Airline Delay and Cancellation Analysis

Group Members: Himanshu Gandhi (1770138), Venkata Shreya Kala (1764875), Shubham Prasad Sahoo (1824661)

1 Abstract

Flight delays are on a steady rise, presenting significant financial challenges and fostering dissatisfaction among airline companies' customers. Addressing this pressing issue, the study employed supervised machine learning models to forecast flight delays. The prediction was based on a dataset containing flight information from the United States for one year-2015. Four algorithms - Logistic Regression, K-Nearest Neighbor, Random Forest, and XGBoost were trained and tested to complete the binary classification of flight delays. The evaluation of algorithms was fulfilled by comparing the values of four measures: accuracy, precision, recall, and f1-score. These measures were weighted to adjust the imbalance of the selected data set. The comparative analysis showed that the XGBoost algorithm has the best performance with an accuracy of 63% and KNN has the worst performance with an F1 Score of 58%. Tree-based ensemble classifiers generally have better performance than other base classifiers.

2 Introduction

As people increasingly choose to travel by air, the number of flights that fail to take off on time also increases. This growth aggravates the crowded situation at airports and causes financial difficulties within the airline industry. Air transportation delay indicates the lack of efficiency of the aviation system. It is a high cost to both airline companies and their passengers. The overall economic impact of airline delays is a critical aspect of the aviation industry, as highlighted in several studies. The annual economic impact of airline delays was estimated to be $31.2 billion in 2010 [1], while other recent studies estimated these expenditures to be $40.2 billion (the paper can be found on our GitHub repository too). Hence, forecasting flight delays holds the potential to enhance airline operations and elevate passenger satisfaction, thereby bringing a positive impact on the economy.

In this study, our main objective is to compare the performance of machine learning algorithms in predicting flight delays. Delving into the realm of Airline Delay prediction, we aim to harness the power of ML algorithms to distinguish and evaluate critical factors by uncovering patterns and relationships among them, which significantly impact the success of delay prediction. Our goal is to gauge the effectiveness of these algorithms in predicting and optimizing the success of airline prediction. The GitHub repository with all files associated with this project is given here - https://github.com/Hyshubham2504/Airline_Delay_and_Cancellation_Analysis.

3 Data

The dataset is sourced from Kaggle [2] and encompasses a vast repository of airline delay and cancellation data, offering a comprehensive perspective on the operational challenges faced by the aviation industry from 2009 to 2018.

For our study, we have specifically focused on data from the year 2015, providing insights into the performance and reliability of air transportation services during this period. The dataset consists of a total of 5,819,079 rows and 28 columns. This includes a broad spectrum covering temporal dynamics, Operational details, Geo-spatial information, and delay classifications. The dataset comprises Integer and categorical (Including Binary) data types, reflecting various essential information crucial for investigating Airline delays. This extensive dataset presents abundant opportunities for thorough examination and analysis, facilitating a detailed exploration of airline delays and cancellations. Table 1 and 2 is a brief description of some of the important features of the dataset.

Feature Name	**Data Type**	**Sample Values**	**Feature Description**
FL_DATE	Object	2015-01-01	The Date of the Flight
OP_CARRIER	Object	'NK', 'MQ', 'OO', 'EV', 'HA'	The Name of the Carrier
OP_CARRIER_FL_NUM	Int64	195, 197, 198	Flight Number of the Carrier
ORIGIN	Object	'MCO', 'LGA', 'FLL', 'IAH'	Origin Airport
DEST	Object	'FLL', 'MCO', 'LAS', 'ORD'	Destination airport
CRS_DEP_TIME	Int64	2147, 1050, 700	Schedule Departure Time (HHMM)
DEP_TIME	float64	2147, 1050, 700	Actual Departure Time (HHMM)
DEP_DELAY	float64	-4., 14., 12.	Difference in minutes between scheduled and actual departure time. Early departures show negative numbers.

Table 1: Important Features of the Dataset

Feature Name	**Data Type**	**Sample Values**	**Feature Description**
TAXI_OUT	float64	15., 20., 19., 8.	Taxi Out Time, in Minutes; The time elapsed between departure from the origin airport gate and wheels off.
TAXI_IN	float64	7., 9., 10., 4., 5.	Wheels down and arrival at the destination airport gate, in minutes
WHEELS_OFF	float64	2158., 1124., 731.	Wheels Off Time (local time) in HHMM
WHEELS_ON	float64	2158., 1124., 731.	Wheels On Time (local time) in HHMM
CRS_ARR_TIME	Int64	2250, 1404, 757	Scheduled Arrival time (HHMM)
ARR_TIME	float64	2245., 1403., 813.	Actual Arrival time (HHMM)
ARR_DELAY	float64	-5.0, -1.0, 16.0	Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.
CANCELLED	float64	0., 1.	Cancelled Flight Indicator (1=Yes); was the flight cancelled?
CANCELLATION_CODE	Object	'A', 'B', 'C', 'D'	Reason for cancellation (A = carrier, B = weather, C = NAS, D = security
DIVERTED	float64	0., 1.	Diverted Flight Indicator (1 = Yes)
CRS_ELAPSED_TIME	float64	63., 194., 57., 196.	Estimated Elapsed Time of Flight, in Minutes
ACTUAL_ELAPSED_TIME	float64	63., 194., 57., 196.	Elapsed Time of Flight, in Minutes
AIR_TIME	float64	40., 150., 32., 164.	Flight time in Minutes
DISTANCE	float64	177., 1076., 1222.	Distance between airports (miles)
CARRIER_DELAY	float64	1., 15., 127., 174.	Carrier Delay, in Minutes
WEATHER_DELAY	float64	31., 17., 24., 61.	Weather Delay, in Minutes
NAS_DELAY	float64	16., 18., 25., 19.	National Air System Delay, in Minutes
SECURITY_DELAY	float64	8., 21., 6., 14.	Security Delay, in Minutes
LATE_AIRCRAFT_DELAY	float64	8., 29., 21., 10.	Late Aircraft Delay, in Minutes

Table 2: Important Features of the Dataset

We have also included weather data using [3] to examine its relationship with flight delays and improve the forecasting precision of our models, given its substantial influence on aviation activities. The weather dataset contains daily weather data for various airports including details such as average temperature (tavg), minimum temperature (tmin), maximum temperature (tmax), precipitation (prcp), snowfall (snow), wind direction (wdir), wind speed (wspd), peak wind gust (wpgt), pressure (pres), and sunshine duration (tsun). Each record is associated with a specific date and airport code, allowing for detailed analysis of weather patterns and trends at different locations over the specified period.

The inclusion of weather data is essential for our analysis as it provides valuable insights into the environmental conditions experienced by airports. Understanding weather patterns and their impact on flight operations is crucial for predicting delays and optimizing airline performance. Table 3 is a brief description of some of the important features of the weather dataset.

Feature Name	**Data Type**	**Sample Values**	**Feature Description**
Date	DateTime	2015-01-01	The Date of the Weather
tavg	float64	1.7, 5.3, 4, 9.8	Average Temperature
tmin	float64	-2.7, 1.7, 1.1, 5.6	Minimum Temperature
tmax	float64	8.3, 9.4, 5.6, 19.4	Maximum Temperature
prcp	float64	0, 0, 12.7, 5.1	Precipitation
snow	float64	0, 30, 50, 20	Snow
wdir	float64	210, 309, 263, 189	Wind Direction
wspd	float64	8.5, 8, 8.3, 14.7	Wind Speed
wpgt	float64	140.4, 154.8, 118.8, 0	Peak Wind Gust
pres	float64	1021.6, 1034, 1036.5	Wind Pressure
tsun	float64	0, 1, 2, 3	Sunshine Duration
Airport Code	Object	ABE,ABI,DCA,ACV	Airport Unique Code

Table 3: Weather Dataset

4 Exploratory Data Analysis

During the Exploratory Data Analysis (EDA) phase, we thoroughly investigated the extensive data on airline delays and cancellations for 2015. This step forms the foundation of our analysis, providing valuable insights into the operational challenges and dynamics of the aviation industry during that period. Through a comprehensive exploration of the dataset, our objective is to discover hidden patterns, trends, and anomalies that could potentially impact the prediction of flight delays.

