Flight Delay Prediction
The Regressionists - Project Report

Ashwin Bhola ab8084
Chinmay Singhal cs5597
Sarthak Agarwal sa5154

Contents

1 Business Understanding

The challenge of managing the disruption caused by flight delays has always been a serious matter of concern for airlines and airports around the world. In the last two years, almost 20% of the flights departing from the John F. Kennedy International airport (JFK), New York have been delayed causing trouble not only to the airlines but also to the travellers. Domestic flight delays put a $32.9 billion dent in the U.S. economy, and about half that cost is borne by airline passengers, according to a study led by UC Berkeley researchers[1]. Flight delays and disruption cost the industry an estimated $8 billion every year and $20 billion to the customers in lost time and money. This number was calculated based on lost passenger time due to flight delays, cancellations and missed connections, as well as expenses for food and accommodations as a result of being away from home. The study found that airlines with high rates of delay also have higher operating costs overall. The direct cost to airlines included increased expenses for crew, fuel, and maintenance, among others. Inefficiency in air transportation also had indirect effects on the U.S. economy, the report said, decreasing productivity in other business sectors and reducing the 2007 U.S. gross domestic product (GDP) by $4 billion. In the airline world, delays build as the day wears on.

With this in mind, we decided to pursue the issue of predicting flight delays using machine learning. More precisely, our aim was to predict flight delays for flights departing from JFK using information like weather conditions, flight carrier, whether the day of flight falls in the holiday season and the time and month of flight. We trained our model on historical data of flight delays. Such a model is useful because:

- The analysis of air delays becomes vital since a better knowledge of their existence, and corresponding triggers, can improve the operational performance of airlines and, consequently airports, by anticipating delay and preparing schedules accordingly for example.
- Predicting departure delay is vital for customer satisfaction, reduced congestion at airports, preventing the ripple effect of one delay leading to another and also save costs.

2 Data Understanding

2.1 Data collection & feature extraction

Our main dataset includes every domestic flight departing from the John F Kennedy airport in the past five years. The data was collected from the United States Department of Transportation, Bureau of Transportation Statistics (BTS) website[2] for a period ranging from August 2013 to July 2018.
For each flight instance we extracted the following features-

1. **Flight carrier** - The dataset included flights details from 14 different domestic airlines. This feature is important as there may exist a hidden trend which causes some flight carriers to be more delayed than others.

2. **Departure month** - This feature captures the seasonality trend in the dataset. Certain months may be more busier than others which may cause greater delays in those specific months.

3. **Departure Time** - This feature captures the hourly trend in the dataset i.e. some hours of the day can be associated with greater delay as compared to others.

4. **Departure delay** - This feature tells us the number of minutes a particular flight was delayed after its scheduled departure time.

According to the BTS statistics, the major causes of departure delays are - air carrier delay, weather delay and national aviation system delay[3]. To account for these delays we used additional datasets.

1. **Air Carrier delay**- Carrier delay is within the control of the air carrier. The aviation data from BTS helps us account for this delay.

2. **Weather Delay** - Weather delay is caused by extreme or hazardous weather conditions that are forecasted or occur at the point of departure, enroute, or on point of arrival. To account for this delay, we used the weather data at the point of departure. Data was collected from the National Oceanic and Atmospheric Administration (NOAA) website[4] which included features like wind speed, precipitation (in mm), snow (in mm) and average temperature amongst other binary variables indicating the weather condition.

3. **National Aviation System Delay** - These delays are within the control of the National Airspace System (NAS) and may include: non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc. To account for these delays, a list of the major bank holidays in the US from 2013 to 2018 was taken and we engineered a feature to account for heavy traffic volume around these dates.

2.2 Selection Bias

Selection Bias in the sampling mechanism means favouring the presence of some particular groups over others. Our dataset contains publicly available data of all domestic flights that departed from JFK in the last five years, ensuring no selection bias during data preparation (since our datset represents the whole population). Although the BTS provides flight departure data for all the domestic carriers on all days, there is no way to check if selection bias may have been implicitly present in the data collection procedure used by BTS.

