Predicting Domestic Flight Delays in the U.S

Final Report

By: Shahzaib Jahanzeb

Flight delays are a major operational challenge for airlines and airports across the United States. Each year, delays cost the aviation industry billions of dollars in lost productivity, crew misalignment, and passenger disruption. According to industry estimates, delays lead to roughly $33 billion in economic losses. Beyond the financial impact, delays also lower customer satisfaction and create inefficiencies throughout the transportation system. Because of how frequent and costly these delays are, being able to *predict* whether a scheduled flight will be delayed has strong practical value for airlines, airports, and travelers.

The goal of this project is to build a machine learning model that can predict whether a flight will be delayed upon arrival. The prediction is formulated as a binary classification problem, where a flight is labeled as *delayed* if its arrival is more than 15 minutes late, following the FAA standard for a delayed flight. This 15-minute threshold is widely used in industry, and it provides a clear and interpretable output for operational decision-making.

To carry out this prediction task, I used a dataset of U.S. domestic commercial flights from 2019 to 2023, sourced from the U.S. Department of Transportation. The *inputs* to the model include features such as scheduled departure time, actual elapsed time, air time, taxi-in and taxi-out durations, day and month of travel, distance flown, and airline carrier indicators. The *output* is a binary value representing whether the flight arrived late under the FAA definition.

This project explores three supervised learning methods to predict delays: a Bagging classifier, a baseline XGBoost model, and a tuned XGBoost model with optimized learning rate and number of boosting rounds. The objective is not only to achieve strong predictive accuracy but also to understand which factors contribute most to delays and how machine learning can support real-world operational improvements.

Related Work

One of the main studies that guided this project was a paper by Li and Jing that explored flight delays as part of a larger air transport network. Their work showed that delays do not happen in isolation. Instead, airports and flights are connected in a network where a delay at one point can spread throughout the system. They used historical flight data and network modeling to simulate how delays develop and travel through different airports. This approach achieved 97.2% accuracy, outperforming previous models and demonstrating the effectiveness of advanced preprocessing and optimization techniques for flight delay classification.

A second study by the same authors expanded on this idea by generating delay scenarios based on actual air traffic behavior before predicting future delays. They focused on how delays evolve and how early disruptions create downstream effects later in the day. This helped show that flight delay prediction improves when models consider both local information about a single flight and broader patterns happening across the entire air traffic network.

In addition to academic work, many practitioners on Kaggle have tackled this problem using the same U.S. DOT dataset. Their approaches often rely heavily on gradient boosting methods like XGBoost because of how well it performs on structured data. Kaggle projects also emphasize the importance of careful preprocessing and feature engineering, especially around time variables and operational factors such as taxi times and departure hours.

Together, these studies and applied examples helped shape this project by showing what types of features matter, how delays behave in real-world networks, and which model families tend to perform well for this type of classification problem.

Data Description

The dataset used in this project comes from the U.S. Department of Transportation's Bureau of Transportation Statistics (via Kaggle). It contains detailed operational records for U.S. domestic flights from 2019 to 2023. After filtering for completeness and relevance, a working sample of 150,000 flights was created for modeling.

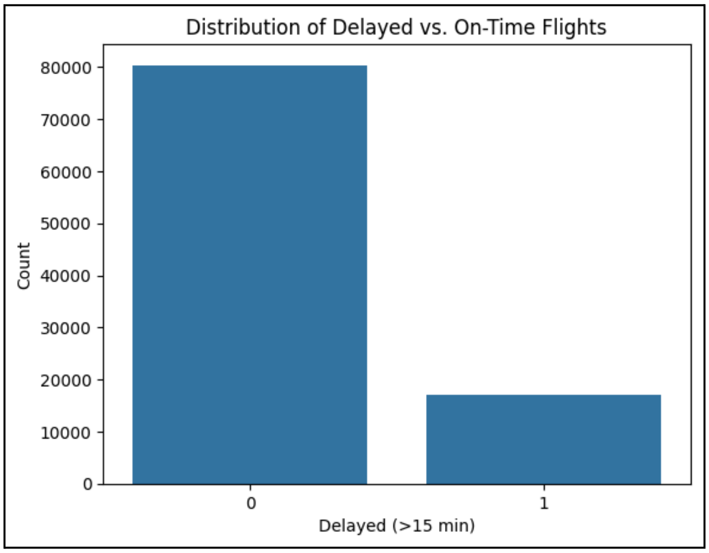
Each observation includes variables describing the scheduled and actual behavior of the flight, such as departure time, arrival time, taxi durations, air time, and cause-of-delay indicators. The target variable, *DELAYED*, follows the FAA definition: a flight is labeled 1 if it arrived more than 15 minutes late, and 0 otherwise. This binary structure makes the problem well-suited to supervised classification.

