Flight Delay Analysis and Prediction Using 2024 US Flight Data

Pratik Narendra Gupta
Western University
London, ON, Canada
pgupta85@uwo.ca

Luke Blommesteyn
Western University
London, ON, Canada
lblommes@uwo.ca

Augusts Zilakovs
Western University
London, ON, Canada
azilakov@uwo.ca

Justin Zomer
Western University
London, ON, Canada
jzomer@uwo.ca

***Abstract*—This report presents a comprehensive analysis of the 2024 US domestic flight dataset to identify patterns in delay behavior and develop predictive models. By leveraging a dataset of approximately 7 million flights, we implemented a data science pipeline including preprocessing, exploratory data analysis (EDA), and machine learning modeling. Our approach utilized Isolation Forests for anomaly detection and XGBoost for delay prediction. The results highlight significant variation in airline reliability and demonstrate that while identifying individual flight delays remains challenging (ROC-AUC $\approx$ 0.69), aggregate daily delay trends can be predicted with high accuracy ($R^2 \approx$ 0.78) using rolling window features.**

***Index Terms*—Flight Delay Prediction, XGBoost, Anomaly Detection, Machine Learning, Data Science.**

I. Introduction

Flight delays represent a continuous challenge within the global aviation industry. The US Bureau of Transportation Statistics (BTS) reported that in 2024, approximately 236 million passengers were affected by delays or cancellations, with an overall on-time arrival rate of 78.1% [1]. Major disruptions in 2024 included the global IT outage caused by a CrowdStrike update in July and severe weather events like Hurricane Debby [2].

The primary objective of this project is to identify key factors contributing to departure and arrival delays in US domestic flights during 2024 and to construct predictive models that estimate the likelihood of delay. The motivation extends beyond descriptive analytics; accurate predictions can support airline efficiency and improve passenger experiences.

This report outlines our methodology, from raw data ingestion to the deployment of advanced machine learning models. We address the problem through three specific goals:

1) **Characterization**: Quantifying delay distributions and identifying carrier-specific reliability.
2) **Detection**: Using unsupervised learning to identify and remove anomalous flight records.
3) **Prediction**: Training supervised models to forecast delays at both the individual flight level and the aggregate daily level.

The complete source code and implementation details are available in our Git repository: https://github.com/lblommesteyn/ds3000_project.

II. Dataset Description

The analysis utilizes the **2024 Domestic Flights Dataset** sourced from Kaggle. This comprehensive dataset contains detailed records of scheduled and actual departure/arrival times, carrier identifiers, flight numbers, and origin/destination airports.

A. Key Attributes

The raw dataset comprises approximately 7,079,081 flight records with 35 columns, including:

- **Temporal**: `fl_date`, `crs_dep_time`, `dep_time`, `crs_arr_time`.
- **Categorical**: `op_unique_carrier` (Airline Code), `origin`, `dest`.
- **Numerical**: `dep_delay`, `arr_delay`, `air_time`, `distance`.
- **Labels**: Binary delay indicators were derived for supervised learning, where a flight is considered "delayed" if it arrives more than 15 minutes past its scheduled time.

$$y = \begin{cases} 1, & \text{if } \texttt{arr\_delay} > 15 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

B. Data Quality and Preprocessing

Initial inspection revealed issues common in large-scale operational data, such as missing values for cancelled flights and inconsistencies in time formats. Our preprocessing pipeline included:

1) **Standardization**: Converting HHMM integer time formats into standard datetime objects.
2) **Imputation**: Missing values for delay fields were handled based on carrier and route-specific historical medians, while cancelled flights were flagged separately.
3) **Outlier Removal**: Logical filters were applied to remove impossible values (e.g., negative air times).

III. Methodology

Our project methodology followed a structured data science lifecycle: Exploratory Data Analysis (EDA), Anomaly Detection, and Predictive Modeling.

A. Exploratory Data Analysis (EDA)

We conducted extensive EDA to understand the underlying distributions of the data. This involved generating statistical summaries of delay variation across hours of the day, days of the week, and seasonal trends. We specifically analyzed carrier performance to identify the most and least reliable airlines.

