

# Beyond the Model: A Deployable Machine Learning System for Flight Delay Prediction Using Comparative Regression Analysis

1<sup>st</sup> **Rajashekhara G C**, Assistant Professor and Director, Faculty of Computing and IT, G M University  
Davanagere

2<sup>nd</sup> **Vinayak Nagendra Pisale**, Student 4<sup>th</sup> Semester MCA, GM University Davanagere

## Abstract

Flight delays remain one of the most significant operational challenges in the aviation industry, affecting passengers, airlines, and airport management through increased travel uncertainty, scheduling disruptions, and financial losses. Accurate prediction of flight delays before departure can support better travel planning and improve operational decision-making. This paper presents a machine learning-based flight delay prediction system that estimates arrival delays using historical flight information from Indian domestic airlines. The proposed system employs four supervised machine learning algorithms, namely Ridge Regression, Decision Tree Regression, Random Forest Regression, and Extra Trees Regression, to evaluate their predictive capabilities. We have a set of information about flights, around 80,000 records. We clean this data. Make it ready for our model to learn from it. We do this by cleaning the data changing some features into codes that the model can understand and making sure all the numbers are on the scale. We then use this model to make predictions. We check how good it is at doing this. We use some ways to measure this like Mean Absolute Error and the coefficient of determination, which is also called  $R^2$ . The model is part of a website that we made using FastAPI, HTML, CSS, JavaScript and SQLite. This website lets users do a things. They can get predictions about flight delays look at predictions and even download reports about these predictions. The flight delay predictions and the reports are all available, on this website made with FastAPI, HTML, CSS, JavaScript and SQLite. Comparative analysis identifies the most suitable model for accurate delay prediction while demonstrating the practical application of machine learning in aviation decision support. The proposed system offers an efficient and user-friendly solution that can assist passengers in travel planning and provide a foundation for future integration with real-time operational data.

**Keywords:** *Flight Delay Prediction, Machine Learning, Ridge Regression, Decision Tree Regression, Random Forest Regression, Extra Trees Regression, Predictive Analytics, FastAPI, Aviation Data, Regression Models.*

## I. Introduction

When you have flown on a plane a lot you know what to do. You check in for your flight you get to the gate. Then you wait for the plane to take off. The thing is, a lot of flights are delayed nowadays. This is because more and more people are flying every year. Delays are pretty normal now. They usually do not happen because of just one thing. Sometimes the weather is bad. The airport is really crowded or there is a problem with air traffic control or the plane has a mechanical issue or it is just running late from its last flight. Any of these things can make a flight late.. It is not just the people on the flight who get upset. The airlines do not work well as they should the airports have to change things around like which gate to use and who works where and this causes problems for the whole system. Most of the things that people use to check on their flights are not very helpful. They tell you that your flight is late after it is already late. They do not tell you before it is late. There are some things that show you where a flight is and if it is, on time. They only show you what is happening right now. They do not try to guess what will happen. Meanwhile, researchers have spent years building machine learning models that predict delays fairly accurately — but a lot of that work stays locked in academic papers and never turns into something a regular traveler could open and actually use. There's a real gap between having a good prediction model and having something practical built around it. That gap is what this project tries to close. Machine learning is well suited to this kind of problem — it can dig through large, messy operational datasets and pick up on patterns that traditional statistical methods tend to miss. But not every algorithm works the same way on every dataset, so it made sense to test a few and see what

actually performs best rather than assuming one approach would win. Working with historical data from Indian domestic flights, I built and compared four regression models — Ridge, Decision Tree, Random Forest, and Extra Trees — to figure out which one predicts delays most accurately. I cleaned up the data. Made sure I had the right features. Then I took the model that worked the best. I used it to build a real application that actually works. The backend of the application is handled by FastAPI. The front end of the application is run by HTML and CSS and JavaScript. FastAPI and HTML and CSS and JavaScript and SQLite all work together. FastAPI handles the backend. HTML and CSS and JavaScript run the end.. Sqlite takes care of the storage for the application. Users can type in their flight details, get a delay prediction back, look up their past searches, and download a report if they need one. The main goal here is twofold: figure out which algorithm handles this dataset best, and show that the result doesn't have to stay trapped in a spreadsheet or a research paper. Turning it into a working tool — however simple — is a small step toward helping people actually plan their trips with a bit more confidence. Looking ahead, there's room to grow this further with live weather feeds, real-time flight tracking, and possibly deep learning models for even sharper predictions.

