AIRLINE DELAY AND CANCELLATION ANALYSIS

Amina Sadiq(1867400), Nikhila Boosam(1855023)

April 9, 2025

UNIVERSITY OF ALBERTA

Abstract

Flight delays are steadily increasing, presenting financial challenges and dissatisfaction among airline customers. This study addresses the issue using supervised machine learning models to forecast flight delays based on a 2018 U.S. flight dataset. Four algorithms—Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and XGBoost—were trained for binary classification of delays. Model performance was evaluated using accuracy, precision, recall, and F1-score. XGBoost demonstrated the best performance with 63% accuracy, while KNN performed worst with an F1-score of 58%. Ensemble tree-based methods outperformed other classifiers. The findings suggest that machine learning, when carefully tuned and applied, can serve as a powerful tool in minimizing airline operational disruptions and enhancing passenger satisfaction. The GitHub repository with all files associated with this project is given here: https://github.com/Nikhila002/AIRLINE-DELAY-AND-CANCELLATION-ANALYSIS

https://github.com/amenasadiq7/MATH509

1. Introduction

In the high-stakes aviation industry, minimizing disruptions and ensuring timely operations are vital to both economic success and customer satisfaction. With increasing air traffic and complex logistical operations, flight delays have become a significant problem. This report seeks to address the challenge by analyzing data from 2018, identifying delay patterns, and proposing solutions using machine learning techniques.

Research Questions:

1. What are the main causes of flight delays and cancellations?
2. How do delays vary by airline, airport, and time of day?
3. What scheduling strategies could airlines adopt to reduce delays?
4. Can a predictive model anticipate delays and cancellations?
5. How do airline-specific operational strategies influence delay management?

2. Background and Problem Description

Flight delays disrupt not only passenger plans but also airline schedules and resource allocations. According to the U.S. Department of Transportation, delays cost airlines over $30 billion annually. Delays also result in missed connections, additional fuel consumption, and increased stress for ground and flight staff. This project aims to model and predict delays using features such as departure time, carrier, weather conditions, and airport congestion, enabling proactive mitigation strategies.

3. Data Collection

The dataset used in this project was sourced from Kaggle and originally compiled from the U.S. Bureau of Transportation Statistics. It contains 196,257 rows and 28 columns related to flight operations in 2018. Features range from temporal attributes (e.g., scheduled departure time) to geographic details (e.g., origin and destination airport), and operational metrics (e.g., taxi-out time, wheels-off time, weather delays).

Selected Key Features:

- CRS_DEP_TIME and DEP_TIME
- OP_CARRIER (Carrier Code)
- ORIGIN and DEST (Airport Codes)
- DEP_DELAY and ARR_DELAY
- CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY
- CANCELLED, CANCELLATION_CODE

These features capture the essential elements that can influence the likelihood and severity of a delay.

Categories	Feature Name	Sample Values	Feature Description
DATE & TIME	FL_DATE	2018-01-01	The Date of the Flight
	CRS_DEP_TIME	2147, 1050, 700	Schedule Departure Time (HHMM)
	DEP_TIME	2147, 1050, 700	Actual Departure Time (HHMM)
	CRS_ARR_TIME	2250, 1404, 757	Scheduled Arrival time (HHMM)
	ARR_TIME	2245., 1403., 813.	Actual Arrival time (HHMM)
FLIGHT DETAILS	OP_CARRIER	'NK', 'MQ', 'OO', 'EV', 'HA'	The Name of the Carrier
	OP_CARRIER_FL_NUM	195, 197, 198	Flight Number of the Carrier
	ORIGIN	'MCO', 'LGA', 'FLL', 'IAH'	Origin Airport
	DEST	'FLL', 'MCO', 'LAS', 'ORD'	Destination airport
	DISTANCE	177., 1076., 1222.	Distance between airports (miles)
TIME METRICS	TAXI_OUT	15., 20., 19., 8.	Taxi Out Time, in Minutes; The time elapsed between departure from the origin airport gate and wheels off.
	WHEELS_OFF	2158., 1124., 731.	Wheels Off Time (local time) in HHMM
	WHEELS_ON	2158., 1124., 731.	Wheels On Time (local time) in HHMM
	TAXI_IN	7., 9., 10., 4., 5.	Wheels down and arrival at the destination airport gate, in minutes
	CRS_ELAPSED_TIME	63., 194., 57., 196.	Estimated Elapsed Time of Flight, in Minutes
	ACTUAL_ELAPSED_TIME	63., 194., 57., 196.	Elapsed Time of Flight, in Minutes
	AIR_TIME	40., 150., 32., 164.	Flight time in Minutes
DELAY INFORMATION	DEP_DELAY	-4., 14., 12.	Difference in minutes between scheduled and actual departure time. Early departures show negative numbers.
	ARR_DELAY	-5.0, -1.0, 16.0	Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.
	CANCELLED	0., 1.	Cancelled Flight Indicator (1=Yes); was the flight cancelled?
	CANCELLATION_CODE	'A', 'B', 'C', 'D'	Reason for cancellation (A = carrier, B = weather, C = NAS, D = security
	DIVERTED	0., 1.	Diverted Flight Indicator (1 = Yes)
	CARRIER_DELAY	1., 15., 127., 174.	Carrier Delay, in Minutes
	WEATHER_DELAY	31., 17., 24., 61.	Weather Delay, in Minutes
	NAS_DELAY	16., 18., 25., 19.	National Air System Delay, in Minutes
	SECURITY_DELAY	8., 21., 6., 14.	Security Delay, in Minutes
	LATE_AIRCRAFT_DELAY	8., 29., 21., 10.	Late Aircraft Delay, in Minutes

