Two-Stage Flight Delay Prediction and Classification Pipeline

Muhammad Waseem H

Abstract

Flight delay brings a negative impression towards the airline company. It necessarily need not be the sole cause of the airline company itself. Several factors that cause airlines to get delayed are accidents, natural disasters, system failures, and most importantly weather. Harsh weather like heavy rainfall, heavy windstorm, heavy snow etc would definitely cause the flight to get delayed. The aim of this two-stage pipeline project is to find out whether a flight would get delayed provided the weather conditions and for the flights that got delayed the model also predicts the arrival delay time.

1 Introduction

Flight delays lead to delayed arrival time and it affects passengers. Many have their own schedule and the level of importance varies from person to person. If there exists a model that could detect the possibility of flight delays with prior weather assumptions to some extent and accuracy it would be easier for passengers to look for alternatives without any serious loss.

In the worst case, even if flight delay is unpredictable if there exists a regression model that could predict the arrival delay minutes, that could help passengers to schedule their work in prior accordingly. This two-stage pipeline project is built with a classification and a regression model that classifies whether a flight would get delayed provided the weather conditions and would predict the delayed arrival time in minutes for those flights that are delayed.

2 Dataset

The weather details of 15 US States are provided between the years 2013 and 2017 (Both inclusive) in JSON format. It includes several attributes but we are only interested in the following attributes.

WindSpeedKmph	WindDirDegree	WeatherCode	precipMM
Visibility	Pressure	Cloudcover	DewPointF
WindGustKmp	tempF	WindChillF	Humidity
date	time	airport	

Table 1: Weather Information

From the weather dataset, we only need to retrieve weather information for 2016 and 2017 and that too only with the above attributes.

Another dataset contains details of flights that flew in the years 2016 and 2017 including columns like source/destination airports, departure, arrival time etc. It actually contains 110 columns. For our purpose, we consider the following columns alone.

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

Table 2: Flight Information

But the dataset for 2016 and 2017 also has details about all other airports but we are interested only in the following US states.

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

Table 3: Interested Airports

The datasets are raw. In the sense, it is to be preprocessed and merged with weather and flight information based on the date and time of departure.

3 Model

3.1 Preprocessing steps

The following preprocessing steps are performed before merging the weather and flight datasets.

- Null values removed
- Weather data for only the years 2016 and 2017 are extracted.
- Irrelevant columns dropped from both the weather and flight datasets.

3.2 Classification

The first stage of the pipeline is classification, i.e classifying whether flights would be delayed for departure or not provided the weather conditions. The merged data frame contains 18,51,422 rows after all the pre-processing steps mentioned above. Python Sklearn library is used for all classification models. The models used are XGboost, Decision Trees, Logistic Regression and Extra Trees classifier. The data frame contains a column DepDel15 which is the target variable, i.e it is equal to 1 if the flight departure delay minutes is greater than 15 minutes, else 0. The data is appropriately standardised. One hot/label encoding is applied to convert airport codes to integer values.

3.3 Classification Metrics

- **Accuracy**

  Accuracy is the ratio of Correct predictions out of total values.

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

- **Precision**

  It tells us how many predicted positives are actually positives.

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

- **Recall**

  It tells us how many actual positive class is predicted as the positive class.

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

- F_1 **Score**

  F_1 It combines Precision and recall in a way such that it is the harmonic mean of them.

$$F_1 Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Decision Trees	0.86	0.43	0.86	0.43	0.86	0.43	0.76
Logistic Regression	0.79	0.52	1.00	0.02	0.88	0.03	0.79
Extra Trees Classifier	0.94	0.34	0.85	0.60	0.89	0.44	0.82
XG Boost Classifier	0.98	0.17	0.82	0.71	0.90	0.27	0.82

Table 4: Classifier Performance before class imbalance treatment

The results shown above are calculated before dealing with the class imbalance problem

Data Imbalance Problem

While training the model, the dataset is divided into train and test with 75% and 25% weightage. When we look into the distribution of data, the availability of class 0 (not delayed flights) is almost 4 times that of class 1 (Delayed flights). Such a situation is said to be a data imbalance problem.

When a data imbalance problem occurs accuracy can't be the perfect metrics. We hence need F1 score to deal with it.