## II. Literature Review

Recent improvements in machine learning have really helped us get better at figuring out when flights will be late. We can use flight information to make these predictions. People have been trying ways to predict delays like using regular machine learning combining different methods and using deep learning models. The goal is to make predictions more accurate so airlines can

make decisions about flights. Machine learning is helping us make these predictions, about flight delays. Several studies have also explored feature engineering, class imbalance handling, hyperparameter optimization, and explainable artificial intelligence to improve model performance. Despite these developments, many existing approaches remain limited by restricted datasets, high computational complexity, or insufficient comparison among predictive models. Therefore, a comprehensive evaluation of multiple machine learning algorithms using a consistent dataset remains an important research direction.

| N o. | Authors                      | Object ive                           | Method                                      | Key Findin g                      | Gap Identified                   |
|------|------------------------------|--------------------------------------|---------------------------------------------|-----------------------------------|----------------------------------|
| 1    | Mtimkul u & Maphos a (2023)  | Predict flight delays                | RF, AdaBoost, GB, Bagging, Stacking         | Stacking achieved 92.4% accuracy  | Only ensemble models compared    |
| 2    | Jha et al. (2024)            | Predict departure and arrival delays | Hybrid ML, ANN, RF, XGBoost                 | Hybrid model performed best       | High complexity; U.S. data only  |
| 3    | Hatipoğlu & Tosun (2024)     | Predict delays over 15 min           | LR, NB, NN, RF, XGBoost, CatBoost, LightGBM | XGBoost achieved highest accuracy | Single airport dataset           |
| 4    | AlBassam & AlShahrani (2025) | Compare ML models                    | DT, RF, SVM, LR, KNN, NB, SMOTE             | RF and DT gave best results       | No deep learning comparison      |
| 5    | Dai (2024)                   | Predict delays using big data        | DBSCAN, ANOVA, FSFS, COWRF                  | Achieved 97.2% accuracy           | High computational cost          |
| 6    | Guiet al. (2020)             | Predict delays using aviation data   | RF, LSTM, ADS-B Data                        | RF reached 90.2% accuracy         | Limited deep learning evaluation |
| 7    | Zhong et al. (2025)          | Predict delays with deep learning    | LSTM, GCN, 3D-CNN, Attention                | Reduced prediction error          | Complex model architecture       |

|    |                       |                                       |                         |                              |                        |
|----|-----------------------|---------------------------------------|-------------------------|------------------------------|------------------------|
| 8  | Kothari et al. (2023) | Predict airline delays                | RF, Neo4j, Path Finding | RF achieved 98.2% accuracy   | No model comparison    |
| 9  | Liet al.              | Predict delays using spatial features | LSTM + RF               | Improved prediction accuracy | Limited explainability |
| 10 | Cai et al.            | Predict delays using graph learning   | GCN, Deep Learning      | Better network modeling      | High computation time  |

### III.Dataset Description

The quality of any machine learning model comes down to the data behind it, so this study relies on a structured dataset built from historical Indian domestic flight records — roughly 80,000 flights in total. Each record captures details like flight schedules, airline information, departure and destination airports, weather conditions, traffic density, and past operational performance. Together, these give the models enough signal to pick up on the patterns that usually lead to delays. Before training anything, I spent time exploring the data — looking at how individual features were distributed and how they related to the target variable. The dataset mixes numerical and categorical attributes, so it needed proper preprocessing before it could be fed into any model. The features chosen here reflect the kinds of operational conditions that typically affect whether a flight runs on time. The actual target being predicted is arrival delay — how many minutes late a flight is expected to be. Using supervised regression, the models were trained to learn the relationship between all these input variables and that delay figure.

