IE 7275 DATA MINING

SHUBHAM GAUR (gaur.sh@northeastern.edu)

SHUO ZHUANG (zhuang.shuo@northeastern.edu)

MALHAR GHOGARE (ghogare.m@northeastern.edu)

PROJECT PROPOSAL

Introduction:

The aviation industry faces significant challenges related to flight delays, which can lead to inconvenience for passengers and financial losses for airlines. By leveraging data mining techniques, we aim to develop predictive models for both regression (predicting delay time) and classification (identifying the type of airline) tasks. This project seeks to enhance understanding and provide actionable insights into the factors contributing to flight delays and the characteristics of different airlines.

Data Description: (Dataset link: https://figshare.com/articles/dataset/flights_csv/9820139)

Our dataset covers a diverse range of airlines, airports, and flight routes, providing a holistic perspective on the aviation landscape.

Data Dictionary:

Variable Name	Description
YEAR	The year of the flight
MONTH	The month of the flight.
DAY	The day of the month of the flight.
DAY_OF_WEEK	The day of the week of the flight
AIRLINE	The code representing the airline operating the flight
FLIGHT_NUMBER	The unique identification number assigned to the flight
TAIL_NUMBER	The registration number of the aircraft
ORIGIN_AIRPORT	The code representing the origin airport of the flight
DESTINATION_AIRPORT	The code representing the destination airport of the flight
SCHEDULED_DEPARTURE	The scheduled departure time of the flight (in local time)

DEPARTURE_TIME	The actual departure time of the flight (in local time)
DEPARTURE_DELAY	The delay in departure time (in minutes)
TAXI_OUT	The time taken for taxiing out (in minutes)
WHEELS_OFF	The time at which the aircraft's wheels leave the ground (in local time)
SCHEDULED_TIME	The scheduled duration of the flight (in minutes)
ELAPSED_TIME	The actual elapsed time of the flight (in minutes)
AIR_TIME	The time spent in the air during the flight (in minutes)
DISTANCE	The distance traveled by the flight (in miles)
WHEELS_ON	The time at which the aircraft's wheels touch the ground upon arrival (in local time)
TAXI_IN	The time taken for taxiing in (in minutes)
SCHEDULED_ARRIVAL	The scheduled arrival time of the flight (in local time)
ARRIVAL_TIME	The actual arrival time of the flight (in local time)
ARRIVAL_DELAY	The delay in arrival time (in minutes)

Problem Statement:

Through rigorous data mining and analysis, we aim to uncover patterns, trends, and correlations that shed light on the performance metrics of different airlines. By leveraging advanced regression and classification techniques, we seek to develop predictive models capable of forecasting delay times with precision and accurately classifying airlines based on their operational profiles. These models hold immense potential for informing strategic decision-making within the aviation industry, enabling airlines to proactively address delay-related challenges and optimize their operations for enhanced efficiency and customer satisfaction.

Exploratory Data Analysis (EDA):

Data Pre-Processing:

We used the isnull() function and the sum() method to calculate the total number of null values in the dataframe. This made it easier to comprehend how much data was missing from our dataset. Columns with no data were eliminated since they were not relevant to the goals and analysis of our study. It makes sense to move forward with deleting the rows with null values instead of replacing them since we have found a maximum of 105,000 rows with null values, which is a small portion of our possible sample size of 5,000,000 rows. This protects our dataset's quality and guarantees data integrity for analysis and model training.

In [ ]:
```
df.isnull().sum()
```

Calculating the sum of null values in the dataframe

In [ ]:
```
# columns_to_drop = ['CANCELLATION_REASON', 'AIR_SYSTEM_DELAY', 'SECURITY_DELAY', 'AIRLINE_DELAY', 'LATE_AIRCRAFT_DELAY', 'WEATHE
columns_to_drop = ['DIVERTED', 'CANCELLED']
df.drop(columns=columns_to_drop, inplace=True)

print(df)
```

