Airtime Analytics and Prediction

Xinran Shen, Carina Zhang, Xingchen Li, Xinzhi Zhang, Zeqi Li

Background & Introduction

-Focusing on the examination and **forecasting of flight delays** through the utilization of historical records of flight operations and climatic data

-Integrating these data sources is key to identifying underlying trends and constructing a model capable of predicting upcoming delays,

-Improving the efficiency of travel planning and management processes

Data Overview

The primary dataset used is the "Flight Delay and Cancellation Dataset (2019-2023)" provided by the US Department of Transportation. This dataset contains approximately 30 million rows. We will only be focusing on the 2023 portion.

We will also be integrating this dataset with the following climate data from NOAA, taking local weather information of each flight into consideration when constructing the predictive model.

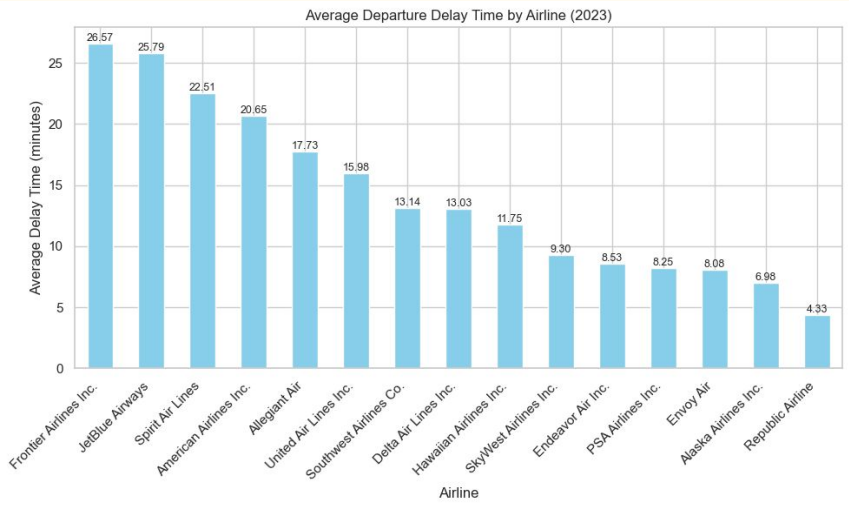
Flights Data Cleaning and Preprocessing for EDA

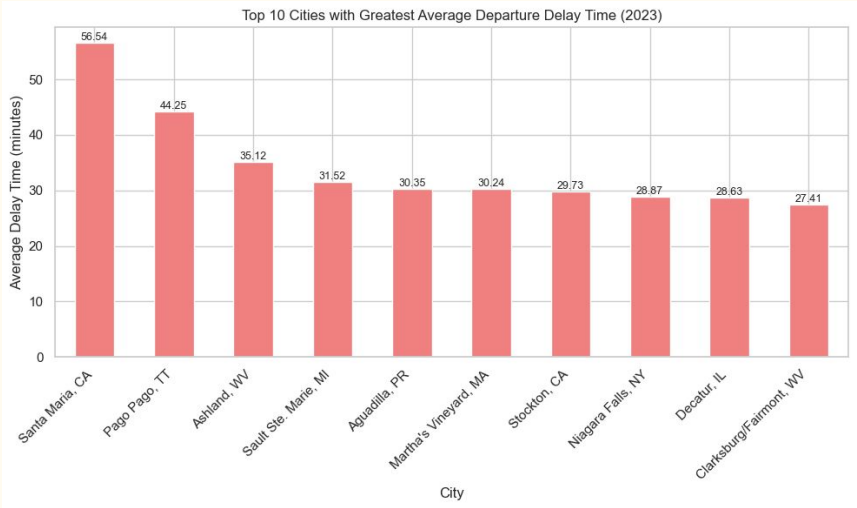
- Loaded in the dataset:
  - “2023.csv”: flight information from Jan 2023 - Aug 2023
  - "AIRLINE_CODE_DICTIONARY.csv": airline codes and their corresponding airline names
- Merged the datasets on airline code to display airline names
- Converted flight dates into Datetime format for easier analysis
- Dropped irrelevant columns (ie. dot_code, wheel off time, etc.)
- Renamed and reorder columns for better readability
- Sorted the Dataframe by date in ascending order
- Checked for null values in Dataframe, since we have a big dataset, we decided to drop the null entries for better analysis