#### A. Dataset Summary

| Parameter             | Description                                     |
|-----------------------|-------------------------------------------------|
| Dataset Type          | Historical Indian Domestic Flight Data          |
| Number of Records     | 80,000                                          |
| Number of Features    | 15                                              |
| Prediction Type       | Regression                                      |
| Target Variable       | Arrival Delay (minutes)                         |
| Airlines Covered      | IndiGo, Air India, SpiceJet, Vistara, Akasa Air |
| Data Format           | CSV                                             |
| Machine Learning Task | Flight Delay Prediction                         |

Table 2. Summary of the Dataset

## B. Input Features

The dataset consists of fifteen variables representing operational and environmental conditions associated with each flight. These attributes are used as input variables during model training.

| Feature          | Type               | Description                             |
|------------------|--------------------|-----------------------------------------|
| Flight Number    | Categorical        | Unique flight identifier                |
| Airline          | Categorical        | Airline operating the flight            |
| Source           | Categorical        | Departure airport                       |
| Destination      | Categorical        | Arrival airport                         |
| Month            | Numerical          | Month of travel                         |
| Day              | Numerical          | Day of the month                        |
| Day of Week      | Numerical          | Scheduled weekday                       |
| Departure Hour   | Numerical          | Scheduled departure time                |
| Distance         | Numerical          | Route distance (km)                     |
| Air Time         | Numerical          | Estimated flight duration (minutes)     |
| Departure Delay  | Numerical          | Departure delay (minutes)               |
| Weather Score    | Numerical          | Weather severity index                  |
| Traffic Score    | Numerical          | Airport traffic congestion index        |
| Historical Delay | Numerical          | Average historical delay of the airline |
| Arrival Delay    | Numerical (Target) | Actual arrival delay (minutes)          |

**Table 3. Description of Dataset Features**

## IV. Data Preprocessing

The quality of the data is really important for the model so before we started training we cleaned up the flight dataset. We did the things: we fixed the missing values removed the duplicates cleaned up the messy parts changed the categories into numbers made sure all the numbers were on the same scale and split everything into two groups. One for training and one, for testing.

### A. Missing Values

First step was checking for gaps in the data. Where numerical fields were missing, I filled them in using the mean or median — nothing fancy. Missing categorical entries got a reasonable placeholder category where it made sense. By the end, there were no half-empty records left to confuse the training process.

### B. Duplicate Records

Duplicates are sneaky — they don't look wrong, but they quietly bias a model by making certain flights count more than they

should. So these got flagged and removed, leaving one row per actual unique flight.

### C. Data Cleaning

This part was more manual. Airline names, airport codes, and date formats needed to actually match up across the dataset — small inconsistencies like that add up. I also checked for outliers and values that just didn't make sense (like negative distances or impossible delay times) and dealt with them before moving on.

### D. Categorical Encoding

Airline, departure airport, destination airport, flight number — all text, and regression models don't understand text. So each of these got converted into numbers through encoding, without losing what made each category distinct.

### E. Feature Scaling

Flight distance, departure delay, weather score, traffic score, historical delay — these numbers live on completely different scales. Left alone, the bigger numbers would just drown out the smaller ones during training. Normalizing brought everything onto the same footing, which made training noticeably more stable.

### F. Train-Test Split

Last step: split the data 80/20. Eighty percent went into training, the rest was set aside so the models could be tested on flights they'd never actually seen.

| Dataset      | Percentage | Number of Records |
|--------------|------------|-------------------|
| Training Set | 80%        | 64,000            |
| Testing Set  | 20%        | 16,000            |
| Total        | 100%       | 80,000            |

**Table 4. Dataset Partition**

### G. Preparation of Data for Machine Learning

The data was cleaned up. Changed into a format that works well for supervised machine learning. Input features representing operational flight characteristics were separated from the target variable, which corresponds to the arrival delay in minutes. The dataset was then given to the four regression algorithms. Ridge Regression, Decision Tree Regression, Random Forest Regression and Extra Trees Regression. So they could learn from it. Be tested. This process made sure the models got data that was correct and consistent which's important for predicting flight delays, with Ridge Regression, Decision Tree Regression, Random Forest Regression and Extra Trees Regression. This helps us get an accurate prediction of flight delays with these regression algorithms.

