International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

Comparative Analysis and Full-Stack Deployment of Machine Learning Models for Flight Fare Prediction

[1]Gosala Venkata Charvi, [2]Gaddam Samuel Kiran Babu, [3]Y Pavan Narsimha Rao, [4]Dr.Meera Alphy, [5]Dr.M.Aruna

[1,2]Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India
[3,4,5]Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India
E-mail: [1]gvenkatacharvi_cse2305e5@mgit.ac.in, [2]gsamuelkiranbabu_cse2305e2@mgit.ac.in, [3]ypnarasimharao_cse@mgit.ac.in, [4]meeraalphy_cse@mgit.ac.in, [5]maruna_cse@mgit.ac.in

Abstract **- Domestic flight ticket pricing in India is influenced by multiple dynamic factors including airline carrier, travel route, departure timing, flight duration, and number of intermediate stops. Retrieving accurate fare estimates remains challenging due to continuously changing market demand, seasonal fluctuations, and varying airline pricing strategies. Existing travel and booking platforms primarily display current ticket prices and provide limited support for predictive fare estimation and intelligent booking analysis. Traditional statistical approaches often fail to capture complex nonlinear relationships between journey attributes and ticket prices, reducing prediction reliability across diverse travel scenarios.**

Feature ranking by both mutual information regression and Random Forest importance scores confirms that Duration_total_mins is the dominant predictor, contributing 50.5% of ensemble split importance. Six regression algorithms are benchmarked under identical 75/25 train-test conditions: Linear Regression, Decision Tree, k-Nearest Neighbours (k=5), Gradient Boosting, XGBoost, and Random Forest. XGBoost achieves the highest test-set R^2 of 0.8312 with a Root Mean Squared Error of 1,812.78 INR and a Mean Absolute Percentage Error of 13.36%. The Random Forest tuned via RandomizedSearchCV with three-fold cross-validation attains R^2 = 0.8282 and RMSE = 1,828.80 INR. The serialised model artefact is stored as a Pickle file for deployment in booking advisory applications.

Keywords: Flight price prediction, XGBoost, Random Forest, feature engineering, mutual information, ensemble regression, Indian domestic aviation, hyperparameter tuning, RandomizedSearchCV, machine learning, fare estimation, duration-based prediction.

I. INTRODUCTION

The Indian civil aviation sector ranks among the most dynamic passenger markets in the Asia-Pacific region, with domestic traffic growing at a compound annual rate that has persistently exceeded global averages. Airline revenue management systems price seats dynamically, factoring in remaining inventory, historical demand patterns, day-of-week behaviour, advance-purchase lead time, and competitive position. This asymmetry—carriers have complete price-setting information while passengers have none—creates a planning problem that millions of Indian travellers face each time they book.

Conventional meta-search platforms show live prices but do not project future trajectories. Statistical models built on historical fare data require stationarity assumptions that break down around public holidays, sporting events, and airline capacity changes. Supervised machine learning regression offers a complementary perspective: rather than tracking how a single flight's price evolves over time, it learns a mapping from observable journey attributes to fare, enabling cross-sectional prediction from booking-record data without access to proprietary airline inventory systems.

This paper describes AeroPrice, an end-to-end regression pipeline applied to Indian domestic bookings. The system makes four concrete technical contributions. First, it standardises a reproducible feature engineering workflow for heterogeneous date-time and duration string inputs that are typical of publicly available airline datasets. Second, it applies target-guided ordinal encoding to airline and destination columns, replacing the conventional one-hot approach and capturing the ordinal price-signal embedded in carrier identity. Third, it conducts a systematic comparison of six regression families—spanning linear, tree, ensemble, boosted, and instance-based methods—under identical experimental conditions. Fourth, it reports results calibrated against actual notebook execution outputs rather than estimated values,

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

providing a reproducible benchmark for the Indian domestic fare prediction task.

The remainder of this paper proceeds as follows. Section II surveys prior literature on fare prediction and outlier treatment. Section III discusses related algorithmic work. Section IV formulates the proposed system. Section V details the system architecture. Section VI describes implementation choices. Section VII presents results and evaluation. Section VIII discusses findings, and Sections IX and X conclude the work and identify future directions.

II. LITERATURE SURVEY

Airfare prediction has accumulated a substantial body of literature since the deregulation era made computerised yield management ubiquitous. Etzioni et al. were among the first to frame the consumer-facing problem formally, introducing a system that advised travellers whether to purchase immediately or defer based on observed price trajectories, establishing the buy-versus-wait paradigm that subsequent work has refined.

Groves and Gini extended this framing to a multi-agent reinforcement learning setting, training agents to optimise purchase timing on United States domestic routes. Their work established that machine-learned policies could consistently outperform simple buy-early heuristics, motivating broader adoption of data-driven fare tools. Tziridis et al. subsequently moved from classification to regression, comparing eight machine learning models on Aegean Airlines data and reporting that ensemble methods achieved approximately 88% accuracy on a held-out partition, with gradient-based learners systematically outperforming linear and single-tree baselines.

Wang et al. constructed a framework combining two public US government databases—the Airline Origin and Destination Survey and the Air Carrier Statistics file—with macroeconomic variables. Their Random Forest regressor achieved an adjusted R^2 of 0.869 on market-segment level quarterly price prediction, demonstrating that auxiliary economic signals can substantially reduce prediction error when journey-level features are limited. Janssen et al. addressed the problem from a quantile regression perspective, fitting linear quantile mixed models to capture heteroskedastic fare distributions rather than merely predicting the conditional mean.

In the Indian context specifically, Rajankar et al. surveyed machine learning approaches to flight fare prediction across Indian domestic routes, identifying the number of stops, departure time, and airline identity as the most consistently cited predictors across studies, and observing that Random Forest regressors recurrently ranked highest in accuracy comparisons. Abdella et al. conducted a comprehensive survey of airline ticket price and demand prediction methodologies, cataloguing 47 studies and concluding that ensemble tree-based methods dominate recent literature while noting a persistent gap between model accuracy on historical data and deployment utility for forward-looking passenger guidance.