Exploratory Data Analysis (EDA)

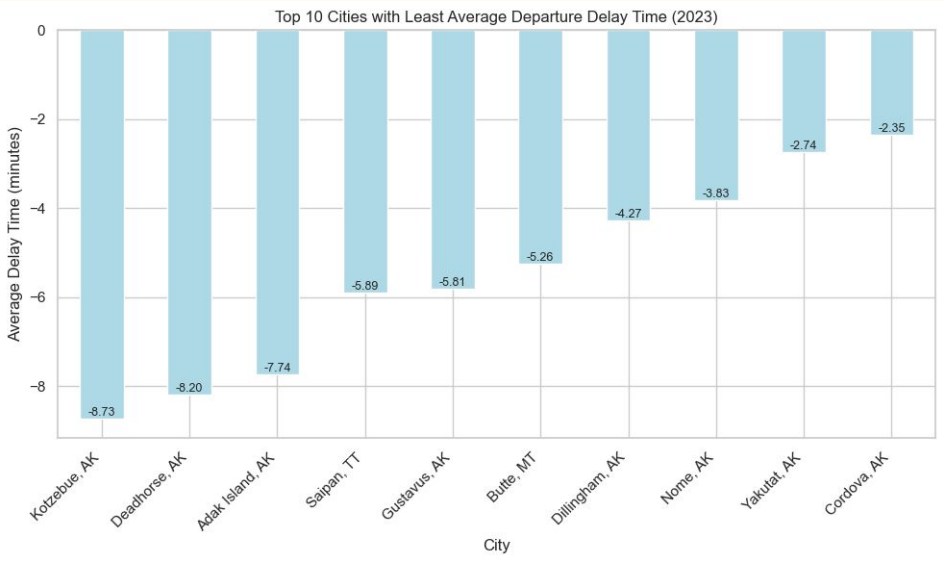
01	**Which airline has the most/least average departure delay time?**	• Ranking 15 airline companies by mean departure delay time • Deriving the most delayed/punctual airline
02	**Which city experiences the most/least average departure delay time?**	• Ranking cities by mean departure delay time • Displaying Top 10 most/least delayed cities • Focusing specifically on 12 major cities in the U.S.
03	**Which airport is the most/least affected by delays?**	• Ranking airports by mean departure delay time • Showcasing Top 10 most/least delayed airports • Comparing delay levels among the three airports in NY
04	**Which time period of a day is the most susceptible to significant delays?**	• Dividing daytime into 4 time periods • Grouping data and contrasting different time periods' delay levels

