Predictive Machine Learning Study for Accurate Forecasting of Flight Delays

Abstract

This study aims to design and implement a predictive machine learning model to accurately forecast flight delays. An expansive data set is used which consists of a set of flight records from 2016 to 2017 in the USA, featuring essential elements such as departure and arrival times, weather conditions, and airport data from 15 airports given in Table 1. The methodology includes a robust preprocessing stage, wherein the flight and weather data are meticulously cleaned and merged to ensure their suitability for further analysis. The subsequent stage involves training a machine learning models to classify flights that are delayed and further predict the delay time using regressors. The results of this research offer significant insights into flight delay factors, providing airlines with potential strategies to mitigate such delays and minimize associated costs.

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

Table 1: Airport codes

1 Introduction

The punctuality of flight arrivals is a crucial component in the aviation industry, exerting substantial influence on both passengers' experience and airlines' operations. Therefore, the ability to forecast flight delays and understand their implications has become an indispensable element in the management of aviation operations.

This research is structured around three main stages. The first stage involves data preprocessing, where weather and flight data are merged into a single comprehensive dataset, ready for analysis. The subsequent stage is centered on constructing a classification model capable of predicting whether a flight is likely to be delayed. The concluding stage involves the creation of a regression model to estimate the duration of delay for flights flagged as delayed by the classification model.

2 Dataset

In this study, an extensive dataset comprising flight and weather data has been employed to analyze flight performance across the United States. The dataset encompasses information from 15 diverse airports throughout the country, as delineated in Table 1, and spans the years 2016 and 2017.

The flight dataset, enriched with features detailed in Table 2, incorporates salient attributes such as actual and scheduled arrival times. Moreover, it includes a binary indicator denoting whether the flight was delayed by 15 minutes or more, alongside the quantified duration of the delay in minutes. Such data allows for a comprehensive insight into the factors affecting flight delays.

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

Table 2: Features from Flight data

The weather data set used consists of meteorological records at the aforementioned airports during the same period. The key features of the weather dataset are outlined in Table 3. Meteorological conditions play a significant role in flight delays, making this dataset essential for understanding the correlations between weather and flight performance.

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 meticulously merged such that each flight record was coupled with corresponding weather information.

2.1 Data Preprocessing

Data Preprocessing step involves conducting a series of operations on the dataset to render it appropriate for subsequent use.

2.1.1 Feature Selection and Dimensionality Reduction

A primary objective of this project is to ascertain whether a flight is liable to be delayed, investigating the influence of weather and various other factors on this probability. In such predictive scenarios, precise departure and arrival times of the flight are typically unknown, thus rendering these columns extraneous for our purpose. Therefore, these columns have been eliminated from the dataset.

Moreover, the *Origin* and *Destination* columns were found to be redundant as they merely replicated the information contained in *OriginAirportID* and *DestAirportID*. Hence, these columns were also removed from our dataset. Further, the columns *Month*, *DayofMonth*, and *Year* were dismissed as the *FlightDate* column encapsulates this date-related information. Consequently, by dropping these columns, we have effectively reduced the dimensionality of the dataset.

Lastly, any rows containing NaN, ∞ or $-\infty$ values are removed. The features and labels are then created by dropping the *DepDel15* column from merged data and assigning it to respective variables.

2.1.2 Adding Additional Features

According to the *FlightDate* and *OriginAirportID* columns, a new feature *flight per day* was added which signifies the count of flights departing from a specific airport on a given day.

2.1.3 Categorical Feature Preprocessing using OneHotEncoding

Next, the OneHotEncoder is used to encode the categorical variables *FlightDate*, *OriginAirportID*, *DestAirportID*, *visibility*, *weatherCode* 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.

2.1.4 Numeric Feature Preprocessing using StandardScaler

The StandardScaler is a preprocessing utility widely utilized in machine learning tasks. It standardizes the features by removing the mean and scaling to unit variance. The transformation is defined as follows:

$$z = (x - u)/s \tag{1}$$

where x is the feature value, u is the mean of the feature values, and s is the standard deviation of the feature values.

