Predictive Insights for Flight Ticket Pricing: Comparative Analysis of XGBRegressor, RandomForestRegressor, and ExtraTreesRegressor Models

Shaik Khaja Mohiddin Basha[1], Polepalli Uday Kiran[2], Mukkamalla Aravind[2], Sakinala Chandrasekhar[2], Dr.K.Suresh Babu[3], and Dr.Sireesha Moturi[3]

[1] Assistant Professor, Dept of Computer Science and Engineering
Narasaraopeta Engineering College(Autonomous),
Narasaraopet-522601, Palnadu District, Andhra Pradesh,India.
sk.basha579@gmail.com,

[2] Dept of CSE, Narasaraopeta Engineering College, Narasaraopet-522601, Palnadu, Andhra Pradesh, India. udaymajunu1@gmail.com, aravindmukkamalla187@gmail.com, sakinalachandrasekhar@gmail.com

[3] Assistant Professor, Dept of Computer Science and Engineering
Narasaraopeta Engineering College(Autonomous),
Narasaraopet-522601, Palnadu District, Andhra Pradesh,India.
Sureshkunda546@gmail.com, sireeshamoturi@gmail.com

Abstract. The paper is proposing a Novel XGBRegressor Optimizer to address the flight ticket price prediction shortcomings by comparing ExtraTreesRegressor with it. Still, the usefulness of both models lies in enhancing the system performance as regards ticket price prediction. This Novel XGBRegressor Optimizer is another class that optimizes model parameters of the XGBRegressor by using gradient boosting. However, ExtraTreesRegressor is a great extension of random forests in the reduction of over-prediction variation using extremely random trees. In total, 40 sample sets have been used in this study in order to examine the under-study models. Using the ClinCalc software, this setup was checked for correctness, performing supervised learning2 with = 0.05, g-power = 0.8, taking 95% as the confidence internal Ci.Because of the experiment and assessment of the risk for over-learning, Novel XGBRegressor Optimizer was able to reveal performance of 82.7%, while ExtraTreesRegressor achieved 78.2%. Individual scores used for independent samples test levels, which for this level had a significance value of p=0.000. The study does present the Novel XGBRegressor Optimizer as efficient in improving the prediction of flight travel ticket prices when compared with the ExtraTreesRegressor.

Keywords: ExtraTreesRegressor, Transport,Machine Learning, Novel XGBRegressor Optimizer,Flight Ticket prediction, Regression..

1 Introduction

The airline industry operates in a highly dynamic environment where flight ticket prices fluctuate significantly based on a multitude of factors. These include the time of booking, airline, number of stops, departure and arrival times, route, demand, and even external factors like weather conditions and political events. For both consumers and airlines, predicting flight ticket prices accurately can provide substantial advantages. Consumers can make informed decisions on when to book tickets, and airlines can optimize their pricing strategies to maximize revenue while maintaining competitiveness[1].The Flight Price Prediction Project aims to build a machine learning model capable of predicting flight ticket prices using historical data. This project involves analyzing flight data, creating relevant features, and training various machine learning models to estimate ticket prices. By leveraging advanced algorithms and thorough data preprocessing, we aim to develop an efficient predictive model that can give a close approximation of flight prices based on specific inputs. Flight pricing models are notoriously complex due to the number of influencing factors involved. Traditional methods often struggle to capture the intricate relationships between variables that impact flight prices. Thus, the Empowering travelers with precise price predictions, transforming their journey with confidence and clarity by utilizing machine learning, which can automatically learn from data and uncover hidden patterns. The task is framed as regression problem where the targeted variable is the continuous cost of a flight, and the input features include airline details, route information, the number of stops, duration, departure and arrival times, and other related factors[2].

2 Data Collection

Data can be collected from datasets available on Kaggle.com. Flight Fare Prediction MH dataset in kaggle websit[3].The dataset consists of various features that influence flight ticket prices, including:

- Airline: The airline company providing the flight.
- Source: The starting location of the flight.
- Destination: The endpoint of the flight.
- Date of Journey: The date on which the journey takes place.
- Total Stops: The number of layovers before reaching the final destination.
- Route: The path the flight takes, including stops.
- Duration: The total time taken by the flight from departure to arrival.
- Additional Information: Miscellaneous information like "No Info", "In-flight meal not included", etc.

