Flight Delay Prediction

Ashwini M

Abstract: Flight delay incur huge losses in the aviation industry, hence there is a need to optimize flight operation. The on-time performance of flights is crucial in the decision making process of the aviation industry. Hence, in this project, a two stage predictive machine learning engine that forecasts the on-time performance of commercial flights by predicting the delay is developed. Flight information of US Domestic flights along with weather data of various airports across USA are analyzed and processed. Using classifiers, they are classified as delayed or not to calculate the delay in minutes for the delayed flights using regressors.

1 Introduction

The advent of modern globalization and industrialization in the 19th century catalyzed revolutionary growth in many fields, notably in the Aviation Industry. This resulted in increased air travel – both commercial, including government and private airlines. However, one of the significant downside of this growth is the increasing difficulty in air traffic supervision, which is further aided by extreme weather, faulty airport operations, etc. This leads to flight delays, causing large economic and environmental losses. In the US, FAA considers a flight to be delayed when difference between scheduled and actual arrival time is greater than 15 minutes. The losses incurred due to flight delays for domestic flights in the US is estimated to be in billions of dollars.

This calls for the urgent need to optimize flight operations and minimize losses by predicting flight delay during arrival. Against this background, machine learning models were developed after processing data sets for the years 2016 and 2017. Several classification techniques (Logistic Regression, Decision Tree Classification, Extra Trees Classification, Random Forest Classification, XGBoost Classification) were tested to classify if each flight was delayed during arrival or not. Following this, the delay in minutes (if any) was predicted using Regression techniques (Linear Regression, Extra Tree Regression, XGBoost Regression, Random Forest Regression) for every flight that was classified as delayed. The developed models were then compared with each other to obtain the best performing model.

2 Dataset

The first input dataset was the flight data, which contained records of flight information, specifically ArrDel15 – a binary valued column assuming either 1, indicating Arrival Delay or 0, indicating no Arrival Delay of various stations for the years 2016 and 2017. This was pre-processed to extract only those flights which were within selected airports in the United States, along with their respective flight feature columns.

Table 1. Airports codes

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

Table 2. Flight Columns

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

The other input dataset was weather data of various airports in United States. The data was available as a collection of JSON files, each file holding hourly weather information, month-wise for the years 2016 and 2017. Since it expressed external and environmental conditions for a particular moment, feature selection was very important to maximize robustness of prediction models. Relevant weather features considered for this model is mentioned in Table 3.

Table 3. Weather Columns

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

After relevant information was extracted from Flight Data and Weather Data, to have the corresponding records together, the two sets were merged to give a final compact dataset such that each record of the flight had the corresponding weather data available. The two sets were merged based on the columns of CRSDepTime, FlightDate and Origin. The final set had 1851433 records.

3 Classification

In order to predict the delay in minutes, the flights were first categorised as delayed or not by deploying classification models that categorized given set of data into classes. This was done since the delay was calculated only for those flights classified as delayed. To do this, the pre-processed dataset was split into training and validation sets in the ratio 80:20. Keeping ArrDel15 as the ground truth value, classification models were trained to predict a flight as delayed or not (0 for not delayed and 1 for delayed). The classifiers used the feature columns in Table 4 to make the predictions. The different classifiers used included Logistic Regression, Decision Tree Classification, Extra Trees Classification, and XGBoost Classification.

Table 4. Features used by Classifiers

DepDelayMinutes	CRSArrTime	CRSDepTime	DepDel15	windspeedKmph	winddirDegree
weatherCode	precipMM	visibility	pressure	cloudcover	DewPointF
WindGustKmph	tempF	WindChillF	humidity	airport	

3.1 Classification Metrics

Evaluation of performance of a Classification model was done via Accuracy, Precision, Recall and F1 Score metrics which give a clear idea on the robustness of a model, the basis of which are four parameters – True Positives (TP), True Negatives (TN), False Positives (FN) and False Negatives (FN) which can be understood with the help of a confusion matrix illustrated in Table 5

Table 5. Confusion Matrix

		Prediction outcome	
		N (0)	**P (1)**
Actual value	**N (0)**	True Negative	False Positive
	P (1)	False Negative	True Positive

The four main classification metrics used are –

- Accuracy – Accuracy is the ratio of correctly predicted observation to the total observations. This metric is ideally suited for balanced datasets.

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

- Precision – Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate as it estimates number of positive class predictions that actually belong to the positive class.