By transforming the data in this manner, the StandardScaler mitigates the impact of outliers and ensures that no single feature dominates others due to differences in scale.

2.1.5 Addressing Data Imbalance

Data Imbalance refers to the situation where the distribution of classes within the target variable exhibits a marked inequality.

These methodologies broadly encompass two techniques: undersampling and oversampling. Undersampling involves reducing the quantity of the majority class instances, while oversampling pertains to increasing the instances of the minority class. The adoption of either strategy hinges on the specific characteristics of the dataset at hand.

Figure 1 demonstrates that Class 0 represents the majority class, indicating a class imbalance in the dataset.

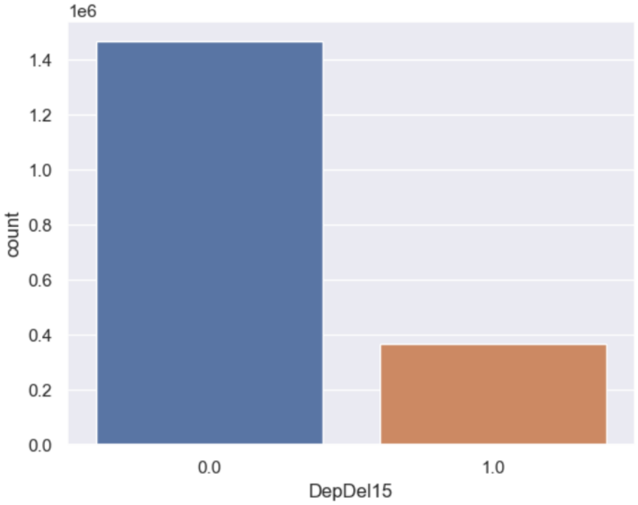

Figure 1: Data Imbalance

In this study, we address class imbalance through oversampling, particularly by increasing the instances of class 1.

The Synthetic Minority Oversampling Technique (SMOTE) is used for this purpose which works by selecting a minority class instance and computing its k-nearest neighbors.

Figure 2 provides a graphical illustration of the class distribution after the application of the Synthetic Minority Oversampling Technique (SMOTE).

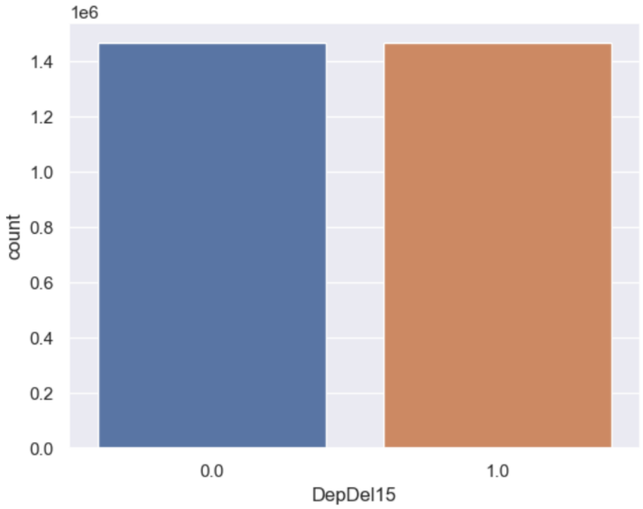

Figure 2: Classes Balanced by SMOTE

2.1.6 Data Splitting

The conventional data partitioning approach was adopted in this project, allocating approximately 80% of the data for model training and the remaining 20% for model evaluation.

The training set encompasses both the features and labels while the testing set is reserved for an unbiased evaluation of the trained model. This partitioning strategy ensures a robust assessment of the model's generalizability and its capability to handle unseen data, thus reflecting the model's real-world predictive performance.

3 Classification

The objective of a classification task is to accurately predict an instance's class label based on its feature values. In the context of our flight delay data, we aim to determine the probability of a flight being delayed considering various features such as weather conditions, airport of origin, departure time, among others. Several classifiers including **Decision Trees**, **Random Forest**, **Logistic Regression**, **ExtraTrees**, **XGBoost**, and **LightGBM** have been used in this project.