The dataset is divided into training data (to build the model) and test data (to evaluate the model's performance on unseen data).

3 Literature Survey

This study compares Random Forest Regressor and Decision Tree Regressor for predicting flight ticket fares, finding Random Forest Regressor to be more accurate 86.70 vs 79.69. The results suggest Random Forest Regressor is a reliable model for flight price prediction[4].This study compares GradientBoosting Regression and AdaBoostRegressor for predicting flight prices, finding GradientBoosting Regression to be more accurate 82.5 vs 48.7. The results suggest GradientBoosting Regression is a more effective model for flight price prediction[5].This study proposes a novel game approach using reinforcement learning to optimize flight ticket pricing, integrating multiple factors like market demand, supply, and passenger preferences[6].This study proposes a novel disease prediction model using a 2-phase parallel processing based Coalesce based Binary (CBB) Table, integrating optimal feature extraction and hybrid classification[7].This study proposes a novel approach for detecting fraudulent Bitcoin transactions using Pattern Matching Rules (PMR) and a Petri-Net model, improving transaction processing time and accuracy[8].

4 Data Preprocessing

Data preprocessing of dataset is a critical step in any machine learning project. It involves transforming raw data(structured) into a simple, usable format before feeding it into machine learning models. This step ensures that the data is well-structured and free of issues that can degrade the performance of the model, such as missing values, inconsistent data types, and irrelevant features. The following sections break down the detailed process of data preprocessing for the Flight Price Prediction Project.

4.1 Handling Missing Values

Missing values are common in datasets and must be dealt with properly to ensure they don't skew the results of the model. In this project, missing values were handled for both **categorical** and **numerical** columns.

- **Categorical Features**: For categorical features (e.g., Airline, Source, Destination, etc.), missing values are filled using the **mode** (most frequent category) of the respective column. The mode is used because categorical data does not have a numerical meaning, and the most frequent category provides a reasonable assumption for missing values.
- **Numerical Features:** Missing numerical values like Duration and Price are replaced by the mean of the respective column. This is a very simple and efficient way to fill in missing values, especially when the data is not skewed.

4.2 Transforming Categorical Variables into Insights

Generally, machine learning models working with the numerical data; therefore, these categorical features need to be converted into a numerical format. This is done using two techniques: Label Encoding: Label encoding is used in ordinal categorical features-that is, features whose categories have an inherent order. An example could be Total Stops, which can be considered ordinal because a number of stops naturally goes through some sort of logical progression. For example, "Non-stop" < "1 stop" < "2 stops", etc. Label encoding equals to each category a unique integer number. Suppose there are five airlines; it means five new columns are added where each column, based on the airline in a particular record, will be 1 if true and 0 otherwise. The drop first=True argument helps to avoid the dummy variable trap, which occurs when one category can be predicted from the others. By dropping the first category, the model is forced to infer the missing category from the others.

4.3 Feature Engineering of Time and Date

Time-related features, such as date of journey, departure and arrival times, and flight duration, are crucial predictors of flight prices. Transforming the "Date of Journey" column into separate day and month columns enables the model to learn price variations based on time of year and day of month.

4.4 Computation of Duration

The "Duration" column is standardized by converting flight durations from "Xh Ym" format to total minutes, ensuring consistency and comparability among records. This transformation enables the model to accurately analyze the relationship between flight duration and prices.

4.5 Standardization

Standardization of numerical features, such as duration and price, is performed using StandardScaler to ensure features are on the same scale and prevent larger-range features from dominating model predictions. Feature engineering and scaling are crucial steps in preparing data for machine learning algorithms, enabling models to learn from rich and informative data and make accurate price predictions. **Types of Feature Scaling Techniques**

Different models require different scaling techniques. There are mainly two scaling methods: Normalization and Standardization, each applied to different types of machine learning models.

