UNIVERSITÀ DI PISA

DEPARTMENT OF COMPUTER SCIENCE

Master program in Data Science and Business Informatics

Project Report

Flight Delay Prediction Dataset

Students

Phuong Chi Huynh
Minh Duc Pham

ACADEMIC YEAR 2025/2026

Contents

1 Introduction

Flight delay is an unexpected incident in the field of aviation in particular and transportation in general. One of the most frustrating experiences for passengers is when a flight is delayed or even cancelled. In this project, we apply distributed machine learning techniques to analyze a large-scale flight dataset and predict whether a flight will be delayed or not. The goal of this project is to find patterns in the data and identify the main factors that may cause flight delays. Our business questions to do this project are:

1. What are the main factors that influence flight delays?
2. How can we predict the actual arrival delay time, and which features contribute the most to this prediction?

The dataset used in this project was obtained from Kaggle, which collected the raw data from the Bureau of Transportation Statistics (BTS). The dataset contains about 2,000,000 records and 32 attributes. We used flight data from January 1, 2019 to August 31, 2023. It includes information from 18 airline companies operating flights between 380 different locations across 52 states in the United States. Each row in the dataset represents one flight and contains several types of information about that flight. In addition, the dataset includes important target variables such as *DEP_DELAY, ARR_DELAY*, which indicate whether a flight was delay and delay time.

2 Data Understanding

2.1 Data semantics

The dataset includes both numerical and categorical variables. Table 1 presents the variables used in the dataset, together with their type and a brief description.

Variable	Type	Description
FL_DATE	date	The date of the flight.
AIRLINE	string	The airline name.
AIRLINE_DOT	string	Official DOT (Department of Transportation).
AIRLINE_CODE	string	Airline short code.
DOT_CODE	integer	DOT code of airline.
FL_NUMBER	integer	Flight number assigned by the airline for this route.
ORIGIN	string	Airport code where the flight departed.
ORIGIN_CITY	string	City of the origin airport.
DEST	string	Airport code where the flight arrived.
DEST_CITY	string	City of the destination airport.
CRS_DEP_TIME	integer	Scheduled departure time according to the flight plan.
DEP_TIME	double	Actual time when the flight took off.
DEP_DELAY	double	Difference between scheduled departure time and actual time (minutes).
TAXI_OUT	double	Time taken to taxi from the gate to the runway before takeoff (minutes).
WHEELS_OFF	double	Exact time when the aircraft's wheels leave the runway.
WHEELS_ON	double	Exact time when the aircraft's wheels touch down on the runway.
TAXI_IN	double	Time taken to taxi from runway to arrival gate (minutes).
CRS_ARR_TIME	integer	Scheduled arrival time according to the flight plan.
ARR_TIME	double	Actual arrival time of the flight.
ARR_DELAY	double	Arrival delay in minutes.
CANCELLED	double	Flight cancelled (0 = no, 1 = yes).
CANCELLATION_CODE	string	Reason for cancellation.
DIVERTED	double	Flight diverted (0 = no, 1 = yes).
CRS_ELAPSED_TIME	double	Scheduled flight duration (minutes).
ELAPSED_TIME	double	Actual duration of the flight (minutes).
AIR_TIME	double	Time spent in the air between takeoff and landing (minutes).
DISTANCE	double	Distance between airports (miles).
DELAY_DUE_CARRIER	double	Delay caused by airline (minutes).
DELAY_DUE_WEATHER	double	Delay caused by weather (minutes).
DELAY_DUE_NAS	double	Delay caused by National Airspace System issues (traffic or congestion).
DELAY_DUE_SECURITY	double	Delay due to security procedures or checks.
DELAY_DUE_LATE_AIRCRAFT	double	Delay caused by the aircraft arriving late.

Table 1: Variables in the Flight Delay Dataset

We identified that *AIRLINE*, *AIRLINE_CODE*, *AIRLINE_DOT*, and *DOT_CODE* are nominal variables with overlapping information. To simplify the analysis, we dropped *AIRLINE* and *AIRLINE_DOT*, keeping only *AIRLINE_CODE* and *DOT_CODE*. We also dropped *ORIGIN_CITY, DEST_CITY* due to their irrelevance to the analysis. Based on the dataset, the sequence of events for a flight can be represented as: *DEP_TIME* $\rightarrow$ *TAXI_OUT* (min) $\rightarrow$ *WHEELS_OFF* $\rightarrow$ *AIR_TIME* (min) $\rightarrow$ *WHEELS_ON* $\rightarrow$ *TAXI_IN* (min) $\rightarrow$ *ARR_TIME*.

2.2 Data Quality

In this section, we evaluated the quality of the dataset by checking for duplicate rows and incorrect data types. Initially, we examined the summary statistics of the numerical features. We observed that *DEP_DELAY* and *ARR_DELAY* contain negative values. After checking their meaning, we found that negative values indicate that a flight departed or arrived earlier than the scheduled time. We also noticed that *CRS_ELAPSED_TIME* had a minimum value of -77, which is not a valid value for flight duration. To investigate this issue, we searched for rows where this variable was lower than 0. Only one such record was found, and we removed it then. We also verified that the dataset contains no duplicate records.

