Real-Time Flight Delay Forecasting Using Apache Spark Streaming

Nolan Pettit
Elizabethtown College
pettitn@etown.edu

Matthew Smith
Elizabethtown College
smithm6@etown.edu

ABSTRACT

Flight delays cost an estimated $33 billion annually [1], making flight delay forecasting an important challenge in the aviation industry. Accurate prediction of flight delays can benefit airline companies and their customers by helping them be more informed about potential risks. This project aims to develop a real-time flight delay prediction model using Apache Spark Streaming and Kafka. This research would provide innovations to the fields of data science and aviation by incorporating weather and flight data. By utilizing machine learning models such as Logistic Regression, Random Forest, and Gradient-Boosted Trees, it seeks to accurately predict flight delays in real-time. The expected result is a robust system that will accurately forecast flight delays in real-time. The results of the machine learning models will be measured using accuracy, precision, and F1-score, and they will be sent to a database where a dashboard can display the predictions in a well-formatted view. The machine learning models will be evaluated and compared, and the best model will be chosen for the streaming pipeline. The latency will also be measured to determine the speed of the real-time predictions.

1 INTRODUCTION

This project will provide a real-time flight delay forecasting model by implementing Apache Spark and Kafka. It will use machine learning models like Random Forest and Gradient-Boosted Trees to provide accurate insights for flight delays. The results will be sent to a dashboard to be easily accessible for customers and airline companies. However, despite advances in this field, several challenges remain. Firstly, model accuracy sometimes fails to incorporate the full story for flight delays. Many current models fail to include weather data in their models which may lead to inconsistent results in the accuracy of these models. Another problem is that due to the high volume of data, current models tend to have high latency in their predictions. These models also have high computation costs, making them difficult to implement effectively.

An existing solution to this problem involves using support vector regression that fine-tunes the data to provide prediction analysis for flight delays [2]. This study strictly uses flight data from Beijing International Airport, so it fails to capture the big picture of flight delays. It also does not include weather data, so the accuracy metrics may not be entirely true. Another solution involves using Mixture Density Networks and Random Forest to provide a robust solution to this problem. While it achieved high accuracy values, this research does not provide a real-time system for predicting flights that would be useful for customers.

Part of our solution includes determining feature importance between the flight data and weather patterns to better pinpoint the causes of flight delays. This will help prioritize and understand key components of delays to provide better accuracy in the machine learning models. Additionally, we will utilize the resources from Apache Spark and Kafka to provide a real-time solution to flight delay analysis. This will create a unified system that incorporates flight patterns and weather data to give a more well-rounded approach to this issue. We also aim to implement a real-time dashboard that can provide detailed information regarding flight status in a well-formatted manner. The use of Random Forest, Gradient-Boosting Trees, and Logistic Regression models in the Spark pipeline will allow for scalability and low-latency results.

The rest of this proposal includes a background section where relevant research will be reviewed and analyzed. The design section will outline our approach to this big data issue, highlighting the usage of Apache Spark and Kafka. The experiments section will demonstrate the use of our Logistic Regression, Random Forest, and Gradient-Boosting models to provide accurate insights on flight delays. Finally, timeline and contributions sections will ensure the organized progression and documentation of our research.

Reference	Advantages	Disadvantages
Yu et al. (2019) Beijing SVR Model	- Handles large-scale air travel data - Identifies key delay factors - Improves model accuracy with Support Vector Regression	- Poor performance in real-time - Limited by latency handling
Lambelho et al. (2020) Heathrow Scheduling Model	- Optimizes flight schedules - Reduces delays strategically - Machine learning–driven scheduling	- Scalability challenges - Airport-specific applicability
Zoutendijk & Mitici (2021) European MDN & RF Model	- Achieves MAE < 15 mins - Reduces gate conflicts by 74% - Probabilistic modeling improves robustness	- Focused only on European airports - Needs testing on broader datasets

Figure 1: Background

2 BACKGROUND

Researchers from Beijing, China developed a model for flight delay prediction using Support Vector Regression [2]. They gathered data from flights leaving Beijing International Airport to find patterns in flight delays. This research advances this field of study by managing the large amount of air travel data. These researchers were also able to determine some of the key factors that influence delays, allowing them to generate a better overall model. Ultimately this research faces limitations in handling latency issues as it does not perform effectively in real-time.

