Real Time Flight Delay Prediction

Abstract

In this project, we aim to to develop a predictive machine learning engine that can accurately forecast flight delays. The dataset utilized for this study comprises a vast collection of flight records from 2016 to 2017 in the USA, encompassing critical factors such as departure and arrival times, weather conditions, and airport information for 15 airports given in Table 1.

Data preprocessing involves cleaning and merging both flight and weather data to ensure the data is suitable for analysis. Subsequently, we move onto the classification stage, where we train a machine learning model to identify the key factors responsible for flight delays. The classification results are then utilized to perform regression and make accurate predictions regarding the delay times.

The outcome of this project provides valuable insights into the factors that contribute to flight delays, thus helping airlines mitigate delays and reduce the associated costs.

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

Table 1: Airport codes

1 Introduction

The timely arrival of flights is an important aspect of air travel and has a significant impact on passengers and airlines alike. Flight delays can cause inconvenience to passengers, result in missed connections, and lead to additional expenses. For airlines, flight delays can lead to decreased operational efficiency and reduced customer satisfaction. Hence, forecasting flight delays and their impact has become a crucial aspect of aviation operations.

The project comprises of three distinct stages. The initial stage deals with data preprocessing, which involves merging the weather and flight data to create a comprehensive dataset. The second stage focuses on the development of a classification model that predicts whether a flight will be delayed or not. Finally, the last stage involves building a regression model that predicts the duration of the delay for flights that have been classified as delayed.

2 Dataset

As already mentioned in Table 1, The flight data and weather data used in this project includes information from 15 different airports in the United States. This comprehensive data set allows for a thorough analysis of flight performance across a diverse range of locations, offering a comprehensive view of flight performance in the country over the course of 2016 and 2017.

The flight data includes key information as given in Table 2 and further enriched with the actual and scheduled arrival times of the flights, along with a binary indicator of whether the arrival was delayed by 15 minutes or more, and the corresponding number of minutes of delay. This provides a comprehensive view of the factors that impact flight delays, allowing for accurate predictions and insights.

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

Table 2: Features from Flight data

The weather data, on the other hand, consists of weather information recorded at different airports in the USA during the same period as mentioned in Table 3.

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

Table 3: Features from Weather data

The two datasets were merged appropriately to ensure that each record in the flight data has corresponding weather information. The merged data was then used as the input for building a predictive machine learning engine. The merged dataset provides crucial information that can be leveraged to understand the relationship between flight delays and weather conditions, which is important for the model forecasting the on-time performance of flights and estimating the arrival delay period (in minutes) for delayed flights.

2.1 Data preprocessing

Data preprocessing step prepares the data for analysis and modeling. Various operations are performed on the dataset to prepare it for further use. The merged dataset contains several columns, but only a few of them are relevant for our analysis.

The time column contains the scheduled departure and arrival times of the flights, which are not necessary for predicting flight delays. Similarly, the airport column contains the name of the airports from which the flights originated or arrived, which may not be particularly relevant for determining the likelihood of a flight being delayed. Lastly, the date column specifies the day on which the flight was scheduled, which is already accounted for in the other time-related columns such as year, month, day, and day of the week. Therefore, dropping these columns can help to reduce the dimensionality of the dataset and focus on the features that are more relevant for our analysis.

Next, the OneHotEncoder is used to encode the categorical variables *Origin* and *DestAirportID* into numerical form so that they can be used in machine learning algorithms. The encoded data is then transformed into a dataframe, and the columns are renamed with the name of the features.

After encoding, the original dataset is concatenated with the encoded data and the columns *Origin* and *DestAirportID* are dropped. Lastly, any rows containing NaN, ∞ or -∞ values are removed. The features and labels are then created by dropping the *DepDel15* column from merged data and assigning it to respective variables.

2.1.1 Data Splitting

The features and labels are divided into training and testing sets to evaluate the performance of the model. We split the data typically 80% for training and 20% for testing as shown below in Figure 1. We will use training features and labels to train the machine learning model and testing features and labels to evaluate the performance of the model.