By examining the distribution, relationships, and characteristics of the data, we aim to gain a deeper understanding of the factors that influence airline delays. This will enable us to make informed decisions and develop effective models. At the end of our investigation, we have obtained informative visual representations from our data analysis which offer a visual depiction of the trends and patterns identified in the dataset.

Analyzing the flight count per month offers valuable insights into the seasonal trends and overall activity within the aviation industry. By examining the fluctuations in flight volumes over the year, we can determine patterns that shed light on factors such as travel demand, peak seasons, and operational dynamics. In Figure 1, the line chart depicts seasonal fluctuations in monthly flight counts, highlighting a notable peak in July indicative of the summer travel surge. A significant dip in February possibly reflects a post-holiday travel slowdown, while the decline in September and October suggests a shoulder season with lower travel demand. The slight increase in November may correlate with holiday travel, which then dips in December, potentially due to the holiday season's tail-end extending into the new year. These trends highlight the cyclical nature of air travel and its correlation with seasonal travel behaviours.

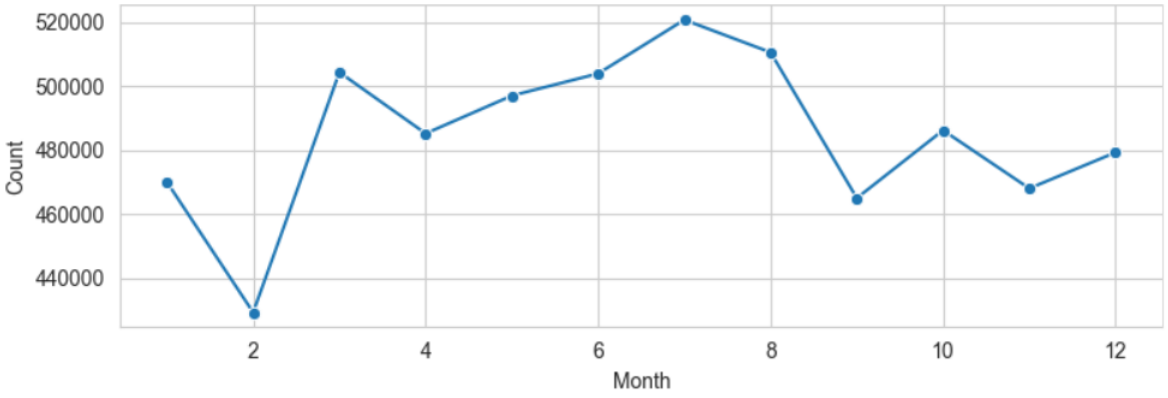

Figure 1: Flights Count Per Month

Analyzing the mean departure delay each month offers a detailed view of the temporal fluctuations in flight punctuality over the year. By examining these delays in various months, we can identify patterns, irregularities, and factors that influence the promptness of flights. In Figure 2, the average departure delay per month is plotted, with a noticeable peak

in June and December, possibly due to increased travel volumes during summer vacations and end-of-year holidays, respectively. The graph shows a plunge in September, which could be indicative of fewer operational issues or lower passenger traffic post-summer. This visual suggests that departure delays are influenced by seasonal highs and lows in travel demand, which are crucial for airlines to manage operational efficiency and customer satisfaction.

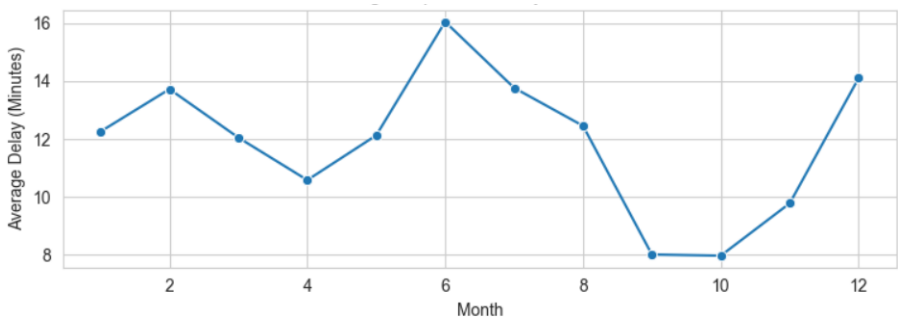

Figure 2: Average Departure Delay Per Month

Examining the mean departure delay across various days provides valuable insights into the weekly trends of flight punctuality. Gaining an understanding of how delays vary throughout the week can shed light on operational difficulties, passenger behaviours, and other factors that impact on-time performance. In Figure 3, the average departure delay fluctuates over the days of the week, with Monday starting at the highest point, suggesting a challenging start to the week for on-time departures. A notable drop on Wednesday indicates improved punctuality mid-week, followed by a spike on Friday, possibly due to increased travel at the week's end. The sharp decrease on Saturday could reflect lighter travel schedules, leading to fewer delays, before rising again on Sunday as travellers return home for the upcoming week.

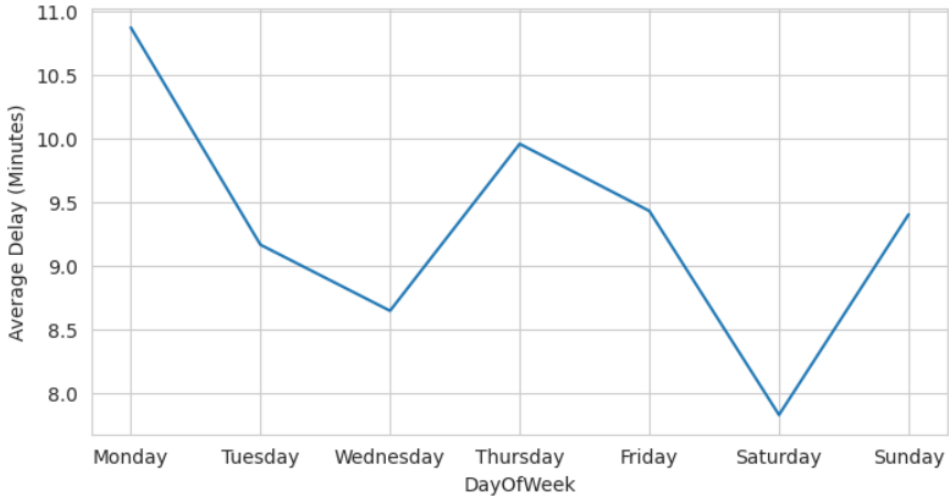

Figure 3: Departure Delay Over Day Of Week

Exploring the typical departure delay throughout different hours of the day offers insights into the timing trends in flight delays. By examining delays during specific time frames throughout the day, we can understand when delays are most common, identify operational challenges, and specify areas for improvement. In Figure 4, the line graph shows a general increase in average departure delay as the day progresses, with early morning hours exhibiting the shortest delays. The delays peak during the late afternoon and evening, possibly due to air traffic congestion and the cumulative effect of earlier delays. The downward trend after the peak suggests a nighttime improvement as traffic volume decreases, allowing for recovery in schedule adherence.

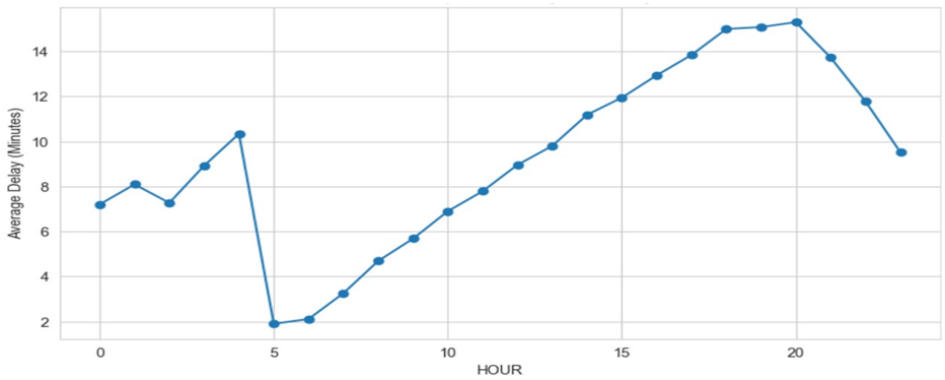

Figure 4: Average Departure Delay Over Hours of a Day

The efficiency of various carriers in terms of punctual departures can be inferred by visualizing the average departure delay per airline. By analyzing the differences in delays among airlines, we can estimate the overall dependability of airline services.