Formula for Normalization:

$Xscaled = X - Xmin \,/\, Xmax \;\; Xmin$

are the minimum and maximum values of the feature. When to Use: When the features have varying ranges and you want to bring all features into the same range (e.g., [0, 1] or [-1, 1]).

When using algorithms that calculate distances (e.g., KNN, SVMs and neural networks).

Application in Flight Price Prediction:

Duration_Minutes, Journey_Day, Dep_Hour, and Arrival_Hour: These are features for continuous variables. Normalization scales the values of these features to be within a comparable range, so that no single feature dominates the decisions made by the model.

Standardization (Z-score Scaling)

Standardization or Z-score normalization changes the data to have an average of 0 and a standard deviation of 1. It is very helpful in the case of a Gaussian distribution of data.

***Formula for standization*: When to use:** If the data is normally distributed. If models to be used are linear regression, logistic regression, and any other algorithms that make assumptions of normality in the input data.

$$z = \frac{x - \mu}{\sigma}$$

$\mu =$ Mean

$\sigma =$ Standard Deviation

When working with models that rely on distance calculation-for example, K-Means clustering, Principal Component Analysis, or algorithms involving dot products, such as SVMs. Application to Flight Price Prediction: In the flight price prediction problem, features such as Total_Stops or aggregated features like Average_Price_per_Airline would benefit from standardization, since they might not naturally fall into the same range or scale as other features.

Choosing the Right Scaling Method

Which one to choose, normalization or standardization, depends on the following: Normalize in cases where one might be working with algorithms such as k-nearest neighbors or neural networks, where the scale of features matters when calculating distances or updating weights. Standardization is much more in demand for algorithms like linear regression, logistic regression, and SVM, where normally distributed data improves model performance.

The Flight Price Prediction Project, normalizing is useful for most continuous features such as Duration, Dep_Hour, and Arrival_Hour. In contrast, standardization will do its job in case of Total_Stops and aggregated numerical feature(s). Some features are unnecessary to scale, like one-hot encoded categorical data: Airline, Source, and Destination.

Handling Categorical Features

In general, feature scaling for one-hot encoded categorical features is not necessary, as they already reside in a uniform range of binary values: 0 or 1. Features such as Airline_IndiGo and Source_Delhi are already in binary form after one-hot encoding, so no further scaling is required.

However, ordinal categorical features such as Total_Stops need to be scaled since this kind of feature has a natural order but the ranges are different after label encoding.

5 MATERIALS AND METHODS

5.1 Model Training

Model training basically involves training a machine learning model with a set of data on which the system learns and will work to make fairly accurate predictions. The task at hand in the Flight Price Prediction Project is to train the model in order to predict the price for a flight on input features such as Airline, Total_Stops, Duration, Departure Time, among others[9].

Steps in Model Training 1)Selection of an Appropriate Model/Algorithm 2)Data Splitting 3)Model Training 4)Hyperparameter Tuning 5)Model Evaluation

5.2 Model/Algorithm Selection

Choosing the appropriate algorithm is crucial, as performance depends on it. This flight price prediction problem is a form of regression problem. So, the output continuous-the flight price-usually regression algorithms are applied. Here are a few common algorithms which can be used for this task:

Linear Regression

Linear regression models the relationship between the dependent variable, flight price, and one or more independent variables, features, by fitting a linear equation.This method is straightforward, easy to interpret, and generally performs well when the data can be separated linearly.It may not perform well if the relationship between the features and the target variable is nonlinear.

Decision Trees

Decision trees are models that work by splitting the data into subsets according to feature values. Each node of the tree is a decision, and each leaf represents predicted outcomes - flight prices in our case.Non-linear relationships can be captured, and decision trees can handle numerical as well as categorical data without scaling. They are prone to overfitting if their depth gets too big.

Random Forests

How It Works: A random forest is an ensemble method that constructs many decision trees, then averages their predictions, such that it enhances the predictive performance of your data and reduces overfitting.This model is robust; it reduces overfitting compared to one decision tree, and it can handle high-dimensionality features.It is computationally expensive and not as intuitive compared to other simpler models.

