Flight Delay Prediction

M Badri Narayanan

Abstract

A flight delay occurs when a flight takes off and(or) lands later than its scheduled time. The factors affecting the delay are airline glitches, congestion in air traffic, inclement weather such as a thunderstorm, hurricane or blizzard and security issues. This project aims to predict the arrival delay of flights using a two-stage machine learning model after its departure. If a plane is predicted as delayed, we predict the arrival delay in minutes.

1 Introduction

Flight delays affect airlines, airports, and passengers. The prediction of flight delays plays an essential role for airlines and travelers because they cause tremendous economic loss and also potential security risks. Due to flight delays, air-traffic supervision is becoming increasingly challenging. Hence there is a need to predict the delay accurately.

This project aims to model a two-stage machine learning engine to classify the flights whether as delayed or non delayed. It also aims to predict the arrival delay of flights based on the weather data of fifteen airports in the USA during 2016 and 2017 and the flight data of all flights in the USA during the same time frame. The project makes use of classification and regression models, and their performances are studied and compared.

2 Dataset

The flight dataset contains data about all the flights that flew in the USA during 2016 and 2017. The individual flight details which have their origin and destination in the 15 airports specified is used to get the Flight Dataset. The weather data was a json file, and it was restructured into csv to get the Weather Dataset. Flight and Weather Datasets were merged based on Date, Departure Time, and Departure Airport.

Table 1 shows the airport codes considered. The weather features considered are listed in Table 2, and the flight features considered are listed in Table 3.

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

Classification is the first stage of this pipeline model. Using classification models, we classify if the flights are delayed or not. Flights having target variable ArrDel15 = 1 are considered to be delayed while the flights having target variable ArrDel15 = 0 are considered to be not delayed. The final dataset consists of 18,51,433 data points out, of which 75 % of the data points were taken as training data while the remaining 25 % is were taken as testing data.

Classification Metrics

- **Accuracy**

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

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

- **Precision**

  It tells us what proportion of predicted positives are truly positive.

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

- **Recall**

  It tells us what proportion of actual positives are correctly classified.

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

- **F_1 Score**

  F_1 Score is the harmonic mean of precision and recall.

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

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.92	0.89	0.98	0.69	0.95	0.77	0.92
Gradient Boosting Classifier	0.93	0.91	0.98	0.71	0.95	0.80	0.92
ExtraTrees Classifier	0.94	0.87	0.97	0.75	0.95	0.81	0.93
Random Forest Classifier	0.93	0.92	0.98	0.74	0.96	0.82	0.93

Table 4: Classifier Performance

Table 4 shows the classifier performance before overcoming the bias.

4 Data Imbalance Problem

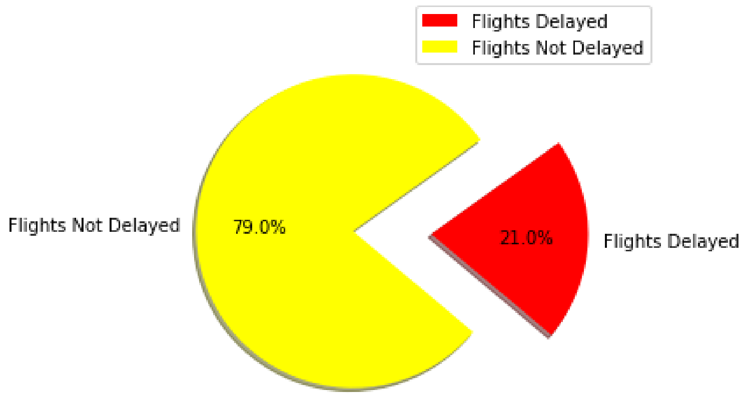


Figure 1: Dataset Distribution Before SMOTE

In the above classification algorithms, the Class 1 (delayed flights) performance is weaker than Class 0 (non delayed flights). This is due to more number of non Delayed flight data points present in the dataset, as shown in Fig 2.

This bias in the dataset can be overcome by applying Oversampling or Undersampling Techniques. Few known Oversampling and Undersampling techniques are

- **Random Over Sampler**

  Random Oversampling involves randomly duplicating data from the minority class and adding it to the training data.

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

  It is one of the few common Oversampling technique. In this technique, the new instances are generated by randomly selecting one or more of

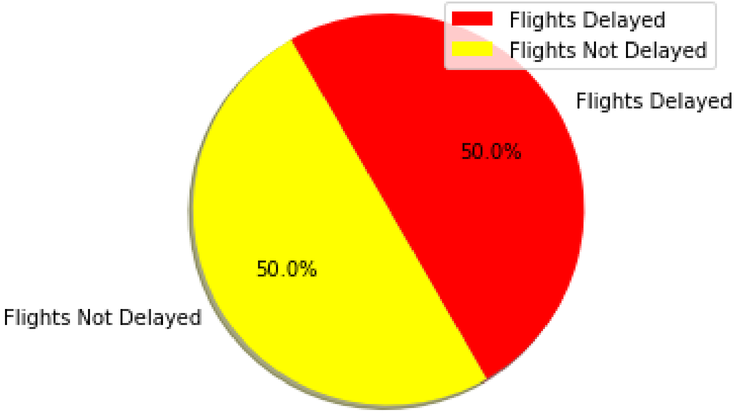


Figure 2: Dataset Distribution After SMOTE

the k-nearest neighbors for each instance in the feature space in the minority class.

