Flight Delay Prediction

Shreyas Srinivasan

Abstract

A flight delay occurs if a flight departes or arrives later than the scheduled time. This delay is affected by factors such as weather conditions, air traffic, thunderstorm and other weather glitches. This project aims to predict whether a flight will be delayed or not and the extent of delay time, using a two-stage machine learning model trained on a dataset, based on the weather data given.

1 Introduction

Flight delay affects airlines, airports and passengers. Loss for airline companies due to flight delays has been increasing over the years with an average loss of over 6 billion dollars annually. Annual loss to passengers surpass an average of 14 billion dollars. Hence prediction of flight delays is essential as it can help passengers and airlines reduce financial loss significantly and also give passengers a fair idea of how much time they will be losing, so that they can reschedule their plans.

This project aims to build a two-stage model to predict whether a flight will be delayed or not, and the extent of delay based on Weather and Flight Data across 15 airports in USA from the years 2016 to 2017. Different classification and regression models are studied and compared in this project.

2 Dataset and Preprocessing

The flight data contains data of all domestic flights in the USA between the years 2016 and 2017. The individual flight details which have their origin and destination in the 15 specified airports comprise the required Flight Dataset.

The weather data comprises of hourly weather conditions recorded in the 15 specified airports. It was a json file that was restructured into csv to get the Weather Dataset.

Flight and Weather Datasets are merged twice seperately for the purpose of integrating departure and arrival details of the flights in the final dataset. First the Weather Dataset is merged with the Flight Dataset based on, Departure Date, Absolute Departure Time and Departure Airport. The merged dataset is then merged with Flight Dataset again based on, Arrival Date, Absolute Arrival Time and Arrival Airport to obtain the final dataset.

The tables below consist of Airport Codes, Weather and Flight Details respectively.

ATL	CLT	DEN	DFW	EWR
IAH	JFK	LAS	LAX	MCO
MIA	ORD	PHX	SEA	SFO

Table 1: Chosen Airport Codes

WindSpeedKmph	WindDirDegree	WeatherCode	precipMM
Visibility	Pressure	Cloudcover	DewPointF
WindGustKmp	tempF	WindChillF	Humidity
date	time	airport	

Table 2: Weather Details

FlightDate	Quarter	Year	Month
DayofMonth	DepTime	DepDel15	CRSDepTime
DepDelayMinutes	OriginAirportID	DestAirportID	ArrTime
CRSArrTime	ArrDel15	ArrDelayMinutes	

Table 3: Flight Details

3 Classification

It is essential to know if a flight will be delayed or not first before finding out the extent of delay. Hence, the first stage is classification. In this stage, we classify whether the flight is delayed or not by 15 minutes or more. The ground truth used is ArrDel15, which gives details of the same.

$$ArrDel15 = 1.0 \quad if \;\; flightdelay >= 15\, minutes \tag{1}$$

$$ArrDel15 = 0.0 \quad if \;\; flightdelay < 15\, minutes \tag{2}$$

3.1 Classification Metrics

To evaluate classifier models, we use the following metrics.

TN: True Negative
The model has predicted correctly that a flight will not be delayed.

TP: True Positive
The model has predicted correctly that a flight will be delayed.

FN: False Negative
The model has predicted wrongly that a flight will not be delayed.

FP:False Positive
The model has predicted wrongly that a flight will be delayed.

- **Accuracy**

  Accuracy is the ratio of true results to the total number of results that are examined.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

- **Precision**

  Ratio of true positives to total positives. Gives a proportion of predictive positives which are truly positive.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

- **Recall**

  It tells the proportion of actual positives are correctly classified.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

- **F_1-Score**

  F_1-Score is the harmonic mean of precision and recall.

$$F_1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

3.2 Classifier Models

Different classifier models are used and the best classifier is chosen based on $F1$-Score as it gives a more accurate performance of the classifier by taking the harmonic mean of the Precision and Recall values. The different classifier models used are Logistic Regressor, Decision Tree Classifier, Gradient Boosting Classifier, Extra Trees Classifier and Random Forest Classifier.

3.3 Classifier Performance

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regressor	0.92	0.89	0.98	0.68	0.95	0.77	0.92
Decision Tree Classifier	0.92	0.69	0.91	0.71	0.92	0.70	0.87
Gradient Boosting Classifier	**0.92**	**0.90**	**0.98**	**0.70**	**0.95**	**0.79**	**0.92**
Extra Trees Classifier	0.89	0.57	0.89	0.57	0.89	0.57	0.82
Random Forest Classifier	0.92	0.89	0.98	0.70	0.95	0.78	0.92

Table 4: Classifier Perfomance

Based on the above results, the **Gradient Boosting Classifier** can be inferred as the best classifier, as it has the highest $F1$-Score for both classes

0 and 1, compared to other classifiers. But it can be observed that the F1-Score of Class 0 is higher than Class 1. This could be due to the imbalance of delayed and non-delayed flights which causes an inflation in results due to larger proportion of negative values compared to positive values.