Basic exploratory analysis reveals several important characteristics of the data. First, delays are relatively less frequent than on-time arrivals, which introduces a mild class imbalance. Second, the delay rate varies meaningfully across airlines, suggesting that carrier-specific operational patterns contribute to performance differences. Third, a correlation heatmap of numerical predictors highlights strong relationships among time-related features such as elapsed time, air time, and taxi durations. These correlations are expected, as these measures are components of total flight time, but they also indicate that multicollinearity will be present and should be handled carefully in modeling.

Overall, the dataset provides a rich combination of operational, temporal, and carrier-level information that is suitable for building predictive models of flight delays.

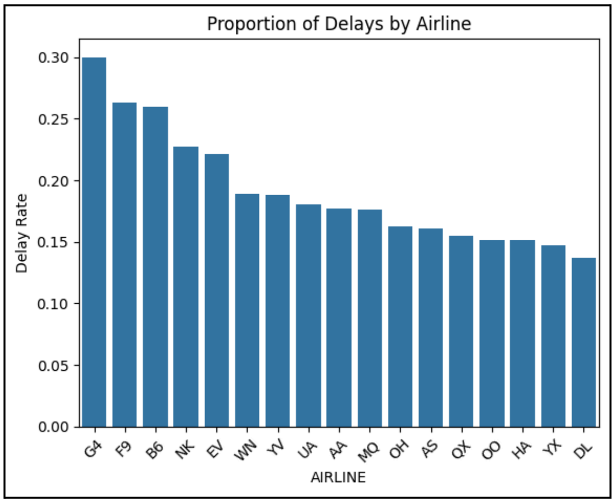
Figures

Distribution of Delayed vs. On-Time Flights



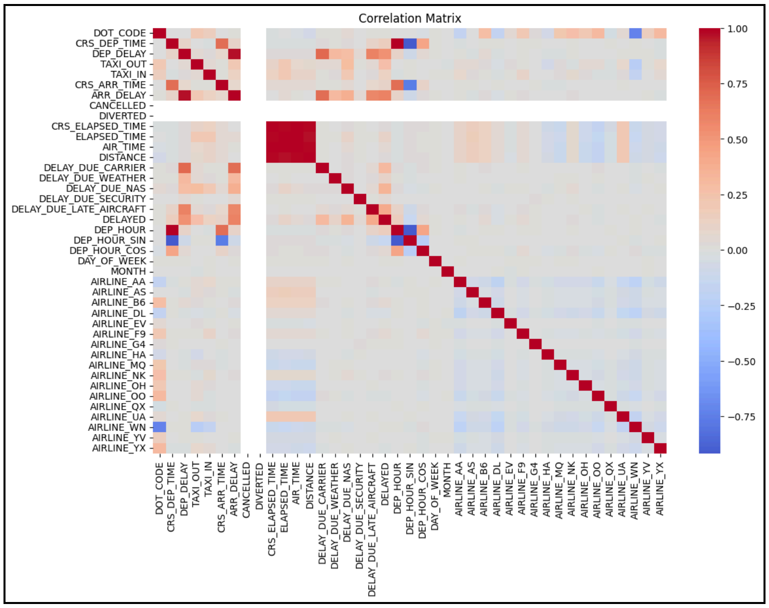
This bar chart shows that most flights in the dataset arrive on time, while a smaller portion experience an arrival delay of more than 15 minutes. This imbalance is important for choosing evaluation metrics and comparing model performance.