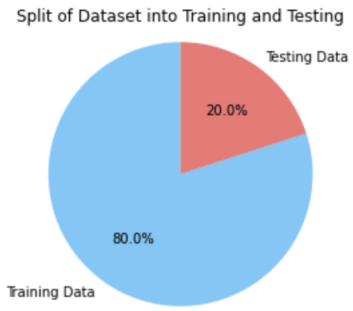

Figure 1: Data Split

3 Classification

The objective of a classification task is to predict the class label of an instance, based on its feature values. In the context of flight data, predicting the likelihood of a flight being delayed, based on a set of features such as weather conditions, airport, time of departure, etc. Some popular classifiers used in the analysis of flight data include **Decision Trees**, **Random Forests**, **Logistic Regression**, **ExtraTrees** and **XGBoost**.

Decision Trees are simple and intuitive models that work by recursively dividing the feature space into regions, based on the class label of the instances in those regions. They can handle both categorical and numerical features, and the prediction for a new instance is based on the path taken through the tree to reach the leaf node that corresponds to that instance.

Random Forests are an extension of Decision Trees, where multiple trees are grown, and the final prediction is made by averaging the predictions of all the trees. This approach reduces the variance in the predictions, and makes the model more robust to overfitting.

ExtraTrees are similar to Random Forests, but instead of growing a single tree at each node, multiple trees are grown, and the best split is selected based on the average improvement in the impurity of the classes.

XGBoost is a gradient boosting algorithm that trains an ensemble of decision trees, and updates the prediction for each instance based on the errors made by the previous trees. This process is repeated until the prediction error reaches a minimum. XGBoost is known for its high accuracy and speed, and is widely used in various machine learning tasks.

3.1 Evaluation Metrics

In classification problems, several evaluation metrics are used to assess the accuracy of the model and identify the strengths and weaknesses of the classifier. The most commonly used metrics are precision, recall, f1-score, and accuracy which use the number of true positives and false positives as part of their calculation.

True positives are instances where the model correctly predicts a positive class (i.e., the class of interest), while false positives refer to instances where the model incorrectly predicts a positive class when the true class is negative.

Precision is defined as the number of true positive predictions divided by the number of true positive predictions plus false positive predictions. It measures the proportion of positive predictions that are actually correct. Precision is

particularly important in cases where false positive predictions have severe consequences.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall is defined as the number of true positive predictions divided by the number of true positive predictions plus false negative predictions. It measures the proportion of actual positive instances that are correctly identified by the model. Recall is particularly important in cases where false negatives have significance.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

The F1 score is the harmonic mean of precision and recall, and it provides a single metric that summarizes the trade-off between precision and recall. A high F1 score indicates that the classifier has a good balance between precision and recall.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Accuracy is the proportion of correct predictions made by the model, and it measures the overall correctness of the model. However, accuracy can be misleading in cases where the class distribution is imbalanced, as the model may have a high accuracy even if it is not making the correct predictions for the minority class.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Observations}$$

The F1 score is particularly practical when the cost of false positives and false negatives is not significantly different, as it gives equal weight to both types of errors. This is important in flight delay prediction, where both types of errors can have significant consequences for airlines and passengers. Other metrics, such as accuracy and precision, do not take into account the importance of both false positives and false negatives and may provide misleading results. Therefore, in this project, we use the F1 score as the primary evaluation metric for our classifiers.

3.2 Performance

In the context of the project, **Class 0** refers to the situation where a flight is not delayed, and **Class 1** refers to the situation where a flight is delayed.

The results produced by the classifiers are shown in Table 4:

Classifiers	**Precision**		**Recall**		**F1-Score**		**Accuracy**
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
LogisticRegression	0.80	0.56	1.00	0.00	0.89	0.00	0.80
RandomForest	0.82	0.58	0.97	0.18	0.89	0.28	0.81
XGBoost	0.81	0.65	0.99	0.09	0.89	0.15	0.81
DecisionTree	0.83	0.32	0.82	0.34	0.82	0.33	0.72
ExtraTrees	0.82	0.54	0.97	0.16	0.89	0.25	0.80

Table 4: Classifier Performance

Based on the F1-Score, it can be seen that all the classifiers perform well in the classification task for Class 0 but not so well for Class 1. The Logistic Regression, Random Forest, XGBoost, Decision Tree, and Extra Trees classifiers have an F1-Score of 0.00, 0.28, 0.15, 0.33, and 0.25 for class 1 respectively. These results show that there is heavy data imbalance between the classes and the best classifier cannot be chosen without resampling the data.

