Project Report   
EAI 6010

**Predicting Airline flight delays**   
**Members**: Abhijeet Gupta, Paritosh Singh, Sujay Torvi, Yash Yadav

**Summary:**

Flight delays have always been a huge source of financial loss and customer dissatisfaction for airlines, but recent years have seen the problem get noticeably worse. In 2018, FAA/Nextor estimated the annual costs of delays (direct cost to airlines and passengers, lost demand, and indirect costs) to be $30.2billion[[1]](#endnote-1). In 2019, this figure jumped to $33 billion[[2]](#endnote-2). This is in stark contrast to the 2012 estimated cost of $19.2 billion. While figures for more recent years have not been made available yet, covid has put a much worse strain on the airline industry, with many flight delays and cancellations being juggled with extra demand from travelers looking for vacations abroad as nations have relaxed covid restrictions on air travel.

There is a need to revisit and reduce inefficiencies in air travel, especially as demand increases and the knock-on effect of delays induces extra indirect costs and losses.

Through this project we would be trying to assess the potential for Machine Learning algorithms to assist in estimating flight delays and potentially highlighting areas of improvement. The hope is that models like these may be able to assist in monitoring and ultimately reducing airline delays as much as possible.

**Note:**

Refer to the Project description for high level overview of the dataset and our approach, this report will go into details of the project itself.

**Detailed Approach:**

Understanding the[**dataset**](https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGK&QO_fu146_anzr=b0-gvzr), Initially, we had 37 columns(Year, Quarter, Month, Day of Month, Day of the week, Flight date, Carrier name, Aircraft Tail number, Carrier flight number, Origin airport, Origin state name, Destination airport, Destination state name, Departure time, Departure delay in hhmm, Departure delay more than 15 mins, Departure delay group, Departure time block, Taxi in and out time in hhmm, Wheels off and wheels on time in hhmm, Arrival time in hhmm, Arrival delay in minutes and arrival time more than 15 mins, etc). To get information regarding the dataset you can refer to Description document. The dataset had 4-5 columns to define the date and time for flight, so we tried to combine it which will again provide the same information regarding a particular flight. Out of 37 columns we had 10 columns of object types. So to visualize the data, we chose to visualize numerical values because the categorical values were just static data about a particular aircraft(it’s, airline carrier, departure state name, etc.) We analyzed the flights which were delayed more than 15 minutes, by plotting bar graph of certain features, and also tried to understand average percentage of flights delayed.

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

First graph shows that on an average 20 % flights are getting delayed. Therefore, from the first graph, we got to know that in June and July months the percentage of flight delays increases and in month of December it increases. This shows that in extreme temperatures the flights generally get delayed, because June, July and December have extreme temperatures. In second graph where we are plotting average flight delays more than 15 minutes with day of week provides a lot of information added to previous graph. It shows that Thursdays and Fridays have more flight delay than Tuesday and Wednesdays. This also provides some information regarding the percentage of flights differ from flights on Thursdays/Fridays to those of on Tuesdays and Wednesdays, i.e., in case of Wednesdays and Fridays, the difference between average flight delay percentage is ~5%, which means that 5% of the total flights delays happening in single day of a week is around 5 %.

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

To analyze more about the percentage of delays of more than 15 mins, we plotted a bar graph with the Arrival time block and airlines. The percentage of flights delayed more than 15 mins with arrival time blocks shows that whichever flight arrives between dawn to dusk(nighttime) is delayed. And it is applicable all over US because the dataset that we scraped is for every state. It is obvious that flights which are arriving in time block from 11 PM to 6 AM have a high delay time. And arrival delay increases after5 PMM. For this information there are several backings, like, flights might be delayed due to natural reasons, or there is huge traffic of incoming flights at airports. The aim must be clear, we are predicting whether a flight’s arrival is delayed more than 15 mins or not. In the case of graph on right, there are some airlines whose aircraft delays more than 15 mins. For complete information refer to the below table. Airlines like B6(Jet Blues), G4(Allegiant Air LLC), and WN(Southwest Airlines) have the most delayed flight percentage. The next step of our project is EDA which is explained below.

|  |  |
| --- | --- |
| Carrier Code | Airline Name |
| 9E | Endeavor Air Inc. |
| AA | American Airlines Inc. |
| AS | Alaska Airlines Inc. |
| B6 | JetBlue Airways |
| C5 | Commutair Aka Champlain Enterprises, Inc |
| DL | Delta Air Lines Inc. |
| F9 | Frontier Airlines Inc. |
| G4 | Allegiant Air |
| G7 | GoJet Airlines LLC d/b/a United Express |
| HA | Hawaiian Airlines Inc. |
| MQ | Envoy Air |
| NK | Spirit Air Lines |
| OH | PSA Airlines Inc. |
| OO | SkyWest Airlines Inc. |
| PT | Piedmont Airlines |
| QX | Horizon Air |
| UA | United Air Lines Inc. |
| WN | Southwest Airlines Co. |
| YV | Mesa Airlines Inc. |
| YX | Republic Airways |
| ZW | Air Wisconsin Airlines Corp |