Proportion of Delays by Airline



This visualization highlights how delay rates differ significantly across carriers. Some airlines show delay rates near 25-30%, while others operate closer to 14–16%. This variation suggests that airline identity provides a predictive signal for modeling.

Correlation Matrix of Flight Features



The correlation heatmap illustrates strong positive relationships among time-based variables, such as scheduled elapsed time, actual elapsed time, air time, and taxi durations. These dependencies reflect the structure of flight operations and help explain why these features become highly influential in the model.

Methods

This project uses three supervised learning approaches to classify flights as delayed or on time: Bagging, XGBoost, and a tuned XGBoost model. All three methods are tree-based, which makes them appropriate for structured aviation data containing nonlinear relationships, mixed variable types, and interactions between operational features. Bagging works by training many decision trees on different bootstrap samples and averaging their outputs. This reduces variance and produces stable predictions, which is useful when dealing with noisy real-world variables such as taxi times or air traffic delays. However, because Bagging treats each tree independently and does not focus specifically on correcting earlier mistakes, it tends to favour the majority class, which is a limitation in datasets where delayed flights are less frequent than on-time flights.

XGBoost builds on these ideas by training decision trees sequentially, where each tree attempts to correct the errors of the previous ones. This allows the model to learn more complex patterns that Bagging may overlook. A tuned version of XGBoost was also implemented by adjusting the learning rate and number of boosting rounds through cross-validation, which are discussed further in the report. The tuned configuration uses a smaller learning rate and more boosting iterations, allowing the model to generalise better and give better results.

Although boosting methods can achieve higher accuracy and better capture subtle delay patterns, they require careful parameter tuning and may still be limited by the information available in the dataset. Many causes of delays, such as weather disruptions or air traffic control interventions, are not fully represented in the features, which constrains the ability of any model to perfectly classify delayed flights.

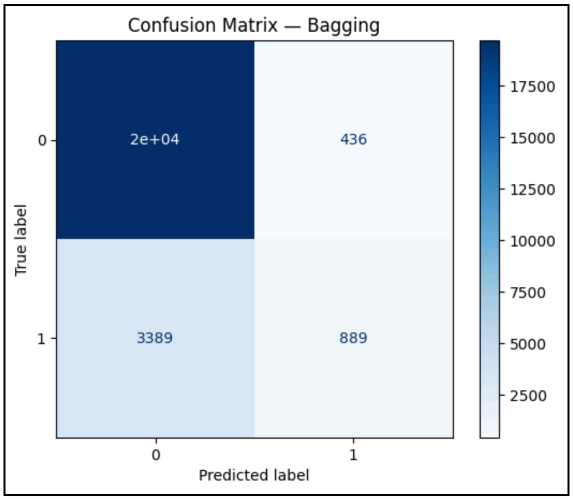
Results

Model performance was assessed using accuracy, test error, and AUC. Confusion matrices were additionally used to determine how well each model handled the imbalance between on-time and delayed flights.

Bagging Classifier (Baseline)

The Bagging model served as the baseline. It was fit using 100 decision-tree estimators with bootstrap sampling. Since Bagging reduces variance by averaging many trees trained on random subsets of the data, it provides stable predictions even without heavy tuning.

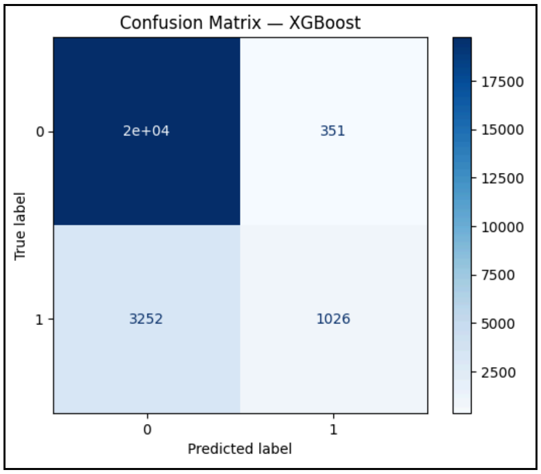
The model achieved an accuracy of 84.31%. The confusion matrix shows that Bagging correctly identified most on-time flights, which dominate the dataset. It predicted approximately 20,000 on-time flights correctly. However, it struggled with predicting the delayed flights, correctly identifying only 889 delayed flights while misclassifying 3,389 as on time. This pattern reflects the class imbalance and Bagging's tendency to favor the majority class. Although the accuracy is strong, the confusion matrix indicates that a large portion of delays remains undetected.