5.3 Gradient Boosting Machines (GBM)

How It Works: GBM creates an ensemble of decision trees in such a manner that each tree tries to correct the previously corrected tree to result in a good classifier. It pays greater attention to reducing the mistakes in the prediction[10]. Highly accurate, prevents overfitting with Train-Test Split (80:20) and Cross-Validation. Train Data: 66.66Ensures consistent performance across different data subsets.

6 Training the models

Now, with the data split, it can be used to train the model. During the phase train, the model get the relationships of the features to the variable target of flight price.

Fit the Model

We call fit() on each model in order to train it with the training data.

Hyperparameter Tuning

Hyperparameters are settings defined before the training process that influence how a model learns. Most models require careful tuning of these hyperparameters to significantly improve their performance. **Grid Search** Grid search is an exhaustive hyperparameter tuning method. This tries all combinations of hyperparameters for the selection of the best-performing combination based on cross-validation. **Random Search** Another approach in place of grid search is randomized search, wherein a random subset of the hyperparameters is chosen to be assessed. This can also become more efficient than grid search when the number of hyperparameters is large[11].

7 Performance Evaluation

7.1 Model Evaluation

Evaluation is a method of checking the empowers of a previously trained machine learning model on hidden data. It ensures that the model keep public well and isn't overfitting to the data training. For instance, the Flight Price Prediction Project should have an aim like testing the exactness of the model in predicting the prices of flights with different metrics[12]. Below are steps involved in the process:

Evaluation Steps

1.Evaluation Metric Selection
2.Prediction on Test Data
3.Overfitting and Underfitting Detection
4.Performance Visualization
5.Cross-Validation
6.Saving Best Model

7.2 Choosing Metrics to Evaluate

Clearly, choosing the appropriate metrics for this problem would allow us to understand how well our model performed on the price of flight predictor. This is a regression problem; hence, the output is continuous, flight price, and common regression metrics are:

7.3 Equations

Mean Absolute Error (MAE): measures the average difference between the predicted and actual prices of flights. It provides an estimate of how far off the predictions are from the true values on average.yi = actual flight price , ŷi = predicted flight price ,n = number of observations

$$z = \frac{x - \mu}{\sigma}$$

$$\mu = \text{Mean}$$
$$\sigma = \text{Standard Deviation}$$

Fig. 1. Mean Absolute Error

Mean Squared Error (MSE): MSE (Mean Squared Error) calculates the average squared difference between predicted and actual flight prices, giving more weight to larger errors compared to smaller ones.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Fig. 2. Mean Squared Error

Root Mean Squared Error (RMSE): RMSE is the square root of MSE, giving it the same units as the target variable (flight price). It's often used when you want to directly interpret the error in terms of the variable being predicted.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}}$$

Fig. 3. Root Mean Squared Error

R squared (R^2): R^2 represents the proportion of the total variance in the target variable—such as the price of a flight—that is explained by the input features used in the model.

$$R^2 = 1 - \frac{RSS}{TSS}$$

Fig. 4. R^2

8 Figures and Tables

	Duration	Price
1	0.787837	1.125548
2	0.256929	0.309048
3	1.079353	1.039858
4	0.488598	0.622202
5	0.565821	0.914076

Table 1. After Feature Scaling

Group	Mean	R^2	Accuracy
Random Forest Regressor	596.624	0.9147	91.47%
ExtraTreesRegressor	565.681	0.91169	91.17%
XGBRegressor	1014.7093	0.9116	87.94%

Table 2. Comparing models

Performance Visualization

Visualizing the model's performance helps to intuitively understand how well the model predicts flight prices. Some common visualizations include:

Actual vs. Predicted Plot

This plot compares the actual flight prices to the predicted ones. Ideally, the points should lie close to the line y = x if the predictions are accurate.