4 Data Imbalance Problem

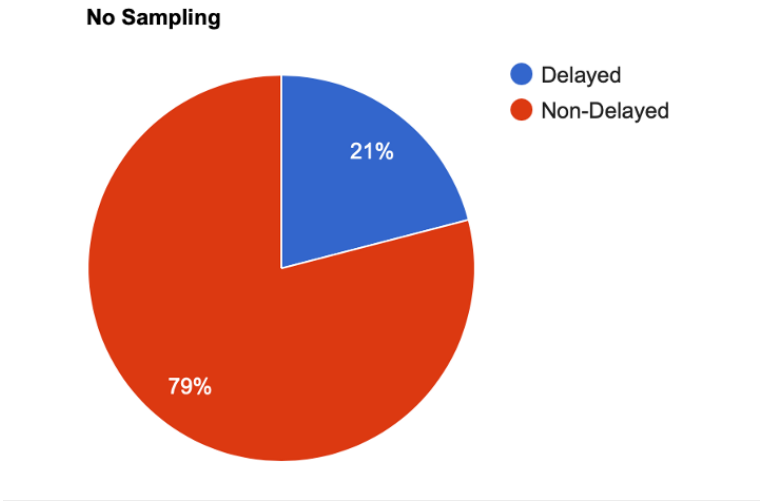

Figure 1: Dataset Distribution Before Sampling

In the above classification algorithms, the performance of Class 1 (delayed flights) is poorer than Class 0 (non-delayed flights). This could be due to larger number of non-delayed flights present in the dataset, as shown in Fig 1. Hence a process called sampling is applied which makes the dataset balanced by making the number of delayed and non-delayed flights equal either by adding or removing data from the training dataset. The best classifier can be chosen after applying some sampling techniques.

Table 4 shows the classifier performance before overcoming the bias.

This bias in the dataset can be overcome by applying Oversampling or Undersampling Techniques like the following

- **Random Under Sampler**

  This technique involves randomly removing data from the majority class.

- **Near Miss**

  In this technique, we eliminate majority class examples by checking if there are instances of two different classes that are very close to each other in the feature space. We remove the instances of the majority class to increase the space between the two classes.

- **Random Over Sampler**

  This technique involves randomly duplicating data from the minority class and adding it to the training data.

- **Synthetic Memory Oversampling Technique(SMOTE)**

  In this technique, the new instances are generated by randomly selecting one or more of the k-nearest neighbors for each instance in the feature space in the minority class.

4.1 Undersampling

Undersampling involves removal or elimination of data in the majority class so that the number of flights in each class will be equal. The number of flights in each class after undersampling has been shown in Fig 2 below. Random Under Sampler and Near Miss techniques are used for undersampling in this project.

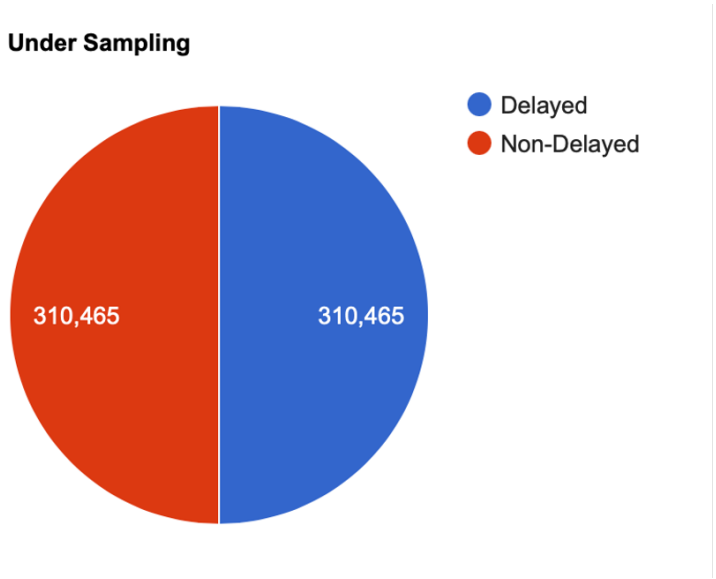

Figure 2: Dataset Distribution After Undersampling

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	0.90
Decision Tree Classifier	0.94	0.51	0.79	0.81	0.86	0.62	0.80
Gradient Boosting Classifier	**0.95**	**0.73**	**0.92**	**0.81**	**0.93**	**0.76**	**0.90**
Extra Trees Classifier	0.90	0.39	0.70	0.71	0.79	0.50	0.71
Random Forest Classifier	0.95	0.72	0.92	0.81	0.93	0.76	0.89