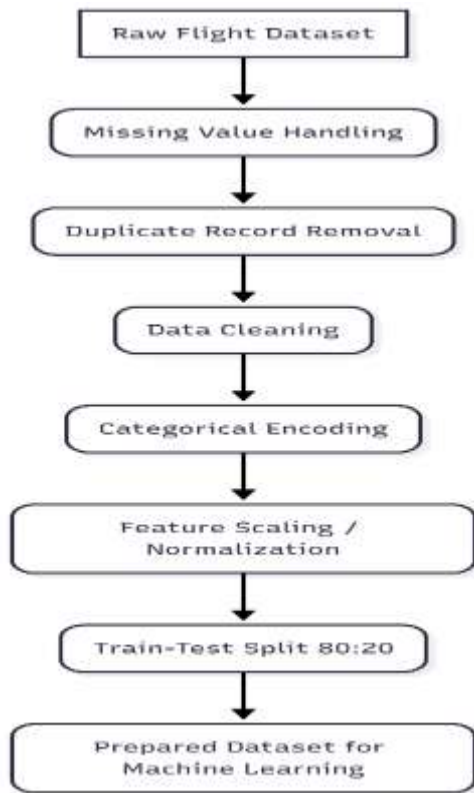


Figure 3. Data Preprocessing Workflow

## V. Feature Engineering

Feature engineering is really about picking and shaping the right inputs so the model has something meaningful to learn from — because no matter how good an algorithm is, it's only as useful as the features feeding into it. Here, the goal was to represent flight operations in a way that actually captures what drives delays.

The dataset has fifteen attributes covering flight schedules, airline details, airport locations, weather, traffic congestion, and past operational performance. Each of these was reviewed for how much it actually contributes to predicting arrival delay. Both time-based and operational factors were kept in the mix, since delays rarely come down to one cause — it's usually a combination of things happening together.

### A. Feature Selection

Features were chosen based on how directly they connect to flight operations and delay risk. Departure delay, weather, airport traffic, and historical airline performance all stayed in, since they hint at possible disruptions before they happen. On the time side, month, day of the month, day of the week, and scheduled departure hour were included too — delays often follow seasonal or daily patterns, so these help the model pick up on that.

Together, these features give the regression models enough to work with to learn how flight conditions translate into delays.

### B. Feature Categories

To make the data easier to interpret, the selected attributes were grouped into four categories based on their operational role.

| Category               | Features                                       |
|------------------------|------------------------------------------------|
| Flight Information     | Flight Number, Airline, Source, Destination    |
| Time-related Features  | Month, Day, Day of Week, Departure Hour        |
| Operational Features   | Distance, Air Time, Departure Delay            |
| Environmental Features | Weather Score, Traffic Score, Historical Delay |

Table 5. Categories of Selected Features

Grouping the variables this way made the data easier to work with, and it helped the models pick up on how different operational conditions relate to one another.

### C. Target Variable

What's actually being predicted here is Arrival Delay, measured in minutes — the difference between a flight's scheduled landing time and when it actually touched down. Because this is a continuous number and not a category, the whole problem is framed as regression, not classification. Once the model is trained, it spits out an estimated delay based on whatever input features it's given. If a simpler yes/no view is more useful down the line, those numeric predictions can also be converted into On-Time or Delayed labels using a set threshold.

### D. Importance of Selected Features

Not every feature contributes equally. Departure delay tends to matter most — a flight that leaves late almost always lands late too, since that time rarely gets made up mid-air. Weather plays into how smoothly ground operations and aircraft movement run at the airport. Traffic levels reflect something similar: how congested the runways are and how stretched air traffic control is at that particular time. Historical delay data is useful in a different way — it says a lot about how consistently reliable a given airline tends to be over time. The temporal features, like the travel month and the departure hour help us understand things like when there are holiday rushes seasonal spikes and the congestion that builds up during peak hours. The travel month and the departure hour are important because they show us when things get busy. These features, like the travel month and the departure hour give the model the information it needs to see the reasons, behind most delays and how these reasons can overlap.

### E. Final Feature Set

After preprocessing and feature engineering, the dataset was narrowed down to fourteen predictor variables plus the one target — arrival delay. This final feature matrix is what actually went into training the machine learning models.

| Variable Type   | Number of Variables |
|-----------------|---------------------|
| Input Features  | 14                  |
| Target Variable | 1                   |
| Total Variables | 15                  |

Table 6. Final Feature Set

The engineered feature set provides a comprehensive representation of flight operations while maintaining a manageable number of variables for efficient model training and evaluation.

