ML4UVA Project Checkpoint: Flight Delays

Alex Foster, Brennen Muller, Gabi Turriago-Lopez

Abstract

This project explores the application of categorization and regression techniques to predict weather-induced departure delays for airline flights in the United States. Through the analysis of these weather- and airline- related factors, this project aims to improve understanding of aircraft departure delays for the benefit of travelers throughout Virginia and the wider United States.

The code for this project is available on GitHub.

Introduction

Every year, almost a quarter of domestic flights in the United States are delayed or canceled, totaling nearly four million delayed aircraft annually [**<empty citation>**]. Delayed and canceled flights are notoriously disruptive, especially because passengers often receive very little notice.

In addition, passengers are rarely informed about risk factors that could delay their flight, some of which, including weather systems and aircraft type, can be forecasted hours or days in advance. For some passengers, particularly those making connections on a tight schedule, knowing which flights are the most reliable is of great importance.

Therefore, the goal of this project is two-fold:

1. Predict the departure delay of flights based on known flight data and weather forecasts.
2. Analyze the relative impacts of each feature and identify potential risk groups.

Methods

Unfortunately, the root cause(s) of most departure delays are poorly documented. Although the causes for flight cancellations and arrival delays are clearly delineated within widely available flight datasets, the departure delay is not, making it more difficult to analyze the impacts of each feature individually. In addition, many departure delays are caused by factors that are inherently unpredictable and hard to account for using traditional algorithms: medical emergencies, personnel issues, and human error, among others.

With the expressed goal of predicting the departure delay for each flight, we prioritized training models and performing analysis based on factors that could be determined in advance, including the weather forecast, flight details and carrier airline for each aircraft.

Because of these limitations within our dataset, generating high-certainty predictions is nearly impossible. Instead, regression models must significantly outperform the standard linear regression across a variety of delayed and non-delayed flights to be considered successful.

Our models are based on the dataset titled *Airline Delay and Cancellation Data, 2009 – 2018* published on Kaggle by Yuanyu 'Wendy' Mu [1], which contains statistics for millions of domestic flights, including carrier airline, flight number, departure time, and others. Using these flight statistics, Alex Foster wrote a script to gather hourly weather data for each flight's origin airport, which is combined with the flight data to create our final dataset.

The data is processed in Python using the Pandas package. This is achieved through two pre-processing stages, a label extraction stage, and one final transformation stage described in more detail below.

Finally, using a random sample of the data, we trained seven models to predict flight departure delays:

- Linear Regression
- Random Forest Regression
- Hybrid K-Means / Linear Regression
- Hybrid SVM / Linear Regression
- XGBoost
- LightGBM
- MLPClassifier

Experiments

The following stages were used to process the data for training:

- Stage 1: Type Conversion and Correction

The first stage focuses on correcting erroneous values and manipulating data types.

Because Sci-kit Learn does not support date-time values automatically, each date is converted into separate day, month, and year values. Similarly, all hourly information is converted into a numeric type representing the number minutes after midnight of the current day (i.e. 1:20 AM becomes '80'). This prevents issues where hourly values are misinterpreted directly as integers (for example, 12:30 PM could be interpreted as '1230' and 1:30 PM as '130', which could cause adverse side effects during scaling and training) and ensures consistency in the training process.

- Stage 2: Feature Selection

Next, many features from the original dataset are dropped. Most of these features represent some form of duplicate data, including weather readings in both imperial and metric units and multiple versions of the date, time, and location values. In addition, all the statistics measured after departure must be removed from the final data, including airtime, taxi duration, and diversion statistics.

- Stage 3: Label Extraction

After pre-processing the data, we extract two different types of labels.

The first, which represents the departure delay in minutes, is the target label for the regression models. In order to train and analyze on-time, delayed, and canceled flights simultaneously, canceled flights are assigned a departure delay of two hours.

Second, each departure delay is converted into a 'delay status.' All flights delayed by at least fifteen minutes are considered 'delayed,' while any flight that departs within those first fifteen minutes are considered 'on-time,' which is used for training the categorical models. This threshold is consistent with the standard for arrival delay statuses [2].

- Stage 4: Transformation

Finally, the remaining features are transformed into machine-friendly data.

Missing numerical values are first imputed with that feature's median value, then all the values for that feature are scaled to match the standard distribution with a mean of zero and a standard deviation of one.

Categorical values are also imputed, instead using the most frequent value or category for that feature. As there were no ordered or sequential 'category' or 'object' features, all categorical values are encoded using one-hot vectors.

Finally, the dataset is divided into training, validation, and testing sets.

. . .

For this project, four models are trained and analyzed. The implementation for each model is described in more detail below:

- Linear Regression

Serving as the experimental baseline, the linear regression model is trained using Sci-kit Learn's SGDRegressor and evaluated on the validation set using the root mean squared error metric.

- Random Forest Regression

The first model of comparison is random forest regression, which can find more nuanced patterns in the training data than linear regression. This model is trained using a five-fold cross-validation grid search that adjusts the number of estimators and the maximum number of features from 5-30 and 5-50, respectively.

- Hybrid K-means / Linear Regression

The hybrid k-means algorithm consists of six total models: one k-means algorithm designed to separate the flights into five different 'risk groups,' followed by a linear regression for each risk group. Each linear model is trained using only the flights associated with that risk group.