3 Data Preparation and Analysis

The final dataset thus includes features from the BTS flight statistics dataset, the NOA weather dataset, and one additional feature about major bank holidays in the years 2013 to 2018 that we engineered. Pandas and numpy libraries in Python were used to perform preprocessing on the data. The year, month and date in each dataset were converted to python date-time format and then used as the key for merging the flight, weather and holiday datasets.

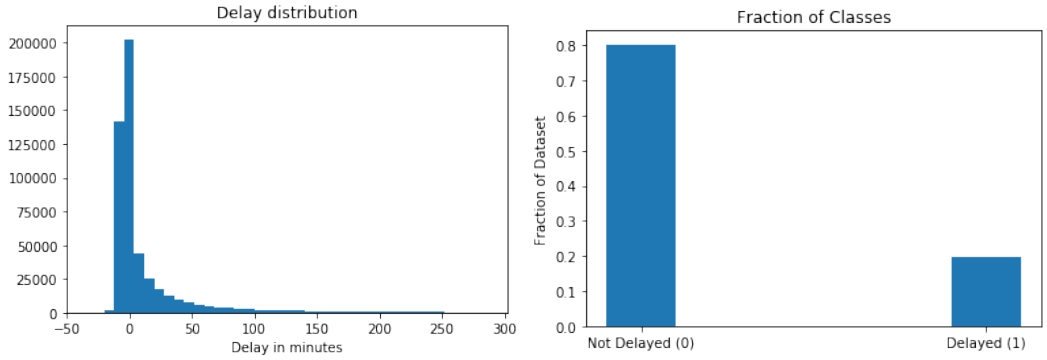
3.1 Target Variable

The target feature is departure delay, which tells the number of minutes a flight is delayed after its scheduled departure time. The original feature had a wide range of integer values, but since our business problem just focuses on predicting whether a flight will be delayed or not and not the actual time by which the flight is delayed, only two classes were used: delayed and non-delayed.

According to the Federal Aviation Administration (FAA), a flight is considered to be delayed if takes off and/or lands 15 minutes later than its scheduled time[4]. Taking this into account, our target variable was modified to take only binary values (0 or 1). 0 representing the flight is not-delayed (departure delay ≤ 15) and 1 representing the flight is delayed (departure delay > 15).

3.2 Summary of Feature Transformations

We plotted a distribution of departure delay (target variable) and also a plot of the fraction of the dataset that belongs in each class -



About 80% of the instances lie in the non-delayed region. Also, the 0.99 quantile of the distribution is around 200 and the 0.01 quantile is -20, so the instances having departure delay > 200 minutes or $<$ -20 minutes (flight departed more than 20 minutes before scheduled time) were considered outliers and removed.

Following is a summary of the further main transformations performed on our dataset -

- Changing departure delay to a binary feature. Following the FAA standard, departure delay values greater than 15 were transformed to 1 and those less than or equal to 15 were transformed to 0.
- Dropping the YEAR and DATE features from the dataset and applying a one-hot encoder to the MONTH feature, for getting dummy variables.
- Applying a floor function to the departure time feature to convert it to the lowest integer less than the given value and then applying a one-hot encoder to it.

- Handling carriers: A one-hot encoder was applied to the flight carrier names as well because algorithms require a numerical input and since no ordinal relationship exists between the different carriers, integer encoding was not enough.

3.3 Feature Engineering

A new feature named "HOLIDAY" was engineered and added to dataset that incorporates the effect of major bank holidays on flight delays. A list of major bank holidays was scraped from the internet[5] and a new weighted feature was created having values -

- 1 : If flight date is same as that of a holiday.
- 0.67 : If flight date is one day before or after a holiday
- 0.33 : If flight date is two days before or after a holiday.

The final dataset, after preprocessing, contains $499,766$ data instances and 63 features per instance, exluding the target variable.

4 Modelling and Evaluation

The task at hand is binary classification. As we have a class imbalance in our dataset, it is not feasible to use accuracy as a metric evaluating classifiers. So we chose the AUC value as our evaluation metric as it is base rate invariant. We also use k-fold cross-validation, with number of folds equal to 5, in order to get a robust estimate for our classifier performance and also to avoid overfitting.