- **Random Under Sampler**

  In this technique, samples from the majority class are randomly removed, to get an even distribution. This technique may discard useful or important samples.

- **NearMiss**

  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.

SMOTE technique is preferred as it creates new samples in the minority class instead of duplicating the existing samples present in the minority class. By creating new samples SMOTE technique mitigates the problem of over fitting.

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
Gradient Boosting Classifier	0.93	0.84	0.96	0.73	0.95	0.78	0.91
Extra Trees Classifier	0.94	0.83	0.96	0.76	0.95	0.80	0.92
Random Forest Classifier	0.93	0.88	0.97	0.74	0.95	0.80	0.92

Table 5: Classifier Performance After SMOTE

Table 5 shows the classifier performance after overcoming the bias.

F_1 Score is preferred in the case of uneven class distribution because it takes both false positives and false negatives into account.

The classifier with the best performance is Random Forest Classifier, as it has the highest F_1 Score. SMOTE technique improves recall values of Class 1, which is shown in Table 5.

ROC Curve

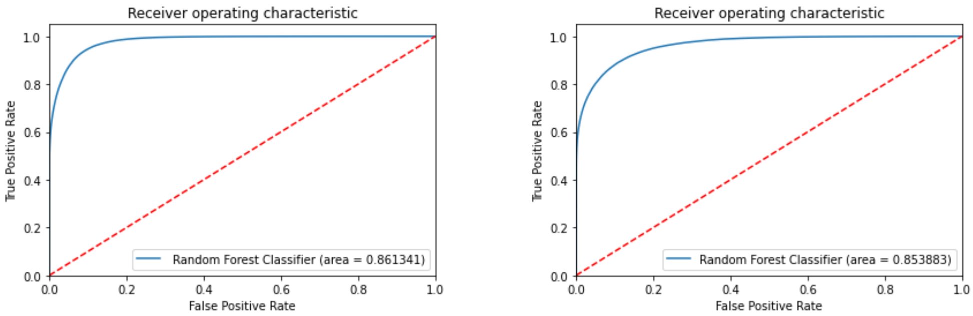
ROC Curve is used to show the classifying ability of the classifier model. AUC Score is the area under the ROC Curve. Greater the area under the ROC curve, the higher the value of the AUC score, the better the performance of the model.

$$True\ Positive\ Rate(TPR) = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate(FPR) = \frac{FP}{TN + FP}$$

Figure 3 shows the ROC Curve of Random Forest Classifier before and after overcoming the bias.



(a) Before SMOTE

(b) After SMOTE

Figure 3: ROC Curve

Classifier Model	ROC Value	
	Before SMOTE	After SMOTE
Logistic Regression	0.83	0.85
Gradient Boosting Classifier	0.84	0.84
Extra Trees Classifier	0.86	0.86
Random Forest Classifier	0.86	0.85

Table 6: ROC Values

Table 6 shows the ROC values before and after applying SMOTE.

5 Regression

Regression is the second stage of this pipeline model. Using regression models, we predict the arrival delay in minutes for the flights that have been classified as delayed by the classifier. The flights which have ArrDel15 $= 1$ are used to train the regressor.

Regression Metrics

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

The following notations stand for :
$\bar{Y}$: Mean Value Of Y

$\hat{Y}$: Predicted Value Of Y
N: Number of Data Points

- **Mean Absolute Error**

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

- **Mean Square Error**

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

- **Root Mean Square Error**

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

- R^2 **Score**

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

Regression Models

In this machine learning model, the regressors used are

- Linear Regressor
- Gradient Boosting Regressor
- Extra Trees Regressor
- Random Forest Regressor

Regressor Performance

Regression Model	RMSE	MAE	R^2 Score
Linear Regressor	17.73	12.25	0.94
Extra Trees Regressor	12.98	8.34	0.97
Random Forest Regressor	10.65	6.10	0.98
Gradient Boosting Regressor	16.07	11.07	0.95

Table 7: Performance of The Regressors

R^2 Score indicates how close the data is fitted to the regression line and higher the R^2 value better the model fits the data. Low MAE value indicates better performance of the model and lower the RMSE value better the fit. Due to the high R^2 Score, low MAE value, and low RMSE values from Table 7, the regressor with the best performance is Random Forest Regressor.

