, "max": 0.5153278033193249}}}
#metrics {"StartTime": 1730680766.5290053, "EndTime": 1730680766.529021, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 0, "model": 31}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5138155469491561, "count": 1,
"min": 0.5138155469491561, "max": 0.5138155469491561}}}
[11/04/2024 00:39:26 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=0, validation binary_classification_cross_entropy_objective <loss>=0.49
259978556004874
[11/04/2024 00:39:26 INFO 140618102482752] #early_stopping_criteria_metric:
host=algo-1, epoch=0, criteria=binary_classification_cross_entropy_objectiv
e, value=0.49259978556004874
[11/04/2024 00:39:26 INFO 140618102482752] Epoch 0: Loss improved. Updating

```
best model
[11/04/2024 00:39:26 INFO 140618102482752] Saving model for epoch: 0
[11/04/2024 00:39:26 INFO 140618102482752] Saved checkpoint to "/tmp/tmpn7r
123g8/mx-mod-0000.params"
[11/04/2024 00:39:26 INFO 140618102482752] #progress_metric: host=algo-1, c
ompleted 6.666666666666667 % of epochs
#metrics {"StartTime": 1730680718.7340493, "EndTime": 1730680766.5365672,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 0, "Meta": "training_data_iter"}, "Metrics": {"Tot
al Records Seen": {"sum": 1320472.0, "count": 1, "min": 1320472, "max": 132
0472}, "Total Batches Seen": {"sum": 1321.0, "count": 1, "min": 1321, "ma
x": 1321}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count":
1, "min": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"su
m": 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}}}
[11/04/2024 00:39:26 INFO 140618102482752] #throughput_metric: host=algo-1,
train throughput=27372.371264347912 records/second
[2024-11-04 00:40:08.764] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 5, "duration": 42209, "num_examples":
1309, "num_bytes": 549558240}
#metrics {"StartTime": 1730680808.7648804, "EndTime": 1730680808.7649906,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 0}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49205524920542304, "count": 1,
"min": 0.49205524920542304, "max": 0.49205524920542304}}}
#metrics {"StartTime": 1730680808.7650743, "EndTime": 1730680808.765094, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 1}, "Metrics": {"train_binary_classificati
on_cross_entropy_objective": {"sum": 0.49305095401962235, "count": 1, "mi
n": 0.49305095401962235, "max": 0.49305095401962235}}}
#metrics {"StartTime": 1730680808.7651455, "EndTime": 1730680808.7651615,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 2}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49206523955925524, "count": 1,
"min": 0.49206523955925524, "max": 0.49206523955925524}}}
#metrics {"StartTime": 1730680808.7652223, "EndTime": 1730680808.7652395,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 3}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49304697280557147, "count": 1,
"min": 0.49304697280557147, "max": 0.49304697280557147}}}
#metrics {"StartTime": 1730680808.7652867, "EndTime": 1730680808.7653024,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 4}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49603806848555165, "count": 1,
"min": 0.49603806848555165, "max": 0.49603806848555165}}}
#metrics {"StartTime": 1730680808.7653801, "EndTime": 1730680808.765396, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 5}, "Metrics": {"train_binary_classificati
on_cross_entropy_objective": {"sum": 0.5138299025555999, "count": 1, "min":
0.5138299025555999, "max": 0.5138299025555999}}}
#metrics {"StartTime": 1730680808.7654498, "EndTime": 1730680808.7654655,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 6}, "Metrics": {"train_binary_classifi
```

cation_cross_entropy_objective": {"sum": 0.4960379461819243, "count": 1, "m
in": 0.4960379461819243, "max": 0.4960379461819243}}}
#metrics {"StartTime": 1730680808.7655087, "EndTime": 1730680808.7655241,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 7}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.5138288244311598, "count": 1, "m
in": 0.5138288244311598, "max": 0.5138288244311598}}}
#metrics {"StartTime": 1730680808.7655668, "EndTime": 1730680808.765581, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 8}, "Metrics": {"train_binary_classificati
on_cross_entropy_objective": {"sum": 0.49223377778406174, "count": 1, "mi
n": 0.49223377778406174, "max": 0.49223377778406174}}}
#metrics {"StartTime": 1730680808.7656362, "EndTime": 1730680808.765652, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 9}, "Metrics": {"train_binary_classificati
on_cross_entropy_objective": {"sum": 0.49318936094231564, "count": 1, "mi
n": 0.49318936094231564, "max": 0.49318936094231564}}}
#metrics {"StartTime": 1730680808.765694, "EndTime": 1730680808.7657084, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 10}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.4922331574699565, "count": 1, "mi
n": 0.4922331574699565, "max": 0.4922331574699565}}}
#metrics {"StartTime": 1730680808.7657514, "EndTime": 1730680808.7657652,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 11}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.4931893890100882, "count": 1,
"min": 0.4931893890100882, "max": 0.4931893890100882}}}
#metrics {"StartTime": 1730680808.765818, "EndTime": 1730680808.7658336, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.4959937975355609, "count": 1, "mi
n": 0.4959937975355609, "max": 0.4959937975355609}}}
#metrics {"StartTime": 1730680808.76588, "EndTime": 1730680808.7658951, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5125046441081103, "count": 1, "mi
n": 0.5125046441081103, "max": 0.5125046441081103}}}
#metrics {"StartTime": 1730680808.7659445, "EndTime": 1730680808.7659597,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 14}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.4959953844962864, "count": 1,
"min": 0.4959953844962864, "max": 0.4959953844962864}}}
#metrics {"StartTime": 1730680808.7660065, "EndTime": 1730680808.7660205,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 15}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5125044446706043, "count": 1,
"min": 0.5125044446706043, "max": 0.5125044446706043}}}
#metrics {"StartTime": 1730680808.7660723, "EndTime": 1730680808.7660866,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 16}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5077906266192048, "count": 1,
"min": 0.5077906266192048, "max": 0.5077906266192048}}}
#metrics {"StartTime": 1730680808.7661319, "EndTime": 1730680808.7661476,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 17}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5080331715108423, "count": 1,

"min": 0.5080331715108423, "max": 0.5080331715108423}}}
#metrics {"StartTime": 1730680808.7661936, "EndTime": 1730680808.7662082,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 18}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5077905497419725, "count": 1,
"min": 0.5077905497419725, "max": 0.5077905497419725}}}
#metrics {"StartTime": 1730680808.7662578, "EndTime": 1730680808.7662723,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 19}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5080334153714895, "count": 1,
"min": 0.5080334153714895, "max": 0.5080334153714895}}}
#metrics {"StartTime": 1730680808.7663271, "EndTime": 1730680808.7663414,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 20}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5086412821311834, "count": 1,
"min": 0.5086412821311834, "max": 0.5086412821311834}}}
#metrics {"StartTime": 1730680808.7663877, "EndTime": 1730680808.7664018,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 21}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5115468648274739, "count": 1,
"min": 0.5115468648274739, "max": 0.5115468648274739}}}
#metrics {"StartTime": 1730680808.7664475, "EndTime": 1730680808.7664618,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 22}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5086414054147329, "count": 1,
"min": 0.5086414054147329, "max": 0.5086414054147329}}}
#metrics {"StartTime": 1730680808.7665083, "EndTime": 1730680808.7665334,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 23}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5115463972179168, "count": 1,
"min": 0.5115463972179168, "max": 0.5115463972179168}}}
#metrics {"StartTime": 1730680808.7665882, "EndTime": 1730680808.7666032,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 24}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137818282707751, "count": 1,
"min": 0.5137818282707751, "max": 0.5137818282707751}}}
#metrics {"StartTime": 1730680808.7666528, "EndTime": 1730680808.7666678,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 25}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5138786307052006, "count": 1,
"min": 0.5138786307052006, "max": 0.5138786307052006}}}
#metrics {"StartTime": 1730680808.766714, "EndTime": 1730680808.7667284, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 26}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5137812400841203, "count": 1, "mi
n": 0.5137812400841203, "max": 0.5137812400841203}}}
#metrics {"StartTime": 1730680808.7667744, "EndTime": 1730680808.7667894,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 27}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5138793853386462, "count": 1,
"min": 0.5138793853386462, "max": 0.5138793853386462}}}
#metrics {"StartTime": 1730680808.766846, "EndTime": 1730680808.7668607, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 28}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5141613477421099, "count": 1, "mi
n": 0.5141613477421099, "max": 0.5141613477421099}}}

#metrics {"StartTime": 1730680808.7669156, "EndTime": 1730680808.7669315,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 29}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5152496174803568, "count": 1,
"min": 0.5152496174803568, "max": 0.5152496174803568}}}
#metrics {"StartTime": 1730680808.7669888, "EndTime": 1730680808.7670062,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 30}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5141608154343538, "count": 1,
"min": 0.5141608154343538, "max": 0.5141608154343538}}}
#metrics {"StartTime": 1730680808.7670634, "EndTime": 1730680808.7670796,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 31}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5152500363504485, "count": 1,
"min": 0.5152500363504485, "max": 0.5152500363504485}}}
[11/04/2024 00:40:08 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=1, train binary_classification_cross_entropy_objective <loss>=0.4920552
4920542304
[2024-11-04 00:40:12.517] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 5, "duration": 3730, "num_example
s": 164, "num_bytes": 68694780}
#metrics {"StartTime": 1730680812.5246232, "EndTime": 1730680812.5246792,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 0}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4922615350597754, "count":
1, "min": 0.4922615350597754, "max": 0.4922615350597754}}}
#metrics {"StartTime": 1730680812.5247633, "EndTime": 1730680812.5247846,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 1}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49387464017846294, "count":
1, "min": 0.49387464017846294, "max": 0.49387464017846294}}}
#metrics {"StartTime": 1730680812.5248518, "EndTime": 1730680812.52487, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 2}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4922655835707723, "count": 1,
"min": 0.4922655835707723, "max": 0.4922655835707723}}}
#metrics {"StartTime": 1730680812.524932, "EndTime": 1730680812.524948, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 3}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49387276481741443, "count": 1,
"min": 0.49387276481741443, "max": 0.49387276481741443}}}
#metrics {"StartTime": 1730680812.5250094, "EndTime": 1730680812.5250258,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 4}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49394108908467044, "count":
1, "min": 0.49394108908467044, "max": 0.49394108908467044}}}
#metrics {"StartTime": 1730680812.5250793, "EndTime": 1730680812.5250945,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 5}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5137162311790847, "count":
1, "min": 0.5137162311790847, "max": 0.5137162311790847}}}
#metrics {"StartTime": 1730680812.525146, "EndTime": 1730680812.5251615, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 6}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49394053231645146, "count": 1,
"min": 0.49394053231645146, "max": 0.49394053231645146}}}

#metrics {"StartTime": 1730680812.525213, "EndTime": 1730680812.525229, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 7}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137159063354289, "count": 1,
"min": 0.5137159063354289, "max": 0.5137159063354289}}}
#metrics {"StartTime": 1730680812.5252802, "EndTime": 1730680812.5252957,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 8}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49246732062372894, "count":
1, "min": 0.49246732062372894, "max": 0.49246732062372894}}}
#metrics {"StartTime": 1730680812.5253673, "EndTime": 1730680812.5253828,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 9}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49405164769180676, "count":
1, "min": 0.49405164769180676, "max": 0.49405164769180676}}}
#metrics {"StartTime": 1730680812.5254343, "EndTime": 1730680812.5254498,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 10}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.4924669080853469, "count":
1, "min": 0.4924669080853469, "max": 0.4924669080853469}}}
#metrics {"StartTime": 1730680812.5254874, "EndTime": 1730680812.5254955,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 11}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.49405181076668053, "coun
t": 1, "min": 0.49405181076668053, "max": 0.49405181076668053}}}
#metrics {"StartTime": 1730680812.5255342, "EndTime": 1730680812.525545, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 12}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.4940839896934053, "count": 1,
"min": 0.4940839896934053, "max": 0.4940839896934053}}}
#metrics {"StartTime": 1730680812.525587, "EndTime": 1730680812.5256004, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 13}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5122471458031056, "count": 1,
"min": 0.5122471458031056, "max": 0.5122471458031056}}}
#metrics {"StartTime": 1730680812.525637, "EndTime": 1730680812.52565, "Dim
ensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "t
raining", "epoch": 1, "model": 14}, "Metrics": {"validation_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49408288175450304, "count": 1,
"min": 0.49408288175450304, "max": 0.49408288175450304}}}
#metrics {"StartTime": 1730680812.52569, "EndTime": 1730680812.5257032, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 15}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5122483317343388, "count": 1,
"min": 0.5122483317343388, "max": 0.5122483317343388}}}
#metrics {"StartTime": 1730680812.5257485, "EndTime": 1730680812.5257618,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 16}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5079445505320925, "count":
1, "min": 0.5079445505320925, "max": 0.5079445505320925}}}
#metrics {"StartTime": 1730680812.5258052, "EndTime": 1730680812.5258193,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 17}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5085199012635301, "count":
1, "min": 0.5085199012635301, "max": 0.5085199012635301}}}
#metrics {"StartTime": 1730680812.525864, "EndTime": 1730680812.5258787, "D

imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 18}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5079451453635543, "count": 1,
"min": 0.5079451453635543, "max": 0.5079451453635543}}}
#metrics {"StartTime": 1730680812.5259283, "EndTime": 1730680812.525943, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 19}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5085199260792718, "count": 1,
"min": 0.5085199260792718, "max": 0.5085199260792718}}}
#metrics {"StartTime": 1730680812.5259886, "EndTime": 1730680812.526002, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 1, "model": 20}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5080180045676608, "count": 1,
"min": 0.5080180045676608, "max": 0.5080180045676608}}}
#metrics {"StartTime": 1730680812.5261054, "EndTime": 1730680812.5261226,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 21}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5099187176017618, "count":
1, "min": 0.5099187176017618, "max": 0.5099187176017618}}}
#metrics {"StartTime": 1730680812.5261848, "EndTime": 1730680812.5262024,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 22}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5080185917491571, "count":
1, "min": 0.5080185917491571, "max": 0.5080185917491571}}}
#metrics {"StartTime": 1730680812.5262604, "EndTime": 1730680812.5262775,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 23}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5099185539671345, "count":
1, "min": 0.5099185539671345, "max": 0.5099185539671345}}}
#metrics {"StartTime": 1730680812.5263376, "EndTime": 1730680812.5263538,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 24}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5137536716027956, "count":
1, "min": 0.5137536716027956, "max": 0.5137536716027956}}}
#metrics {"StartTime": 1730680812.5264125, "EndTime": 1730680812.5264287,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 25}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.514552243848765, "count":
1, "min": 0.514552243848765, "max": 0.514552243848765}}}
#metrics {"StartTime": 1730680812.5264847, "EndTime": 1730680812.5264995,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 26}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5137697210437776, "count":
1, "min": 0.5137697210437776, "max": 0.5137697210437776}}}
#metrics {"StartTime": 1730680812.5265515, "EndTime": 1730680812.5265665,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 27}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5145766780250767, "count":
1, "min": 0.5145766780250767, "max": 0.5145766780250767}}}
#metrics {"StartTime": 1730680812.5266209, "EndTime": 1730680812.5266366,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 1, "model": 28}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5139145938509287, "count":
1, "min": 0.5139145938509287, "max": 0.5139145938509287}}}
#metrics {"StartTime": 1730680812.5266855, "EndTime": 1730680812.5267005,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio

n": "training", "epoch": 1, "model": 29}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5135515399235712, "count": 1, "min": 0.5135515399235712, "max": 0.5135515399235712}}}
#metrics {"StartTime": 1730680812.5267608, "EndTime": 1730680812.5267768, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 30}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5139170920311177, "count": 1, "min": 0.5139170920311177, "max": 0.5139170920311177}}}
#metrics {"StartTime": 1730680812.526838, "EndTime": 1730680812.5268548, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 1, "model": 31}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5135692029475148, "count": 1, "min": 0.5135692029475148, "max": 0.5135692029475148}}}
[11/04/2024 00:40:12 INFO 140618102482752] #quality_metric: host=algo-1, epoch=1, validation binary_classification_cross_entropy_objective <loss>=0.4922615350597754
[11/04/2024 00:40:12 INFO 140618102482752] #early_stopping_criteria_metric: host=algo-1, epoch=1, criteria=binary_classification_cross_entropy_objective, value=0.4922615350597754
[11/04/2024 00:40:12 INFO 140618102482752] Saving model for epoch: 1
[11/04/2024 00:40:12 INFO 140618102482752] Saved checkpoint to "/tmp/tmpy8pxqfvs/mx-mod-0000.params"
[11/04/2024 00:40:12 INFO 140618102482752] #progress_metric: host=algo-1, completed 13.333333333333334 % of epochs
#metrics {"StartTime": 1730680766.5554628, "EndTime": 1730680812.5328496, "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}}}
[11/04/2024 00:40:12 INFO 140618102482752] #throughput_metric: host=algo-1, train throughput=28458.936198134823 records/second
[2024-11-04 00:40:54.802] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 7, "duration": 42268, "num_examples": 1309, "num_bytes": 549558240}
#metrics {"StartTime": 1730680854.8022742, "EndTime": 1730680854.8023663, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 0}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4918954697355218, "count": 1, "min": 0.4918954697355218, "max": 0.4918954697355218}}}
#metrics {"StartTime": 1730680854.8024564, "EndTime": 1730680854.802476, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 1}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4929053160174542, "count": 1, "min": 0.4929053160174542, "max": 0.4929053160174542}}}
#metrics {"StartTime": 1730680854.8025265, "EndTime": 1730680854.8025415, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 2}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49190450230869676, "count": 1, "min": 0.49190450230869676, "max": 0.49190450230869676}}}