4.1 Choice of Algorithms

Our **baseline model** is a logistic regression model (without hyperparameter tuning) with its performance assessed on just the historical data of flights without adding the holidays and weather features.

To improve upon our performance on baseline model, we used the following algorithms:

- **Logistic regression:** Logistic Regression has the advantage of being interpretable and computationally inexpensive and thus, we decided to use it as our baseline model, but it can't perform well in a large feature space and large sample sizes. Regularization strength and choice of norm in the penalty (l1 or l2) are the hyper-parameters to be tuned.

- **K nearest neighbours classifier:** Our baseline model was a linear classifier which didn't perform well and in order to increase the performance, we decided to choose a non-linear classifier. Also, KNN is easy to understand and is an intuitive algorithm but since it gets all of its information from the input's neighbors, localized anomalies affect outcomes significantly, compared to an algorithm that uses a generalized view of the data. k is one of the hyper-parameters to tune along with the choice of distance metric.

- **Decision tree Classifier:** Our decision to apply decision trees was motivated by the implicit feature selection performed by this non-linear classifier. We use entropy as the criterion and use a grid search to tune the parameters `min_samples_split` and `min_samples_leaf`. They can be interpreted easily but are prone to overfitting and can easily create complex trees that do not generalise well.

- **Random Forest:** Decision trees have high variance associated with them because of their unstable nature and thus, we decided to proceed to ensemble methods. We performed a grid search over the number of trees and max tree depth. The drawback of using random forests is that it performs worse than decision trees when the feature space is partially sparse.

- **XGBoost:** To improve upon the computational speed and performance of random forests, we decided to use XGBoost. Although a drawback is that it has sparsity penalties are associated with it.

Train-test split with a test set size fraction of 0.25 was used. 5-fold cross validation was performed on the training set for hyperparameter tuning, using grid search, for each of the above mentioned models.

We used Bootstrap to evaluate the effect of more data on the performance of the above mentioned models. It was found that the curve plateaus after

utilizing the five year data which indicates that it is not worth the effort to invest in more data. Fig 1 shows the effect of increasing the sample size on mean AUC for Decision Trees.

