Two Stage Predictive Machine Learning Engine that Forecasts the On-Time Performance of Flights

Nitish Kumar M

Abstract. The uncertainty in weather prediction has been creating problems in the air transportation sector. This project focuses on training a two stage machine learning model that will forecast if a flight will be delayed and by how much using the forecasted weather of the destination and origin of the aircraft after departure.

Keywords: Classifiers · Regressors · Sampling · Metrics

1 Introduction

Flights all over the world are cancelled or delayed by hours due to the sudden changes in weather. These delay in flights have led to reduced productivity. Hence, it is highly desirable to forecast the delay of a flight using the forcasted weather and can be helpful for the aircraft company to schedule the flights early or late depending upon the weather, also keep the passengers informed.

2 Dataset and Preprocessing

Data required to train and test the machine learning models consists of two separate datasets namely the flight and weather data. These two datasets were merged such that it contains the flight data mapped along with the weather data of the destination and origin of the flight.

Flight Data comprises of the information of the flights over a period of two years (2016-2017) for every month and for the following airport codes which are mentioned in the Table 1. Out of all the available flight features, only certain features were considered which are mentioned in the Table 2.

Table 1. Airport Codes

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

Table 2. Flight Features

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

Weather Data comprises of weather details over a period of 5 years and are grouped by cities. The files of the years 2016 and 2017 was accessed to retrieve weather features for 24 hours everyday for every air station code. From all the available weather data, only certain weather features were considered which are mentioned in Table 3. The retrieved data from the files were processed and stored as the final weather data.

Table 3. Weather Features

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

Merging Flight and Weather Data: At first, the flight and weather data was merged on date, time and Origin airport code and merged another time with Destination airport code. The generated dataset consists of the flights data as well as the weather details of the origin city as well as the destination city. The final merged dataframe will be used for the training of machine learning models. The final dataframe was divided into training and testing datasets such that training data consisted of 95.8 % of the entire data.

3 Classification

In the development of a two stage machine learning model, the first stage involves a classification model that must be trained to classify if flights would be delayed or not. The dataset obtained previously consists of a column "ArrDel15" that provides a binary classification for delayed flights, which serves as the ground truth. This column was considered because a delay greater than 15 minutes is considered to be a delay. In this case, the models used are as follows:

- XGBoost Classifier
- Extra Trees Classifier

- AdaBoost Classifier
- Logistic Regressor.

Models were evaluated using the following metrics to determine the performance with respect to the dataset:

- Area Under the Curve (AUC)
- Accuracy
- Precision
- Recall
- F1 Score

A Receiver Operating Characteristic curve, or ROC curve, is a graphical plot that illustrates the ability of a binary classifier system as its discrimination threshold is varied. The curve is a two dimensional graph in which the false positive rate is plotted on the X axis and the true positive rate is plotted on the Y axis. An AUC of 0.5 suggests no discrimination, 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding.

Accuracy can be defined as the percentage of correctly classified instances. Accuracy is calculated as:

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

TP - True Positives (Correct prediction of delayed flights), FN - False Negatives (Incorrect prediction of flights not delayed), TN - True Negatives (Correct prediction of flights not delayed) and FP -False Positives (Incorrect prediction of delayed flights)

Recall is calculated as the number of true positives divided by the total number of true positives and false negatives. Similarly, Precision is calculated as the number of true positives divided by the total number of true positives and false positives. Recall is also termed as true positive rates.

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

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

The F1-score is one class of the F-score, is a measure of a model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model and is calculated as:

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

The models and their metrics are given in the Table 4.

Table 4. Classifier Model and Metrics

Class	Model	AUC	Accuracy	Precision	Recall	F1 Score
Class 0	XGBoost Classifier	0.8416	0.92	0.93	0.98	0.95
	Extra Trees Classifier	0.8234	0.91	0.92	0.97	0.94
	Logistic Regressor	0.8319	0.92	0.92	0.98	0.95
	AdaBoost Classifier	0.8331	0.92	0.92	0.98	0.95
Class 1	XGBoost Classifier	0.8416	0.92	0.90	0.70	0.79
	Extra Trees Classifier	0.8234	0.91	0.86	0.67	0.76
	Logistic Regressor	0.8319	0.92	0.89	0.69	0.78
	AdaBoost Classifier	0.8331	0.92	0.89	0.69	0.78

Data Imbalance Upon observing the final dataset, it can be seen that the number of delayed flights constitute 21 percent of the entire data. The distribution of the data is given in the Fig 1. This leads to reduced performance of the classifier models as the number of data with which it is trained of is very less in number. This condition of the datasets is termed as data imbalance and can be taken care using sampling.

