**FLY FEST**

**Economical travel with Cultural Highlights**

Suman Mahato

Computing Technology

SRM Institute of Science and Technology

Kattankulathur, Chennai

ss5292@srmist.edu.in

Dr T Ragunthar

Dhruv Saini

Computing Technology

SRM Institute of Science and Technology

Kattankulathur, Chennai

dy7354@srmist.edu.in

Dr. Jane Olive Sharon

Computing Technology

SRM Institute of Science and Technology

Kattankulathur, Chennai

janeolip@srmist.edu.in

*Abstract*— Nowadays in our budget conscious world, travelers search for the most economical flight options while exploring new experiences. "Fly Fest: Economical Travel with Cultural Highlights," aims to create an integrated system that combines flight fare data with events and inform various experiences to recommend the most affordable travel opportunities. By analyzing flight data and experiences data, the system preprocesses and normalizes data for accurate analysis. A machine learning model, specifically a Random Forest Regressor, is trained on flight data to predict the lowest fare trends. In addition, a recommendation engine powered by collaborative filtering suggests travel destinations aligned with users' preferences and cultural interests. The system would update recommendations based on new fare and event data, offering dynamic, travel suggestions. As a real-life example, a user interested in doing water sports receives suggestions for the cheapest travel places from their location, ensuring that their trip is both culturally enriching and affordable. The system's user-friendly interface visualizes fare trends and event details, allowing users to make informed decisions. This solution caters to budget travelers, enhancing their ability to experience cultures and activities without burning a hole in their pockets.

Keywords— budget travel, flight fare prediction, machine learning, Random Forest, event recommendation

# **Introduction**

In today’s highly competitive aviation industry, airfare pricing is dynamic and influenced by a variety of factors such as demand, supply, seasonality, and special events. Budget conscious traveller’s look for various ways to relax and reduce their travel expenses, predicting flight prices becomes a crucial tool for both consumers and travel agencies. The growing importance of dynamic pricing in airfare, alongside the availability of large volumes of flight data, has led to the application of machine learning models to predict airfare trends and recommend optimal booking fares.

This project focuses on the prediction of flight fares while integrating events, festival, and activity recommendations to enhance the traveler experience. Unlike traditional and normal booking systems, this approach not only predicts the best time to purchase a ticket but also suggests travel by correlating fare with significant events and cultural activities at the destination. The domain of this project falls under data

science and machine learning applications in the travel industry.

In previous works, various machine learning techniques have been used for airfare prediction. Tziridis et al. [1] demonstrated the use of Support Vector Machines (SVMs) for predicting airfare prices. Etzioni et al. [2] focused on mining airfare data using decision tree-based approaches to minimize ticket purchase costs. Similarly, Panigrahi et al. [3] employed regression models to analyze and forecast flight fares. Meanwhile, Knijnenburg and Willemsen [4] explored recommender systems for enhancing the user experience by suggesting personalized travel recommendations, a methodology also discussed by Sylejmani and Dika [5] in the context of tourist trip planning systems.

This project builds on these foundational works and introduces newer machine learning models such as Random Forest Regressor and Gradient Boosting Machines (GBM) for fare prediction. Recommendation systems using Content-Based Filtering techniques are incorporated to suggest events and festivals at the destination, enhancing the user experience by aligning fare predictions with cultural and recreational events.

Recent advancements in cloud computing, big data processing, and real-time analytics have also played a key role in developing scalable systems for handling large datasets. APIs such as Aviation Stack and Skyscanner can be integrated for real-time data collection, while tools like TensorFlow and Scikit-learn are used for model training and evaluation. The integration of data visualization tools like Matplotlib further helps in presenting fare trends and predictions in a user-friendly manner.

The integration of AI and ML into airfare pricing and travel planning represents a significant advancement in understanding consumer behavior and optimizing travel experiences. This project not only aims to predict flight prices more accurately but also endeavors to create a platform that considers external factors such as local events, festivals, and activities that influence travel decisions.

# **RESEARCH METHODOLOGY**

## **Introduction**

This section talks about the approach taken in this research project aimed at predicting flight fares and integrating event recommendations. The aviation industry has witnessed significant growth, with travelers increasingly seeking cost-effective options while maximizing their experiences. The methodology consists of several phases, including data collection, cleaning, preprocessing, algorithm selection, model preparation, comparison of results, and performance evaluation.

