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

Nitish Kumar M

[1] SSN College of Engineering, Kalavakkam, Chennai 603 110
[2] nitish.visva@gmail.com

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 predict if a flight will be delayed and by how much with respect to the predicted weather of the destination and origin of the aircraft.

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 hold ups in flights have led to the loss of productive man hours. Apart from mechanical issues, the major factor responsible for the delay is weather. Hence it is highly desirable to predict the delay of a flight in advance with respect to the predicted 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 different data sets namely the flights data and weather data. The merging was done using pandas and NumPy libraries of python. These two data sets were merged such that it contains the flight data mapped along with the weather data of the destination and origin of the flight accordingly.

Flights Data; Data comprises of the information of the flights over a period of two years (2016-2017) and has several features (Origin, Destination, Departure time & Delay, Arrival time & Delay, etc) and this data is available as a .csv file for every month over two years which are stored individually in separate directories. Every .csv was accessed one after the other in which Arrival time

*

and Departure time was changed accordingly in order to facilitate the merging of this data along with the weather data. The Flights Dataframe was generated by compiling the data from the individual .csv files taking into consideration only some of the important features.

Weather Data; Data comprises of weather details over a period of 5 years and are individually stored in .json format in directories named after the cities. The weather data was accessed using the json library in python. The json files of the years 2016 and 2017 was accessed to retrieve weather features for 24 hours everyday for every city code. From all the available weather data, only certain weather features (humidity, visibility, wind speed, cloud cover,etc) were considered. The retrieved data from the json files were processed and stored in the form of a pandas dataframe which was saved as a .csv file.

Merging Flight and Weather Data; Merging the two sets of data was done using pandas.Merge function and in two steps. At first, the flights and weather data was merged upon date, time and Origin city code. This merged dataframe was merged the second time with the weather data upon date,time and Destination city 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 be used for the running of machine learning models.

The final dataframe was processed in order to remove the Nan's and divided into training and testing datasets such that training data consists of more than 70 % of the entire data and are saved separately as .csv files.

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 flights on whether they would be delayed or not. The dataset generated previously consists of a column "ArrDel15" that contains a 1 for flights that has been delayed over 15 minutes and 0 otherwise, which serves as the ground truth for the classifier. This column was considered because a delay greater than 15 minutes is significant rather than a delay by few minutes.The training and testing datasets were imported, the ground truth and the other features were separated to train the classifiers.

The "scikit-learn" library of python was used to develop and train different kinds of classifier models, in this case the models used were XG Boost Classifier, Extra Trees Classifier, Ada Boost Classifier and Logistic Classifier. These models were trained with the training data by passing the input features and ground truth. The trained models were used to predict the output by passing the input features of the test data and the metrics were calculated for every model to determine the performance with respect to the dataset as the performance of every model varies with different types of datasets. The performance evaluation of the classifiers are done using the metrics Area Under the Curve (AUC), Accuracy, Precision, Recall and 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 Area of the ROC curve is known as AUC and is calculated as:

$$TruePositiveRate(TPR) = \frac{TP}{TP + FN} \tag{1}$$

$$FalsePositiveRate(FPR) = 1 - TPR = \frac{FP}{FP + TN} \tag{2}$$

TP - True Positives, FN - False Negatives, TN - True Negatives and FP -False Positives

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

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

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.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

The F-score, also called the F1-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:

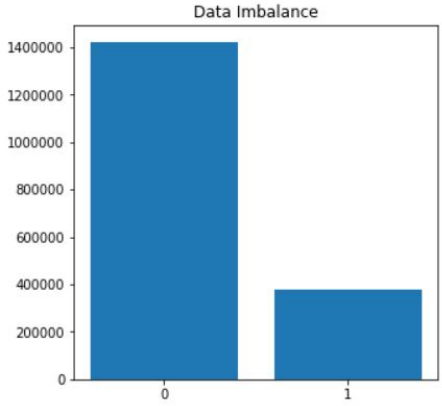
$$Precision = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{6}$$

The models and their metrics are given in the table below.

