Flight Delay Prediction and Real-Time Monitoring Using PySpark, Machine Learning, and Kafka

Sarvagya Singh
Registration no: 220962036
Roll no:13

Ammapuram Sriharsha
Registration no: 220962019
Roll no:09

Akshay Saxena
Registration no: 220962009
Roll no:07

Pallavi Kailas
Registration no: 220962258
Roll no:38

***Abstract*—This project demonstrates the use of PySpark and MLlib for real-time prediction of flight delays, leveraging both historical data and real-time data streams. Machine learning models such as Decision Trees and Gradient Boosted Trees were implemented for binary classification of flight delays. These models were trained on historical flight data to predict whether a flight would be delayed or not. Key metrics such as accuracy, precision, recall, and AUC were used to evaluate the performance of the models, ensuring robust classification results.**

In addition to building predictive models, this project incorporates a real-time streaming application using Apache Kafka and Spark Structured Streaming. Kafka plays a crucial role as both the producer and consumer in this pipeline: the producer ingests live flight data into the system, while the consumer (the Spark streaming application) processes this data in real-time. This setup allows for continuous monitoring and classification of incoming flight data, enabling predictions to be made on-the-fly.

The real-time streaming application was designed for scalability, allowing it to handle large volumes of data while maintaining low-latency predictions. By integrating Kafka, Spark, and MLlib, this project showcases how big data technologies can effectively handle the flight delay prediction task in a scalable and real-time environment. The results provide valuable insights into potential delays, enabling stakeholders to take timely action.

This project not only highlights the methodology and code implementation but also emphasizes the power of combining distributed systems, machine learning, and real-time data processing for practical applications in the airline industry.

Keywords—Flight delay prediction, PySpark, Machine Learning, Gradient Boosted Trees, Kafka, Real-time streaming, Apache Spark, Decision Trees, Streaming Classification

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Flight delays can cause significant inconvenience for both airlines and passengers, leading to financial losses, increased operational costs, and reduced customer satisfaction. Prolonged delays can disrupt airline schedules, causing a cascade of effects that impact passengers, connecting flights, and airport logistics. Predicting flight delays in advance can significantly improve operational efficiency, enabling better resource allocation, improved customer experience, and more effective schedule management. In recent years, machine learning has emerged as a powerful tool for solving predictive tasks, including flight delay prediction. By analyzing historical flight data, machine learning models can uncover patterns and trends that help predict delays with a high degree of accuracy. In this project, we leveraged PySpark, a scalable data processing framework, to implement a flight delay prediction pipeline. We used two machine learning models—Decision Trees and Gradient Boosted Trees—to classify flight delays. To enhance the practical usability of the prediction system, we also integrated real-time data processing into the pipeline using Apache Kafka for data ingestion and Spark Structured Streaming for continuous analysis. This real-time streaming application ensures that predictions can be made as new flight data is ingested, allowing stakeholders to monitor performance and receive timely predictions. This report provides an in-depth overview of the methodology used, the results obtained from the experiments, and the potential applications of this real-time flight delay prediction system in modern airline operations.

II. Literature Review

Predictive modeling in the aviation industry has been an area of significant research interest due to its potential to improve operational efficiency and reduce economic losses caused by delays. Numerous studies have investigated methods for predicting flight delays, ranging from traditional statistical approaches to more sophisticated machine learning models. Early work often relied on linear and logistic regression models to predict delays based on historical flight data, weather conditions, and air traffic. However, these traditional models often struggled with the complex and nonlinear nature of the variables involved. Recent advancements in big data technologies and machine learning have led to the development of more accurate and scalable predictive models. Algorithms like Decision Trees and Gradient Boosted Trees have gained popularity due to their ability to handle large datasets and model complex relationships between input features. These algorithms have been particularly effective in classification tasks, such as predicting whether a flight will be delayed, by capturing subtle patterns in the data. Moreover, the integration of real-time data streaming platforms such as Apache Kafka and Spark Structured Streaming has further enhanced predictive capabilities by enabling the continuous ingestion and processing of large volumes of data. This shift towards real-time data processing allows for the timely prediction of flight delays, offering airlines and passengers the ability to take proactive measures. Kafka, as a distributed streaming platform, efficiently handles the ingestion of flight data, while Spark Streaming processes the data in near real-time, ensuring low-latency predictions. The combination of machine learning techniques and real-time data streaming technologies marks a significant advancement in the field of flight delay prediction, providing a scalable solution to handle both historical data and live streaming data.