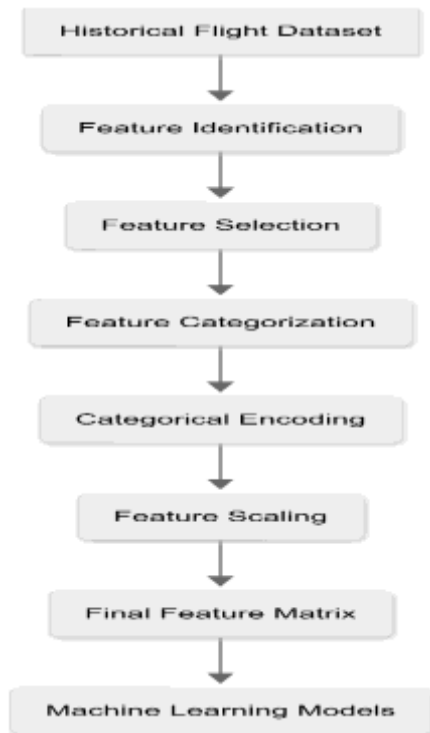


Figure 4. Feature Engineering Workflow

## VI. Proposed Methodology

This study builds a machine learning framework to predict flight arrival delays using historical data from Indian domestic airlines. The process runs through data collection, preprocessing, feature engineering, model training, prediction, and evaluation — with the end goal of finding which regression algorithm predicts delays most accurately without being computationally heavy. The winning model then gets wrapped into a web app so users can actually check delay estimates themselves.

### A. Overall Workflow

The workflow is made up of six parts. These parts are data collection, preprocessing, feature engineering, model training, prediction and evaluation. We do these things to make sure the data collection and the model training and the prediction and the evaluation all work together smoothly. The workflow is important because it includes data collection, preprocessing, feature engineering, model training, prediction and evaluation. These six parts are what make up the pipeline. Data collection is the part of the pipeline. Then comes preprocessing, followed by feature engineering. After that we have model training, prediction and finally evaluation.

### B. Data Collection

The dataset starts with roughly 80,000 historical flight records from Indian domestic operations — covering airlines, airports, departure schedules, weather, traffic, past delays, and arrival delays. This is what the models train on.

### C. Data Preparation

Before we start training we need to clean up the data. We handle missing values in the data. We also drop any values. The categorical fields in the data get encoded. The numerical fields, in the data get normalized. When we have cleaned up the data we split it into two sets. One set is for training. The other set is, for testing. We use most of our data for training eighty percent of the data is for training. The rest, which is twenty percent of the data is, for testing. This is how we split our data into a training set and

a testing set. We have a training set that's eighty percent of the data and a testing set that is twenty percent of the data. This way the evaluation is unbiased. We can trust the results of the data preparation and the overall workflow.

### D. Model Training

Four regression models were trained on the prepared data — Ridge, Decision Tree, Random Forest, and Extra Trees. Each one learns its own version of the relationship between flight characteristics and arrival delay, and the trained models get saved for later use on new flights.

### E. Flight Delay Prediction

Users interact with the app by entering flight details — airline, source and destination airport, travel date, departure time, and other parameters. The model takes these inputs and outputs an estimated arrival delay in minutes, along with an On-Time/Delayed label based on a set threshold.

### F. Comparative Performance Analysis

To compare the four models fairly, each was tested on identical training/testing splits and scored using Mean Absolute Error (MAE) and  $R^2$  Score. This comparison is what determines which algorithm actually works best for this dataset.

### G. Web-Based Deployment

The best-performing model was built into a FastAPI web application where users can register, log in, enter flight details, get delay predictions, check their prediction history, and download PDF reports. It's a working example of how machine learning can support real travel planning, not just sit in a research paper.