Statistic	CRS_DEP_TIME	DEP_TIME	CRS_ARR_TIME	ARR_TIME	WHEELS_OFF	WHEELS_ON
count	200000	1948570	200000	1946909	1947719	1946909
mean	1327.164	1329.81	1490.40	1466.37	1352.37	—1462.21
std	485.74	499.13	511.62	531.85	500.71	527.32
min	1	1	1	1	1	1
max	2359	2400	2400	2400	2400	2400

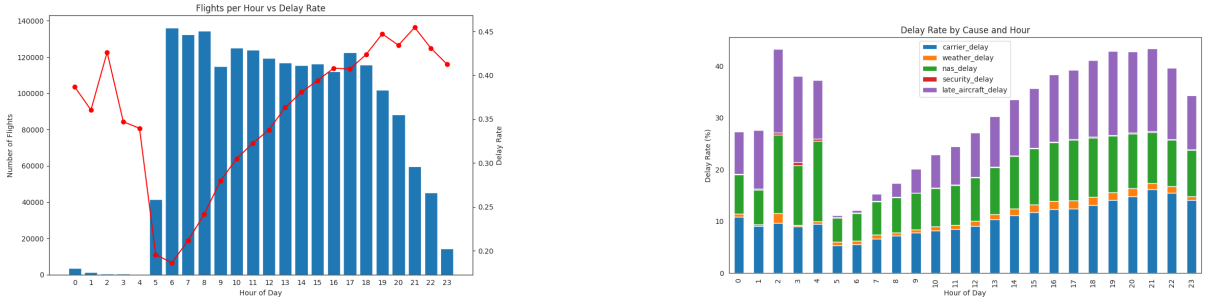
Table 2: Summary statistics of selected features

Next, we examined the minute component of several time-related variables, including *CRS_DEP_TIME, DEP_TIME, WHEELS_OFF, WHEELS_ON, CRS_ARR_TIME*, and *ARR_TIME*. From the table 2, we observed that some values were recorded as 2400, which should represent 00:00. Therefore, we converted all occurrences of 2400 to 00:00. We also verified that the minute values are all below 60, indicating no further formatting errors.

2.3 Missing value

For missing values, we examined the number of rows with NULL values in each feature. We found that 17 features contain missing values, most of them around 50,000 records. The column *CRS_ELAPSED_TIME* has only 17 missing values, so we removed those records. We observed that missing values are strongly related to flight status. When $CANCELLED = 1$, several variables such as *ARR_DELAY*, *ELAPSED_TIME*, *AIR_TIME*, *ARR_TIME*, *TAXI_IN*, *WHEELS_ON*, *TAXI_OUT*, *WHEELS_OFF* are missing. Similarly, when $DIVERTED = 1$, variables like *ARR_DELAY*, *ELAPSED_TIME*, and *AIR_TIME* are also missing. Since cancelled flights account for only about 2.5% of the dataset and our goal is to analyze flight delays, we removed rows where $CANCELLED = 1$ and $DIVERTED = 1$. The columns *CANCELLED*, *CANCELLATION_CODE*, and *DIVERTED* were then dropped. For delay reason features such as *DELAY_DUE_CARRIER*, *DELAY_DUE_WEATHER*, missing values occur because delay reasons are recorded only when *ARR_DELAY* >15 minutes. Therefore, we replaced these missing values with 0.

2.4 Distribution



(a) Flights per hour vs Delay rate

(b) Delay rate by cause and hour

Figure 1: Delay Flight analysis by Hour and Cause.

First, we calculated the departure and arrival delay rates. The delay rate is approximately 31% for departures and 34% for arrivals. Next, we analyzed the relationship between flight delays and the hour

of the day. We also examined the delay rate by different causes across hours to better understand how delay patterns vary over time. As shown in Figure 1a and 1b, we see the delay rate increase throughout the day, peaking in the late evening. This suggests that delays tend to accumulate over time, regardless of the total number of flights.

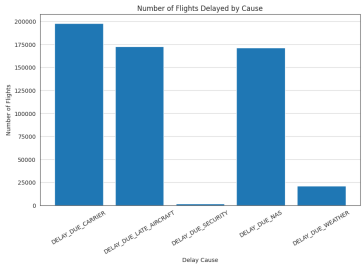
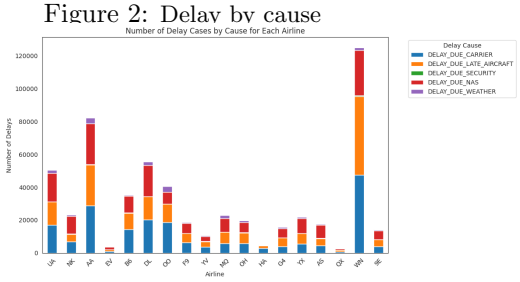
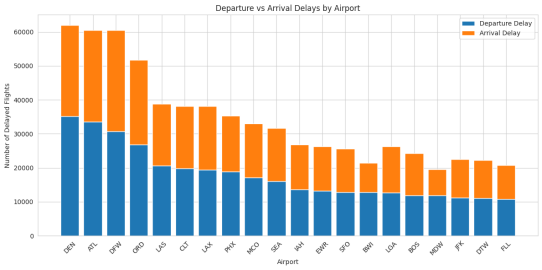