Table 1: Sample Dataset

4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis plays a crucial role in understanding the structure and quality of a dataset before applying any modeling techniques. It helps uncover the underlying patterns, detect anomalies or missing values, identify relationships between variables, and highlight trends that may otherwise go unnoticed. Performing EDA allows me to ask better questions, form data-driven hypotheses, and choose the right methods for further analysis. Since the dataset focuses on airline delays and cancellations in 2018, this step is especially important to make sense of the complex interplay between operational, temporal, and geographical features that potentially influence flight performance. By visually and statistically exploring the dataset, we gain valuable context that helps inform both interpretation and decision-making in later stages of the project.

As a first step in the EDA process,wel generate a correlation matrix to examine how numerical variables in the dataset relate to one another. This matrix helps identify which features are strongly or weakly correlated, which in turn can provide early insight into which variables might play a significant role in delay prediction. It also serves as a useful guide for detecting multicollinearity, ensuring that features used in future modeling are informative and not redundant. The correlation matrix offers a clear visual summary of linear relationships and is a foundational step in guiding further analysis.

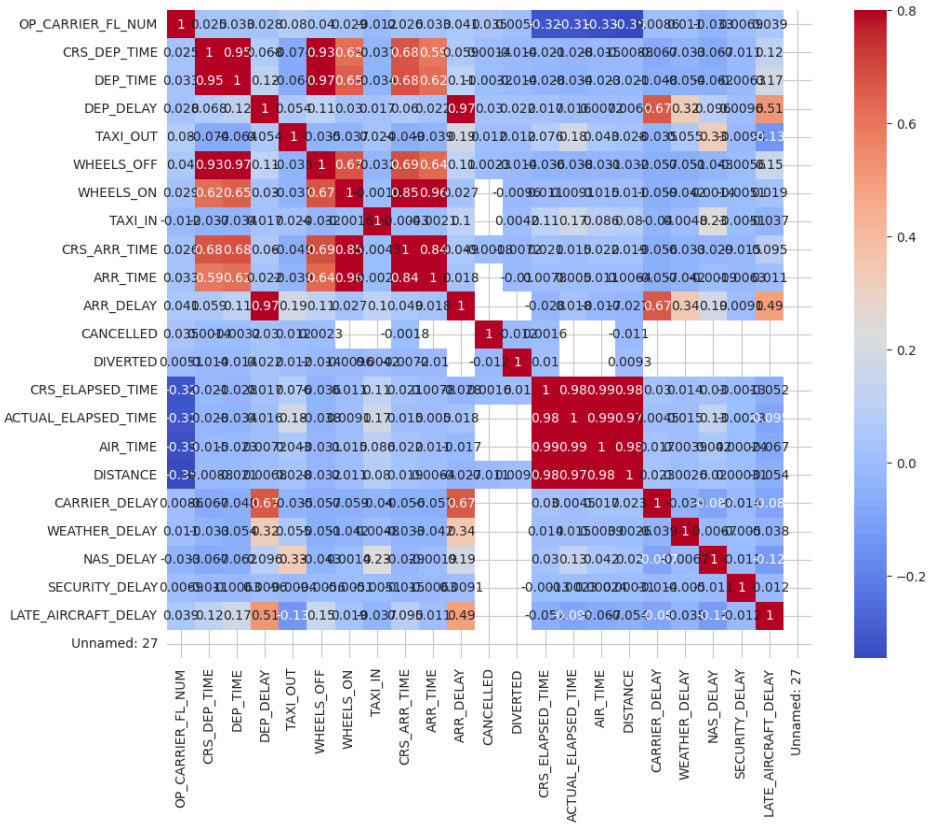

Figure 1: Correlation Matrix

The correlation matrix reveals several important relationships between variables, helping to address the first research question: *"What are the main causes of flight delays and cancellations?"* Strong positive correlations are observed between variables that are expected to be closely linked. For example, **CRS_DEP_TIME** and **DEP_TIME** show a very high correlation (~0.97), indicating that actual departures generally align with their scheduled times. Similarly, **WHEELS_OFF** and **WHEELS_ON** are strongly correlated (~0.87), as these variables directly reflect the takeoff and landing times. The correlation between **CRS_ELAPSED_TIME** and **ACTUAL_ELAPSED_TIME** (~0.98) also aligns with expectations, suggesting that planned and actual durations of flights are often very similar. Notably, **LATE_AIRCRAFT_DELAY** shows a strong correlation with **ARR_DELAY** (~0.61), highlighting that late-arriving aircraft significantly impact subsequent arrival delays.

Moderate correlations provide further insight into the dynamics of delays. **DEP_DELAY** and **ARR_DELAY** are moderately correlated (~0.66), showing that departure delays contribute meaningfully to arrival delays, although other factors are also at play. **DEP_DELAY** and **CARRIER_DELAY** (~0.50), as well as **ARR_DELAY** and **CARRIER_DELAY** (~0.48), indicate that airline operational issues are relevant contributors to both departure and arrival delays.