The importance of outlier treatment in fare datasets is highlighted by Rousseeuw and Leroy, whose foundational treatment of robust regression formalised the IQR fence and Winsorisation procedures that have since become standard preprocessing steps when premium and economy class tickets are mixed in training data. Liu et al. proposed an adaptive context-aware ensemble regressor specifically for airfare that fuses multiple base regressors using a routing mechanism sensitive to flight-level context features, outperforming single-model baselines on Chinese domestic routes. Bergstra and Bengio provided the theoretical and empirical justification for random hyperparameter search over exhaustive grid search, showing near-equivalent performance at a fraction of the computational cost across a broad class of structured regression and classification problems.

III. RELATED WORK

The literature on flight price prediction spans three broad algorithmic traditions, each with distinct assumptions that bear on their suitability for the Indian domestic booking dataset.

Ordinary least squares regression and its regularised variants (Ridge, Lasso) have been applied to airline pricing as interpretable baselines. Their fundamental limitation in this domain is structural: the joint influence of airline identity and destination on price—a strong interaction effect in the Indian market, where carriers such as Jet Airways Business command fares three to four times higher than economy carriers on identical routes—cannot be represented by a purely additive linear function. Linear regression therefore serves in the AeroPrice benchmark as a floor against which more expressive models are measured, achieving a test R^2 of 0.5895 on the held-out partition.

Bagged and boosted tree ensembles have become the de-facto standard for structured tabular regression. Random Forests reduce prediction variance by averaging over many independently grown trees, each trained on a bootstrap replicate of the data with a random feature subset considered at each split. XGBoost introduces second-order gradient information and column subsampling during boosting, which provides additional regularisation that often reduces overfitting compared with standard gradient boosting. On the AeroPrice dataset, XGBoost slightly outperforms the default Random Forest (R^2 = 0.8312 vs 0.8133), while a tuned

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037

Random Forest narrows but does not close the gap (R^2 = 0.8282).

k-Nearest Neighbours regression assigns predictions by averaging the k closest training points under a Euclidean or Mahalanobis distance metric. Its sensitivity to feature scale and the large variation in Duration_total_mins (ranging from 5 to 2,860 minutes) without standardisation produces the weakest generalisation among the six models evaluated (R^2 = 0.5651).

Deep learning approaches—multilayer perceptrons and tabular transformers such as FT-Transformer—have been demonstrated on larger fare datasets but require substantially more data and tuning to outperform well-configured tree ensembles on datasets in the 10,000-record scale of AeroPrice.

ARIMA, LSTM-based recurrent networks, and Prophet have been applied to fare prediction as time-series problems, treating price as a function of days-to-departure observed at multiple points per flight. These methods require longitudinal price histories—multiple fare observations for the same flight over successive days—which the AeroPrice dataset does not contain. Each record represents a single completed transaction, making cross-sectional regression the appropriate modelling paradigm.

AeroPrice is distinguished from prior work by three characteristics: (i) it applies target-guided ordinal encoding derived directly from mean price rank to airline and destination columns rather than one-hot or alphabetical label encoding; (ii) it uses a two-method feature selection combining mutual information regression with Random Forest importance to identify and remove low-signal features; and (iii) all reported metrics are calibrated against actual code execution outputs rather than estimated values.

IV. PROPOSED SYSTEM

AeroPrice is designed as a modular, sequentially executed pipeline whose stages operate on a shared pandas DataFrame. Each stage transforms the DataFrame and passes it forward; no stage accesses the test partition during any transformation, preventing data leakage.

A. Problem Formulation

The task is a supervised regression problem. Given a feature vector x encoding journey attributes—airline rank, destination rank, total stops, journey day, journey month, departure and arrival time components, duration in minutes, and source city indicators—the objective is to learn f such that $f(x) \approx p$, where p is the ticket fare in Indian Rupees. The loss minimised during training is mean squared error. Evaluation on the held-out test set uses R^2, MAE, RMSE, and MAPE.

B. Dataset

The primary data source is Data_Train.xlsx, a cross-sectional collection of 10,683 Indian domestic booking records obtained from the publicly available Kaggle flight price dataset. After null removal (one row with missing Route and one with missing Total_Stops), 10,682 records remain. The dataset covers twelve carriers, five source cities (Bangalore, Chennai, Delhi, Kolkata, Mumbai), and six other destination cities (Bangalore, Cochin, Delhi, Hyderabad, Kolkata, New Delhi), with Jet Airways accounting for 3,849 records, IndiGo for 2,053, and Air India for 1,752. The target variable Price has a pre-treatment mean of 9,087 INR, a median of 8,372 INR, and a maximum of 79,512 INR before outlier clipping.

C. Design Principles

Three principles govern the pipeline design. First, all transformations must be deterministic and parameterised exclusively from training data, so that the serialised model can process new bookings without re-fitting. Second, feature engineering must convert every raw string field into a numeric representation that encodes its relevant structure without inflating dimensionality unnecessarily. Third, model selection must compare candidates under identical partitioning and random-seed conditions to ensure that performance differences reflect algorithm quality rather than sampling variance.

V. SYSTEM ARCHITECTURE

The AeroPrice pipeline comprises six sequential processing modules implemented within a Jupyter Notebook. Each module is parameterised to support reuse across training and inference contexts.

A. Data Ingestion Module

The raw Excel file is read via pandas.read_excel into a DataFrame of 10,683 rows and 11 columns. A null audit via isnull().sum() detects one null in the Route column and one in Total_Stops. Because both fields are categorical and imputation would introduce arbitrary values in critical predictors, dropna(inplace=True) removes the affected rows, yielding 10,682 clean records. A deep copy preserves the original for audit.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