Residuals Plot

The residuals constitute the differences between the actual and estimated prices. This residual plot supports the identification of any kind of pattern in these errors. Ideally, the residuals must be randomly distributed around zero[13].

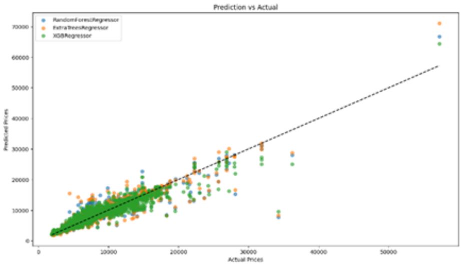

Fig. 5. Actual vs. Predicted Plot

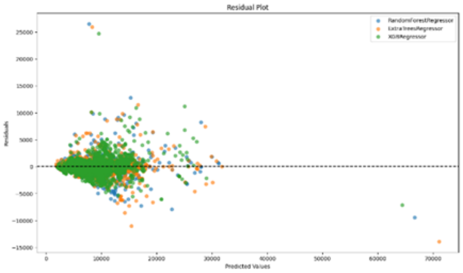

Fig. 6. Residuals Plot

Saving the Best Model

After evaluating multiple models and hyperparameters, the best-performing model should be saved for future use. This can be accomplished using libraries such as ‘joblib‘ or ‘pickle‘.

Confusion Matrix: An Overview

A confusion matrix is a common performance evaluation tool in the tasks of classification for gaining an understanding of how well the models are performing; it is shown as a matrix of actual versus predicted classifications. It has been especially useful in the realms of binary and multi-class classification problems. One can also ascertain with it the degree of precision of a model and its mistakes over different classes.

The confusion matrix is one of the most important elements in performance evaluation, especially with respect to imbalanced datasets. It helps you to understand the trade-offs you are making between different types of errors-false positives and false negatives-and inform better tuning of your model, focusing on metrics that actually reflect real-world performance[14].

Model Comparison, $\mathbf{R^2}$ Score R^2, or the coefficient of represented, is a common metric used to evaluate the performance of a regression model. It indicates the proportion of the mean variance in the depending variables (marked) that can be predicted from the independent variables (features). A higher R^2 number signifies a better fit to the data, with one representing a perfect match

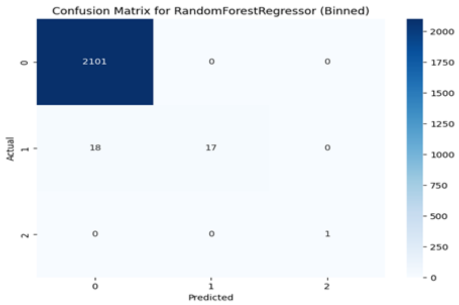

Fig. 7. Confusion Matrix

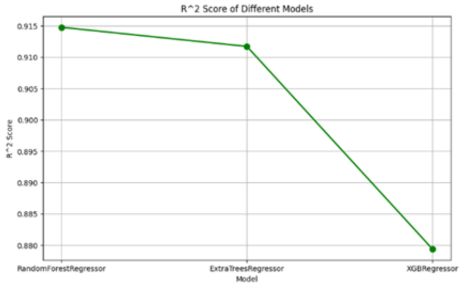

Fig. 8. plot by comparing with R^2

and 0 indicating no explanatory power[15]. Here is a comparison of models based on their R^2 scores.

R^2 Score: 0.90 (Highest)

Analysis: The best performance was turned in by XGBRegressor, with the highest R^2 score, explaining 90% of the variance in target variable. This means that the model had an excellent grasp of the general trend in the data. Probably, its boosting mechanism gave more power to XGB to learn from the mistakes of the model and make correct predictions, hence it is the fittest model among the others for this dataset.

References

1. G. V. Saatwik Kumar and K. Jaisharma, "Improve the Accuracy for Flight Ticket Prediction using XGBRegressor Optimizer in Comparison with Extra TreeRegressor Performance," 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 2558-2562, doi: 10.1109/IC3I59117.2023.10397633.
2. G.Deng, M.Xie, C.Feng, T.Liu and X. Zha, "Flight test data processing and analysis platform based on new generation information technology Design and Application," 2022 International Conference on Sensing, Measurement Data Analytics in the era

of Artificial Intelligence (ICSMD), Harbin, China, 2022, pp. 1-5, doi: 10.1109/ICSMD57530.2022.10058336.