#metrics {"StartTime": 1730680854.8025854, "EndTime": 1730680854.8025997, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 3}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4929039644713796, "count": 1, "min": 0.4929039644713796, "max": 0.4929039644713796}}}
#metrics {"StartTime": 1730680854.8026512, "EndTime": 1730680854.8026674, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 4}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4929917178372724, "count": 1, "min": 0.4929917178372724, "max": 0.4929917178372724}}}
#metrics {"StartTime": 1730680854.8027112, "EndTime": 1730680854.8027265, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 5}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5111829315792166, "count": 1, "min": 0.5111829315792166, "max": 0.5111829315792166}}}
#metrics {"StartTime": 1730680854.8027718, "EndTime": 1730680854.8027868, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 6}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49299169427247586, "count": 1, "min": 0.49299169427247586, "max": 0.49299169427247586}}}
#metrics {"StartTime": 1730680854.8028357, "EndTime": 1730680854.8028505, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 7}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5111826934581138, "count": 1, "min": 0.5111826934581138, "max": 0.5111826934581138}}}
#metrics {"StartTime": 1730680854.8029027, "EndTime": 1730680854.8029187, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 8}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49208999857771285, "count": 1, "min": 0.49208999857771285, "max": 0.49208999857771285}}}
#metrics {"StartTime": 1730680854.8029675, "EndTime": 1730680854.8029823, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 9}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49306037914497775, "count": 1, "min": 0.49306037914497775, "max": 0.49306037914497775}}}
#metrics {"StartTime": 1730680854.8030336, "EndTime": 1730680854.8030481, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 10}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49208966883388133, "count": 1, "min": 0.49208966883388133, "max": 0.49208966883388133}}}
#metrics {"StartTime": 1730680854.8030999, "EndTime": 1730680854.8031147, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 11}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49306038371794814, "count": 1, "min": 0.49306038371794814, "max": 0.49306038371794814}}}
#metrics {"StartTime": 1730680854.8031666, "EndTime": 1730680854.8031824, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 12}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.493145427902175, "count": 1, "min": 0.493145427902175, "max": 0.493145427902175}}}
#metrics {"StartTime": 1730680854.8032324, "EndTime": 1730680854.8032465, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 13}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5100733419295844, "count": 1, "min": 0.5100733419295844, "max": 0.5100733419295844}}}
#metrics {"StartTime": 1730680854.803293, "EndTime": 1730680854.8033075, "D

imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 14}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49314552790090577, "count": 1, "min": 0.49314552790090577, "max": 0.49314552790090577}}}
#metrics {"StartTime": 1730680854.8033557, "EndTime": 1730680854.8033705, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 15}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5100733001662321, "count": 1, "min": 0.5100733001662321, "max": 0.5100733001662321}}}
#metrics {"StartTime": 1730680854.8034167, "EndTime": 1730680854.8034315, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 16}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.507754690444433, "count": 1, "min": 0.507754690444433, "max": 0.507754690444433}}}
#metrics {"StartTime": 1730680854.8034854, "EndTime": 1730680854.8035007, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 17}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5080136292950458, "count": 1, "min": 0.5080136292950458, "max": 0.5080136292950458}}}
#metrics {"StartTime": 1730680854.8035567, "EndTime": 1730680854.8035705, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 18}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5077547050032776, "count": 1, "min": 0.5077547050032776, "max": 0.5077547050032776}}}
#metrics {"StartTime": 1730680854.8036196, "EndTime": 1730680854.8036366, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 19}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5080136809742779, "count": 1, "min": 0.5080136809742779, "max": 0.5080136809742779}}}
#metrics {"StartTime": 1730680854.803686, "EndTime": 1730680854.8037002, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 20}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5080167046876494, "count": 1, "min": 0.5080167046876494, "max": 0.5080167046876494}}}
#metrics {"StartTime": 1730680854.8037543, "EndTime": 1730680854.8037694, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 21}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5110996317732225, "count": 1, "min": 0.5110996317732225, "max": 0.5110996317732225}}}
#metrics {"StartTime": 1730680854.803819, "EndTime": 1730680854.8038332, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 22}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5080167359751664, "count": 1, "min": 0.5080167359751664, "max": 0.5080167359751664}}}
#metrics {"StartTime": 1730680854.803881, "EndTime": 1730680854.803895, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 23}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5110996113115129, "count": 1, "min": 0.5110996113115129, "max": 0.5110996113115129}}}
#metrics {"StartTime": 1730680854.803942, "EndTime": 1730680854.8039563, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 24}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5137729155280903, "count": 1, "min": 0.5137729155280903, "max": 0.5137729155280903}}}
#metrics {"StartTime": 1730680854.804, "EndTime": 1730680854.8040144, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "tr

aining", "epoch": 2, "model": 25}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5138806332462789, "count": 1, "min": 0.5138806332462789, "max": 0.5138806332462789}}}
#metrics {"StartTime": 1730680854.8040588, "EndTime": 1730680854.8040729, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 26}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5137731941059824, "count": 1, "min": 0.5137731941059824, "max": 0.5137731941059824}}}
#metrics {"StartTime": 1730680854.8041189, "EndTime": 1730680854.804133, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 27}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5138805981557304, "count": 1, "min": 0.5138805981557304, "max": 0.5138805981557304}}}
#metrics {"StartTime": 1730680854.804179, "EndTime": 1730680854.8041928, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 28}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5138744779814274, "count": 1, "min": 0.5138744779814274, "max": 0.5138744779814274}}}
#metrics {"StartTime": 1730680854.8042347, "EndTime": 1730680854.804248, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 29}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5150635678819195, "count": 1, "min": 0.5150635678819195, "max": 0.5150635678819195}}}
#metrics {"StartTime": 1730680854.8042932, "EndTime": 1730680854.8043082, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 30}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.513874470118718, "count": 1, "min": 0.513874470118718, "max": 0.513874470118718}}}
#metrics {"StartTime": 1730680854.8043513, "EndTime": 1730680854.8043654, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 31}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5150639005421863, "count": 1, "min": 0.5150639005421863, "max": 0.5150639005421863}}}
[11/04/2024 00:40:54 INFO 140618102482752] #quality_metric: host=algo-1, epoch=2, train binary_classification_cross_entropy_objective <loss>=0.4918954697355218
[2024-11-04 00:40:58.561] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/validation", "epoch": 8, "duration": 3734, "num_examples": 164, "num_bytes": 68694780}
#metrics {"StartTime": 1730680858.5687816, "EndTime": 1730680858.5688367, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 0}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.4921819552677068, "count": 1, "min": 0.4921819552677068, "max": 0.4921819552677068}}}
#metrics {"StartTime": 1730680858.5689147, "EndTime": 1730680858.56893, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 1}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.4937280978129929, "count": 1, "min": 0.4937280978129929, "max": 0.4937280978129929}}}
#metrics {"StartTime": 1730680858.5689838, "EndTime": 1730680858.5689998, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 2, "model": 2}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.4921858679451705, "count": 1, "min": 0.4921858679451705, "max": 0.4921858679451705}}}
#metrics {"StartTime": 1730680858.569052, "EndTime": 1730680858.5690675, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":

"training", "epoch": 2, "model": 3}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4937274942120585, "count": 1,
"min": 0.4937274942120585, "max": 0.4937274942120585}}}
#metrics {"StartTime": 1730680858.5691168, "EndTime": 1730680858.5691328,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 4}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.492907732867815, "count":
1, "min": 0.492907732867815, "max": 0.492907732867815}}}
#metrics {"StartTime": 1730680858.5691783, "EndTime": 1730680858.5691931,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 5}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5121777936417081, "count":
1, "min": 0.5121777936417081, "max": 0.5121777936417081}}}
#metrics {"StartTime": 1730680858.5692425, "EndTime": 1730680858.5692573,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 6}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4929077306288007, "count":
1, "min": 0.4929077306288007, "max": 0.4929077306288007}}}
#metrics {"StartTime": 1730680858.5693026, "EndTime": 1730680858.5693376,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 7}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.512177687101945, "count":
1, "min": 0.512177687101945, "max": 0.512177687101945}}}
#metrics {"StartTime": 1730680858.569385, "EndTime": 1730680858.5694008, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 8}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49237718798416447, "count": 1,
"min": 0.49237718798416447, "max": 0.49237718798416447}}}
#metrics {"StartTime": 1730680858.5694487, "EndTime": 1730680858.5694637,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 9}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4939150055011685, "count":
1, "min": 0.4939150055011685, "max": 0.4939150055011685}}}
#metrics {"StartTime": 1730680858.5695138, "EndTime": 1730680858.5695293,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 10}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.49237681910656095, "coun
t": 1, "min": 0.49237681910656095, "max": 0.49237681910656095}}}
#metrics {"StartTime": 1730680858.5695834, "EndTime": 1730680858.5695982,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 11}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.4939150491619471, "count":
1, "min": 0.4939150491619471, "max": 0.4939150491619471}}}
#metrics {"StartTime": 1730680858.569639, "EndTime": 1730680858.569648, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 12}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.49310001586953617, "count": 1,
"min": 0.49310001586953617, "max": 0.49310001586953617}}}
#metrics {"StartTime": 1730680858.5696788, "EndTime": 1730680858.5696864,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 13}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5107256801805966, "count":
1, "min": 0.5107256801805966, "max": 0.5107256801805966}}}
#metrics {"StartTime": 1730680858.569725, "EndTime": 1730680858.5697389, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 14}, "Metrics": {"validation_binary_classi

fication_cross_entropy_objective": {"sum": 0.49310009927281834, "count": 1,
"min": 0.49310009927281834, "max": 0.49310009927281834}}}
#metrics {"StartTime": 1730680858.5697834, "EndTime": 1730680858.5697923,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 15}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5107256727172157, "count":
1, "min": 0.5107256727172157, "max": 0.5107256727172157}}}
#metrics {"StartTime": 1730680858.569824, "EndTime": 1730680858.5698366, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 16}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5078876558730097, "count": 1,
"min": 0.5078876558730097, "max": 0.5078876558730097}}}
#metrics {"StartTime": 1730680858.5698922, "EndTime": 1730680858.5699065,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 17}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5085207203695895, "count":
1, "min": 0.5085207203695895, "max": 0.5085207203695895}}}
#metrics {"StartTime": 1730680858.569956, "EndTime": 1730680858.5699658, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 18}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5078879572070156, "count": 1,
"min": 0.5078879572070156, "max": 0.5078879572070156}}}
#metrics {"StartTime": 1730680858.5699952, "EndTime": 1730680858.5700076,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 19}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5085208485531574, "count":
1, "min": 0.5085208485531574, "max": 0.5085208485531574}}}
#metrics {"StartTime": 1730680858.5700607, "EndTime": 1730680858.570075, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 20}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5081749290557875, "count": 1,
"min": 0.5081749290557875, "max": 0.5081749290557875}}}
#metrics {"StartTime": 1730680858.5701237, "EndTime": 1730680858.5701373,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 21}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5101465397308965, "count":
1, "min": 0.5101465397308965, "max": 0.5101465397308965}}}
#metrics {"StartTime": 1730680858.5701709, "EndTime": 1730680858.5701787,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 22}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.508174968425122, "count":
1, "min": 0.508174968425122, "max": 0.508174968425122}}}
#metrics {"StartTime": 1730680858.570209, "EndTime": 1730680858.5702162, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 23}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5101463615426763, "count": 1,
"min": 0.5101463615426763, "max": 0.5101463615426763}}}
#metrics {"StartTime": 1730680858.5702457, "EndTime": 1730680858.570253, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 24}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5137589661252424, "count": 1,
"min": 0.5137589661252424, "max": 0.5137589661252424}}}
#metrics {"StartTime": 1730680858.570287, "EndTime": 1730680858.5703, "Dime
nsions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "tr
aining", "epoch": 2, "model": 25}, "Metrics": {"validation_binary_classific
ation_cross_entropy_objective": {"sum": 0.5145155152450124, "count": 1, "mi

n": 0.5145155152450124, "max": 0.5145155152450124}}}
#metrics {"StartTime": 1730680858.570339, "EndTime": 1730680858.5703473, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 26}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5137585336223163, "count": 1,
"min": 0.5137585336223163, "max": 0.5137585336223163}}}
#metrics {"StartTime": 1730680858.5703778, "EndTime": 1730680858.570385, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 27}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5144958507288822, "count": 1,
"min": 0.5144958507288822, "max": 0.5144958507288822}}}
#metrics {"StartTime": 1730680858.5704145, "EndTime": 1730680858.5704215,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 28}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5139656584899815, "count":
1, "min": 0.5139656584899815, "max": 0.5139656584899815}}}
#metrics {"StartTime": 1730680858.570451, "EndTime": 1730680858.5704613, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 29}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5136981255766506, "count": 1,
"min": 0.5136981255766506, "max": 0.5136981255766506}}}
#metrics {"StartTime": 1730680858.570507, "EndTime": 1730680858.5705156, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 2, "model": 30}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5139747774354123, "count": 1,
"min": 0.5139747774354123, "max": 0.5139747774354123}}}
#metrics {"StartTime": 1730680858.5705462, "EndTime": 1730680858.5705533,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "model": 31}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5136981739020423, "count":
1, "min": 0.5136981739020423, "max": 0.5136981739020423}}}
[11/04/2024 00:40:58 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=2, validation binary_classification_cross_entropy_objective <loss>=0.49
21819552677068
[11/04/2024 00:40:58 INFO 140618102482752] #early_stopping_criteria_metric:
host=algo-1, epoch=2, criteria=binary_classification_cross_entropy_objectiv
e, value=0.4921819552677068
[11/04/2024 00:40:58 INFO 140618102482752] Saving model for epoch: 2
[11/04/2024 00:40:58 INFO 140618102482752] Saved checkpoint to "/tmp/tmps5q
5u56s/mx-mod-0000.params"
[11/04/2024 00:40:58 INFO 140618102482752] #progress_metric: host=algo-1, c
ompleted 20.0 % of epochs
#metrics {"StartTime": 1730680812.5333934, "EndTime": 1730680858.5765712,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 2, "Meta": "training_data_iter"}, "Metrics": {"Tot
al Records Seen": {"sum": 3937416.0, "count": 1, "min": 3937416, "max": 393
7416}, "Total Batches Seen": {"sum": 3939.0, "count": 1, "min": 3939, "ma
x": 3939}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count":
1, "min": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"su
m": 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}}}
[11/04/2024 00:40:58 INFO 140618102482752] #throughput_metric: host=algo-1,
train throughput=28418.276787523537 records/second

[2024-11-04 00:41:42.022] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 9, "duration": 43445, "num_examples": 1309, "num_bytes": 549558240}
#metrics {"StartTime": 1730680902.0228667, "EndTime": 1730680902.0229633, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 0}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4918286125346426, "count": 1, "min": 0.4918286125346426, "max": 0.4918286125346426}}}
#metrics {"StartTime": 1730680902.023065, "EndTime": 1730680902.0230865, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 1}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4927683104768806, "count": 1, "min": 0.4927683104768806, "max": 0.4927683104768806}}}
#metrics {"StartTime": 1730680902.023136, "EndTime": 1730680902.0231524, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 2}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4918526237791102, "count": 1, "min": 0.4918526237791102, "max": 0.4918526237791102}}}
#metrics {"StartTime": 1730680902.0232003, "EndTime": 1730680902.0232158, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 3}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49276774566837045, "count": 1, "min": 0.49276774566837045, "max": 0.49276774566837045}}}
#metrics {"StartTime": 1730680902.0232747, "EndTime": 1730680902.0232906, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 4}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4921221060796615, "count": 1, "min": 0.4921221060796615, "max": 0.4921221060796615}}}
#metrics {"StartTime": 1730680902.0233521, "EndTime": 1730680902.023368, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 5}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5088830552174046, "count": 1, "min": 0.5088830552174046, "max": 0.5088830552174046}}}
#metrics {"StartTime": 1730680902.0234203, "EndTime": 1730680902.023436, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 6}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49212210334987816, "count": 1, "min": 0.49212210334987816, "max": 0.49212210334987816}}}
#metrics {"StartTime": 1730680902.0234954, "EndTime": 1730680902.0235116, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 7}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5088830160671782, "count": 1, "min": 0.5088830160671782, "max": 0.5088830160671782}}}
#metrics {"StartTime": 1730680902.0235724, "EndTime": 1730680902.0235887, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 8}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4920300830759404, "count": 1, "min": 0.4920300830759404, "max": 0.4920300830759404}}}
#metrics {"StartTime": 1730680902.0236497, "EndTime": 1730680902.0236666, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 9}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4929334979975989, "count": 1, "min": 0.4929334979975989, "max": 0.4929334979975989}}}
#metrics {"StartTime": 1730680902.0237281, "EndTime": 1730680902.023743, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 10}, "Metrics": {"train_binary_classificat

ion_cross_entropy_objective": {"sum": 0.4920442713232944, "count": 1, "mi
n": 0.4920442713232944, "max": 0.4920442713232944}}}
#metrics {"StartTime": 1730680902.0237906, "EndTime": 1730680902.0238035,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 11}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.4929334996074711, "count": 1,
"min": 0.4929334996074711, "max": 0.4929334996074711}}}
#metrics {"StartTime": 1730680902.023854, "EndTime": 1730680902.023867, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 12}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.49232030771704627, "count": 1, "mi
n": 0.49232030771704627, "max": 0.49232030771704627}}}
#metrics {"StartTime": 1730680902.0239165, "EndTime": 1730680902.0239305,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 13}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5079500616697725, "count": 1,
"min": 0.5079500616697725, "max": 0.5079500616697725}}}
#metrics {"StartTime": 1730680902.0239801, "EndTime": 1730680902.0239952,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 14}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.4923203154630982, "count": 1,
"min": 0.4923203154630982, "max": 0.4923203154630982}}}
#metrics {"StartTime": 1730680902.0240448, "EndTime": 1730680902.0240602,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 15}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5079500550203002, "count": 1,
"min": 0.5079500550203002, "max": 0.5079500550203002}}}
#metrics {"StartTime": 1730680902.0241148, "EndTime": 1730680902.0241306,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 16}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5077430653703322, "count": 1,
"min": 0.5077430653703322, "max": 0.5077430653703322}}}
#metrics {"StartTime": 1730680902.0241823, "EndTime": 1730680902.024198, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 17}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5079842141994279, "count": 1, "mi
n": 0.5079842141994279, "max": 0.5079842141994279}}}
#metrics {"StartTime": 1730680902.0242586, "EndTime": 1730680902.0242753,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 18}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5077524060123922, "count": 1,
"min": 0.5077524060123922, "max": 0.5077524060123922}}}
#metrics {"StartTime": 1730680902.0243337, "EndTime": 1730680902.0243504,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 19}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5079842236253465, "count": 1,
"min": 0.5079842236253465, "max": 0.5079842236253465}}}
#metrics {"StartTime": 1730680902.0244095, "EndTime": 1730680902.0244257,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 20}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5078132000611099, "count": 1,
"min": 0.5078132000611099, "max": 0.5078132000611099}}}
#metrics {"StartTime": 1730680902.0244865, "EndTime": 1730680902.0245025,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 21}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.510694306446507, "count": 1, "m