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

- Recall (Sensitivity) – Recall is the ratio of correctly predicted positive observations to the all observations in actual class, giving a clear idea on proportion of actual positives identified correctly.

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

- F1 score – F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$F1Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

The summary of the classification algorithms using the above metrics is tabulated in Table 6 to measure the quality of predictions.

Table 6. Classification Results

Classifier	**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 Classification	0.92	0.72	0.93	0.70	0.93	0.71	0.88
Extra Tree Classification	0.93	0.80	0.95	0.73	0.94	0.76	0.91
Random Forest Classification	0.92	0.83	0.96	0.70	0.94	0.76	0.91
XGBoost Classification	0.92	0.90	0.98	0.69	0.95	0.78	0.92

4 Data Balancing

The input dataset used was imbalanced – a situation where the number of observations belonging to one class is significantly lower than those belonging to the other class, resulting in models having poor predictive performance. Imbalanced datasets lead to biased classifications which pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of data for each class. Fig. 1 visually illustrates the imbalance in this dataset with the help of a pie chart.

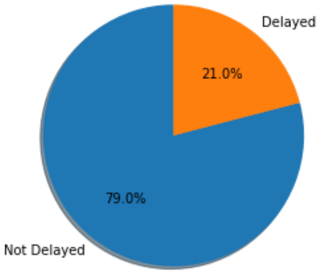

Fig. 1. Imbalance in distribution of delays of flight

The technical approach to solve data imbalance is to either increase the frequency of the minority class or decrease the frequency of the majority class, the entire process known as Sampling.

Sampling Techniques:

- Random Over Sampling – Randomly duplicating minority class examples.
- Random Under Sampling – Randomly eliminating majority class examples.

Here, SMOTE – an oversampling technique – was used to oversample the minority class and resulted in an equal distribution between the two classes, the metrics of which are summarized in Table 7 for all the classifiers.

Table 7. Sampling Results

Classifier	**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 Classification	0.92	0.72	0.93	0.70	0.92	0.71	0.88
Extra Tree Classification	0.93	0.77	0.94	0.75	0.94	0.76	0.90
Random Forest Classification	0.93	0.80	0.95	0.72	0.94	0.76	0.90
XGBoost Classification	0.93	0.86	0.97	0.71	0.95	0.78	0.92

From the table, it can be observed that the sampling technique used was ineffective as the scores remained nearly identical. Hence, the process for this dataset was concluded as redundant, and XGBoost Classifier was chosen as the best classifier, as it exhibited best scores, particularly F1 score.

5 Regression

In order to determine the delay in minutes of delayed flights, regression models were deployed, whose task was the prediction of a dependent variable with the help of other correlated independent variables. The pre-processed data was split into training and validation sets in the same ratio of 80:20. Keeping ArrDelayMinutes as the ground truth value and with the feature columns referenced in Table 8, regression models were trained to predict the flight delay.

Table 8. Features used by Regressors

DepDelayMinutes	CRSArrTime	CRSDepTime	DepDel15	windspeedKmph	winddirDegree
weatherCode	precipMM	visibility	pressure	cloudcover	DewPointF
WindGustKmph	tempF	WindChillF	humidity	airport	ArrDel15

5.1 Regression Metrics

The performance standard of regression models were measured using four important metrics, namely - Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2.

- Mean Absolute Error – MAE is the mean absolute difference between the target value and the value predicted by the model, hence is a linear score (all the individual differences are weighted equally).

  $$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

  where: $\hat{y}$ = predicted value of y

- Mean Squared Error – Being the average of the squared difference between the target value and the value predicted by the regression model, it penalizes even a small error which helps in clear understanding of the standard of the model.

  $$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

- Root Mean Squared Error – It is the square root of the averaged squared difference between the target value and the value predicted by the model. RMSE is useful when large errors are undesired.

  $$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

- R^2 – The metric compares the current model with a constant baseline by taking the mean of the data and drawing a line at the mean. R^2 is a scale-free score, hence will always be less than or equal to 1.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

where: $\bar{y}$ = mean value of y

Multiple regression models trained included Linear Regression, Random Forest Regression, XGBoost Regression etc., whose performance metrics are summarized in Table 9. Owing to the best performance metrics, Random forest regressor was chosen to be the best regressor.