Figure 2: Delay by cause

Next, we analyzed flight delays by cause. As shown in Figure 2, carrier-related issues (accounting for 35.1%) and late aircraft arrivals (30.6%) are the primary drivers of delays. In contrast, security and weather-related problems account for only a very small fraction of the total cases. Furthermore, our analysis by airline and airport shows that Southwest Airlines (WN) has the highest number of delayed flights, mainly due to carrier and late aircraft issues, followed by American Airlines (AA) and Delta Airline (DL). Regarding airports, Denver International Airport (DEN) recorded the highest delay rate.



(a) Delay Flights by Airlines



(b) Delay rate by airport

Figure 3: Delay Flight analysis by airline and airport.

2.5 Correlation

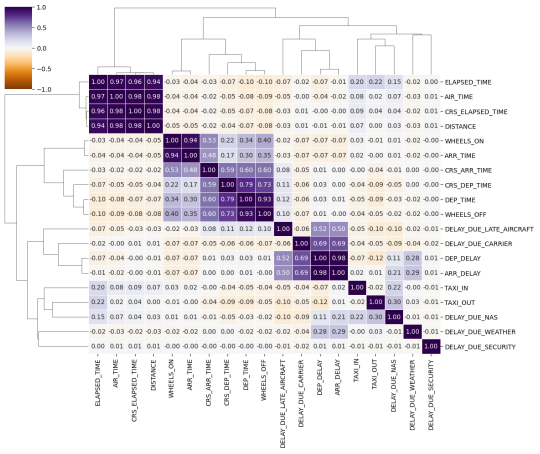


Figure 4: Heatmap of feature correlations based on the Pearson correlation matrix.

To analyze the relationships between features in our dataset, we leveraged several key functionalities. First, we used the VectorAssembler to combine all numerical feature columns into a single vector column, which is a prerequisite for correlation analysis in PySpark. Next, we utilized the Correlation module from 'pyspark.ml.stat' to compute the Pearson correlation matrix, allowing us to quantify pairwise relationships between features. The resulting correlation matrix is visualized in figure 4. From this figure, it indicated *AIR_TIME ELAPSED_TIME CRS_ELAPSED_TIME DISTANCE* may be redundant. And *DEP_DELAY, ARR_DELAY* May be lead to data leakage if using both in learning tasks.

2.6 Outliers Detection

We used boxplots to detect outliers in several numerical features. The plots show that some variables, especially *DEP_DELAY* and *ARR_DELAY*, contain extreme values that can exceed 1000 minutes. These

cases are rare but may affect the distribution of the data. To reduce the impact of these extreme values, we filtered the dataset using the following thresholds: $DEP\_DELAY \leq 600$, $ARR\_DELAY \leq 600$, $TAXI\_OUT \leq 90$, $TAXI\_IN \leq 60$, and $DISTANCE \leq 3000$. This filtering removes rare extreme values and makes the dataset more stable for analysis and modeling.

2.7 Features Engineering

To capture time-related patterns in flight delays, several new features were created from existing variables. From *CRS_DEP_TIME*, we derived the variable *HOUR*, which represents the scheduled departure hour. From *FL_DATE*, we extracted *WEEKDAY* and *MONTH* to capture weekly and seasonal patterns that may affect flight delays.

For the classification purpose (we would do later), we created a binary target variable called *DELAYED*. A flight is labeled as 1 if *DEP_DELAY* is greater than 15 minutes, and 0 otherwise. This threshold follows the standard definition used in aviation statistics. Moreover, several aggregated features were computed to capture delay patterns, including *AIRPORT_DELAY_RATE*, *ROUTE_DELAY_RATE*, and *AIRLINE_DELAY_RATE*. These variables represent the average delay rate by airport, route, and airline.

We also removed irrelevant features because they act as identifiers, contain redundant raw time information, or may cause data leakage. These include nominal identifiers such as *FL_DATE, DOT_CODE, ...*, raw time variables such as *CRS_DEP_TIME, DEP_TIME, ...*, and delay cause variables. The delay cause features were only used for exploratory data analysis (EDA) and were excluded from learning tasks.