in": 0.510694306446507, "max": 0.510694306446507}}}
#metrics {"StartTime": 1730680902.0245595, "EndTime": 1730680902.024576, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 22}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5078132006910598, "count": 1, "mi
n": 0.5078132006910598, "max": 0.5078132006910598}}}
#metrics {"StartTime": 1730680902.024635, "EndTime": 1730680902.0246518, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 23}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5106943101795441, "count": 1, "mi
n": 0.5106943101795441, "max": 0.5106943101795441}}}
#metrics {"StartTime": 1730680902.0247023, "EndTime": 1730680902.0247169,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 24}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137682394325186, "count": 1,
"min": 0.5137682394325186, "max": 0.5137682394325186}}}
#metrics {"StartTime": 1730680902.0247622, "EndTime": 1730680902.0247767,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 25}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5138687739474329, "count": 1,
"min": 0.5138687739474329, "max": 0.5138687739474329}}}
#metrics {"StartTime": 1730680902.0248244, "EndTime": 1730680902.0248334,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 26}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137699542031011, "count": 1,
"min": 0.5137699542031011, "max": 0.5137699542031011}}}
#metrics {"StartTime": 1730680902.0248692, "EndTime": 1730680902.024882, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 27}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5138690392963747, "count": 1, "mi
n": 0.5138690392963747, "max": 0.5138690392963747}}}
#metrics {"StartTime": 1730680902.0249302, "EndTime": 1730680902.0249445,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 28}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137890123606458, "count": 1,
"min": 0.5137890123606458, "max": 0.5137890123606458}}}
#metrics {"StartTime": 1730680902.0249906, "EndTime": 1730680902.0250087,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 29}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5148566347174688, "count": 1,
"min": 0.5148566347174688, "max": 0.5148566347174688}}}
#metrics {"StartTime": 1730680902.0250547, "EndTime": 1730680902.0250704,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 30}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137889759868657, "count": 1,
"min": 0.5137889759868657, "max": 0.5137889759868657}}}
#metrics {"StartTime": 1730680902.025135, "EndTime": 1730680902.02515, "Dim
ensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "t
raining", "epoch": 3, "model": 31}, "Metrics": {"train_binary_classificatio
n_cross_entropy_objective": {"sum": 0.5148566444233652, "count": 1, "min":
0.5148566444233652, "max": 0.5148566444233652}}}
[11/04/2024 00:41:42 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=3, train binary_classification_cross_entropy_objective <loss>=0.4918286
125346426
[2024-11-04 00:41:45.619] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 11, "duration": 3575, "num_exampl

es": 164, "num_bytes": 68694780}
#metrics {"StartTime": 1730680905.6261718, "EndTime": 1730680905.626226, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 0}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49217695666831474, "count": 1,
"min": 0.49217695666831474, "max": 0.49217695666831474}}}
#metrics {"StartTime": 1730680905.6263037, "EndTime": 1730680905.626319, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 1}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4935410352596626, "count": 1,
"min": 0.4935410352596626, "max": 0.4935410352596626}}}
#metrics {"StartTime": 1730680905.6263797, "EndTime": 1730680905.6263967,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 2}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49218468518587427, "count":
1, "min": 0.49218468518587427, "max": 0.49218468518587427}}}
#metrics {"StartTime": 1730680905.6264498, "EndTime": 1730680905.6264615,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 3}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4935408037082686, "count":
1, "min": 0.4935408037082686, "max": 0.4935408037082686}}}
#metrics {"StartTime": 1730680905.6265078, "EndTime": 1730680905.6265228,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 4}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49229355427014776, "count":
1, "min": 0.49229355427014776, "max": 0.49229355427014776}}}
#metrics {"StartTime": 1730680905.6265805, "EndTime": 1730680905.6265953,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 5}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5098846760017361, "count":
1, "min": 0.5098846760017361, "max": 0.5098846760017361}}}
#metrics {"StartTime": 1730680905.6266453, "EndTime": 1730680905.6266606,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 6}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.492293544754337, "count":
1, "min": 0.492293544754337, "max": 0.492293544754337}}}
#metrics {"StartTime": 1730680905.6267068, "EndTime": 1730680905.626721, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 7}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.5098846159215196, "count": 1,
"min": 0.5098846159215196, "max": 0.5098846159215196}}}
#metrics {"StartTime": 1730680905.6267705, "EndTime": 1730680905.6267827,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 8}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49236674615446013, "count":
1, "min": 0.49236674615446013, "max": 0.49236674615446013}}}
#metrics {"StartTime": 1730680905.62681, "EndTime": 1730680905.6268227, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 9}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49373277250165176, "count": 1,
"min": 0.49373277250165176, "max": 0.49373277250165176}}}
#metrics {"StartTime": 1730680905.6268697, "EndTime": 1730680905.6268835,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 10}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.49237674521908953, "coun
t": 1, "min": 0.49237674521908953, "max": 0.49237674521908953}}}

#metrics {"StartTime": 1730680905.6269312, "EndTime": 1730680905.626947, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 11}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.4937327807113708, "count": 1,
"min": 0.4937327807113708, "max": 0.4937327807113708}}}
#metrics {"StartTime": 1730680905.6270027, "EndTime": 1730680905.6270173,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 12}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.4925224352664549, "count":
1, "min": 0.4925224352664549, "max": 0.4925224352664549}}}
#metrics {"StartTime": 1730680905.6270702, "EndTime": 1730680905.627085, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 13}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5087560163284087, "count": 1,
"min": 0.5087560163284087, "max": 0.5087560163284087}}}
#metrics {"StartTime": 1730680905.6271298, "EndTime": 1730680905.627144, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 14}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.49252243582620847, "count": 1,
"min": 0.49252243582620847, "max": 0.49252243582620847}}}
#metrics {"StartTime": 1730680905.6271927, "EndTime": 1730680905.627208, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 15}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.50875591986421, "count": 1, "m
in": 0.50875591986421, "max": 0.50875591986421}}}
#metrics {"StartTime": 1730680905.6272604, "EndTime": 1730680905.6272752,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 16}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5078388087910221, "count":
1, "min": 0.5078388087910221, "max": 0.5078388087910221}}}
#metrics {"StartTime": 1730680905.6273232, "EndTime": 1730680905.6273375,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 3, "model": 17}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.508474389006736, "count":
1, "min": 0.508474389006736, "max": 0.508474389006736}}}
#metrics {"StartTime": 1730680905.6273818, "EndTime": 1730680905.627396, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 18}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5078861758845671, "count": 1,
"min": 0.5078861758845671, "max": 0.5078861758845671}}}
#metrics {"StartTime": 1730680905.627445, "EndTime": 1730680905.6274595, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 19}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.508474444235755, "count": 1,
"min": 0.508474444235755, "max": 0.508474444235755}}}
#metrics {"StartTime": 1730680905.627504, "EndTime": 1730680905.6275187, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 20}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.507996037225338, "count": 1,
"min": 0.507996037225338, "max": 0.507996037225338}}}
#metrics {"StartTime": 1730680905.6275644, "EndTime": 1730680905.627579, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 3, "model": 21}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5096874337699574, "count": 1,
"min": 0.5096874337699574, "max": 0.5096874337699574}}}
#metrics {"StartTime": 1730680905.627625, "EndTime": 1730680905.627639, "Di

mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 22}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5079960424497046, "count": 1, "min": 0.5079960424497046, "max": 0.5079960424497046}}}
#metrics {"StartTime": 1730680905.6276848, "EndTime": 1730680905.6276984, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 23}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5096873766750931, "count": 1, "min": 0.5096873766750931, "max": 0.5096873766750931}}}
#metrics {"StartTime": 1730680905.627743, "EndTime": 1730680905.6277578, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 24}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5137649411214474, "count": 1, "min": 0.5137649411214474, "max": 0.5137649411214474}}}
#metrics {"StartTime": 1730680905.6278152, "EndTime": 1730680905.6278315, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 25}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5144281964866511, "count": 1, "min": 0.5144281964866511, "max": 0.5144281964866511}}}
#metrics {"StartTime": 1730680905.6278894, "EndTime": 1730680905.627906, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 26}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5137584782067127, "count": 1, "min": 0.5137584782067127, "max": 0.5137584782067127}}}
#metrics {"StartTime": 1730680905.6279547, "EndTime": 1730680905.6279697, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 27}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5144084073320588, "count": 1, "min": 0.5144084073320588, "max": 0.5144084073320588}}}
#metrics {"StartTime": 1730680905.628024, "EndTime": 1730680905.6280382, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 28}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5137727771116926, "count": 1, "min": 0.5137727771116926, "max": 0.5137727771116926}}}
#metrics {"StartTime": 1730680905.6280937, "EndTime": 1730680905.6281092, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 29}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5135732697437951, "count": 1, "min": 0.5135732697437951, "max": 0.5135732697437951}}}
#metrics {"StartTime": 1730680905.6281655, "EndTime": 1730680905.6281812, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 30}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5137685571295183, "count": 1, "min": 0.5137685571295183, "max": 0.5137685571295183}}}
#metrics {"StartTime": 1730680905.628238, "EndTime": 1730680905.6282537, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 3, "model": 31}, "Metrics": {"validation_binary_classification_cross_entropy_objective": {"sum": 0.5135732712364713, "count": 1, "min": 0.5135732712364713, "max": 0.5135732712364713}}}
[11/04/2024 00:41:45 INFO 140618102482752] #quality_metric: host=algo-1, epoch=3, validation binary_classification_cross_entropy_objective <loss>=0.49217695666831474
[11/04/2024 00:41:45 INFO 140618102482752] #early_stopping_criteria_metric: host=algo-1, epoch=3, criteria=binary_classification_cross_entropy_objective, value=0.49217695666831474
[11/04/2024 00:41:45 INFO 140618102482752] Saving model for epoch: 3

```
[11/04/2024 00:41:45 INFO 140618102482752] Saved checkpoint to "/tmp/tmp5e5
hmrss/mx-mod-0000.params"
[11/04/2024 00:41:45 INFO 140618102482752] #progress_metric: host=algo-1, c
ompleted 26.666666666666668 % of epochs
#metrics {"StartTime": 1730680858.577168, "EndTime": 1730680905.63485, "Dim
ensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "t
raining", "epoch": 3, "Meta": "training_data_iter"}, "Metrics": {"Total Rec
ords Seen": {"sum": 5245888.0, "count": 1, "min": 5245888, "max": 5245888},
"Total Batches Seen": {"sum": 5248.0, "count": 1, "min": 5248, "max": 524
8}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count": 1, "mi
n": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"sum": 13
09.0, "count": 1, "min": 1309, "max": 1309}, "Reset Count": {"sum": 6.0, "c
ount": 1, "min": 6, "max": 6}, "Number of Records Since Last Reset": {"su
m": 1308472.0, "count": 1, "min": 1308472, "max": 1308472}, "Number of Batc
hes Since Last Reset": {"sum": 1309.0, "count": 1, "min": 1309, "max": 130
9}}}
[11/04/2024 00:41:45 INFO 140618102482752] #throughput_metric: host=algo-1,
train throughput=27805.615299257697 records/second
[2024-11-04 00:42:27.657] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/train", "epoch": 11, "duration": 42021, "num_examples":
1309, "num_bytes": 549558240}
#metrics {"StartTime": 1730680947.6577084, "EndTime": 1730680947.6578, "Dim
ensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "t
raining", "epoch": 4, "model": 0}, "Metrics": {"train_binary_classification
_cross_entropy_objective": {"sum": 0.49180100390233034, "count": 1, "min":
0.49180100390233034, "max": 0.49180100390233034}}}
#metrics {"StartTime": 1730680947.6579032, "EndTime": 1730680947.6579256,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 1}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.4926461112579439, "count": 1, "m
in": 0.4926461112579439, "max": 0.4926461112579439}}}
#metrics {"StartTime": 1730680947.6579812, "EndTime": 1730680947.6579986,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 2}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.49185105781671834, "count": 1,
"min": 0.49185105781671834, "max": 0.49185105781671834}}}
#metrics {"StartTime": 1730680947.6580448, "EndTime": 1730680947.6580577,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 3}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.4926458321900907, "count": 1, "m
in": 0.4926458321900907, "max": 0.4926458321900907}}}
#metrics {"StartTime": 1730680947.6581082, "EndTime": 1730680947.6581216,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 4}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.491900448743721, "count": 1, "mi
n": 0.491900448743721, "max": 0.491900448743721}}}
#metrics {"StartTime": 1730680947.6581624, "EndTime": 1730680947.6581762,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 5}, "Metrics": {"train_binary_classifi
cation_cross_entropy_objective": {"sum": 0.5068938369750977, "count": 1, "m
in": 0.5068938369750977, "max": 0.5068938369750977}}}
#metrics {"StartTime": 1730680947.6582236, "EndTime": 1730680947.658238, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics": {"train_binary_classificati
on_cross_entropy_objective": {"sum": 0.4919004465038988, "count": 1, "min":
0.4919004465038988, "max": 0.4919004465038988}}}
```

#metrics {"StartTime": 1730680947.6582892, "EndTime": 1730680947.6583042, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 7}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5068938284124439, "count": 1, "min": 0.5068938284124439, "max": 0.5068938284124439}}}
#metrics {"StartTime": 1730680947.6583383, "EndTime": 1730680947.6583462, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 8}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4920026745810786, "count": 1, "min": 0.4920026745810786, "max": 0.4920026745810786}}}
#metrics {"StartTime": 1730680947.6583698, "EndTime": 1730680947.6583765, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 9}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49281887749721515, "count": 1, "min": 0.49281887749721515, "max": 0.49281887749721515}}}
#metrics {"StartTime": 1730680947.6583989, "EndTime": 1730680947.6584058, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 10}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.49204346398403154, "count": 1, "min": 0.49204346398403154, "max": 0.49204346398403154}}}
#metrics {"StartTime": 1730680947.6584282, "EndTime": 1730680947.6584346, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 11}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4928188778238559, "count": 1, "min": 0.4928188778238559, "max": 0.4928188778238559}}}
#metrics {"StartTime": 1730680947.6584728, "EndTime": 1730680947.6584873, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 12}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4921139252222277, "count": 1, "min": 0.4921139252222277, "max": 0.4921139252222277}}}
#metrics {"StartTime": 1730680947.6585364, "EndTime": 1730680947.6585531, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 13}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5061214391025928, "count": 1, "min": 0.5061214391025928, "max": 0.5061214391025928}}}
#metrics {"StartTime": 1730680947.658599, "EndTime": 1730680947.6586134, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 14}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.4921139268087685, "count": 1, "min": 0.4921139268087685, "max": 0.4921139268087685}}}
#metrics {"StartTime": 1730680947.658661, "EndTime": 1730680947.6586766, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 15}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5061214373527317, "count": 1, "min": 0.5061214373527317, "max": 0.5061214373527317}}}
#metrics {"StartTime": 1730680947.6587274, "EndTime": 1730680947.6587422, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 16}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5077465337817457, "count": 1, "min": 0.5077465337817457, "max": 0.5077465337817457}}}
#metrics {"StartTime": 1730680947.6587968, "EndTime": 1730680947.6588118, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 17}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5079556549258918, "count": 1, "min": 0.5079556549258918, "max": 0.5079556549258918}}}
#metrics {"StartTime": 1730680947.658857, "EndTime": 1730680947.6588724, "D

imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 18}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5077523518133601, "count": 1, "min": 0.5077523518133601, "max": 0.5077523518133601}}}
#metrics {"StartTime": 1730680947.6589186, "EndTime": 1730680947.658933, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 19}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5079556570723881, "count": 1, "min": 0.5079556570723881, "max": 0.5079556570723881}}}
#metrics {"StartTime": 1730680947.6589844, "EndTime": 1730680947.659, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 20}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5077618997323039, "count": 1, "min": 0.5077618997323039, "max": 0.5077618997323039}}}
#metrics {"StartTime": 1730680947.6591132, "EndTime": 1730680947.6591322, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 21}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.510381959291044, "count": 1, "min": 0.510381959291044, "max": 0.510381959291044}}}
#metrics {"StartTime": 1730680947.6591723, "EndTime": 1730680947.659181, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 22}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5077618992656743, "count": 1, "min": 0.5077618992656743, "max": 0.5077618992656743}}}
#metrics {"StartTime": 1730680947.6592078, "EndTime": 1730680947.6592155, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 23}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5103819590343976, "count": 1, "min": 0.5103819590343976, "max": 0.5103819590343976}}}
#metrics {"StartTime": 1730680947.6592627, "EndTime": 1730680947.6592767, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 24}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.51376620791374, "count": 1, "min": 0.51376620791374, "max": 0.51376620791374}}}
#metrics {"StartTime": 1730680947.6593277, "EndTime": 1730680947.6593392, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 25}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5138555431074323, "count": 1, "min": 0.5138555431074323, "max": 0.5138555431074323}}}
#metrics {"StartTime": 1730680947.6593661, "EndTime": 1730680947.6593773, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 26}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.513770029120489, "count": 1, "min": 0.513770029120489, "max": 0.513770029120489}}}
#metrics {"StartTime": 1730680947.659421, "EndTime": 1730680947.6594338, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 27}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5138557057978546, "count": 1, "min": 0.5138557057978546, "max": 0.5138557057978546}}}
#metrics {"StartTime": 1730680947.6594846, "EndTime": 1730680947.6595001, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training", "epoch": 4, "model": 28}, "Metrics": {"train_binary_classification_cross_entropy_objective": {"sum": 0.5137755978085579, "count": 1, "min": 0.5137755978085579, "max": 0.5137755978085579}}}
#metrics {"StartTime": 1730680947.659548, "EndTime": 1730680947.6595633, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":

