Flight-Delay Prediction

Mohan Vamsi

November 12, 2020

Abstract

The main objective of the project is to predict the arrival delay of the flights using a two-stage model. If a flight is classified a delayed, then it is piplined with the regressor to predict the delay. To predict the flight delay real world weather data is used, as weather has a major impact on the delay.

1 Introduction

Flights play an important role in the day-to-day life and it is considered as quickest means of international transport since the day it was invented. Flights follow a schedule to avoid air traffic and other difficulties. But when a flight is not on schedule it is said to be delayed. There are many reasons for the delay, most common reasons are due to bad weather and other human errors which affect the passengers, airport authority, and airline crew members.

So, the flight delay is predicted to maintain the air-traffic and to prevent economic loss for the airport authorities. The model predicts the delay in two stages one (by classifying the flights as delayed or not delayed) and two (by predicting the flight delay for the flights classified as delays with a regression model). The flight data from 15 airports in the United States and the corresponding weather data in 2016 & 2017 are used to train the model.

2 Data Preprocessing

The data used for building the model are: **Flight data** & **Weather data**

- **Flight data**

  The flight data consists of the 15 specific airports in United States that flew during 2016 & 2017. The airports that are considered are listed in the Table 1. The attributes that are considered from the flight data are shown in Table 2.

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

Table 1: Airports

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

Table 2: Flight Attributes

- **Weather data**

  The weather data from the above airports is considered from the year 2016 & 2017. The Table 3 shows the attributes that are taken into consideration.

WindSpeedKmph	WindDirDegree	WeatherCode
Visibility	Pressure	Cloudcover
WindGustKmph	tempF	WindChillF
date	time	airport
precipMM	DewPoint	Humidity

Table 3: Weather Features

The data contains null values so they are replaced with the mean value of the column to increase the performance of the model. The flight data and weather data are merged using attributes **date, time** and **airport**. Weather conditions at the origin airport and the destination airport are considered. SelectKBest is a popular feature selection technique that is used to retain relevant features and drop unwanted features. It is used to find most correlated features with the target variable(i.e ArrDel15/ArrDelayMinutes) and the least correlated features which are to be dropped and then the model is trained.

3 Classification

Classication models are used to classify if the flights are delayed or not.

According to the dataset, flights having attribute **ArrDel15 = 1** are considered to be delayed while the flights having attribute **ArrDel15 = 0** are considered not delayed. The train set and test set are split in a ratio of 70:30 i.e 70% of the data from the dataset is considered as training data and rest of the data is considered as testing data.

A set of classification metrics are used to evaluate the performance of the classifier and to compare it with the performance of other classifiers.

3.1 Classification Metrics

- **Accuracy** is the ratio of true results to the total number of results that are examined.

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

- **Precision** tells what proportion of predicted positives are truly positive.

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

- **Recall** tells what proportion of actual positives are correctly classified.

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

- **F1-Score** is the harmonic mean of precision and recall.

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

where TP-True Positives, TN-True Negatives, FP-False Positive, FN-False Negative.

With respect to the dataset **True Positive** is an outcome where the model correctly classifies the delayed flights. **True Negative** is an outcome where the model correctly classifies the non-delayed flights. **False Positive** is an outcome where the model incorrectly classifies non-delayed flights as delayed. **False Negative** is an outcome where the model incorrectly classifies the delayed flights as non-delayed.

The Classifiers considered are:

- Decision Tree
- XG Boost
- Random Forest
- Logistic Regression

CLASSIFIER	*CLASS*	*PRECISION*	*RECALL*	*F1-SCORE*	*ACCURACY*
Decision Tree	0	0.92	0.92	0.92	0.87
	1	0.68	0.72	0.69	
XG Boost	0	0.92	0.98	0.95	0.92
	1	0.90	0.69	0.78	
Random Forest	0	0.92	0.97	0.95	0.91
	1	0.87	0.70	0.77	
Logistic Regression	0	0.92	0.98	0.95	0.92
	1	0.89	0.68	0.77	

Table 4: Classifier Performance

Class 0- Non-delayed Flights **Class 1**- Delayed flights

Table 4 shows the classifier performance for the dataset. However, In all the metrics "Class 1" performance is weaker when comparing with "Class 0". This is due to data imbalance between the two classes.