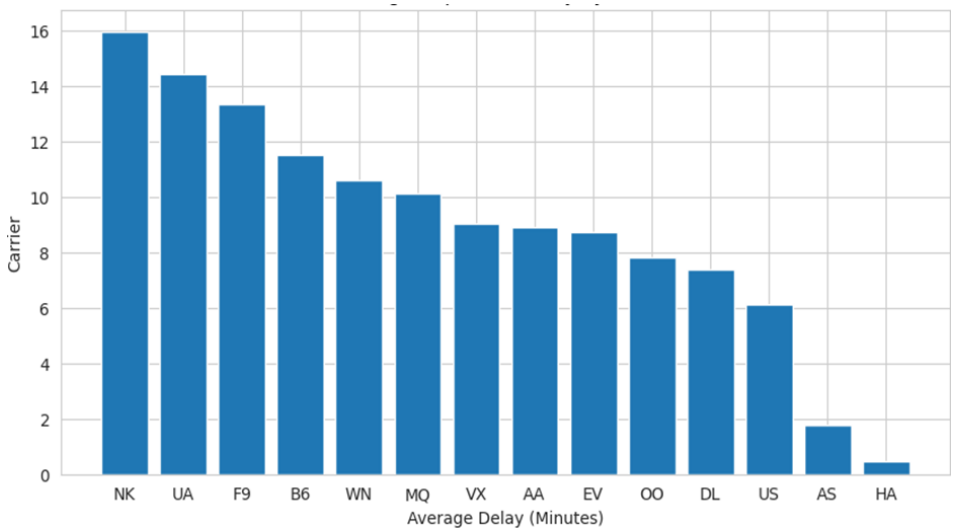

Figure 5: Average Departure Delay by Carrier

In Figure 5 above, the ranking of various airlines based on their average departure delay times is seen. Carriers such as NK (Spirit Airlines) and UA (United Airlines) are at the higher end of the spectrum, indicating longer average delays that could reflect operational challenges or larger network complexities. On the other end, AS (Alaska Airlines) and HA (Hawaiian Airlines) exhibit the shortest average delays, suggesting a higher punctuality in their departures. These differences may be influenced by the size of operations, hub locations, and efficiency of ground operations, with more efficient carriers likely benefiting from robust operational strategies and possibly less congested airports or more favourable weather conditions.

Studying the average departure delay at the top 20 airports provides valuable insights into the efficiency of air travel infrastructure at crucial hubs. Capturing the factors that contribute to delays at these major airports is essential to implementing focused strategies aimed at enhancing punctuality and improving passenger satisfaction.

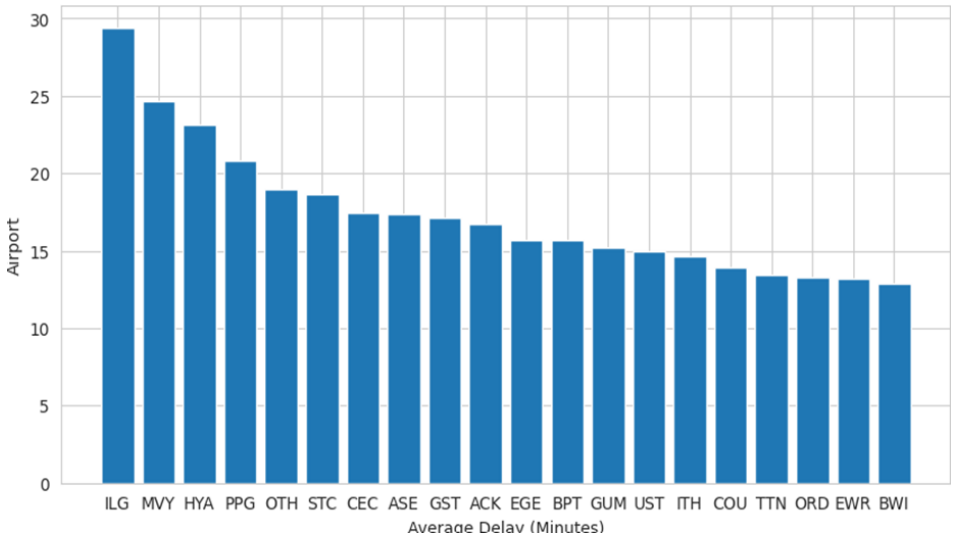

Figure 6: Top 20 Airports By Average Departure Delay

In Figure 6, ILG (Wilmington Airport) leads with the highest average departure delays among the top 20 airports, which may reflect operational constraints or regional weather challenges. Notably, major hubs like ORD (O'Hare International Airport) and EWR (Newark Liberty International Airport) are also listed, suggesting that high traffic volumes and complicated flight operations contribute to delays. BWI (Baltimore/Washington International Thurgood Marshall Airport) shows better performance relative to other large airports, hinting at more efficient handling of operations. This shows the need for improvements in traffic management and infrastructure enhancements at busy airports to minimize delay causes such as congestion, weather, or logistical inefficiencies.

Reasons behind flight cancellations give us insight into the operational challenges of airlines effectively. In Figure 7, weather-related issues are the leading cause of flight cancellations, reflecting the significant impact of adverse meteorological conditions on air travel safety and schedules. Carrier-related problems represent the second most common reason, which could encompass maintenance issues, crew shortages, or operational hiccups. NAS (National Airspace System)-related cancellations follow, indicating infrastructure and traffic management challenges. Security reasons account for the least number of cancellations, suggesting effective security protocols are in place. Understanding these factors is key for airlines to mitigate risks and enhance reliability for passengers.

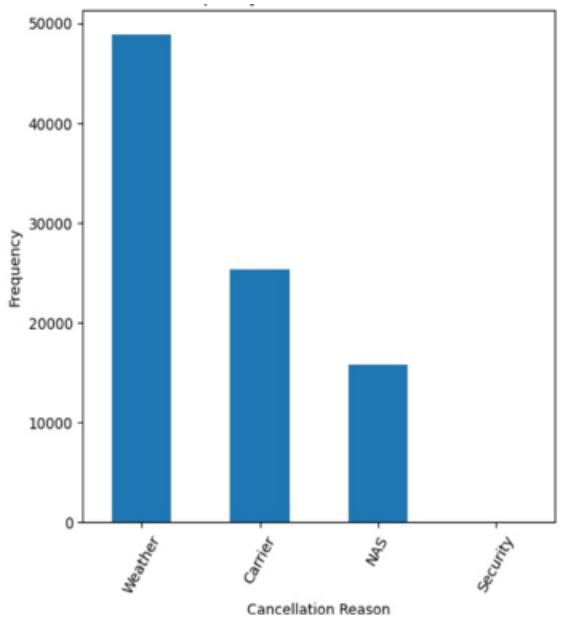

Figure 7: Flight Cancellation Reasons

Figure 8, WN (Southwest Airlines) shows the highest number of cancelled flights, which could be a reflection of its large volume of domestic flights and the impact of operational disruptions. DL (Delta Air Lines) and AA (American Airlines) also have a significant number of cancellations, indicating their extensive international and domestic networks which are subject to a variety of logistical challenges. On the other end of the spectrum, HA (Hawaiian Airlines) and VX (Virgin America, which has since merged with Alaska Airlines) show fewer cancellations, due to their smaller size or more contained route structures. These trends highlight the need for airlines to implement robust schedules and operational flexibility to minimize disruptions. Exploring how cancelled flights are distributed among various carriers offers valuable insights into the operations and reliability within the airline industry.