"training", "epoch": 4, "model": 29}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5147192236757424, "count": 1, "mi
n": 0.5147192236757424, "max": 0.5147192236757424}}}
#metrics {"StartTime": 1730680947.6596208, "EndTime": 1730680947.6596336,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 30}, "Metrics": {"train_binary_classif
ication_cross_entropy_objective": {"sum": 0.5137756835284218, "count": 1,
"min": 0.5137756835284218, "max": 0.5137756835284218}}}
#metrics {"StartTime": 1730680947.659678, "EndTime": 1730680947.659693, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics": {"train_binary_classificat
ion_cross_entropy_objective": {"sum": 0.5147192272687906, "count": 1, "mi
n": 0.5147192272687906, "max": 0.5147192272687906}}}
[11/04/2024 00:42:27 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=4, train binary_classification_cross_entropy_objective <loss>=0.4918010
0390233034
[2024-11-04 00:42:31.505] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 14, "duration": 3826, "num_exampl
es": 164, "num_bytes": 68694780}
#metrics {"StartTime": 1730680951.5131524, "EndTime": 1730680951.5132067,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 0}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.49217736696768255, "count":
1, "min": 0.49217736696768255, "max": 0.49217736696768255}}}
#metrics {"StartTime": 1730680951.513289, "EndTime": 1730680951.5133255, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 1}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49334084424607005, "count": 1,
"min": 0.49334084424607005, "max": 0.49334084424607005}}}
#metrics {"StartTime": 1730680951.5133922, "EndTime": 1730680951.5134099,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 2}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4921836828538125, "count":
1, "min": 0.4921836828538125, "max": 0.4921836828538125}}}
#metrics {"StartTime": 1730680951.5134652, "EndTime": 1730680951.5134816,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 3}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4933407513269772, "count":
1, "min": 0.4933407513269772, "max": 0.4933407513269772}}}
#metrics {"StartTime": 1730680951.5135314, "EndTime": 1730680951.513547, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 4}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4921685148381208, "count": 1,
"min": 0.4921685148381208, "max": 0.4921685148381208}}}
#metrics {"StartTime": 1730680951.513606, "EndTime": 1730680951.5136218, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 5}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.5072567943799022, "count": 1,
"min": 0.5072567943799022, "max": 0.5072567943799022}}}
#metrics {"StartTime": 1730680951.5136611, "EndTime": 1730680951.513675, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 6}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.4921685161442125, "count": 1,
"min": 0.4921685161442125, "max": 0.4921685161442125}}}
#metrics {"StartTime": 1730680951.5137305, "EndTime": 1730680951.5137455,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio

n": "training", "epoch": 4, "model": 7}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.5072567794531403, "count":
1, "min": 0.5072567794531403, "max": 0.5072567794531403}}}
#metrics {"StartTime": 1730680951.5138018, "EndTime": 1730680951.5138178,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 8}, "Metrics": {"validation_binary_cla
ssification_cross_entropy_objective": {"sum": 0.4923661132597555, "count":
1, "min": 0.4923661132597555, "max": 0.4923661132597555}}}
#metrics {"StartTime": 1730680951.513871, "EndTime": 1730680951.5138857, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 9}, "Metrics": {"validation_binary_classif
ication_cross_entropy_objective": {"sum": 0.49353486528263024, "count": 1,
"min": 0.49353486528263024, "max": 0.49353486528263024}}}
#metrics {"StartTime": 1730680951.5139244, "EndTime": 1730680951.5139384,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 10}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.4923768909416026, "count":
1, "min": 0.4923768909416026, "max": 0.4923768909416026}}}
#metrics {"StartTime": 1730680951.5139894, "EndTime": 1730680951.5140045,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 11}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.49353487088016595, "coun
t": 1, "min": 0.49353487088016595, "max": 0.49353487088016595}}}
#metrics {"StartTime": 1730680951.514039, "EndTime": 1730680951.5140524, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 12}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.49238857766324956, "count": 1,
"min": 0.49238857766324956, "max": 0.49238857766324956}}}
#metrics {"StartTime": 1730680951.5140946, "EndTime": 1730680951.514109, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 13}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5064340047688377, "count": 1,
"min": 0.5064340047688377, "max": 0.5064340047688377}}}
#metrics {"StartTime": 1730680951.514161, "EndTime": 1730680951.5141768, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 14}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.4923885800888484, "count": 1,
"min": 0.4923885800888484, "max": 0.4923885800888484}}}
#metrics {"StartTime": 1730680951.5142257, "EndTime": 1730680951.5142403,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 15}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5064339827518639, "count":
1, "min": 0.5064339827518639, "max": 0.5064339827518639}}}
#metrics {"StartTime": 1730680951.5142932, "EndTime": 1730680951.514309, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 16}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5078115390897026, "count": 1,
"min": 0.5078115390897026, "max": 0.5078115390897026}}}
#metrics {"StartTime": 1730680951.514347, "EndTime": 1730680951.514356, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 17}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.508399391917156, "count": 1,
"min": 0.508399391917156, "max": 0.508399391917156}}}
#metrics {"StartTime": 1730680951.5144057, "EndTime": 1730680951.51442, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 18}, "Metrics": {"validation_binary_classi

fication_cross_entropy_objective": {"sum": 0.5078862313001706, "count": 1,
"min": 0.5078862313001706, "max": 0.5078862313001706}}}
#metrics {"StartTime": 1730680951.5144656, "EndTime": 1730680951.5144804,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 19}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5083994068439179, "count":
1, "min": 0.5083994068439179, "max": 0.5083994068439179}}}
#metrics {"StartTime": 1730680951.5145335, "EndTime": 1730680951.5145485,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 20}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5078981131892295, "count":
1, "min": 0.5078981131892295, "max": 0.5078981131892295}}}
#metrics {"StartTime": 1730680951.514589, "EndTime": 1730680951.5146036, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 21}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5092230103366752, "count": 1,
"min": 0.5092230103366752, "max": 0.5092230103366752}}}
#metrics {"StartTime": 1730680951.5146525, "EndTime": 1730680951.5146673,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 22}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5078981150550748, "count":
1, "min": 0.5078981150550748, "max": 0.5078981150550748}}}
#metrics {"StartTime": 1730680951.5147054, "EndTime": 1730680951.514714, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 23}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5092230021269561, "count": 1,
"min": 0.5092230021269561, "max": 0.5092230021269561}}}
#metrics {"StartTime": 1730680951.5147598, "EndTime": 1730680951.5147684,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 24}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5137642528111391, "count":
1, "min": 0.5137642528111391, "max": 0.5137642528111391}}}
#metrics {"StartTime": 1730680951.5148172, "EndTime": 1730680951.514827, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 25}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5142932553871161, "count": 1,
"min": 0.5142932553871161, "max": 0.5142932553871161}}}
#metrics {"StartTime": 1730680951.514871, "EndTime": 1730680951.514885, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 26}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5137586847557805, "count": 1,
"min": 0.5137586847557805, "max": 0.5137586847557805}}}
#metrics {"StartTime": 1730680951.5149415, "EndTime": 1730680951.514957, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 27}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5142862862685684, "count": 1,
"min": 0.5142862862685684, "max": 0.5142862862685684}}}
#metrics {"StartTime": 1730680951.515007, "EndTime": 1730680951.5150194, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 28}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5137595253190601, "count": 1,
"min": 0.5137595253190601, "max": 0.5137595253190601}}}
#metrics {"StartTime": 1730680951.5150576, "EndTime": 1730680951.5150712,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "model": 29}, "Metrics": {"validation_binary_cl
assification_cross_entropy_objective": {"sum": 0.5135371352251663, "count":

1, "min": 0.5135371352251663, "max": 0.5135371352251663}}}
#metrics {"StartTime": 1730680951.515126, "EndTime": 1730680951.515141, "Di
mensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 30}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5137580843267829, "count": 1,
"min": 0.5137580843267829, "max": 0.5137580843267829}}}
#metrics {"StartTime": 1730680951.515197, "EndTime": 1730680951.5152125, "D
imensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation":
"training", "epoch": 4, "model": 31}, "Metrics": {"validation_binary_classi
fication_cross_entropy_objective": {"sum": 0.5135371257093555, "count": 1,
"min": 0.5135371257093555, "max": 0.5135371257093555}}}
[11/04/2024 00:42:31 INFO 140618102482752] #quality_metric: host=algo-1, ep
och=4, validation binary_classification_cross_entropy_objective <loss>=0.49
217736696768255
[11/04/2024 00:42:31 INFO 140618102482752] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objectiv
e, value=0.4921685148381208
[11/04/2024 00:42:31 INFO 140618102482752] Saving model for epoch: 4
[11/04/2024 00:42:31 INFO 140618102482752] Saved checkpoint to "/tmp/tmp3ab
tf7m0/mx-mod-0000.params"
[11/04/2024 00:42:31 INFO 140618102482752] Early stop condition met. Stoppi
ng training.
[11/04/2024 00:42:31 INFO 140618102482752] #progress_metric: host=algo-1, c
ompleted 100 % epochs
#metrics {"StartTime": 1730680905.6357706, "EndTime": 1730680951.5216358,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "epoch": 4, "Meta": "training_data_iter"}, "Metrics": {"Tot
al Records Seen": {"sum": 6554360.0, "count": 1, "min": 6554360, "max": 655
4360}, "Total Batches Seen": {"sum": 6557.0, "count": 1, "min": 6557, "ma
x": 6557}, "Max Records Seen Between Resets": {"sum": 1308472.0, "count":
1, "min": 1308472, "max": 1308472}, "Max Batches Seen Between Resets": {"su
m": 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}}}
[11/04/2024 00:42:31 INFO 140618102482752] #throughput_metric: host=algo-1,
train throughput=28515.704014695675 records/second
[11/04/2024 00:42:31 WARNING 140618102482752] wait_for_all_workers will not
sync workers since the kv store is not running distributed
[11/04/2024 00:42:31 WARNING 140618102482752] wait_for_all_workers will not
sync workers since the kv store is not running distributed

2024-11-04 00:42:40 Uploading - Uploading generated training model[2024-11-
04 00:42:35.156] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/i
nput/data/validation", "epoch": 17, "duration": 3617, "num_examples": 164,
"num_bytes": 68694780}
[11/04/2024 00:42:35 INFO 140618102482752] #early_stopping_criteria_metric:
host=algo-1, epoch=4, criteria=binary_classification_cross_entropy_objectiv
e, value=0.4921685148381208
[2024-11-04 00:42:35.766] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/validation", "epoch": 19, "duration": 598, "num_example
s": 164, "num_bytes": 68694780}
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('b
inary_classification_cross_entropy_objective', 0.49259978556004874)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('b

```
inary_classification_accuracy', 0.7900879804841067)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('b
inary_f_1.000', 0.004234461556309637)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('p
recision', 0.5)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('r
ecall', 0.002126234235283838)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('r
oc_auc_score', 0.6433222280900706)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('b
inary_balanced_accuracy', 0.5)
[11/04/2024 00:42:36 INFO 140618102482752] #validation_score (algo-1) : ('b
inary_log_loss', 0.7535580726090925)
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation binary_classification_cross_entropy_objective <loss>=0.49259978556
004874
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation binary_classification_accuracy <score>=0.7900879804841067
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation binary_f_1.000 <score>=0.004234461556309637
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation precision <score>=0.5
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation recall <score>=0.002126234235283838
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation roc_auc_score <score>=0.6433222280900706
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation binary_balanced_accuracy <score>=0.5
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, va
lidation binary_log_loss <score>=0.7535580726090925
[11/04/2024 00:42:36 INFO 140618102482752] Best model found for hyperparame
ters: {"optimizer": "adam", "learning_rate": 0.005, "l1": 0.0, "wd": 0.000
1, "lr_scheduler_step": 10, "lr_scheduler_factor": 0.99, "lr_scheduler_mini
mum_lr": 1e-05}
[11/04/2024 00:42:36 INFO 140618102482752] Saved checkpoint to "/tmp/tmp84o
ywd19/mx-mod-0000.params"
[2024-11-04 00:42:36.027] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 0, "duration": 237623, "num_examples":
1, "num_bytes": 420000}
[2024-11-04 00:42:36.637] [tensorio] [info] epoch_stats={"data_pipeline":
"/opt/ml/input/data/test", "epoch": 1, "duration": 609, "num_examples": 16
4, "num_bytes": 68694780}
#metrics {"StartTime": 1730680956.0237098, "EndTime": 1730680956.8811777,
"Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operatio
n": "training", "Meta": "test_data_iter"}, "Metrics": {"Total Records See
n": {"sum": 163559.0, "count": 1, "min": 163559, "max": 163559}, "Total Bat
ches Seen": {"sum": 164.0, "count": 1, "min": 164, "max": 164}, "Max Record
s Seen Between Resets": {"sum": 163559.0, "count": 1, "min": 163559, "max":
163559}, "Max Batches Seen Between Resets": {"sum": 164.0, "count": 1, "mi
n": 164, "max": 164}, "Reset Count": {"sum": 1.0, "count": 1, "min": 1, "ma
x": 1}, "Number of Records Since Last Reset": {"sum": 163559.0, "count": 1,
"min": 163559, "max": 163559}, "Number of Batches Since Last Reset": {"su
m": 164.0, "count": 1, "min": 164, "max": 164}}}
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('binary_
classification_cross_entropy_objective', 0.49370398605473836)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('binary_
```

classification_accuracy', 0.7902408305259876)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('binary_f_1.000', 0.00579575750550597)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('precision', 0.5714285714285714)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('recall', 0.0029126496373751204)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('roc_auc_score', 0.6396880607098765)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('binary_balanced_accuracy', 0.5)
[11/04/2024 00:42:36 INFO 140618102482752] #test_score (algo-1) : ('binary_log_loss', 0.7537304357983453)
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test binary_classification_cross_entropy_objective <loss>=0.49370398605473836
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test binary_classification_accuracy <score>=0.7902408305259876
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test binary_f_1.000 <score>=0.00579575750550597
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test precision <score>=0.5714285714285714
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test recall <score>=0.0029126496373751204
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test roc_auc_score <score>=0.6396880607098765
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test binary_balanced_accuracy <score>=0.5
[11/04/2024 00:42:36 INFO 140618102482752] #quality_metric: host=algo-1, test binary_log_loss <score>=0.7537304357983453
#metrics {"StartTime": 1730680718.395206, "EndTime": 1730680956.8893576, "Dimensions": {"Algorithm": "Linear Learner", "Host": "algo-1", "Operation": "training"}, "Metrics": {"initialize.time": {"sum": 299.08061027526855, "count": 1, "min": 299.08061027526855, "max": 299.08061027526855}, "epochs": {"sum": 15.0, "count": 1, "min": 15, "max": 15}, "check_early_stopping.time": {"sum": 2.0112991333007812, "count": 6, "min": 0.20456314086914062, "max": 0.7789134979248047}, "update.time": {"sum": 232752.0728111267, "count": 5, "min": 45882.93361663818, "max": 47799.317836761475}, "finalize.time": {"sum": 4498.605966567993, "count": 1, "min": 4498.605966567993, "max": 4498.605966567993}, "setuptime": {"sum": 2.417325973510742, "count": 1, "min": 2.417325973510742, "max": 2.417325973510742}, "totaltime": {"sum": 238615.04125595093, "count": 1, "min": 238615.04125595093, "max": 238615.041255950933}}}

2024-11-04 00:42:53 Completed - Training job completed
Training seconds: 391
Billable seconds: 391

Model evaluation

In this section, you will evaluate your trained model.

First, examine the metrics for the training job:

In [61]:
```
sagemaker.analytics.TrainingJobAnalytics(classifier_estimator._current_job_r
                                         metric_names = ['test:objective_los
                                                         'test:binary_f_beta
                                                         'test:precision',
                                                         'test:recall']
                                        ).dataframe()
```

Out[61]:

	timestamp	metric_name	value
0	0.0	test:objective_loss	0.493704
1	0.0	test:binary_f_beta	0.005796
2	0.0	test:precision	0.571429
3	0.0	test:recall	0.002913

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.

