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. High accuracy 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. One of the main reasons why this is essential for airlines is that, the customers' choice can be influenced by the delay caused, which will in turn have a negative affect on the airlines' economy. It is equally essential to inform the passengers about the delay in the flights beforehand. Therefore, the proposed method uses Machine Learning techniques to prevent this and predict flight delay efficiently.

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 or schedule of both the parties gets disrupted. It therefore becomes necessary for the commercial airlines to predict flight delay efficaciously. In order to address this problem, a machine learning model employing on-time flight data combined with weather data is implemented.

The dataset used consists of real-time data of 15 airports in USA, most of which include features which can potentially cause delay in flights. This data is then pre-processed, merged and fed to five different Machine Learning classifiers. Through classification, the information about the delayed flights is procured. Further on, regression algorithms are utilised to obtain the amount of delay in the flights. The metrics of classifiers and regressors are then analysed and compared after which, the most efficient algorithm among them is adopted. On this wise, the final metrics obtained after implementing the model clearly shows that the flight delay prediction is highly efficient.

2 Data Pre-processing

The dataset used, consisted of the on-time performance of the flights from specific airports in USA out of which only 15 airports were extracted for further manipulation. Table 1. shows the airport codes of the 15 said airports. From these 15 airports the flight features that were used for the prediction in flight delay are given in Table 2. The weather data of the corresponding 15 airports in USA for the years 2016 and 2017 only, were extracted from the entire weather data available. Table 3. shows the different weather features which were considered.

Data Pre-processing of the flight and weather data was done to alter the raw or unreadable data to a useful and efficient format. From this data, irrelevant data was ignored which could have hampered the performance of the Machine Learning techniques. This is called Data Cleaning, which is the first step of Data Pre-processing. Following which, Data Transformation was done, to bring the data in a form that will aid in further implementation. After appropriate extraction of the features that could have influenced the flight delay from both flight and weather data, these datasets were merged such that each flight 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 recorded while observing the corresponding weather data. It was concluded after looking at the dataset that there was a class imbalance that existed. It was observed that the delayed flights were less in number than the flights that were on time. The pie chart in Fig. 1. shows the class imbalance in the dataset.

In this way, the flight and weather data were pre-processed with the required features.

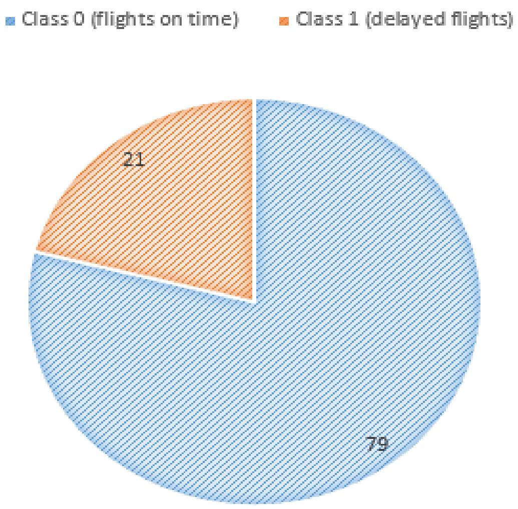

Fig. 1. Pie chart representing class imbalance

Table 1. Airports considered

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

Table 2. Flight features considered

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

3 Classification

In this section, classifiers were modelled using different classification algorithms to predict whether a flight will be delayed or not by making use of the preprocessed data. A train-test split of 70:30 was done on the entire dataset ahead of classification. In this process, the classes of given data points were predicted. The classes that have to be predicted are also called as targets/ labels or categories.

3.1 Metrics in Classification

[Table 4. explains the variables used in the following metrics]

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, or the accuracy of minority class predictions.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

2. Recall: It is calculated as the number of true positives divided by the total number of true positives and false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Table 3. Weather features considered

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

Table 4. Variables used in Classification Metrics

TruePositive (TP)	correctly predicted as delayed flights
FalsePositive (FP)	wrongly predicted as delayed flights
TrueNegative (TN)	correctly predicted as on-time flights
FalseNegative (FN)	wrongly predicted as on-time flights

3. F1 Score: It is a weighted harmonic mean of precision and recall. The weighted average of F1 should be used to compare classifier models and not global accuracy.
$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$
4. Support: It is the number of predicted occurrences of the class in the specified dataset.

3.2 Classification algorithms

The process of classification was carried out by making use of different classification algorithms. This was done to analyse the performance of each algorithm and to study the metric values obtained to decide on the best one out of all of the ones used. Table 4. shows the performance of each classifier which was implemented.

Linear Regression is a Machine Learning algorithm predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. Classification tasks have discrete categories, unlike regressions tasks.

Decision Trees is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Random Forest uses the principle that a forest is made up of trees and more trees means more robust forest. This algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

Extra Trees aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. When it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class.

Table 5. Classification results - Imbalanced Data

Algorithm	Category	Precision	Recall	F1-score	Support
Linear Regression	0.0	0.92	0.98	0.95	438945
	1.0	0.89	0.68	0.77	116384
Decision Trees	0.0	0.92	0.91	0.92	438945
	1.0	0.68	0.70	0.69	116384
Random Forest	0.0	0.92	0.98	0.95	439076
	1.0	0.89	0.70	0.78	116253
Extra Trees	0.0	0.93	0.96	0.94	439076
	1.0	0.81	0.74	0.77	116253
XGBoost	0.0	0.92	0.98	0.95	439076
	1.0	0.90	0.69	0.78	116253

3.3 Sampling

Imbalanced Data Class imbalance in the data was seen through a pie-chart in the pre-processing section where the two categories of classes, class 0 and class 1 had a vast difference. This typically refers to a problem in classification where the classes are not represented equally owing to which machine learning techniques often fail or give misleadingly optimistic performance on classification datasets with an imbalanced class distribution. Data sampling provides a collection of techniques that transform a training dataset in order to balance or better balance the class distribution.

