Flight Delay Prediction

Nivedhitha D

Sri Sivasubramaniya Nadar College of Engineering
nivedhitha18104@cse.ssn.edu.in

Abstract. Flights are classified as delayed when they arrive later than the scheduled arrival time. This delay is predominantly influenced by environmental conditions. Flight delay is vexatious for passengers and also incurs an agonizingly high financial loss to airlines and countries. A structured prediction system is an indispensable tool that can help aviation authorities effectively alleviate flight delays. This project aims to build a two-stage machine learning engine to effectively predict the arrival delay of a flight after departure based on real-time flight and weather data.

Keywords: Machine Learning · Two-stage Model · Flight Delay Prediction.

1 Introduction

Flight delay is extremely troublesome to passengers and aviation authorities alike. Apart from the disruption of the schedule, flight delays cause monumental financial losses to the airline company. To accommodate the unforeseen delay in the arrival of a flight, a reallocation of airport resources, impromptu crew management and a redraft of flight schedules may arise. In some cases, the airline may be required to compensate the passengers for the delay.

To address this issue, this project aims to design a two-stage machine learning engine to predict the arrival delay of flights accurately. Flight delay prediction involves the pipelined operation of two sequential tasks: predicting whether a flight will be delayed or not (classification) and if the flight is delayed, to predict the arrival delay in minutes (regression). The model is trained on a dataset synthesized from 15 airports in the USA for which weather data is available and merged with the corresponding flight data from 2016 to 2017. Flights with an arrival delay greater than 15 minutes are categorized as delayed. The performance of various classification and regression models is studied and compared before constructing the pipelined engine.

Section 2 explains how the flight and weather data were processed and merged to construct the dataset. Section 3 deals with how different classifiers were trained and analyzed on the dataset. Section 4 deals with how different regressors were trained and analyzed on the dataset. Finally, Section 5 details the two-stage pipelined model to predict flight delay.

2 Dataset

The flight data contains the details of the flight schedules and their on-time performance in all the airports in the USA for the years 2016 and 2017. The weather data contains details of the atmospheric parameters that were recorded every one hour, each month over 2016 and 2017 for 15 airports (Table 1) in the USA. The airports for which the weather data is available were selected and the corresponding flight data was merged with the weather data based on the Origin, Destination, date and time attributes. The time attribute of the flight data was rounded off to the nearest hour before merging with the weather data. The features selected from the weather dataset are listed in Table 2. The features selected from the weather dataset are listed in Table 3. The raw dataset was then subject to data cleaning to handle missing data and redundant attributes. The processed data set consists of 18,51,436 data points. The dataset was split to designate 80 per cent of the data points for training and the remaining 20 per cent of the data points for testing.

Table 1. The airports for which weather data is available.

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

Table 2. The weather data attributes considered.

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

Table 3. The flight schedule and performance attributes considered.

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

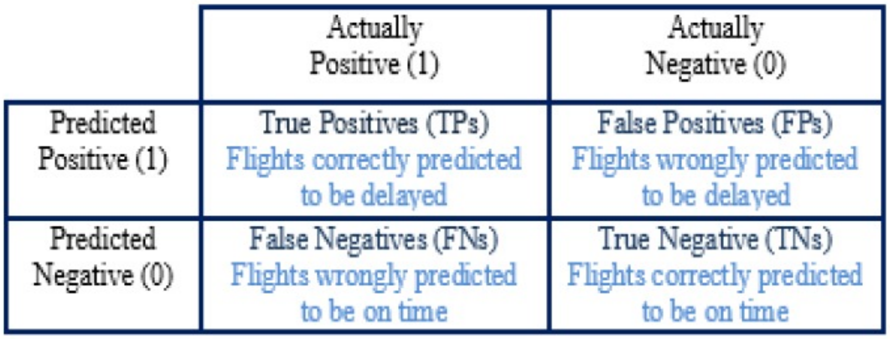
3 Classification

3.1 Overview

Classification is the first stage of the pipelined model and aims to predict whether a scheduled flight will be delayed or not. Flights that are delayed have the target variable ‘ArrDel15’ set to 1 and those which are on time have the target variable ‘ArrDel15’ set to 0. The performance of different models was studied and compared based on the performance metrics detailed in the next subsection.

3.2 Performance Metrics

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The columns in a confusion matrix represent the true values of the category and the rows represent the predicted values. Some of the important terms to be noted are explained in the confusion matrix depicted in Fig 1.