In [62]:
```
import io
# Having to find the LabBucketName was very interesting
bucket='sagemaker-us-east-1-905418072867'
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)
```

```
INFO:botocore.credentials:Found credentials from IAM Role: BaseNotebookInst
anceEc2InstanceRole
```

In [64]:
```
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,'
    target_predicted_df = pd.read_json(io.BytesIO(obj['Body'].read()),orient
    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.

In [65]:
```
test_labels, target_predicted = batch_linear_predict(test, classifier_estima
```

```
INFO:sagemaker.image_uris:Same images used for training and inference. Defa
ulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating model with name: linear-learner-2024-11-04-00-44-22
-307
INFO:sagemaker:Creating transform job with name: linear-learner-2024-11-04-
00-44-22-994
```

```
.....................................Docker entrypoint called with argum
ent(s): serve
Running default environment configuration script
[11/04/2024 00:51:06 INFO 139940816336704] Memory profiler is not enabled b
y 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: Syntax
Warning: "is" with a literal. Did you mean "=="?
  if cons['type'] is 'ineq':
/opt/amazon/lib/python3.8/site-packages/scipy/optimize/_shgo.py:743: Syntax
Warning: "is not" with a literal. Did you mean "!="?
  if len(self.X_min) is not 0:
[11/04/2024 00:51:10 WARNING 139940816336704] Loggers have already been set
up.
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm.serve.server_config:config_api
[11/04/2024 00:51:10 INFO 139940816336704] loading entry points
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/jsonlines
[11/04/2024 00:51:10 WARNING 139940816336704] Loggers have already been set
up.
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm.serve.server_config:config_api
[11/04/2024 00:51:10 INFO 139940816336704] loading entry points
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm:model
[11/04/2024 00:51:10 INFO 139940816336704] Number of server workers: 4
[11/04/2024 00:51:10 INFO 139940816336704] loading model...
[11/04/2024 00:51:10 INFO 139940816336704] ...model loaded.
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/x-recordio-protobuf
```

```
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm:model
[11/04/2024 00:51:10 INFO 139940816336704] Number of server workers: 4
[11/04/2024 00:51:10 INFO 139940816336704] loading model...
[11/04/2024 00:51:10 INFO 139940816336704] ...model loaded.
[2024-11-04 00:51:10 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-11-04 00:51:10 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-11-04 00:51:10 +0000] [1] [INFO] Using worker: sync
[2024-11-04 00:51:10 +0000] [43] [INFO] Booting worker with pid: 43
[2024-11-04 00:51:10 +0000] [52] [INFO] Booting worker with pid: 52
[2024-11-04 00:51:11 +0000] [61] [INFO] Booting worker with pid: 61
[2024-11-04 00:51:11 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681471.521616, "D
imensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operati
on": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "co
unt": 1, "min": 1, "max": 1}}}
[2024-11-04 00:51:10 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-11-04 00:51:10 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-11-04 00:51:10 +0000] [1] [INFO] Using worker: sync
[2024-11-04 00:51:10 +0000] [43] [INFO] Booting worker with pid: 43
[2024-11-04 00:51:10 +0000] [52] [INFO] Booting worker with pid: 52
[2024-11-04 00:51:11 +0000] [61] [INFO] Booting worker with pid: 61
[2024-11-04 00:51:11 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681471.521616, "D
imensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operati
on": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "co
unt": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.1536648,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 133.10599327087
402, "count": 1, "min": 133.10599327087402, "max": 133.10599327087402}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.1536648,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 133.10599327087
402, "count": 1, "min": 133.10599327087402, "max": 133.10599327087402}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681471.5218246, "EndTime": 1730681474.1617696,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.40594482421
875, "count": 1, "min": 136.40594482421875, "max": 136.40594482421875}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2701616,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.33935356140
137, "count": 1, "min": 137.33935356140137, "max": 137.33935356140137}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2703965,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 138.33999633789
062, "count": 1, "min": 138.33999633789062, "max": 138.33999633789062}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681471.5218246, "EndTime": 1730681474.1617696,
```

"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.40594482421875, "count": 1, "min": 136.40594482421875, "max": 136.40594482421875}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2701616, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.33935356140137, "count": 1, "min": 137.33935356140137, "max": 137.33935356140137}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2703965, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 138.33999633789062, "count": 1, "min": 138.33999633789062, "max": 138.33999633789062}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681474.1538303, "EndTime": 1730681475.0288732, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 62.47138977050781, "count": 1, "min": 62.47138977050781, "max": 62.47138977050781}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681474.1538303, "EndTime": 1730681475.0288732, "Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 62.47138977050781, "count": 1, "min": 62.47138977050781, "max": 62.47138977050781}, "invocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
2024-11-04T00:51:11.541:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD

Docker entrypoint called with argument(s): serve
Running default environment configuration script
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[11/04/2024 00:51:06 INFO 139940816336704] Memory profiler is not enabled by the environment variable ENABLE_PROFILER.
[11/04/2024 00:51:06 INFO 139940816336704] 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:
[11/04/2024 00:51:10 WARNING 139940816336704] Loggers have already been setup.
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit

```
hm.serve.server_config:config_api
[11/04/2024 00:51:10 INFO 139940816336704] loading entry points
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/jsonlines
[11/04/2024 00:51:10 WARNING 139940816336704] Loggers have already been set
up.
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm.serve.server_config:config_api
[11/04/2024 00:51:10 INFO 139940816336704] loading entry points
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded request iterator text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/json
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/jsonlines
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm:model
[11/04/2024 00:51:10 INFO 139940816336704] Number of server workers: 4
[11/04/2024 00:51:10 INFO 139940816336704] loading model...
[11/04/2024 00:51:10 INFO 139940816336704] ...model loaded.
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder applicat
ion/x-recordio-protobuf
[11/04/2024 00:51:10 INFO 139940816336704] loaded response encoder text/csv
[11/04/2024 00:51:10 INFO 139940816336704] loaded entry point class algorit
hm:model
[11/04/2024 00:51:10 INFO 139940816336704] Number of server workers: 4
[11/04/2024 00:51:10 INFO 139940816336704] loading model...
[11/04/2024 00:51:10 INFO 139940816336704] ...model loaded.
[2024-11-04 00:51:10 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-11-04 00:51:10 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-11-04 00:51:10 +0000] [1] [INFO] Using worker: sync
[2024-11-04 00:51:10 +0000] [43] [INFO] Booting worker with pid: 43
[2024-11-04 00:51:10 +0000] [52] [INFO] Booting worker with pid: 52
[2024-11-04 00:51:11 +0000] [61] [INFO] Booting worker with pid: 61
[2024-11-04 00:51:11 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681471.521616, "D
imensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operati
on": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "co
```

```
unt": 1, "min": 1, "max": 1}}}
[2024-11-04 00:51:10 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2024-11-04 00:51:10 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080
(1)
[2024-11-04 00:51:10 +0000] [1] [INFO] Using worker: sync
[2024-11-04 00:51:10 +0000] [43] [INFO] Booting worker with pid: 43
[2024-11-04 00:51:10 +0000] [52] [INFO] Booting worker with pid: 52
[2024-11-04 00:51:11 +0000] [61] [INFO] Booting worker with pid: 61
[2024-11-04 00:51:11 +0000] [70] [INFO] Booting worker with pid: 70
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681471.521616, "D
imensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Operati
on": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "co
unt": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.1536648,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 133.10599327087
402, "count": 1, "min": 133.10599327087402, "max": 133.10599327087402}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.1536648,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 133.10599327087
402, "count": 1, "min": 133.10599327087402, "max": 133.10599327087402}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681471.5218246, "EndTime": 1730681474.1617696,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.40594482421
875, "count": 1, "min": 136.40594482421875, "max": 136.40594482421875}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2701616,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.33935356140
137, "count": 1, "min": 137.33935356140137, "max": 137.33935356140137}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2703965,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 138.33999633789
062, "count": 1, "min": 138.33999633789062, "max": 138.33999633789062}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681471.5218246, "EndTime": 1730681474.1617696,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 136.40594482421
875, "count": 1, "min": 136.40594482421875, "max": 136.40594482421875}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2701616,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 137.33935356140
137, "count": 1, "min": 137.33935356140137, "max": 137.33935356140137}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681470.8273335, "EndTime": 1730681474.2703965,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 138.33999633789
062, "count": 1, "min": 138.33999633789062, "max": 138.33999633789062}, "in
vocations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681474.1538303, "EndTime": 1730681475.0288732,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 62.471389770507
```

```
81, "count": 1, "min": 62.47138977050781, "max": 62.47138977050781}, "invoc
ations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
#metrics {"StartTime": 1730681474.1538303, "EndTime": 1730681475.0288732,
"Dimensions": {"Algorithm": "LinearLearnerModel", "Host": "UNKNOWN", "Opera
tion": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 62.471389770507
81, "count": 1, "min": 62.47138977050781, "max": 62.47138977050781}, "invoc
ations.count": {"sum": 1.0, "count": 1, "min": 1, "max": 1}}}
2024-11-04T00:51:11.541:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPay
loadInMB=6, BatchStrategy=MULTI_RECORD
```

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

In [117…

```
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=colorma
    plt.title("Confusion Matrix")
    plt.tight_layout()
    plt.ylabel("True Class")
    plt.xlabel("Predicted Class")
    plt.show()
```

In [118…

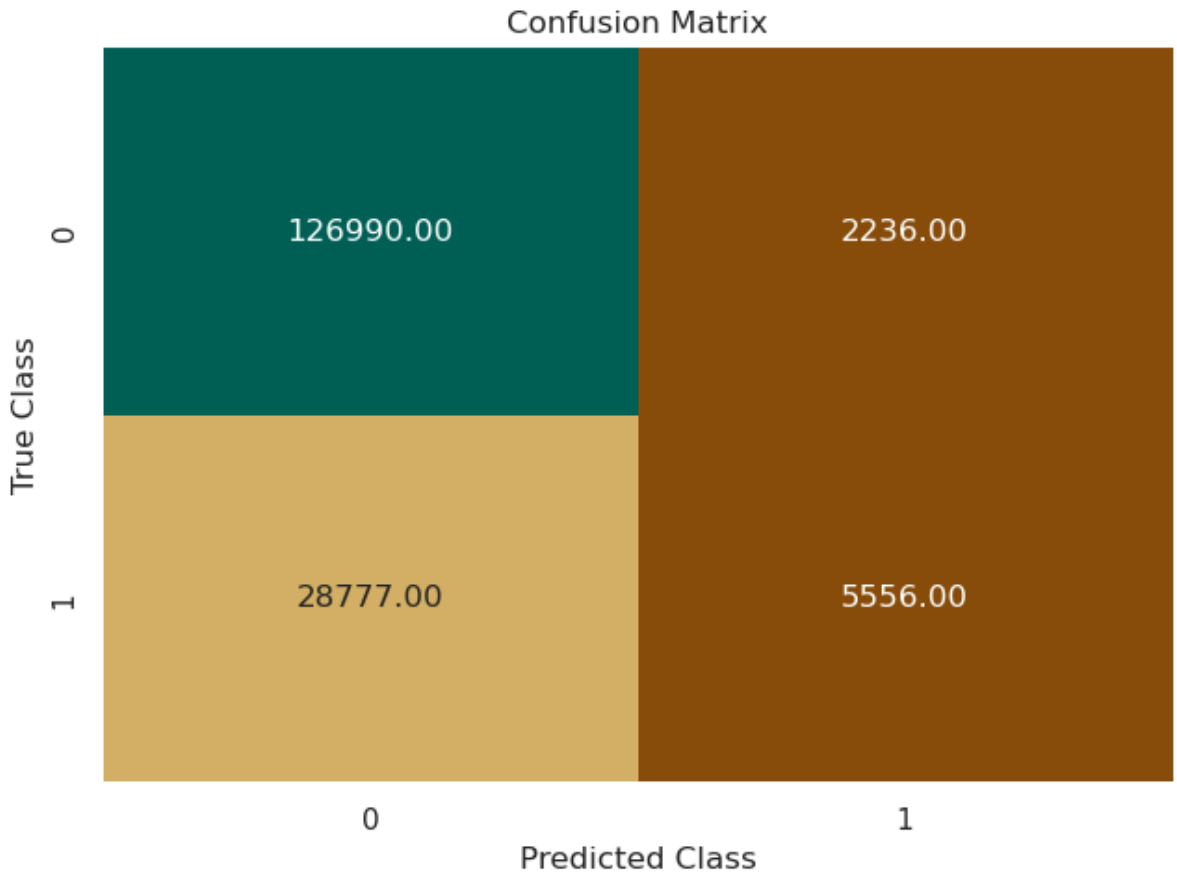
```
from sklearn import metrics

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_predic

    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=
    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:

In [119…

```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



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?

In [ ]:
```
# The TP is 129151 and the TN is 100
```

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.

Iteration II

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?

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.

In [69]:
```
# Source: http://www.calendarpedia.com/holidays/federal-holidays-2014.html

holidays_14 = ['2014-01-01',  '2014-01-20', '2014-02-17', '2014-05-26', '201
holidays_15 = ['2015-01-01',  '2015-01-19', '2015-02-16', '2015-05-25', '201
holidays_16 = ['2016-01-01',  '2016-01-18', '2016-02-15', '2016-05-30', '201
holidays_17 = ['2017-01-02', '2017-01-16', '2017-02-20', '2017-05-29' , '201
holidays_18 = ['2018-01-01', '2018-01-15', '2018-02-19', '2018-05-28' , '201
holidays = holidays_14+ holidays_15+ holidays_16 + holidays_17+ holidays_18

### Add indicator variable for holidays
data_orig['is_holiday'] = np.isin(data_orig['FlightDate'], holidays)
```

Weather data was fetched from https://www.ncei.noaa.gov/access/services/data/v1?dataset=daily-summaries&stations=USW00023174,USW00012960,USW00003017,USW00094846,USW0C 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.

In [70]:
```
!aws s3 cp s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_project
#!wget 'https://www.ncei.noaa.gov/access/services/data/v1?dataset=daily-summ
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/flight_delay_projec
t/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.

In [71]:
```
weather = pd.read_csv('/home/ec2-user/SageMaker/project/data/daily-summaries
station = ['USW00023174','USW00012960','USW00003017','USW00094846','USW00013
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*.

In [72]:
```
weather['MONTH'] = weather['DATE'].apply(lambda x: x.split('-')[1])
weather.head()
```

Out[72]:

	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	0
1	USW00023174	2014-01-02	22	0	NaN	NaN	159.0	256.0	100.0	LAX	0
2	USW00023174	2014-01-03	17	0	NaN	NaN	140.0	178.0	83.0	LAX	0
3	USW00023174	2014-01-04	18	0	NaN	NaN	136.0	183.0	100.0	LAX	0
4	USW00023174	2014-01-05	18	0	NaN	NaN	151.0	244.0	83.0	LAX	0

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.

In [73]:
```
weather.SNOW.fillna(0, inplace=True)
weather.SNWD.fillna(0, inplace=True)
weather.isna().sum()
```

Out[73]:
```
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.

In [76]:
```
idx = np.array([i for i in range(len(weather))])
TAVG_idx = idx[weather.TAVG.isna()]
TMAX_idx = idx[weather.TMAX.isna()]
TMIN_idx = idx[weather.TMIN.isna()]
TAVG_idx
```

Out[76]:
```
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])
```

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`.

In [77]:
```
weather_impute = weather.groupby(['MONTH', 'STATION']).agg({'TAVG':'mean','T
weather_impute.head(2)
```

Out[77]:

	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.

In [78]:
```
weather = pd.merge(weather, weather_impute,  how='left', left_on=['MONTH','S
.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.

In [79]:
```
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()
```

Out[79]:
```
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.

In [80]:
```
weather.drop(columns=['STATION','MONTH','TAVG_AVG', 'TMAX_AVG', 'TMIN_AVG',
```

Add the origin and destination weather conditions to the dataset.

In [81]:
```
### Add origin weather conditions
data_orig = pd.merge(data_orig, weather,  how='left', left_on=['FlightDate',
.rename(columns = {'AWND':'AWND_O','PRCP':'PRCP_O', 'TAVG':'TAVG_O', 'SNOW':
.drop(columns=['DATE','airport'])

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

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

In [82]:
```
sum(data.isna().any())
```

Out[82]:
```
0
```

In [83]:
```
data_orig.columns
```

Out[83]:
```
Index(['Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek', 'FlightDate',
       'Reporting_Airline', 'Origin', 'OriginState', 'Dest', 'DestState',
       'CRSDepTime', 'Cancelled', 'Diverted', 'Distance', 'DistanceGroup',
       'ArrDelay', 'ArrDelayMinutes', 'is_delay', 'AirTime', 'DepHourofDa
y',
       '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.