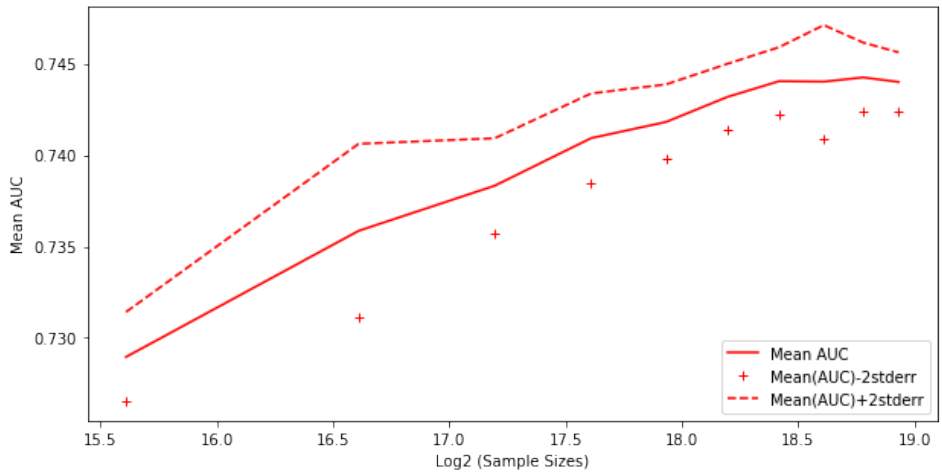


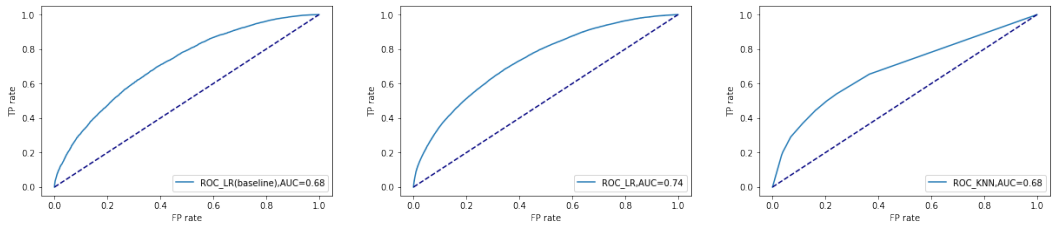
Figure 1: Bootstrap

4.2 Results

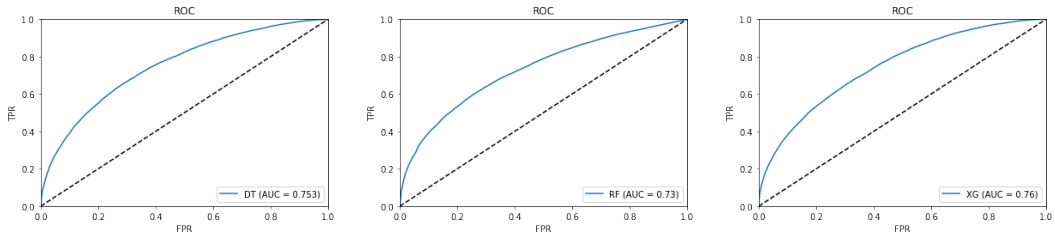
The performance of each of the model was evaluated on the basis of AUC and the results are showed in the following table -

Model	Precision	Recall	F1-Score	AUC
Baseline	0.79	0.81	0.75	0.68
Logistic Regression	0.79	0.82	0.76	0.74
KNN	0.78	0.81	0.79	0.68
Decision Trees	0.80	0.82	0.79	0.75
Random Forest	0.81	0.83	0.80	0.73
XGBoost	0.81	0.83	0.78	0.78

Among the models applied, we find that XGBoost has the highest AUC value (0.78), which is a considerable improvement over our baseline model (0.68). Figure 2 shows the ROC curves of all the models.



Baseline model | Logistic Regression | KNN



Decision Trees | Random Forest | XG-Boost

Figure 2: ROC Curves

4.3 Model's ability to solve the business problem

The model that we are trying to build is beneficial to the travelers, airlines and the JFK authorities. For the travellers, it would help them plan their schedule accordingly. Airport authorities and airlines will benefit as they can take into account the possibility of the delay and take preventive measures.
As mentioned above, we are using AUC to evaluate our model's performance as it is base rate invariant. AUC is a measure of our model's ability to separate between the two classes and so a good AUC indicates a good class separation capacity. Thus, our model has a reasonable performance and can predict delays with a significant confidence as compared to the baseline model.

5 Deployment

The model can be deployed in the form of a mobile application which takes the flight confirmation number and the airline name as input from the user and using our model, predicts if that particular flight will be delayed. Using the confirmation number and airline name we can extract the other details regarding the flight. For the weather features, we can use the weather forecast

for the flight departure time. Also, as holidays are known beforehand, the holiday features can be constructed easily. Using these features our model will output the probability of delay which will be displayed to the users.

As mentioned, the model will be deployed using an application. The model will use predicted weather which may affect the confidence of our prediction. To avoid this effect, the app will only predict the delay of flights departing in the next eight hours. Also, to evaluate the accuracy of model predictions, one may use A/B testing with the null hypothesis being that the model is able to accurately predict flight delays. In addition, A/B testing can also be used to improve upon the UI of the app.

Regarding the risks associated with the model, all the predictions should be taken with a grain of salt i.e. travellers should still arrive on the airport ahead of time if the model shows high probability of delay and the airport authorities should be skeptical even if the model predicts no delay with significant probability. As all the data used is open and public and the model is not designed to discriminate against any specific subset of airlines, the model can be assumed to have no ethical issues.

6 References

1. https://engineering.berkeley.edu/2010/11/flight-delays-cost-more-just-time
2. https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236
3. https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp
4. https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/GHCND:USW00094789/detail
5. https://en.wikipedia.org/wiki/Flight_cancellation_and_delay
6. https://gist.github.com/shivaas/4758439
7. http://cs229.stanford.edu/proj2016/report/DuperierSauvestreLeaf-ModelingFlightDelays-report.pdf
8. Alice Sternberg, Jorge Soares, Diego Carvalho, Eduardo Ogasawara, A Review on Flight Delay Prediction