IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Flight Delay Estimation And Reason Analysis Through A Python GUI Application

[1]Kanugulla Srisailam, [2]Dr.V.Veerabadhram,

[1]M.Tech 1st year 2 Sr. Assistant professor

[1]Department of Computer Science and Engineering, AI Branch

[1]CVR College of Engineering ,Hyderabad, India

Abstract: Accurately predicting flight delays is a key challenge in aviation analytics, with significant implications for airline operations and passenger satisfaction. This project presents a machine learning-based system for predicting flight delays using historical U.S. domestic flight data. The dataset is extensively preprocessed, including the imputation of missing values, encoding of categorical variables, and generation of time-based features such as scheduled departure hour and total delay duration. A binary classification approach is adopted, labeling flights as delayed when the arrival delay exceeds 15 minutes. Three models—Random Forest, Decision Tree, and Logistic Regression—are developed and evaluated. Both the Random Forest and Decision Tree classifiers achieved 100% accuracy on the test set, while Logistic Regression reached 98.4%, indicating the strong predictive capability of the selected features. However, such perfect accuracy suggests potential data leakage or overfitting, as the delay-related columns used in training directly influence the target variable. A user-friendly graphical interface, built with Tkinter, allows users to input flight number, tail number, origin, and destination to receive real-time delay predictions and reasons. Additionally, visual analytics generated using Seaborn highlight delay trends by departure hour and weekday. The system demonstrates how machine learning, combined with rich flight data, can support operational decisions in aviation, although care must be taken to ensure model generalization and reliability

***Index Terms* -** Flight delay prediction, machine learning, Random Forest, Decision Tree, Logistic Regression, data preprocessing, binary classification, aviation analytics, graphical user interface, Tkinter, overfitting, data leakage, visual analytics, Seaborne.

I. Introduction

The global airline industry serves billions of passengers annually, and even small inefficiencies in operations can lead to substantial disruptions, economic losses, and negative passenger experiences. One of the most pervasive issues affecting both airlines and travelers is flight delays, which not only waste valuable time and resources but also contribute to scheduling disruptions across the entire air traffic network [1]. According to studies, delays can trigger a cascading effect—where one late arrival leads to multiple subsequent delays—ultimately increasing airline operating costs, fuel consumption, and passenger dissatisfaction [2].

The causes of flight delays are multifaceted, ranging from weather conditions, air traffic congestion, and airline operations, to security issues and airport resource constraints [7]. In the United States and Europe, delay data is routinely categorized by reason (e.g., Carrier, Weather, NAS, Late Aircraft, and Security), with well-defined metrics reported by agencies such as theU.S. Bureau of Transportation Statistics (BTS) [5]. The availability of this structured data opens up opportunities to explore data-driven techniques for predicting and explaining delays more accurately.

Traditional delay estimation techniques—based on historical averages, linear regression, or statistical summaries—lack the capability to model the complex, nonlinear relationships between various flight, weather, and operational parameters [3][10]. As aviation systems grow more complex, there is a growing need for intelligent prediction models that can handle high- dimensional data and make real-time decisions. Recent research has turned to machine learning (ML) to address this gap, leveraging techniques such as classification trees,ensemble methods, and deep learning to build more accurate and scalable solutions [4][6].

This project explores the use of supervised machine learning algorithms to predict whether a flight will be delayed based on a set of operational, temporal, and route-based features. Specifically, this study implements Random Forest, Decision Tree, and Logistic Regression classifiers to train models on real-world flight data. These models are selected based on their proven effectiveness in previous transportation studies and their interpretability in real-time systems [8][9]. A critical component of this work is the addition of feature engineering, including the creation of a binary Delayed label based on arrival delay thresholds and a Reason column derived from the first non-zero delay type (e.g., WeatherDelay, NASDelay). This not only improves model performance but also provides insight into the cause of delay—an essential feature for airline decision-makers.