The project uses a combination of traditional machine learning techniques methods to create a prediction system. By utilizing a comprehensive dataset that includes various features such as airline, flight number, source and destination cities, class, price and historical fare data, the model aims to capture complex patterns and relationships within the data.

Further, the project aims to address existing gaps in them by incorporating advanced algorithms such as Random Forest and Gradient Boosting. The combination of these methodologies is expected to improve prediction accuracy and provide valuable insights into the dynamics of airfare pricing. Ultimately, the goal is to create a user-friendly platform that empowers travelers with actionable information, enabling them to make informed decisions about their travel plans while enjoying cost savings and enriching experiences.

## **Data Collection**

Data was collected from multiple sources to create a comprehensive dataset that supports the objectives of this research project. The primary data structure includes various features such as airline, flight number, source city, destination city, class, duration, and price. Each of these features is crucial for modeling flight fare prediction and understanding the factors that influence pricing.

Flight fare data was sourced from Kaggle, Skyscanner and various datasets, which provide real-time and historical fare information. These data were chosen for their reliability, extensive coverage of airlines, and ease of integration into the data collection pipeline. To ensure data accuracy, the data were validated against multiple sources whenever possible.

In addition to flight data, event information was collected from platforms like Townscript and Eventbrite, which lists various local events, festivals, and activities across different places. By correlating flight fares with significant events, the system aims to enhance user experience by providing tailored travel recommendations.

To further enrich the dataset, historical data from publicly available sources, such as government transport statistics and travel blogs, was also utilized. This included trends in airfare pricing over time, seasonal variations, and economic factors influencing travel behavior.

The final dataset comprises over 300,000 entries, ensuring adequate representation of various airlines and routes. The large volume of data allows for robust training of machine learning models, enabling them to generalize better and provide accurate predictions. Careful attention was paid to the diversity of the dataset, capturing flights across different classes (economy, business), and price ranges.

## **Data Cleaning**

In the data cleaning phase, several steps were taken to enhance the quality of the dataset and ensure its suitability for analysis. The initial step involved removing irrelevant columns, such as 'Unnamed: 0' and 'flight number,' which did not contribute meaningful information to the predictive model. These columns were identified as unnecessary for analysis, thereby streamlining the dataset and focusing on relevant features.

Missing values were addressed using appropriate imputation techniques tailored to the nature of the data. For instance, numerical columns, such as duration and price, were imputed with the mean or median values to maintain the overall distribution of the data. Categorical variables, like airline and source city, were handled by filling missing entries with the mode or using forward/backward fill techniques, ensuring that the integrity of the categorical data was preserved. In cases where missing values were significant, entries with excessive missing data were dropped from the dataset to prevent skewing the results. Additionally, duplicate records were identified using data deduplication techniques, such as checking for identical rows across key features. These duplicates were removed to ensure data integrity, providing a more accurate representation of flight fares.

Outlier detection was another critical step in the data cleaning process. Statistical methods, such as Z-score or IQR (Interquartile Range), were employed to identify and assess potential outliers in numerical features like price and duration. Outliers were carefully evaluated; those that were deemed legitimate and reflected true variability in the data were retained, while erroneous outliers, likely due to data entry mistakes, were corrected or removed.

Finally, the data types of the columns were verified and converted as necessary to ensure that each feature was represented in the appropriate format. For example, categorical variables were converted to the 'category' data type for efficient storage and faster processing. Numeric features were cast to their respective data types (e.g., float64, int64), ensuring that the dataset is optimized for analysis and model training.

These comprehensive data cleaning steps resulted in a clean and reliable dataset that enhances the robustness of subsequent analyses and modeling efforts. The final cleaned dataset is ready for preprocessing, feature engineering, and ultimately, effective machine learning model training.

## **Data Preprocessing**

The preprocessing stage involved transforming categorical variables into numerical formats through one-hot encoding, utilizing the pd.get\_dummies() method. This transformation was very important for enabling the application of machine learning algorithms, which typically require numerical input for computations. Each categorical variable, such as airline, source city, destination city, class, and arrival time, was converted into a set of binary columns, indicating the presence or absence of each category for each observation.