On the other hand, weak correlations highlight variables with limited influence. **WEATHER_DELAY**, for instance, has a relatively low correlation with most other features (~0.37 max), suggesting that while weather does impact delays, it is not the most dominant cause. **DISTANCE** shows almost no correlation with delay variables, indicating that longer flights do not necessarily experience greater delays.

These findings provide an early understanding of key delay drivers and serve as a foundation for deeper analysis and model development in later sections of the study.

4.1 Main causes for flight delays and cancellations

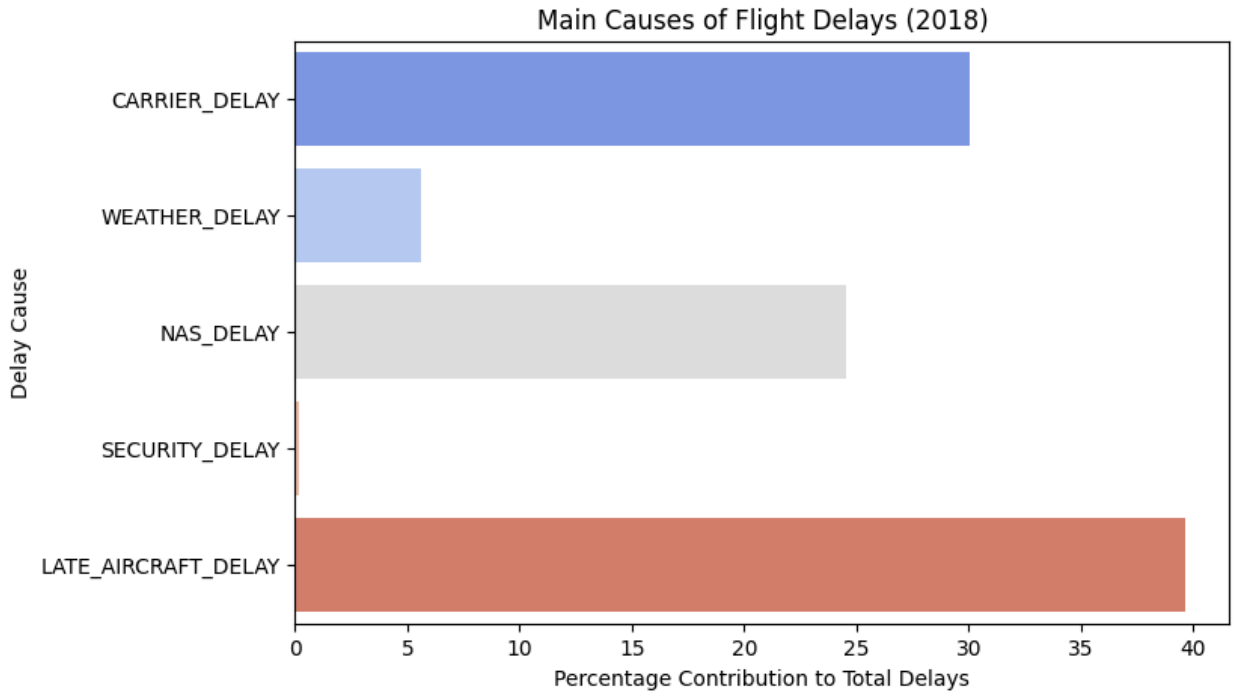

Figure 2: Main Causes for Flight Delays

From the graph, it is clear that **late aircraft delay** is the leading cause, contributing to nearly **40%** of all delays. This is followed by **carrier delays**, which account for approximately **31%**. **NAS delays** (National Aviation System delays) and **weather-related delays** follow with smaller shares. This visualization supports the findings from the correlation matrix and highlights late aircraft and carrier issues as the most significant contributors to flight delays.

4.2 Flight delays by airline, airport, and time of day

After identifying the primary causes of delays, the next step is to explore how these delays vary across different airlines, airports, and times of day. Analyzing these patterns helped uncover whether certain carriers or airports experience more delays than others, and whether delays are more likely to occur during specific times. This type of breakdown provides a deeper understanding of where and when delays are most common, which was useful for both operational improvements and strategic scheduling which airlines could adopt to reduce delays.

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.