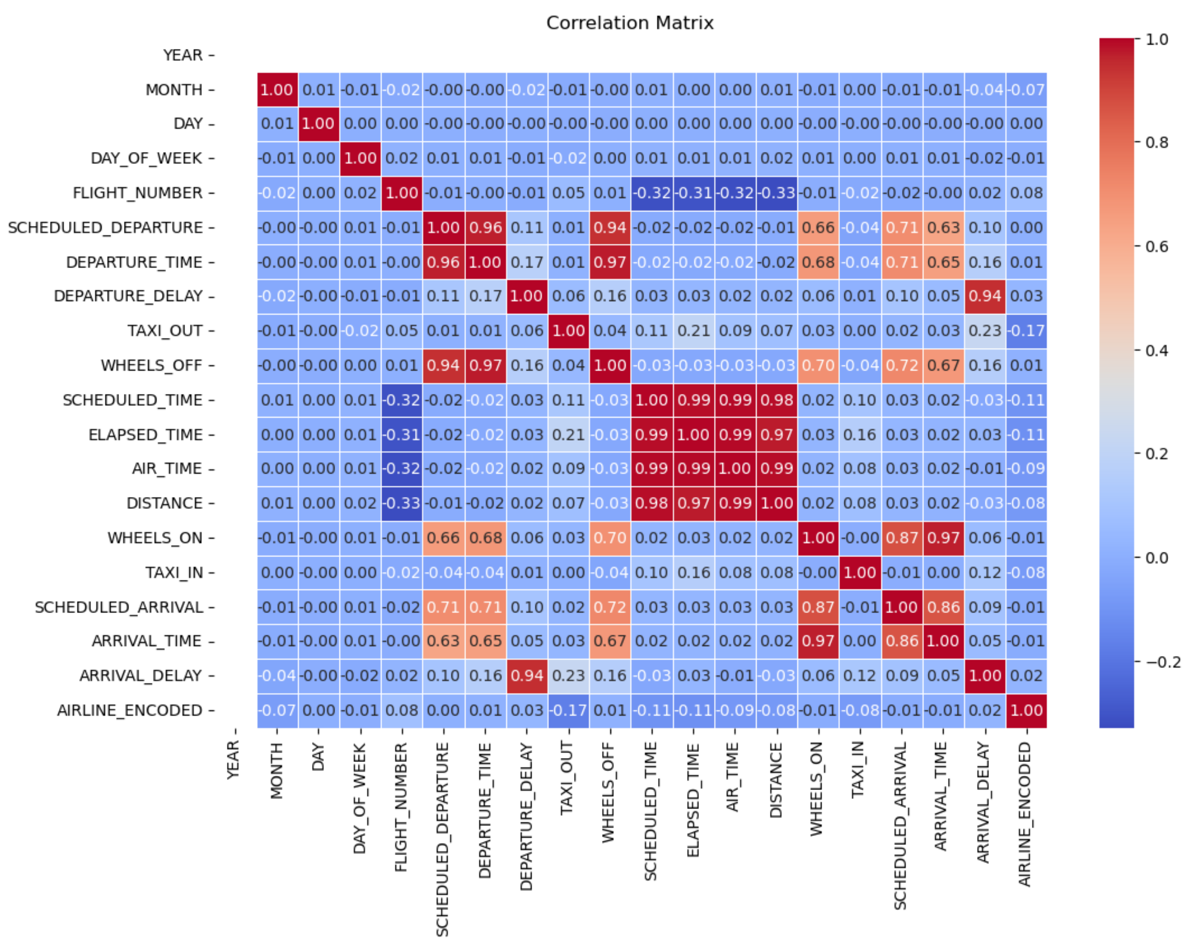
Dropped the columns because of there was no data present in them making them irrelevant to our project

In [ ]:
```
df.dropna(inplace=True)
```

Dropping rows with null values as there are maximum 105,000 rows with null values. Therefore it makes sense to drop these as we have a possible sample size of 5,000,000 rows even if we exclude the rows with null values