In [84]:
```
data = data_orig.copy()
data = data[['is_delay', 'Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWee
       'Reporting_Airline', 'Origin', 'Dest','Distance','DepHourofDay','is_h
       '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')
```

In [85]:
```
data_dummies = pd.get_dummies(data[['Year', 'Quarter', 'Month', 'DayofMonth'
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.

In [86]:
```
data.shape
```

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

In [87]:
```
data.columns
```

Out[87]:
```
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_U
A',
       'Reporting_Airline_WN', 'Origin_CLT', 'Origin_DEN', 'Origin_DFW',
       'Origin_IAH', 'Origin_LAX', 'Origin_ORD', 'Origin_PHX', 'Origin_SF
O',
       'Dest_CLT', 'Dest_DEN', 'Dest_DFW', 'Dest_IAH', 'Dest_LAX', 'Dest_OR
D',
       'Dest_PHX', 'Dest_SFO', 'is_holiday_True'],
      dtype='object')
```

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.

In [88]:
```
data.rename(columns = {'is_delay':'target'}, inplace=True )# Enter your code
```

Create the training sets again.

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

In [89]:
```
# Enter your code here
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
```

New baseline classifier

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

In [90]:
```
# Instantiate the LinearLearner estimator object
classifier_estimator2 = sagemaker.LinearLearner(role=sagemaker.get_executior
                                                instance_count=1,
                                                instance_type='ml,m4.xlarge'
                                                predictor_type='binary_class
                                                binary_classifier_model_sele
```

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 =
'cross_entropy_loss')
```

In [91]:
```
train_records = classifier_estimator2.record_set(train.values[:, 1:].astype(
val_records = classifier_estimator2.record_set(validate.values[:, 1:].astype
test_records = classifier_estimator2.record_set(test.values[:, 1:].astype(np
```

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

In [95]:
```
# Enter your code here
# Training the model with the specified datasets
classifier_estimator2.fit([train_records,val_records,test_records])
```

```
INFO:sagemaker.image_uris:Same images used for training and inference. Defa
ulting to image scope: inference.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
INFO:sagemaker:Creating training-job with name: linear-learner-2024-11-04-0
1-17-11-657
ERROR:sagemaker:Please check the troubleshooting guide for common errors: h
ttps://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-python-sdk-trouble
shooting.html#sagemaker-python-sdk-troubleshooting-create-training-job
```

```
---------------------------------------------------------------------------
ClientError                               Traceback (most recent call last)
Cell In[95], line 3
      1 # Enter your code here
      2 # Training the model with the specified datasets
----> 3 classifier_estimator2.fit([train_records,val_records,test_records])

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/workflow/pipeline_context.py:346, in runnable_by_pipeline.<locals>.wrapper(*args, **kwargs)
    342         return context
    344     return _StepArguments(retrieve_caller_name(self_instance), run_func, *args, **kwargs)
--> 346 return run_func(*args, **kwargs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/amazon/amazon_estimator.py:266, in AmazonAlgorithmEstimatorBase.fit(self, records, mini_batch_size, wait, logs, job_name, experiment_config)
    263 self._prepare_for_training(records, job_name=job_name, mini_batch_size=mini_batch_size)
    265 experiment_config = check_and_get_run_experiment_config(experiment_config)
--> 266 self.latest_training_job = _TrainingJob.start_new(
    267     self, records, experiment_config=experiment_config
    268 )
    269 if wait:
    270     self.latest_training_job.wait(logs=logs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/estimator.py:2498, in _TrainingJob.start_new(cls, estimator, inputs, experiment_config)
   2495 train_args = cls._get_train_args(estimator, inputs, experiment_config)
   2497 logger.debug("Train args after processing defaults: %s", train_args)
-> 2498 estimator.sagemaker_session.train(**train_args)
   2500 return cls(estimator.sagemaker_session, estimator._current_job_name)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:1055, in Session.train(self, input_mode, input_config, role, job_name, output_config, resource_config, vpc_config, hyperparameters, stop_condition, tags, metric_definitions, enable_network_isolation, image_uri, training_image_config, infra_check_config, container_entry_point, container_arguments, algorithm_arn, encrypt_inter_container_traffic, use_spot_instances, checkpoint_s3_uri, checkpoint_local_path, experiment_config, debugger_rule_configs, debugger_hook_config, tensorboard_output_config, enable_sagemaker_metrics, profiler_rule_configs, profiler_config, environment, retry_strategy, remote_debug_config, session_chaining_config)
   1050         logger.error(
   1051             "Please check the troubleshooting guide for common errors: %s", troubleshooting
   1052         )
   1053         raise e
-> 1055 self._intercept_create_request(train_request, submit, self.train.__name__)
```

```
File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:6606, in Session._intercept_create_request(self, request, create, func_name)
   6589 def _intercept_create_request(
   6590     self,
   6591     request: typing.Dict,
   (...)
   6594     # pylint: disable=unused-argument
   6595 ):
   6596     """This function intercepts the create job request.
   6597
   6598     PipelineSession inherits this Session class and will override
   (...)
   6604         func_name (str): the name of the function needed intercepting
   6605     """
-> 6606     return create(request)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:1053, in Session.train.<locals>.submit(request)
   1046 troubleshooting = (
   1047     "https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-python-sdk-troubleshooting.html"
   1048     "#sagemaker-python-sdk-troubleshooting-create-training-job"
   1049 )
   1050 logger.error(
   1051     "Please check the troubleshooting guide for common errors: %s", troubleshooting
   1052 )
-> 1053 raise e

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:1044, in Session.train.<locals>.submit(request)
   1042     logger.info("Creating training-job with name: %s", job_name)
   1043     logger.debug("train request: %s", json.dumps(request, indent=4))
-> 1044     self.sagemaker_client.create_training_job(**request)
   1045 except Exception as e:
   1046     troubleshooting = (
   1047         "https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-python-sdk-troubleshooting.html"
   1048         "#sagemaker-python-sdk-troubleshooting-create-training-job"
   1049     )

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/botocore/client.py:569, in ClientCreator._create_api_method.<locals>._api_call(self, *args, **kwargs)
    565     raise TypeError(
    566         f"{py_operation_name}() only accepts keyword arguments."
    567     )
    568 # The "self" in this scope is referring to the BaseClient.
--> 569 return self._make_api_call(operation_name, kwargs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/botocore/client.py:1023, in BaseClient._make_api_call(self, operation_name, api_params)
```

```
   1019     error_code = error_info.get("QueryErrorCode") or error_info.get
(
   1020         "Code"
   1021     )
   1022     error_class = self.exceptions.from_code(error_code)
-> 1023     raise error_class(parsed_response, operation_name)
   1024 else:
   1025     return parsed_response

ClientError: An error occurred (ValidationException) when calling the Creat
eTrainingJob operation: 1 validation error detected: Value 'ml,m4.xlarge' a
t 'resourceConfig.instanceType' failed to satisfy constraint: Member must s
atisfy enum value set: [ml.r5.12xlarge, ml.r5d.12xlarge, ml.m6i.xlarge, ml.
trn1.32xlarge, ml.p2.xlarge, ml.m5.4xlarge, ml.m4.16xlarge, ml.r5.24xlarge,
ml.r5d.24xlarge, ml.m6i.12xlarge, ml.p5.48xlarge, ml.m6i.24xlarge, ml.p4d.2
4xlarge, ml.t3.xlarge, ml.r5.16xlarge, ml.r5d.16xlarge, ml.trn2.48xlarge, m
l.g5.2xlarge, ml.c5n.xlarge, ml.p5e.48xlarge, ml.p3.16xlarge, ml.m5.large,
ml.m6i.16xlarge, ml.g6.2xlarge, ml.p2.16xlarge, ml.g5.4xlarge, ml.g6.4xlarg
e, ml.c4.2xlarge, ml.c5.2xlarge, ml.c6i.32xlarge, ml.c4.4xlarge, ml.c6i.xla
rge, ml.g6e.xlarge, ml.g5.8xlarge, ml.c5.4xlarge, ml.c6i.12xlarge, ml.g6e.1
2xlarge, ml.g6.8xlarge, ml.c5n.18xlarge, ml.g4dn.xlarge, ml.c6i.24xlarge, m
l.g6e.24xlarge, ml.g4dn.12xlarge, ml.c4.8xlarge, ml.g4dn.2xlarge, ml.c6i.2x
large, ml.g6e.48xlarge, ml.g6e.2xlarge, ml.c6i.16xlarge, ml.g6e.16xlarge, m
l.c5.9xlarge, ml.g4dn.4xlarge, ml.c6i.4xlarge, ml.g6e.4xlarge, ml.c5.xlarg
e, ml.g4dn.16xlarge, ml.c4.xlarge, ml.trn1n.32xlarge, ml.g6.xlarge, ml.g4d
n.8xlarge, ml.c6i.8xlarge, ml.g6e.8xlarge, ml.g6.12xlarge, ml.g5.xlarge, m
l.c5n.2xlarge, ml.t3.2xlarge, ml.t3.medium, ml.g6.24xlarge, ml.g5.12xlarge,
ml.g5.24xlarge, ml.c5n.4xlarge, ml.trn1.2xlarge, ml.g6.48xlarge, ml.c5.18xl
arge, ml.p3dn.24xlarge, ml.m6i.2xlarge, ml.g6.16xlarge, ml.g5.48xlarge, ml.
p5en.48xlarge, ml.r5.2xlarge, ml.r5d.2xlarge, ml.g5.16xlarge, ml.p3.2xlarg
e, ml.m6i.4xlarge, ml.m5.xlarge, ml.m4.10xlarge, ml.r5.4xlarge, ml.t3.larg
e, ml.r5d.4xlarge, ml.c5n.9xlarge, ml.m5.12xlarge, ml.m4.xlarge, ml.m5.24xl
arge, ml.m4.2xlarge, ml.m6i.8xlarge, ml.m6i.large, ml.p2.8xlarge, ml.m5.2xl
arge, ml.r5.8xlarge, ml.r5.xlarge, ml.r5.large, ml.r5d.8xlarge, ml.r5d.xlar
ge, ml.r5d.large, ml.m6i.32xlarge, ml.p4de.24xlarge, ml.p3.8xlarge, ml.m4.4
xlarge]
```

Perform a batch prediction by using the newly trained model.

In [96]:
```
# Enter your code here
test_labels, target_predicted = batch_linear_predict(test, classifier_estima
```

```
WARNING:sagemaker.estimator:No finished training job found associated with
this estimator. Please make sure this estimator is only used for building w
orkflow config
INFO:sagemaker:Creating transform job with name: linear-learner-2024-11-04-
01-18-41-419
```

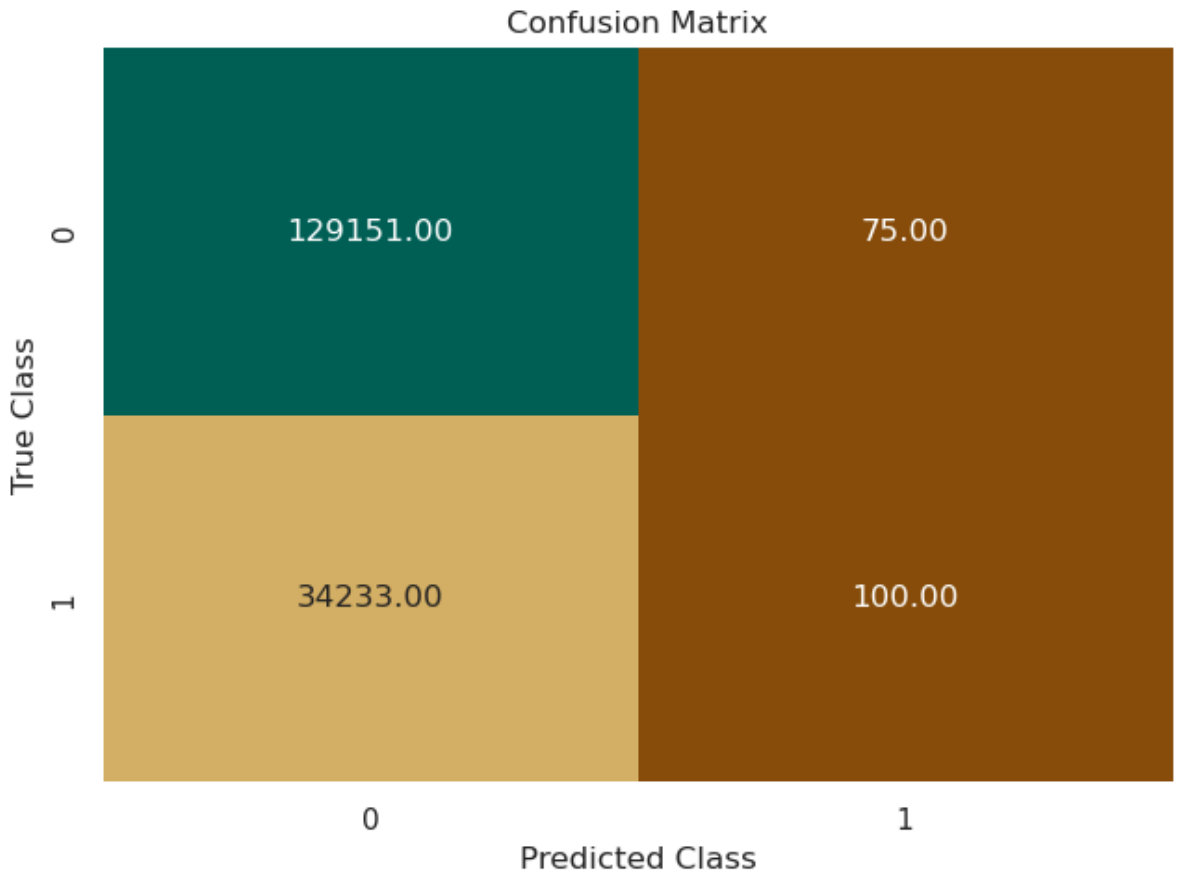
```
---------------------------------------------------------------------------
ClientError                               Traceback (most recent call last)
Cell In[96], line 2
      1 # Enter your code here
----> 2 test_labels, target_predicted = batch_linear_predict(test, classifi
er_estimator2)

Cell In[64], line 15, in batch_linear_predict(test_data, estimator)
      7 batch_input = "s3://{}/{}/batch-in/{}".format(bucket,prefix,batch_X
_file)
      9 classifier_transformer = estimator.transformer(instance_count=1,
     10                                                instance_type='ml.m4.xlarg
e',
     11                                                strategy='MultiRecord',
     12                                                assemble_with='Line',
     13                                                output_path=batch_output)
---> 15 classifier_transformer.transform(data=batch_input,
     16                                   data_type='S3Prefix',
     17                                   content_type='text/csv',
     18                                   split_type='Line')
     20 classifier_transformer.wait()
     22 s3 = boto3.client('s3')

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/workfl
ow/pipeline_context.py:346, in runnable_by_pipeline.<locals>.wrapper(*args,
**kwargs)
    342         return context
    344     return _StepArguments(retrieve_caller_name(self_instance), run_
func, *args, **kwargs)
--> 346 return run_func(*args, **kwargs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/transf
ormer.py:302, in Transformer.transform(self, data, data_type, content_type,
compression_type, split_type, job_name, input_filter, output_filter, join_s
ource, experiment_config, model_client_config, batch_data_capture_config, w
ait, logs)
    292 experiment_config = check_and_get_run_experiment_config(experiment_
config)
    294 batch_data_capture_config = resolve_class_attribute_from_config(
    295     None,
    296     batch_data_capture_config,
   (...)
    299     sagemaker_session=self.sagemaker_session,
    300 )
--> 302 self.latest_transform_job = _TransformJob.start_new(
    303     self,
    304     data,
    305     data_type,
    306     content_type,
    307     compression_type,
    308     split_type,
    309     input_filter,
    310     output_filter,
    311     join_source,
    312     experiment_config,
    313     model_client_config,
```

```
    314     batch_data_capture_config,
    315 )
    317 if wait:
    318     self.latest_transform_job.wait(logs=logs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/transformer.py:636, in _TransformJob.start_new(cls, transformer, data, data_type, content_type, compression_type, split_type, input_filter, output_filter, join_source, experiment_config, model_client_config, batch_data_capture_config)
    619 """Placeholder docstring"""
    621 transform_args = cls._get_transform_args(
    622     transformer,
    623     data,
   (...)
    633     batch_data_capture_config,
    634 )
--> 636 transformer.sagemaker_session.transform(**transform_args)
    638 return cls(transformer.sagemaker_session, transformer._current_job_name)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:3886, in Session.transform(self, job_name, model_name, strategy, max_concurrent_transforms, max_payload, input_config, output_config, resource_config, experiment_config, env, tags, data_processing, model_client_config, batch_data_capture_config)
   3883     logger.debug("Transform request: %s", json.dumps(request, indent=4))
   3884     self.sagemaker_client.create_transform_job(**request)
-> 3886 self._intercept_create_request(transform_request, submit, self.transform.__name__)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:6606, in Session._intercept_create_request(self, request, create, func_name)
   6589 def _intercept_create_request(
   6590     self,
   6591     request: typing.Dict,
   (...)
   6594     # pylint: disable=unused-argument
   6595 ):
   6596     """This function intercepts the create job request.
   6597
   6598     PipelineSession inherits this Session class and will override
   (...)
   6604         func_name (str): the name of the function needed intercepting
   6605     """
-> 6606     return create(request)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/session.py:3884, in Session.transform.<locals>.submit(request)
   3882 logger.info("Creating transform job with name: %s", job_name)
   3883 logger.debug("Transform request: %s", json.dumps(request, indent=4))
-> 3884 self.sagemaker_client.create_transform_job(**request)
```

```
File ~/anaconda3/envs/python3/lib/python3.10/site-packages/botocore/client.
py:569, in ClientCreator._create_api_method.<locals>._api_call(self, *args,
**kwargs)
    565     raise TypeError(
    566         f"{py_operation_name}() only accepts keyword arguments."
    567     )
    568 # The "self" in this scope is referring to the BaseClient.
--> 569 return self._make_api_call(operation_name, kwargs)

File ~/anaconda3/envs/python3/lib/python3.10/site-packages/botocore/client.
py:1023, in BaseClient._make_api_call(self, operation_name, api_params)
   1019     error_code = error_info.get("QueryErrorCode") or error_info.get
(
   1020         "Code"
   1021     )
   1022     error_class = self.exceptions.from_code(error_code)
-> 1023     raise error_class(parsed_response, operation_name)
   1024 else:
   1025     return parsed_response

ClientError: An error occurred (ValidationException) when calling the Creat
eTransformJob operation: Could not find model "linear-learner-2024-11-04-01
-18-41-418".
```

Plot a confusion matrix.