XGBoost (Default Parameters)

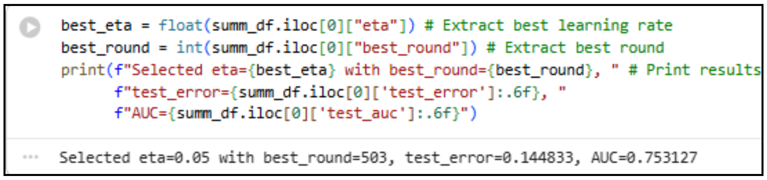
The XGBoost model was trained using its default configuration, aside from specifying a binary logistic objective and enabling AUC evaluation during training. Unlike Bagging, XGBoost improves performance by sequentially adding trees that correct the mistakes of previous iterations.

With this setup, XGBoost improved accuracy to 85.22%. Its confusion matrix shows that it continues to perform very well on on-time flights, again identifying roughly 20,000 correctly. It also captured more delayed flights than the Bagging model, predicting 1,026 correctly. The number of missed delays decreased to 3,252, demonstrating XGBoost's ability to extract more complex patterns from the data. This improvement reflects the advantage of boosting, which can learn finer distinctions between delayed and non-delayed flights.



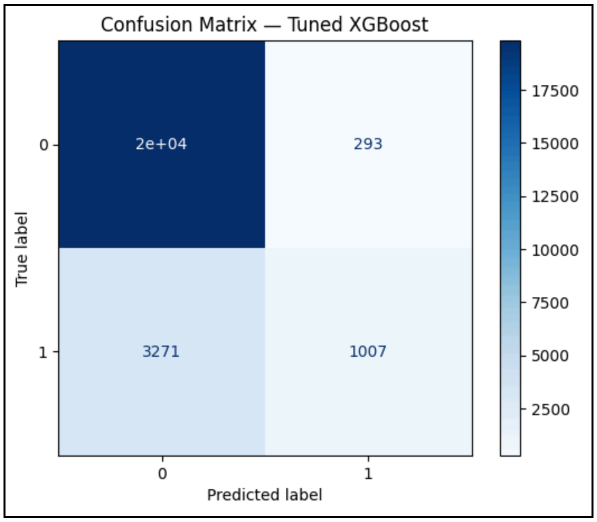
Tuned XGBoost

To further increase performance, XGBoost was tuned using five-fold cross-validation. Several learning rates and numbers of boosting rounds were tested. The tuning process identified a learning rate of 0.05 and 503 boosting rounds as the best-performing combination. A smaller learning rate slows the model's updates and allows it to learn more gradually, while the larger number of rounds gives the model enough depth to refine its decision boundaries.



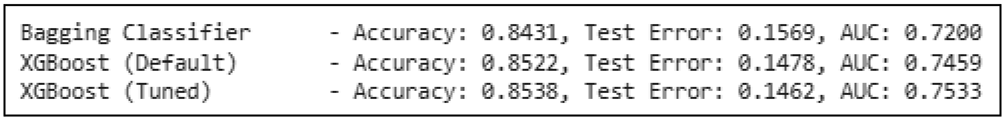
```
best_eta = float(summ_df.iloc[0]["eta"]) # Extract best learning rate
best_round = int(summ_df.iloc[0]["best_round"]) # Extract best round
print(f"Selected eta={best_eta} with best_round={best_round}, " # Print results
      f"test_error={summ_df.iloc[0]['test_error']:.6f}, "
      f"AUC={summ_df.iloc[0]['test_auc']:.6f}")

Selected eta=0.05 with best_round=503, test_error=0.144833, AUC=0.753127
```

The tuned model reached the highest accuracy at 85.38%. The confusion matrix shows that it correctly identifies more than 20,000 on-time flights, similar to the previous models, and correctly predicts 1,007 delayed flights. Misclassified delays remained challenging, with 3,271 incorrectly labeled as on time. The tuned model performs slightly better than the untuned version, but the improvement is modest, showing that delay prediction is limited by data variability and class imbalance.