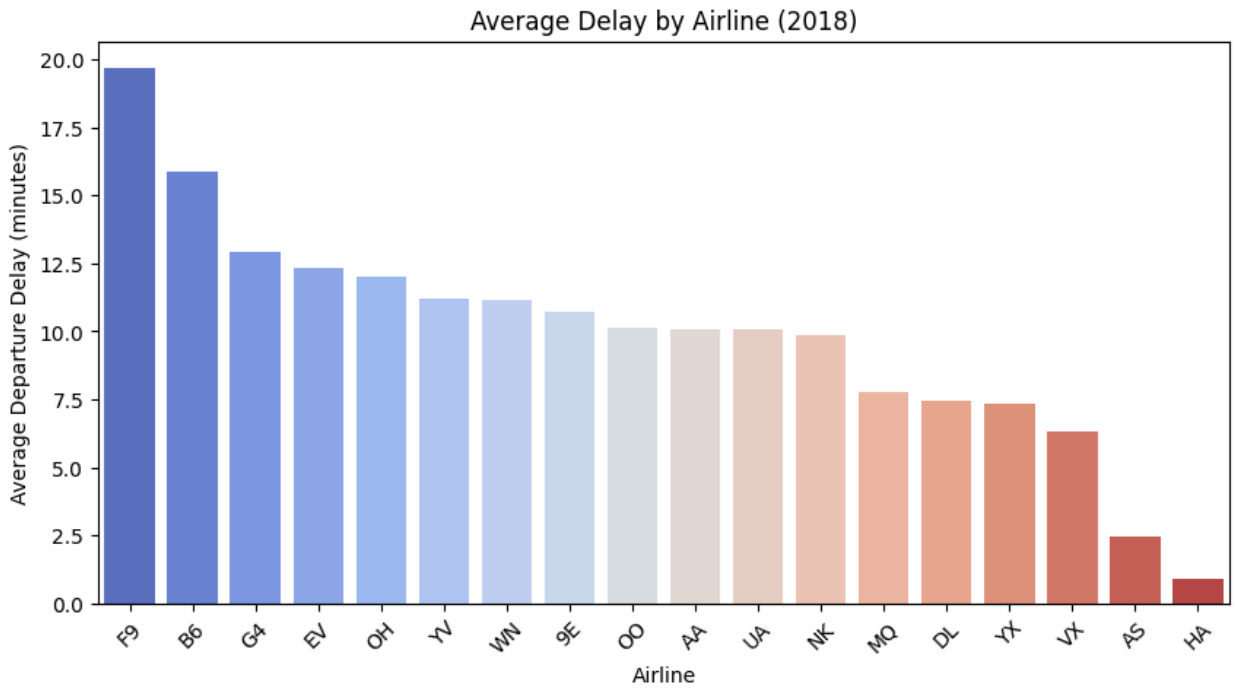

Figure 3: Average Delay by Airline

In figure 3, the top 18 airlines are ranked based on the total number of flight delays recorded in 2018. Carriers such as **F9 (Frontier Airlines)**, **B6 (JetBlue Airways)**, **G4 (Allegiant Air)**, **EV (ExpressJet)**, and **OH (PSA Airlines)** appear at the higher end, indicating a greater frequency of delays. These patterns may reflect a combination of operational inefficiencies, tight scheduling, or challenges related to regional connectivity and resource allocation. In contrast, airlines ranked lower on the chart exhibit fewer delays, which could be linked to more streamlined operations, less congested hubs, or more effective delay mitigation strategies. Such insights are valuable in identifying which carriers are more prone to delays and may benefit from targeted performance improvements.

Studying the average departure delay at the top 10 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.