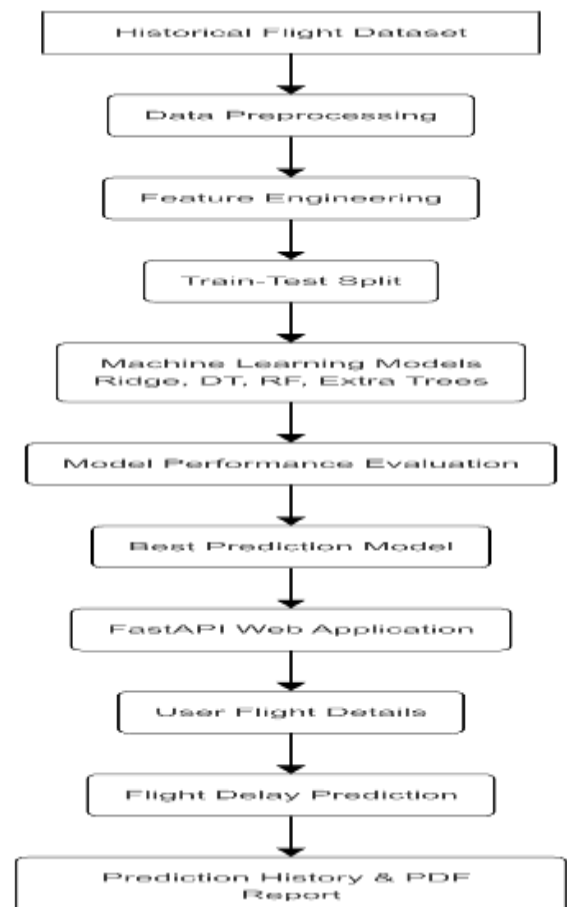


Figure 5. Proposed Methodology Flowchart

## VII. Machine Learning Models

Machine learning algorithms are like tools that help computers look for patterns, in data and make guesses without being told exactly what to do every time. Random Forest Regression is one of these machine learning algorithms. Since many things can cause flight delays like how things run, when things happen and the environment picking the right prediction model is very important to get good results. In this study, four supervised machine learning regression algorithms were selected and compared to estimate flight arrival delays. The models were chosen because they represent different learning approaches and have demonstrated effectiveness in regression-based prediction problems. All models were trained using the same pre processed dataset and evaluated under identical experimental conditions to ensure a fair comparison.

### A. Ridge Regression

Ridge Regression is a way to make predictions that is related to regression but it also includes a penalty to prevent the coefficients from getting too big. This helps when the things we are looking at are connected to each other so the predictions are more stable. We do not get bad results because the model is too complicated. Ridge Regression gives us predictions and is also fast which makes it good for problems where we have to look at many things that are connected to each other.

The objective function of Ridge Regression is expressed as:

$$J(\beta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

where:

- $y_i$  = Actual arrival delay
- $\hat{y}_i$  = Predicted arrival delay
- $\beta_j$  = Regression coefficients
- $\lambda$  = Regularization parameter

The goal of Ridge Regression is to control how complicated the model is by not letting the coefficients get too big which helps the predictions be more stable and accurate.

### B. Decision Tree Regression

Decision Tree Regression is a way to predict values by dividing the data into smaller groups based on the values of the things we are looking at. Each part of the tree represents a decision. Each end part represents the predicted delay value. The algorithm automatically learns nonlinear relationships between flight characteristics and arrival delays without requiring complex mathematical assumptions.

The process of making a prediction starts at the beginning of the tree and follows a sequence of decisions until we get to the end, where we find the predicted delay value, for flight delays.

### C. Random Forest Regression

Random Forest Regression is a way of learning that puts together decision trees to make better predictions and reduce mistakes. Each decision tree is trained using a group of the data we have and the features we are looking at. The final prediction is obtained by averaging the outputs of all decision trees.

The Random Forest prediction is calculated as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N T_i(x)$$

where:

- $N$  = Number of decision trees
- $T_i(x)$  = Prediction produced by the  $i^{th}$  tree

When we combine separate models Random Forest Regression makes predictions that are more reliable than a single decision tree.

### D. Extra Trees Regression

Extra Trees (Extremely Randomized Trees) Regression is another ensemble learning technique that extends the Random Forest approach by introducing additional randomness during tree construction. Instead of searching for the optimal split at each node, Extra Trees randomly selects split thresholds, reducing computational complexity and increasing model diversity.

The prediction is obtained by averaging the outputs of all generated trees:

$$\hat{y} = \frac{1}{M} \sum_{i=1}^M E_i(x)$$

where:

- $M$  = Number of Extra Trees
- $E_i(x)$  = Prediction generated by the  $i^{th}$  tree

This increased randomness often improves generalization performance while reducing variance.