3 Clustering

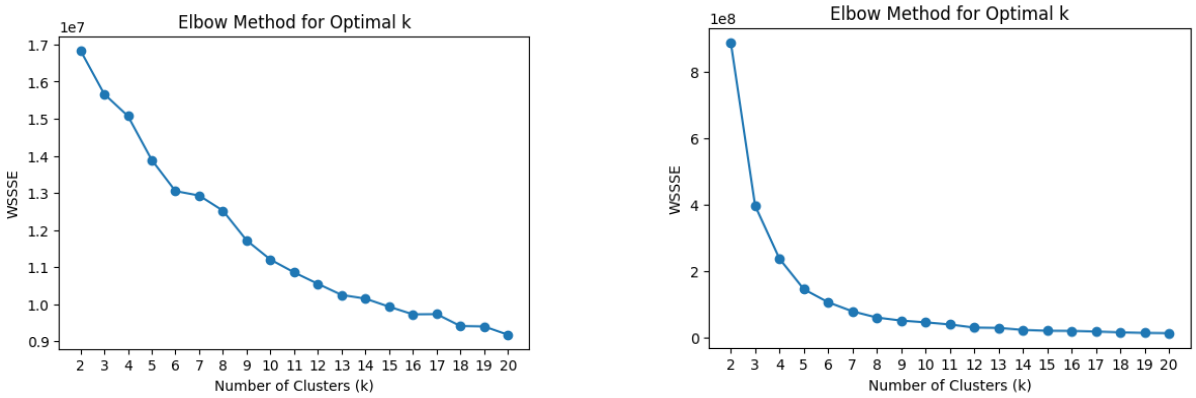
3.1 Dataset

In this section, clustering is applied to better understand delay patterns in the dataset. To perform the analysis, the data was divided into two parts. The first part is the **flight** dataset, which is the cleaned dataset obtained from the preprocessing stage. From this dataset, 10 relevant features were selected for clustering: *TAXI_OUT, TAXI_IN, DISTANCE, Hour, DayOfWeek, Month, AIRPORT_DELAY_RATE, ROUTE_DELAY_RATE, AIRLINE_DELAY_RATE, and ARR_DELAY.*

The second part is the **route** dataset. A route is defined as the combination of origin and destination airports. The data was aggregated by *ROUTE* to summarize the overall characteristics of flights operating on the same route. During this aggregation process, several statistics were computed, including the total number of flights, average departure delay, average arrival delay, average taxi-out time, average taxi-in time, and average flight distance. After aggregation, each route is represented by a set of numerical features describing its operational and delay characteristics. The features we used include *avg_dep_delay, avg_arr_delay, delay_rate, avg_taxi_out, avg_taxi_in, distance, airport_delay_rate, route_delay_rate*. We applied the route-level dataset to explore hidden patterns among routes, such as groups of routes with high delays, more stable routes, or routes affected by factors such as airport congestion or long taxi times. This route-level analysis helps reveal broader patterns in the flight network and provides additional insights into delay behavior.

3.2 Implementation

In this section, we applied the K-Means clustering algorithm, which groups observations based on distance similarity. Before training the model, a preprocessing step was performed to both datasets. All selected numerical features were standardized using Z-score normalization. To determine the optimal number of clusters k, we applied the Elbow method and evaluated the clustering structure. After selecting the appropriate k, the K-Means model was trained in PySpark with *maxIter = 50* and *initSteps = 5*. Running the algorithm with multiple initialization steps and sufficient iterations helps improve clustering stability and ensures convergence. Finally, we interpreted the resulting clusters using two visualization techniques. A parallel coordinates plot was used to analyze the distribution of features across clusters, while a PCA-based visualization was applied to project the high-dimensional data into a two-dimensional space for easier cluster interpretation.



(a) The Elbow method for Delay Flight

(b) The Elbow method for Delay Route

Figure 5: The Elbow method for optimal k

3.3 Result

From the figure 5a for flight dataset, the WSSE decreases rapidly when increasing k from 2 to around 8-10, indicating that adding more clusters significantly improves cluster compactness in this range. However, after this point, the decrease becomes much slower and the curve starts to flatten. This pattern suggests diminishing returns when increasing the number of clusters further. Based on this elbow-shaped curve, we selected $k = 5$, as a reasonable balance between cluster compactness and model simplicity. Then, we implemented the training process with that value.

As shown in Figure 5b, the WSSE drops sharply when the number of clusters increases from $k = 2$ to $k = 3$, and the decrease becomes more gradual afterward. This pattern suggests that the elbow point occurs at $k = 3$. Therefore, we chose $k = 3$. After choosing k value, we implemented the training process.

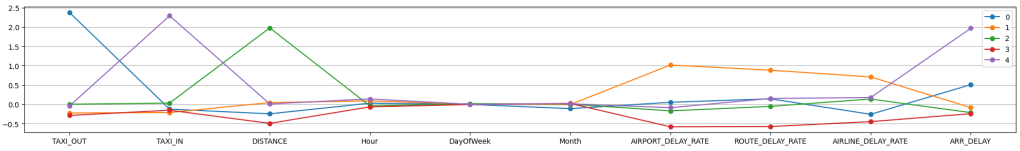


Figure 6: The parallel chart at Flight dataset

Figure 6 presents the parallel chart for the clustering results on the flight dataset. The figure shows clear and distinct differences among the five clusters, with very little overlap in their main characteristics. This indicates that the clustering process successfully separated flights into groups with different behavioral patterns. In contrast, Figure 7 shows the clustering results for the route dataset. The main difference between clusters appears only in the distance feature: Cluster 1 mainly represents long-distance routes, Cluster 2 corresponds to medium-distance routes, while Cluster 0 contains the remaining routes. Other features do not show clear separation among the clusters. Because the patterns in the route-level clustering are not clearly distinguishable, the insights obtained from this analysis are limited. Therefore, we decided not to further explore the route clustering results, as they did not provide meaningful findings compared to our initial expectations.