3.1 Evaluation Metrics for Classification Models

Four commonly used metrics in classification tasks are precision, recall, F1-score, and accuracy, each of which incorporates the concept of true positives and false positives in their computations.

True positives refer to instances where the model accurately predicts a positive class whereas false positives are instances where the model incorrectly identifies a negative class instance as positive.

Precision is calculated as the ratio of true positive predictions to the sum of true positive and false positive predictions. This metric quantifies the fraction of positive predictions that are indeed accurate.

$$\textbf{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2}$$

Recall, is the ratio of true positive predictions to the sum of true positive and false negative predictions. It measures the model's ability to correctly identify actual positive instances.

$$\textbf{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3}$$

The **F1 score**, computed as the harmonic mean of precision and recall, provides a comprehensive metric encapsulating the trade-off between these two measures.

$$\textbf{F1} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Accuracy, which represents the overall correctness of the model, is the proportion of correct predictions. However, accuracy can be misleading in situations with imbalanced class distributions, as a model may exhibit high accuracy while failing to predict correctly for the minority class.

$$\textbf{Accuracy} = \frac{True\ Positives + True\ Negatives}{Total\ Observations} \tag{5}$$

The **F1 score** proves especially useful when the costs associated with false positives and false negatives are not significantly different. By equally weighting these types of errors, it aligns well with the objectives of flight delay prediction, where both types of errors can cause substantial repercussions for airlines and passengers. Metrics like accuracy and precision may overlook the significance of both false positives and false negatives, potentially yielding misleading results. Hence, in this project, we adopt the F1 score as the primary evaluation metric for our classifiers

3.2 Performance of Classification Models

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

Classifiers	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Logistic Regression	0.99	0.74	0.85	0.86
Random Forest	0.93	0.82	0.87	0.87
XGBoost	0.96	0.81	0.87	0.88
Decision Tree	0.86	0.82	0.84	0.84
Extra Trees	0.93	0.85	0.90	0.93
LightGBM	0.97	0.75	0.85	0.87

Table 4: Classifier Performance

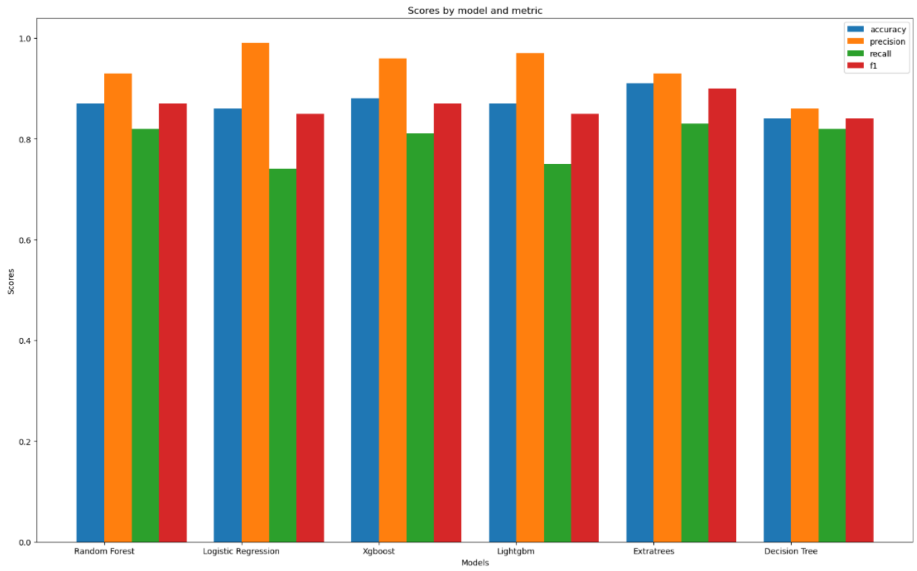

Figure 3: Bar Graphs of Classifier Performance

A range of machine learning models were assessed for their performance in predicting flight delays, as demonstrated in Table 4. The evaluation metrics displayed suggest that each model achieved respectable results. However, the **Extra Trees classifier** surpassed other models in terms of the **F1-Score**, an evaluation metric that balances both precision and recall.