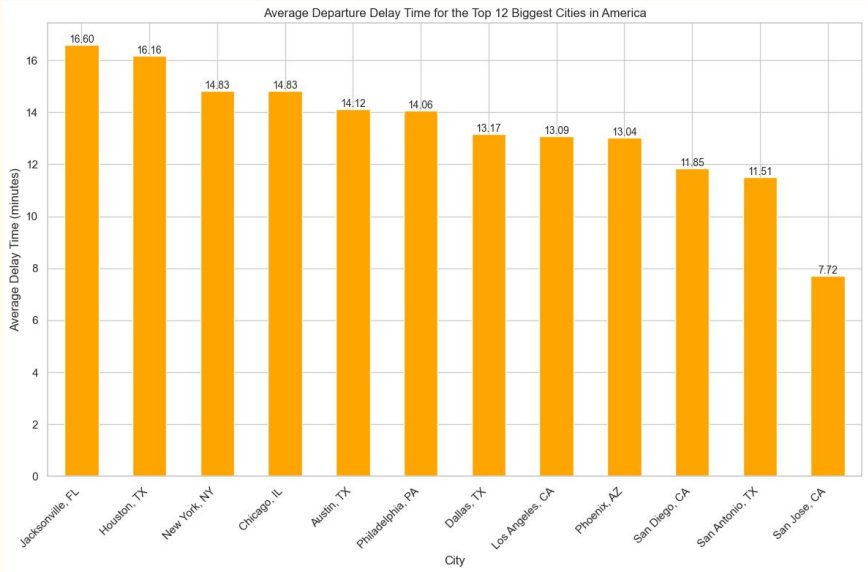
01. Delving into Airline Delays





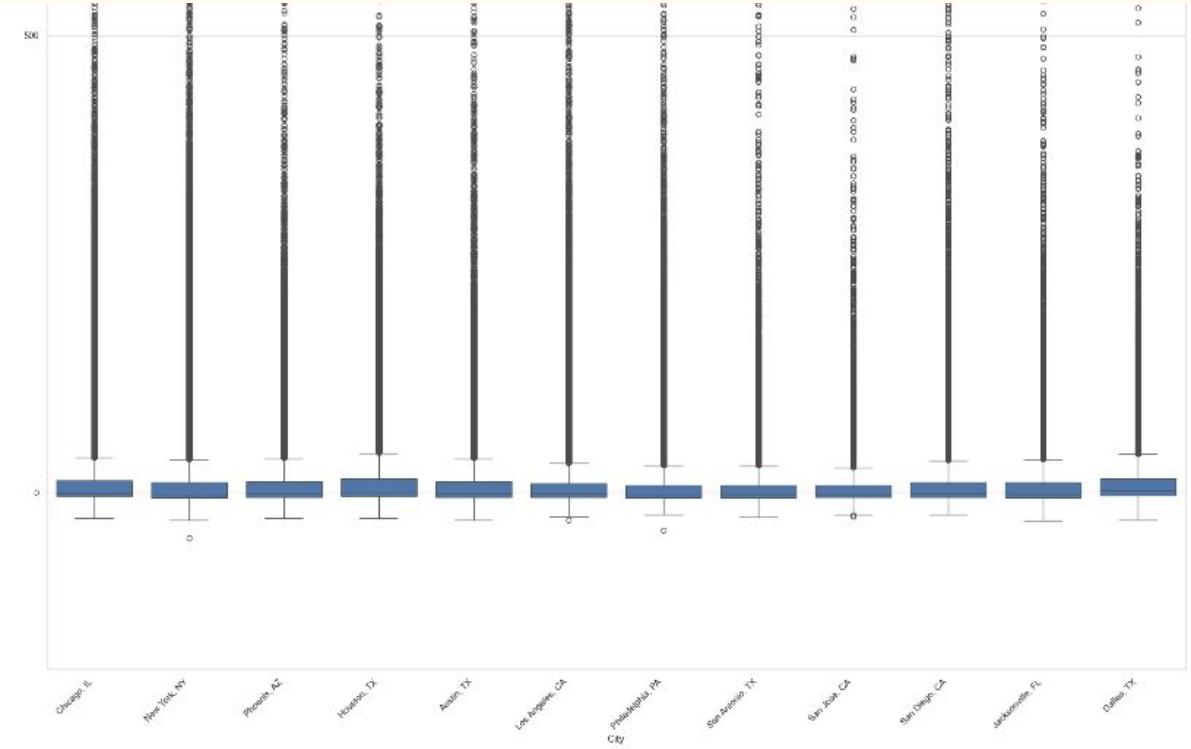
02. Understanding Delays by City





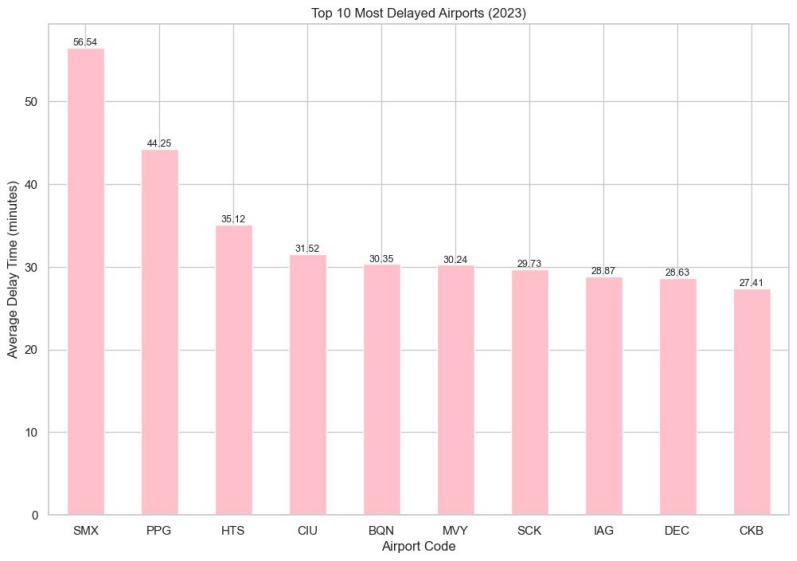
02. Understanding Delays by City

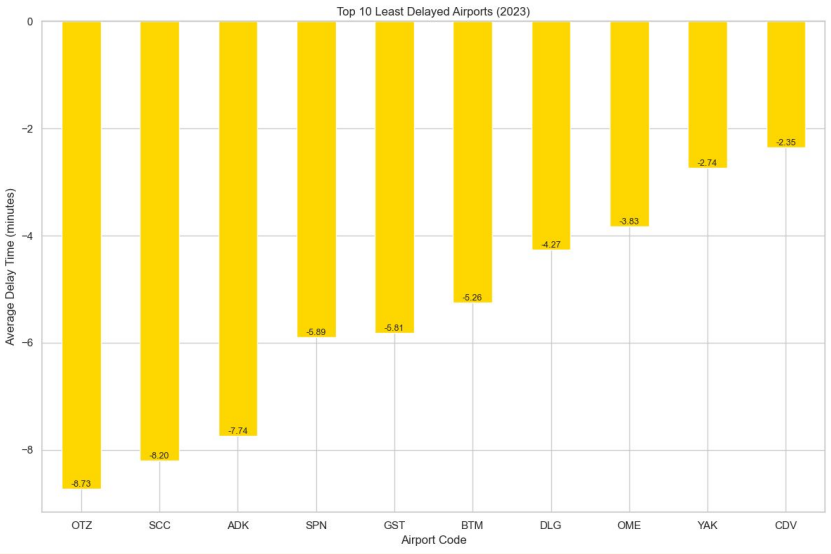
With a focus on the 12 largest cities by population in the U.S.



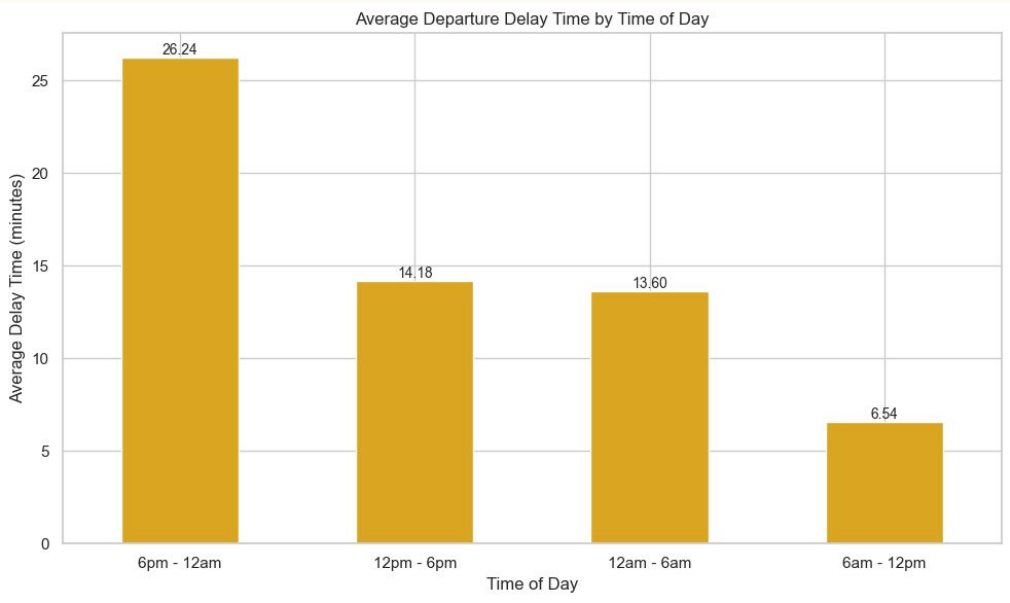
The central tendency (median, quartiles) of the box plot is not as prominent as the outliers. This indicates that most flights might be on time or have minor delays, but a few flights have very long delays, which skews the visualization.

03. Airport-Based Delay Insights





03. Airport-Based Delay Insights and Time of Day & Delays



How to predict delays given flight information and weather conditions?