Figure 1 shows a clear increase in delays throughout the day, with afternoon and evening flights experiencing the largest slowdowns.

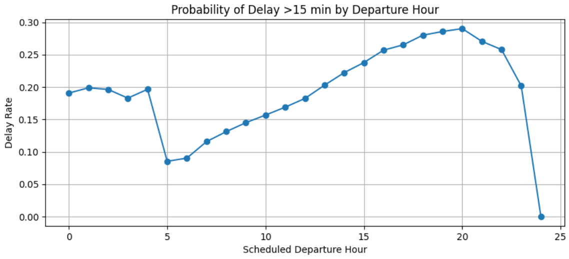

Fig. 1. Average delay by hour of the day, highlighting the accumulation of delays as the day progresses.

B. Anomaly Detection (Isolation Forest)

To improve the robustness of our predictive models, we employed an **Isolation Forest** algorithm. This unsupervised learning technique isolates anomalies by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

- **Input**: Scaled features including `dep_delay`, `air_time`, and `distance`.
- **Outcome**: The model identified records with extreme deviation from normal operations, such as data entry errors or extreme weather events irrelevant to standard modeling.

C. Predictive Modeling (XGBoost)

We utilized **XGBoost (Extreme Gradient Boosting)** for the classification task of predicting flight delays. XGBoost was selected for its efficiency and ability to handle non-linear relationships and interactions between features.

- **Baseline Model**: Trained on standard flight metadata.
- **No-Leakage Model**: A strict time-based split was enforced to prevent data leakage (using future information to predict the past).
- **Evaluation**: Models were evaluated using ROC-AUC, Precision, Recall, and F1-score.

D. Time-Series Forecasting (Rolling/Lag Models)

Recognizing the time-based nature of delays, we implemented a “Rolling Stone” prediction model. This approach aggregates delays on a daily basis and uses rolling averages (e.g., 3-day and 7-day lag) and standard deviations (`std_delay`) as features to predict the average delay for subsequent days.

Let D_t denote the mean arrival delay on day t. Rolling features are defined as:

$$\text{3-day average:} \quad \bar{D}_t^{(3)} = \frac{1}{3}\sum_{i=1}^{3} D_{t-i} \tag{2}$$

$$\text{7-day average:} \quad \bar{D}_t^{(7)} = \frac{1}{7}\sum_{i=1}^{7} D_{t-i} \tag{3}$$

The regression model then estimates:

$$\hat{D}_t = f(\sigma_{t-1}, \bar{D}_t^{(3)}, \bar{D}_t^{(7)}, N_t, c_t, d_t) \tag{4}$$

where σ_{t-1} is the historical delay standard deviation, N_t is daily flight volume, and c_t, d_t represent calendar effects.

E. Data Leakage and Model Refinement

During development, an early XGBoost model (`inflated_xgboost.py`) achieved an unusually high ROC-AUC of 0.938. Upon further inspection, this performance was determined to be the result of **data leakage**. Specifically, the model included post-departure features such as `late_aircraft_delay` and `nas_delay`, which directly encode information that would not be available at prediction time.

While these variables are informative for retrospective analysis, their inclusion violates the fundamental constraint of predictive modeling: only information available *before departure* may be used. As a result, this model was deemed invalid for real-world deployment.

To address this issue, a refined “No-Leakage” XGBoost model was constructed using only strict pre-departure features:

- Scheduled departure hour
- Month and day of week
- Distance bucket
- Weekend and peak-summer indicators
- Origin, destination, and carrier

Additionally, class imbalance was corrected using `scale_pos_weight`. This refined model achieved a more realistic ROC-AUC of approximately 0.69. Although numerically lower, this performance accurately reflects the inherent uncertainty of predicting individual flight delays without real-time weather or operational data. This refinement step was critical in ensuring scientific validity and preventing misleading model conclusions.