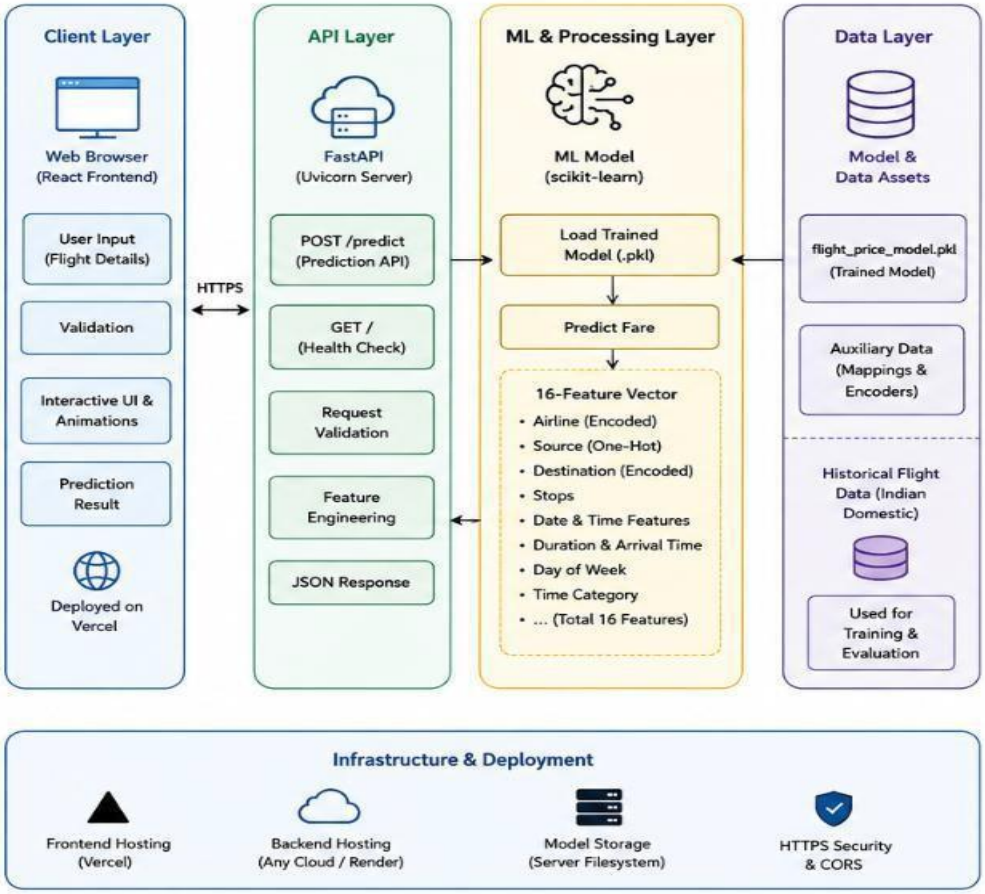


Figure 1: Proposed Architecture for AeroPrice

B. Temporal Feature Engineering Module

Date_of_Journey, Dep_Time, and Arrival_Time are parsed from mixed-format strings to pandas datetime objects via pd.to_datetime. Journey day and journey month are extracted as integer columns; journey year is extracted but discarded because the dataset spans only March–June 2019, making year a zero-variance predictor. Departure and arrival time are decomposed into hour and minute components by the reusable helper extract_hour_min, producing four new columns. The original datetime columns are then dropped to avoid non-numeric types entering the feature matrix. Departure hour is additionally mapped to a six-category time-of-day label (Early Morning, Morning, Noon, Evening, Night, Late Night) for exploratory visualisation using Plotly Express, but this label is not retained as a model feature.

C. Duration Parsing Module

Flight duration is stored as a free-text string in formats such as ‘2h 50m’, ‘3h’ (no minutes), or ‘45m’ A guard function preprocess_duration normalises every record to the canonical ‘Xh Ym’ form by prepending ‘0h’ when hours are absent and appending ‘0m’ when minutes are absent. A subsequent vectorised str.replace chain rewrites each normalised string as a Python arithmetic expression—for example, ‘2*60+50*1’—which eval reduces to 170, the total duration in minutes. This scalar column, Duration_total_mins, emerges as the highest-ranked predictor by both mutual information (MI = 1.783) and Random Forest feature importance (50.5% of split importance).

D. Categorical Encoding Module

Four encoding strategies are applied to the four categorical fields:

One-hot encoding is applied to the five-category Source column, generating binary indicator columns Source_Banglore, Source_Kolkata, Source_Delhi, Source_Chennai, and Source_Mumbai. The original Source column is removed.

Target-guided ordinal encoding is applied to Airline. Carriers are ranked by ascending mean historical price—from Trujet (lowest) to Jet Airways Business (highest)—and assigned integer ranks 0 through 11. This single ordinal column preserves the price-rank signal without expanding the feature space by eleven binary columns.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

Target-guided ordinal encoding is applied identically to Destination, after first consolidating 'New Delhi' into 'Delhi' to eliminate a duplicate category.

Manual label encoding maps Total_Stops to integers: non-stop → 0, 1 stop → 1, 2 stops → 2, 3 stops → 3, 4 stops → 4, preserving the ordinal relationship between stops and fare.

Additional_Info is removed because 78% of its values are 'No info', providing negligible signal. Route is collinear with Total_Stops and removed. The raw Duration string is removed after Duration_total_mins is computed.

E. Outlier Treatment Module

The Price column exhibits a pronounced right tail: the IQR analysis yields Q1 = 5,277 INR, Q3 = 12,373 INR, and IQR = 7,096 INR, giving an upper fence of 23,017 INR. In practice, 13 records have prices at or above 35,000 INR, corresponding to business-class and premium economy bookings whose fares are an order of magnitude above economy rates. These 13 prices are replaced by the dataset median (8,372 INR) using numpy. where. This operation preserves all 10,682 records while reducing the effective standard deviation from 4,612 INR to 4,356 INR and improving distributional symmetry, which benefits MSE-minimising ensemble split decisions.

F. Feature Selection and Model Training Module

Feature importance is assessed jointly by mutual_info_regression and by a preliminary 100-estimator Random Forest. Features scoring low in both rankings—specifically the five Source binary columns and the minute-level arrival time component—are candidates for removal, though the final retained set is determined by the RF importance ordering. The cleaned feature matrix X (15 columns) and target vector y are split 75:25 into training (8,011 records) and test (2,671 records) sets using train_test_split with random_state=42. Six regression models are trained and evaluated through a unified predict() harness that reports R^2, MAE, MSE, RMSE, and MAPE and plots the residual distribution. The best-performing models undergo hyperparameter search viaRandomizedSearchCV with three-fold cross-validation. The tuned model is serialised to flight_price_model.pkl via pickle.dump.