Table 9. Regression Results

Regressor	**MAE**	**MSE**	**RMSE**	**R^2**
Linear Regression	12.18	308.23	17.55	0.94
Extra Tree Regression	11.94	288.76	16.99	0.94
XG Boost Regression	11.87	292.28	17.09	0.94
Random Forest Regression	11.74	278.58	16.69	0.94

6 Regression Analysis

To visualize how the delay in minutes was distributed and how this distribution affected the model, Random Forest Regression was considered and tested within different intervals of the validation set. Table 10 summarizes the metrics of the model under different intervals.

Table 10. Regression Analysis

Delay Interval (mins)	**MAE**	**RMSE**	**R^2**
15 - 100	10.49	13.82	0.6
100 - 200	17.80	26.13	0.05
200 - 1000	19.01	29.09	0.95
1000 - 2000	22.19	31.42	0.97

From the values, it was concluded that the regressor predicted more accurately in those intervals it was more familiar with, and less accurately in those intervals where the delay was scantily distributed and hence a decrease in credibility was seen because of this distribution of values. Nonetheless, owing to the fact that regression's optimization process applies chance correlations in the data, this inflated the R^2 value in the upper ranges, since there existed more points in lower range but with greater variance as compared to less points in upper range with smaller variance. This tailored the regressor model to accurately predict flights with greater delay, hence in the range of 1000 - 2000, though the value of MAE is large, it is better than the MAE value of 10.49 in the range of 15 - 100, since it is the mean of errors in a much smaller range with more points.

Fig. 2 illustrates the distribution of delay of flights in multiple intervals.

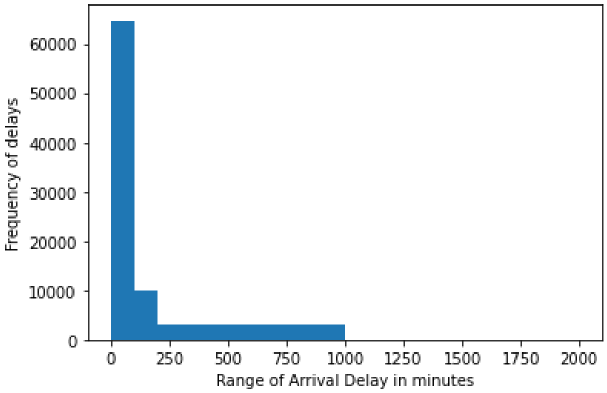

Fig. 2. Distribution of delay of flights in multiple intervals

7 Pipeline

The method of splitting data processing elements (machine learning modules) such that they can be pipelined together, often executed parallelly make the modules more efficient and simplified. Fig. 3 illustrates the flowchart on how the process was carried out.

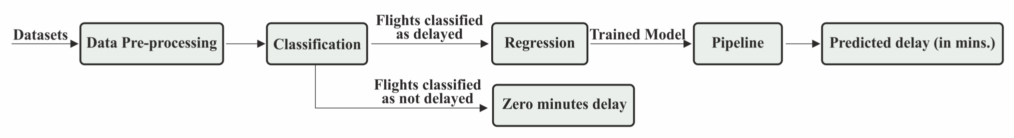

Fig. 3. Two stage predictive model flowchart

Against the optimum performance standards for this task, after observation and evaluation of metrics of Classifier and Regressor, XGBoost Classifier and Random Forest Regressor were used for the pipelined predictive model, the results of which are summarized in Table 11. The delay in minutes for flights classified as not delayed was deduced to be zero minutes.

Table 11. Pipeline Metrics

MAE	**MSE**	**RMSE**	**R^2**
13.649	340.134	18.442	0.947

8 Conclusion

Since the issue of flights being on-time is very important, predicting flight delay using models having high precision and accuracy is also important. Hence, in this project, real world dataset was used after processing and clean up. Suitable Classifiers were tested to classify the flights as delayed or not. Comparing the classifiers against each other in terms of their performance, XGBoost Classifier yielded the most satisfying predictive classification. Subsequently, the delay in minutes for the delayed flights were predicted using Regressors, which were compared with each other as well. Random forest regressor exhibited ideal performance and was chosen along with XGBoost classifier for the pipelined model. The issue of data imbalance was also addressed along with compensating techniques to finally yield the two-stage predictive pipelined model, where the Classifier and Regressor (trained beforehand) classified and predicted the delay in the validation set. The analysis thus retrieved can be used as a basic prototype by any Aviation Industry for further research by using advanced pre-processing techniques, sampling algorithms, and machine learning models. This is appreciable since it is not only important for the commuters and the aviation industry, it will help reduce their economic and environmental impact as well.