EDA:

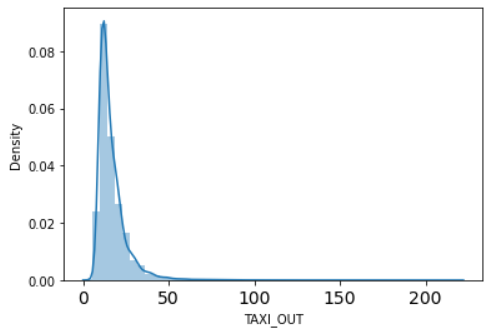
Chart, box and whisker chart

Description automatically generatedChart, histogram

Description automatically generatedThe first step that we chose is to check sparsity per column so that we can remove those columns which have high sparsity(>80%). Sparsity simply means missing data in the dataset. Here we denoted missing data in terms of percentage, and we will remove columns having greater than 80 percent of missing values. There for the column having high sparsity were, CANCELLATION\_CODE, CARRIER\_DELAY, WEATHER\_DELAY, NAS\_DELAY, SECURITY\_DELAY, LATE\_AIRCRAFT\_DELAY. After removing those columns we decided to remove those redundant columns which makes less sense and also the extra columns(it is mentioned in the code file) we made before that. FL\_DATE means the date of a particular flight, it was of data type ‘object’, and as we already added a column with date of flight having data type ‘datetime’. That is why we removed that. After this step the columns we were left with were: ['YEAR', 'QUARTER', 'MONTH', 'DAY\_OF\_MONTH', 'DAY\_OF\_WEEK', 'OP\_UNIQUE\_CARRIER', 'TAIL\_NUM', 'OP\_CARRIER\_FL\_NUM', 'ORIGIN', ’ORIGIN\_STATE\_NM', 'DEST', 'DEST\_STATE\_NM', 'DEP\_TIME', 'DEP\_DELAY', 'DEP\_DELAY\_NEW', 'DEP\_DEL15', 'DEP\_DELAY\_GROUP', 'DEP\_TIME\_BLK', 'TAXI\_OUT', 'WHEELS\_OFF', 'WHEELS\_ON', 'TAXI\_IN', 'ARR\_TIME', 'ARR\_DELAY', 'ARR\_DEL15', 'ARR\_DELAY\_GROUP', 'ARR\_TIME\_BLK', 'CANCELLED', 'AIR\_TIME', 'FLIGHTS', 'DATE']. In next step we try to analyze data in depth, for that we split the data into numerical and categorical variables. For categorical variables, we used Label Encoder because it encodes unique classes in integers. The reason why we did not use One Hot Encoder because it will increase the number of columns in the dataset as we had more than 1000 categories in in one of the columns and other than that we had many categories in other columns as well. In case of Numerical variable, we tried looking for outliers. So out of these numerical columns ('YEAR','QUARTER', 'MONTH', 'DAY\_OF\_MONTH', 'DAY\_OF\_WEEK', 'DEP\_TIME', 'DEP\_DELAY\_GROUP', 'TAXI\_OUT', 'WHEELS\_OFF', 'WHEELS\_ON', 'ARR\_TIME', 'ARR\_DEL15', 'AIR\_TIME', 'FLIGHTS') we got outliers in AIR\_TIME, DEP\_DELAY\_GROUP and TAXI\_OUT. We tried to understand their distribution and Kernel Density Estimation. But in case of DEP\_DELAY\_GROUP, we have to ignore our search for outliers, since it represents a category.

In the above distribution plot of AIR\_TIME, there are extreme outliers that go up to 750 units. And for assistance there is a bar plot is also given. The second quartile range is around 95units. As there are a lot of outliers so instead of imputing those outliers to mean we imputed those to the 90-percentile(205.0) range. So that we won’t make major changes in the dataset and can effectively use models on it.