3.3 Data Imbalance

Data imbalance is a common problem in the field of machine learning, where the distribution of classes in the target variable is unequal. This can lead to poor performance of machine learning models, as they are unable to learn the characteristics of the minority class correctly. In such cases, the imbalance can be addressed by either undersampling or oversampling the data. Figure 2 demonstrates that Class 0 represents roughly 80% of the total data, indicating a class imbalance in the dataset.

3.3.1 Undersampling

Undersampling involves reducing the size of the majority class to match the size of the minority class. RandomUndersampler is used for undersampling the dataset and the classifiers performed are as given in the below table:

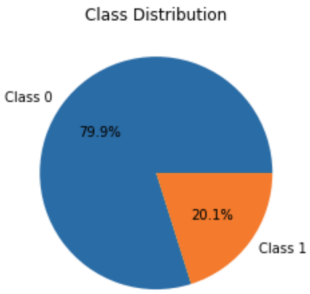

Figure 2: Data Imbalance

Classifiers	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
LogisticRegression	0.85	0.27	0.58	0.61	0.69	0.37	0.59
RandomForest	0.88	0.32	0.66	0.63	0.76	0.43	0.66
XGBoost	0.88	0.33	0.65	0.66	0.75	0.44	0.66
DecisionTree	0.85	0.26	0.58	0.59	0.69	0.36	0.58
ExtraTrees	0.87	0.32	0.66	0.62	0.75	0.42	0.65

Table 5: Classifier Performance (Undersampling)

Table 5 indicates that undersampling has significantly improved the performance of the classifiers on Class 1. The F1-score of all the classifiers improved as compared to the F1-score in the table with no undersampling. The highest accuracy score achieved was 0.66, which was recorded by three classifiers: RandomForest, XGB, and ExtraTrees. In addition, the recall of class 1, which is a critical metric for detecting delayed flights, was notably increased for RandomForest and XGB. These improvements indicate that undersampling was an effective strategy for addressing the class imbalance problem in the dataset, leading to better performance in predicting flight delays.

3.3.2 Oversampling

Oversampling involves increasing the size of the minority class by generating new samples. The most commonly used oversampling method is random oversampling, where new samples are generated by randomly selecting existing minority samples and adding them to the dataset. The classifiers performance are as given in the below table:

Classifiers	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
LogisticRegression	0.85	0.27	0.58	0.61	0.69	0.37	0.58
RandomForest	0.84	0.47	0.92	0.29	0.87	0.36	0.79
XGBoost	0.88	0.33	0.66	0.65	0.76	0.44	0.66
DecisionTree	0.83	0.32	0.82	0.33	0.82	0.32	0.72
ExtraTrees	0.82	0.55	0.97	0.15	0.89	0.24	0.80

Table 6: Classifier Performance (Oversampling)

Table 6 indicates that the oversampling technique improved the performance of the classifiers: Random Forest and Extra Trees classifiers had an increase in accuracy from 0.66 to 0.79 and 0.65 to 0.80, respectively. The Decision Tree classifier had a moderate improvement in its F1 score from 0.58 to 0.72. The Precision for Random Forest, XGB, Decision Tree, and Extra Trees remained relatively the same. The Recall for Random Forest, XGB, Decision Tree decreased while that of Extra Trees increased. Overall, the oversampling technique improved the performance of the classifiers, with Extra Trees showing the most improvement in accuracy.

4 Regression

Regression models are a fundamental part of predictive modeling and are used in a variety of applications, including predicting stock prices, housing prices, and sales forecasts. In the context of this flight data, regression models can be used to predict the arrival delay time, which can help airlines to better allocate resources and plan flight schedules.

The dataset used contains data of all the flights labelled to be delayed in the merged dataset. The independent variable, which do not include the *ArrDelayMinutes* column, and dependent variable, which is the *ArrDelayMinutes* column is split into a training set and a test set, with the test set making up 15% of the total data.

There are various regression algorithms available including **Linear Regression**, **Random Forest Regressor**, **ExtraTrees Regressor** and **XGBoost Regressor**. Each algorithm has its own strengths and weaknesses and the choice of the appropriate algorithm depends on the nature of the data and the problem being solved. In the context of flight data, we use them to predict arrival delay times and improve the efficiency of flight schedules.

4.1 Evaluation Metrics

In regression analysis, the metrics help in measuring the difference between the predicted values and the actual values. The three commonly used evaluation

metrics are Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score.