In [97]:
```
# 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.

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.

In [98]:
```
bucket='c134412a3409745l8230134t1w90541807286-flightbucket-yv3dkayyocwn'
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)

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: BaseNotebookInst
anceEc2InstanceRole
```

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

In [99]:
```
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}
```

In [100…
```
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.
```

In [101…
```
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-2024-11-0
4-01-21-09-174
```

```
2024-11-04 01:21:11 Starting - Starting the training job...
2024-11-04 01:21:26 Starting - Preparing the instances for training...
2024-11-04 01:21:55 Downloading - Downloading input data...
2024-11-04 01:22:35 Downloading - Downloading the training image......
2024-11-04 01:23:16 Training - Training image download completed. Training
in progress.[2024-11-04 01:23:26.868 ip-10-0-223-163.ec2.internal:7 INFO ut
ils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
INFO:sagemaker-containers:Imported framework sagemaker_xgboost_container.tr
aining
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 bi
nary: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 algo
rithm 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.
[01:23:31] 1308472x85 matrix with 111220120 entries loaded from /opt/ml/inp
ut/data/train?format=csv&label_column=0&delimiter=,
[01:23:32] 163559x85 matrix with 13902515 entries loaded from /opt/ml/inpu
t/data/validation?format=csv&label_column=0&delimiter=,
[2024-11-04 01:23:32.021 ip-10-0-223-163.ec2.internal:7 INFO json_config.p
y:91] Creating hook from json_config at /opt/ml/input/config/debughookconfi
g.json.
[2024-11-04 01:23:32.022 ip-10-0-223-163.ec2.internal:7 INFO hook.py:201] t
ensorboard_dir has not been set for the hook. SMDebug will not be exporting
tensorboard summaries.
[2024-11-04 01:23:32.022 ip-10-0-223-163.ec2.internal:7 INFO profiler_confi
g_parser.py:102] User has disabled profiler.
[2024-11-04 01:23:32.023 ip-10-0-223-163.ec2.internal:7 INFO hook.py:255] S
aving to /opt/ml/output/tensors
[2024-11-04 01:23:32.023 ip-10-0-223-163.ec2.internal:7 INFO state_store.p
y: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:0.65455#011validation-auc:0.65569
[2024-11-04 01:23:37.032 ip-10-0-223-163.ec2.internal:7 INFO hook.py:423] M
onitoring the collections: metrics
[2024-11-04 01:23:37.034 ip-10-0-223-163.ec2.internal:7 INFO hook.py:486] H
ook is writing from the hook with pid: 7
[1]#011train-auc:0.66319#011validation-auc:0.66430
[2]#011train-auc:0.67159#011validation-auc:0.67320
[3]#011train-auc:0.67357#011validation-auc:0.67489
[4]#011train-auc:0.67590#011validation-auc:0.67763
[5]#011train-auc:0.67874#011validation-auc:0.68021
[6]#011train-auc:0.68116#011validation-auc:0.68237
[7]#011train-auc:0.68241#011validation-auc:0.68375
[8]#011train-auc:0.68540#011validation-auc:0.68684
```

```
[9]#011train-auc:0.68720#011validation-auc:0.68842
[10]#011train-auc:0.68866#011validation-auc:0.68991
[11]#011train-auc:0.69096#011validation-auc:0.69245
[12]#011train-auc:0.69187#011validation-auc:0.69335
[13]#011train-auc:0.69463#011validation-auc:0.69593
[14]#011train-auc:0.69552#011validation-auc:0.69669
[15]#011train-auc:0.69712#011validation-auc:0.69829
[16]#011train-auc:0.69836#011validation-auc:0.69953
[17]#011train-auc:0.70039#011validation-auc:0.70139
[18]#011train-auc:0.70120#011validation-auc:0.70210
[19]#011train-auc:0.70237#011validation-auc:0.70304
[20]#011train-auc:0.70343#011validation-auc:0.70388
[21]#011train-auc:0.70453#011validation-auc:0.70483
[22]#011train-auc:0.70576#011validation-auc:0.70589
[23]#011train-auc:0.70646#011validation-auc:0.70642
[24]#011train-auc:0.70732#011validation-auc:0.70714
[25]#011train-auc:0.70762#011validation-auc:0.70739
[26]#011train-auc:0.70844#011validation-auc:0.70819
[27]#011train-auc:0.70922#011validation-auc:0.70889
[28]#011train-auc:0.70960#011validation-auc:0.70911
[29]#011train-auc:0.71100#011validation-auc:0.71051
[30]#011train-auc:0.71167#011validation-auc:0.71115
[31]#011train-auc:0.71202#011validation-auc:0.71150
[32]#011train-auc:0.71270#011validation-auc:0.71210
[33]#011train-auc:0.71325#011validation-auc:0.71264
[34]#011train-auc:0.71403#011validation-auc:0.71341
[35]#011train-auc:0.71506#011validation-auc:0.71438
[36]#011train-auc:0.71643#011validation-auc:0.71563
[37]#011train-auc:0.71686#011validation-auc:0.71600
[38]#011train-auc:0.71777#011validation-auc:0.71692
[39]#011train-auc:0.71817#011validation-auc:0.71730
[40]#011train-auc:0.71849#011validation-auc:0.71759
[41]#011train-auc:0.71878#011validation-auc:0.71787
[42]#011train-auc:0.71898#011validation-auc:0.71807
[43]#011train-auc:0.72016#011validation-auc:0.71916
[44]#011train-auc:0.72051#011validation-auc:0.71948
[45]#011train-auc:0.72110#011validation-auc:0.72001
[46]#011train-auc:0.72164#011validation-auc:0.72056
[47]#011train-auc:0.72203#011validation-auc:0.72092
[48]#011train-auc:0.72236#011validation-auc:0.72113
[49]#011train-auc:0.72323#011validation-auc:0.72184
[50]#011train-auc:0.72358#011validation-auc:0.72212
[51]#011train-auc:0.72381#011validation-auc:0.72228
[52]#011train-auc:0.72406#011validation-auc:0.72250
[53]#011train-auc:0.72437#011validation-auc:0.72276
[54]#011train-auc:0.72454#011validation-auc:0.72290
[55]#011train-auc:0.72474#011validation-auc:0.72306
[56]#011train-auc:0.72490#011validation-auc:0.72319
[57]#011train-auc:0.72511#011validation-auc:0.72338
[58]#011train-auc:0.72558#011validation-auc:0.72382
[59]#011train-auc:0.72589#011validation-auc:0.72418
[60]#011train-auc:0.72648#011validation-auc:0.72477
[61]#011train-auc:0.72674#011validation-auc:0.72503
[62]#011train-auc:0.72696#011validation-auc:0.72518
[63]#011train-auc:0.72720#011validation-auc:0.72537
[64]#011train-auc:0.72740#011validation-auc:0.72552
```

```
[65]#011train-auc:0.72800#011validation-auc:0.72617
[66]#011train-auc:0.72849#011validation-auc:0.72655
[67]#011train-auc:0.72855#011validation-auc:0.72662
[68]#011train-auc:0.72887#011validation-auc:0.72686
[69]#011train-auc:0.72918#011validation-auc:0.72713
[70]#011train-auc:0.72937#011validation-auc:0.72733
[71]#011train-auc:0.72957#011validation-auc:0.72755
[72]#011train-auc:0.72985#011validation-auc:0.72778
[73]#011train-auc:0.72992#011validation-auc:0.72784
[74]#011train-auc:0.73007#011validation-auc:0.72792
[75]#011train-auc:0.73024#011validation-auc:0.72805
[76]#011train-auc:0.73043#011validation-auc:0.72816
[77]#011train-auc:0.73055#011validation-auc:0.72831
[78]#011train-auc:0.73059#011validation-auc:0.72832
[79]#011train-auc:0.73092#011validation-auc:0.72862
[80]#011train-auc:0.73108#011validation-auc:0.72878
[81]#011train-auc:0.73138#011validation-auc:0.72908
[82]#011train-auc:0.73148#011validation-auc:0.72916
[83]#011train-auc:0.73160#011validation-auc:0.72920
[84]#011train-auc:0.73177#011validation-auc:0.72935
[85]#011train-auc:0.73191#011validation-auc:0.72947
[86]#011train-auc:0.73208#011validation-auc:0.72958
[87]#011train-auc:0.73215#011validation-auc:0.72967
[88]#011train-auc:0.73233#011validation-auc:0.72978
[89]#011train-auc:0.73254#011validation-auc:0.72995
[90]#011train-auc:0.73274#011validation-auc:0.73014
[91]#011train-auc:0.73288#011validation-auc:0.73023
[92]#011train-auc:0.73298#011validation-auc:0.73030
[93]#011train-auc:0.73310#011validation-auc:0.73042
[94]#011train-auc:0.73354#011validation-auc:0.73085

2024-11-04 01:27:17 Uploading - Uploading generated training model[95]#011t
rain-auc:0.73375#011validation-auc:0.73101
[96]#011train-auc:0.73386#011validation-auc:0.73109
[97]#011train-auc:0.73394#011validation-auc:0.73115
[98]#011train-auc:0.73411#011validation-auc:0.73124
[99]#011train-auc:0.73428#011validation-auc:0.73141

2024-11-04 01:27:30 Completed - Training job completed
Training seconds: 335
Billable seconds: 335
```

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

In [102…
```
batch_X = test.iloc[:,1:];
batch_X_file='batch-in.csv'
upload_s3_csv(batch_X_file, 'batch-in', batch_X)
```

In [103…
```
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-2024-11-04-01-28
-03-963
INFO:sagemaker:Creating transform job with name: sagemaker-xgboost-2024-11-
04-01-28-04-587
```

```
.....................................[2024-11-04:01:35:16:INFO] No GP
Us detected (normal if no gpus installed)
[2024-11-04:01:35:16:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:16: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 "{}";
    }
  }
}
[2024-11-04 01:35:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 01:35:16 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 01:35:16 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 01:35:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 01:35:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 01:35:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 01:35:16 +0000] [29] [INFO] Booting worker with pid: 29
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
2024-11-04T01:35:22.452:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPay
```

```
loadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652612 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652444 "-" "Go-http-client/1.1"
[2024-11-04:01:35:27:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652580 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652653 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652612 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652444 "-" "Go-http-client/1.1"
[2024-11-04:01:35:27:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652580 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652653 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:29 +0000] "POST /invocations HTTP/1.
1" 200 602646 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:29 +0000] "POST /invocations HTTP/1.
1" 200 602646 "-" "Go-http-client/1.1"

[2024-11-04:01:35:16:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:16:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:16:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
[2024-11-04:01:35:16:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:16:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:16: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 "{}";
    }
  }
}
[2024-11-04 01:35:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 01:35:16 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 01:35:16 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 01:35:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 01:35:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 01:35:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 01:35:16 +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 "{}";
    }
  }
}
[2024-11-04 01:35:16 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 01:35:16 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 01:35:16 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 01:35:16 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 01:35:16 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 01:35:16 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 01:35:16 +0000] [29] [INFO] Booting worker with pid: 29
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
```

```
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:01:35:22:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:01:35:22 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
[2024-11-04:01:35:23:INFO] Determined delimiter of CSV input is ','
2024-11-04T01:35:22.452:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPay
loadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652612 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652444 "-" "Go-http-client/1.1"
[2024-11-04:01:35:27:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652580 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652653 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652612 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652444 "-" "Go-http-client/1.1"
[2024-11-04:01:35:27:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652580 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:27 +0000] "POST /invocations HTTP/1.
1" 200 652653 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:29 +0000] "POST /invocations HTTP/1.
1" 200 602646 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:01:35:29 +0000] "POST /invocations HTTP/1.
1" 200 602646 "-" "Go-http-client/1.1"
```

Get the predicted target and test labels.

In [104…

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

Calculate the predicted values based on the defined threshold.

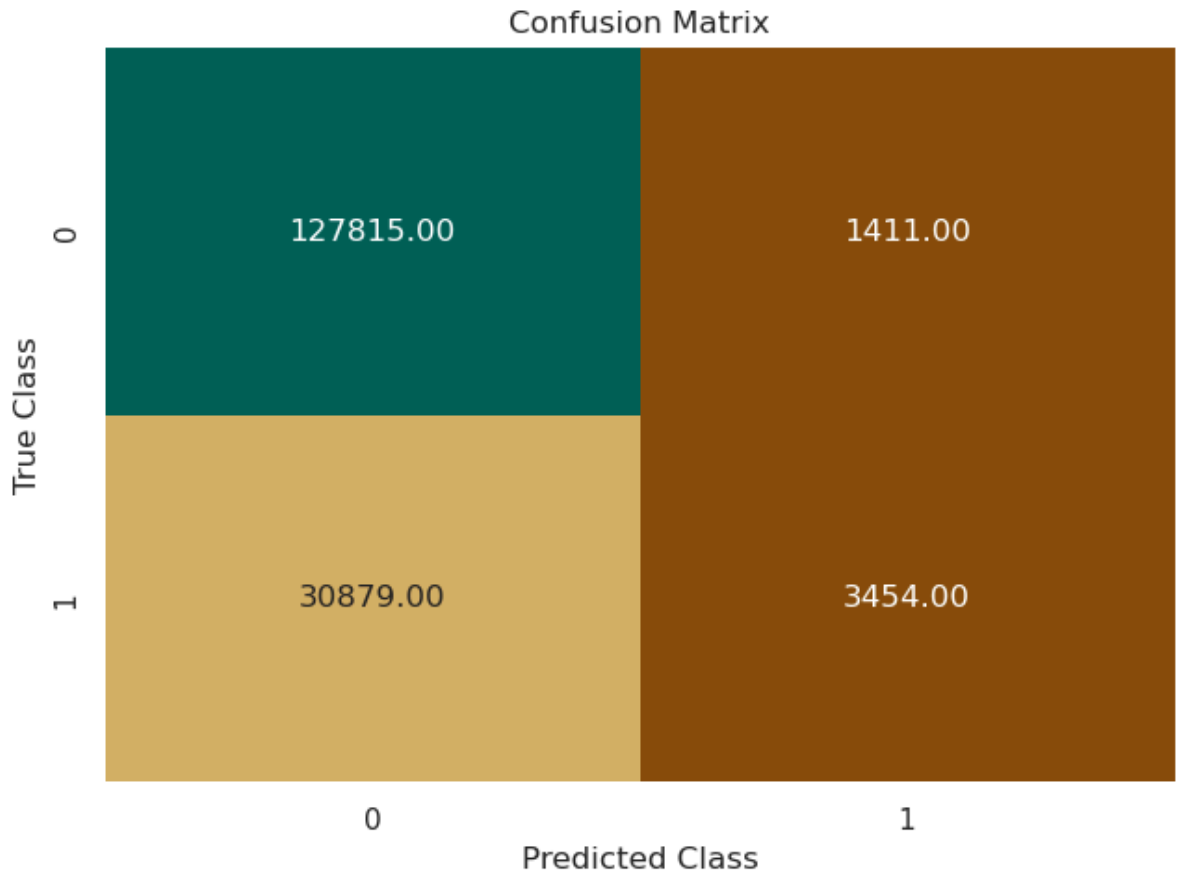
Note: The predicted target will be a score, which must be converted to a binary class.

In [105…
```
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  0.067604
1  0.102508
2  0.345982
3  0.075726
4  0.152937
   target
0       0
1       0
2       0
3       0
4       0
```

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

In [106…
```
# Inserted another confusion matrix
# Getting real familiar inserting these

plot_confusion_matrix(test_labels, target_predicted)
```



Try different thresholds

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

In [107…
```
#Enter your answer here
```

Hyperparameter optimization (HPO)

In [108…
```
from sagemaker.tuner import IntegerParameter, CategoricalParameter, Continuc

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

xgb = sagemaker.estimator.Estimator(container,
                                    role=sagemaker.get_execution_role(),
                                    instance_count= 1, # make sure you have
                                    instance_type=instance_type,
                                    output_path='s3://{}/{}/output'.format(b
                                    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=
```

```
                            'eta': ContinuousParameter(0.1, 0.5, scaling_type='
                            'min_child_weight': ContinuousParameter(3, 10, scal
                            '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
                            max_parallel_jobs=1)
```

In [109…
```
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 w
orkflow config
WARNING:sagemaker.estimator:No finished training job found associated with
this estimator. Please make sure this estimator is only used for building w
orkflow config
INFO:sagemaker:Creating hyperparameter tuning job with name: sagemaker-xgbo
ost-241104-0136
.....................................................................
!
```

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.

In [110…
```
boto3.client('sagemaker').describe_hyper_parameter_tuning_job(
    HyperParameterTuningJobName=tuner.latest_tuning_job.job_name)['HyperPara
```

Out[110]:
```
'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.

In [111…
```
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(HyperPar
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).dataf
tuner_df.head()
```

```
INFO:botocore.credentials:Found credentials from IAM Role: BaseNotebookInst
anceEc2InstanceRole
tuning job name:sagemaker-xgboost-241104-0136
best training job: sagemaker-xgboost-241104-0136-010-aa821baf

2024-11-04 02:39:18 Starting - Found matching resource for reuse
2024-11-04 02:39:18 Downloading - Downloading the training image
2024-11-04 02:39:18 Training - Training image download completed. Training
in progress.
2024-11-04 02:39:18 Uploading - Uploading generated training model
2024-11-04 02:39:18 Completed - Resource retained for reuse
```

Out[111]:

	alpha	eta	min_child_weight	num_round	subsample	TrainingJobName	Trai
0	9.396102	0.462110	6.666324	143.0	0.895170	sagemaker-xgboost-241104-0136-010-aa821baf	
1	16.008314	0.172030	8.033633	149.0	0.626674	sagemaker-xgboost-241104-0136-009-c03fda13	
2	48.622399	0.488527	6.411129	144.0	0.955533	sagemaker-xgboost-241104-0136-008-3226fa77	
3	106.782664	0.395059	8.026973	62.0	0.890813	sagemaker-xgboost-241104-0136-007-bb6d8768	
4	40.040316	0.357424	9.146715	118.0	0.904106	sagemaker-xgboost-241104-0136-006-2d749963	

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

Tip: See the previous XGBoost estimator fit function.