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs) Flights correctly predicted to be delayed	False Positives (FPs) Flights wrongly predicted to be delayed
Predicted Negative (0)	False Negatives (FNs) Flights wrongly predicted to be on time	True Negative (TNs) Flights correctly predicted to be on time

Fig. 1. Confusion Matrix.

From the confusion matrix, we can compute the following scores to evaluate the performance of the different classifiers.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Area under Receiver Operating Characteristic Curve is another indicator of the performance of the classifier model trained. The higher the area, better the performance of the model.

Table 4. Area Under Receiver Operating Characteristic Curve.

AUC Value	**Inference**
AUC = 0	The classifier is predicting all negatives as positives, and all positives as negatives
AUC = 0.5	The classifier is unable to distinguish the positive and negative class points thereby predicting a random or constant class for all the data points
Between 0.5 and 1	There is a high chance that the classifier will be able to distinguish between the two classes
AUC = 1	The classifier is able to perfectly distinguish between all the positive and the negative class points correctly

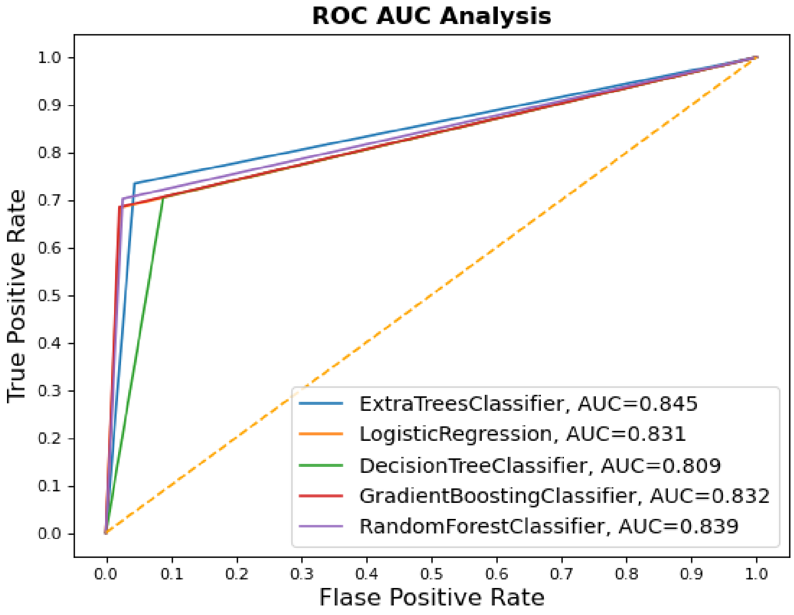


Fig. 2. Area under ROC for the different classification models.

Table 5. Results from the different classification models.

Algorithm	Precision		Recall		F1-Score		Accuracy
	0	**1**	**0**	**1**	**0**	**1**	
Logistic Regression	0.92	0.89	0.98	0.68	0.95	0.77	0.92
Decision Tree	0.92	0.68	0.91	0.71	0.92	0.69	0.87
Extra Trees	0.93	0.82	0.96	0.73	0.94	0.77	0.91
Gradient Boosting	0.92	0.90	0.88	0.68	0.95	0.78	0.92
Random Forest	0.93	0.86	0.97	0.70	0.95	0.78	0.92

3.3 Class Imbalance

It is observed that the models obtained higher recall and F1 score for the negative class (class 0) when compared to the positive class (class 1). The poor performance of the classifiers on class 1 relative to class 0 on the dataset is because of the inherent skew towards the class ‘Not-Delayed’ flights. This skew arises from the number of rows labeled ‘Not Delayed’ accounting for 79 per cent of the dataset (Fig 3). This leads to incorrect learning and misleading optimistic performance (accuracy) because the information is biased to one class. Out of 18,51,436 data points, only 3,88,058 data points are delayed flights. Since the classifier models have lesser rows with the label ‘Delayed’ to train on, the results of classification for this class are poor.