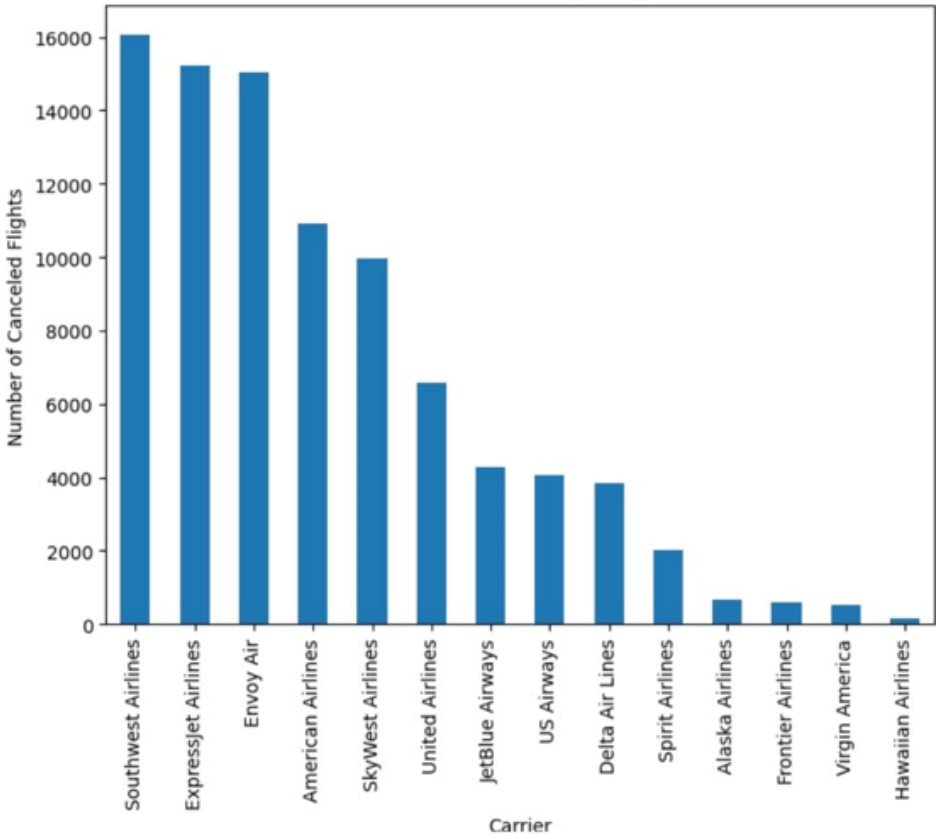

Figure 8: Number of Cancelled Flights By Carrier

The visualizations during EDA have produced significant findings and provide a comprehensive understanding of the patterns, distributions, and relationships within the data. Equipped with these valuable insights, we now move forward to the next stage of our project: Data Pre-processing. In this phase, we will enhance and organize the data to prepare it for model training, ensuring that our predictive algorithms are constructed using reliable data.

5 Methodology

5.1 Data Pre-processing and Analysis

Data Pre-processing is a crucial step in preparing the raw data for effective analysis. This section outlines the methods used to clean and transform the data into a usable format for predicting airline delays.

5.1.1 Null Value Handling

Temperature Average, Precipitation, and Wind Speed: These features (tavg, prcp, and wspd) are crucial for assessing the impact of weather conditions on flight delays. Null values in these fields were filled using the mean value calculated from the available data in the dataset. This approach helps maintain continuity in the weather data and ensures that our model can accurately evaluate the influence of weather conditions on flight delays.

Departure Delay: This feature (dep_delay) is critical as it forms the basis of our target variable for predicting flight delays. The dataset initially contained 86,153 records with null values in the 'dep_delay' field. Given the importance of accurate delay data for our analysis, we opted to remove these records entirely. This decision was made to ensure the integrity and reliability of our target variable, avoiding any skewness (asymmetry) or inaccuracies in the predictive modelling process.

These methods of handling null values were chosen to optimize the dataset for further analysis, ensuring that the data used in our model is both complete and representative. Each step was taken considering how it impacts the dataset's quality and the accuracy of our predictions regarding flight delays.

5.1.2 Feature Engineering

Feature engineering is a crucial step in preparing our dataset for predictive modelling. This process involves creating new features from existing data to improve the model's ability to determine patterns and make accurate predictions. We focused on extracting temporal dynamics, simplifying categorical data, and incorporating comprehensive weather information:

Extraction of Time Features: We extracted the 'day of week' and 'hour of day' from the Flight_date and CRS_DEP_TIME fields, respectively. These features are vital as they capture the variability in flight delays based on different times of the day and week. For instance, weekend flights or early morning flights might have different delay patterns compared to weekday or midday flights.

Weather Data Integration: Initially, we encountered over 300 unique airport identifiers ('Origin'), a scenario that could lead to the curse of dimensionality if one-hot encoded. To mitigate this and simplify the model, we substituted the 'Origin' airport data with comprehensive weather information corresponding to the geographical location of each flight's origin. This significantly enhances the model's ability to assess the impact of local weather conditions, which primarily influence flight delays. We gathered and scraped the weather data and integrated it with the main airline dataset as mentioned in Section 3. By leveraging weather information rather than specific airport locations, our model more effectively addresses delays linked to geographical and environmental factors.

Binary Target Transformation: The DEP_DELAY was transformed into a dichotomous/binary variable where delays less than or equal to zero minutes are marked as 0 (no delay) and delays greater than zero are marked as 1 (delayed).

Encoding of Categorical Variables: To use categorical data in machine learning models, which inherently require numerical input, we applied one-hot encoding to the 'Op_carrier', 'hour', and 'day of week' features. This transformation converts categorical variables into a format that can be provided to machine learning algorithms to improve model performance by treating each category as a separate binary feature.

These engineered features are expected to improve the predictive accuracy of our model by providing it with structured and relevant information that directly impacts flight delays. This process is vital for creating a robust model for predicting airline delays.

5.1.3 Feature Selection

Feature selection involves identifying the most relevant features for use in predictive modelling. This step improves model performance by reducing complexity and helps avoid overfitting and enhances computational efficiency. In our project, we adopted a systematic approach to drop redundant and irrelevant features, focusing on those that directly influence the prediction of airline delays:

Removal of Redundant Features: Flight-specific Operational Details: Features such as OP_CARRIER_FL_NUM, TAXI_OUT, WHEELS_OFF, WheelsOn, TAXI_IN, and Diverted were removed. These details, occurring post-departure, do not influence predictions regarding the likelihood of pre-departure delays.

Cancellation-related Features: Given that our model targets delay predictions rather than cancellations, CANCELLED and CANCELLATION_CODE were excluded from the analysis to prevent misleading the model with irrelevant data.

Destination and Arrival Timings: We also eliminated DEST (destination airport), CRS_ARR_TIME, ARR_TIME, and ARR_DELAY as these are focused on post-departure events and arrival metrics, which are not relevant for predicting departure delays.

Handling of Highly Sparse Features: We removed features categorizing the reasons for delays, such as carrier, weather, NAS (National Airspace System), security, and late aircraft, due to their high sparsity, with over 98 percent null values. The removal of these features was critical as their sparse nature could significantly hinder the model's ability to learn effectively and generalize from the training data.

This strategic feature selection process has refined our dataset, focusing solely on variables that directly impact the predictive accuracy regarding flight delays. Doing so has enhanced the model's efficiency and reliability, paving the way for more precise delay predictions.