Another team developed a flight delay model using Mixture Density Networks and Random Forest [3]. They gathered data from European airports to determine flight delays. They were able to make substantial progress in this field by achieving a Mean Absolute Error of less than 15 minutes. These researchers were also able to create a flight-to-gate model that managed to reduce aircraft conflicts by about 74%. This research can be expanded to areas such

as North America to include a wider variety of data which can hopefully improve the results even further.

A third group took a different approach to this issue by using machine learning to determine strategic flight schedules [4]. The goal of this would be to reduce delays by scheduling flights in a more organized way. They managed to create an optimized schedule that reduced flight delays at the London Heathrow Airport. This research can be improved by looking at a wider scope of airports. However, it is limited by challenges related to scalability.

3 DESIGN

This research will include three machine learning algorithms with Logistic Regression, Random Forest, and Gradient-Boosting Trees. Additionally, we will implement these models under the Apache Spark Streaming pipeline and use Kafka for simulation analysis. To start, we will implement data ingestion with Kafka that gathers real-time flight data, weather data, and airline traffic. Before moving into the Spark Streaming and Kafka stages of this research, the data must be processed to allow for consistency and accuracy in the results. The first model that we will test will be a Logistic Regression model, which we expect to perform at a base-line level due to its simplicity and implementation for initial comparisons. The second model that we will test will be a Random Forest model which we believe will perform well due to its capabilities in handling non-linear and complex data. Factors such as weather and airline schedules will likely be non-linear, allowing for Random Forest to perform well. We will also implement a Gradient-Boosting Tree model due to its sequential nature and its ability to handle interacting features. The flowcharts for the real time flight prediction and offline flight models are shown in Figures 2 and 3.

3.1 Data Preprocessing

The preprocessing stage started with loading the vast amount of data into a Spark session. Columns that were duplicates of others—such as Destination and Destination City, where Destination City contained all the information from Destination—were removed. The null values for arrival delays, departure delays, origin cities, destination cities, and distance were removed because these are our predictive columns. Having null or synthetic values in these rows would negatively influence the performance of the machine learning models. Next, the null values in the columns that indicated the cause of delay were replaced with zero. It is a fair assumption that if these rows are null, then the delay was not caused by that reason.

The flight date column was also converted to a standard YYYY-MM-DD format. The numerical columns were scaled using a MinMaxScaler to normalize the data. There was also a weight of 0.2 added to the majority class of severe delays since there was a severe imbalance in the delays. The flight distance was also filtered to not include anything above three-thousand since that would skew the data. The logistic distribution of arrival delays can be seen in Figure 4. Figure 5 is a bar chart that shows the representation of delays by destination airport. Lastly, a graph shows the average delay by airline in Figure 6.

3.2 Feature Engineering

The first step in the feature engineering stage was to create a column labeled "Severe Delays" that indicate when a delay is longer than 45 minutes. Other variables were extracted from the arrival time and the departure time to allow more analysis on the hour of the day, the day of the week, and the month of the year to name a few. The distance per minute of the flight was also calculated to determine how efficient each flight was from its comparison of distance traveled to elapsed time of the trip.

Real-time data streaming with Kafka will be achievable with the combination of flight schedules and a weather data API. Spark Streaming will be implemented for data cleansing, feature extraction, and data transformation to provide an accurate real-time prediction model. The trained machine learning model will be applied to Kafka to generate a real-time flight delay simulation. An interactive dashboard will be developed to display the results using interactive graphs and tables. The models will be evaluated using mean absolute error and they will be analyzed on feature importance. We plan on utilizing Apache Parquet to store the data in a centralized location.