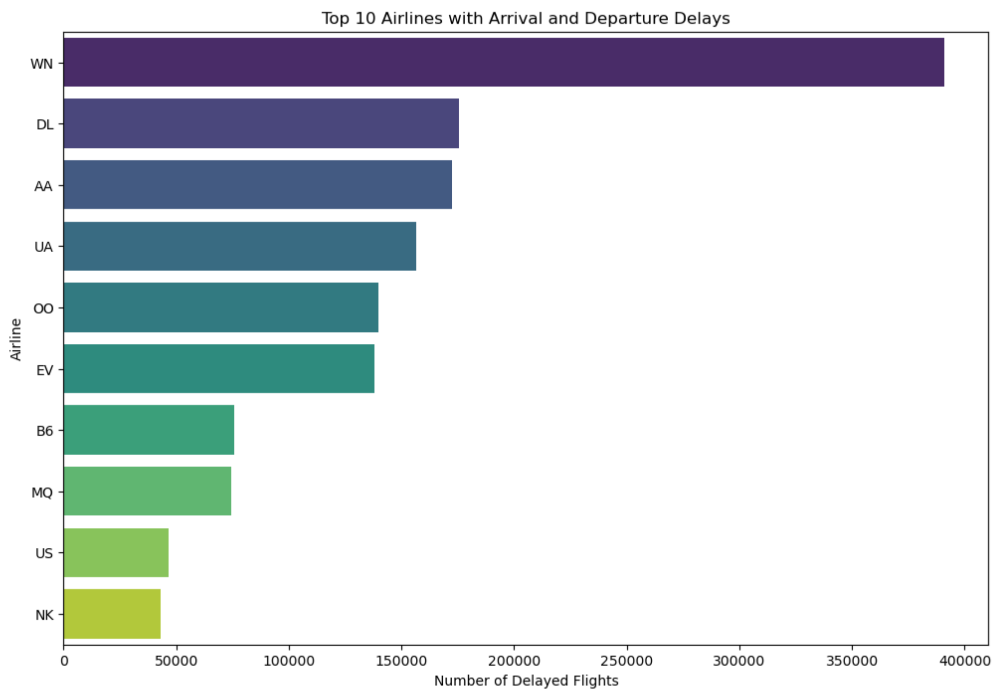
Data Visualization:

Visualization 1: Correlation Matrix



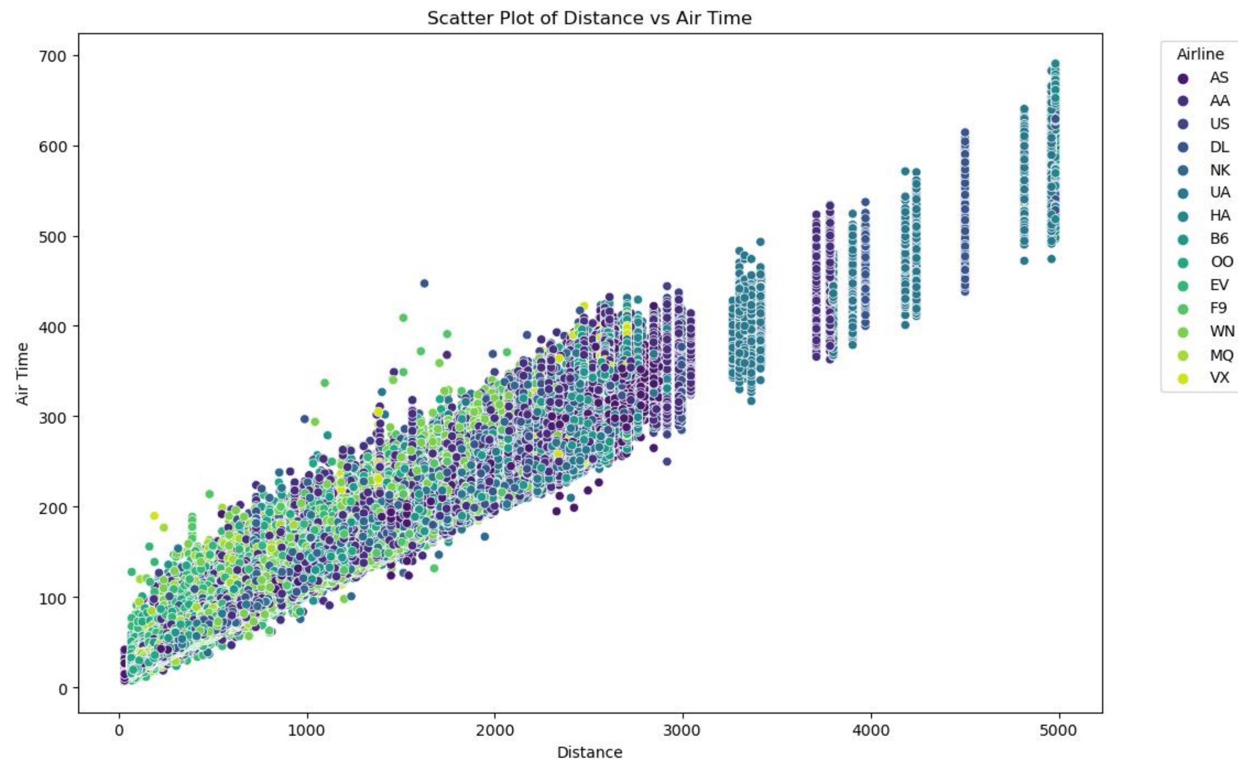
Correlation matrix to observe the relationships between the variables and develop an understanding of the interdependencies. It reveals that "SCHEDULED_DEPARTURE" has a significant correlation with several other time-related variables such as "WHEELS_OFF," "DEPARTURE_TIME," "WHEELS_ON," "SCHEDULED_ARRIVAL," and "ARRIVAL_TIME." This suggests a tight relationship between the scheduled departure time and various stages of the flight process, from takeoff to arrival. In contrast, "DEPARTURE_DELAY" shows a notable correlation only with "ARRIVAL_DELAY," indicating that delays in departure tend to correspond with delays in arrival.