Weather Data and Preprocessing

-Acquire: To predict flight delays, we need weather information from both the origins and destinations. For a place like New York, this means data from hundreds of climate stations. Instead of directly using the online search tool from NOAA, we called their API to do the bulk data downloading.

-Preprocess: PRCP, SNOW, SNWD, TMAX and TMIN are five core values of GHCN dataset. We drop the SNWD cause it contains too much NaN value. Besides, considering the availability and consistency of data across various stations and which data can influence the taking off and landing of flights, we also add AWND and WSF2 into our datasets. After dropping those stations which have NaN in line, we finally has a datasets coming from 247 stations, including New York.

Preprocessing

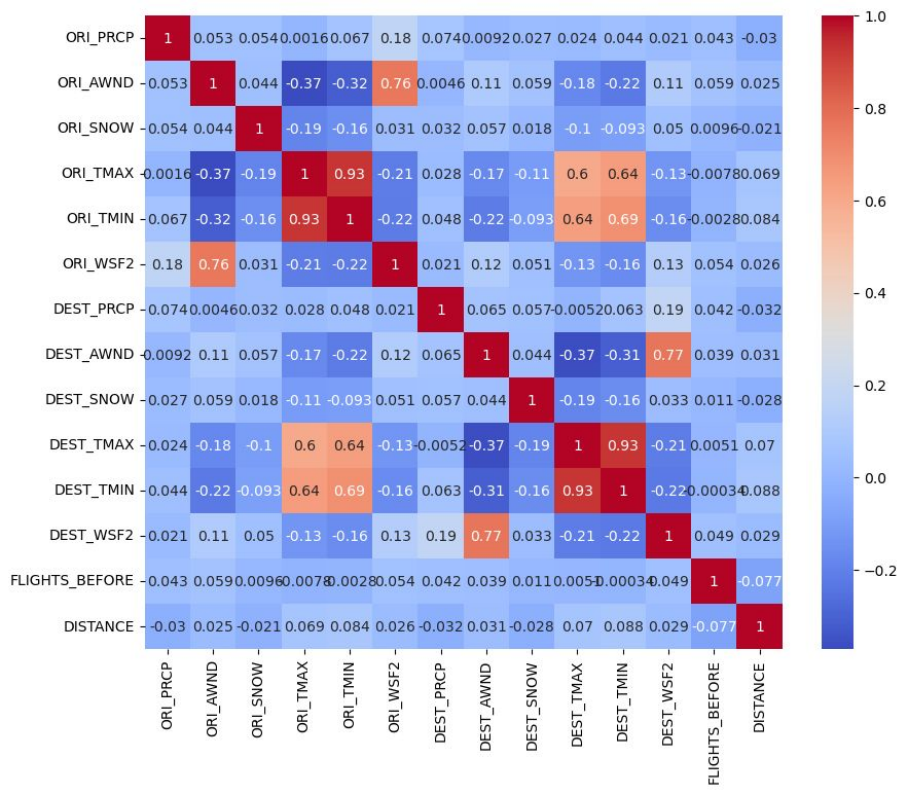
Due to the limitations of the scale of data we can process, we have focused our predictive data sample on flights where either the departure or arrival airport is located in New York.

Delayed flights vs. On-time flights: 28.57% vs. 71.43%. Unbalanced! Keep all the delayed flights and randomly sample same number of on-time flights. Merge the 2 datasets to build a balanced datasets.

Splitting the dataset into training and testing sets (80-20 split).

Standardizing the data using StandardScaler.