4 Regression

Within the scope of this project, regression models are utilized to anticipate the departure delay durations. This predictive capability facilitates airlines in more effective resource allocation and flight scheduling, thereby enhancing their operational efficiency.

In this analytical framework, we specifically use the segment of our comprehensive dataset that includes only flights identified as delayed. Our model separates the dataset into independent variables, excluding the *DepDelayMinutes* column, and the dependent variable, represented by the *DepDelayMinutes* column. Subsequently, this data is split into a training set, constituting 80% of the total data, and a test set comprising the remaining 20%.

A diverse array of regression algorithms are at our disposal for this analysis, including **Linear Regression**, **Random Forest Regressor**, **DecisionTree Regressor**, **XGBoost Regressor**, and **LightGBM Regressor**.

4.1 Evaluation Metrics for Regression Models

Primarily, three metrics are frequently employed for regression, namely Mean Absolute Error (MAE), Mean Squared Error (MSE), and the $\mathrm{R}^{\mathbf{2}}$ score.

Mean Absolute Error (MAE) denotes the average magnitude of the absolute discrepancies between the predicted and actual values. It is computed as the total of these absolute differences, divided by the number of observations.

$$\mathbf{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \tag{6}$$

Mean Squared Error (MSE), on the other hand, is the average of the squared disparities between the predicted and actual values. It is derived by taking the sum of these squared differences and dividing it by the number of observations. By squaring the error values, MSE provides a measure of the magnitude of the errors, with a particular emphasis on larger deviations due to the squaring operation.

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{7}$$

The $\mathbf{R^2}$ **score**, often termed the coefficient of determination, gauges how well the model adheres to the data. This score can range from 0 to 1, where a score of 1 suggests a perfect fit of the model to the data, while a score of 0 indicates a poor fit. The $\mathbf{R^2}$ score is ascertained by taking the ratio of the explained variance to the total variance.

$$\mathbf{R^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2} \tag{8}$$

where:

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

4.2 Performance of Regression Models

The performance of the Regressors on the test data is shown in the Table 5.

Regressors	**MAE**	**MSE**	$\mathbf{R^2}$
Linear Regression	4.46	78.13	0.957
Random Forest	3.684	57.108	0.967
XGBoost	3.605	52.542	0.983
Decision Tree	3.745	58.959	0.966
LightGBM	3.673	56.292	0.968

Table 5: Regressor Performance

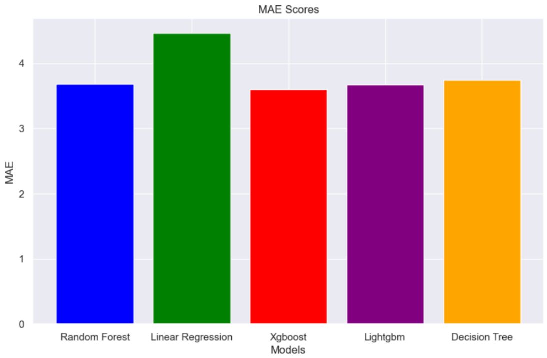

Figure 4: MAE of Regressors

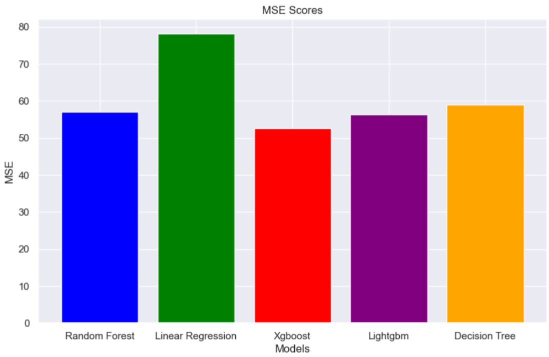

Figure 5: MSE of Regressors

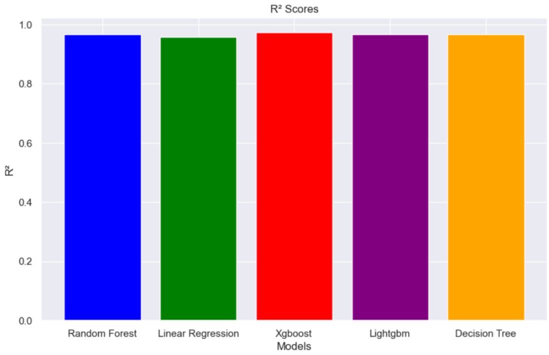

Figure 6: R^2 of Regressors

Upon examination of the comparison metrics outlined in Table 5, the **XGB Regressor** demonstrates superior performance relative to other evaluated models.