Visualization 2: Top 10 Airlines with Arrival and Departure Delays



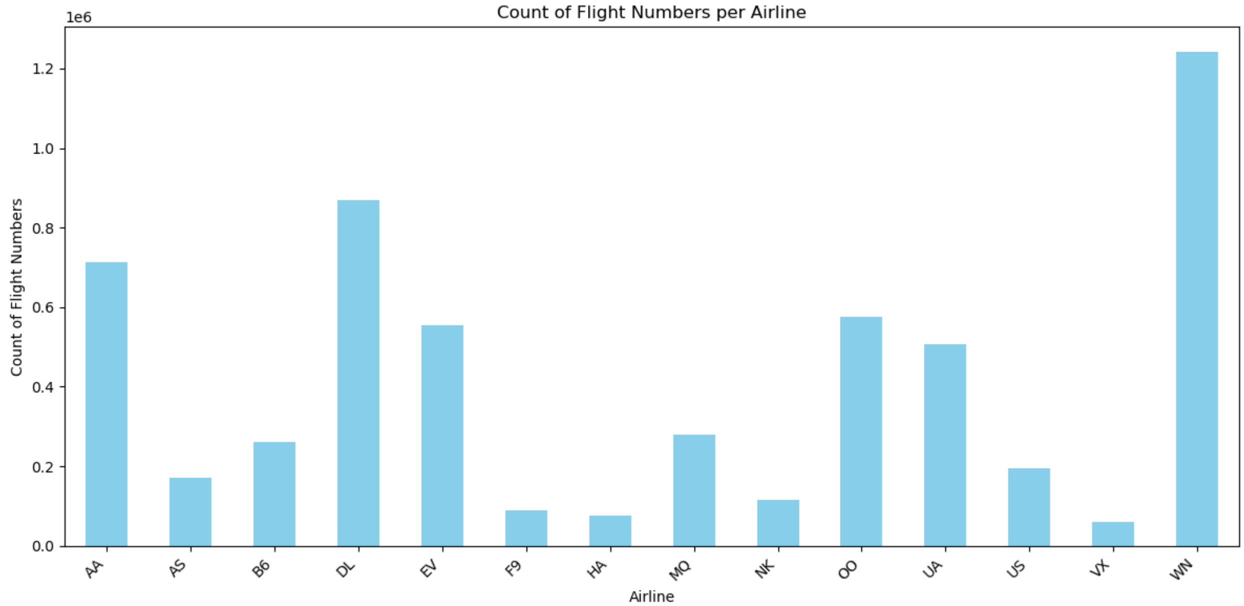
The horizontal bar chart provides a clear visualization of the frequency of delayed flights across different airlines. It underscores the significant variation in delay occurrences among the top 10 airlines. WN emerges as the airline with the highest number of delayed flights, nearing the 400,000 mark, which could be attributed to its extensive fleet and route network. Conversely, NK appears at the bottom of the list with a noticeably lower count of around 35,000 delayed flights. This disparity suggests varying levels of operational efficiency and management practices among airlines in handling and mitigating flight delays.