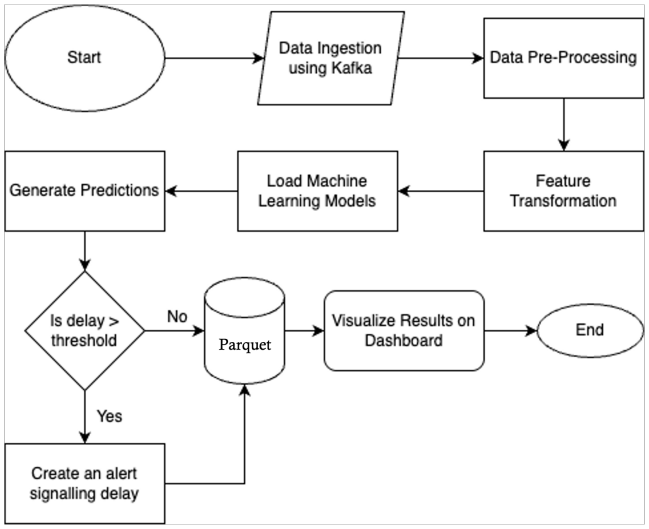

Figure 2: Real Time Flight Flowchart

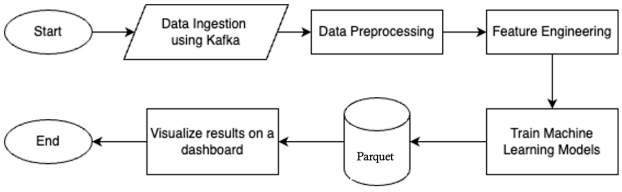

Figure 3: Offline Flight Flowchart

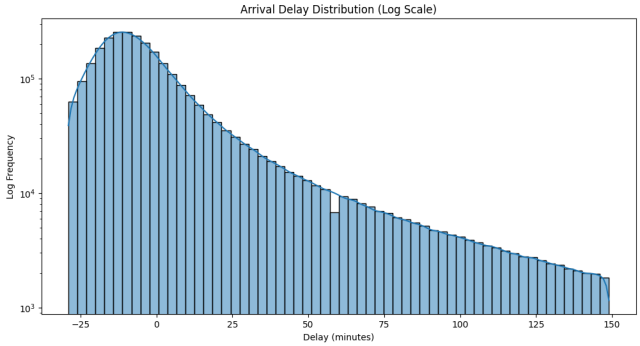

Figure 4: Arrival Delay Distribution

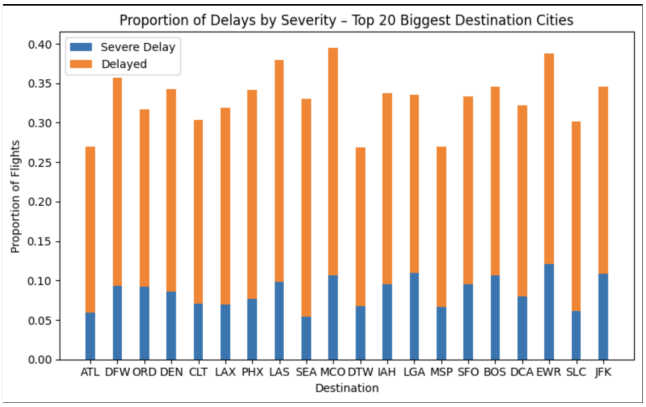

Figure 5: Delay by Destination City

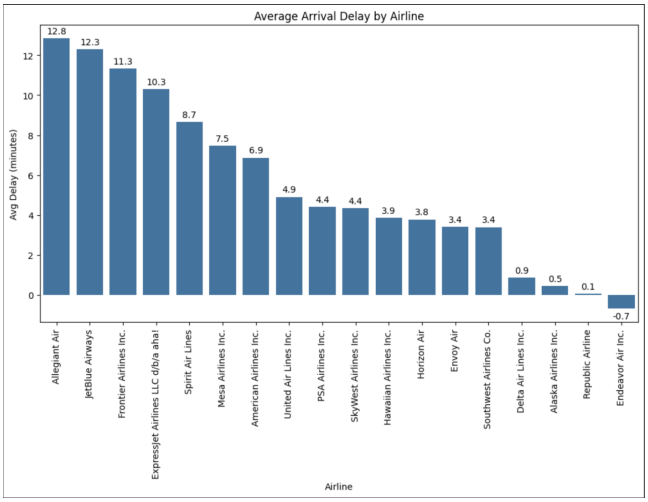

Figure 6: Delay by Airline

4 EXPERIMENTS

In this section, we discuss the experiments conducted to evaluate the performance of various machine learning models in flight delay prediction. The tested models are Logistic Regression, Random Forest, and Gradient-Boosting Trees. Each model is designed to solve the complexity of the flight data, with unique strengths and weaknesses in different aspects of prediction. We also explain how we can integrate these models into a real-time system in practice with Apache Kafka and Spark Streaming to provide real-time updates. From our experiments, we aim to measure the accuracy, scalability, and the real-time suitability of each model when applied to predicting flight delays.

4.1 Logistic Regression

We selected Logistic Regression as our baseline model due to its simplicity and effectiveness in binary classification problems. It was used to predict flight delay probabilities from variables such as airport congestion, weather, and historical delay data. While a useful baseline model, Logistic Regression is poor at capturing complex, non-linear variable relationships due to its linear nature. it was expected for its performance to be consistent but inferior to sophisticated models like Random Forest and Gradient-Boosting Trees, but it had the exact same results as Random Forest surprisingly.

4.2 Random Forest

Random Forest utilizes multiple decision trees to improve prediction accuracy and prevent overfitting. The model was chosen because it can handle large datasets in addition to resisting noise. It should show significant improvement over Logistic Regression since it can handle non-linear relationships as well as interactions between