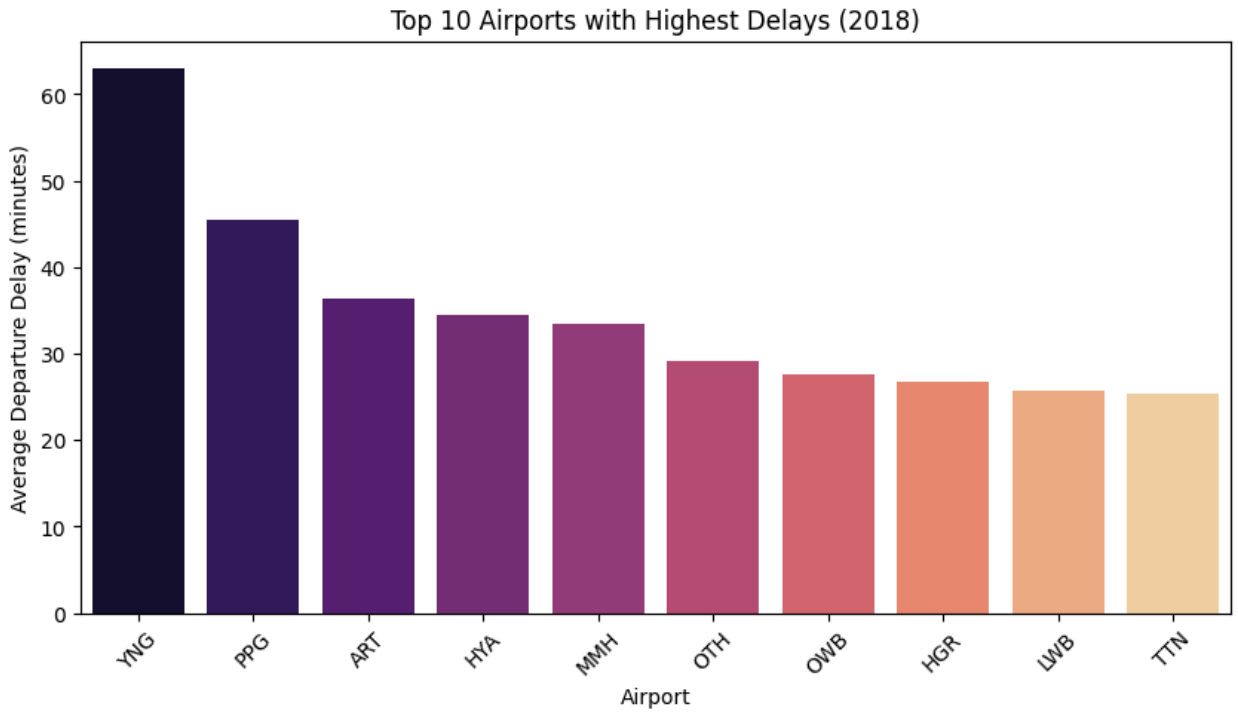

Figure 4: Top 10 Airports with highest delays

In Figure 4, **YNG (Youngstown–Warren Regional Airport)** stands out with the highest average departure delay of approximately **63 minutes**, indicating potential operational bottlenecks or regional challenges. **PPG (Pago Pago International Airport)** and **ART (Watertown International Airport)** follow with average delays of about **45 minutes** and **37 minutes**, respectively. The presence of smaller or regional airports such as **HYA**, **MMH**, **OTH**, and **OWB** in the top 10 suggests that limited infrastructure, fewer available resources, or irregular flight schedules may be contributing to longer delays. Unlike major hubs, these airports might lack the capacity to absorb disruptions efficiently, leading to more significant delays when issues arise. These findings highlight the importance of addressing infrastructure limitations and optimizing operations, even at smaller airports, to improve overall network reliability.

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 5, the line graph shows a general increase in average departure delay as the day progresses. The delays peak during busy hours such as early mornings and late evenings, 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.

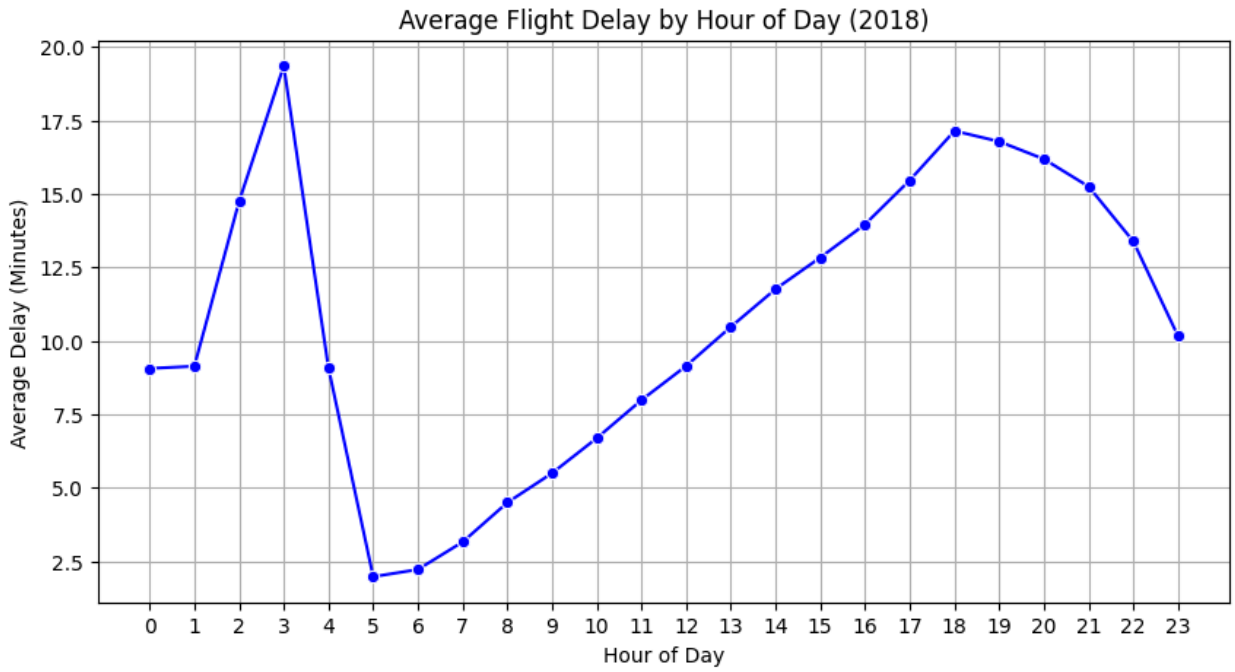

Figure 5: Average flight delays by hour of day

5. Model Formulation and Methods

The classification task involves predicting whether a flight is delayed (1) or not delayed (0). The methodology includes preprocessing, feature engineering, model training, and evaluation.

5.1 Data Pre-processing and Analysis

Data preprocessing 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.

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.

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. 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. This step was taken as the default solver did not converge even in 1000 iterations.

$$\text{Log Loss} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

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. 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. 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. 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. 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.

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.. 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.

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.

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. The loss function and regularization term are optimized to enhance model performance. XGBoost's functionality is given in the XGBClassifier from the xgboost library.. 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.

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.

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. 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': 1
- **Random Forest:** 'max depth': 13, 'min samples split': 3, 'n estimators': 400
- **KNN:** 'n neighbors': 1,2,3,4,5,6,7, 'weights': 'uniform'
- **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.

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).

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

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

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

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

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

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. 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.

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

6.2 Results and Analysis

Logistic Regression:

For logistic regression, Figure 6 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.