## VII. Results and Comparative Analysis

We tried out four models on the same information about flight delays: Ridge Regression, Decision Tree, Random Forest and Extra Trees. We did the thing with each of these models so any differences, in how well they work come from how each model uses the information to learn. Not because something was set up differently with one model.

Two metrics were used to judge them. MAE shows how far off predictions typically are, in real terms — lower is better. The  $R^2$  value tells us how well the model explains the delay pattern. When the  $R^2$  value is close to 1 it means the model is a fit.

Ridge Regression is a type of model that assumes the relationship between the inputs and the delay is pretty straight. It also uses something called regularization to prevent the model from getting too complicated. It's reliable, but not great at picking up on messier, conditional patterns. The Decision Tree model can find relationships that're not straight. If the tree gets too big it can start to fit the data too closely. This is where the Random Forest model comes in. It uses trees and averages their predictions, which helps to make the predictions more stable. The Random Forest model is basically an improvement, on the Decision Tree model. Extra Trees pushes randomness even further during training, usually making it generalize a bit better without slowing things down much.

Across the board, the tree-based models did a noticeably better job than Ridge at capturing the kind of interactions you'd expect in real flight data — like delays compounding when bad weather hits an already congested airport. The ensemble models (Random Forest and Extra Trees) were the most consistent overall, since combining multiple trees smooths out the noise a single tree might latch onto.

What this really shows is that flight delays aren't random — there's enough signal in departure delay, weather, airport traffic, and airline history to predict them with reasonable accuracy, as long as the model can handle nonlinear relationships. Data cleaning and feature selection mattered here too; better inputs meant better predictions across all four models.

## VIII. Discussion

Testing four models side by side made it pretty clear that the algorithm you pick actually matters here, not just as a technical detail but for how usable the final predictions are.

Ridge Regression is the safe, stable option — good when you want consistency and don't want the model reacting too strongly to correlated features. Decision Tree is easy to interpret and catches nonlinear patterns, but on its own it's prone to overfitting. Random Forest solves that by averaging across trees, at the cost of being harder to interpret. Extra Trees goes a step further with extra randomness in how it builds each tree, which often helps it generalize better while still training fast.

Feature selection ended up mattering just as much as model choice. Departure delay was the standout predictor — once a flight leaves late, that delay usually doesn't just disappear, it tends to carry through the rest of the trip. Weather and airport congestion added more disruption on top of that, and historical airline performance gave a sense of how reliably a carrier typically operates. Time-based features like departure hour and travel month picked up on seasonal and peak-period effects. All of this gave the models real patterns to learn from, not just noise.

Beyond the comparison itself, this project also tried to answer a more practical question: can this kind of prediction actually be useful to someone? Wrapping the trained model in a FastAPI web app was the attempt at that — a place where someone can enter flight details, get a prediction, look back at past searches, and download a report. It's a small step, but it's the difference between a model sitting in a notebook and something a person could actually open and use.

This lines up with earlier research showing ensemble methods like Random Forest and XGBoost tend to perform well on delay prediction, and newer work pushing further into deep learning. Most of that research stops at the model, though. What's different here is pairing the comparison with something people can actually interact with — logins, prediction history, PDF reports — instead of leaving the results in a spreadsheet.

## IX. Conclusion

This study presented a machine learning-based approach for predicting flight arrival delays using historical Indian domestic flight data. The primary objective was to develop an accurate and practical prediction system while conducting a comparative performance analysis of multiple regression algorithms. We used four machine learning models. Ridge Regression, Decision Tree Regression, Random Forest Regression and Extra Trees

Regression. We implemented these machine learning models. Then checked how well they worked. We did this by using the steps to prepare the data and the same standards to evaluate the results of Ridge Regression and the other machine learning models. Our approach was very structured. We started by getting the dataset ready. Then we cleaned up the data. This included fixing missing values getting rid of records and making sure the numbers were on the same scale. We also changed the categories into numbers that the machine learning models could understand. After that we split the data into two parts. One for training and one for testing. This way we could make sure that our results, for Ridge Regression and the other machine learning models were fair and accurate. Then we trained the machine learning models. Checked how well they worked. Finally we put the machine learning models into a web application. The comparative evaluation framework demonstrated the importance of assessing multiple algorithms rather than relying on a single predictive technique. The selected regression models provide different learning characteristics, enabling the study to analyze their effectiveness in estimating flight arrival delays under identical experimental conditions. This comparative approach supports the identification of the most appropriate prediction model based on accuracy, reliability, and generalization capability. The developed Flight Delay Prediction System extends beyond model development by integrating the selected prediction model into a FastAPI-based web application. The system enables users to enter flight information, obtain predicted arrival delays, review previous prediction records, and generate downloadable PDF reports. This integration illustrates how machine learning models can be transformed into practical decision-support tools that improve user accessibility and operational usability. Overall, the proposed framework demonstrates the applicability of machine learning for flight delay prediction and provides a systematic methodology for comparing predictive models using historical aviation data. The study contributes to the growing use of predictive analytics in the aviation sector by combining comparative model evaluation with a user-oriented software implementation, thereby supporting improved travel planning and informed operational decision-making.