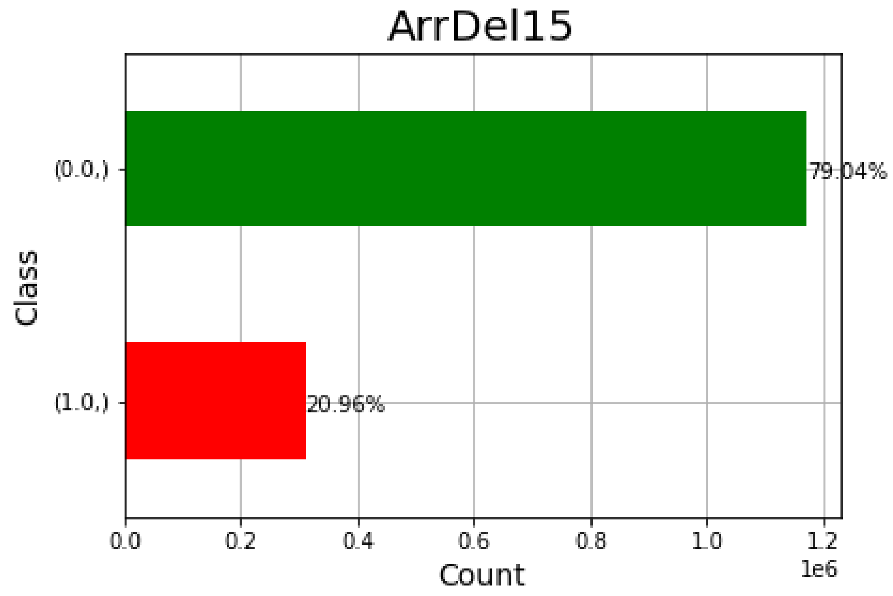


Fig. 3. Imbalanced data set class distribution.

3.4 Overcoming Imbalance

To overcome this bias, we need to perform sampling to ensure equal representation of the two classes. There are two sampling methods to make the dataset balanced:

- **Under-sampling**
  The majority class is under-sampled to ensure balance by randomly selecting examples from the majority class to delete from the training dataset repeatedly until the desired class distribution is achieved.
- **Over-sampling**
  The minority class is over-sampled to ensure balance by synthesizing data points from the minority class from the training dataset repeatedly until the desired class distribution is achieved.

To preserve the existing data, over-sampling was employed to balance the dataset. **Synthetic Minority Over-sampling TEchnique (SMOTE)** is an oversampling technique which works by selecting examples that are close in the feature space, deriving a line between the examples in the feature space and drawing a new sample at a point along that line. SMOTE was employed to sample the dataset as it synthesises data points that have smooth variation and high correlation with the existing dataset.

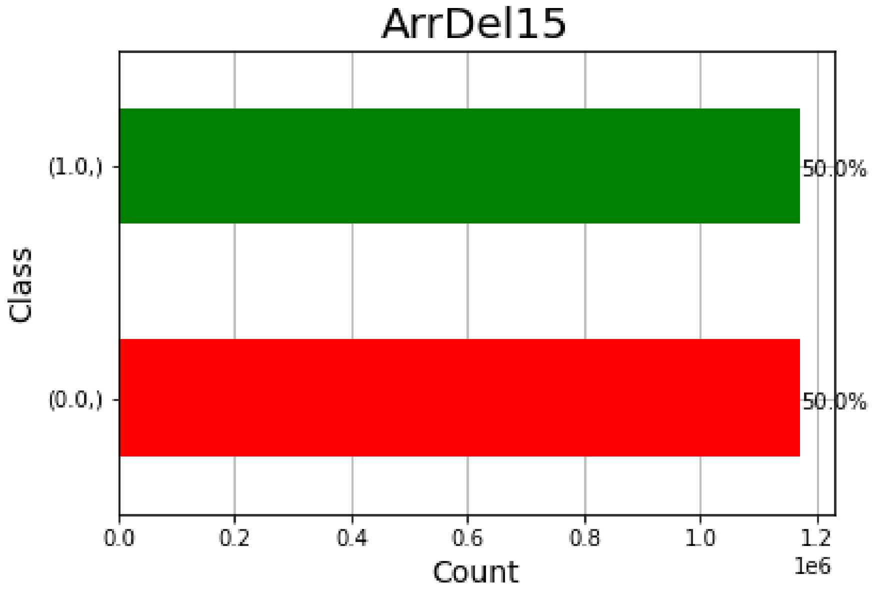


Fig. 4. Class distribution after SMOTE.

3.5 Classifier Performance Comparison after SMOTE

SMOTE improves the performance of the classification models. The Random Forest Classifier having the highest F1 Score (0.78) and Area under ROC (0.85) was chosen.