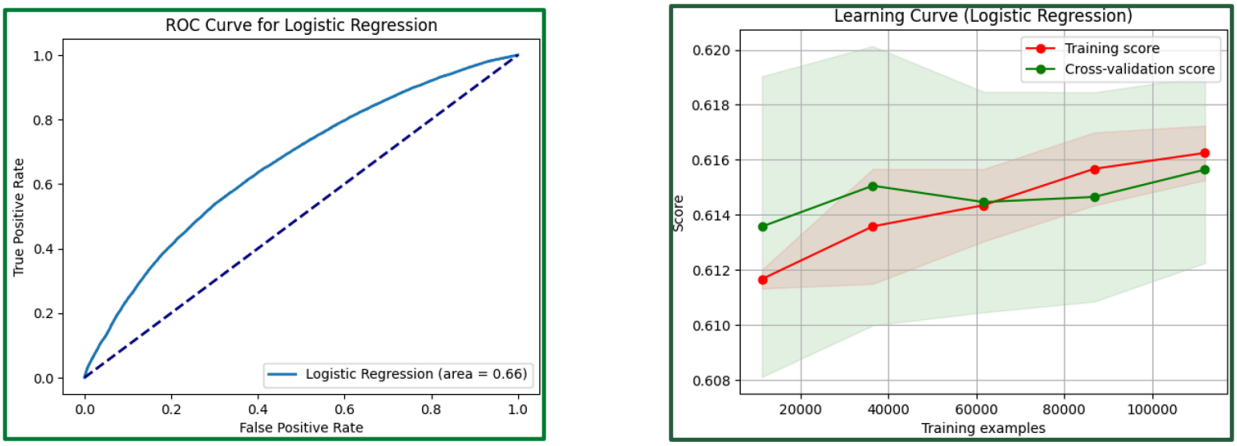

Figure 6: Logistic Regression: ROC Curve and Learning Curve

KNN:

For KNN, Figure 7 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.

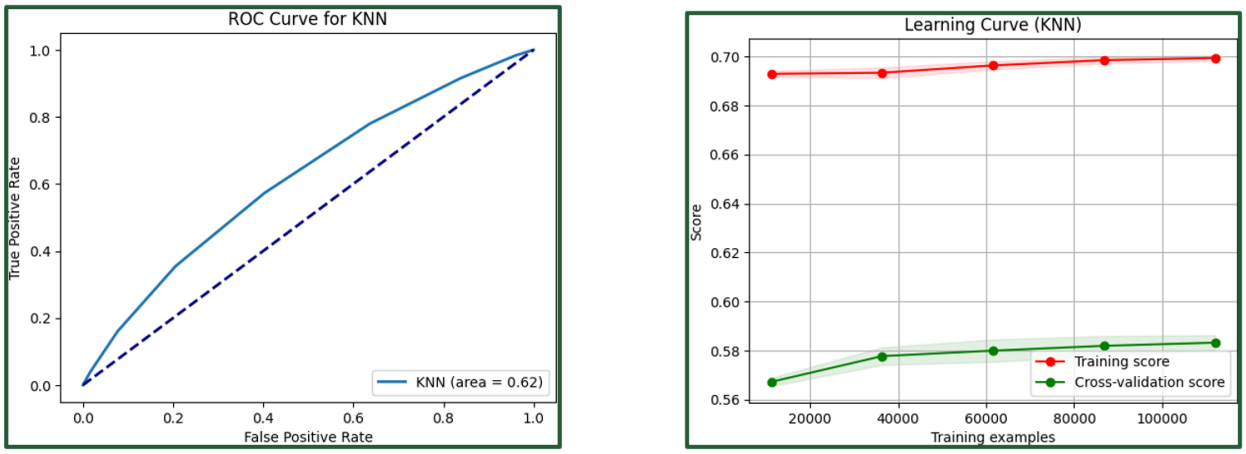

Figure 7: KNN: ROC Curve and Learning Curve

Random Forest:

For Random Forest, Figure 8 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.

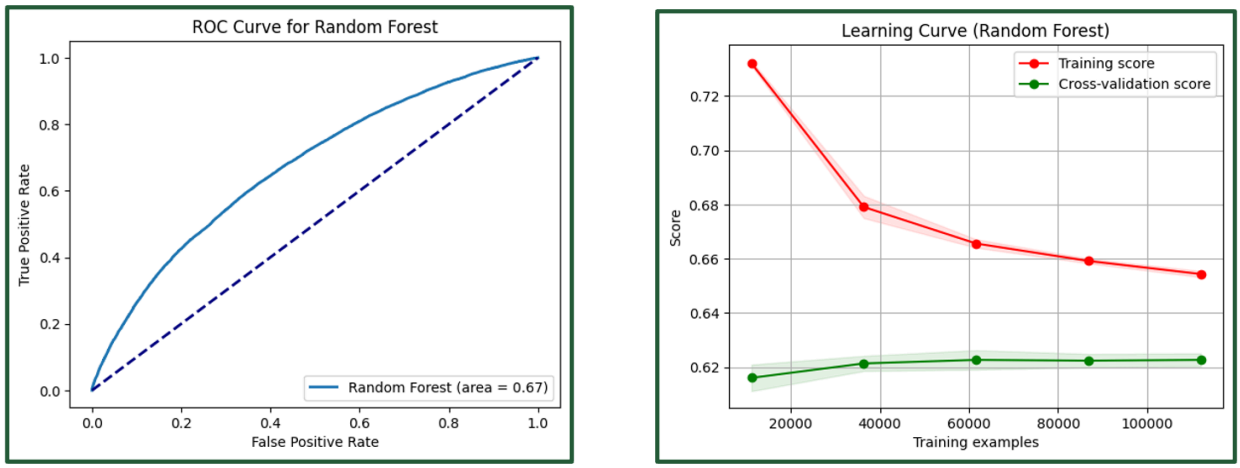

Figure 8: Random Forest: ROC Curve and Learning Curve

XGBOOST:

For the XGBoost model, Figure 9 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.