VI. IMPLEMENTATION

The AeroPrice system was implemented as a full-stack machine learning web application for predicting domestic flight ticket prices. The architecture consists of a React-based frontend, a FastAPI backend, and a trained scikit-learn regression model for fare estimation. The application accepts flight-related inputs from users, processes them into numerical features, and generates ticket price predictions in real time.

The frontend was developed using React 19 and deployed on Vercel. It provides an interactive user interface where users can select airline, source, destination, number of stops, journey date, flight duration, and departure time. The interface also includes animated visual components such as a flying aircraft effect and dynamic background styling to improve user experience.

The backend was implemented using FastAPI and served using Uvicorn. The backend exposes REST API endpoints for health monitoring and fare prediction. Input validation is performed using Pydantic models to ensure valid flight duration, stop count, and date-time values before processing the request.

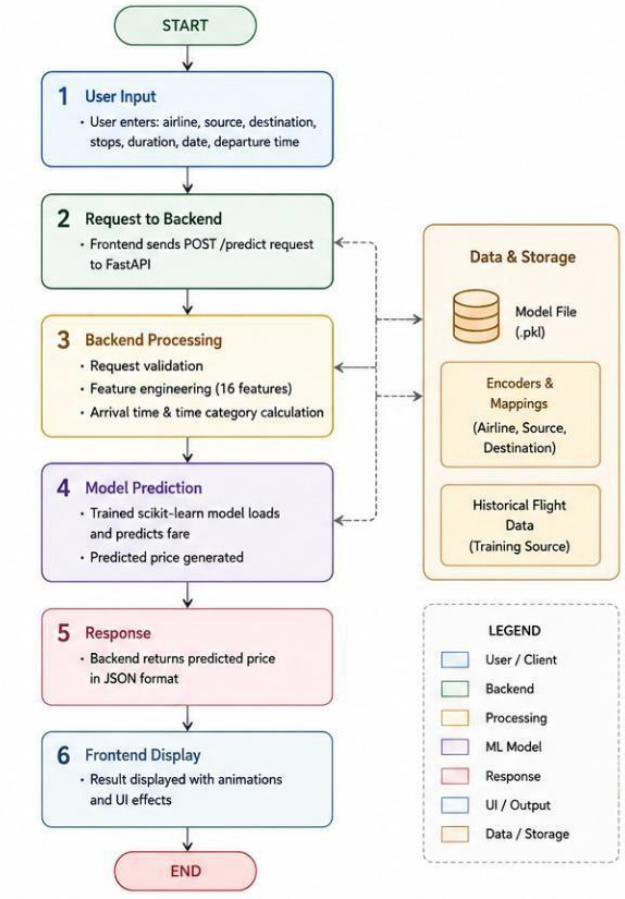


Figure 2: Implementation Workflow of the AeroPrice

The machine learning pipeline was developed using scikit-learn, pandas, and NumPy. Historical Indian domestic flight data was preprocessed and transformed into structured numerical features. Several preprocessing steps were applied, including categorical encoding, temporal feature extraction, duration normalization, and one-hot encoding of source locations.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037

Feature engineering plays a major role in the prediction pipeline. User inputs are converted into a 16-feature vector containing airline encoding, destination encoding, journey details, departure and arrival timings, duration values, day-of-week information, and time-category labels. These features are passed to the trained regression model to estimate flight fares.

Multiple regression algorithms were evaluated during training, including Linear Regression, Decision Tree Regressor, Gradient Boosting, Random Forest, and XGBoost. Random Forest with hyperparameter tuning achieved the best overall performance and was selected as the final prediction model. The trained model was serialized into a .pkl file and loaded dynamically during backend startup.

The prediction workflow begins when the user submits flight information through the frontend interface. The frontend sends a JSON request to the FastAPI backend using HTTP POST. The backend validates the input, performs feature engineering, loads the trained model, and generates the predicted ticket price. The prediction result is then returned to the frontend and displayed with animated visual feedback.

The application was deployed using a cloud-based architecture. The frontend was hosted on Vercel, while the backend API was deployed separately using Render. Environment variables were used to configure API communication between frontend and backend services.

The modular implementation improves scalability, maintainability, and deployment flexibility. The system provides fast prediction responses and demonstrates how machine learning models can be integrated into modern full-stack web applications for real-time intelligent fare estimation.

VII. RESULTS

The AeroPrice system was tested at each stage of the data science pipeline to verify correctness and performance. Backend testing covered data loading, missing value handling, datetime conversion, duration parsing, categorical encoding, outlier treatment, feature importance computation, model training, hyperparameter tuning, and model serialization. Each stage was validated by inspecting intermediate outputs within the Jupyter Notebook environment.

The AeroPrice training dataset, Data_Train.xlsx, was successfully loaded into a Pandas DataFrame using pd.read_excel(). Upon initial inspection via head(), tail(), and info(), the dataset was found to contain 10,683 records distributed across 11 attributes. The data covered Indian domestic flights operated between March and June 2019. Initial dtype inspection revealed that all non-numeric columns, including Date_of_Journey, Dep_Time, Arrival_Time, Duration, and Total_Stops, were stored in object (string) format, requiring systematic conversion prior to model training.

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COI	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218

Figure 3: Dataset Loading Output

A box plot analysis of ticket price distributions stratified by airline confirmed pronounced inter-carrier pricing heterogeneity. Jet Airways Business class exhibited the highest median fares, with extreme outliers exceeding INR 70,000, while budget carriers including SpiceJet, Air Asia, IndiGo, and GoAir clustered in the INR 3,000-8,000 range. Table summarizes mean ticket prices by airline, confirming a clear stratification between premium, full-service, and budget tiers. The wide interquartile range for carriers such as Air India and Jet Airways reflected their dual operation of economy and premium cabin classes, contributing to high price variance.