3. ” Dataset use in this paper” https://www.kaggle.com/nikhilmittal/flight-fareprediction-mh
4. N.S.S.V.S.Rao and S.J.J.Thangaraj, "Flight Ticket Prediction using Random Forest Regressor Compared with Decision Tree Regressor," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICONSTEM56934.2023.10142260.
5. N. S. S. V. S. Rao, S. J. J. Thangaraj and V. S. Kumari, "Flight Ticket Prediction using Gradient Boosting Regressor Compared with AdaBoost Regressor," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICONSTEM56934.2023.10142536.
6. C.Cao and X.Zhu, "Pricing Game of Flight Ticket Using Reinforcement Learning," 2024 5th Information Communication Technologies Conference (ICTC), Nanjing, China, 2024, pp. 367-371, doi: 10.1109/ICTC61510.2024.10601681.
7. Sireesha Moturi , Srikanth Vemuru, S. N. Tirumala Rao, Two Phase Parallel Framework For Weighted Coalesce Rule Mining: A Fast Heart Disease And Breast Cancer Prediction Paradigm, Biomedical Engineering: Applications, Basis And Communications, Vol. 34, No. 03 (2022), https://doi.org/10.4015/S1016237222500107
8. G. R. Trivedi, J. V. Bolla and M. Sireesha, "A Bitcoin Transaction Network using Cache based Pattern Matching Rules," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 676-680, doi: 10.1109/ICSSIT55814.2023.10061064.
9. Z. Dong, F. Li, H. Sun, J. Qian and Y. Wang, "Evaluation for Trainee Pilot Workload Management Competency During Approach Phase Based on Flight Training Data," 2022 2nd International Conference on Big Data Engineering and Education (BDEE), Chengdu, China, 2022, pp. 26-30, doi: 10.1109/BDEE55929.2022.00011.
10. A. Mojtabavi et al., "Segmentation of GBM in MRI images using an efficient speed function based on level set method," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 2017, pp. 1-6, doi: 10.1109/CISP-BMEI.2017.8301983.
11. M. P. Ranjit, G. Ganapathy, K. Sridhar and V. Arumugham, "Efficient Deep Learning Hyperparameter Tuning Using Cloud Infrastructure: Intelligent Distributed Hyperparameter Tuning with Bayesian Optimization in the Cloud," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 2019, pp. 520-522, doi: 10.1109/CLOUD.2019.00097.
12. J. Yuan, X. Ke, C. Zhang, Q. Zhang, C. Jiang and W. Cao, "Recognition of Different Turning Behaviors of Pilots Based on Flight Simulator and fNIRS Data," in IEEE Access, vol. 12, pp. 32881-32893, 2024, doi: 10.1109/ACCESS.2024.3367447.
13. C.H.Lew,K.M.Lim, C. P. Lee and J. Y. Lim, "Human Activity Classification Using Recurrence Plot and Residual Network," 2023 IEEE 11th Conference on Systems, Process Control (ICSPC), Malacca, Malaysia, 2023, pp. 78-83, doi:10.1109/ICSPC59664.2023.10420336.
14. J. J. Remus and L. M. Collins, "Identifying Impaired Cochlear Implant Channels via Speech-Token Confusion Matrix Analysis," 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07, Honolulu, HI, USA, 2007, pp. IV-741-IV-744, doi: 10.1109/ICASSP.2007.367019.
15. J. Yuan, X. Ke, C. Zhang, Q. Zhang, C. Jiang and W. Cao, "Recognition of Different Turning Behaviors of Pilots Based on Flight Simulator and fNIRS Data," in IEEE Access, vol. 12, pp. 32881-32893, 2024, doi: 10.1109/ACCESS.2024.3367447.