5 Pipeline Architecture

The dataset is used for pipeline analysis 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 7 below.

From Table 5 and Table 6 , we can infer that **XGBoost** delivers the best predictions compared to other models.

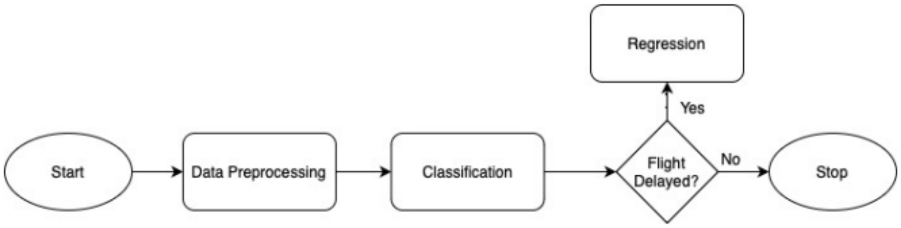

Figure 7: Pipeline Architecture

Regressors	**MAE**	**MSE**	**R^2**
Linear Regression	10.15	237.92	0.961
Random Forest	6.85	107.108	0.977
XGBoost	5.1	67.542	0.994
Decision Tree	7.89	187.78	0.969
LightGBM	8.54	155.463	0.972

Table 6: Regressor Performance on New Dataset created using Best Classifier

6 Regression Analysis

In Regression Analysis, **XGBoost Regressor** is trained across a series of intervals divided into six distinct ranges: 15-100, 100-200, 200-500, 500-1000, 1000-2000, and 2000+ in order to analyze its performance in these intervals.

Intervals	**MAE**	**MSE**	**R^2**	**Datapoints**
15-100	2.219	10.984	0.977	1157252
100-200	2.954	17.999	0.975	186754
200-500	1.485	5.375	0.998	55479
500-1000	0.481	0.452	0.999	4486
1000-2000	0.386	0.288	0.999	667
2000+	1.24	2.154	0.999	11

Table 7: Regressor Performance on Different Intervals of Flight Delays

As shown in Table 7, the efficiency of the regressor fluctuates across distinct delay intervals. A marginal deterioration is noticeable in the model's performance for flights experiencing delays between 100 and 200 minutes, compared to its efficacy in predicting delays between 15 and 100 minutes.

However, the regressor delivers commendable results for flights delayed between 200 and 500 minutes, despite the more limited quantity of data points in this range. The model exhibits even stronger performance in predicting delays spanning 500-1000 minutes and 1000-2000 minutes, approaching near-perfect predictions because the number of points in these intervals are relatively less and it is also easier for the model to predict longer delays as these are essentially just outliers of the dataset.

The model is particularly adept at predicting longer delays, signifying its wide-ranging utility for anticipating flight delays under diverse scenarios.

7 Conclusion

This project pursued the objective of predicting flight delays. It embarked on a three-staged approach. Initially, during the data preprocessing stage, the weather and flight data were consolidated and transformed via multiple statistical methods into a format suitable for further analysis, which included segregating the dataset into a training set and a testing set.

In the subsequent stage, a classification model was constructed to discern whether a flight would encounter a delay. Among several classifiers evaluated, the **Extra Trees** model surfaced as the most proficient.

Finally, the project shifted its focus onto regression modelling. Various regression models were evaluated based on their metrics. The **XGB Regressor** emerged superior with the minimal MAE and MSE scores, along with the maximal R-squared score. In the end, for the prediction of delay duration, the XGB Regressor was identified as the optimal model.