IV. Results

A. Carrier Reliability Analysis

Our analysis of over 7 million flights identified distinct tiers of airline reliability. The top 5 airlines by on-time rate were:

1) **YX (Republic Airways)**: 84.7%
2) **HA (Hawaiian Airlines)**: 84.5%
3) **9E (Endeavor Air)**: 82.5%
4) **DL (Delta Air Lines)**: 82.4%
5) **OO (SkyWest Airlines)**: 80.4%

These findings align with industry reports indicating Delta (DL) as a top-performing major carrier in 2024 [3]. Conversely, carriers such as **OH (PSA Airlines)** and **G4 (Allegiant Air)** showed lower reliability, with on-time rates around 77%.

B. Anomaly Detection

The Isolation Forest model successfully detected **104,479 anomalies**. The primary causes for flagging were:

- **Extreme Departure Delay**: 85,280 instances.
- **Abnormally Long Air Time**: 17,427 instances.
- **Abnormally Short Air Time**: 963 instances.

Removing these outliers ensured that our supervised models were trained on representative operational data.

C. Classification Performance (XGBoost)

The classification task of predicting individual flight delays proved challenging due to the high variance and random nature of individual flight operations.

- **Baseline XGBoost ROC-AUC**: 0.7039
- **No-Leakage XGBoost ROC-AUC**: 0.6923

TABLE I
COMPARISON OF XGBOOST MODELS WITH AND WITHOUT DATA LEAKAGE

Model	**ROC-AUC**	**Validity**
Inflated XGBoost (With Leakage)	0.938	Invalid
No-Leakage XGBoost (Clean)	0.692	Valid

Detailed classification metrics for the No-Leakage model:

- **Class 0 (On-time)**: Precision 0.87, Recall 0.65.
- **Class 1 (Delayed)**: Precision 0.31, Recall 0.62.

While the model captures 62% of delayed flights (Recall), the low precision (31%) indicates a high False Positive Rate, suggesting that while the model learns signal, individual delays are heavily influenced by unobserved variables (e.g., real-time weather, maintenance).

As shown in Figure 2, the standard deviation of previous delays is the strongest predictor in the daily forecasting model.

D. Daily Aggregate Prediction

The daily delay prediction model yielded significantly stronger results, validating the presence of clear time-based trends.

The model's ability to follow real delay trends is illustrated in Figure 3, which shows predicted versus actual daily average delays.

- **R-squared** (R^2): 0.786
- **Mean Absolute Error (MAE)**: 2.67 minutes
- **ROC-AUC (Daily Classification)**: 0.975

Feature Importance: The most influential feature was `std_delay` (Standard Deviation of Delay), followed by `delay_3d_avg` and `flight_count`. This indicates that the volatility of delays in the preceding days is a strong predictor of future system performance.

V. CONCLUSION AND DISCUSSION

This study demonstrates the effectiveness of machine learning in analyzing aviation data. We successfully processed a massive dataset of 7 million flights to benchmark carrier performance and detect operational anomalies.

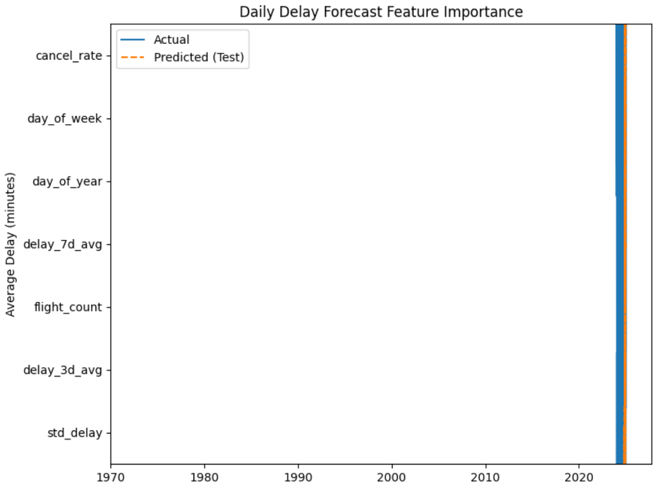

Fig. 2. Feature Importance for Daily Delay Prediction. Standard deviation of previous delays is the dominant predictor.