Table 5: Classifier Performance using Random Under Sampler

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	0.90
Decision Tree Classifier	0.94	0.51	0.79	0.81	0.86	0.62	0.80
Gradient Boosting Classifier	**0.95**	**0.73**	**0.92**	**0.81**	**0.93**	**0.76**	**0.90**
Extra Trees Classifier	0.90	0.39	0.70	0.71	0.79	0.50	0.71
Random Forest Classifier	0.95	0.72	0.92	0.81	0.93	0.76	0.89

Table 6: Classifier Performance using Near Miss

After applying Near Miss Undersampling and Random Under Sampler techniques, the $F1$-Scores are poorer compared to the the results before sampling as shown in Table 4. Hence, the dataset is oversampled to observe if results get better.

4.2 Oversampling

Oversampling involves duplication of data from the minority class and adding it to the training data so that number of flights will be equal in each class. The number of flights in each class has been shown in Fig 3 below. Random Over Sampler and Synthetic Memory Oversampling Technique(SMOTE) are used for oversampling in this project.

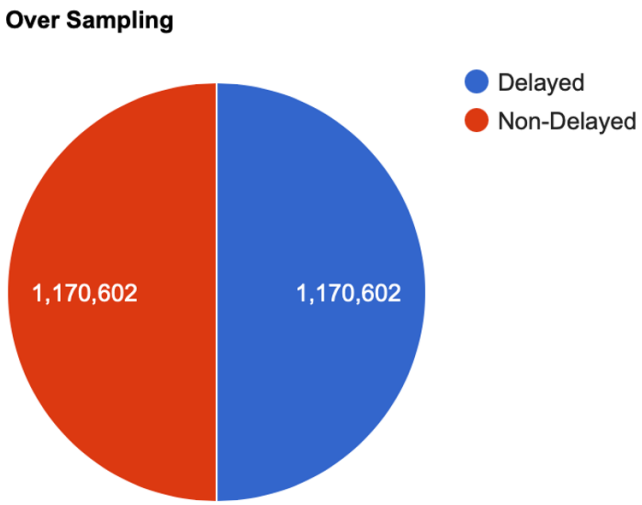

Figure 3: Dataset Distribution After Oversampling

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	0.90
Decision Tree Classifier	0.92	0.69	0.92	0.70	0.92	0.70	0.87
Gradient Boosting Classifier	**0.95**	**0.73**	**0.92**	**0.81**	**0.93**	**0.77**	**0.90**
ExtraTrees Classifier	0.89	0.60	0.89	0.59	0.89	0.59	0.83
Random Forest Classifier	0.93	0.83	0.96	0.74	0.95	0.78	0.91

Table 7: Classifier Performance using Random Over Sampler

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.94	0.74	0.93	0.78	0.93	0.76	0.90
Decision Tree Classifier	0.92	0.67	0.91	0.71	0.91	0.69	0.87
Gradient Boosting Classifier	**0.93**	**0.87**	**0.97**	**0.71**	**0.95**	**0.78**	**0.92**
Extra Tree Classifier	0.89	0.50	0.84	0.62	0.86	0.55	0.79
Random Forest Classifier	0.94	0.80	0.95	0.77	0.94	0.78	0.91

Table 8: Classifier Performance using SMOTE

After applying Random Over Sampler and SMOTE techniques, the F1-Scores don't significantly change than before sampling as shown in Table 4. However, SMOTE algorithm involves selecting one or more of the k-nearest members for each instance. This is better than random duplication of data done using Random Under Sampler. Hence, SMOTE is considered as a better oversampling technique.

As **Gradient Boosting Classifier** has the best F1-Score compared to other classifiers after sampling using **SMOTE**, it is chosen as the best classifier.

5 Regression

After finding out if a flight is delayed or not, it is important to know the extent of delay. The delayed flights are filtered based on ArrDel15 = 1.0. Different regression models are used to obtain the delay period (in minutes) of flights. The ground truth will be ArrDelayMinutes, which is the time difference between the CRSArrTime (scheduled Arrival Time) and ArrTime (actual Arrival Time).