Visualization 3: Scatter Plot of Distance VS Air Time for Airlines



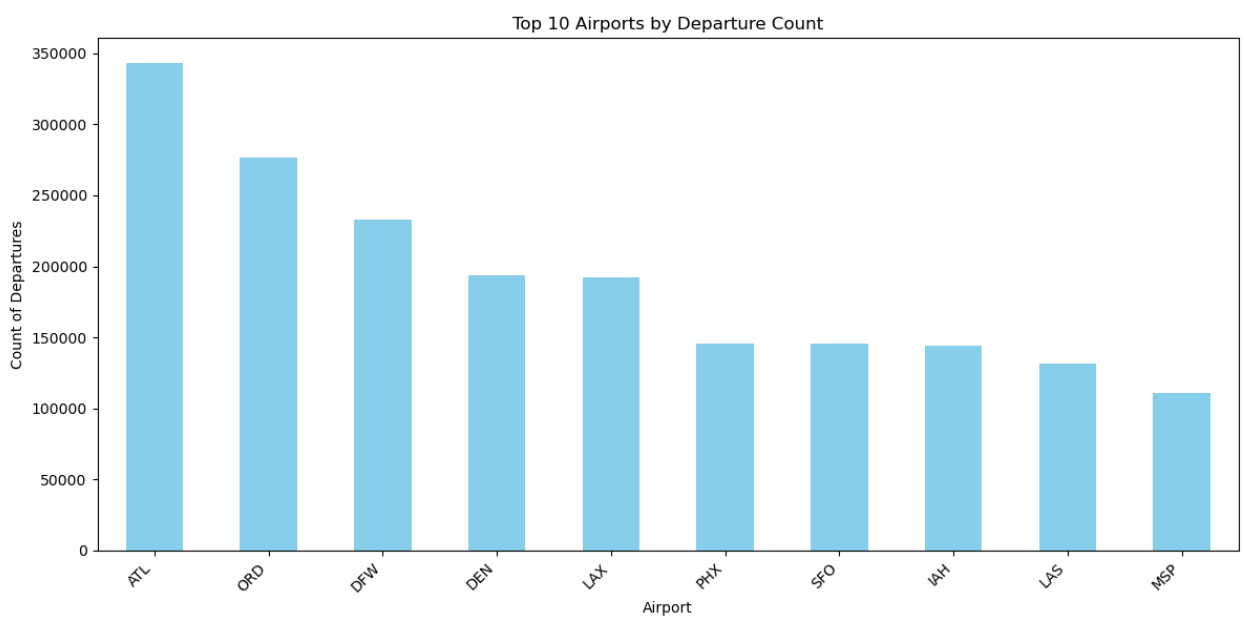
The scatter plot visually represents a linear correlation between the distance of a flight and its corresponding airtime. Notably, it demonstrates a consistent distribution pattern along both axes across all airlines included in the analysis. This uniform spread suggests that regardless of the airline, flights with longer distances tend to have proportionally longer airtime. Additionally, the absence of any noticeable clustering or outliers along the axes indicates a relatively stable relationship between flight distance and airtime across the dataset. This observation underscores the importance of considering flight distance as a key factor influencing airtime duration across various airline operations.