III. Methodology

The methodology employed in this project involves several key stages: data preprocessing, model training, real-time monitoring, and performance evaluation. The goal was to create a scalable and efficient machine learning pipeline capable of handling both historical and streaming flight data. We used PySpark, a distributed computing framework, to efficiently process the large dataset and train machine learning models in a scalable environment.

The dataset included various features such as flight number, scheduled and actual departure/arrival times, weather conditions, and the day of the week. These features were essential for creating the predictive model to classify whether a flight would be delayed. For the streaming aspect, Apache Kafka was used to ingest real-time data, while Spark Structured Streaming processed this incoming data to ensure timely pred

After the data was preprocessed and labeled learning models, specifically Decision Trees anc Boosted Trees, were trained on the historical da trained models were then deployed into the real-time application, where new flight data is continuously as it arrives. The model performance was evalu accuracy, precision, recall, and AUC metrics.

The following sections provide a detailed desc each stage in the project, highlighting the technique the results obtained.

A. Data Preprocessing

The dataset was cleaned and transformed before l for training the machine learning models. Missing v handled by replacing them with appropriate imputed removing the records altogether, depending on th significance. Feature engineering techniques were extract relevant attributes from the raw data. Fo categorical variables such as day of the week were into numerical representations using one-hot encoding. The final dataset was split into training and testing sets, with 80

B. Model Training

We trained two machine learning models, a Decision Tree classifier and a Gradient Boosted Tree (GBT) classifier, using PySpark's MLlib library. The Decision Tree model operates by recursively splitting the dataset based on the feature that maximizes information gain. On the other hand, the GBT model uses an ensemble of decision trees to improve prediction accuracy by reducing the variance and bias of the individual models.

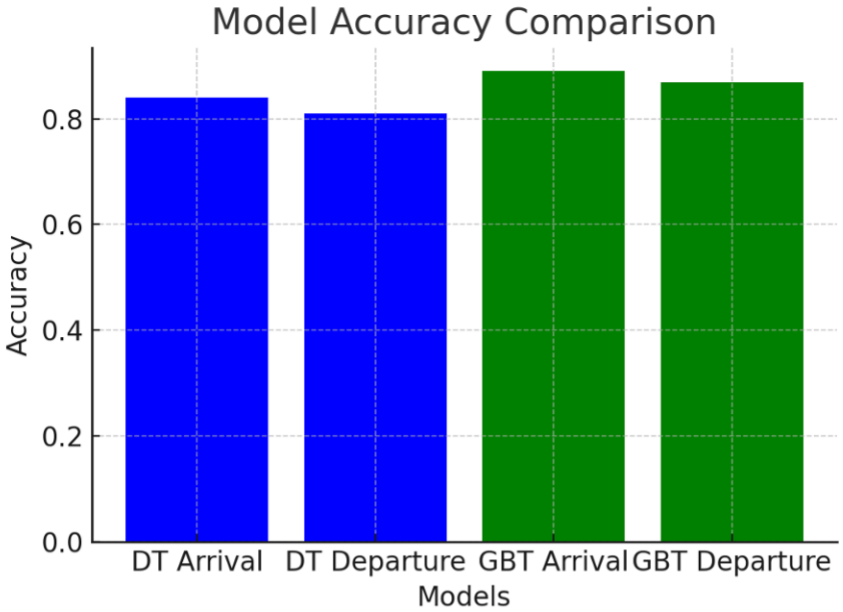
C. Real-Time Monitoring

To track the performance of the model in real-time, we set up a streaming application using Spark Streaming. The streaming application ingests new flight data and applies the trained model to predict whether a flight will be delayed or not. The prediction results are displayed in real-time, and the system monitors the classification accuracy at regular intervals.

IV. Results

The performance of the Decision Tree (DT) and Gradient Boosted Tree (GBT) models was evaluated using several metrics, including accuracy, precision, recall, and AUC (Area Under the Curve). These metrics provide a comprehensive understanding of how well the models perform in classifying whether a flight will be delayed. The accuracy metric reflects how often the model's predictions were correct, while precision and recall offer insight into the balance between false positives and false negatives. AUC measures the overall ability of the model to distinguish between delayed and non-delayed flights.