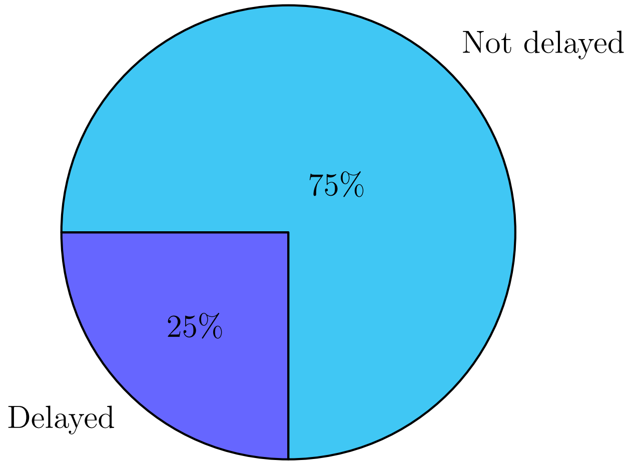

Figure 1: Before Applying Smote

From the above figure, we can clearly see there is an imbalance in the dataset. In order to properly train such a model we need to either upscale or downscale minority class or majority class respectively. There are many such approaches with which we can achieve it.

Few known Oversampling and Undersampling techniques are

- **Random Over Sampler**

  Random Oversampling involves randomly duplicating data from the minority class and adding it to the training data. It might lead to overfitting.

- **Synthetic Minority Oversampling Technique(SMOTE)**

  This is also an oversampling technique. The new instances are generated by randomly selecting one or more of the k-nearest neighbours for each instance in the feature space in the minority class.

- **Random Under Sampler**

  Random Under Sampling involves randomly removing data from the majority class and adding it to the training data. It might lead to loss of information.

- **NearMiss**

  It is an undersampling technique where we remove majority class instances by checking if there are instances of two different classes that are very close to each other in the feature space. We remove the instances of the majority class to increase the space between the two classes.

SMOTE technique is usually preferred. It also considerably prevents overfitting compared to randomly upsampling minority class.

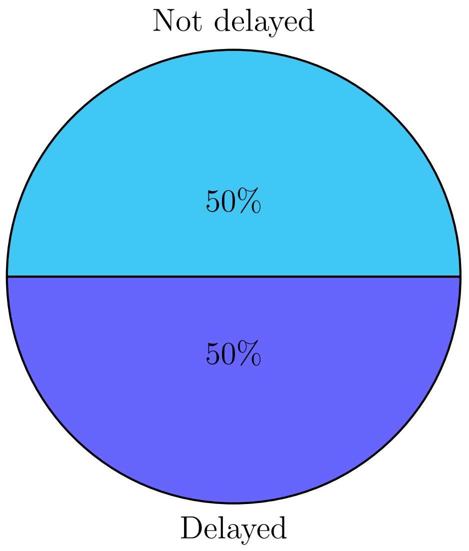

Figure 2: After applying smote

The classifier with the best performance is Extra Trees Classifier, as it has the highest F_1 Score.

Algorithm	Precision		Recall		F_1 Score		Accuracy
	0	1	0	1	0	1	
Decision Trees	0.84	0.46	0.86	0.42	0.85	0.44	0.76
Logistic Regression	0.85	0.25	0.61	0.61	0.71	0.40	0.60
ExtraTrees Classifier	0.92	0.42	0.86	0.57	0.89	0.48	0.82
XG Boost Classifier	0.93	0.26	0.83	0.49	0.88	0.34	0.80

Table 5: Classifier Performance after smote

It can be seen from Tables 4 and 5 that the F1 score was improved when smote is applied.

3.4 Regression

The second stage of the pipeline is Regression, i.e predicting arrival delay time for flights that are delayed for departure. Ideally, We should consider flights that are classified as delayed by the classification model to consider for regression. But here for building a robust model all rows with ArrDel15 as 1 are used for training the regression model. Here ArrDelayMinutes is the target variable.

Regression Metrics

To evaluate the regressor models, we use the following metrics.