In [112…

```
# Enter your code here
best_estimator.fit(inputs=data_channels)
```

```
INFO:sagemaker:Creating training-job with name: sagemaker-xgboost-2024-11-0
4-02-39-36-851
```

```
2024-11-04 02:39:38 Starting - Starting the training job...
2024-11-04 02:39:51 Starting - Preparing the instances for training...
2024-11-04 02:40:18 Downloading - Downloading input data...
2024-11-04 02:40:54 Downloading - Downloading the training image......
2024-11-04 02:41:44 Training - Training image download completed. Training
in progress.[2024-11-04 02:41:53.419 ip-10-2-95-24.ec2.internal:7 INFO util
s.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
INFO:sagemaker-containers:Imported framework sagemaker_xgboost_container.tr
aining
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 bi
nary: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 algo
rithm 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
[02:41:58] 1308472x85 matrix with 111220120 entries loaded from /opt/ml/inp
ut/data/train?format=csv&label_column=0&delimiter=,
[02:41:58] 163559x85 matrix with 13902515 entries loaded from /opt/ml/inpu
t/data/validation?format=csv&label_column=0&delimiter=,
[2024-11-04 02:41:58.642 ip-10-2-95-24.ec2.internal:7 INFO json_config.py:9
1] Creating hook from json_config at /opt/ml/input/config/debughookconfig.j
son.
[2024-11-04 02:41:58.643 ip-10-2-95-24.ec2.internal:7 INFO hook.py:201] ten
sorboard_dir has not been set for the hook. SMDebug will not be exporting t
ensorboard summaries.
[2024-11-04 02:41:58.643 ip-10-2-95-24.ec2.internal:7 INFO profiler_config_
parser.py:102] User has disabled profiler.
[2024-11-04 02:41:58.644 ip-10-2-95-24.ec2.internal:7 INFO hook.py:255] Sav
ing to /opt/ml/output/tensors
[2024-11-04 02:41:58.644 ip-10-2-95-24.ec2.internal:7 INFO state_store.py:7
7] The checkpoint config file /opt/ml/input/config/checkpointconfig.json do
es 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:0.66288#011validation-auc:0.66374
[2024-11-04 02:42:04.960 ip-10-2-95-24.ec2.internal:7 INFO hook.py:423] Mon
itoring the collections: metrics
[2024-11-04 02:42:04.962 ip-10-2-95-24.ec2.internal:7 INFO hook.py:486] Hoo
k is writing from the hook with pid: 7
[1]#011train-auc:0.67921#011validation-auc:0.68027
[2]#011train-auc:0.68522#011validation-auc:0.68647
[3]#011train-auc:0.69046#011validation-auc:0.69158
[4]#011train-auc:0.69483#011validation-auc:0.69562
```

```
[5]#011train-auc:0.69854#011validation-auc:0.69898
[6]#011train-auc:0.70229#011validation-auc:0.70188
[7]#011train-auc:0.70502#011validation-auc:0.70442
[8]#011train-auc:0.70806#011validation-auc:0.70738
[9]#011train-auc:0.71210#011validation-auc:0.71095
[10]#011train-auc:0.71411#011validation-auc:0.71302
[11]#011train-auc:0.71668#011validation-auc:0.71509
[12]#011train-auc:0.71756#011validation-auc:0.71593
[13]#011train-auc:0.71947#011validation-auc:0.71782
[14]#011train-auc:0.72038#011validation-auc:0.71869
[15]#011train-auc:0.72145#011validation-auc:0.71949
[16]#011train-auc:0.72320#011validation-auc:0.72098
[17]#011train-auc:0.72501#011validation-auc:0.72284
[18]#011train-auc:0.72603#011validation-auc:0.72365
[19]#011train-auc:0.72703#011validation-auc:0.72463
[20]#011train-auc:0.72816#011validation-auc:0.72570
[21]#011train-auc:0.72933#011validation-auc:0.72668
[22]#011train-auc:0.72982#011validation-auc:0.72714
[23]#011train-auc:0.73044#011validation-auc:0.72764
[24]#011train-auc:0.73127#011validation-auc:0.72836
[25]#011train-auc:0.73168#011validation-auc:0.72859
[26]#011train-auc:0.73235#011validation-auc:0.72907
[27]#011train-auc:0.73318#011validation-auc:0.72992
[28]#011train-auc:0.73364#011validation-auc:0.73025
[29]#011train-auc:0.73480#011validation-auc:0.73127
[30]#011train-auc:0.73517#011validation-auc:0.73159
[31]#011train-auc:0.73580#011validation-auc:0.73204
[32]#011train-auc:0.73628#011validation-auc:0.73239
[33]#011train-auc:0.73686#011validation-auc:0.73301
[34]#011train-auc:0.73743#011validation-auc:0.73339
[35]#011train-auc:0.73753#011validation-auc:0.73344
[36]#011train-auc:0.73812#011validation-auc:0.73393
[37]#011train-auc:0.73855#011validation-auc:0.73429
[38]#011train-auc:0.73894#011validation-auc:0.73450
[39]#011train-auc:0.73919#011validation-auc:0.73464
[40]#011train-auc:0.73970#011validation-auc:0.73519
[41]#011train-auc:0.74011#011validation-auc:0.73542
[42]#011train-auc:0.74037#011validation-auc:0.73553
[43]#011train-auc:0.74098#011validation-auc:0.73612
[44]#011train-auc:0.74115#011validation-auc:0.73627
[45]#011train-auc:0.74173#011validation-auc:0.73657
[46]#011train-auc:0.74209#011validation-auc:0.73691
[47]#011train-auc:0.74262#011validation-auc:0.73728
[48]#011train-auc:0.74297#011validation-auc:0.73755
[49]#011train-auc:0.74342#011validation-auc:0.73789
[50]#011train-auc:0.74361#011validation-auc:0.73804
[51]#011train-auc:0.74418#011validation-auc:0.73846
[52]#011train-auc:0.74452#011validation-auc:0.73868
[53]#011train-auc:0.74499#011validation-auc:0.73898
[54]#011train-auc:0.74551#011validation-auc:0.73933
[55]#011train-auc:0.74576#011validation-auc:0.73955
[56]#011train-auc:0.74611#011validation-auc:0.73970
[57]#011train-auc:0.74642#011validation-auc:0.73992
[58]#011train-auc:0.74678#011validation-auc:0.74006
[59]#011train-auc:0.74714#011validation-auc:0.74036
[60]#011train-auc:0.74756#011validation-auc:0.74064
```

```
[61]#011train-auc:0.74787#011validation-auc:0.74090
[62]#011train-auc:0.74847#011validation-auc:0.74158
[63]#011train-auc:0.74855#011validation-auc:0.74165
[64]#011train-auc:0.74869#011validation-auc:0.74178
[65]#011train-auc:0.74917#011validation-auc:0.74211
[66]#011train-auc:0.74932#011validation-auc:0.74221
[67]#011train-auc:0.74961#011validation-auc:0.74238
[68]#011train-auc:0.74974#011validation-auc:0.74242
[69]#011train-auc:0.75006#011validation-auc:0.74271
[70]#011train-auc:0.75037#011validation-auc:0.74287
[71]#011train-auc:0.75064#011validation-auc:0.74309
[72]#011train-auc:0.75090#011validation-auc:0.74326
[73]#011train-auc:0.75104#011validation-auc:0.74330
[74]#011train-auc:0.75132#011validation-auc:0.74341
[75]#011train-auc:0.75178#011validation-auc:0.74374
[76]#011train-auc:0.75198#011validation-auc:0.74383
[77]#011train-auc:0.75231#011validation-auc:0.74409
[78]#011train-auc:0.75273#011validation-auc:0.74438
[79]#011train-auc:0.75293#011validation-auc:0.74458
[80]#011train-auc:0.75332#011validation-auc:0.74491
[81]#011train-auc:0.75375#011validation-auc:0.74516
[82]#011train-auc:0.75402#011validation-auc:0.74537
[83]#011train-auc:0.75440#011validation-auc:0.74567
[84]#011train-auc:0.75447#011validation-auc:0.74572
[85]#011train-auc:0.75468#011validation-auc:0.74582
[86]#011train-auc:0.75498#011validation-auc:0.74605
[87]#011train-auc:0.75511#011validation-auc:0.74615
[88]#011train-auc:0.75529#011validation-auc:0.74624
[89]#011train-auc:0.75557#011validation-auc:0.74639
[90]#011train-auc:0.75587#011validation-auc:0.74662
[91]#011train-auc:0.75601#011validation-auc:0.74679
[92]#011train-auc:0.75630#011validation-auc:0.74695
[93]#011train-auc:0.75642#011validation-auc:0.74702
[94]#011train-auc:0.75681#011validation-auc:0.74734
[95]#011train-auc:0.75703#011validation-auc:0.74752
[96]#011train-auc:0.75721#011validation-auc:0.74764
[97]#011train-auc:0.75745#011validation-auc:0.74773
[98]#011train-auc:0.75768#011validation-auc:0.74793
[99]#011train-auc:0.75791#011validation-auc:0.74804
[100]#011train-auc:0.75829#011validation-auc:0.74829
[101]#011train-auc:0.75853#011validation-auc:0.74832
[102]#011train-auc:0.75872#011validation-auc:0.74841
[103]#011train-auc:0.75893#011validation-auc:0.74854
[104]#011train-auc:0.75912#011validation-auc:0.74869
[105]#011train-auc:0.75939#011validation-auc:0.74882
[106]#011train-auc:0.75958#011validation-auc:0.74887
[107]#011train-auc:0.75980#011validation-auc:0.74894
[108]#011train-auc:0.75997#011validation-auc:0.74908
[109]#011train-auc:0.76015#011validation-auc:0.74926
[110]#011train-auc:0.76040#011validation-auc:0.74935
[111]#011train-auc:0.76065#011validation-auc:0.74953
[112]#011train-auc:0.76083#011validation-auc:0.74965
[113]#011train-auc:0.76095#011validation-auc:0.74969
[114]#011train-auc:0.76124#011validation-auc:0.74980
[115]#011train-auc:0.76148#011validation-auc:0.74989
[116]#011train-auc:0.76163#011validation-auc:0.74991
```

```
[117]#011train-auc:0.76179#011validation-auc:0.75000
[118]#011train-auc:0.76202#011validation-auc:0.75013
[119]#011train-auc:0.76229#011validation-auc:0.75031
[120]#011train-auc:0.76244#011validation-auc:0.75044
[121]#011train-auc:0.76264#011validation-auc:0.75052
[122]#011train-auc:0.76287#011validation-auc:0.75075
[123]#011train-auc:0.76307#011validation-auc:0.75088
[124]#011train-auc:0.76331#011validation-auc:0.75103
[125]#011train-auc:0.76344#011validation-auc:0.75113
[126]#011train-auc:0.76366#011validation-auc:0.75133
[127]#011train-auc:0.76389#011validation-auc:0.75153
[128]#011train-auc:0.76407#011validation-auc:0.75165
[129]#011train-auc:0.76422#011validation-auc:0.75175
[130]#011train-auc:0.76438#011validation-auc:0.75184
[131]#011train-auc:0.76461#011validation-auc:0.75189
[132]#011train-auc:0.76475#011validation-auc:0.75198
[133]#011train-auc:0.76488#011validation-auc:0.75206
[134]#011train-auc:0.76502#011validation-auc:0.75214
[135]#011train-auc:0.76516#011validation-auc:0.75221
[136]#011train-auc:0.76536#011validation-auc:0.75223
[137]#011train-auc:0.76550#011validation-auc:0.75235
[138]#011train-auc:0.76569#011validation-auc:0.75242
[139]#011train-auc:0.76591#011validation-auc:0.75256
[140]#011train-auc:0.76614#011validation-auc:0.75270
[141]#011train-auc:0.76634#011validation-auc:0.75275
[142]#011train-auc:0.76651#011validation-auc:0.75286

2024-11-04 02:48:34 Uploading - Uploading generated training model
2024-11-04 02:48:34 Completed - Training job completed
Training seconds: 496
Billable seconds: 496
```

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

In [113…

```
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-2024-11-04-02-49
-02-584
INFO:sagemaker:Creating transform job with name: sagemaker-xgboost-2024-11-
04-02-49-03-147
```

```
.....................................[2024-11-04:02:55:55:INFO] No GPUs
detected (normal if no gpus installed)
[2024-11-04:02:55:55:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:55:55: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 "{}";
    }
  }
}
[2024-11-04 02:55:55 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 02:55:55 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 02:55:55 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 02:55:55 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 02:55:55 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 02:55:55 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 02:55:56 +0000] [29] [INFO] Booting worker with pid: 29

[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
```

```
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
2024-11-04T02:56:01.979:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPay
loadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653444 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653452 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653212 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653444 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653452 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653212 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653483 "-" "Go-http-client/1.1"
[2024-11-04:02:56:08:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653483 "-" "Go-http-client/1.1"
[2024-11-04:02:56:08:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:02:56:10 +0000] "POST /invocations HTTP/1.
1" 200 603178 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:10 +0000] "POST /invocations HTTP/1.
1" 200 603178 "-" "Go-http-client/1.1"
[2024-11-04:02:55:55:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:55:55:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:55:55:INFO] nginx config:
worker_processes auto;
daemon off;
[2024-11-04:02:55:55:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:55:55:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:55:55: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 "{}";
    }
  }
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 "{}";
    }
  }
}
[2024-11-04 02:55:55 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 02:55:55 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 02:55:55 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 02:55:55 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 02:55:55 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 02:55:55 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 02:55:56 +0000] [29] [INFO] Booting worker with pid: 29
}
```

```
[2024-11-04 02:55:55 +0000] [19] [INFO] Starting gunicorn 19.10.0
[2024-11-04 02:55:55 +0000] [19] [INFO] Listening at: unix:/tmp/gunicorn.so
ck (19)
[2024-11-04 02:55:55 +0000] [19] [INFO] Using worker: gevent
[2024-11-04 02:55:55 +0000] [26] [INFO] Booting worker with pid: 26
[2024-11-04 02:55:55 +0000] [27] [INFO] Booting worker with pid: 27
[2024-11-04 02:55:55 +0000] [28] [INFO] Booting worker with pid: 28
[2024-11-04 02:55:56 +0000] [29] [INFO] Booting worker with pid: 29
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2024-11-04:02:56:01:INFO] No GPUs detected (normal if no gpus installed)
169.254.255.130 - - [04/Nov/2024:02:56:01 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] No GPUs detected (normal if no gpus installed)
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
[2024-11-04:02:56:03:INFO] Determined delimiter of CSV input is ','
2024-11-04T02:56:01.979:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPay
loadInMB=6, BatchStrategy=MULTI_RECORD
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653444 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653452 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653212 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653444 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653452 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653212 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653483 "-" "Go-http-client/1.1"
[2024-11-04:02:56:08:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:02:56:07 +0000] "POST /invocations HTTP/1.
1" 200 653483 "-" "Go-http-client/1.1"
[2024-11-04:02:56:08:INFO] Determined delimiter of CSV input is ','
169.254.255.130 - - [04/Nov/2024:02:56:10 +0000] "POST /invocations HTTP/1.
1" 200 603178 "-" "Go-http-client/1.1"
169.254.255.130 - - [04/Nov/2024:02:56:10 +0000] "POST /invocations HTTP/1.
1" 200 603178 "-" "Go-http-client/1.1"
```

In [114…
```
s3 = boto3.client('s3')
obj = s3.get_object(Bucket=bucket, Key="{}/batch-out/{}".format(prefix,'batc
target_predicted = pd.read_csv(io.BytesIO(obj['Body'].read()),sep=',',names=
test_labels = test.iloc[:,0]
```

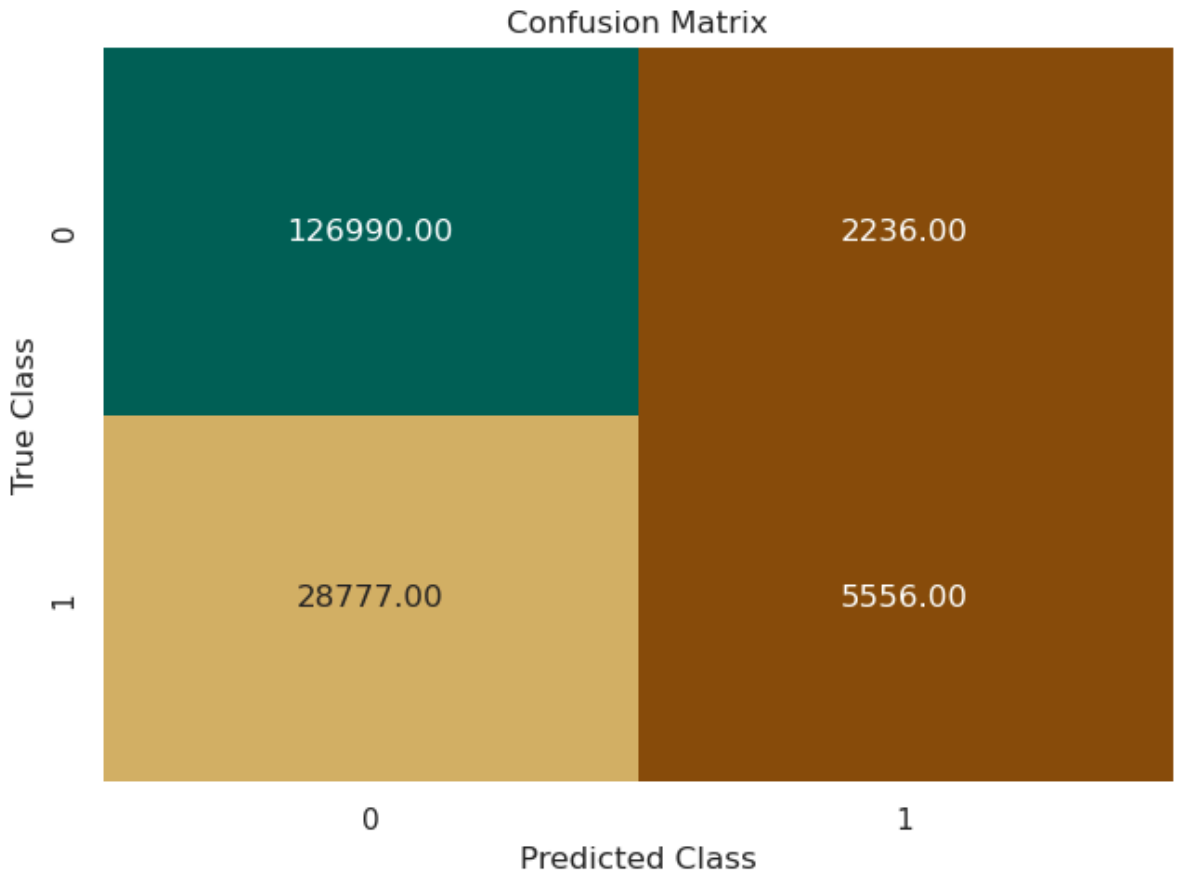
Get the predicted target and test labels.

In [115…
```
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  0.092488
1  0.090602
2  0.601977
3  0.049336
4  0.117867
   target
0       0
1       0
2       1
3       0
4       0
```

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

In [116…
```
# Enter your code here
plot_confusion_matrix(test_labels, target_predicted)
```



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

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.

In [ ]:
```
# Write your answers here
# I do feel like the model performance meets the business goal.
# Tried to keep it as simple possible ran into alot of issues on things I fe
# The biggest challenge I faced was classifying the data correctly
# Not to sure why I couldn't train my three datasets and peform a batch on t
# The biggest thing I learned is correctly structuring and gaining more unde
```