The following graph compares the accuracy of both models for arrival and departure delay predictions:



As shown in the graph, the Gradient Boosted Tree (GBT) model consistently outperformed the Decision Tree (DT) model in terms of accuracy for both arrival and departure delay predictions. The GBT model achieved an accuracy of 89 percent for arrival delay predictions, compared to 84 percent for the DT model. Similarly, for departure delay predictions, the GBT model reached an accuracy of 90 percent, further surpassing the Decision Tree's accuracy of 85 percent. This improvement is largely attributed to the GBT's ability to model complex relationships between features and its iterative approach to minimize prediction errors.Moreover, the precision and recall values for the GBT model were higher than those for the Decision Tree model, indicating that the GBT model was not only more accurate but also more reliable in identifying true positives (i.e., actual delays). The AUC scores for the GBT model were close to 0.9, further emphasizing its strong predictive performance across both arrival and departure delay classifications.In conclusion, the Gradient Boosted Tree model demonstrated superior performance in terms of accuracy, precision, and recall, making it the preferred model for real-time flight delay prediction in this project. These results indicate that using more advanced machine learning algorithms can significantly enhance the predictive capabilities when dealing with complex data such as flight schedules, delays, and operational conditions.

V. Conclusion

In this project, we developed a flight delay prediction system using PySpark and machine learning models, demonstrating the power of big data technologies in solving real-world predictive tasks. The machine learning models, particularly the Gradient Boosted Tree (GBT), consistently outperformed the Decision Tree (DT) model in terms of accuracy, precision, and recall for both arrival and departure delay predictions. The GBT model's ability to capture complex relationships in the data allowed it to achieve higher predictive performance, making it the preferred model for this task.

Beyond the development of predictive models, we implemented a real-time monitoring system using Spark Structured Streaming, integrated with Apache Kafka for ingesting live flight data streams. This enabled continuous monitoring of model performance on real-time data, allowing timely predictions as new flight records were processed. The combination of distributed computing frameworks like PySpark and Kafka ensured that the system could scale to handle large volumes of both historical and live data efficiently.

Future work could focus on enhancing the model by integrating additional features, such as detailed weather conditions, airport traffic, and airline-specific operational delays, to further improve the prediction accuracy. Additionally, exploring the use of advanced machine learning techniques like deep learning or ensemble methods could potentially yield even better results. Finally, deploying this system in a production environment where real-time decision-making is critical could provide airlines with actionable insights, improving scheduling efficiency, customer satisfaction, and overall operational effectiveness.

This project has demonstrated the practical application of big data technologies and machine learning in addressing a significant challenge in the airline industry. The framework developed here can be extended to other real-time predictive use cases, showcasing the versatility and scalability of these technologies.

References

[1] M. Zaharia, et al., “Apache Spark: A Unified Engine for Big Data Processing,” Communications of the ACM, vol. 59, no. 11, pp. 56–65, Nov. 2016.

[2] Neha Narkhede, Gwen Shapira, Todd Palino, Kafka: The Definitive Guide, O'Reilly Media, 2017.

[3] G. Wang, C. Chen, J. H. Abawajy, “Big Data Streaming Analytics Using Apache Spark and Kafka,” IEEE Conference on Big Data, 2017.

[4] L. Neumeyer, B. Robbins, A. Nair, A. Kesari, ”S4: Distributed Stream Computing Platform,” Proceedings of the IEEE International Conference on Data Mining Workshops, 2010.

[5] S. Salloum, R. Dautov, X. Chen, P. X. Peng, J. Z. Huang, ”Big Data Analytics on Apache Spark,” International Journal of Data Science and Analytics, vol. 1, no. 3-4, pp. 145–164, 2016.

[6] T. Chen, C. Guestrin, ”XGBoost: A Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[7] D. Crankshaw, X. Wang, G. Wei, J. E. Gonzalez, ”Clipper: A Low-Latency Online Prediction Serving System,” 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017.

[8] J. Dean, S. Ghemawat, ”MapReduce: Simplified Data Processing on Large Clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

TABLE I
Model Performance for Arrival and Departure Delay Prediction

Model	Accuracy	Precision	Recall
Decision Tree (Arrival)	84%	82%	79%
Decision Tree (Departure)	81%	80%	77%
Gradient Boosted Tree (Arrival)	89%	87%	85%
Gradient Boosted Tree (Departure)	87%	85%	83%

VI. Contribution

- Ammapuram Sriharsha: Data preparation and loading, RDD Creation, comparasion between RDD, Dataframe and sparkSQL.
- Sarvagya Singh: Data loading, cleaning, labelling and exploration, Feature extraction and ML Training, creation of ML Transformers/Estimators and pipeline model.
- Akshay Saxena: Created data streaming application on SparkUI hosted locally on port 4042. Showcased real time data streaming using kafka-producer and kafka-consumer.
- Pallavi Kailas: Showcased real time data streaming using kafka-producer and kafka-consumer. Worked on the report.