features. Its increased accuracy and adaptability made it a potential candidate for predicting flight delays, but it still faces challenges due to its computational intensity. Unfortunately, it did not live up to its expectations as it had the exact same results as the Logistic Regression: 0.9388 accuracy, 0.8813 precision, 0.9388 recall, and 0.9091 F1 score.

4.3 Gradient-Boosting Tree

Gradient-Boosting Trees (GBT) is a powerful ensemble method that develops trees sequentially to improve the errors of previous models, and is therefore highly applicable to complex prediction tasks. GBT in this research worked better in predicting flight delays with a high capacity to model complex interactions in the data. By incrementally optimizing the model, GBT achieved more accuracy and precision compared to Random Forest. However, just like Random Forest, even GBT had scalability issues, and they proved to be the main contributing factor in implementing the model in a real-time system. GBT's results were: 0.9369 accuracy, 0.8996 precision, 0.9369 recall, and 0.9104 F1 score. GBT was the best model for our uses as it only had a small tick of .0018 below random forest's accuracy and recall but it had a more significant jump of more than one-percent in both precision and F1 score.

4.4 Integrating into Real-Time

To integrate these models into an actual real-time system, we utilized Apache Kafka to stream data and Apache Spark Streaming to run the real-time flight data through processing. Kafka supplied the ever-going ingestion of the flight and weather data, while Spark Streaming provided us with the infrastructure that would allow us

to apply trained models to incoming data as it streamed. This setup allowed us to create real-time predictions of flight delays and output the result to a dashboard for immediate access by airline companies and travelers. Low-latency predictions and real-time processing of large-scale data, however, were huge challenges, especially with more computationally intensive models like Gradient-Boosting Trees.