6 Regression Analysis

The arrival delay for the flights classified as delayed was between 0 to 2028 minutes. Figure 4 shows the frequency distribution of the flights. From Table 8, it is clear that most of the flights had a delay between 15 - 100 minutes. The performance of Random Forest Regressor in these ranges is given in Table 8.

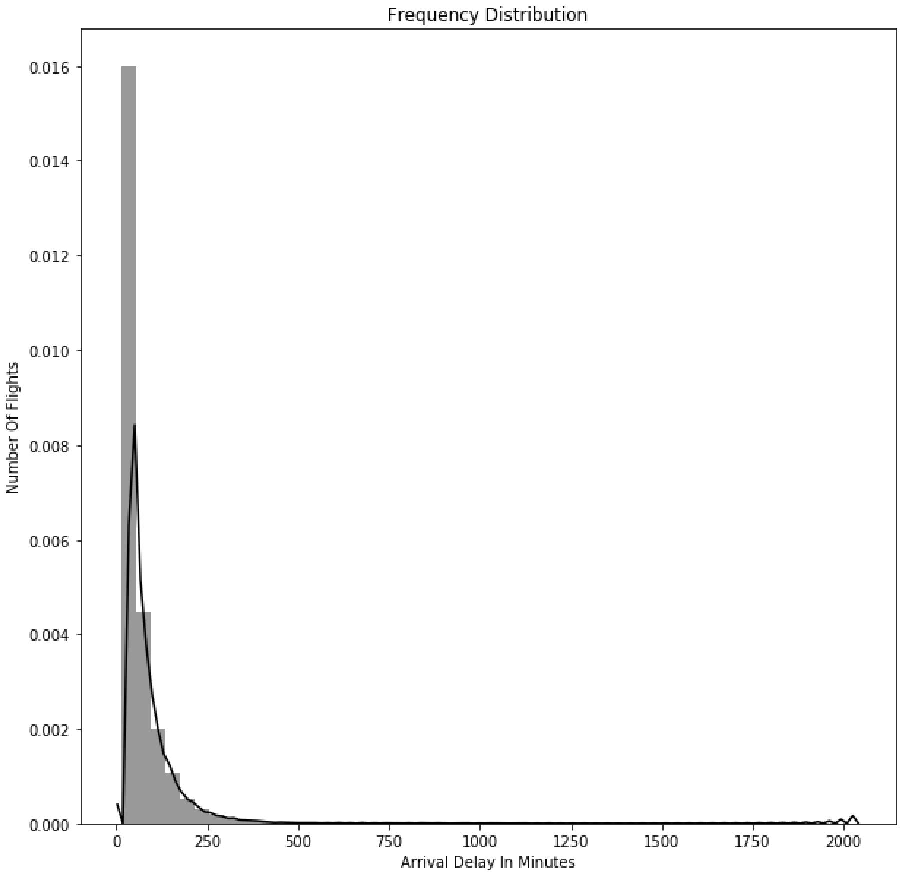


Figure 4: Frequency DistributionOf Flights

ArrivalDelayMinutes	No Of Flights	RMSE	MAE
15 - 100	81095	4.81	7.94
100 - 200	12160	11.67	17.58
200 - 500	3574	15.66	23.47
500 - 1000	290	16.39	23.07
1000 - 2000	40	29.67	41.55

Table 8: Frequency Distribution And Range Wise Regressor Scores Of The Flights

RMSE values tell us how close the predicted data is to the original data. Low MAE value indicates better performance by the regressor. From Table 8, we can say that in the range 500 - 1000, the model performance is better compared to the data in the range 1000 - 2000.

7 Pipelining

Random Forest Classifier and Random Forest Regressor were chosen to build the pipeline model. The regressor performance is shown in Table 9. Fig 5 shows the structure of the pipeline model.

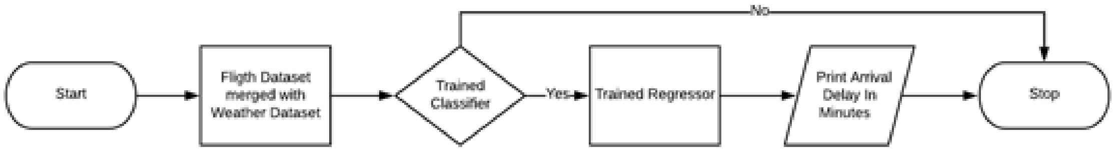

Start
Fligth Dataset merged with Weather Dataset
Trained Classifier
No
Yes
Trained Regressor
Print Arrival Delay In Minutes
Stop

Figure 5: Pipeline Model Structure

Metric	Value
MAE	6.10
MSE	113.38
RMSE	10.65
$R^2 Score$	0.98

Table 9: Performance of the pipeline model

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. Random Forest 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 those flights classified as delayed. Random Forest Regressor was chosen for the pipeline model as it had a high R^2 Score, low RMSE and low MAE values. The pipeline model with the selected classifier and regressor performed with reasonable accuracy.