5.1.4 Feature Scaling

Feature scaling is a crucial pre-processing step in data preparation, especially when dealing with variables that vary significantly in magnitude, units, and range. Inconsistent variable scales can lead to a biased or inefficient performance in many machine learning algorithms, particularly those that rely on distance calculations such as k-nearest neighbours (KNN) or gradient descent-based algorithms like logistic regression. To address these concerns, we implemented feature scaling using **'Standard Scaler'** to our dataset. This technique adjusts the features so that they have a mean of zero and a standard deviation of one. By doing so, each feature contributes equally to the distance computations, ensuring that no single feature dominates the model due to its scale. This is important in our context as features like temperature, wind speed, and flight times vary widely in their natural units and ranges.

Standardization helps in normalizing the data, providing a standardized level for all features to influence the algorithm's learning process effectively. This step is essential to ensure that our model behaves as expected and improves its ability to generalize from the training data to unseen data.

5.2 Modeling Techniques

Logistic Regression

Logistic Regression is a classic classification algorithm, that predicts binary outcomes by modelling the relationship between features and a binary target variable. It utilizes the logistic function to compute the log odds of the event happening and then applies it to obtain the predicted probability. The logistic function transforms the input features into a probability score for the positive class [4].

The logistic regression formula represents the probability that the target variable of a flight being delayed or not equals 1 given the independent variables or features that could impact delay, where each β represents the coefficient for a feature:

Logistic or Sigmoid Function:

$$p = \frac{1}{1 + e^{-\text{logit}(p)}}$$

Logit Function for the model:

$$\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Logistic regression learns the relationship between the input features (such as flight departure time, airline, weather conditions, etc.) and the target variable- whether a flight is delayed. It learns each feature's coefficients (or beta values), indicating their importance in predicting if the flight is delayed or not. The logistic function is then applied to compute the chance of a flight getting delayed based on these coefficients.

To implement logistic regression, we utilize the Logistic Regression function from sklearn.linear_model. The parameter used is C: The inverse of regularization strength, where smaller values indicate stronger regularization [5]. Additionally, the maximum number of iterations to run was set as 1000 and the solving algorithm was set to 'sag' or 'Stochastic Average Gradient' descent. It is used to minimize the loss function during model training by efficiently updating parameters by averaging gradients for each data point, making it suitable for large datasets [6]. This step was taken as the default solver did not converge even in 1000 iterations.

Log loss is used in logistic regression to penalize the incorrect classifications made, thus enabling the model to output corrected and improved probabilities [4].
Log Loss $= -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$

where, n is the total number of instances in the dataset,
y_i is the true label of the i-th instance
p_i is the predicted probability that the i-th instance belongs to class 1 or is delayed.

Random Forest

Random Forest is an ensemble learning technique that combines multiple decision trees to make predictions. A decision tree is a flowchart-like structure used in machine learning that makes decisions based on attributes in the dataset. Each tree is trained on a random subset of the data, and then their predictions are combined to determine the most frequently predicted class [7]. Each tree in the forest is trained and tested on a random subset of the data and features, which helps to reduce overfitting and improve generalization as given in Figure 9. So, it recognizes delayed flights by analyzing various features and their interactions. It has been implemented using the RandomForestClassifier function from sklearn.ensemble. The parameters used are the number of estimators- n_estimators: Number of trees in the forest. Other parameters are- max_depth: Maximum depth of each tree in the forest and min_samples_split: The minimum number of samples required to split an internal node [8]. It is essential to balance model complexity and performance by controlling these to prevent overfitting. The loss function in Random Forest by the min_samples_split parameter [8]. For classification tasks, Random Forest employs a voting mechanism where each tree's

prediction is considered, and the class with the majority of votes among the trees becomes the final prediction, enhancing accuracy and robustness.

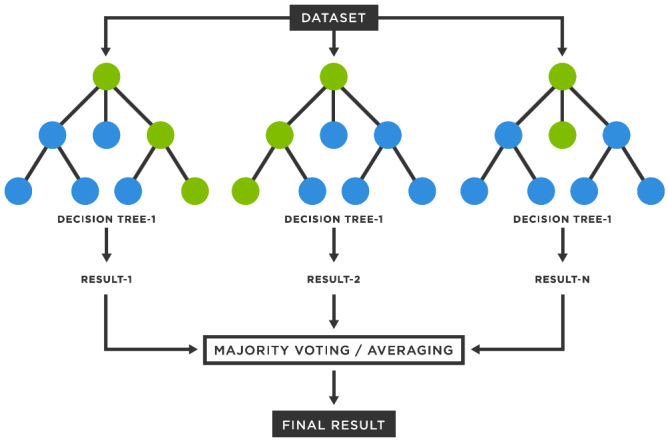

Figure 9: Random Forest [7]

K-Nearest Neighbours (KNN)

K-Nearest Neighbors (KNN) is a simple algorithm that emphasizes local patterns in the data. It classifies instances based on the majority class among their 'k' number of nearest neighbours in the feature space. The distance between a new observation and the training instances is calculated to determine the 'k' nearest neighbours [9]. The label observed most frequently among these neighbours is then assigned to the new observation. It does not assume the underlying data distribution, making it a non-parametric method. It has been imported from sklearn.neighbors as KNeighborsClassifier and the number of neighbours to consider for classification, denoted as 'k' denoted by n_neighbors was passed as a parameter. Higher values of 'k' lead to smoother decision boundaries but may lead to an oversimplified model. Along with it, the weight function to use for the instances was experimented with too. The parameter determines how the neighbouring points contribute to predictions, with "uniform" treating all neighbours equally and "distance" giving more weight to closer instances [10].

This model identifies similar instances of delayed flights based on their features and assigns them the same label as their nearest neighbours in the feature space similar to the working given in Figure 10.

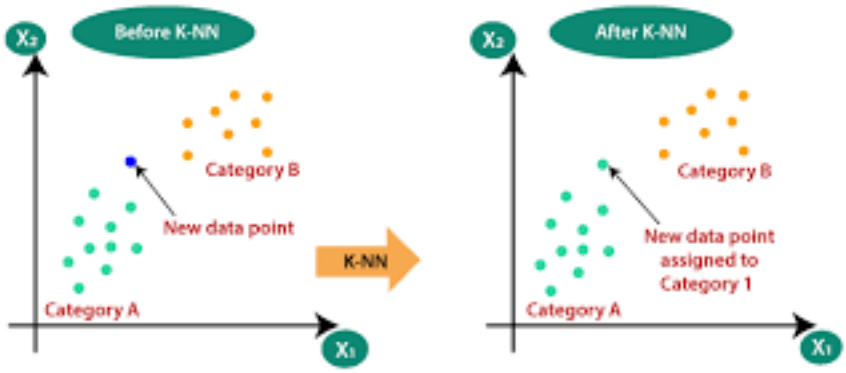

Figure 10: KNN [9]

XGBoost

XGBoost is an ensemble method that leverages boosting to combine weak learners, typically decision trees, to improve prediction accuracy or predictive power. It employs gradient boosting to construct sequential, interpretable trees by iteratively correcting errors and updating residuals [11]. The loss function and regularization term are optimized to enhance model performance. XGBoost's functionality is given in the XGBClassifier from the xgboost library. It is illustrated in Figure 11. Three parameters were passed to the model- n_estimators: Number of trees to build, max_depth: Maximum depth of each tree, learning_rate: Step size at each iteration, influencing the speed of learning and convergence [12].

The final prediction in XGBoost is obtained by summing up the predictions from all individual trees:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i)$$

This model works by iteratively improving predictions through sequential trees, enabling the identification of delayed flight instances based on various features in the dataset.