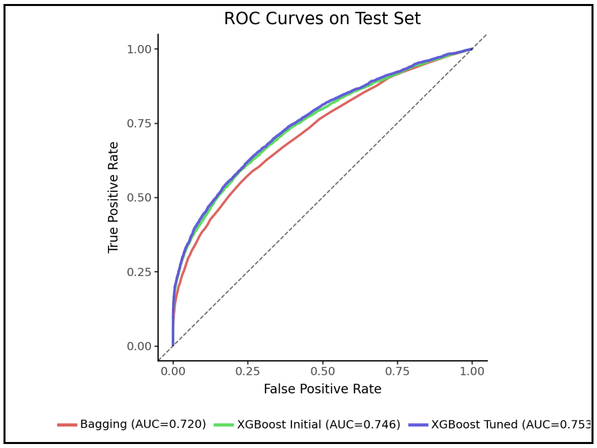
Comparison Across Models

The ROC curves provide a clearer picture of how each model performs across different decision thresholds. All three models follow a similar upward-curving shape, which means they are all meaningfully better than random guessing, but the separation between them shows small performance differences. Bagging has the lowest AUC at around 0.72, which matches what we saw in its higher test error. The default XGBoost model improves on this with an AUC of roughly 0.746, showing that it captures more of the signal in the data and maintains better ranking ability between delayed and on-time flights.



```
Bagging Classifier      - Accuracy: 0.8431, Test Error: 0.1569, AUC: 0.7200
XGBoost (Default)       - Accuracy: 0.8522, Test Error: 0.1478, AUC: 0.7459
XGBoost (Tuned)         - Accuracy: 0.8538, Test Error: 0.1462, AUC: 0.7533
```

The tuned XGBoost model performs the best overall, reaching an AUC of about 0.753. This improvement is modest but consistent with the lower test error we observed. The tuning process essentially helped the model learn the delay patterns more smoothly, and the ROC curve reflects that by sitting slightly above the other two models across most threshold values. Overall, the comparison shows a steady gain from Bagging to XGBoost, with tuning providing a final incremental boost in discriminative power.



Challenges Encountered

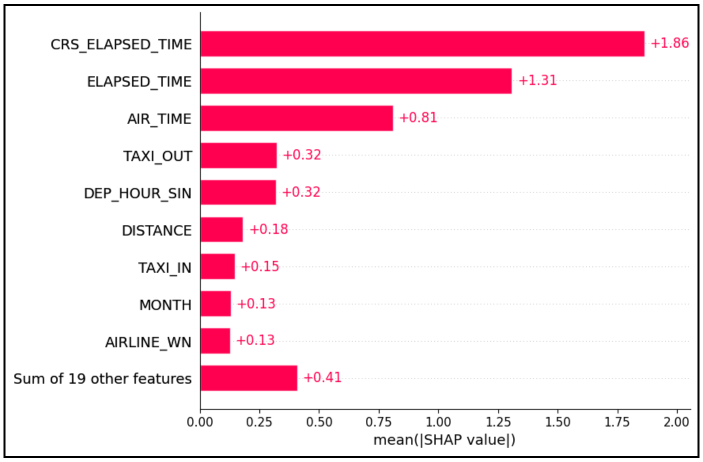
Two key challenges emerged during the modeling process. First, the natural class imbalance in the dataset made it difficult for the models to correctly identify delayed flights. Although accuracy remained high, confusion matrices revealed the limitations of using accuracy alone on imbalanced data. Second, several operational delay causes (weather, security, and late aircraft) contained missing values, necessitating careful preprocessing. These missing values were treated as zeros, which is appropriate for this dataset but may reduce precision in capturing the true source of some delays.