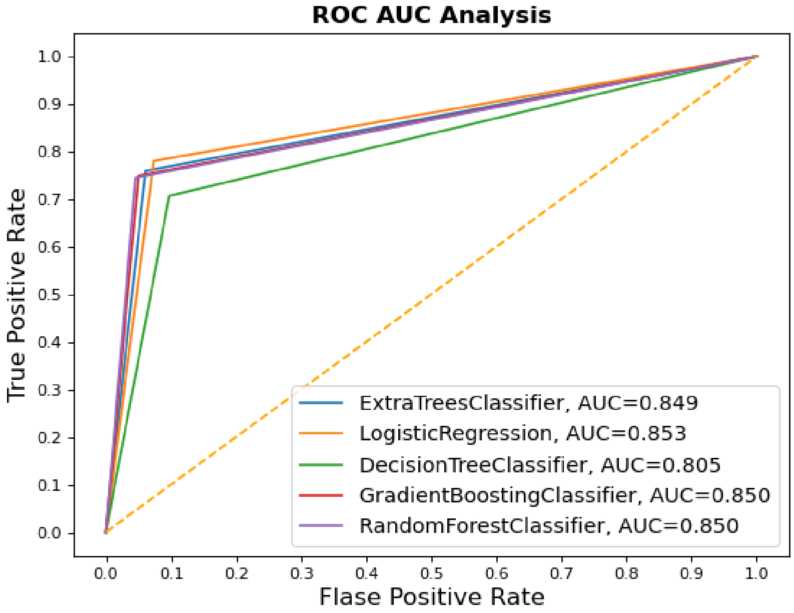


Fig. 5. Area under ROC for the different classification models after SMOTE.

Table 6. Results from the different classification models after SMOTE.

Algorithm	**Precision**		**Recall**		**F1-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	0.92	0.66	0.90	0.71	0.91	0.68	0.86
Extra Trees	0.94	0.77	0.94	0.76	0.94	0.76	0.86
Gradient Boosting	0.93	0.80	0.95	0.75	0.94	0.77	0.91
Random Forest	0.93	0.81	0.95	0.74	0.94	0.78	0.91

4 Regression

4.1 Overview

Regression is the second stage of the pipelined model and aims to predict the arrival delay in minutes if the flight is classified as ‘Delayed’ by the classifier. The flights having ‘ArrDelayMinutes’ greater than 15 were used to train the regression model. The performance of different regression models was studied and compared (Table 7).

4.2 Performance Metrics

$$Mean\ Squared\ Error(MSE) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2 \tag{5}$$

$$Root\ Mean\ Squared\ Error(RMSE) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2} \tag{6}$$

$$Mean\ Absolute\ Error(MAE) = \frac{1}{n}\sum_{i=1}^{n} \mid \hat{y_i} - y_i \mid \tag{7}$$

$$R^2\ Score = 1 - \frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{\sum_{i=1}^{n}(\hat{y_i} - \bar{y})^2} \tag{8}$$

R-squared Score is a measure of the ability of a model to predict the variances in the data set accurately. R-squared is a goodness-of-fit measure for linear regression models. The Random Forest Regressor having the most promising R-squared Score (0.937) and RMSE (15.038) was chosen.

Table 7. Results from the different regression models.

Algorithm	**RMSE**	**MAE**	R^2 **Score**
Linear Regression	15.576970	10.590204	0.933416
Decision Tree	21.758238	14.584725	0.870087
Extra Trees	15.143081	10.465352	0.937073
Gradient Boosting	15.195708	10.323951	0.936635
Random Forest	15.038856	10.384938	0.937937

4.3 Regression Testing

In this section, the dataset is split into ranges of arrival delay minutes and the performance of the pipelined Random Forest Regressor is studied in each range. The flight arrival delay ranges from 0 to 1210 minutes. The frequency distribution plot of the arrival delay indicates that the majority of data points are observed in the 0 - 200 range. The bar plot reveals that most of the data points have ‘ArrDelayMinutes’ ranging between 0 - 100 minutes. As the range increases, the number of data points decreases, indicating that flights with very high flight delays are less. As the number of data points decrease with each range, the values of RMSE and MAE scores increase excluding the 1000-1210 range. This is attributed to the imbalance in the dataset.