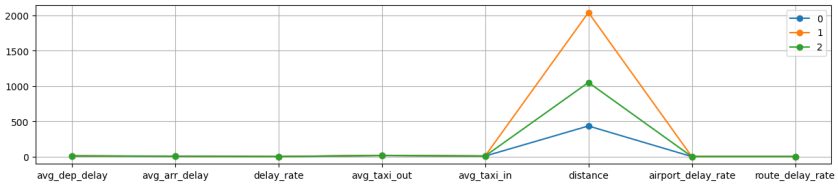


Figure 7: The parallel chart at Route dataset

From the parallel plot 6, we can interpret the clusters generated by the flight analysis as below:

- Cluster 0 (160,960 records): Flights with significantly higher distance compared to other clusters, while maintaining moderate taxi times and average delay levels. These flights likely correspond to long-haul routes where operational complexity is higher but delays are not extreme.
- Cluster 1 (545,176 records): Flights characterized by relatively high airport delay rate and route delay rate, with moderate distance and taxi times. This cluster likely represents flights affected mainly by airport congestion or route-level constraints.
- Cluster 2 (227,695 records) : Flights with relatively low delay rates and low airport and route delay contributions. This cluster indicates more efficient operations and likely corresponds to well-performing routes.
- Cluster 3 (882,088 records): Flights with significantly higher airline delay rate and relatively elevated arrival delays. This cluster may represent cases where delays are mainly caused by airline operational issues.
- Cluster 4 (115,988 records): Flights characterized by very high arrival delays and higher taxi-in times. This pattern suggests arrival congestion or delays accumulated during flight operations, possibly at busy destination airports.

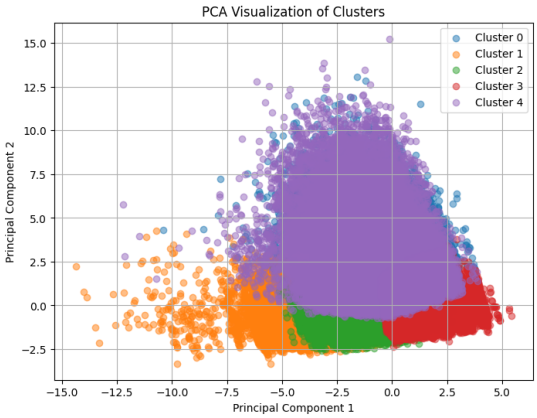


Figure 8: The PCA visualization of Clusters

The figure 8 shows the PCA visualization of the clustering results. Some clusters are separated along the horizontal axis (Principal Component 1). However, several clusters still overlap in the center area. This indicates that the clusters capture some patterns in the data, but the boundaries between groups are not very clear. Overall, the clustering result is acceptable but not very strong. Some separation between clusters exists, but overlap is still visible, suggesting that the data patterns are captured only partially.

4 Classification

In this section, we performed a classification task on the binary variable is "DELAY", which indicates whether a flight was delayed. As observed during the data exploration phase, approximately 33% of flights in the dataset experienced delays.

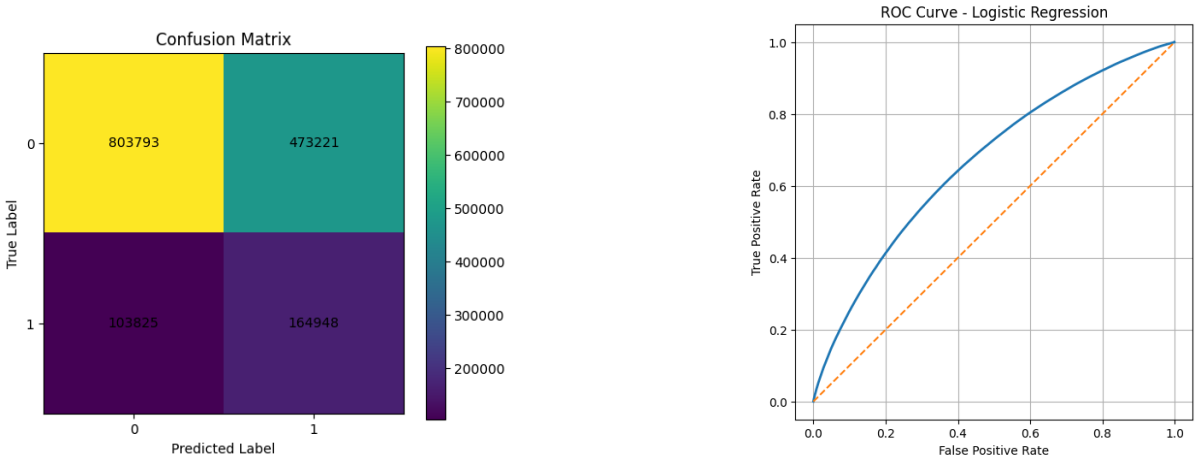
Before training the model, we performed several preprocessing steps to prepare the data for the delay classification task. First, the dataset was split into training and test sets using an 80/20 ratio. Based on the EDA results, the number of delayed flights is smaller than non-delayed flights, indicating an imbalanced dataset. We calculated class weights to handle the imbalance between delayed and on-time flights. We also computed average delay rates by airline, origin airport, and route, and filled missing values with the overall delay rate to handle unseen categories in the test set. Next, we calculated *AIRPORT_DELAY_RATE, ROUTE_DELAY_RATE*, and *AIRLINE_DELAY_RATE* based on the training set to avoid data leakage. The features used for the model include *DISTANCE, CRS_ELAPSED_TIME, TAXI_OUT, TAXI_IN, Hour, DayOfWeek, Month, AIRPORT_DELAY_RATE,*