Table 1. Classifier Model and Metrics

Model	AUC	Accuracy	Precision	Recall	F1 Score
XG Boost Classifier	0.578105	0.81	0.75	0.58	0.59
Extra Trees Classifier	0.617575	0.79	0.67	0.62	0.63
Logistic Classifier	0.506955	0.79	0.72	0.51	0.46
Ada Boost Classifier	0.517382	0.79	0.70	0.52	0.48

Data Imbalance Upon observing the final dataset, it can be seen that the number of delayed flights constitute less than 20 percent of the entire data. The distribution of the data is given in the figure below. Due to this, the performance of the classifier models decreases as the number of data with which it is trained of is very less in proportion. This condition of the datasets is termed as data imbalance and must be taken care using sampling.



Sampling Sampling is a method used on a data imbalanced dataset to improve the performance of a classifier model by changing the training dataset contents according to the model's performance requirements. In this case, sampling can be done in two ways to achieve the required data set which are oversampling and under sampling.

Oversampling involves increasing the number of 1's present in the dataset or in other words increasing the number of "true" ground truths in the dataset. This was done using the "imblearn" library of python.The training data distribution after oversampling is visualised in the figure below. The dataset produced from oversampling was used to train the classifier models again and their metrics were calculated, which are mentioned below.

Under Sampling involves reducing the number of 0's present in the datasets in other words decreasing the number of "false" ground truths in the dataset. The training data distribution after under sampling is visualised in the figure below. The dataset produced from under sampling was used to train the classifier models and their metrics were calculated, which are mentioned below.

Sampling does not necessarily result in improving the quality of the dataset for training. Hence the all the classifier models were compared. Sampling can

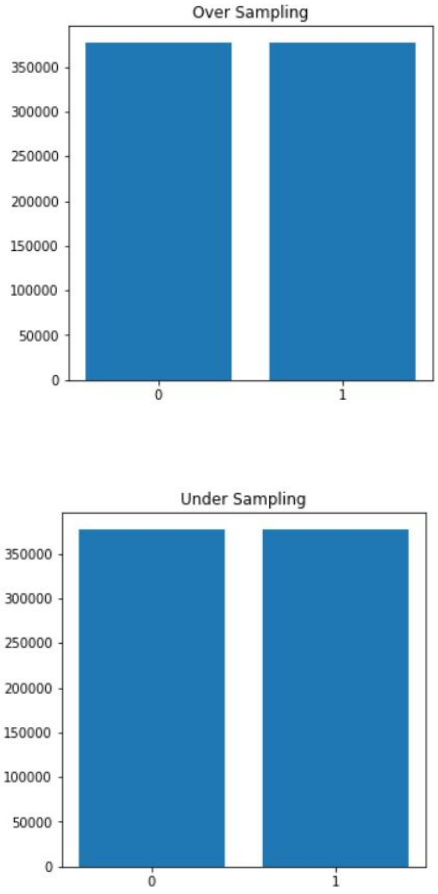


Table 2. Oversampled Classifier Models and Metrics

Model	AUC	Accuracy	Precision	Recall	F1 Score
XG Boost Classifier	0.678735	0.69	0.63	0.68	0.63
Extra Trees Classifier	0.620328	0.78	0.66	0.62	0.63
Logistic Classifier	0.6080403	0.61	0.57	0.61	0.55
Ada Boost Classifier	0.630610	0.63	0.59	0.63	0.57

Table 3. Under Sampled Classifier Models and Metrics

Model	AUC	Accuracy	Precision	Recall	F1 Score
XG Boost Classifier	0.676409	0.69	0.62	0.68	0.62
Extra Trees Classifier	0.667277	0.68	0.62	0.67	0.62
Logistic Classifier	0.607951	0.61	0.57	0.61	0.55
Ada Boost Classifier	0.630891	0.63	0.59	0.63	0.57

only be done to improve the performance of a classifier but not a regressor as absolute values cannot be sampled.