Mean Absolute Error (MAE) is the average of the absolute differences between the actual and predicted values. It is calculated as the sum of absolute differences between the actual and predicted values divided by the number of observations. MAE provides a measure of the magnitude of the errors in terms of the absolute values.

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

Mean Squared Error (MSE) is the average of the squared differences between the actual and predicted values. It is calculated as the sum of squared differences between the actual and predicted values divided by the number of observations. MSE provides a measure of the magnitude of the errors in terms of the squared values.

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

R^2 score, also known as the coefficient of determination, is a measure of how well the model fits the data. It ranges between 0 and 1, where 1 indicates that the model perfectly fits the data, and 0 indicates that the model is not a good fit for the data. R^2 score is calculated as the ratio of explained variance to the total variance.

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

where:

y_i = True value
$\hat{y}_i$ = Predicted value
$\bar{y}$ = Mean of the true values
n = Number of samples

A lower MAE indicates that the model is more accurate in predicting the target values. MSE is more sensitive to outliers and large errors, as the squared differences amplify their impact.

4.2 Performance

The performance of the Regressors on the test data is shown in the Table 7.

Regressors	**MAE**	**MSE**	**R^2 score**
LinearRegression	11.51	272.33	0.940
RandomForest	8.4	174.41	0.961
XGBoost	4.5	47.18	0.990
ExtraTrees	10.18	221.92	0.951

Table 7: Performance

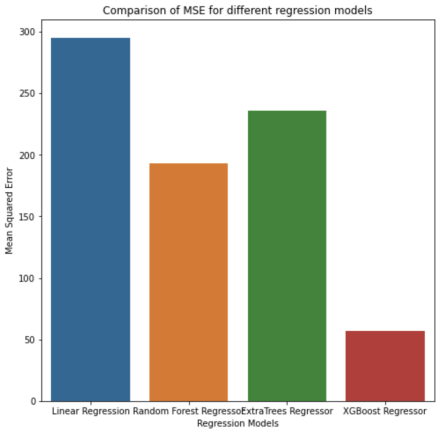

Figure 3: Comparison of MSE

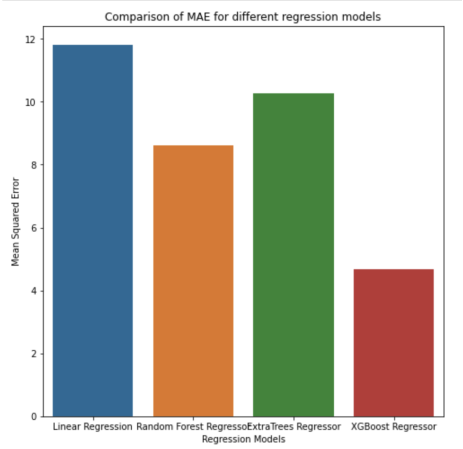

Figure 4: Comparison of MAE

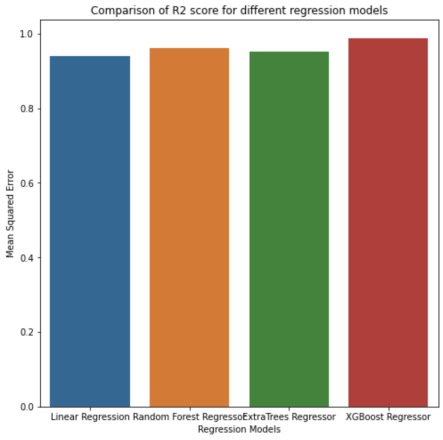

Figure 5: Comparison of R^2 score

When comparing the performance of different regression models, it is important to look at all three metrics and not just rely on one. A low MAE and MSE score, combined with a high R2 score, indicate a well-performing model. From the above table, it can be clearly observed that **XGB Regressor** has performed way better than all the other regressors.

5 Pipeline Architecture

A new dataset is used for pipeline analysis which comprises of data of the flights predicted to be delayed by the best classifier (ExtraTrees Classifier). The process is as shown as in the Figure 6 below.