AIRLINE_DELAY_RATE, and *ROUTE_DELAY_RATE*. These features were combined into a single feature vector using *VectorAssembler*, since Spark ML requires features to be stored in a vector column. Finally, the features were standardized using *StandardScaler* before training the model.

After preprocessing, we trained a Logistic Regression model to predict flight delays. We used the standardized feature vector (scaled_features) as input and the target variable *DELAYED* as the label. To handle the imbalanced dataset, we included the previously computed classWeight column, which assigns higher weight to delayed flights, helping the model focus more on the minority class. The model was trained with a maximum of 50 iterations to ensure convergence. Once trained, we applied the model to the test set to generate predictions. For each flight, the model produced both a predicted class (0 or 1) and a probability score for being delayed. We evaluated the model using two metrics:

- **AUC (Area Under the ROC Curve)** - measures how well the model distinguishes between delayed and on-time flights. The model achieved AUC = 0.666, indicating a moderate ability to separate the two classes. The result is shown at the figure 9b.
- **Accuracy** – measures the proportion of correct predictions. The model achieved accuracy ≈ 0.627, reflecting the overall correctness but showing that predicting the minority class (delayed flights) remains challenging.

The confusion matrix 9a shows that some delayed flights were misclassified as on-time, which is expected given the class imbalance. Using class weights helped reduce this bias, but further improvements could be explored using additional features or more advanced models.



(a) The confusion matrix by Logistic Regression

(b) ROC Curve of Logistic Regression

Figure 9: The result of Confusion matrix and ROC by Logistic regression

After training the Logistic Regression model, we analyzed the feature coefficients to understand which factors most influence flight delays. In logistic regression, the magnitude and sign of a coefficient indicate the importance and direction of the relationship with the target. The figure 10 represents our results after doing feature importance. From the results, *Hour, ROUTE_DELAY_RATE, and AIRLINE_DELAY_RATE* have the largest positive coefficients, meaning that flights during certain hours, specific routes, or operated by certain airlines are more likely to be delayed. *TAXI_OUT* and *CRS_ELAPSED_TIME* also contribute positively but to a lesser extent. On the other hand, *DISTANCE* and *AIRPORT_DELAY_RATE* have small negative coefficients, suggesting that longer flights and certain airports are slightly less associated with delays.

This analysis provides practical insights: delays are strongly linked to the route, airline, and departure time rather than just distance or airport. Overall, the feature importance aligns with expectations, though some coefficients (e.g., negative for distance) may reflect correlations in the dataset rather than causal effects. It helps prioritize which factors to monitor or optimize to reduce delays.

5 Regression

In this section, we implemented a Regression task to predict the Arrival Delay time of the vector. We tested the algorithm Linear Regression, Random Forest Regressor. The ratio of training and

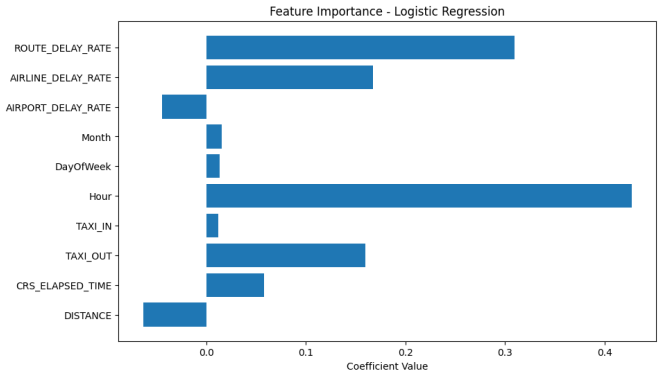


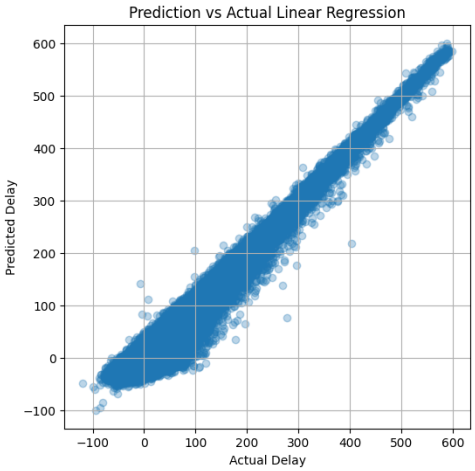
Figure 10: Feature Importance - Logistic Regression