Chart

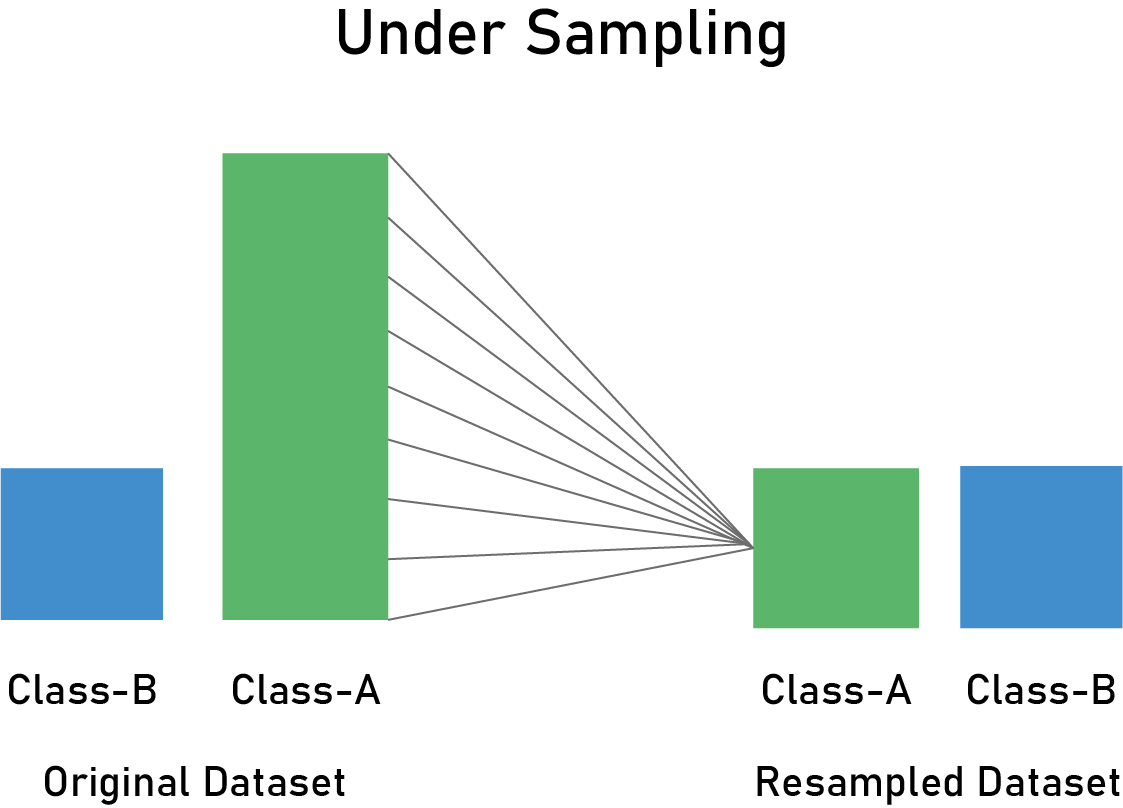
Description automatically generated

And here also we will do the same on the TAXI\_OUT column, as it has outliers ranging till 250 where this column’s mean is ~19. So, we will do the same here and will replace outliers with a 90-percentile range. Even this makes sense, if we replace outliers with mean, it means we are intentionally ignoring outlier values and so will our model. That’s not ideal. Now the next step is dealing with null values. In the final dataset, we had 2% of missing values, so it is easy to remove than imputing values, we are trying to keep the data real, with no synthetic values. After this we plotted the correlation matrix:

Chart

Description automatically generated

The above matrix shows the least correlation between variables, but in YEAR, QUARTER, and MONTH there is a high correlation, but instead of diving into time series data modeling. So, we will keep these three columns in our dataset. Now the last problem self to solve is balancing the dataset. We use the Random Under Sampling technique. Initially, the dataset had 80% of the majority class(0.0: when the flight was not delayed more than 15 mins.) and 20% of the minority class(1.0: when the flight was delayed more than 15 mins). Random Under Sampling(RUS) helped us achieve the 50-50 weightage of both classes. Meaning, RUS chooses randomly samples from the population(complete data of the majority class).



Text

Description automatically generated with medium confidence

The final dataset after RUS :

Just to check how well we did EDA, we tried to run a Decision tree, if our EDA is good then we will get a good f1 score, and will also tell us how RUS has performed.

Table

Description automatically generated

This shows we performed a good EDA on our raw data. This will help in accurately predicting the arrival delay(more than 15 mins) really well and the model will take comparatively less time.

Modeling:

1. Random Forest:

Classification algorithm consisting of many **decision trees**.

It uses **bagging** and **feature randomness** when building each individual tree to try to create an **uncorrelated forest of trees** whose **prediction by committee** is more accurate than that of any individual tree.

Chart, diagram

Description automatically generated

**Logic** - What feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible).

A picture containing diagram

Description automatically generated

**Bagging**-Random Forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as **bagging**.

**Feature Randomness** - each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.

What do we need for our random forest to make accurate class predictions?