## X. References

### Journal and Conference Papers

- [1] B. Mtinkulu and B. Maphosa, "Flight Delay Prediction Using Machine Learning: A Comparative Study of Ensemble Techniques," *Proc. International Conference on Computing and Data Analytics*, 2023.
- [2] A. Jha, *et al.*, "Flight Delay Prediction Using Hybrid Machine Learning Approach: A Case Study of Major Airlines in the United States," *International Journal of Intelligent Systems and Applications*, vol. 16, no. 2, pp. 45–61, 2024.
- [3] M. Hatipoğlu and E. Tosun, "Predictive Modeling of Flight Delays at an Airport Using Machine Learning Methods," *Applied Sciences*, vol. 14, no. 3, 2024.
- [4] A. AlBassam and M. AlShahrani, "Flight Delay Prediction: Evaluating Machine Learning Algorithms for Enhanced Accuracy," *PLOS ONE*, vol. 20, no. 1, 2025.
- [5] Y. Dai, "A Hybrid Machine Learning Model for Flight Delay Prediction Using Aviation Big Data," *Scientific Reports*, vol. 14, 2024.

[6] G. Gui, *et al.*, "Flight Delay Prediction Based on Aviation Big Data and Machine Learning," *IEEE Access*, vol. 8, pp. 197432–197444, 2020.

[7] Y. Zhong, *et al.*, "FlightNet-ST: An Interpretable Spatio-Temporal Deep Learning Framework for Civil Aviation Flight Delay Prediction," *Expert Systems with Applications*, 2025.

[8] R. Kothari, R. Kakkar, S. Agrawal, P. Oza, S. Tanwar, *et al.*, "Selection of Best Machine Learning Model to Predict Delay in Passenger Airlines," *IEEE Access*, vol. 11, pp. 79673–79689, 2023.

[9] J. Rebollo and H. Balakrishnan, "Characterization and Prediction of Air Traffic Delays," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 2014.

[10] J. Qu, X. Dong, W. Zhang, and Y. Qu, "A Machine Learning Approach for Flight Delay Prediction," in *Proc. IEEE International Conference on Big Data Analysis (ICBDA)*, 2016.

#### Books

[11] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

#### Software and Documentation

[13] FastAPI, "FastAPI Documentation." [Online]. Available: <https://fastapi.tiangolo.com>. Accessed: Jul. 2025.

[14] Scikit-learn, "Scikit-learn Documentation." [Online]. Available: <https://scikit-learn.org>. Accessed: Jul. 2025.

[15] Pandas Development Team, "Pandas Documentation." [Online]. Available: <https://pandas.pydata.org>. Accessed: Jul. 2025.

[16] NumPy Developers, "NumPy Documentation." [Online]. Available: <https://numpy.org>. Accessed: Jul. 2025.

[17] SQLite Consortium, "SQLite Documentation." [Online]. Available: <https://www.sqlite.org>. Accessed: Jul. 2025.

[18] ReportLab, "ReportLab User Guide." [Online]. Available: <https://www.reportlab.com/docs/reportlab-userguide.pdf>. Accessed: Jul. 2025.

[19] Chart.js, "Chart.js Documentation." [Online]. Available: <https://www.chartjs.org>. Accessed: Jul. 2025.

#### Aviation Data Sources

[20] Directorate General of Civil Aviation (DGCA), Government of India, "Monthly Domestic Air Traffic Statistics." [Online]. Available: <https://dgca.gov.in>. Accessed: Jul. 2025.