Flight Delay Prediction

Charu Jain

SSN College of Engineering, Chennai

Abstract. Flight delay is a very crucial and important problem for almost all countries in recent times. A structured prediction system is an indispensable tool that can help aviation authorities to effectively alleviate flight delays. For the same reason, a two-stage machine learning engine is implemented to predict the delay of flights based on real-time flight and weather data. It consists of a classifier which helps in the prediction the delayed flights, the output of which is passed to the regressor that predicts the amount of delay. In totality, this model aids in the prediction of flight delay in a systematic way once the flight has taken off.

Keywords: Flight delay, Machine Learning, Predictive analysis.

1 Introduction

Flight delay is a huge problem which can cause disturbances to both passengers and airlines due to which the daily itinerary of both the parties gets disrupted. This will have a negative affect on the airlines' economy and it is equally essential to inform the passengers about the flight delay. It therefore, becomes necessary for the commercial airlines to predict flight delay efficaciously. In order to address this problem, a two-stage pipeline using machine learning models is implemented.

The dataset used consisted of real-time flight and weather data of airports in USA. This data was then pre-processed, merged and fed to different classifiers through which, statistics about the delayed flights was procured. Further on, regression algorithms were implemented to obtain the amount of delay in the flights. The metrics of classifiers and regressors were then analysed and the most efficient algorithm among them was adopted. Subsequently, the classifier and the regressor chosen was implemented successively in pipelining. Promising results were obtained through this two-stage machine learning model.

2 Data Pre-processing

The first dataset used, consisted of the performance of the flights from specific airports in USA out of which only 15 airports were extracted. Table 1. shows the airport codes of the 15 airports afore mentioned. From these 15 airports the attributes pertaining to every trip are shown in Table 2. The second dataset consisted of weather data from which the weather data of the corresponding 15 airports in USA for the years 2016 and 2017, were extracted. Table 3. shows the different weather attributes which were considered.

Pre-processing of the flight and weather data was done to alter the raw data to a format which was understandable by the machine learning models. Irrelevant data was ignored which could have hampered the performance of the machine learning techniques. After appropriate extraction of the given attributes, these datasets were merged such that each trip's airports had its corresponding weather data. This was done based on the location of the origin and the destination airport, the date and also the time.

The target variable for classification, "ArrDel15" holds the value of 0 for flights that have arrived on-time and the value of 1 for flights that arrived late. Therefore "ArrDel15" can be considered a categorical variable with its categories/classes being 0 (on-time flights) and 1 (delayed flights). Upon further inspection it was observed that there is a significant class imbalance between these two classes with class 0 having majority occurrence when compared to class 1. The pie chart in Fig. 1. shows the class imbalance in the dataset.

Table 1. Airports considered

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

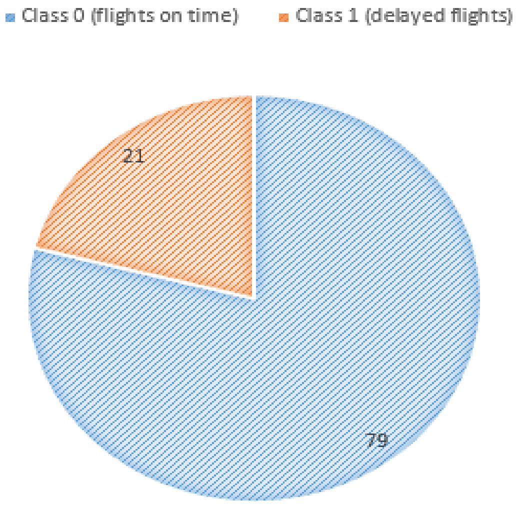


Fig. 1. Pie chart representing class imbalance

Table 2. Flight attributes considered

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

Table 3. Weather attributes considered

WindSpeedKmph	WindDirDegree	WeatherCode
Visibilty	Pressure	Cloudcove
WindGustKmph	tempF	WindChillF
precipMM	Humidity	DewPointF
date	time	airport