3.2 Data Imbalance

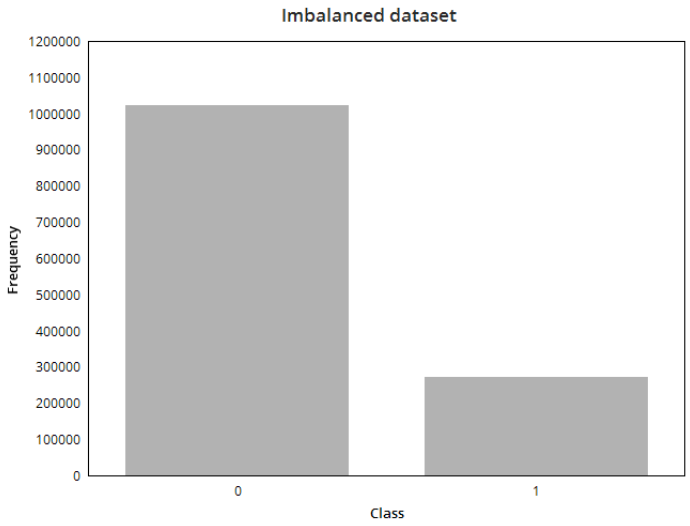

Figure 1: Imbalance dataset

There is a heavy class imbalance, i.e there are more number of flights which are non delayed than the number of flights which are delayed as shown in figure 1. This affects the performance of the model.

When subjected to data imbalance, the predictions have misleading accuracy. This is due to lack of information about the minority class when compared with the majority class. So, Data can be balanced either by oversampling or undersampling the data.

NearMiss technique is used for undersampling. In this technique, it eliminate's majority class examples by checking if there are instances of two different classes that are very close to each other in the feature space. The instances of the majority class are removed to increase the space between the two classes. By removing the miss leading data points undersampling, it reduced the size of the dataset which results in decreasing the run time. Figure 2 shows balanced dataset after applying undersampling.

SMOTE (Synthetic minority oversampling technique) is used for Over sampling. SMOTE technique is preferred as it creates new samples in the minority class instead of duplicating the existing samples present in the minority class. By creating new samples SMOTE technique mitigates the problem of over fitting. Once we perform the sampling the data set will be balanced as shown in the figure 3.