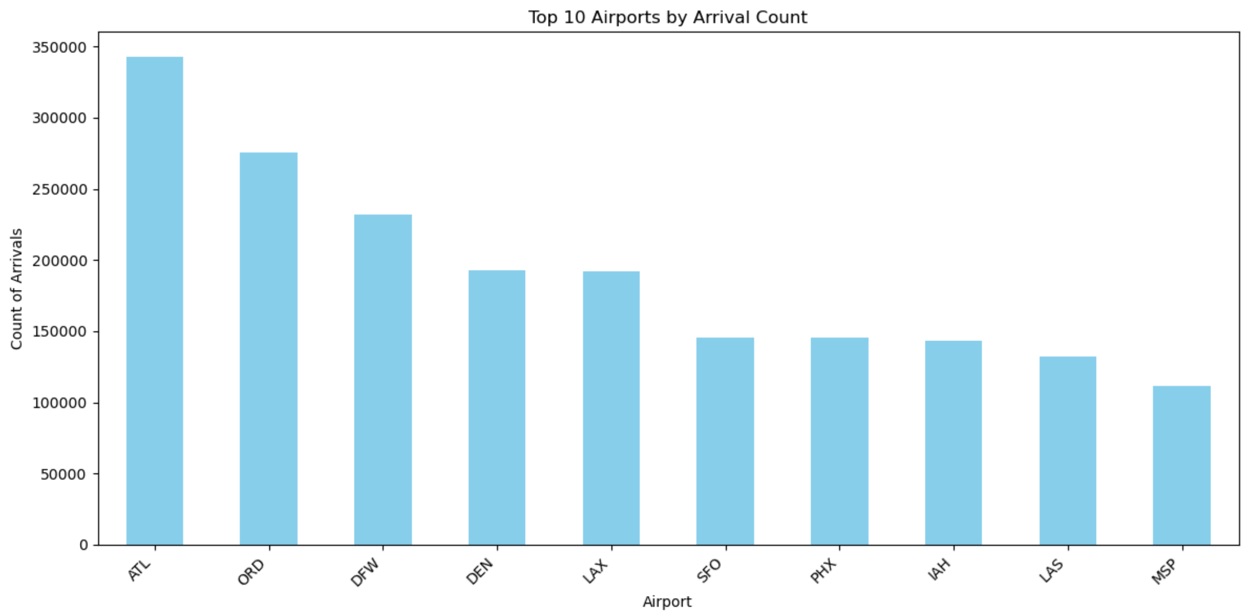
Visualization 4: Number of Airplanes owned by Airlines



The bar chart provides a visual representation of the number of flights operated by each airline, highlighting disparities in flight volumes across the dataset. This variation in the frequency of flights among different carriers may introduce challenges when applying regression and classification algorithms, as models may become biased towards airlines with larger datasets. WN emerges as the airline with the highest flight count, indicating a significant presence in the dataset, whereas VX stands out with the lowest number of flights recorded. This discrepancy underscores the importance of considering potential dataset imbalances and their implications on the analysis outcomes.

Visualization 5: Top 10 Airports by Departure and Arrival Count





The bar charts provide a comprehensive overview of airport activity, highlighting Atlanta as the focal point of flight traffic. Its dominance in both arrival and departure counts underscores its status as a major transportation hub. Despite the overall similarity between the two graphs, the slight discrepancy between departure counts for SFO and PHX, and arrival counts for PHX and SFO, reflects nuanced differences in flight scheduling and regional connectivity. This observation suggests diverse travel patterns and operational dynamics among the airports, contributing to the intricate network of air travel within the region. Understanding these variations is crucial for optimizing air traffic management and airport infrastructure planning.

Feature Engineering:

We started our feature engineering process by encoding the airline information seen in the 'AIRLINE' column. We also encoded the following columns along with airline columns as they contained alphanumeric data: ‘ORIGIN_AIRPORT’, “DESTINATION_AIRPORT’, ‘TAIL_NUMBER’.

```
In [11]: flight_df['ORIGIN_AIRPORT'] = flight_df['ORIGIN_AIRPORT'].astype(str)
         flight_df['DESTINATION_AIRPORT'] = flight_df['DESTINATION_AIRPORT'].astype(str)
         flight_df['TAIL_NUMBER'] = flight_df['TAIL_NUMBER'].astype(str)
         label_encoder = LabelEncoder()
         flight_df['ORIGIN_AIRPORT'] = label_encoder.fit_transform(flight_df['ORIGIN_AIRPORT'])
         flight_df['DESTINATION_AIRPORT'] = label_encoder.fit_transform(flight_df['DESTINATION_AIRPORT'])
         flight_df['TAIL_NUMBER'] = label_encoder.fit_transform(flight_df['TAIL_NUMBER'])
```

```python
In [ ]: from sklearn.preprocessing import LabelEncoder

        # Initialize LabelEncoder
        label_encoder = LabelEncoder()

        # Fit LabelEncoder and transform 'AIRLINE' column
        df['AIRLINE_ENCODED'] = label_encoder.fit_transform(df['AIRLINE'])

        # Get unique encoded values
        unique_encoded_airlines = df['AIRLINE_ENCODED'].unique()

        # Map encoded values back to original airline names
        airline_mapping = dict(zip(df['AIRLINE_ENCODED'], df['AIRLINE']))

        # Display unique encoded values and their corresponding airline names
        print("Unique Encoded Airline Values:")
        print(unique_encoded_airlines)

        print("\nMapping of Encoded Values to Airline Names:")
        print(airline_mapping)
```