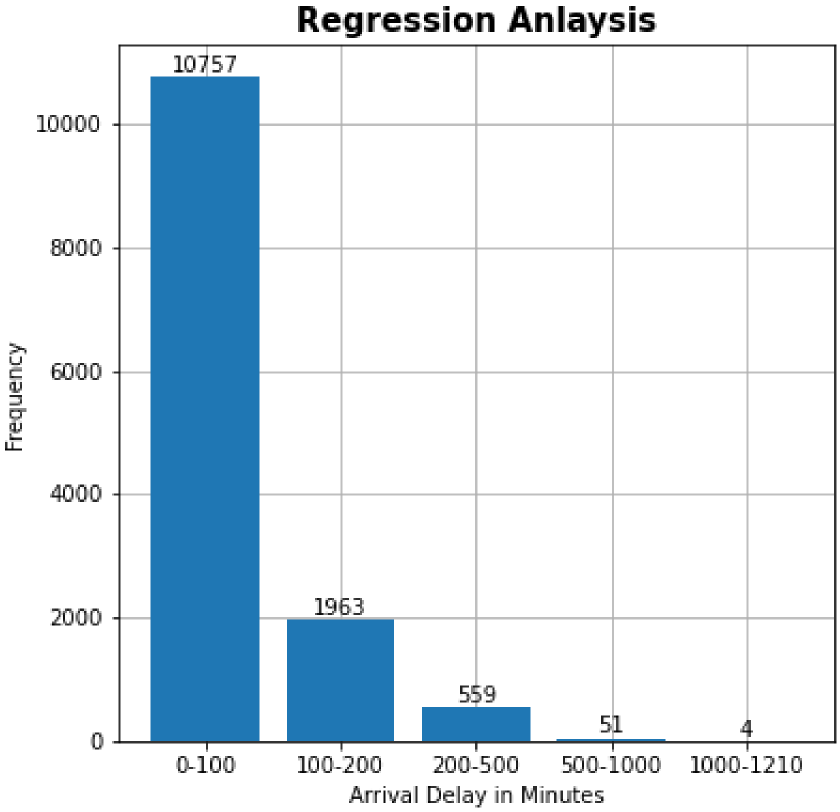


Fig. 6. Range-wise frequency analysis.

Table 8. Range-wise regression analysis.

Range	**RMSE**	**MAE**	R^2 **Score**
0-100	9.834668	6.634876	0.856493
100-200	15.399585	9.194358	0.658591
200-500	18.425296	10.801726	0.914323
500-1000	20.275852	11.399412	0.982030
1000-1210	5.276057	4.457500	0.995092

5 Pipelined Model

The flow chart represents the two-stage flight delay prediction machine learning model. The pipelined model involves chaining the best performing classifier before the best regressor. The data was preprocessed and a model was trained to perform classification using the Random Forest Classifier. The Random Forest Classifier was chosen as it has the maximum F1 Score (0.78) and area under ROC (0.85). The flight delay needs to be calculated only for the flights that will be delayed. Thus, only those data points that were predicted to be delayed by the classifier are selected to perform regression and predict the flight arrival delay in minutes. The Random Forest Regressor was chosen as it has the highest R-squared score (0.937) and lower values of RMSE (15.038) and MAE (10.384).

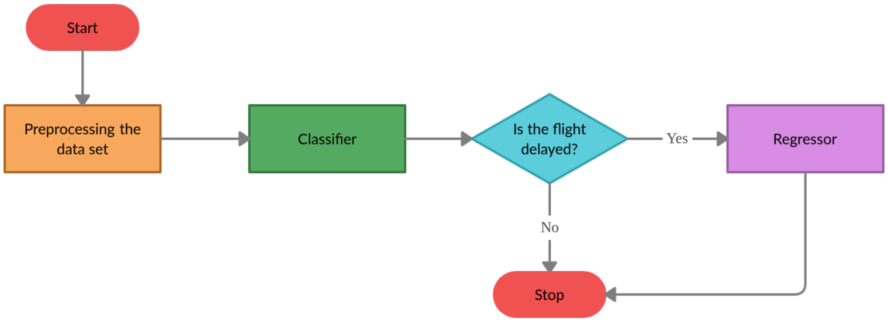


Fig. 7. Flight delay prediction as a pipelined operation of two sequential tasks: predicting whether a flight will be delayed or not (classification) and if the flight is delayed, to predict the arrival delay in minutes (regression).

Table 9. Pipelined model performance evaluation.

Metric	**Value**
RMSE	11.283234
MAE	7.178468
R^2 Score	0.977392

6 Conclusion

The flight and weather data were combined into a single dataset and preprocessed to train a two-stage machine learning model that predicts flight arrival delay. Due to class imbalance, there was an inherent bias towards the majority class, 'Not Delayed' flights (class 0). The data was sampled using SMOTE before classification to overcome the bias. Out of several classification algorithms, the Random Forest classifier gave the best F1 score (0.78) and Recall (0.74) for the delayed flights. Subsequently, the Random Forest regressor was pipelined, giving MAE 7.178 minutes and RMSE 11.283 minutes with an R-squared score of 0.977. In conclusion, the flight delay prediction was efficient and the Machine Learning model exhibited good performance.