Upon observing the metrics of different models, it can be seen that XG Boost Classifier trained with an oversampled training data has the best performance. The "ArrDel15" column in test data is replaced with the predicted output of the XG Boost classifier and the dataframe at which "ArrDel15" = 1 is stored as separate .csv file which will have been later used for pipelining.

4 Regression

After the classification, the second stage of the machine learning engine is a regressor which determines the minutes by which an aircraft would be delayed is predicted for the flights with respect to the weather using regression models. The datasets are initially changed such that it contains data of "ArrDel15"=1 to make sure that the model is not trained with redundant data values. Once again the training and testing dataset was processed to separate the input features and the ground truth. In this case, the ground truth is the column "ArrDelayMinutes".

The "scikit-learn" library of python was used to import and train the regression models. The models used in this case were XG Boost Regressor, Decision Tree Regressor, Linear Regressor and Ada Boost Regressor. These models were imported and trained by passing the input features and ground truth of the training dataset. The trained models were then tested by passing the testing data and their metrics were calculated for every model to compare the performance of models with respect to the dataset. The performance of the models are determined using metrics like Root Mean Squared Error (RMSE) , Mean Absolute Error (MAE) and R Squared (R2). The corresponding equations to determine the respective metrics are as follows.

RMSE is the standard deviation of the residuals which are a measure of how far from the regression line, data points are located and it is calculated as follows.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2} \tag{7}$$

MAE is a measure of errors between paired observations expressing the same phenomenon (one technique of measurement versus an alternative technique of measurement). MAE is calculated as:

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{8}$$

R2 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:

$$R2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2} \tag{9}$$

The models and their metrics are given in the table below.

Table 4. Regression Models and Metrics

Model	RMSE	MAE	R Squared
XG Boost Regressor	80.847	44.15276	-0.2500846
Decision Tree Regressor	92.14008	54.74519	-0.623711
Linear Regressor	85.12767	46.88885	-0.3859676
Ada Boost Regressor	97.038	79.718585	-0.80094

Upon observing the metrics, the XG Boost Regressor has the best performance when compared with other models for this dataset. Hence XG Boost regressor is the best fit for the dataset available.

5 Pipeline

The problem statements 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. 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 below.

Table 5. Pipeline Regression Models and Metrics

Model	RMSE	R Squared	MAE
XG Boost Regressor	50.0646	27.88345	0.115126
Decision Tree Regressor	67.92314	33.40706	-0.628753
Linear Regressor	53.255	28.3093	-0.001262
Ada Boost Regressor	112.136	103.5623	-3.439328

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 engine 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 it's prediction of a particular range. As the XG Boost model has the best performance for the dataset, comparing its metrics on different ranges are mentioned below.

Table 6. XG Boost Regression Metrics in different ranges

Arrival Delay in Minutes	RMSE	R Squared	MAE
15-100	31.5312	23.9467	-1.0708817
100-200	108.156	102.2913	-15.306257
200-500	246.851	232.6989	-11.28709
500-1000	702.0748	683.6350	-19.127661
1000+	1290.4224	1279.2172	-61.674175

From the metrics, we can say that the regressors have the best performance in the range of 15-100 which implies that the model has the ability to accurately predict the outputs if the input features fall in that range and this accuracy reduces in other ranges. The reason behind this difference is due to the unavailability of more number of data in other ranges due to which the learning process of the model in those ranges diminish.

7 Conclusion

The prediction in the delay of Flights is approached with a two step process which involves in the classification of the flights at first and then passing the it's output to predict the delay in minutes. It initially involves in the processing of the data into a suitable format for the purpose of training and testing models. The data must be sampled due to a data imbalance and the classifiers trained and tested accordingly to determine the best fit for the data whose output is then fed as an input to the trained regression models to predict the delay in

minutes as the solution is a pipeline process. It is seen that the models are well learned in the range of 15-100 as the data available in the other ranges are very less in number.

The project involves methods to improve the performance of the models by improving the data (such as sampling) and also several ways to evaluate the performance of the models to realise areas which needed improvements. As we know the machine learning models are data driven, the quality of data available determines the performance of the models. The outputs 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.