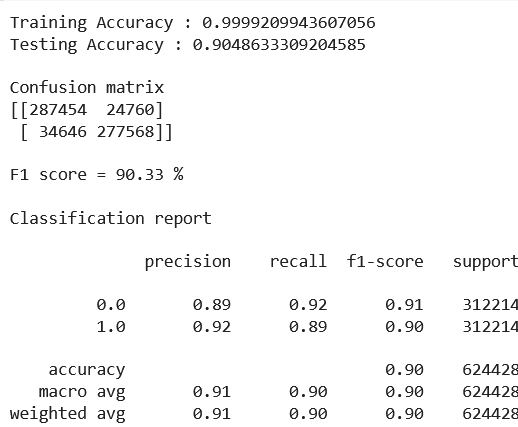
* We need features that have at least some predictive powers. After all, if we put garbage in then we will get garbage out.
* The trees of the forest and more importantly their predictions need to be uncorrelated (or at least have low correlations with each other). While the algorithm itself via feature randomness tries to engineer these low correlations for us, the features we select and the hyper-parameters we choose will impact the ultimate correlations as well.

Here, for the prediction of our data we will perform Grid Search CV and use stratified K-fold technique to tune hyperparameters in RF.



And after that implementing random forest, we got these results:

Chart, line chart

Description automatically generated

1. **AdaBoost Algorithm**:

AdaBoost ML algorithm is a boosting technique that consists of multiple decision tree estimators the model learns by identifying the examples that it incorrectly predicted and it learns to correctly predict them by increasing the importance weights of the wrongly predicted records and ultimately the final prediction is calculated by taking a majority vote of the number of decision trees in the AdaBoost algorithm. In the AdaBoost algorithm, we have ‘n’ number of decision tree stumps. A stump is a decision tree with depth = 1.

**Icon

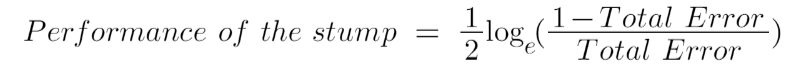
Description automatically generated with medium confidence**

Each stump is a weak learner which tries to predict the label, and those learners which have failed to predict certain records are retrained on datasets with the wrongly predicted records having more weights.

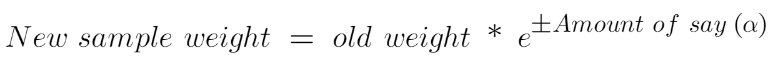
**Diagram

Description automatically generated**

* We calculate the performance of each of the learners using this formula:

****

And we adjust the weights of the dataset based on this formula.



* After adjusting the weights of the dataset we train the learners on those records having more weights as compared to those with fewer weights. At the end the prediction happens based on majority vote of our weak learners.  On our dataset we run the adaboost algorithm by trying out various number of estimators 12,24,48,76. As you can see in the above screenshot, we achieved 0.90 F1 score on both the test and the train data. This means the model is not overfitting and has decently captured the patterns in the data.

Table

Description automatically generated

We get decent precision and recall as well. Below shown is our confusion matrix which shows how our model performed with respect to true and false positives/negatives.

Chart, treemap chart

Description automatically generated

As you can see below, we get a 95.5% AUC score. This means that our model makes a mistake and predicts false positives/negatives only 4.5% time.

Chart

Description automatically generated

1. **XgBoost Algorithm**:

XGBoost ML algorithm is very similar to Adaboosting ML Technique which uses decision tree stumps and boosts the weak learners to achieve strong prediction performance. In XGBoost the weak learners are trained one by one and each learner tries to improve upon the previous learner’s mistake. We improve each learner by using gradient descent algorithm and we optimize the tree by calculating the loss (cross entropy, squared error, etc.) and optimizing it using gradient descent.

Diagram

Description automatically generated

* We try to adjust parameters of the tree and try to improve the predictions of the tree model. We keep doing this until the number of estimators in our model are exhausted. The prediction of the last estimator is the best as it has learned from all its predecessors and gives us strongest prediction. We train our model and we get a decent F1-score and precision/recall. As you can see the AUC is 95.7% which is slightly better than the one recorded on Adaboost.

Table

Description automatically generated

**Chart, line chart

Description automatically generated**

**Results:**

<ACTUAL RESULTS HERE>

Conclusion:

As we have used different ensembling techniques, the cumulative results can be seen below:

Background pattern

Description automatically generated

As Random Forest, AdaBoost has the same f1-score, which means they are predicting well than the Decision tree and Neural network. But keep in mind their AUC score, we will choose Random Forest for further implementation

**References:**

1. <https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2021.pdf> [↑](#endnote-ref-1)
2. <https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2022.pdf> [↑](#endnote-ref-2)