Flight Delay Prediction

Introduction

Flight delays are very common in today's world due to various reasons like weather, air traffic, overbooking and other logistic issues. This in turn affects both customer satisfaction as well as airline economy by bringing a negative economic impact to the airlines and also puts the airlines at a risk of losing loyal customers. Despite the abundance of literature on predicting flight delays, there is still scope for enhancing the efficacy of a streamlined process that can cater to scaling requirements.

So, it is crucial to predict the delays in advance w.r.t the responsible parameters. We propose to leverage delay prediction with various models and compare the results using PySpark and Spark ML. Our motive is to build an efficient and scalable system that can predict flight delays effectively.

Overview

Initially, we aim to refine the data by eliminating data points that contain null values or columns that contain irrelevant information. Once this is done, we will commence the training of all the models(Linear Regression, Gradient Descent Booster and Random Forest) with the same data. Finally, we will test the models and analyze the model with better performance.

Data

Our strategy for predicting flight delays involves utilizing pre-existing datasets on Kaggle (https://www.kaggle.com/datasets/giovamata/airlinedelaycauses). With approximately 200k data points, the dataset is sizable enough to enable the development of a scalable pipeline and draw meaningful insights. The dataset consists of several attributes like origin, destination, year, month, day, TaxiIn, TaxiOut etc. Then, we will carry out data preprocessing tasks to clean the data. In order to make fair comparisons between our model and other established models, we intend to train our model on the identical dataset employed by those standard models.

Metrics Used

Since our motive is to choose a state of art model, we plan to evaluate the models on certain common metrics. This would therefore help us to compare the model accuracy and find the best one. We would then accordingly evaluate the models on RMSE w.r.t the already split test data. The metrics that we use to compare the model performance are speedup, parallelizability and other metrics pertaining to systems in general. Then, we choose the best model by comparing the results of all these metrics.

Goals

Milestone:

- Achieve data pre-processing by removing unwanted attributes and eliminating null values rows.
- Construct pipelines with indexers that utilize transformed data for the required models.
- Use ParamGridBuilder, TrainValidationSplit for hyperparameter tuning and partitioning the validation data respectively.

Final:

- Train Logistic Regression and Random Forest Regression models using the training data.
- Integrate the models with the pipeline estimators.
- Evaluate the models using the metrics mentioned.
- Compare the outcomes of the models and graphically illustrate the superior performer.

Proposed extensions

In order to further enhance this project, we speculate to find the correlation between the features with the ‘delay’ parameter to reduce the dimensionality if possible. In addition, we also want to train more models so that we can find the state-of-art model which is effective in making predictions and which achieves the highest system performance.