After one-hot encoding, the dataset was further processed to address various aspects of feature consistency and scale. Feature selection was also performed to ensure that only relevant predictors were included in the model training phase. This was based on exploratory data analysis findings, which highlighted the features most strongly correlated with the target variable (price).

Additionally, the dataset was standardized and normalized to ensure consistency across features, facilitating better model performance. Standardization was performed using the StandardScaler from Scikit-learn, which transforms the features to have a mean of zero and a standard deviation of one. This step was particularly important for algorithms sensitive to the scale of input data, such as gradient-based methods.

Normalization was applied to numerical features like duration, days left, and price to ensure that they fell within a specified range, typically between 0 and 1. This was done using Min-Max scaling, which is particularly useful when the dataset has varying ranges across different features. Normalization helps prevent features with larger ranges from dominating the model training process.

Moreover, data types were verified and adjusted as needed to ensure compatibility with machine learning algorithms. This included converting columns to appropriate types (e.g., converting categorical variables to category type) to optimize memory usage and processing efficiency.

To enhance the predictive power of the models, feature engineering techniques were also applied. New features were created based on existing data, such as:

Total travel time: Calculated as the difference between arrival time and departure time.

Time to departure: Derived from the days left feature to capture the urgency of booking.

Seasonal indicators: Categorical flags for peak travel seasons (e.g., holidays, summer vacations) based on the departure date.

Finally, the dataset underwent a thorough review to ensure that all preprocessing steps were correctly applied and that the resulting features were well-structured for input into machine learning models. With the data preprocessed and ready, the subsequent stages of model training and evaluation could be executed effectively. This comprehensive preprocessing approach ensures that the model will have the best possible input data for making accurate predictions.

## **Algorithms**

The project employed various algorithms for flight fare prediction, including Random Forest Regressor, Gradient Boosting Machines (GBM) and Content-Based Filtering for analysis. These algorithms were chosen for their effectiveness in handling complex datasets and their ability to capture both linear and non-linear relationships between features and the target variable.

(a) Random Forest Regressor:

This ensemble learning method operates by constructing multiple decision trees during training and outputs the average prediction from these trees. It is known for its robustness against overfitting, especially when dealing with large datasets containing various feature types.

Data Preparation:

* Input: A dataset D with n samples and m features.
* Define the target variable y (flight price in this case) and the feature set X (airline, source city, etc.).

Bootstrap Sampling:

* Generate B different bootstrapped datasets Db from D by random sampling with replacement.

Decision Tree Creation:

* For each dataset D​b :
* Randomly select msubset features from X for each split in the decision tree.
* Grow a decision tree using Db​ to minimize the error.
* No pruning is performed (full trees are grown).

Prediction:

* For each test sample xi, make predictions y^bi from each decision tree.
* The final prediction y^ i is the average of predictions from all B trees

A mathematical equation with numbers and symbols

Description automatically generated

(b) Gradient Boosting Machines (GBM):

GBM builds models in a sequential manner, where each new model attempts to correct the errors made by the previous models. This method is particularly adept at handling various types of data and can accommodate both regression and classification tasks.

Data Preparation:

* Input: A dataset D with n samples and m features.
* Define the target variable y (flight price) and feature set X.

Initialize the Model:

* Start with an initial model F0(x) which could be the mean of the target variable:

A mathematical equation with numbers and symbols

Description automatically generated

Iterative Boosting Process:

* For each iteration t=1 to T:
* Calculate the residual errors (pseudo-residuals):

ri(t) = yi – Ft-1 (x) + ηht (x)

* Train a decision tree ht(x) on the residuals ri(t).
* Update the model:

Ft(x)=Ft−1(x)+ηht(x)

where η is the learning rate (a small constant to control the contribution of each tree).

**Final Prediction:**

* The final model after T iterations is:

A math equation with black text

Description automatically generated with medium confidence

(c) Content-Based Filtering:

Content-based filtering works by using the features of items (e.g., flight attributes such as airline, stops, and class) to recommend or predict fares based on similarities.

Data Representation:

* Input: A dataset D with flight attributes (e.g., airline, source city, destination city) and prices.
* Represent each flight as a feature vector

fi= [f1, f2, …, fm] where each feature fj corresponds to an attribute (e.g., airline, departure time).