The following notations stand for :
$\bar{Y}$: Mean Value Of Y
$\hat{Y}$: Predicted Value Of Y
N: Number of Data Points

- **Mean Absolute Error**

$$Mean\ Absolute\ Error(MAE) = \frac{1}{N}\sum_{i=1}^{N} \mid Y_i - \hat{Y}_i \mid$$

- **Mean Square Error**

$$Mean\ Square\ Error(MSE) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

- R^2 **Score**

$$R^2 Score = 1 - \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}$$

Regression Models

In this Pipeline, the regressors used are

- Linear Regressor
- XG Boost Regressor
- Extra Trees Regressor
- Random Forest Regressor

Regressor Performance

Regression Model	MSE	MAE	R^2 Score
Linear Regressor	396.0	14.63	0.92
Extra Trees Regressor	342.0	12.76	0.93
Random Forest Regressor	328.0	12.56	0.93
XB Boost Regressor	291.3	11.8	0.94

Table 6: Performance of The Regressors

R^2 score indicates how better the model is compared to the average baseline model. Closer to 1 better is the model. Mean squared and mean average error describes how far the predicted value is from the actual value. The lower these metrics are better the performance model. Among the models built, XG Boost has the least MSE, MAE and highest R^2 score. Hence it is selected for the pipeline.

4 Regression Analysis

The arrival delay for the flights classified as delayed was between 0 to 2000 minutes. Figure 3 shows the distribution of the flights in the delay intervals. The performance of the Extra Trees Regressor in these ranges is given in Table 6. From Table 6, it is clear that most of the flights had a delay between 15 - 100 minutes.

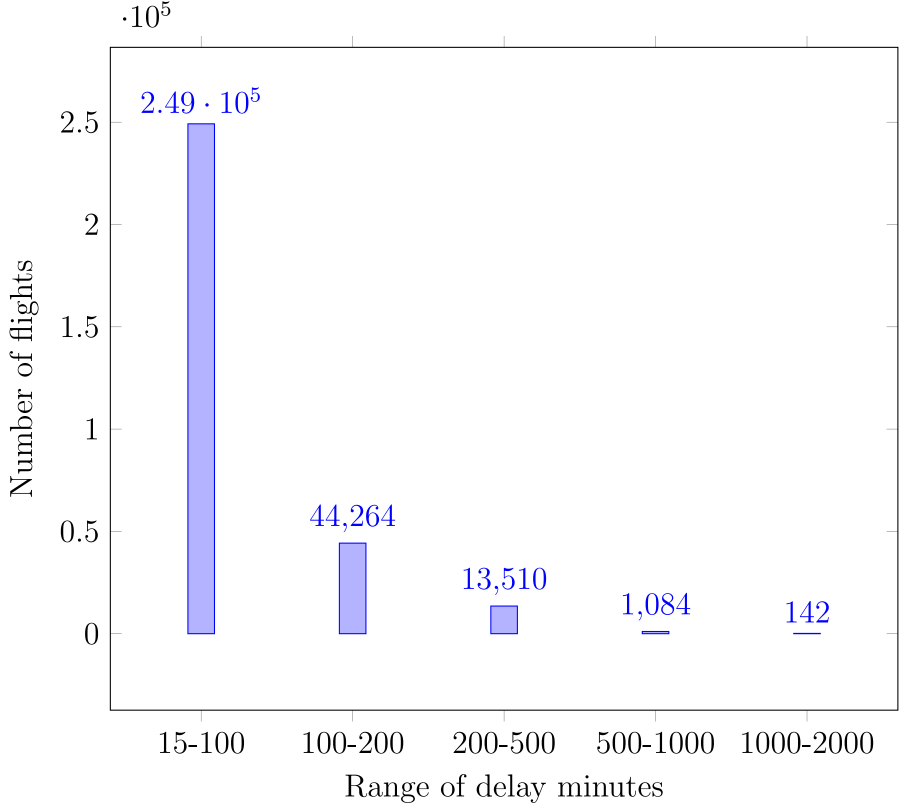

Figure 3: Distribution of flights in different ranges of delays

ArrivalDelayMinutes	No Of Flights	MSE	MAE
15 - 100	249204	350.8	14.03
100 - 200	44264	407.8	13.82
200 - 500	13510	469.84	15.05
500 - 1000	1084	346.16	13.15
1000 - 2000	142	351.16	13.84

Table 7: Frequency distribution and range wise Regression scores of the Flights

MAE or MSE values tell us how close the predicted data is to the original data. A lower MAE or MSE value indicates better performance by the regressor. From Table 7, we can say that in the range 500 - 1000, the model performance is better compared to the data in the other ranges.