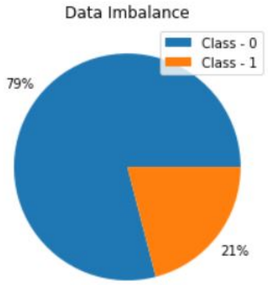

Fig. 1. Data Imbalance

Sampling Sampling is a method used on a imbalanced dataset to improve the performance of a classifier model by modifying the training dataset contents according to the model's performance requirements. In this case, sampling was done using Random oversampling which just increases the size of the training data set through repetition of the original examples. Sampling is done in two ways to reduce the skew in the dataset which are oversampling and under-sampling.

Oversampling involves increasing the number of delay data points in the dataset. Under-Sampling involves decreasing the number of non-delay data points in the dataset. The training data distribution after sampling is visualised in the Fig 2. The datasets produced were used to train the classifier models and their metrics were calculated, which are mentioned in Table 5 and Table 6 respectively.

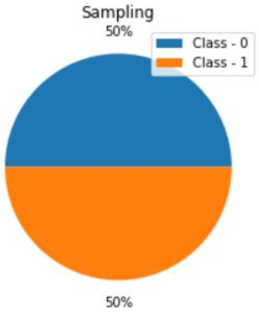

Fig. 2. Sampling

Table 5. Random Oversampling Classifier Models and Metrics

Class	Model	AUC	Accuracy	Precision	Recall	F1 Score
Class 0	XGBoost Classifier	0.8645	0.90	0.95	0.92	0.93
	Extra Trees Classifier	0.8126	0.90	0.91	0.97	0.94
	Logistic Regressor	0.8535	0.90	0.94	0.93	0.93
	AdaBoost Classifier	0.8545	0.89	0.94	0.93	0.92
Class 1	XGBoost Classifier	0.8645	0.90	0.73	0.81	0.77
	Extra Trees Classifier	0.8126	0.90	0.86	0.65	0.74
	Logistic Regressor	0.8535	0.90	0.74	0.78	0.76
	AdaBoost Classifier	0.8545	0.89	0.73	0.79	0.76

Upon observing the metrics of different models, it can be seen that XGBoost Classifier trained without any sampling has the best performance as the value of AUC is 0.84159, which is the highest and as mentioned earlier an AUC of 0.8 to 0.9 is considered excellent.

Table 6. Random Under-sampled Classifier Models and Metrics

Class	Model	AUC	Accuracy	Precision	Recall	F1 Score
Class 0	XGBoost Classifier	0.8651	0.90	0.95	0.92	0.93
	Extra Trees Classifier	0.8578	0.87	0.95	0.88	0.92
	Logistic Regressor	0.8537	0.90	0.94	0.93	0.93
	AdaBoost Classifier	0.8546	0.89	0.94	0.92	0.93
Class 1	XGBoost Classifier	0.8651	0.90	0.73	0.81	0.77
	Extra Trees Classifier	0.8578	0.87	0.65	0.83	0.73
	Logistic Regressor	0.8537	0.90	0.74	0.78	0.76
	AdaBoost Classifier	0.8546	0.89	0.73	0.79	0.76

4 Regression

After the classification, the second stage of the machine learning engine is a regressor which determines the delay in minutes. The dataset is filtered such that it contains data only of delayed flights. The models used in this case are as follows:

- XGBoost Regressor
- Decision Tree Regressor
- Linear Regressor
- AdaBoost Regressor

Models were evaluated using the following metrics to determine the performance with respect to the dataset are as follows:

- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R Squared (R2)

Root Mean Squared Error (RMSE) is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model and is calculated as:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (5)$$

Mean Absolute Error refers to the mean of the absolute values of each prediction error on all instances of the test data-set. Prediction error is the difference between the actual value and the predicted value for that instance. MAE is calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (6)$$

R^2 is a statistical method that represents the proportion of variance for a dependent variable with respect to independent variables in a regression model and is calculated as:

$$R^2 \ = \ 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2} \tag{7}$$

The models and their metrics are given in the Table 7.

Table 7. Regression Models and Metrics

Model	RMSE	MAE	R^2
XGBoost Regressor	30.9803	16.4578	0.8164
Decision Tree Regressor	26.0537	19.3289	0.8701
Linear Regressor	20.8968	15.1296	0.9164
AdaBoost Regressor	21.3434	15.1844	0.9128

Upon observing the metrics, the Linear Regressor has the best performance as the value of RMSE is of 20.89 which is the lowest when compared with other models.