test set was 80:20. Before doing the regression analysis, we did some preprocessing steps. First, we created 2 new features: *PREVIOUS_FLIGHT_DELAY* and *AIRPORT_CONGESTION*. The *PREVIOUS_FLIGHT_DELAY* feature represents the arrival delay of the previous flight operated by the same airline on the same route, capturing the possible propagation of delays between consecutive flights. The *AIRPORT_CONGESTION* feature measures the number of flights departing from the origin airport within the same hour, which reflects how busy the airport is at that time. A higher congestion level may increase the probability of delays, providing additional context for predicting arrival delay.

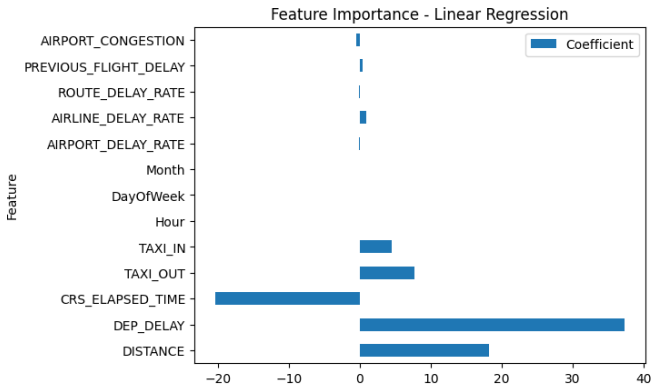
After creating the new features, all selected variables were combined and scaled using StandardScaler. The model uses several features such as distance, departure delay, taxi time, scheduled flight time, time-related variables, delay rate features, and the two new features *PREVIOUS_FLIGHT_DELAY* and *AIRPORT_CONGESTION*. These features provide useful information about flight operations and possible delay factors.

5.1 Linear Regression

First, a Linear Regression model was used as a baseline model. The model was trained with 50 maximum iterations and a regularization parameter of 0.1. After training, the model was evaluated on the test dataset. The results show an RMSE of 9.04 and an MAE of 6.49, which means the prediction error is around 6–9 minutes on average. The model also achieved an R^2 score of 0.95, meaning that about 95% of the variation in arrival delay can be explained by the selected features. This is a strong result and shows that the model performs well, demonstrating that the relationship between departure delay and arrival delay is largely linear. It also suggests that factors such as departure delay, taxi time, and airport congestion have a strong relationship with arrival delay.



(a) Prediction vs Actual Linear Regression



(b) Feature Importance - Linear Regression

Figure 11: The result of Linear Regression

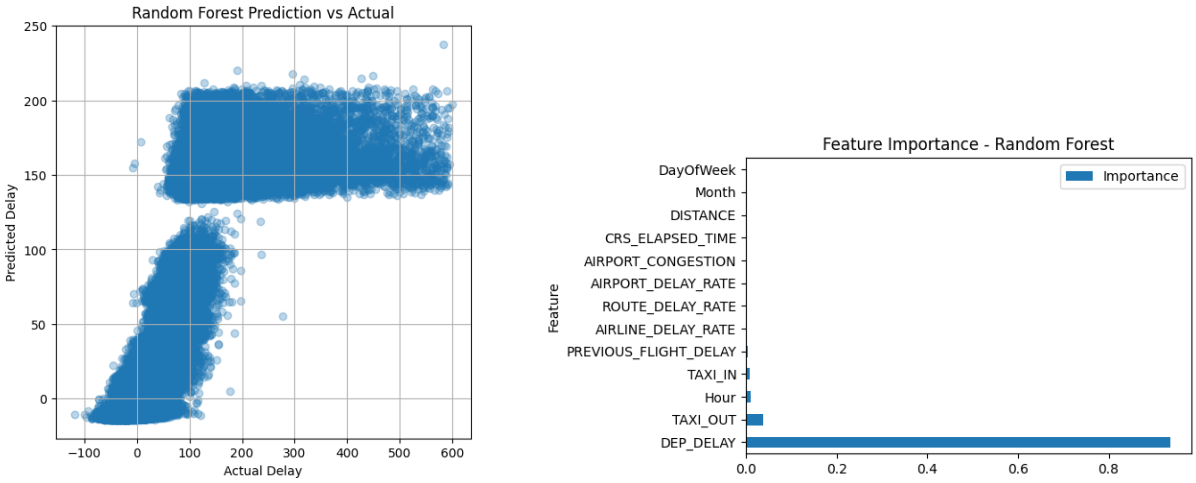
To better understand how each feature affects the prediction, we examined the coefficients of the Linear Regression model. The results show that *DEP_DELAY* has the largest positive coefficient, indicating that departure delay is the most important factor affecting arrival delay. This suggests that delays at the beginning of the flight often continue and lead to delays at arrival. Other features such as *DISTANCE*, *TAXI_OUT*, and *TAXI_IN* also have positive effects, meaning longer routes and longer taxi times may increase the arrival delay. In addition, *AIRLINE_DELAY_RATE* and *PREVIOUS_FLIGHT_DELAY* show positive relationships with arrival delay, which supports the idea that operational conditions and delay propagation can affect flight performance. Overall, these results provide useful insights into which factors contribute most to predicting arrival delay and help explain how delays occur in the flight system. The figure 11 represents the result of linear regression analysis.