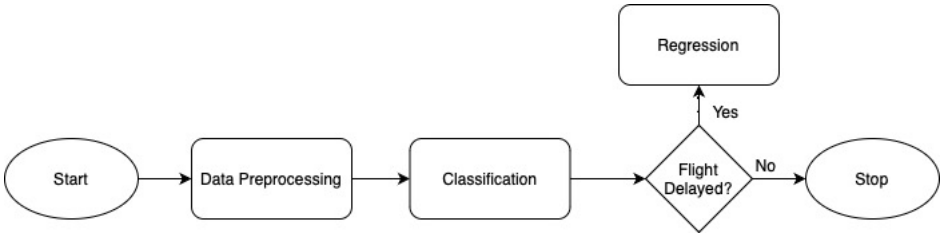

Figure 6: Pipeline Architecture

From Table 8, it can be observed that the performance of all the regressors has increased on the new dataset.

Regressors	**MAE**	**MSE**	R^2 **score**
LinearRegression	11.21	272.59	0.9535
RandomForest			
XGBoost	4.5	43.69	0.9925
ExtraTrees	8.61	167.76	0.9714

Table 8: Performance

From both the tables (Table 7, Table 8) above, it can be seen that XGBoost Regressor has clear edge over the other regressors based on the performance metrics.

6 Regression Analysis

Regression analysis helps to establish the relationship between a dependent variable and one or more independent variables. It is useful in predicting future outcomes, and understanding the effect of one variable on another. Here, the XGBoost Regressor is trained on different intervals on six different ranges (15-100, 100-200, 200-500, 500-1000, 1000-2000 and 2000+). The Delay Minutes density is shown in Figure 7.

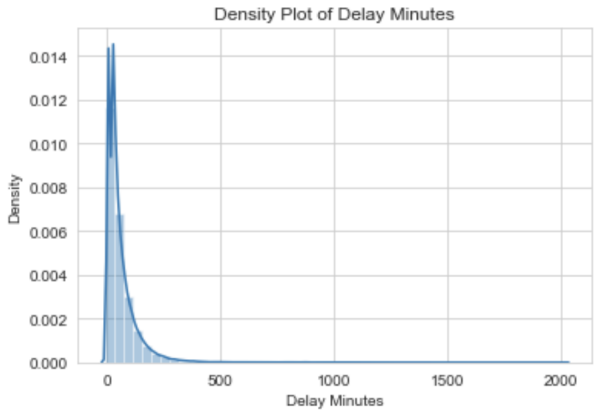

Figure 7: Delay Minutes Density

The scores produced by the regressor on each interval are tabulated in table below.

Intervals	**MAE**	**MSE**	R^2 **score**	**Datapoints**
15-100	1.82	6.20	0.987	189670
100-200	2.48	10.97	0.985	39963
200-500	2.12	7.92	0.998	11910
500-1000	0.27	0.153	0.999	917
1000-2000	0.001	0.0	0.999	151
2000+	0	0	0	1

Table 9: Regression Analysis

The performance of the regressor varies across the different duration intervals. The model's performance deteriorates slightly for flights delayed between 100-200 minutes as compared to the performance in 15-100 minutes. However, for flights delayed between 200-500 minutes, the regressor has an impressive performance despite the much smaller number of data points. The regressor performed even better for flights delayed between 500-1000 minutes and 1000-2000 minutes, as the model's performance is almost perfect. Finally, there is only one flight delayed for over 2000 minutes, and the regressor has a perfect performance on this flight. Overall, the regressor performs well across all duration intervals, with the best performance on longer delays, indicating that the model is useful for predicting flight delays across a wide range of scenarios.

7 Conclusion

The project aimed to predict flight delays and in the first stage of data preprocessing, the weather and flight data were merged, and split into a training set and a test

set. In the second stage, the classification model was developed to predict whether a flight would be delayed or not. The results showed that all classifiers performed well in classifying Class 0, but not as well for Class 1. The data imbalance problem between the classes was addressed by random undersampling and oversampling, which led to an improvement in the F1-score and accuracy for all classifiers. Random Forest and Extra Trees classifiers showed the most significant improvement in their performance. In the last stage of regression modeling, the performance of different regression models was compared using MAE, MSE, and R-squared metrics. The XGB Regressor was found to be the best performing model with the lowest MAE and MSE scores and the highest R-squared score.

In conclusion, The classifiers showed improved performance after resampling, and the XGB Regressor was identified as the best model for predicting the duration of delay. Future work could involve improving the classification model's performance for Class 1 by using other data imbalance techniques and exploring other regression models to improve the accuracy of delay duration prediction.