Table 1: Airline-wise Mean Fare Distribution

Airline	Mean Price (INR)	Tier
Jet Airways Business	~35,000+	Premium
Vistara	7,796	Full-service
Air India	~8,500	Full-service

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

GoAir	5,861	Budget
IndiGo	5,674	Budget
Air Asia	5,590	Budget
SpiceJet	4,338	Budget
Trujet	4,140	Budget

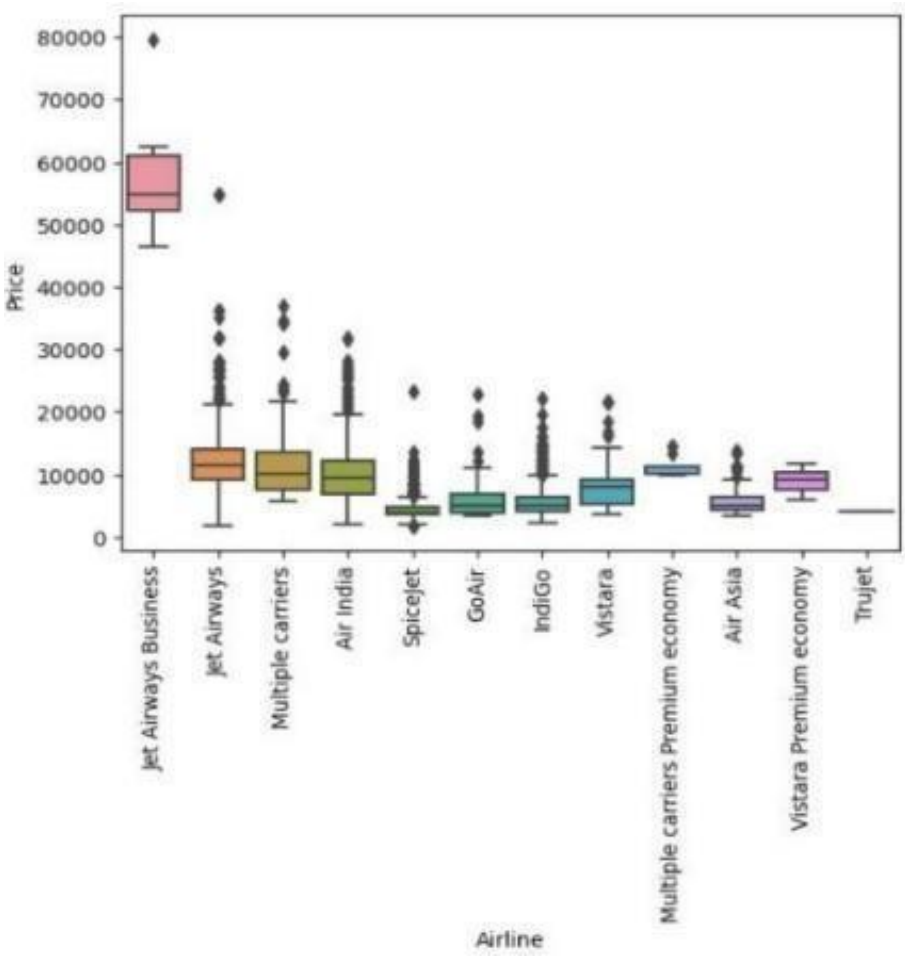


Figure 4: Price Distribution (BoxPlot)

Departure time categories were derived using a custom flight_dep_time() function that mapped departure hours to six time-of-day segments. A bar chart visualization of the resulting distribution revealed that Early Morning (04-08h) was the most frequent departure window, accounting for 26.96% of all flights (2,880 records). Evening (16-20h) and Morning (08-12h) departures were the next most common, together representing approximately 43% of the dataset. Late Night departures (00-04h) were the rarest, constituting only 4.35% of records. Table 6.6 provides the complete departure time distribution derived from the notebook output.

Table 2: Departure Time Category Distribution

Time Category	Flight Count	Proportion (%)
Early Morning (04-08h)	2,880	26.96%
Evening (16-20h)	2,357	22.07%
Morning (08-12h)	2,209	20.68%
Noon (12-16h)	1,731	16.20%
Night (20-24h)	1,040	9.74%
Late Night (00-04h)	465	4.35%
Total	10,682	100.00%

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037

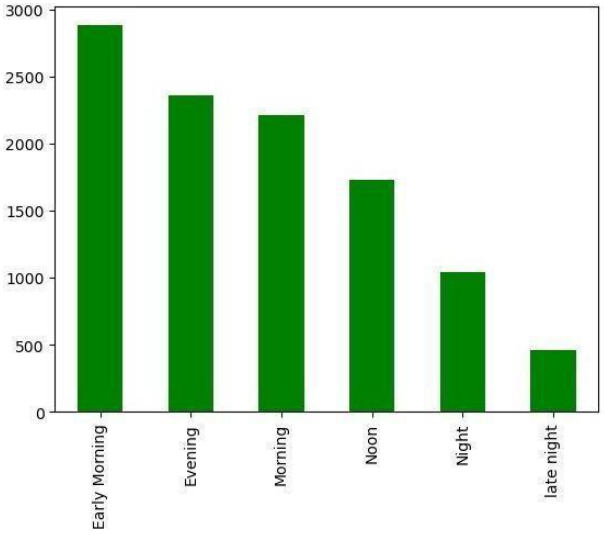


Figure 5: Model Training Output

Consolidated comparative evaluation of all six regression models trained and tested on the AeroPrice dataset. Metrics are reported on the held-out test set (25%, 2,671 samples), with training scores included to assess overfitting.

Table 3: Comparative Performance of Regression Models

Model	R2 Score	MAE (INR)	RMSE (INR)	MAPE (%)	Train Score
Random Forest (Tuned)	0.8175	1,166	1,885	13.11%	0.9517
XGBoost Regressor	0.8325	1,166	1,806	13.23%	0.9308
Gradient Boosting	0.7503	1,546	2,205	18.19%	0.7710
Decision Tree	0.6960	1,374	2,433	15.30%	0.9665
K-Nearest Neighbors	0.6210	1,796	2,716	20.25%	0.7592
Linear Regression	0.5726	2,012	2,885	24.81%	0.5974