5 Pipelining

Extra Trees Classifier and XG Boost Regressor were chosen to build the pipeline model. The classification performance is shown in Table 5. The regression performance is shown in Table 7. Figure 4 shows the structure of the pipeline model.

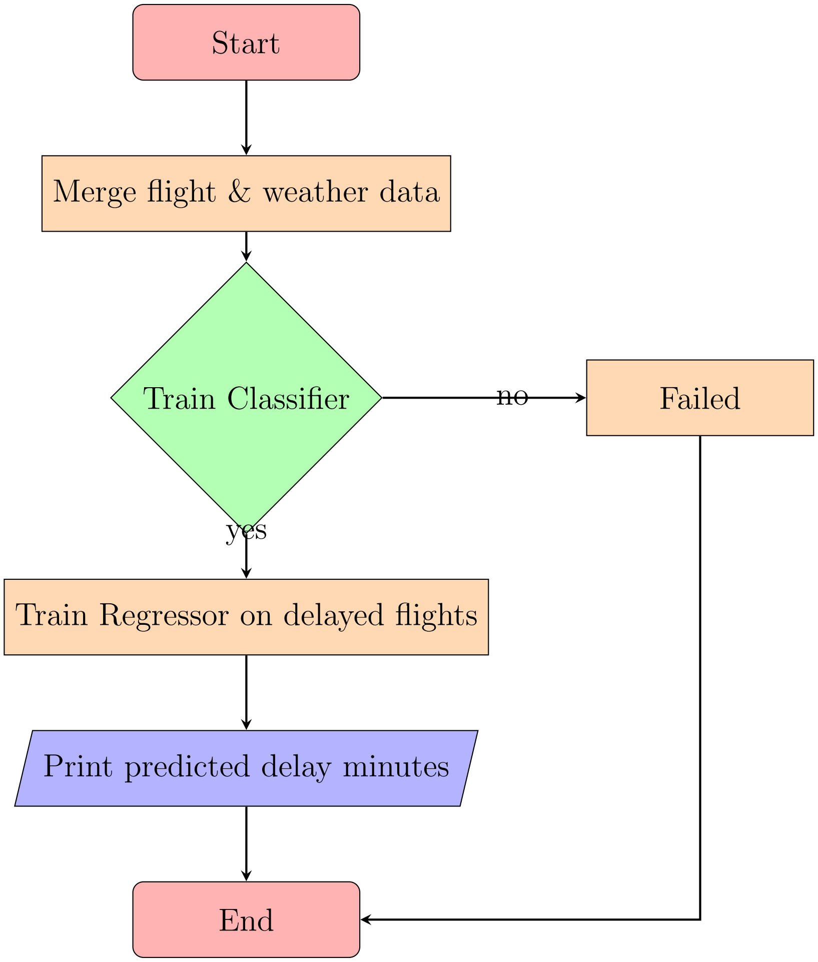
Start
Merge flight & weather data
Train Classifier
no
Failed
yes
Train Regressor on delayed flights
Print predicted delay minutes
End

Figure 4: Two-Stage pipeline

Metric	Value
Accuracy	0.82
MAE	11.8
MSE	291.3
$R^2 Score$	0.94

Table 7: Performance of the pipeline model

6 Conclusion

Initially, the data from weather and flight datasets are merged so that it included only the necessary attributes for the years 2016 and 2017. Two-stage pipeline model is built with Extra trees classifier and XG Boost regressor. The overall accuracy of classification is 0.82. The model performs better with class 0 compared to class 1 due to the class imbalance. To an extent, the intensity of the class imbalance issue is reduced with SMOTE technique. The $F1$ score of the classification model turned out to be close to 0.50 which is a decent model. The MSE, MAE and R^2 scores of the regression model turned out to be 291.3, 11.8, 0.94 which is pretty decent as well. The higher the R^2 scores, the lower the MSE and MAE value means the better the model. Finally, with this pipelined model it would be possible to classify whether a flight would get delayed and if so it would be possible to predict the arrival delay minutes.