5.1 Regression Metrics

To evaluate the regressor models, we use the following metrics.

The following notations stand for :
$\bar{Y}$: Mean Value Of distribution of Y
$\hat{Y}$: Predicted Value Of Y
N: Number of Data Points

- **Mean Absolute Error**

$$Mean\ Absolute\ Error(MAE) = \frac{1}{N}\sum_{i=1}^{N} \mid Y_i - \hat{Y}_i \mid \tag{7}$$

- **Mean Square Error**

$$Mean\ Square\ Error(MSE) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{8}$$

- **Root Mean Square Error**

$$Root\ Mean\ Square\ Error(RMSE) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \tag{9}$$

- R^2 **Score**

$$R^2 Score = 1 - \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2} \tag{10}$$

5.2 Regression Models

Among the different regressor models that will be used, the best regressor is chosen based on RMSE and MAE values as they indicate how close predicted data is to original data and the perfomance of the regressor respectively. The different regressor models include Linear Regressor, Extra Trees Regressor, Random Forest Regressor and Gradient Boosting Regressor.

5.3 Regression Perfomance

Regression Model	RMSE	MAE	R^2 Score
Linear Regressor	19.79	14.49	0.93
Extra Trees Regressor	**18.77**	**12.16**	**0.93**
Random Forest Regressor	19.12	12.21	0.93
Gradient Boosting Regressor	29.56	19.37	0.82

Table 9: Performance of the Regressors

As **Extra Trees Regressor** has the lowest RMSE value and the lowest MAE value, it can be considered as the best regressor.

6 Regression Analysis

The arrival delay time of flights ranges from 0.0 to 2142.0 minutes. Regression analysis is done to check performance of the best regressor, obtained from Table 9, on different time intervals and form an analysis based on the same.

ArrivalDelayMinutes	No Of Flights	RMSE	MAE	R2 Score
0 - 100	323550	13.15	9.89	0.64
100 - 200	48954	15.81	11.98	0.66
200 - 500	14232	20.39	14.61	0.91
500 - 1000	1128	22.72	16.34	0.97
1000 - 2142	175	49.04	29.72	0.96