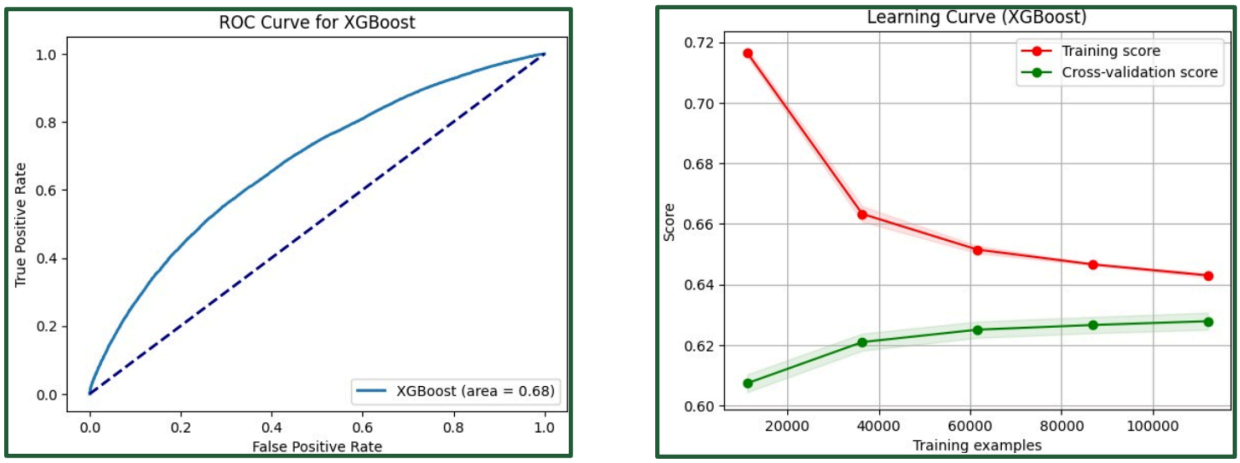

Figure 9: XGBOOST: ROC Curve and Learning Curve

Table 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 2: 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 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.