5.2 Random Forest Regressor

After the Linear Regression model, a Random Forest Regressor was applied to predict arrival delay. Random Forest is an ensemble method that uses multiple decision trees to improve prediction performance and capture more complex relationships between features. In this experiment, the model was trained with 30 trees, a maximum depth of 6, and 32 bins.

The model was then evaluated using RMSE, MAE, and R^2. The results show an RMSE of 18.72 and an MAE of 10.34, meaning the average prediction error is around 10–18 minutes. The model achieved an R^2 score of 0.78, which indicates that about 78% of the variation in arrival delay can be explained by the model.

Compared with the Linear Regression model, the Random Forest model performs worse in this case. This suggests that the relationship between the selected features and arrival delay may be mostly linear, or that the Linear Regression model already captures the main patterns in the data. However, Random Forest still helps confirm that factors such as departure delay, taxi time, and operational conditions are important for predicting arrival delay.



(a) Prediction vs Actual Random Forest Regressor

(b) Feature Importance - Random Forest

Figure 12: The result of Random Forest

The prediction-vs-actual plot in Figure 12a for the Random Forest model shows that the points are more spread out around the diagonal compared to Linear Regression. Although the model still captures the general upward trend between actual and predicted delays, the predictions are less accurate and more scattered, showing a weaker alignment with the ideal diagonal line. Additionally, some horizontal clustering patterns appear in the predictions, indicating that the model tends to produce similar predicted values for groups of observations. This occurs because Random Forest regression outputs the average value of samples within each leaf node, which can lead to less continuous prediction behavior.

5.3 Key insights

Overall, the comparison of the prediction versus actual plots shows that Linear Regression predicts arrival delays more accurately and consistently than Random Forest. This suggests that the relationship between key features, especially departure delay-and arrival delay is mainly linear, so a simple model works better than a complex one. Both models show that departure delay is the most important factor, with the Random Forest model giving it the highest feature importance. Other factors, like taxi times and scheduled elapsed time, also affect predictions but much less. The Linear Regression model predicts large delays particularly well, showing that departure delay strongly drives arrival delay. For flights arriving on time or slightly early, both models tend to slightly overestimate negative delays, reflecting the overall pattern of early arrivals in the dataset.

6 Conclusion

In this project, we analyzed a large-scale flight dataset to understand delay patterns and build predictive models using supervised and unsupervised learning. We used Apache Spark to process a large dataset of about 2 million records. Initial exploration was done on Google Colab, but its 2 CPU cores were not enough for heavy tasks. Therefore, clustering, classification, and regression were run on a local machine with 10 CPU cores. To improve performance, the data was repartitioned (20 partitions for classification and 40 for regression), shuffle partitions were adjusted, and memory was increased to 8GB for both driver and executor. These steps made computation faster and show the importance of tuning Spark for dataset size and hardware.

From the exploratory analysis, we found that delays are not random. They tend to increase during the day, suggesting that delays accumulate over time. Carrier-related issues and late arrivals of aircraft were the main causes, while weather and security had less effect. Clustering gave some insights into flight behavior. Some clusters had different distances or delay rates, but the separation between groups was not very clear. This shows that delay patterns are complex and hard to divide into distinct categories. For classification, the Logistic Regression model had moderate performance, with an AUC of 0.666 and accuracy around 0.627. This shows that predicting whether a flight will be delayed is difficult. Important features were departure hour, route delay rate, and airline delay rate. For regression, Linear Regression worked better than Random Forest. It achieved $R^2 = 0.95$ and low errors. This means the relationship between departure delay and arrival delay is mostly linear. Departure delay was the most important factor, showing that delays often continue through the flight process.

Overall, the project shows that flight delays are mainly caused by operational factors, including airline performance, route characteristics, and departure time. Simple models like Linear Regression can capture the main patterns in the data well. Using more complex models does not always improve results when the relationships are mostly linear. These findings can help airlines and airports understand delays better and improve scheduling and operations in the future.