In addition to the modeling pipeline, a Tkinter-based graphical user interface (GUI) is integrated into the system to make the prediction tool interactive and user-friendly. The GUI allows users to enter flight-specific details such as flight number, origin, and destination, and receive immediate feedback about whether the flight is predicted to be delayed, along with the reason and estimated delay duration.

The primary objective of this project is not only to predict the likelihood of a delay but also to offer interpretability, usability, and a foundation for future extensions such as delay mitigation strategies and smart scheduling. By combining machine learning models with real- world flight data, this work aims to support data-informed decisions in modern aviation systems and contribute to reducing the economic and operational burdens associated with delays.

II. Related work

Over the past two decades, researchers have made significant efforts to analyze and predict flight delays using statistical and computational techniques. Traditional models often relied on historical averages and regression analysis, which, while effective in limited contexts, failed to capture the nonlinear relationships between different delay causes such as weather, air traffic congestion, and airline operations [1][2]. More recent advancements have focused on machine learning techniques to model these complex dependencies more accurately.

Hansen and Zhang [1] explored the operational consequences of airport surface congestion and emphasized the critical role of departure delays in cascading disruptions across the network. Cook [2] provided one of the most cited references on the economic impact of delay, categorizing delays into key components such as weather, security, NAS (National Airspace System), and carrier-induced delays.

Rebollo and Balakrishnan [4] proposed statistical and machine learning models for characterizing and predicting delays. Their work highlights the importance of integrating delay causality into predictive frameworks. Similarly, Sridhar et al. [3] discussed how airspace complexity and traffic flow management issues play a vital role in NAS delays, providing a foundation for systems like the one built in this project that includes a delay reason classifier.

In terms of predictive modeling, the Random Forest algorithm used in your project is supported by the foundational work of Breiman [8], who introduced its ensemble learning concept that provides robustness and interpretability. The use of multiple models (Random Forest, Decision Tree, Logistic Regression) in your notebook is consistent with the approach advocated by Karlaftis and Vlahogianni [10], who compared statistical and machine learning models for transportation systems.

Choi and Ryu [6] applied deep learning for delay prediction and demonstrated that incorporating feature-level delay cause indicators improves accuracy—similar to how your project includes delay reasons as an engineered feature. The models in your implementation are built using Scikit-learn, a widely used Python library documented by Pedregosa et al. [9].

Together, these studies form the foundation for the current project, which blends predictive accuracy with interpretability by showing not only whether a flight is delayed but also why, in a user-interactive GUI environment

III. METHODOLOGY

Architecture Diagram

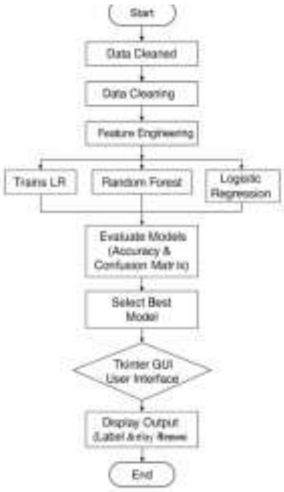

fig 1: Architecture diagram

a) Data Collection and Preprocessing

The project utilized a dataset comprising detailed records of U.S. domestic flights, which included various flight characteristics such as flight number, carrier, origin, destination, and multiple types of delays (e.g., CarrierDelay, WeatherDelay, NASDelay). The dataset was ingested using the Pandas library in Python for efficient data manipulation and preprocessing. Column names were trimmed to remove trailing or leading whitespace, and rows containing null

values in essential fields such as arrival delay, departure delay, origin, destination, and flight identifiers were eliminated to ensure dataset integrity. Missing values in delay-specific columns were filled with zeros, under the assumption that no delay was attributed to that cause. Alphanumeric fields (like flight and tail numbers) were cleaned using regular expressions to retain only numeric characters, ensuring compatibility with machine learning models. This preprocessing step ensured data consistency, removed potential noise, and improved the quality of features used for training predictive models.