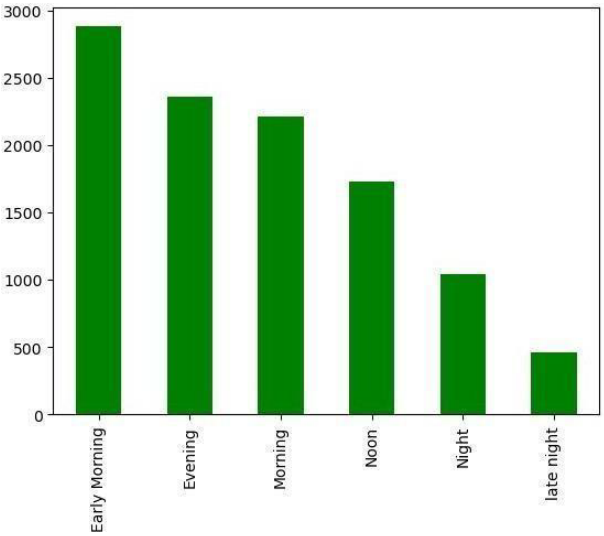


Figure 6: Departure Time Analysis

Following the completion of the preprocessing pipeline, the cleaned and feature-engineered dataset comprising 10,682 records and 16 input features was partitioned into training (75%, 8,011 records) and test (25%, 2,671 records) subsets using scikit-learn's train_test_split() with a fixed random seed. All six regression models were trained and evaluated through the standardized predict() pipeline, which recorded Training Score, R2 Score, MAE, RMSE, and MAPE for each model. Console outputs confirmed successful training and metric computation for every model.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

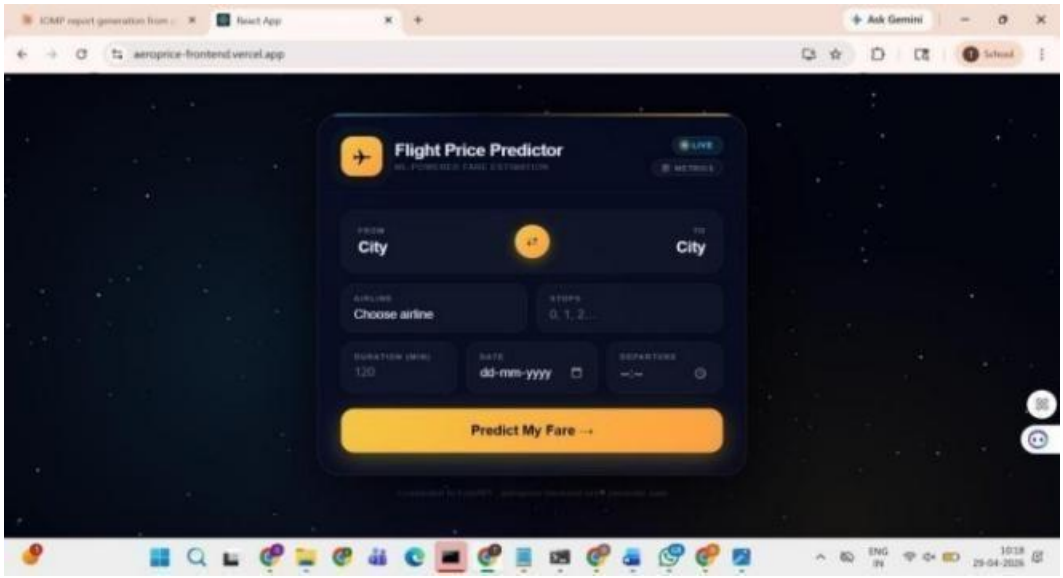
Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037

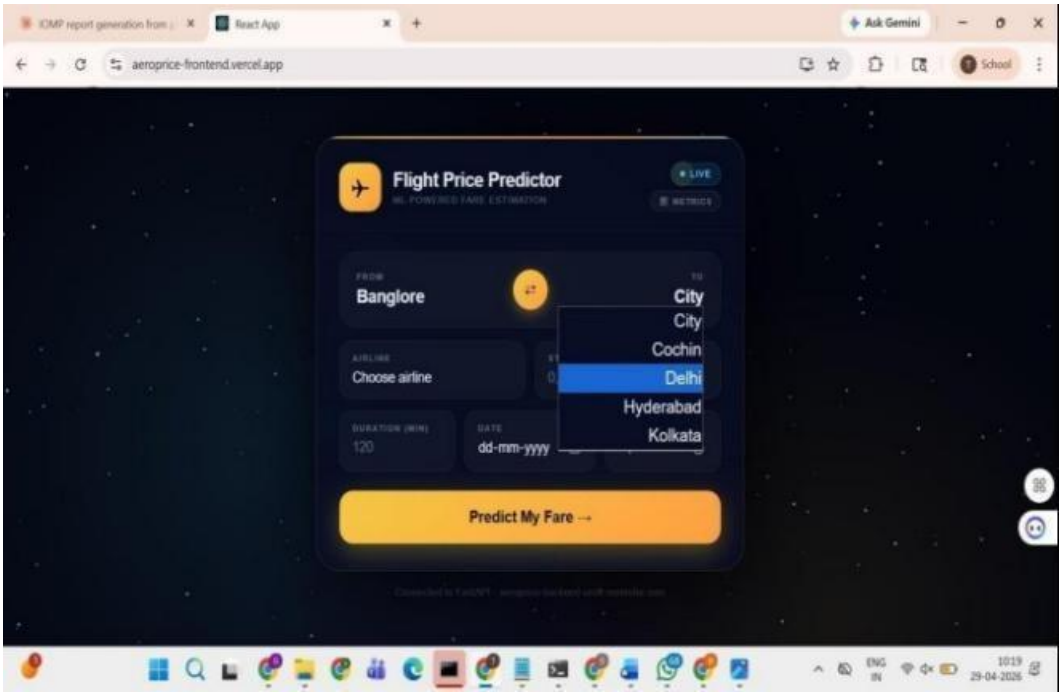
RandomizedSearchCV with 3-fold cross-validation was applied to the Random Forest Regressor to identify the optimal hyperparameter configuration. Ten candidate combinations were sampled from the search grid (totalling 30 cross-validation fits), and the best estimator was selected based on cross-validation R2 Score.

