Week 1: Describe Dataset

Team 13: Ryan Orton, John Slater, Jason Papale

Summary of Tables

Currently, there are 3 tables we are intending to use for this project. The first table is the "weather table" (parquet file imported as a Spark DataFrame) downloaded from the National Oceanic and Atmospheric Administration (NOAA) website and provided to us as part of this project. In this table, each record represents a report that describe weather conditions, such as wind, visibility, and temperature. Each record also contains an identifier for the weather station collecting the data, as well as its latitude and longitude. Based on our initial exploration of this table, it appears as though there is at least one report for a given weather station every hour.

The second table we are intending to use for this project is the "flights table" (parquet file imported as a Spark DataFrame) obtained from the U.S. Department of Transportation (DOT) and provided to us as part of this class. This table contains information at an individual flight level for flights occurring between 2015-2019. Additionally, though we are still exploring the data, specific data elements that we have identified as potentially useful in predicting fight delays include: the departure time (in hourly bins), the departure location, the distance of the fight, the taxi time (for departure), as well as the carrier. This table also contains the outcome variable we intend to use for this project, DepDel15, which is an indication as to whether or not there was a departure delay of 15 minutes or more. Of note, when evaluating flight records which contained these variables (Q1 2015 data only), ~3% of the records contained a null value, and in the vast majority of these instances, the field with a null value was the outcome variable. In terms of the outcome variable, the dataset is a slightly imbalanced, with 74% of the flights having no departure delay of 15 minutes or greater, 23% having a departure delay of 15 minutes or greater, and ~3% having no information regarding this delay.

The third table (csv file imported as a Spark DataFrame) we intend to use is a table which contains the latitude and longitude of each airport. We decided to include this table in our project in order to help bridge the gap that currently appears to exist between the weather table and the flights table. That is, we would ideally like to be able to identify the weather conditions that a given flight was subject to at the time of departure. To do that, however, we will need to identify the weather station which closest to a given airport. Since this third table includes the latitude and longitude of each airport, we can now compare this information to the latitude and longitude of each weather station and find the weather station which is closest (likely using Euclidean distance) to each airport.

Approach

The approach we intend to take for this project is to build a baseline binary classification model and then continue to improve upon it as we better understand the data through exploratory data analysis, improve the usefulness of the data through feature engineering, and improve model performance through evaluating different model architectures and hyperparameter tuning. As mentioned above, one immediate next step we anticipate taking is to integrate the weather table into the flight table so we can include those data elements in the model. We intend to accomplish by creating a lookup table from the weather table which contains just one row per station identifier. We will then have a lookup table for the weather stations and a lookup table for the airports, both with a latitude and longitude for each location. We will then join these tables (using a cartesian join), calculate the distance between each location, and then sort those distances (partitioned by airport code), such that we can easily find the closest weather station to each airport. Though this operation will be computationally expensive (current testing indicates this operation will create ~ 686M records), it will be a necessary, one-time, intermediate step towards finding the closest weather station to each airport.

Once we have joined the datasets together, we will still need to take a number of steps to refine our features. For example, since some of the variables we intend to use as features are categorical (e.g. airline carrier, departure time), we will need to one-hot encode these features before including them in the model. We will also intend to normalize the features using the training data and then apply the mean and standard deviation values obtained in the process to normalize the test dataset before testing it on the model. For the continuous features, we also intend to explore Principal Component Analysis (PCA) to see if doing so can allow us to create a more parsimonious model. Once we complete the feature engineering, we intend to split the data into training, validation, and test datasets, using an 75/10/15 split, respectively. Additionally, though the specific metrics we will utilize to evaluate the effectiveness of our model may depend, in part, on the type of model utilized, we currently anticipate evaluating the effectiveness of our model by assessing the F1 score (mathematical formula shown below) of the positive class (i.e. prediction of delay). We can determine this numerically and we also intend to visualize it using a precision-recall curve. We also intend to evaluate model effectiveness using a Receiver Operating Characteristic (ROC) curve.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$