Encoding Airline column to help us standardize the dataset to prepare it for feature selection and further processing

After that, we standardized our dataset using the StandardScaler() function. This conversion guarantees that every feature has a mean of 0 and a standard deviation of 1, which prepares our dataset for the best possible feature selection procedures later on in our machine learning project.

```python
In [ ]: scaler = StandardScaler()
        df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns[0:])
        display(df_scaled)
```

Standardizing the dataset

Feature Selection:

We then used feature selection approaches to determine which features would have the most influence on our machine learning models.

```python
In [ ]: x = df.drop(columns = ['ARRIVAL_DELAY',])
        y = df['ARRIVAL_DELAY']
        train_X, valid_X, train_y, valid_y = train_test_split(x,
```

Creating Test/Train split

We applied backward selection by defining 'ARRIVAL_DELAY' as our dependent variable.

In [16]:
```python
def train_model(variables):
 model = LinearRegression()
 model.fit(train_X[variables], train_y)
 return model
def score_model(model, variables):
 return AIC_score(train_y, model.predict(train_X[variables]), model)
allVariables = train_X.columns
best_model, best_variables = backward_elimination(allVariables, train_model,
 score_model, verbose=True)
print(best_variables)
regressionSummary(valid_y, best_model.predict(valid_X[best_variables]))
```

```
Variables: YEAR, MONTH, DAY, DAY_OF_WEEK, FLIGHT_NUMBER, TAIL_NUMBER, ORIGIN_AIRPORT, DESTINATION_AIRPORT, SCHEDULED_DEPARTURE,
DEPARTURE_TIME, DEPARTURE_DELAY, TAXI_OUT, WHEELS_OFF, SCHEDULED_TIME, ELAPSED_TIME, AIR_TIME, DISTANCE, WHEELS_ON, TAXI_IN, SC
HEDULED_ARRIVAL, ARRIVAL_TIME, AIRLINE_ENCODED
Start: score=-215847929.24
Step: score=-222369934.02, remove TAIL_NUMBER
Step: score=-223391701.21, remove TAXI_OUT
Step: score=-224927904.57, remove DISTANCE
Step: score=-226051160.83, remove DESTINATION_AIRPORT
Step: score=-227857594.57, remove AIRLINE_ENCODED
Step: score=-227857594.57, remove None
['YEAR', 'MONTH', 'DAY', 'DAY_OF_WEEK', 'FLIGHT_NUMBER', 'ORIGIN_AIRPORT', 'SCHEDULED_DEPARTURE', 'DEPARTURE_TIME', 'DEPARTURE_
DELAY', 'WHEELS_OFF', 'SCHEDULED_TIME', 'ELAPSED_TIME', 'AIR_TIME', 'WHEELS_ON', 'TAXI_IN', 'SCHEDULED_ARRIVAL', 'ARRIVAL_TIM
E']

Regression statistics

               Mean Error (ME) : 0.0000
Root Mean Squared Error (RMSE) : 0.0015
     Mean Absolute Error (MAE) : 0.0000
```

This method maximizes the performance of our model by methodically removing features until only the most essential ones remain.

Recommended Features:

```
['YEAR', 'MONTH', 'DAY', 'DAY_OF_WEEK', 'FLIGHT_NUMBER', 'ORIGIN_AIRPORT', 'SCHEDULED_DEPARTURE', 'DEPARTURE_TIME', 'DEPARTURE_
DELAY', 'WHEELS_OFF', 'SCHEDULED_TIME', 'ELAPSED_TIME', 'AIR_TIME', 'WHEELS_ON', 'TAXI_IN', 'SCHEDULED_ARRIVAL', 'ARRIVAL_TIM
E']
```

The characteristics "TAIL_NUMBER," "TAXI OUT," "DISTANCE," "DESTINATION_AIRPORT," and "AIRLINE_ENCODED" showed a lack of connection to the dependent variable "ARRIVAL_DELAY" for regression. As such, we decided to exclude these features in our model training procedure. Instead, in order to successfully predict the target variable, we will train our regression model using the remaining features.