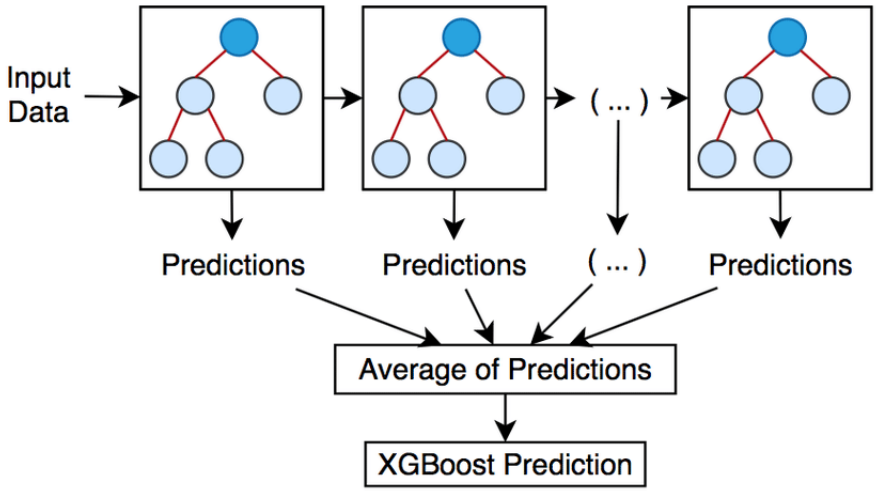

Figure 11: XGBOOST Model [13]

5.3 Model Improvements

To improve the performance and efficiency of the models, the following improvements were attempted.

Grid Search

Hyperparameters significantly impact model performance, and finding the optimal values in an iterative matter can be useful. GridSearchCV from sklearn.model_selection automates this by testing different hyperparameter combinations using cross-validation [14]. A grid with the different values for each hyperparameter or setting of a model is defined. The grid search function takes in the parameter and conducts an exhaustive search over the grid, evaluating each combination's performance using cross-validation. GridSearchCV trains and evaluates the model for each hyperparameter combination, allowing us to select the best-performing model based on the chosen evaluation metric- accuracy. By using GridSearchCV, we can optimize our model's hyperparameters, leading to better performance and generalization to unseen data.

The optimal values of the hyperparameters found are:

- **Logistic Regression**: 'C':
- **Random Forest**: 'max_depth': 13, 'min_samples_split': 3, 'n_estimators': 400
- **KNN**: 'n_neighbors': , 'weights':
- **XGBOOST**: 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 1000

Regularization

Regularization is used to prevent overfitting by penalizing the weights assigned to features or the cost function to deal with the complexity of the model. It has been applied in Logistic Regression to ensure they generalize well to unseen data. As the number of features increases, the prediction function becomes more complicated, increasing the risk of overfitting [15].

The regularization term is determined by a regularization parameter, which controls the strength of regularization. By adjusting this parameter, we can balance between fitting the training data well and having a simple model.

6 Results and Inferences

6.1 Evaluation Metrics

We generated classification reports and plotted AUC-ROC curves for each of the classification models. Learning curves were also plotted to demonstrate the model's performance as the training increased.

Classification report: The classification report presents an overview of a classification model's performance, summarizing key metrics including precision (the accuracy of positive predictions), recall (the capability to identify positive instances), F1 score (a balanced measure of precision and recall), and accuracy (correct classifications made).

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

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

The F1 score is the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{\text{Number of correctly predicted instances}}{\text{Total number of instances}}$$

The models were termed the best depending on the highest accuracy achieved, followed by a better precision, recall, and F1 score and then by considering the lesser difference in the training and testing accuracy if the former is higher.

Receiver Operating Characteristics (ROC): A graphical representation of the trade-off between True Positive Rate and False Positive Rate [16]. A high true positive rate and a low false positive rate indicate a well-performing model. The true Positive Rate is given by the

proportion of correctly identified actual positives while the false positive rate indicates the proportion of actual negatives that are falsely identified as positives [16].

Area Under the Curve (AUC): Area under the ROC curve. An AUC score close to 1 indicates a good classifier, while a score close to 0 indicates a good classifier in reverse, and a score close to 0.5 suggests a poor classifier (or a random guess).

$$\text{AUC} = \frac{\sum_{x \in AN} \sum_{y \in AP} 1_{f(y)>f(x)}}{|AP| \times |AN|}$$

Learning Curve: Learning curves provide valuable insights into the training of a model by plotting the change in a performance metric over time or with the number of steps. These curves are representations of the learning process, with the x-axis representing time or progress, and the y-axis representing the metric. These curves help in detecting issues and optimizing prediction performance [17].

6.2 Results and Analysis

Logistic Regression:

For logistic regression, Figure 12 below, the AUROC (Area Under the Receiver Operating Characteristic curve) is 0.66, indicating moderate predictive performance. In the learning curve analysis, both the training score and cross-validation score initially increase in parallel, and then converge at a point, suggesting optimal model complexity for the given dataset, striking a balance between bias and variance. In short, the model can generalize well to unseen data.

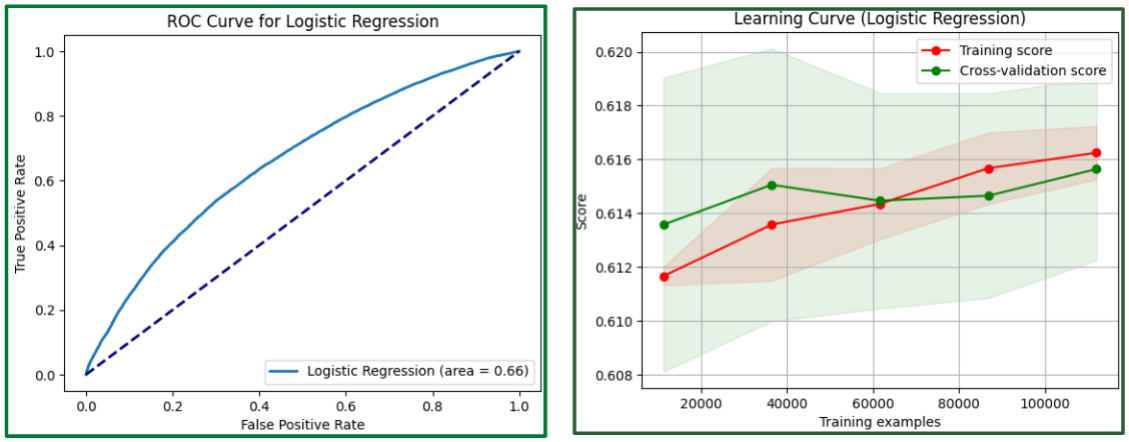

Figure 12: Logistic Regression: ROC Curve and Learning Curve

KNN:

For KNN, Figure 13 below, the AUROC is 0.62, indicating fair predictive performance. In the learning curve analysis, we observe that both the training score and cross-validation score exhibit a similar trend and increase together with increasing training set size. However, the notable gap between the two curves suggests that the model might be suffering from overfitting or high variance. Overfitting occurs when a model captures noise or random fluctuations in the training data, leading to poor generalization of the model to unseen data. The gap between the training and cross-validation scores indicates that the model performs significantly better on the training data compared to the validation data, which is a common indicator of overfitting.

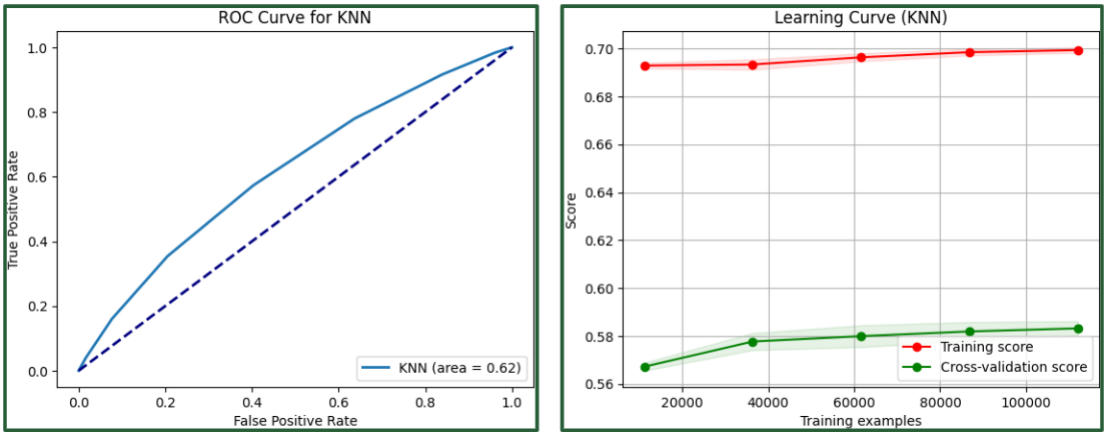

Figure 13: KNN: ROC Curve and Learning Curve

Random Forest:

For Random Forest, Figure 14 below, an AUROC of 67% indicates moderate predictive performance, suggesting that the model demonstrates some ability to discriminate between classes. AUROC values closer to 1 indicate better performance, so while 67% is not particularly high, it still suggests some level of predictive power.