Here, the minority class is class 1 which represents the delayed flights and the majority class is class 0 upon which two sampling methods are used: SMOTE Oversampling and Random Undersampling:

1. SMOTE Oversampling: Synthetic Minority Oversampling Technique is used to address imbalanced datasets is to oversample the minority class (delayed flights). It involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples.
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.

3.4 Analysis

Upon observing the metrics of the five classifiers, the F1 score and Recall of the Random Forest classifier was the found to be leading. This can be seen from Tables 5, 6 and 7. Hence, it was concluded that Random Forest performs better than the rest of the classification algorithms employed.

Table 6. Classification results - SMOTE Oversampling

Algorithm	Category	Precision	Recall	F1-score	Support
Linear Regression	0.0	0.94	0.93	0.93	439076
	1.0	0.74	0.78	0.76	116253
Decision Trees	0.0	0.92	0.90	0.91	439076
	1.0	0.66	0.70	0.68	116253
Random Forest	0.0	0.93	0.96	0.95	439076
	1.0	0.84	0.74	0.78	116253
Extra Trees	0.0	0.94	0.94	0.94	439076
	1.0	0.78	0.76	0.77	116253
XGBoost	0.0	0.93	0.97	0.95	439076
	1.0	0.87	0.71	0.78	116253

Table 7. Classification results - Random Undersampling

Algorithm	Category	Precision	Recall	F1-score	Support
Linear Regression	0.0	0.94	0.93	0.93	439076
	1.0	0.74	0.78	0.76	1162536
Decision Trees	0.0	0.94	0.94	0.94	439076
	1.0	0.78	0.76	0.77	116253
Random Forest	0.0	0.95	0.91	0.93	439076
	1.0	0.71	0.81	0.76	116253
Extra Trees	0.0	0.95	0.90	0.92	439076
	1.0	0.68	0.82	0.74	116253
XGBoost	0.0	0.95	0.92	0.93	439076
	1.0	0.73	0.80	0.76	116253

4 Regression

In this step, 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. In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable, called the 'outcome variable' and one or more independent variables called features. Here, the dependent variable is the flight arrival delay in minutes and the dataset includes the independent variables which can potentially affect the delay.

Table 7. shows the metric scores of each of the given regressors.

4.1 Regression Metrics

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. Mean Squared Error(MSE): It is simply the average of the squared difference between the target value and the value predicted by the regression model. As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is.

$$MSE = \frac{1}{n} \sum_{t=1}^{n} e_t^2 \tag{5}$$

3. 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. It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.

$$RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - x_i)^2} \tag{6}$$

4. R^2Error: It helps to correlate the current model with a constant baseline and tells how much the current model is better. The constant baseline is chosen by taking the mean of the data and drawing a line at the mean. 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.

4.2 Analysis

After completing regression and analysing the metric scores of each regressor, it was concluded that XGBoost was the best algorithm for prediction of the amount of flight delay. This inference was drawn as it had the lowest mean absolute error among the regressors.

Table 8. Regression performance

Algorithm	MAE	MSE	RMSE	R2
Linear Regression	12.203	311.472	17.648	0.940
Random Forest	11.668	277.124	16.647	0.947
Extra Trees	11.720	278.903	16.700	0.946
XGBoost	11.140	260.554	16.141	0.950

5 Pipelining

It is performing classification and regression consecutively such that the best performing classifier is used prior to using the best regressor. 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 9. shows the metrics that were obtained after Pipelining.

Table 9. Final scores using XGBoost Algorithm

MAE	MSE	RMSE	R2
12.959	309.967	17.605	0.950

5.1 Regression Analysis

In this section, the delay in minutes incurred by flights were classified into different sets on which regression was performed. This was done using XGBoost Regressor to analyse the correctness in the prediction of the amount of delay for particular ranges of delay. It was concluded from the scores shown in Table 10. that the class 1000-1560 minutes is the most correctly predicted out of all the other classes. The reason is that, even though its error is more when compared to other classes, the error will not affect this class as much as it affects the rest, as the delay here is maximum.

6 Conclusion

To predict flight delay using the weather data and on-time performance of flights in US, the required data was first pre-processed. On the same data, classification was done using five different algorithms out of which Random Forest classifier gave the f1 score of 0.78 and Recall value of 0.70 for the delayed flights which turned out to be the most efficient of out all. Similarly in regression, XGBoost gave best results having only MAE as 11.14 minutes and RMSE as 16.14 minutes.

Table 10. Regression performance for different classes of delay in minutes

Class	MAE	MSE	RMSE	R2
15-100	10.344	182.892	13.523	0.640
100-200	15.860	545.048	23.346	0.259
200-500	17.729	764.412	27.648	0.825
500-1000	16.448	559.947	23.663	0.970
1000-1650	22.859	1060.912	32.571	0.940

Subsequently, the Random Forest classifier and the XGBoost regressor were implemented consecutively giving MAE as 12.96 minutes and RMSE as 17.6 minutes. Class analysis which was implemented after pipelining, showed that the class 1000-1560 minutes is the most efficiently predicted. Thereby, the flight delay prediction was quite efficient and satisfactory by incorporating Machine Learning techniques.