Corr Matrix


ORI_PRCP 1 0.053 0.054 0.0016 0.067 0.18 0.074 0.0092 0.027 0.024 0.044 0.021 0.043 -0.03
ORI_AWND 0.053 1 0.044 -0.37 -0.32 0.76 0.0046 0.11 0.059 -0.18 -0.22 0.11 0.059 0.025
ORI_SNOW 0.054 0.044 1 -0.19 -0.16 0.031 0.032 0.057 0.018 -0.1 -0.093 0.05 0.0096 -0.021
ORI_TMAX 0.0016 -0.37 -0.19 1 0.93 -0.21 0.028 -0.17 -0.11 0.6 0.64 -0.13 -0.0078 0.069
ORI_TMIN 0.067 -0.32 -0.16 0.93 1 -0.22 0.048 -0.22 -0.093 0.64 0.69 -0.16 -0.0028 0.084
ORI_WSF2 0.18 0.76 0.031 -0.21 -0.22 1 0.021 0.12 0.051 -0.13 -0.16 0.13 0.054 0.026
DEST_PRCP 0.074 0.0046 0.032 0.028 0.048 0.021 1 0.065 0.057 -0.0052 0.063 0.19 0.042 -0.032
DEST_AWND 0.0092 0.11 0.057 -0.17 -0.22 0.12 0.065 1 0.044 -0.37 -0.31 0.77 0.039 0.031
DEST_SNOW 0.027 0.059 0.018 -0.11 -0.093 0.051 0.057 0.044 1 -0.19 -0.16 0.033 0.011 -0.028
DEST_TMAX 0.024 -0.18 -0.1 0.6 0.64 -0.13 -0.0052 -0.37 -0.19 1 0.93 -0.21 0.0051 0.07
DEST_TMIN 0.044 -0.22 -0.093 0.64 0.69 -0.16 0.063 -0.31 -0.16 0.93 1 -0.22 0.00034 0.088
DEST_WSF2 0.021 0.11 0.05 -0.13 -0.16 0.13 0.19 0.77 0.033 -0.21 -0.22 1 0.049 0.029
FLIGHTS_BEFORE 0.043 0.059 0.0096 0.0078 0.0028 0.054 0.042 0.039 0.011 0.0051 0.00034 0.049 1 -0.077
DISTANCE -0.03 0.025 -0.021 0.069 0.084 0.026 -0.032 0.031 -0.028 0.07 0.088 0.029 -0.077 1
ORI_PRCP
ORI_AWND
ORI_SNOW
ORI_TMAX
ORI_TMIN
ORI_WSF2
DEST_PRCP
DEST_AWND
DEST_SNOW
DEST_TMAX
DEST_TMIN
DEST_WSF2
FLIGHTS_BEFORE
DISTANCE
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Feature Engineering

- Binarized the column dep_deplay into a binary code where 1 is when delay time is greater than or equal to 15 mins, 0 otherwise.
- Dummy-coded the 15 airlines and merged into df_newyork as binary columns
- Calculated the count of flights before each flight's departure time within a 30-minute window to take possible airport congestion into account
- Separated dates into specifica days of the week and dummy coded them
- Separated estimated departure time into 4 segments of the day and dummy coded them

Prediction Models

- Features (X):
  - 'ORI_PRCP', 'ORI_AWND', 'ORI_SNOW', 'ORI_TMAX', 'ORI_TMIN', 'ORI_WSF2', 'DEST_PRCP', 'DEST_AWND', 'DEST_SNOW', 'DEST_TMAX', 'DEST_TMIN', 'DEST_WSF2', 'FLIGHTS_BEFORE','AIRLINE_9E', 'AIRLINE_AA', 'AIRLINE_AS', 'AIRLINE_B6', 'AIRLINE_DL', 'AIRLINE_F9', 'AIRLINE_HA', 'AIRLINE_NK', 'AIRLINE_OO', 'AIRLINE_UA', 'AIRLINE_WN', 'AIRLINE_YX', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'
- Target Variable(Y)
  - 'DELAY': Binary, 1 if the delay is greater than 15 min, 0 otherwise

Model Selection

Support Vector Machine (SVM)

Strengths:

Effective in high-dimensional spaces.

Versatile with different kernel functions (linear, polynomial, radial basis function).

Considerations:

Sensitive to the choice of kernel and hyperparameters.

k-Nearest Neighbors (KNN)

Strengths:

Simple and intuitive.

No assumptions about the underlying data distribution.

Considerations:

Sensitive to the choice of the number of neighbors (k) and distance metric.

XGBoost

Strengths:

High performance and efficiency.

Handles missing data well.

Considerations:

Requires careful tuning of hyperparameters.

Random Forest

Strengths:

Robust and resistant to overfitting.

Handles unbalanced data well.

Considerations:

Number of trees and other hyperparameters impact performance.