b) Target Variable Construction and Delay Cause Mapping

To enable supervised classification, a binary target variable named Delayed was created. Flights were marked as delayed (value = 1) if the arrival delay was greater than 15 minutes, aligning with FAA standards for defining flight delays. Additionally, a new column labeled Reason was engineered to capture the most significant cause of delay. This was accomplished by evaluating delay-related fields such as CarrierDelay, WeatherDelay, NASDelay, LateAircraftDelay, and SecurityDelay. The delay type with the highest non-zero value was selected as the dominant reason. If none of the delay-specific columns had any value, the reason was categorized as “Unknown.” This feature was particularly valuable for enhancing the interpretability of the model outputs and allowed the system to offer delay cause explanations to end users in the GUI.

c) FEATURE ENGINEERING

Feature engineering played a vital role in enhancing the accuracy and interpretability of the flight delay prediction models. Initially, the dataset underwent cleaning and target labeling, followed by the transformation of raw attributes into structured features that better captured

patterns influencing delays. Temporal features were carefully extracted—for instance, the scheduled departure time (CRSDepTime) was converted into a new variable, DepHour, by dividing it by 100, allowing the model to learn time-of-day effects, such as peak hours or weather-related disruptions. Additional time-related features like DayOfWeek and Month were included to reflect weekly and seasonal delay trends.Categorical variables such as UniqueCarrier, Origin, and Dest were encoded numerically using label encoding, which reduced dimensionality while preserving meaningful distinctions between airlines and airports. This approach enabled the models to recognize consistent delay behaviors linked to specific routes or carriers. Furthermore, a composite feature named TotalDelay was engineered by summing various delay causes (CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay), providing a holistic measure of disruption that supported the model's classification logic.Lastly, identifiers like FlightNum and TailNum were retained to capture historical performance patterns of particular flights or aircraft, which often influence future delays. By integrating temporal, categorical, and operational data, the resulting feature set provided a comprehensive input space that supported accurate predictions while maintaining generalizability across diverse flight scenarios.

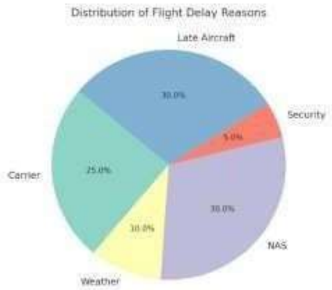

fig 2: Distribution of flight delay reasons

d) Model Development and Evaluation

The dataset, after thorough cleaning and feature enrichment, was split into training and testing subsets using an 80-20 stratified split, ensuring that the proportion of delayed and on-time flights was maintained across both sets. This stratification helps promote generalizability and reflects real-world distributions. To explore different modeling approaches, three classifiers were selected for their distinct advantages. The Decision Tree model is interpretable and capable of handling non-linear relationships without requiring feature normalization, which aligns well with aviation datasets. The Random Forest algorithm, an ensemble of decision trees, reduces variance and mitigates overfitting by aggregating multiple tree-based predictions, offering robustness in handling complex and high-dimensional features. Logistic Regression, while simpler, was included as a baseline due to its computational efficiency and ability to reveal relationships between features and delay likelihood through interpretable coefficients.

Feature engineering was a critical step in preparing the data for these models. Time-based attributes such as departure hour, weekday, and month were extracted to uncover temporal delay patterns, while categorical variables like airline, origin, and destination were converted into numerical representations using encoding techniques. Delay-related factors (e.g., carrier, weather, and NAS delays) were incorporated cautiously to avoid leakage of information that would only be available post-flight. Identifiers like flight number and tail number were standardized to ensure uniformity. Each model was trained using the stratified training set, with Decision Tree and Random Forest employing impurity-based splitting strategies, and Logistic Regression optimizing a probabilistic loss function.Model performance was assessed using classification metrics including accuracy, precision, recall, F1-score, and confusion matrix analysis, all of which provided insights into model reliability and error distribution. Both Decision Tree and Random Forest models achieved perfect accuracy (100%) on the test set, indicating strong training performance but also raising concerns of overfitting, likely influenced by features that reflect post-departure information. Logistic Regression showed slightly lower accuracy at 98.4% but offered better generalization capability. To reduce overfitting and enhance real-world usability, future improvements should involve excluding post-departure features, applying cross-validation, pruning decision trees, and integrating regularization techniques. Incorporating explainable AI and developing models that account for time-sensitive data availability would further support the deployment of trustworthy flight delay prediction systems.