Table 10: Frequency Distribution And Range Wise Regressor Scores Of The Flights

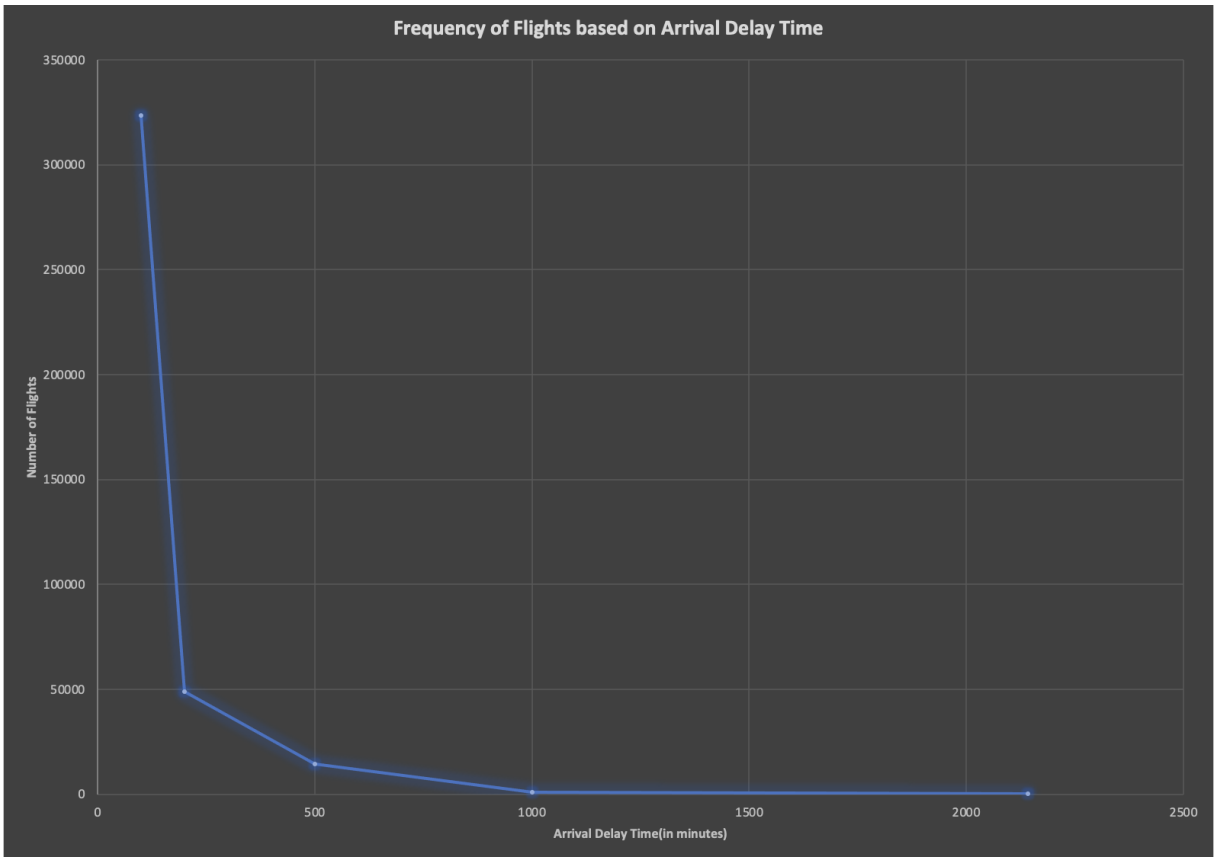

Figure 4: Frequency Distribution of Flight Delay in minutes

Fig 4 indicates the distribution of flights over different time ranges.

Low RMSE value indicates that predicted data is close to original data and low MAE value indicates the performance of the regressor. High R2 Score indicates how well the predicted data has been fit among original data.

From Table 10, it can be observed that the 200-500 range and the 500-1000 range offer low RMSE and MAE values with a high R2 Score. Although 500-100 range offers a better R2 Score with RMSE and MAE values close to the 200-500 range, its training data size is significantly lesser than the 200-500 range. As the 200-500 range offers good results after training a significantly larger dataset, it can concluded that the **best regressor perfomance** is

observed in the **200-500 range**. This indicates that the flight delay predicted around this range is more likely to be correct compared to the other ranges of time.

7 Pipeline

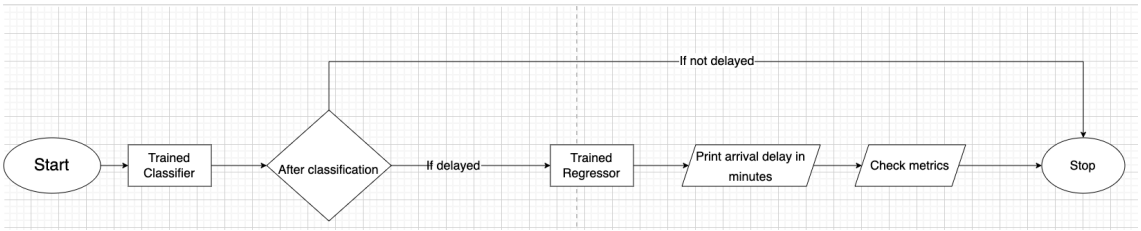

Figure 5: Pipelining Process

In pipelining, the best classifier and the best regressor are used to classify and regress the dataset respectively. First the dataset is classified using the best classifier, **Gradient Boosting Classifier using SMOTE**.

The classified dataset with only delayed flights is then regressed using the best regressor, **Extra Trees Regressor**. After regression, R2 Score, Root Mean Square Error and Mean Absolute Error are observed and noted.

Metric	Value
MAE	11.64
RMSE	16.55
R^2Score	0.95

Table 11: Performance of the Pipeline model

After observation of results in Table 11, the final stage of the project, the pipeline, has been successully completed.

8 Conclusion

Classification models were used to classify the flights as delayed or non-delayed. The classifier performance was observed, which showed the poor

performance of Class 1 with respect to Class 0. The poor performance was due to more number of non-delayed flight data points being present in the dataset. This imbalanced data was overcome by applying SMOTE. After using SMOTE, the recall values of Class 1 increased. Gradient Boost Classifier was chosen for the pipeline model as it had the highest F_1-Score.

Regression models were used to predict the arrival delay,in minutes, for the flights classified as delayed. Extra Trees Regressor was chosen for the pipeline model as it had a high R^2 Score, low RMSE and low MAE values. Regression Analysis was done to check perfomance of the best regressor in smaller intervals from the dataset based on ArrDelayMinutes.

The pipeline model with the selected classifier and regressor performed with reasonable accuracy. Thereby a two-stage predictive machine learning model has been successfully built which predicted if a flight will be delayed or not and by how much time will there be a delay if it exists.

With this information, passengers and airlines can significantly reduce losses caused by delays and also help in better planning of time.