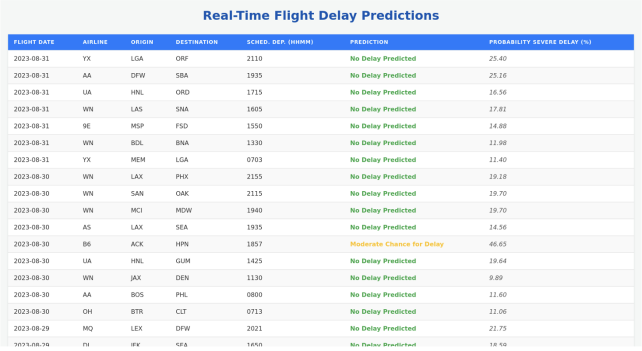

Figure 7: Live Time Application

Figure 8: Visualization Application

5 RESULTS

We decided to implement a Gradient Boosting Decision Tree model to determine flight delays in real-time. Due to the size of the dataset and the capabilities of our GCP, we sampled only half of the dataset when running the model. 1,166,305 samples were used to train the model. The data was also split using an 80/20 division with a seed of 101 so we could replicate the results. The Gradient Boosting Decision Tree model was trained to have 50 trees with a max depth of 5. Due to the imbalance in the dataset, we used the weighted metrics to determine the overall performance of our model.

The Gradient Boosting model performed with an accuracy of 93.69%, a weighted precision of 89.96%, a weighted recall of 93.69%, and an F1 score of 91.04%. Additionally, Kafka was used to allow us to stream in data to predict in real-time. This performed well, creating batches that are 16 KB in size. These batches allow the model to simulate data being streamed in a real-world scenario. Feature Importance analysis was also done on the GBT model seen in Figure 9 and determined that some variables had very high importance like destination with almost 25%, airlines with 17.83%, and origin airports with 16.66%. This shows that location-based factors and airline choice play a significant role in predicting delays in the GBT model which logically makes perfect sense.

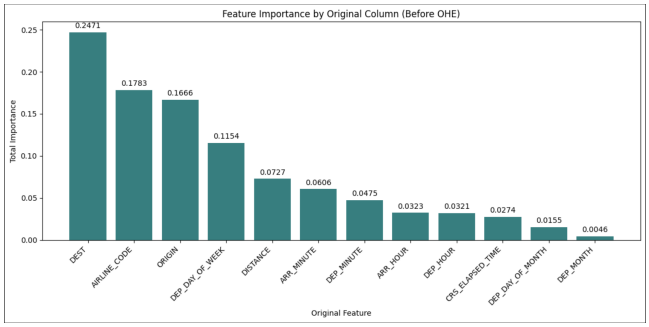

Figure 9: Feature Importances of GBT Model

6 TIMELINE

We will hold weekly meetings to discuss the progress of our parts of this project. Additionally, we aim to have baseline models for Logistic Regression, Random Forest, and Gradient-Boosting Tree to be completed by the first week of March. Following their completion, we will integrate them into Spark Streaming and Kafka to create the real-time model by the end of March. In the later stages of March and early April, we expect to be nearly complete with the dashboard using either PowerBI, Tableau. We ended up choosing the combination of Flask, Spark Streaming, and Kafka to create our dashboard seen in Figure 7. We also created a separate dataset application that showed visualizations of our dataset and also the results of different models and different methods to try to balance the data seen in Figure 8.

7 CONTRIBUTIONS

The project was divided fairly between the two of us. Nolan was responsible for training the majority of our three models, while Matt was in charge of making these models run with low latency using Spark Streaming and Kafka. Together, we developed a solution to integrate all of this research into a dashboard that provides clear insights on flight delays. Nolan contributed primarily to the data preprocessing and visualizations that include the distributions of the data. Matt played a large part in working through feature engineering and additional visualizations that describe in-depth details of our data.

REFERENCES

[1] Airlines for America. (n.d.). U.S. passenger carrier delay costs. Retrieved February 27, 2025, from https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs/

[2] Yu, B., Guo, Z., Asian, S., Wang, H., & Chen, G. (2019). Flight delay prediction for commercial air transport: A deep learning approach. Transportation Research. Part E, Logistics and Transportation Review, 125, 203–221. https://doi.org/10.1016/j.tre.2019.03.013

[3] Zoutendijk, M., & Mitici, M. (2021). Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem. Aerospace, 8(6), 152-. https://doi.org/10.3390/aerospace8060152

[4] Lambelho, M., Mitici, M., Pickup, S., & Marsden, A. (2020). Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions. Journal of Air Transport Management, 82, 101737-. https://doi.org/10.1016/j.jairtraman.2019.101737