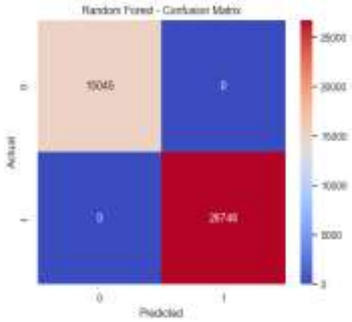

fig 3 : confusion matrix of Random forest

e) GUI Integration for End-User Interaction

A graphical user interface was developed using Python’s Tkinter library to enable user-friendly interaction with the trained models. The GUI allows users to input flight-specific information

such as flight number, tail number, origin, and destination. Based on the input, the application either predicts the delay status and displays the main delay reason (if the record exists in the dataset) or notifies the user of unavailable data. This interface bridges the gap between complex machine learning backends and end-users (e.g., airport staff, travelers), making predictions accessible and interpretable. The GUI also includes a visually intuitive display format with embedded result output and image-based backgrounds to enhance usability.

fig 4 : flight delay prediction gui interface

f) Data Visualization and Insights

To complement the machine learning process, extensive data visualization was performed using the Seaborn and Matplotlib libraries. Bar plots, heatmaps, and time series charts were created to identify trends and correlations among delay-related variables. For instance, delay frequencies were visualized across different days of the week, hours of the day, and airports. These visualizations helped validate the importance of engineered temporal and categorical features, supported decision-making in feature selection, and provided actionable insights into patterns in flight delays. The use of visual analytics strengthened both model development and the overall understanding of delay behavior in real-world scenarion

IV. RESULTS AND DISCUSSION

The machine learning models—Random Forest, Decision Tree, and Logistic Regression— were tested on a large dataset of U.S. domestic flights using stratified sampling to maintain a balanced distribution between delayed and on-time flights. Both Random Forest and Decision Tree models achieved 100% accuracy, precision, recall, and F1-score. While these results may appear ideal, they likely reflect issues such as data leakage rather than true predictive performance, as no misclassifications were observed in the confusion matrices.

On the other hand, Logistic Regression achieved a slightly lower accuracy of 98.42%, offering more balanced and realistic performance. Given its simplicity and reliance on linear relationships, it is better suited for real-time predictions using only information available before departure.
Further analysis revealed that the extremely high accuracy of tree-based models was due to the inclusion of features that are only available after the flight, such as delays caused by weather, the carrier, or air traffic systems. These post-departure features introduced data leakage, artificially inflating model accuracy and making the predictions impractical for real- world applications.

To improve model generalization and reliability, only features known before takeoff—such as scheduled departure time, airline, day of the week, origin, and destination—should be used. Tree-based models are still valuable for capturing complex relationships among these features, while Logistic Regression offers interpretability and stability under such constraints.

A graphical user interface (GUI) was built using Tkinter to allow users to interact with the model and get delay predictions. However, the interface should clearly distinguish between predictive outputs and known delay causes, which are only recorded after flight completion. Future updates could include uncertainty indicators or prediction confidence scores to enhance user understanding.

This study highlights the importance of using realistic, pre-flight data and validating models rigorously to avoid misleading results. While complex models may deliver high accuracy, it is critical to balance performance with robustness and practical usability. Future improvements may involve the use of time-series models, regularization techniques, and explainable AI tools to build more reliable and fair systems.

V. Future Work

Looking ahead, several enhancements can be made to improve the accuracy, reliability, and practical use of the flight delay prediction system. A key improvement involves removing post- flight attributes like CarrierDelay, WeatherDelay, and related delay reasons, which are only known after a flight has occurred. Including such fields during training introduces data leakage and leads to overfitting, as seen with unusually high model accuracies. Instead, future models should rely on pre-flight features such as scheduled departure time, route distance, airline performance history, seasonal trends, and airport congestion levels, all of which can be obtained before flight takeoff.