Discussions:

The SHAP value analysis provides a clear view of which features the tuned XGBoost model relied on most when predicting flight delays. Departure hour emerged as one of the strongest predictors, which aligns with common operational patterns in aviation. Delays tend to accumulate throughout the day, and evening flights often inherit disruptions from earlier in the schedule. The model captures this trend, assigning higher predicted delay probabilities to flights departing later in the day.

Airline indicators also show a strong influence. Some carriers have more consistent on-time performance, while others experience higher variability in operations. The model reflects these differences by adjusting predicted delay probabilities based on the airline operating the flight. This suggests that carrier-level operational efficiency and resource management play a meaningful role in delay outcomes.

The SHAP analysis also highlights the importance of time-related operational features such as *Taxi-Out* and *Taxi-In* durations. Taxi-out time, in particular, is closely connected to airport congestion and ground traffic conditions. Longer-than-normal Taxi-Out times often signal bottlenecks at the departure airport, which can delay takeoff and increase the likelihood of arriving late. Taxi-in time, while less predictive, still contributes information about arrival airport activity and runway availability. These findings point to potential operational improvements: consistently monitoring unusually long taxi times could help airlines and airports anticipate downstream delays and adjust schedules or resources proactively.



Although the tuned model performs the best among the three tested, its confusion matrix and overall accuracy show that predicting flight delays is still a difficult task. Many delayed flights share similar characteristics with on-time flights, and most delays arise from factors such as weather conditions, air-traffic control decisions, or unexpected mechanical issues that are not captured in the dataset. As a result, the model struggles to clearly separate delayed flights from non-delayed ones.

Overall, the analysis demonstrates that machine learning can identify meaningful patterns, including time-of-day effects, scheduling patterns, and differences across airlines. However, the complexity and unpredictability of real aviation operations still limit perfect classification. These findings are useful for highlighting where operational improvements may reduce delays and where collecting additional, more detailed data could enhance future predictive performance.

Conclusion

This project evaluated Bagging, XGBoost, and a tuned XGBoost model for predicting whether a U.S. domestic flight would arrive more than fifteen minutes late. The tuned XGBoost model produced the strongest performance, showing the highest accuracy and AUC and offering the clearest separation between delayed and on-time flights. The SHAP analysis showed that the model learned meaningful operational patterns. Departure hour strongly influenced delay predictions, and airline identity, taxi-out time, and taxi-in time also played important roles. These results demonstrate that machine learning can capture several underlying factors that contribute to delays in real aviation settings.

Although the models performed reasonably well, they still struggled to classify many delayed flights correctly. This is largely due to missing real-world variables that strongly affect delay outcomes. The dataset did not include weather conditions, visibility, storm activity, runway closures, airport congestion levels, or aircraft rotation history. These external factors often trigger delays, but were not available for modelling. Including such information would likely improve the model’s sensitivity and overall predictive performance.

Contribution:

This is a solo project. All aspects of data exploration, modeling, reporting, and visualization will be completed by me.

Bibliography:

1. U.S. Passenger Carrier Delay Costs

   - Airlines for America. (2024). *U.S. Passenger Carrier Delay Costs*. Retrieved from https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs/

2. Hybrid Machine Learning-Based Model for Predicting Flight Delay

   - Guo, S., Zhao, H., Wang, Q., Yu, X., Guo, L., & Ren, T. (2024). A hybrid machine learning-based model for predicting flight delay through aviation big data. *Frontiers in Neurorobotics*, 18, Article 10897135. https://pmc.ncbi.nlm.nih.gov/articles/PMC10897135/

3. Generation and Prediction of Flight Delays in Air Transport

   - Li, X., & Jing, Z. (2021). Generation and prediction of flight delays in air transport. *The Journal of Engineering*, Article 531e7a3585733a374947b0560113d064bc900790. https://www.semanticscholar.org/paper/Generation-and-prediction-of-flight-delays-in-air-Li-Jing/531e7a3585733a374947b0560113d064bc900790

4. Kaggle Flight Delay Prediction Projects:

   - Kaggle. (n.d.). *Flight delay prediction*. Retrieved from https://www.kaggle.com/search?q=flight+delay+prediction