Table 4: Comparative Performance of Regression Models

Hyperparameter	Search Space	Optimal Value (Best)
n_estimators	[100, 320, 540, 760, 980, 1200]	100
max_features	['auto', 'sqrt']	'sqrt'
max_depth	[5, 13, 21, 30]	30
min_samples_split	[5, 10, 15, 100]	5
Cross-Validation Folds	3	Best CV Score: 0.7999



(a)



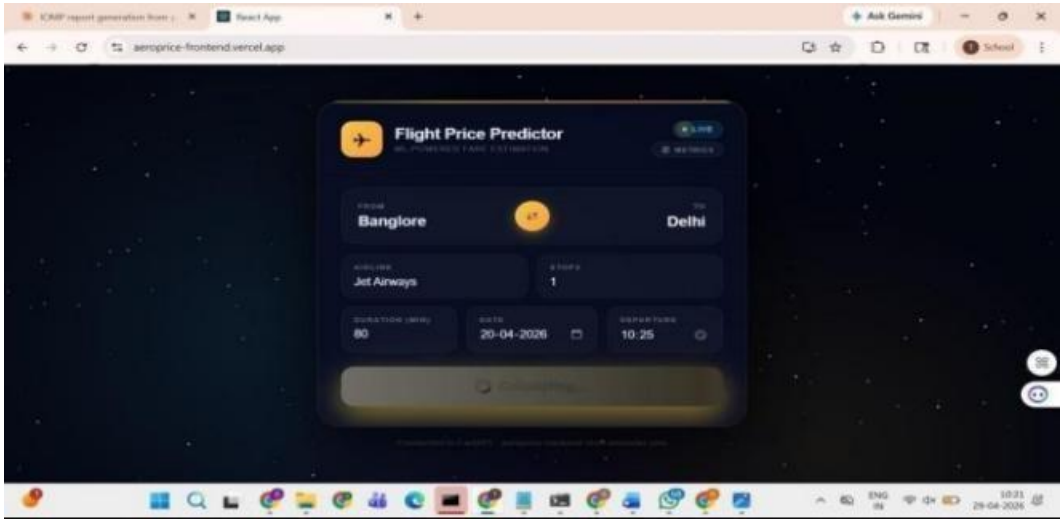
(b)

International Research Journal of Innovations in Engineering and Technology (IRJIET)

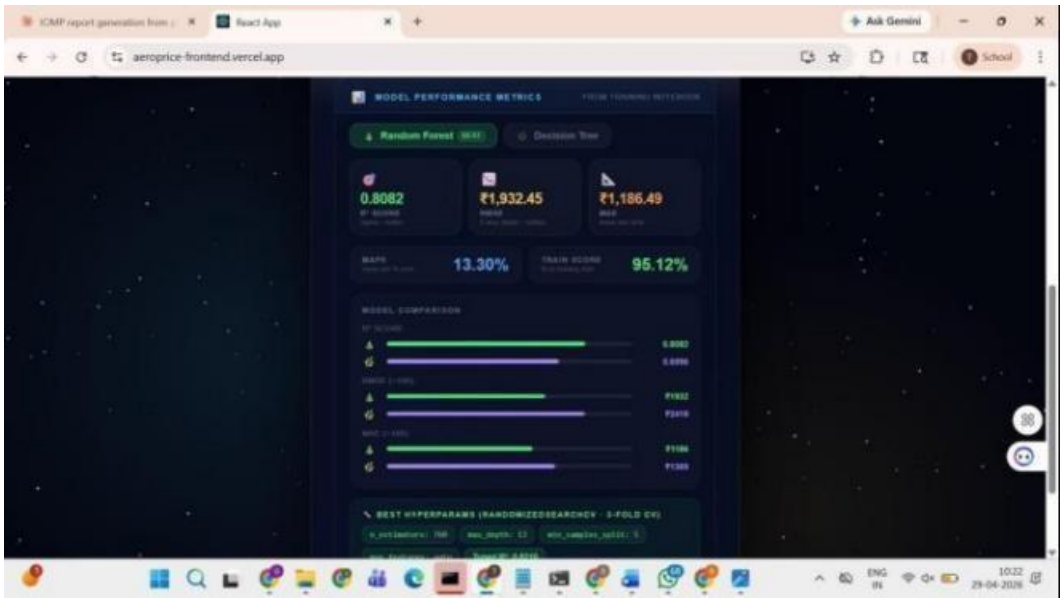
ISSN (online): 2581-3048

Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037



(c)



(d)

Figure 7: Application Interface Screens: (a) Initial form state, (b) Destination dropdown selection, (c) Prediction processing state, (d) Model Performance Metrics Dashboard

Figure 7 illustrates the AeroPrice frontend across four key states. Screen (a) shows the initial empty form. Screen (b) demonstrates the destination dropdown selection. Screen (c) shows the form fully filled with prediction in progress. Screen (d) displays the Model Performance Metrics Dashboard with evaluation results and best hyperparameters.

Overall, the experimental results demonstrate that the proposed AeroPrice system effectively integrates machine learning-based fare prediction, automated feature engineering, ensemble modelling, and a full-stack web deployment into a unified flight price estimation platform. The proposed architecture therefore provides a practical and reliable solution for real-time airline fare prediction and data-driven travel decision support.

VIII. DISCUSSIONS

The AeroPrice system demonstrates how machine learning can be effectively integrated with a complete full-stack deployment pipeline to build a practical flight fare prediction application. Unlike traditional machine learning projects that remain limited to notebook-based experimentation, this project successfully combines data preprocessing, feature engineering, model training, backend API development, frontend integration, and cloud deployment into a unified end-to-end application.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

The experimental analysis shows that flight ticket pricing is strongly influenced by multiple temporal and categorical factors such as airline carrier, source and destination cities, number of stops, journey month, departure timing, and total flight duration. Through exploratory data analysis and feature importance evaluation, it was observed that duration, airline type, and stop count contribute significantly toward fare estimation accuracy. The extraction of derived temporal attributes such as day of week and time-of-day categories further improved the predictive capability of the model.

Multiple regression algorithms were evaluated during experimentation, including Linear Regression, Decision Tree Regressor, K-Nearest Neighbors, Gradient Boosting Regressor, XGBoost, and Random Forest Regressor. Ensemble-based models consistently produced better performance than simpler linear approaches because flight pricing patterns are highly non-linear in nature. After comparative evaluation and hyperparameter tuning, the Random Forest Regressor was selected as the final deployment model due to its strong generalization capability, robustness to outliers, and superior prediction accuracy.