From the learning curve, we observe that both the training score and cross-validation score increase together as the training set size grows. The fact that these curves run nearly in parallel suggests that the model's performance remains consistent across different training set sizes. Additionally, the small gap of approximately 0.02 between the two curves indicates minimal overfitting or variance. This means that the model's performance on the training data closely aligns with its performance on unseen data, as measured by the cross-validation score.

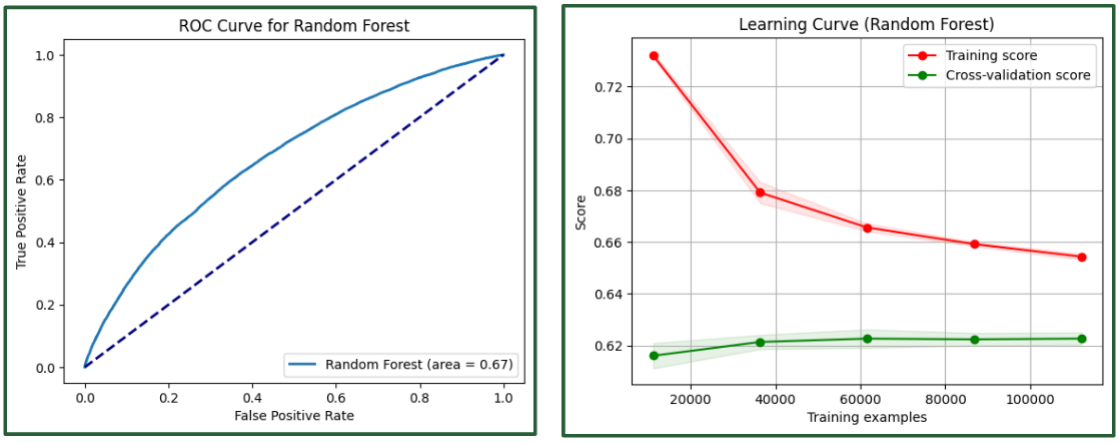

Figure 14: Random Forest: ROC Curve and Learning Curve

XGBOOST:

For the XGBoost model, Figure 15 below, achieving an AUROC of 68% represents the highest predictive performance among all the models we implemented. This indicates that the XGBoost model has the best ability to discriminate between classes compared to Logistic Regression, KNN, and Random Forest.

In the learning curve analysis, we observe a similar pattern to the Random Forest model, where both the training score and cross-validation score increase together as the training set size grows. The fact that these curves run nearly in parallel suggests that the model's performance remains consistent across different training set sizes. Additionally, the small gap of approximately 0.01 between the two curves indicates minimal overfitting or variance.

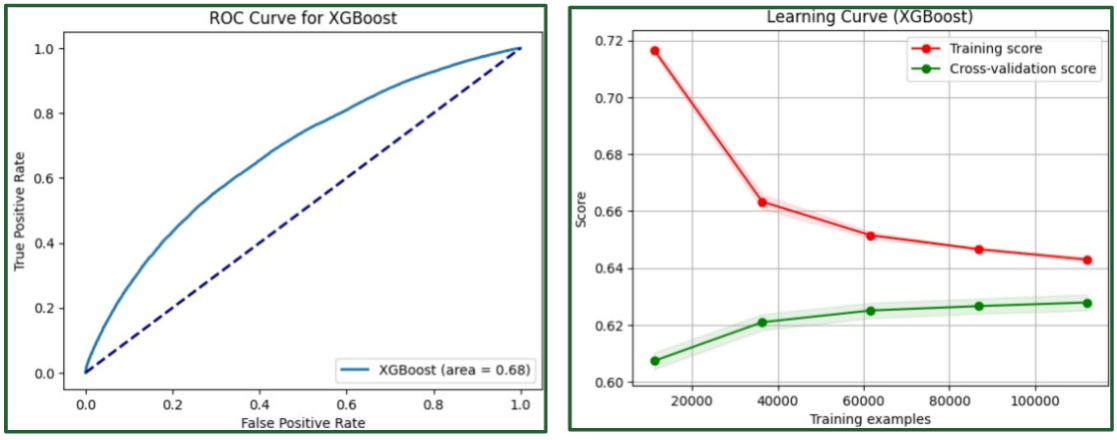

Figure 15: XGBOOST: ROC Curve and Learning Curve

Table 4 below, gives a concise view of all the metrics calculated to evaluate the model using the testing set. Overall, the XGBoost model demonstrates the most promising performance among the models tested with an accuracy of 68%, and the highest AUROC and minimal overfitting as indicated by the learning curve analysis. This suggests that the XGBoost model is well-suited for predicting flight delays based on the features provided in the dataset, ensuring effective flight delay management for improved operational efficiency and passenger satisfaction.

Model	Accuracy%	Precision%	Recall%	F1 Score%	AUC
Logistic Regression	62%	62%	62%	62%	66%
KNN	59%	59%	59%	58%	62%
Random Forest	62%	62%	62%	62%	67%
XGBoost	63%	63%	63%	63%	68%

Table 4: Model Comparison

Utilizing the built-in feature importance functionality, we generated a comprehensive plot that highlights the significance of each feature in contributing to the model's predictions. The feature importance plot, Figure 16 below from the XGBoost model revealed that the average temperature, wind speed, and precipitation were the most influential factors in determining flight delays. These features exhibited significant importance in predicting delays accurately. However, it's worth noting a notable decrease in feature importance scores from 1002 to 202, which can be attributed to the pre-processing techniques such as one-hot encoding applied to the features. This drop emphasizes the impact of the pre-processing steps in refining the features' significance and enhancing the model's overall performance.

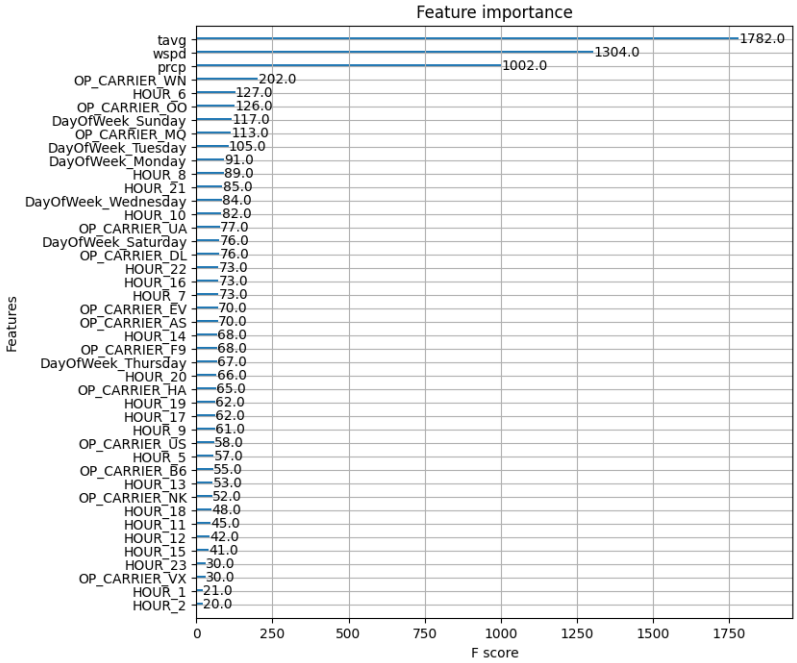

Figure 16: Feature Importances

6.3 Recommendations

The dataset offers valuable insights useful for strategic planning within the airline industry for them to optimize route planning and resource allocation. This enhances operational efficiency and contributes to customer satisfaction which are vital in the airline industry.