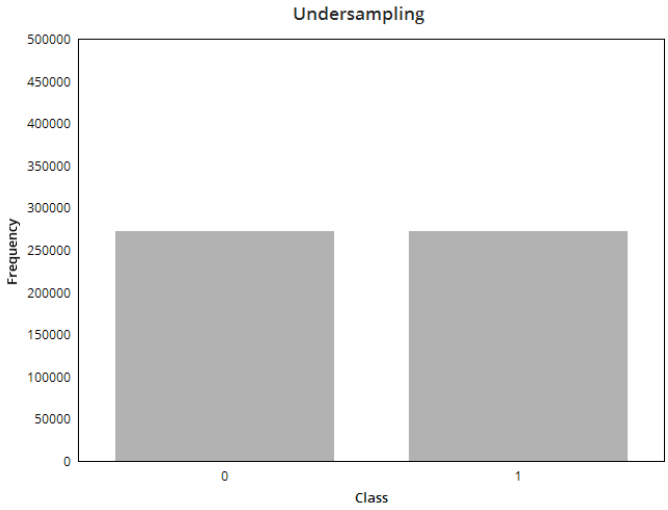

Figure 2: Balanced dataset using Undersampling

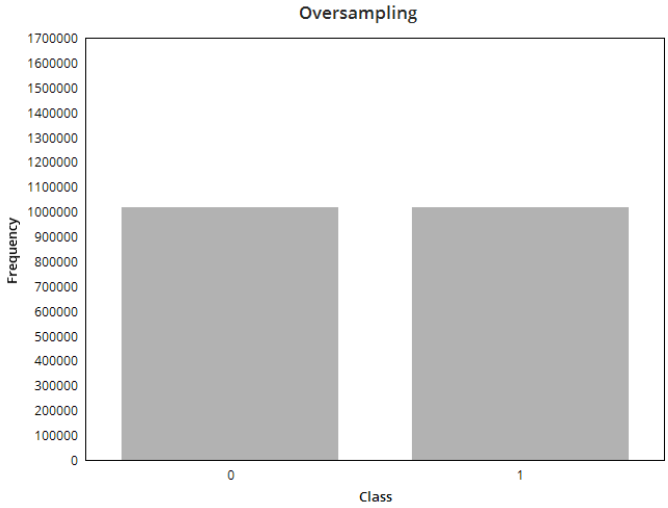

Figure 3: Balanced dataset using Oversampling

From figures 2 and 3 we choose data from oversampling as it increases the data points. Which increases the model's performance in predicting the flight delay.

3.2.1 Classifier After SMOTE

Generally, the F1 Score is preferred in the case of uneven class distribution as it takes both precision and recall into account, i.e indirectly considering false positive and false negative. But according to the problem statement false-negative is more significant than false-positive in predicting the flight delay that is because classifying delayed flights as non-delayed causes more damage than vice-versa. Thus, recall score has more significance than F1 score. Even though XG Boost has high F1 score but due to low recall score it is not considered as our ideal classifier.

CLASSIFIER	*CLASS*	*PRECISION*	*RECALL*	*F1-SCORE*	*ACCURACY*
Decision Tree	0	0.92	0.91	0.92	0.87
	1	0.67	0.70	0.69	
XG Boost	0	0.92	0.98	0.95	0.92
	1	0.89	0.69	0.78	
Random Forest	0	0.93	0.95	0.94	0.91
	1	0.81	0.74	0.77	
Logistic Regression	0	0.94	0.91	0.92	0.87
	1	0.68	0.70	0.69	

Table 5: Classifier Performance After SMOTE

So, we consider the classifier which shows a better recall score instead of the F1 Score. Random Forest classifier has a better performance compared to other classifiers with an increase in recall score after SMOTE.

The results of the test set as predicted by the random forest classifier using the SMOTE model are pipelined to the regressor module for prediction of flight delay minutes.

4 Regression

Using regression models, the arrival delay is predicted for the flights that have been classified as delayed by the classifier. The delayed flights are used to train the regressor. The features which was used in the classifier has been used to train the regressor as well. But the only difference is that the unsampled dataset is used to train the regressor as sampling is only used to balance the bias between two classes for the classifer. And using sampled dataset is like training the model with misleading data points which affects the prediction. So, unsampled dataset is used to train the regressor model.

4.1 Regression Metrics

The following metrics are used to evaluate the regressors:

where Y-Value, N-Total number of points, $\overline{Y} - Mean\ value\ of\ Y, \hat{Y} - Predicted\ value\ of\ Y$

- **Mean Absolute Error** is the absolute difference between the actual or true values and the values that are predicted.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}|$$

- **Mean Square Error** is calculated by taking the average of the square of the difference between the original and predicted values of the data.

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

- **Root Mean Square Error** 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.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}$$

- **R Squared** value lies between 0 and 1 where 0 indicates that this model doesn't fit the given data and 1 indicates that the model fits perfectly to the dataset provided.

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

The Regressors considered are:

- Gradient Boosting
- XG Boost
- Random Forest
- Decision Tree

4.1.1 Regressor Performance

REGRESSOR	***RMSE***	***MAE***	R^2***-SCORE***
Gradient Boosting	16.98	11.78	0.95
XG Boost	16.42	11.45	0.95
Random Forest	17.60	12.52	0.94
Decision Tree	23.47	16.14	0.90

Table 6: Regressor Performance After SMOTE

R^2 Score shows how close the data is fitted to the regressor, higher the R^2 value better the model fitted. A Low MAE value indicates better performance and lower the RMSE value better the fit. Due to the high $R2$ Score, low MAE value, and low RMSE values from Table 6, the regressor with the best performance is XG Boost.

4.2 Regression Analysis

The arrival delay for the flights classified as delayed was between 0 to 2000 minutes. From Table 7, it is clear that most of the flights had a delay between 15 - 100 minutes. The performance of XG Boost regressor in these ranges is given in Table 7.