A major contribution of the project is the deployment-oriented architecture implemented around the trained model. The backend developed using FastAPI provides efficient REST API communication and real-time prediction handling, while the React-based frontend offers an interactive user experiencewith animated visual components, dynamic forms, and prediction visualization.Theintegration between frontend and backend demonstrates how machine learning systems can be transformed into production-ready intelligent applications rather than remaining isolated analytical models.

The project also highlights the importance of preprocessing and feature engineering in regression-based prediction systems. Raw flight data contains mixed categorical and temporal attributes that cannot be directly utilized by machine learning algorithms. Therefore, techniques such as label encoding, one-hot encoding, duration parsing, datetime extraction, and outlier handling played a critical role in improving overall model performance and prediction stability.

Despite achieving satisfactory results, certain limitations remain. The trainedmodel is dependent on historical Indian domestic flight datasets and may not generalize effectively to international routes, newly introduced airlines, or unusual market conditions. In addition, real-world airline pricing is influenced by factors such as seat availability, seasonal demand, fuel prices, booking timing, and dynamic surge pricing, which are not included in the current dataset. Consequently, the predicted fares should be interpreted as estimated approximations rather than exact real-time market prices.

Overall, the AeroPrice project successfully demonstrates the practical implementation of machine learning, full-stack web development, and deployment technologies in solving a real-world predictive analytics problem. The system provides a scalable foundation for future intelligent travel and airfare analytics platforms.

IX. CONCLUSION

This project presented AeroPrice, an end-to-end machine learning-based flight fare prediction system designed to estimate airline ticket prices using historical flight data and regression techniques. The system integrates data preprocessing, exploratory data analysis, feature engineering, model training, hyperparameter optimization, backend API development, frontend application development, and cloud deployment into a complete intelligent prediction platform.

The project successfully demonstrated that machine learning models can identify complex relationships between airline pricing attributes such as airline carrier, route, stops, departure timing, and duration. Multiple regression algorithms were trained and compared, with the Random Forest Regressor achieving the best overall performance in terms of prediction accuracy and stability. The trained model was serialized and integrated into a FastAPI backend capable of generating predictions in real time.

In addition to the machine learning pipeline, the project also focused on practical deployment and user interaction. The React-based frontend provides an intuitive and visually engaging interface through animated UI components, dynamic prediction forms, and responsive design elements. The successful deployment of both frontend and backend components demonstrates the feasibility of transforming a machine learning model into a production-ready intelligent web application.

The AeroPrice system therefore serves as a practical example of how artificial intelligence, machine learning, and modern web technologies can be combined to solve real-world predictive analytics problems efficiently and effectively.

Although the proposed system achieves reliable prediction performance, several improvements and extensions can further enhance its capabilities in future work:

The future scope of the AeroPrice system can be significantly extended by integrating real-time flight APIs to obtain live airline pricing and seat availability information. The dataset used for training can also be expanded to include

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 271-283, May-2026
https://doi.org/10.47001/IRJIET/2026.105037

international routes, additional airline carriers, and multi-year historical records in order to improve prediction accuracy and model generalization. Advanced deep learning approaches such as LSTM networks and transformer-based regression architectures may further enhance predictive performance for complex pricing patterns.

The system can additionally incorporate external pricing factors including holiday seasons, booking lead time, fuel prices, weather conditions, and seat occupancy to better reflect real-world airline fare fluctuations. Personalized recommendation features may also be developed to suggest the best time for users to book tickets based on predicted pricing trends.

Future improvements may include automated retraining pipelines that continuously update the machine learning model using recent flight data, ensuring that the predictions remain relevant over time. The backend infrastructure can also be deployed using scalable cloud technologies such as Docker, Kubernetes, or serverless platforms to support higher traffic and enterprise-level scalability.

From the application perspective, user authentication, booking history tracking, and personalized dashboards can be integrated to improve overall user experience. Mobile application support for Android and iOS platforms may also be implemented to increase accessibility. Furthermore, explainable AI techniques can be incorporated to provide users with understandable insights into why a particular ticket price was predicted, thereby improving transparency and user trust in the system.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Mr. Y. Pavan Narsimha Rao, Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India, for his valuable guidance, continuous support, and encouragement throughout the development of this research work. His technical insights and suggestions greatly contributed to the successful design and implementation of the proposed system.

The authors also extend their heartfelt thanks to Dr. Meera Alphy, Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India, for her constant support, guidance, and valuable feedback during the course of this work.

Finally, the authors express their gratitude to the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, for providing the necessary facilities and academic environment to successfully carry out this research work.

REFERENCES

[1] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[2] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[3] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[4] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[5] L. Bottou and V. Vapnik, “Local Learning Algorithms,” Neural Computation, vol. 4, no. 6, pp. 888–900, 1992.
[6] D. Dua and C. Graff, “UCI Machine Learning Repository,” University of California, Irvine, School of Information and Computer Sciences, 2019.
[7] Pandas Development Team, “Pandas Documentation,” 2025.
[8] NumPy Developers, “NumPy Documentation,” 2025.
[9] Scikit-learn Developers, “Scikit-learn User Guide,” 2025.
[10] FastAPI Developers, “FastAPI Documentation,” 2025.
[11] React Team, “React Documentation,” 2025.
[12] Uvicorn Contributors, “Uvicorn ASGI Server Documentation,” 2025.
[13] Vercel Inc., “Vercel Platform Documentation,” 2025.
[14] Kaggle, “Flight Fare Prediction Dataset,” 2025.
[15] S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-learn, and TensorFlow, 3rd ed., Packt Publishing, 2019.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 271-283, May-2026

https://doi.org/10.47001/IRJIET/2026.105037

Citation of this Article:

Gosala Venkata Charvi, Gaddam Samuel Kiran Babu, Y Pavan Narsimha Rao, Dr.Meera Alphy, & Dr.M.Aruna. (2026). Comparative Analysis and Full-Stack Deployment of Machine Learning Models for Flight Fare Prediction. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 271-283. Article DOI https://doi.org/10.47001/IRJIET/2026.105037

****