3 Classification

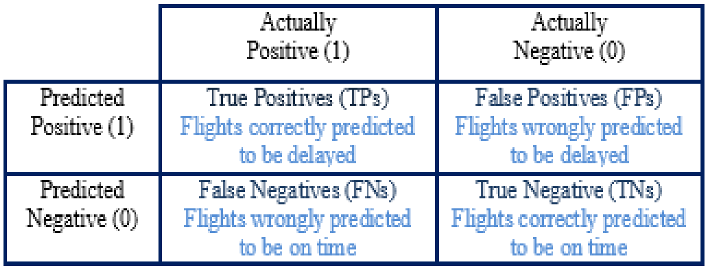
Classification is a process of categorizing a given set of data into classes. In this case, the classifiers were implemented to predict whether a flight will be delayed or not by making use of the pre-processed data. All attributes were considered except for "ArrDel15" and "ArrDelayMinutes" where "ArrDel15" was the target variable. Seventy percent of the dataset was used to train the classifiers and the remaining thirty percent was used to test it.

The different classifiers tested were Logistic Regression, Decision Trees, Random Forest, Extra Trees and XGBoost.

3.1 Metrics in Classification

Certain metrics that are used for the purpose of evaluating the classifiers are mentioned in this section. Some of the important terms to be noted are explained in the confusion matrix depicted in Fig 2. 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 of the same.

Therefore, we can say that, in this case the four outcomes that can be obtained after classification will be:



	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. 2. Confusion matrix

1. Precision: It is calculated as the number of true positives divided by the total number of true positives and false positives.
   Precision is simply the ratio of correct positive predictions out of all positive predictions made.
   $$Precision = \frac{TP}{TP + FP} \tag{1}$$
2. Recall: It is calculated as the number of true positives divided by the total number of true positives and false negatives.
   Recall is the ratio of correctly predicted positive observations to all the observations in the actual class.
   $$Recall = \frac{TP}{TP + FN} \tag{2}$$
3. F1 Score: It is a weighted harmonic mean of precision and recall.
   $$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$

Table 4. Classification results - Unsampled Data

Algorithm	Category	Precision	Recall	F1-score
Logistic Regression	0.0	0.92	0.98	0.95
	1.0	0.89	0.68	0.77
Decision Trees	0.0	0.92	0.91	0.92
	1.0	0.68	0.70	0.69
Random Forest	0.0	0.92	0.98	0.95
	1.0	0.89	0.70	0.78
Extra Trees	0.0	0.93	0.96	0.94
	1.0	0.81	0.74	0.77
XGBoost	0.0	0.92	0.98	0.95
	1.0	0.90	0.69	0.78

Table 4. shows the performance of each classifier which was implemented. Here, Recall takes into account, the flights that are wrongly predicted to be on time and since we are focusing on delayed flights, recall scores are important. Similarly, as F1 score is the harmonic mean of Precision and Recall, both F1 score and Recall values were observed to evaluate the classifiers. It can be observed that the models were better at predicting the recall and F1 scores of the negative class (class 0) when compared to the scores of the positive class (class 1). This difference is seen because classes were not represented equally, owing to which the machine learning models gave misleadingly optimistic performance. To overcome this, the data was sampled prior to classification.

3.2 Sampling

Class imbalance in the data was seen through a pie-chart in the section 2 where the two categories of classes, class 0 (on-time flights) and class 1(delayed flights) had a vast difference. Therefore, sampling was performed, so that the training dataset could be transformed, in order to balance the class distribution.

Here, the two sampling methods used were SMOTE and Random Undersampling.

1. SMOTE: Synthetic Minority Oversampling Technique is used to address imbalanced datasets by oversampling the minority class (delayed flights). SMOTE synthesises new minority instances only between the real minority instances.
2. Random Undersampling: It involves randomly selecting examples from the majority class(on-time flights) to delete from the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.

Table 5. and Table 6. show the metric scores of the sampled data.