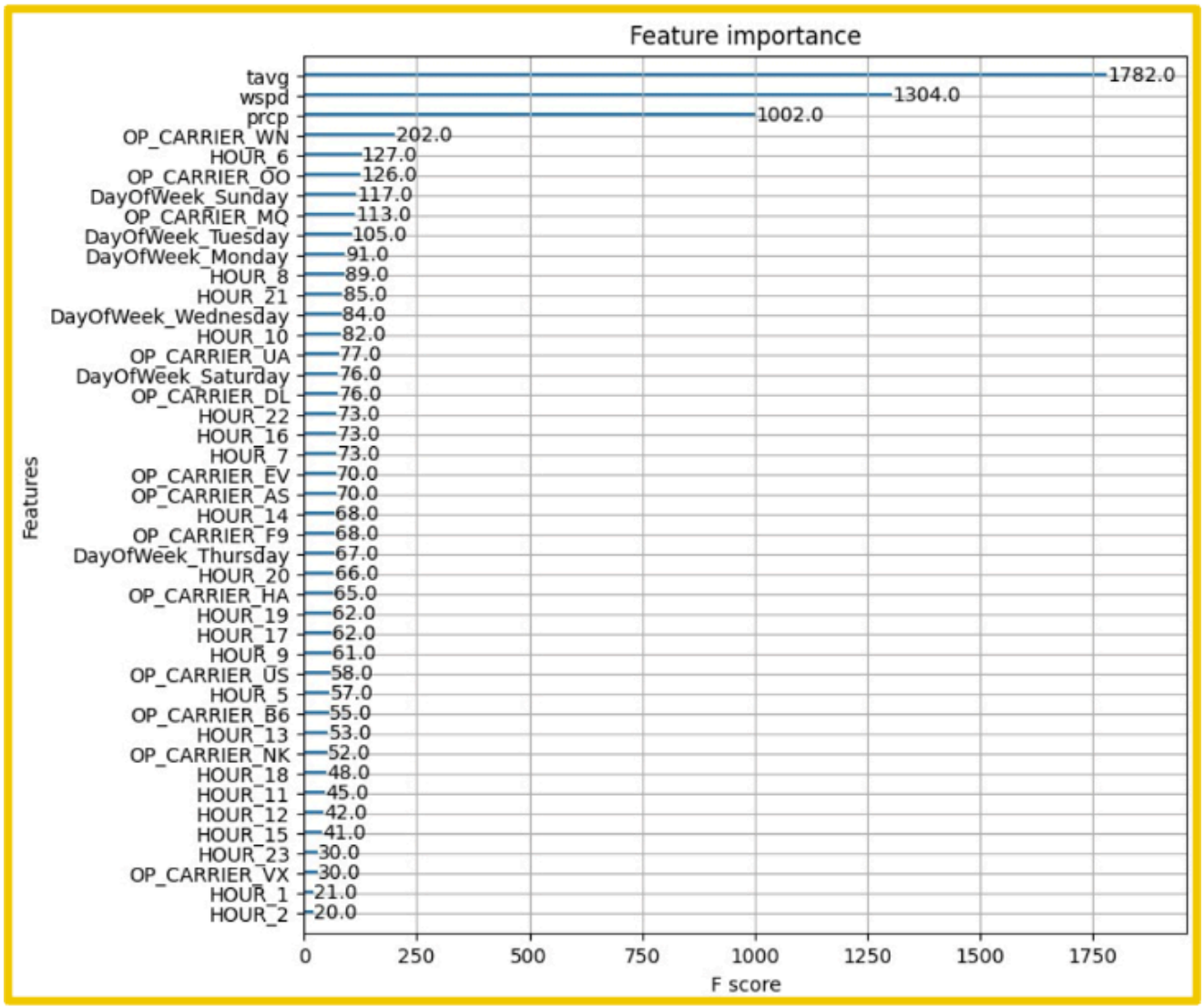

Figure 10: Feature Importances

6.3 How Do Airline-Specific Operational Strategies Influence Delay Management?

Airline-specific operational strategies play a critical role in managing and mitigating flight delays. While external factors such as weather and air traffic congestion are beyond the control of airlines, the internal decisions made by each airline—such as how they schedule crews, maintain aircraft, or design route networks—can significantly impact their ability to respond to and recover from delays.

One key strategy is fleet standardization, where airlines operate a uniform set of aircraft (e.g., Southwest Airlines using only Boeing 737s). This simplifies maintenance, crew training, and aircraft substitutions, thereby reducing turnaround time and minimizing delays due to equipment issues.

Another important factor is crew scheduling and standby availability. Airlines that efficiently manage crew assignments and maintain reserve staff are better equipped to prevent cancellations and minimize delays caused by unavailable pilots or flight attendants. Conversely, poor crew planning can lead to extended delays and missed departure slots.

Turnaround time optimization is also a major contributor. Airlines that streamline boarding, cleaning, refueling, and baggage handling can significantly reduce ground time and improve on-time performance. These improvements not only prevent delays on the current flight but also reduce the risk of subsequent delays across the fleet.

The route network model—whether hub-and-spoke or point-to-point—also affects delay management. Point-to-point carriers (e.g., Ryanair, Southwest) typically experience fewer cascading delays, as they avoid heavy reliance on central hub airports. In contrast, hub-based airlines are more vulnerable to system-wide delays triggered by disruptions at their major hubs.

Finally, the adoption of predictive analytics and AI-driven operations enables airlines to proactively respond to potential disruptions. Airlines that invest in data-driven decision-making tools can reroute flights, adjust schedules, and reassign crews in real time, significantly reducing the impact of delays.

In summary, airlines that focus on operational efficiency—through standardized fleets, optimized turnaround processes, proactive scheduling, and smart use of technology—are far more successful in managing and minimizing delays. These strategic differences explain why some airlines consistently outperform others in on-time performance despite operating under similar external conditions.

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.

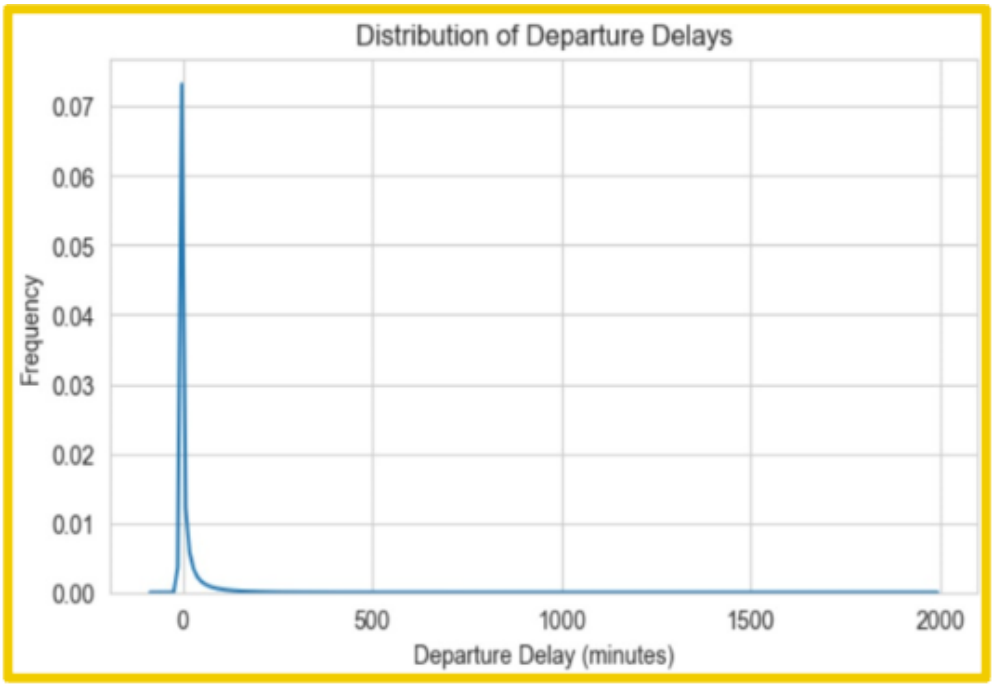

Figure 11: Skewness of the departure delay

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 11 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 for 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.

9. 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-cancellationdata-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-logisticregression-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

10. Appendix

The files related to this project- proposal, presentation, code and report are given in our GitHub repository-

https://github.com/Nikhila002/AIRLINE-DELAY-AND-CANCELLATION-ANALYSIS

https://github.com/amenasadiq7/MATH509