The Figure 4 represents the distribution of the frequency of the flights delay and by minutes.

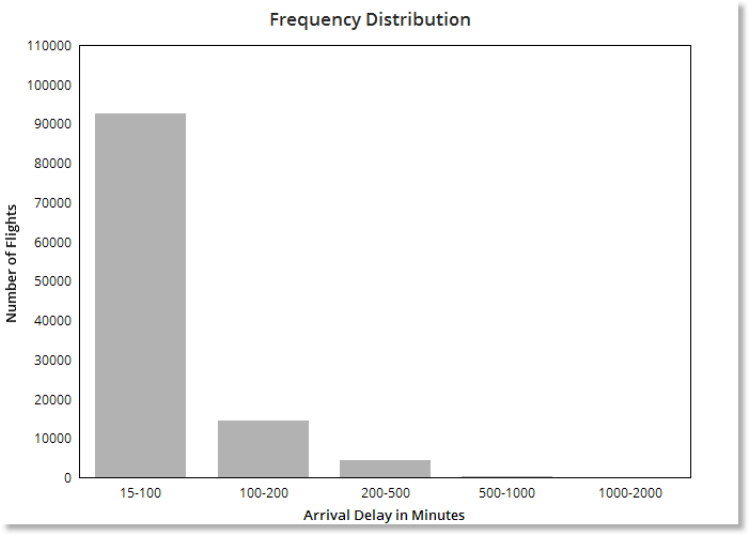

Figure 4: Frequency Distribution of flights

RMSE values tell us how close the predicted data is to the original data. Low MAE value indicates better performance by the regressor. From the table 8, The RMSE is minimum for the range 15-100 but that error is significant considering the magnitude. Even though the the RMSE and MAE is high for the range 1000-2000 while considering its magnitude the error is insignificant. So, the model performed well in the range 1000-2000 even though the model didn't have enough data points to train.

ArrivalDelayMinutes	No. of Flights	RMSE	MAE
15-100	92463	13.56	10.14
100-200	14247	25.97	17.41
200-500	4268	28.65	18.25
500-1000	352	22.24	16.05
1000-2000	58	26.65	19.76

Table 7: Range Wise Regressor Scores

4.3 Pipeline

The Figure 5 shows the structure of the model and Table 8 shows the final score of the model after the pipeline.

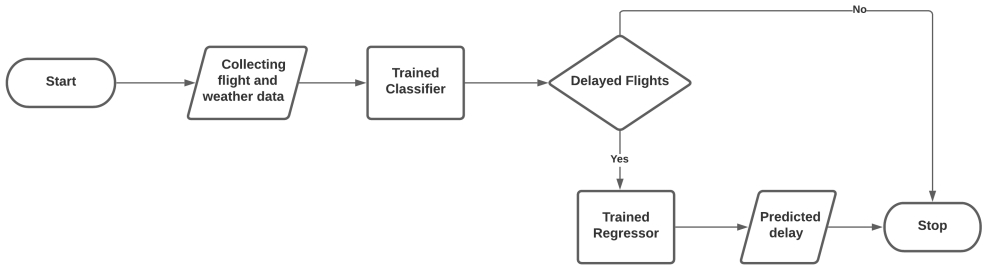

Figure 5: Model Structure

METRIC	***SCORE***
RMSE	18.56
MAE	14.15
R^2-SCORE	0.94

Table 8: Pipeline Performance

Random Forest Classifier and XG Boost regressor were chosen to build the pipeline and to produce the better score as shown in the Table 8.

5 Conclusion

The main objective of the project(to predict the arrival delay) is successfully completed by building a two-stage model. The classification model is used to classify the flights as delayed and not delayed. As the performance of the classifier is observed to be low, sampling techniques(SMOTE) where used to balance the class difference between the non-delayed flights and delayed flights. After using SMOTE the recall value of delayed data points increased. Random forest classifier is chosen for the pipeline as it had the highest recall score. Regression models are used to predict the arrival delay minutes for those flights which are classified as delay by the classifier. XG Boost regressor is chosen for the pipeline model as it had a high R2 Score(0.95), low RMSE (16.42), and low MAE(11.45)values. The pipeline model with the selected classifier and regressor performed with better accuracy.