Table 5. Classification results - SMOTE

Algorithm	Category	Precision	Recall	F1-score
Logistic Regression	0.0	0.94	0.93	0.93
	1.0	0.74	0.78	0.76
Decision Trees	0.0	0.92	0.90	0.91
	1.0	0.66	0.70	0.68
Random Forest	0.0	0.93	0.96	0.95
	1.0	0.84	0.74	0.78
Extra Trees	0.0	0.94	0.94	0.94
	1.0	0.78	0.76	0.77
XGBoost	0.0	0.93	0.97	0.95
	1.0	0.87	0.71	0.78

Table 6. Classification results - Random Undersampling

Algorithm	Category	Precision	Recall	F1-score
Logistic Regression	0.0	0.94	0.93	0.93
	1.0	0.74	0.78	0.76
Decision Trees	0.0	0.94	0.94	0.94
	1.0	0.78	0.76	0.77
Random Forest	0.0	0.95	0.91	0.93
	1.0	0.71	0.81	0.76
Extra Trees	0.0	0.95	0.90	0.92
	1.0	0.68	0.82	0.74
XGBoost	0.0	0.95	0.92	0.93
	1.0	0.73	0.80	0.76

3.3 Inference

It is known from Section 3.1 that evaluation of Recall and F1 scores is important. Upon observing the outcomes of the five classifiers, the F1 score and Recall of the Random Forest classifier was found to be leading. This can be seen from the results obtained after Oversampling (Table 4.). Hence, it was concluded that Random Forest performs better than the rest of the classification algorithms employed.

4 Regression

Regression allows to predict a continuous outcome variable based on the value of one or multiple predictor variables. In this section, the outcome variable was ”ArrDelayMinutes” and the amount of delay in minutes of the delayed flights was predicted. The regression algorithms used were Linear regressor, Random Forest, Extra Trees and XGBoost Regressor. Table 7. shows the metrics obtained for all the regressors.

4.1 Regression Metrics

Here,
x_i - predicted value of y;
y_i - true value of y;
$\overline{y}$ - mean value of y;
n - number of data points

1. Mean Absolute Error(MAE): It is the absolute difference between the target value and the value predicted by the model. MAE is a linear score which means all the individual differences are weighted equally.

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{4}$$

2. Root Mean Squared Error(RMSE): It is the square root of the averaged squared difference between the target value and the value predicted by the model.

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2} \tag{5}$$

3. R^2Error: It helps to correlate the predictions with the ground truth and tells how much the current model agrees with it. R^2 is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R^2 will always be less than or equal to 1.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Table 7. Regression metrics

Algorithm	MAE	RMSE	R^2
Linear Regression	12.203	17.648	0.940
Random Forest	11.668	16.647	0.947
Extra Trees	11.720	16.700	0.946
XGBoost	11.140	16.141	0.950

5 Pipelining

In this segment, classification and regression was implemented consecutively such that the best performing classifier is used prior to using the best regressor. The XGBoost regressor was used, as it has least MAE, RMSE and highest R^2. Here, the output of the Random Forest classifier was fed to the XGBoost regressor which then predicted the amount of the flight delay in minutes. This means that the flight delay was calculated only for the flights predicted to be delayed by the classifier. Table 8. shows the metrics that were obtained after Pipelining.

Table 8. Final scores after pipelining

MAE	RMSE	R^2
12.959	17.605	0.950

6 Regression Analysis

In this section, the delay in minutes incurred by flights were categorised into different sets on which regression was performed. This was done using XGBoost Regressor to analyse the performance of prediction of the amount of delay for particular ranges. It was concluded from looking at the class 1000-1650 minutes in Table 9. that the model had performed well. The reason is that, the error here is negligible when compared to the magnitude of the bin.

Table 9. Regression performance for different classes of delay in minutes

Category	MAE	RMSE
15-100	10.344	13.523
100-200	15.860	23.346
200-500	17.729	27.648
500-1000	16.448	23.663
1000-1650	22.859	32.571

7 Conclusion

To predict flight delay using the weather and flight data in US, the required data was first pre-processed. Due to class imbalance, the data was sampled using SMOTE before classification. Out of the five different algorithms, Random Forest classifier gave the F1 score of 0.78 and Recall value of 0.74 for the delayed flights which turned out to be the most efficient. Subsequently, the Random Forest classifier and the XGBoost regressor were implemented consecutively giving MAE as 12.96 minutes and RMSE as 17.6 minutes. Thereby, the flight delay prediction was efficient due to the incorporation of Machine Learning techniques.