The model's predictions help airlines manage their resources better. From staffing to maintenance and fuel, these forecasts allow airlines to make wise decisions, saving costs and boosting profits. By matching resources with the demand, airlines can run their operations smoothly and achieve better financial results. We recommend the implementation of targeted strategies to address delays on routes prone to disruptions. By adjusting flight schedules and allocating resources strategically, airlines can mitigate the impact of delays. Also, investing in preventive maintenance programs to address underlying issues could be helpful. These measures help reduce delays and improve customer satisfaction.

7 Model Critique

The project began with the aim of employing regression models to predict flight delays which was a continuous variable rather than being binary. However, as we delved deeper into the data and its analysis, we encountered several limitations and challenges that reformed our approach.

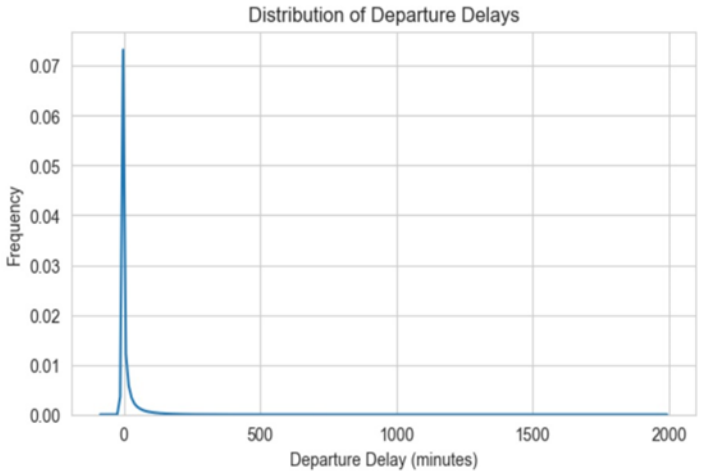

Figure 17: Skewness of the departure delay

One significant problem was we noticed the inherent skewness or asymmetry in the distribution of departure delay data. This skewed distribution posed a challenge for our regression models, as they struggled to capture the variance of the data accurately because they require normally distributed or at least symmetric data. Figure 17 depicts the right-skewness or asymmetry observed with the continuous departure delay variable.

In addition to challenges related to data distribution, the lack of critical features such as detailed air traffic information and explicit causes of the delays such as weather or NAS delays also severely limited our models' ability to make precise predictions. These essential features were initially included in the dataset but were later removed due to their high sparsity. With over 98% of the data containing null values, these features became redundant and were removed.

The results of the regression models were highly unsatisfactory indicating that only 3.5% of the variance in the data could be accounted by the predictions. Therefore, we transitioned towards classification models, which offered a better framework for addressing the data's complexities. The departure delay was converted to a binary variable indicating delay for a positive departure delay value and not otherwise. Despite this, the challenges continued, emphasizing the need for better solutions to overcome them.

The scale of the dataset presented logistical challenges during the training process. Training our models on such a large dataset required substantial computational resources and time. Although Google Colab was used for training the model with its processor and RAM, the training time of the models did not vary. Additionally, Google Colab's RAM would reach full capacity when attempting to use more training data, so the dataset was reduced. 200000 delayed and non-delayed flights each were sampled for training and testing the model further. Moreover, including hyperparameter tuning to optimize model performance added another layer of complexity, extending the training time even further. We tried to run a Support Vector Machine (SVM) model for fitting the data and classifying the flights as delayed or not. However, the training process did not finish despite spending more than 8 hours, as the Colab environment used for execution would timeout.

Future work includes implementing strategies to mitigate these limitations and challenges effectively. Exploring alternative approaches, such as feature engineering to address data skewness and using domain knowledge to impute missing features, could enhance the model's performance. Optimizing efficiency through parallel processing could also accelerate the training of the model.

8 Conclusion

Airline delays are a prevalent issue in the aviation industry, with extensive consequences for both airlines and passengers. Aside from the inconvenience experienced by travellers, these delays incur huge financial losses for airlines, in the form of increased operational costs, decreased productivity, and compensations. Additionally, delays tarnish the reputation of airlines, decreasing customer loyalty and satisfaction.

Thus, using machine learning techniques for predicting airline delays has gained popularity. By using historical flight data, weather forecasts, airport congestion patterns, and other relevant details, machine learning models can predict delays to aid airline companies in managing their operations, optimizing flight schedules to minimize disruptions, and allocating resources strategically.

However, the models' performance depends on the quality and comprehensibility of the data, so having access to flight schedules, aircraft performance metrics, weather data, air traffic information, and historical delay records along with their reasons would be highly valuable. The airline industry which is ever-active and dynamic, requires continuous monitoring of delays and update of models with more information with time.

By investing in data-driven strategies, airlines can navigate the complexities of modern air travel effectively and deliver a smooth and reliable experience for passengers.

References

[1] M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, B. Zou, R. Britto, D. Fearing, P. Swaroop, N. Uman, V. Vaze, and A. Voltes, *Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United States*, Oct. 2010.

[2] "Airline Delay and Cancellation Data, 2009 - 2018." [Online]. Available: https://www.kaggle.com/datasets/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018

[3] "Daily Data | Python Library | Meteostat Developers." [Online]. Available: https://dev.meteostat.net/python/daily.html#example

[4] "Logistic Regression: Definition, Types and Advantages." [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

[5] "sklearn.linear_model.LogisticRegression." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[6] "Comparing various online solvers in Scikit Learn - GeeksforGeeks." [Online]. Available: https://www.geeksforgeeks.org/comparing-various-online-solvers-in-scikit-learn/

[7] "Random Forest. Random Forest is an ensemble machine... | by Deniz Gunay | Medium." [Online]. Available: https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

[8] "sklearn.ensemble.RandomForestClassifier." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] "KNN Classifier from scratch. This article intends to help the reader... | by Shashank Parameswaran | Medium." [Online]. Available: https://medium.com/@shankyp1000/knn-classifier-from-scratch-326d3d4e894e

[10] "sklearn.neighbors.KNeighborsClassifier." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[11] "XGBoost Algorithm in Machine Learning." [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

[12] "XGBoost Parameters — xgboost 2.0.3 documentation." [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html

[13] "Figure 2.3: XGBoost model (Source: Self)." [Online]. Available: https://www.researchgate.net/figure/XGBoost-model-Source-Self_fig2_350874464

[14] "An Introduction to GridSearchCV | What is Grid Search | Great Learning." [Online]. Available: https://www.mygreatlearning.com/blog/gridsearchcv/

[15] "Understanding Regularization in Logistic Regression | by Jun M. | Towards Data Science." [Online]. Available: https://towardsdatascience.com/understanding-regularization-in-machine-learning-5a0369ac73b9

[16] "Guide to AUC ROC Curve in Machine Learning." [Online]. Available: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

[17] "What Is a Learning Curve in Machine Learning? | Baeldung on Computer Science," Jul. 2020. [Online]. Available: https://www.baeldung.com/cs/learning-curve-ml

9 Appendix

The files related to this project- proposal,data, code, presentation, and report are given in our GitHub repository- https://github.com/Hyshubham2504/Airline_Delay_and_Cancellation_Analysis/tree/main.

The **code** has been divided into three Jupyter notebooks as follows owing to extensive code, for better readability and organization:

Exploratory Data Analysis: https://github.com/Hyshubham2504/Airline_Delay_and_Cancellation_Analysis/blob/main/509_EDA.ipynb

Pre-processing: https://github.com/Hyshubham2504/Airline_Delay_and_Cancellation_Analysis/blob/main/Airline_Preprocessing.ipynb

Modelling: https://github.com/Hyshubham2504/Airline_Delay_and_Cancellation_Analysis/blob/main/MATH509_Modelling.ipynb