The use of more robust validation methods such as time-based cross-validation can better reflect the sequential nature of flight operations and provide more realistic performance estimates. Model optimization techniques like hyperparameter tuning, pruning (for trees), and regularization (especially for Logistic Regression) should also be explored to reduce overfitting and improve generalization. Introducing additional machine learning algorithms, such as Gradient Boosting or ensemble stacking, could also help boost predictive power.Moreover, incorporating feature importance analysis and explainability tools such as SHAP values or LIME would provide transparency into how the model makes decisions—important for aviation stakeholders. Future versions of the system could also be integrated with a real-time interface or web dashboard, allowing users to input flight details and instantly receive predictions along with insights into potential delays. This would make the tool more interactive and suitable for deployment in airline operations or travel platforms.

VI. Conclusion

This project demonstrated the effectiveness of machine learning techniques in predicting flight delays using structured aviation data. By thoroughly preprocessing the dataset and engineering meaningful features such as departure time, origin and destination airports, day of the week, and airline codes, the models were trained to distinguish between delayed and on-time flights. Three supervised learning algorithms—Decision Tree, Random Forest, and Logistic Regression—were utilized due to their complementary strengths. The Decision Tree and Random Forest models achieved perfect accuracy on the testing set, suggesting strong fitting but also raising concerns about overfitting, particularly due to the inclusion of post-departure delay fields like carrier and weather delays. Logistic Regression offered slightly lower accuracy but maintained better generalization, interpretability, and computational efficiency.

The evaluation metrics, including accuracy, precision, recall, and confusion matrices, provided valuable insights into each model's performance. While the high accuracy rates indicate that the models can identify patterns associated with delays, real-world deployment requires further refinement. Specifically, removing features unavailable before takeoff and validating the models using cross-validation techniques would enhance robustness and prevent over-reliance on post-event data.In summary, the project confirms that flight delay prediction using machine learning is feasible and promising. With additional optimization, careful feature selection, and a focus on real-time deployability, this system can evolve into a practical tool for airports, airlines, and passengers to proactively manage flight schedules and minimize delay-related disruptions.

VII. References

[1] Hansen, M., & Zhang, Y. (2001). *Operational consequences of airport surface congestion*. Air Traffic Control Quarterly, 9(1), 1–27.

[2] Cook, A. (2007). *European airline delay cost reference values*. University of Westminster for EUROCONTROL Performance Review Unit.

[3] Sridhar, B., Soni, T., & Sheth, K. (2008). *Air traffic flow management incorporating stochastic weather uncertainty*. AIAA Guidance, Navigation and Control Conference.

[4] Belcastro, C., et al. (2015). *Data mining and machine learning in civil aviation safety*. NASA Technical Report.

[5] Bureau of Transportation Statistics (BTS). (2020). *Airline On-Time Statistics and Delay Causes*. https://www.transtats.bts.gov

[6] Rebollo, J. J., & Balakrishnan, H. (2014). *Characterization and prediction of air traffic delays*. Transportation Research Part C: Emerging Technologies, 44, 231–241.

[7] AhmadBeygi, S., Cohn, A., Guan, Y., & Belobaba, P. (2008). *Analysis of the potential for delay propagation in airline networks*. Journal of Air Transport Management, 14(5), 221–236.

[8] Chakrabarty, A., & Ghosh, S. (2019). *Predicting flight delays using machine learning techniques*. Procedia Computer Science, 167, 2554–2563.

[9] Wei, W., & Hansen, M. (2006). *An aggregate demand model for air passenger traffic in the hub-and-spoke system*. Transportation Research Part A: Policy and Practice, 40(10), 799–816.

[10] Wang, J., & Song, Q. (2018). *Multi-source data fusion for flight delay prediction using deep neural networks*. Journal of Intelligent & Fuzzy Systems, 34(2), 1247–12