- Hybrid SVM / Linear Regression

The hybrid SVM model operates on an analogous principle to the hybrid k-means algorithm. First, an SVM model predicts which flights are likely to be delayed. The flights identified as likely to be

'on-time' are predicted to depart with no delay, while the remaining flights are used to train a linear regression that determines how long the flight will be delayed. The SVM model is trained using a five-fold cross-validation grid search that adjusts the kernel, gamma value, and C value independently.

- XG BOOST

XGBoost, is a popular ensemmble method that trains decisions trees sequentially. We decided to try using it after implementing our RandomForest model to see if we could produce more accurate results. To our XGBoost model we added early stopping to avoid overfitting.

- LightGBM

LightGBM, similar to XGBoost, in that it trains trees sequentially correcting the errors of its predecessors we decided to attempt using it due to its speed, high accuracy, and its built-in regulation. We utilized GridSearchCV, to find the best parameters, we tested number of leaves, n-estimators, learning rate, and subsample.

- MLP

Finally, we decided to attempt a baseline Neural Network to see if our results would be drastically better. We utilized early-stopping to reduce overfitting. We also added adaptive learning rates for faster convergence. Last, we utilized GridSearchCV to optimize multiple parameters including the hidden layer sizes, the activation function, the learning rate.

Results

Here we will discuss the RMSE, our scoring metric, for each of the models as evaluated on the test dataset. The results showed an RMSE of around half an hour (31.99 minutes) for all models tested, with the exception of the K-Means and SVM hybrid categorization models which had anomalous RMSEs of 38970550732.92505 and 9940603635.653193, respectively. The dataset's standard deviation, however, was around 31 minutes, meaning our models did not perform any better (in fact, worse) than simply predicting the mean of the dataset every time. The linear regressor achieved an RMSE of 31.94, the random forest model got 31.27, and XGBoost scored 31.04. LightGBM did slightly better, with an RMSE of 30.70. Our FNN scored 33.92 for its RMSE.

Conclusion

Our project hypothesized that it was possible to predict airline delays with a mixture of categorization and regression models. We first explored a hybrid categorization approach in which we utilized k-means to create clusters of our training data, before running each cluster through an SGDRegressor. This initial approach had an extremely high RMSE (in the millions). Our next approach was to utilize the labels as a feature by concatenating it to the original data before running it through the all models only to find that our results did not improve at all. Therefore, we switched approaches and ran our other models without the hybrid approach. Again, these models performed similarly with an RMSE of around half an hour, which included Linear Regressor, Random Forest, XGBoost, LightGBM, and MLP. We concluded that the hybrid categorization approach did not perform appreciably better than the non-categorizing approach. Further, our dataset's standard

deviation, was around 31 minutes, meaning our models were no better than simply predicting the mean of the dataset every time.

Throughout the process, we made several efforts to improve our results including switching out our data, and creating a new dataset that combines weather and flight data by utilizing the WeatherAPI. This new dataset had over 33GB of data, spanning from 2009 to 2018. Unfortunately, we quickly ran into a new issue, we simply were unable to load and run our entire dataset due to computing limitations. Therefore, in the future, we hope to utilize Dask, which will allow us to access all 33GB of our dataset as we believe that the lack of data was a likely culprit behind the problems we faced. Further, although our new dataset was more comprehensive than those found on Kaggle, it lacked key airline-specific information including the make, model, and manufacture date.

In addition, we noticed when training our models that the training accuracy was much better than the validation accuracy, indicating that these models were overfit and simply could not generalize well. We made efforts to solve this problem by by adding GridSearchCV to find better parameters for our models to avoid overfitting in models such as RandomForest or XGBoost. Although the degree of overfitting was reduced, the models still didn't perform any better, indicating that these model architectures simply were not equipped for the problem. As a first step, we attempted a simple neural network, MLP, but in the future we hope to build a custom neural network implementation that could potentially handle the nuances of our data better.

While we were unable to properly predict flight delays, it is beneficial to continue experimenting and exploring this issue as it could have significant net benefits for Virginians. Every year, Virginians, experience thousands of flight delays, having the ability to reliably predict if and for how long a flight delay will occur will allow Virginians to have reduced stress, better planning, and help avoid unplanned expenses that are often unavoidable with flight delays.

Contributions

Alex Foster built the dataset by finding an existing dataset on Kaggle, processing it, and pulling weather data for each existing datapoint from WeatherAPI to create an updated dataset with weather features. He also wrote the script for, created animations for, and edited most of the video presentation. He also wrote parts of the proposal, checkpoint, and final report, and worked on the feature engineering for the hybrid categorization strategy.

Brennen Muller created the four-stage data processing pipeline, devised and trained four of the regression models, and began implementing load-on-demand and parallelization using the Dask framework.

In addition, Brennen contributed diagrams of the hybrid models, wrote and revised portions of the proposal, checkpoint, and final reports, and helped interpret project results.

Gabi Turriago created contributions to the data processing pipeline, devised and trained the XGBoost Model, LightGBM, the MLP Model, and the VotingRegressor Model. In addition, Gabi contributed to the ML4VA Video, working on compiling video data and creating an animation, worked on the script, helped film, and created maps with weather overlays. Worked on the proposal, checkpoint, and final report.

References

[1] Yuanyu Mu. *Airline Delay and Cancellation Data, 2009 - 2018*. 2019. URL: https://www.kaggle.com/datasets/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018/data. Accessed Dec. 7 2024.

[2] Bureau of Transportation. *Airline On-Time Performance and Causes of Flight Delays*. 2021. URL: https://www.bts.gov/explore-topics-and-geography/topics/airline-time-performance-and-causes-flight-delays. Accessed Dec. 7 2024.