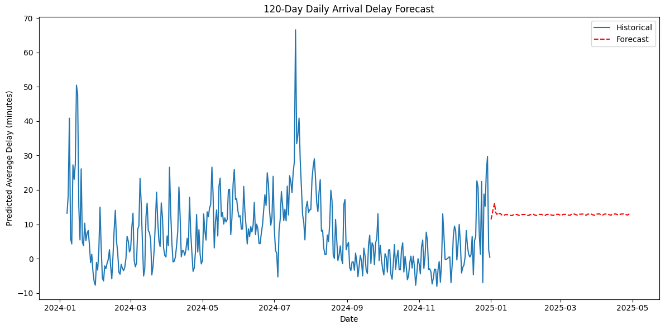

Fig. 3. Forecast of Daily Average Delays showing the model's ability to track trends.

A key lesson from this study was the critical importance of preventing data leakage. An early high-performing model ended up being rejected after identifying that post-flight delay variables had been inadvertently included. The corrective redesign of this feature space led to a lower but more useful analysis.

Our predictive modeling results suggest a clear difference in predictability:

1) **Individual Flight Level**: Predicting specific flight delays is difficult (AUC $\approx$ 0.69) using only schedule and carrier data. The random nature of mechanical issues and local weather creates a high "noise" floor.
2) **System Level**: Aggregated daily delays are highly predictable ($R^2 \approx 0.79$). The strong performance of rolling lag features implies that delays are systemic and persistent; a bad day tends to have cascading effects that are measurable in the statistical variance of the network.

Next Steps: Future work should focus on integrating real-time weather APIs and incoming aircraft tracking to improve individual flight prediction. Additionally, analyzing airport-specific congestion metrics could further refine the models.

Appendix

Appendix A: Individual Contributions

Each team member contributed to specific components of the project as outlined below:

Justin Zomer

Justin was responsible for the development and evaluation of the XGBoost classification models. This included:

- Implementing the initial high-performing `inflated_xgboost` model.
- Identifying and diagnosing data leakage caused by post-flight features.
- Designing and training the final `no_leakage_xgboost` model using only valid pre-departure features.
- Conducting classification evaluation using ROC-AUC, precision, recall, and F1-score.

Augusts Zilakovs

Augusts developed the time-series forecasting models used for daily delay prediction. His contributions included:

- Implementing the lag-based daily delay prediction model.
- Developing the "Rolling Stone" rolling-average forecasting model.
- Generating multi-day delay forecasts.
- Evaluating regression performance using MAE, R^2, and ROC-AUC after converting predictions to classification outputs.

Pratik Narendra Gupta

Pratik led the data preprocessing and cleaning pipeline. His responsibilities included:

- Loading and optimizing the 7-million-row dataset for memory-efficient processing.
- Creating feature engineering functions such as departure hour extraction and delay labeling.
- Handling missing values, invalid records, and data type optimization.
- Building reusable preprocessing utilities used across all models.

Luke Blommesteyn

Luke was responsible for exploratory data analysis, airline performance analysis, and anomaly detection. His work included:

- Performing airline reliability analysis and ranking carriers by on-time performance.
- Conducting route, distance, and traffic-based flight analysis.
- Implementing the Isolation Forest anomaly detection model.
- Analyzing root causes of flight delays by airline and by month.

Appendix B: Use of Generative AI Tools

Generative AI tools were used in this project for repetitive tasks such as assisting with document formatting and generating boilerplate code. All data analysis, algorithm implementation, and interpretation of results were performed manually. Experimental work and conclusions presented in this report were done directly by team members.

References

[1] US Department of Transportation, "Air Travel Consumer Report: September 2024 and Full Year 2024 Numbers," 2024. [Online]. Available: https://www.transportation.gov/.

[2] AirHelp, "US Fight Delays 2024: Statistics and Causes," 2024. [Online].

[3] Cirium, "The On-Time Performance Review 2024: Airlines and Airports," 2024.

[4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. of the 22nd ACM SIGKDD*, 2016.

[5] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," in *Eighth IEEE International Conference on Data Mining*, 2008.

[6] Kaggle, "2024 US Domestic Flights Dataset," 2024. [Online].