Hyperparameter Tuning

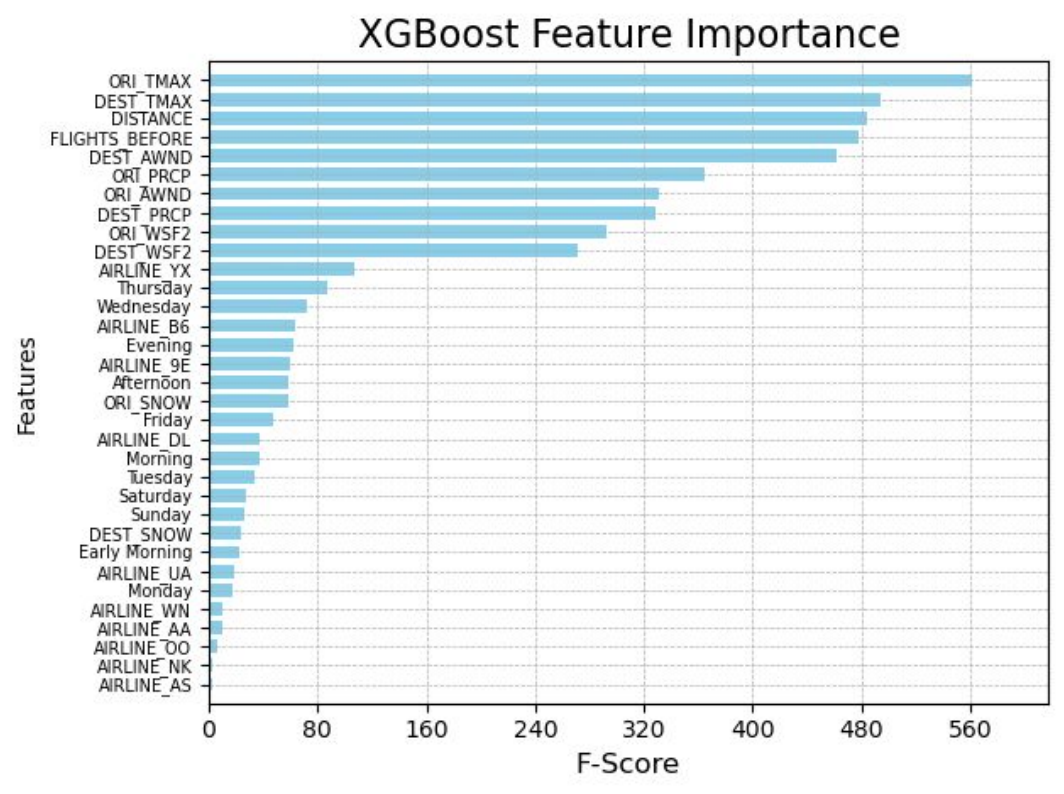
- Performed **grid search** for each model to find the best hyperparameters.
- Grid search helps optimize the model's performance.

  - **SVC**: svm = SVC(C = 0.0001, gamma= 0.0001, kernel='linear')
  - **KNN**: knn = KNeighborsClassifier(algorithm= 'auto', leaf_size= 10, n_neighbors= 50, p= 1, weights='distance')
  - **Xgboost**: xgb_classifier = XGBClassifier(colsample_bytree= 0.8, gamma=0, learning_rate= 0.1, max_depth=3, n_estimators= 100, subsample= 0.8)
  - **Random Forest**: rf_classifier = RandomForestClassifier(n_estimators=20, max_depth=10, min_samples_split=20)

Model Evaluation

	Model	Accuracy	Precision	Recall	F1 Score
0	SVM	0.800303	0.640484	0.800303	0.711530
1	KNN	0.791225	0.664714	0.791225	0.709719
2	XGBoost	0.797277	0.745047	0.797277	0.744224
3	Random Forest	0.804841	0.766162	0.804841	0.739095

Result


XGBoost Feature Importance
Features
ORI_TMAX
DEST_TMAX
DISTANCE
FLIGHTS_BEFORE
DEST_AWND
ORI_PRCP
ORI_AWND
DEST_PRCP
ORI_WSF2
DEST_WSF2
AIRLINE_YX
Thursday
Wednesday
AIRLINE_B6
Evening
AIRLINE_9E
Afternoon
ORI_SNOW
Friday
AIRLINE_DL
Morning
Tuesday
Saturday
Sunday
DEST_SNOW
Early Morning
AIRLINE_UA
Monday
AIRLINE_WN
AIRLINE_AA
AIRLINE_OO
AIRLINE_NK
AIRLINE_AS
0
80
160
240
320
400
480
560
F-Score