5 Pipeline

The problem statement involves classification of the flights when the necessary features are given as an input, whether they would be delayed or not and by how many minutes will it be delayed. This shows that the output of the classifier is linked to the input of the regressor. Hence, this is a pipeline process. A block diagram explaining the pipeline process is shown in the Fig 3. In this case, the output of the classifier that was stored as the classified flights will now act as the input to the regressor models. This classified data was passed into the trained regressor models and the delay in minutes was predicted. The output of every model was predicted and metrics were calculated, which are mentioned in the Table 8.

Table 8. Pipeline Linear Regressor Model and Metrics

Model	RMSE	MAE	R^2
Linear Regressor	18.3455	13.5178	0.9470

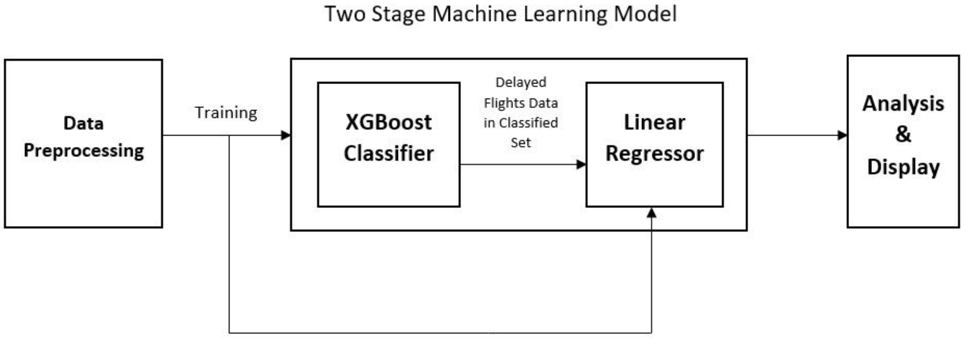

Fig. 3. Block Diagram

From the metrics, it can be seen that the performance of the regression models when pipelined is much better than the regressors with the original test data. This assumption is false on all grounds. The test data that is passed to the regressor models is the output of the classifier and this classifier output data would contain errors as the accuracy of the classifier is neither 1.0 nor nearly equal to one. Hence, there will be a deviation of the output of the two stage model when compared with the actual output.

6 Regression Testing

For further evaluation of the regression models, the trained models were passed with grouped values of the test data. The test data was divided into 5 different categories (15-100, 100-200, 200-500, 500-1000, 1000+) and passed into the models for prediction after which their metrics were evaluated. The purpose of this is to evaluate the performance of the models in a particular range from which we can develop ways to improve the overall performance of the model. As the Linear model has the best performance for the dataset, comparing its metrics on different ranges are mentioned in the Table 9.

The amount of training examples in every range is unequal and decreases as the range increases due to which the learning process of the model in those ranges diminish. From the table 6, the MAE and RMSE values does not define the actual performance of the regressor in that range. A MAE of 26 in the range of 1000+ can be considered acceptable rather than a MAE of 14 in the range of 15-100. Upon observing the metrics of different ranges, we could say that the regressor has performed well in the ranges 200-500, 500-1000 and 1000+. From

Table 9. Linear Regression Metrics in different ranges

Arrival Delay in Minutes	RMSE	MAE	R^2
15-100	19.0587	14.5869	0.2434
100-200	26.6575	16.7485	0.0094
200-500	32.9860	20.5927	0.7805
500-1000	30.9685	26.8483	0.9608
1000+	29.3995	26.5174	0.9674

the metrics, we can say that the regressors have the a better performance in the range of 200-500, 500-1000 and 1000+.

7 Conclusion

In the modern age of data, machine learning approach to determine the delay in flights can be incorporated on a large scale to avoid several losses and this can be made much reliable depending upon the quality of data. The prediction of delay of flights after departure is approached with a two step process which involves in the classification of the flights at first and then passing it's output to predict the delay in minutes. The XG Boost Classifier is the best fit for classification and Linear Regressor is the best fit for regression and their metrics support this claim. As we know the machine learning models are data driven, the quality of data available determines the performance of the models. The output from the machine learning models are accurate to a great extent but with a better and larger dataset as this, the models can be trained to higher level of perfection.

Appendix A

At first, the two stage machine learning engine was trained to forecast the delay in flights before it's departure ie., without considering "DepDelayMinutes" as a feature while training the models. It was considered that "DepDelayMinutes" formed a part of the ground truth and hence must not be included while training. This assumption was later proven wrong by finding the correlation between "DepDelayMinutes" and ground truth. Correlation can be useful to better understand the relationships between variables. The value of correlation was 0.965712, a value of 0.9 and 1.0 indicate variables which can be considered very highly correlated. Hence this input feature can be considered. Finally, the models were re-trained including "DepDelayMinutes" as an input feature.