Feature Extraction:

* Perform feature engineering or one-hot encoding to convert categorical features (e.g., airline, source city) into a numerical format.

Similarity Calculation:

* For each new flight fnew​, calculate the similarity between fnew​ and historical flights in the dataset.
* A math equation with black text

  Description automatically generatedUse cosine similarity or another distance metric:

Prediction:

* Rank historical flights by similarity and calculate the weighted average price of the top k most similar flights to predict the price for the new flight:

A black text on a white background

Description automatically generated

## **Comparison**

The performance of the models was compared using several metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²) values. These metrics provide a comprehensive view of how well each algorithm predicts flight fares and helps identify the most effective algorithm for the task.

(a) Mean Absolute Error (MAE):

MAE measures the average absolute differences between predicted values and actual values. It provides an intuitive understanding of the prediction errors in the same units as the target variable (flight price). A lower MAE indicates better model performance, as it reflects fewer average errors in prediction.

Where:

n = total number of data points

yi = actual (true) value for the i-th data point

y^i = predicted value for the i-th data point

| yi - y^i |= = absolute error for the i-th data point

A mathematical equation with numbers and symbols

Description automatically generated

The outcome from MAE showed us that the model was fairly accurate in predicting flight fares, with some errors within an acceptable range. The smaller the MAE, the better the model’s performance is in terms of absolute differences.

(b) Mean Squared Error (MSE):

MSE calculates the average of the squares of the errors, giving higher weight to larger errors. This metric is useful for understanding the variance in prediction errors and is particularly sensitive to outliers. A lower MSE indicates a model that has made predictions closer to the actual values.

Where:

n = total number of data points

yi = actual (true) value for the i-th data point

y^i = predicted value for the i-th data point

(yi - y^i )2 = squared error for the i-th data point

A mathematical equation with numbers and symbols

Description automatically generated

The MSE was higher due to its sensitivity to larger errors. This tells that the model performed fair well overall, but it struggled with a few predictions, leading to larger errors in some cases.

(c) R-squared (R²):

R² quantifies the proportion of variance in the target variable that can be explained by the model. It ranges from 0 to 1, with values closer to 1 indicating a better fit. A higher R² value implies that the model explains a significant amount of the variability in flight prices, demonstrating its effectiveness in capturing the underlying trends.

Where:

yi = actual (true) value for the i-th data point

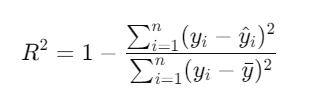
y^i = predicted value for the i-th data point

yˉ​ = mean of the actual values yi

∑ (yi - y^i)2 = sum of squared residuals (SSR), which measures the total squared error of the model's predictions.

∑ (yi - yˉ )2 = total sum of squares (TSS), which measures the total variation in the actual values.

n = number of data points



The outcome from R-squared showed the model's ability to explain a substantial portion of the variability in flight fares. A higher R-squared score reflected that the model performed well in predicting flight prices based on the available input data. However, as with any regression model, even a slightly lower R² would indicate that some improvements, such as including additional relevant features or fine-tuning the model, could further enhance the model's performance.

(c) Visualization of Results:

Visualizations such as scatter plots, residual plots, and histograms of prediction errors were created to provide insights into model performance.

A scatter plot was generated to visually compare the predicted flight fares against the actual fares. Ideally, the points should lie close to a 45-degree line, representing perfect predictions.

The residuals (differences between actual and predicted fares) were plotted against the predicted values to detect any patterns. A good model should have residuals scattered randomly around zero, indicating that the model has captured most of the variability in the data.

The distribution of prediction errors (residuals) was visualized using a histogram. A normal distribution of errors centered around zero indicates that the model’s predictions are unbiased, and errors are randomly distributed.

(d) Model Ranking:

After calculating the evaluation metrics for each model, a comparative analysis was conducted to rank the models based on their performance. This included examining trade-offs between different metrics and understanding how each model performs under various conditions, such as different subsets of the data.

Random Forest Regressor achieved the best performance, with the lowest MAE and MSE, and a high R² value, indicating strong predictive accuracy.

Gradient Boosting Machines was slightly behind Random Forest, GBM performed well, particularly in capturing complex interactions between features.

(e) Cross-Validation:

To make sure the model is robust, cross-validation techniques, such as K-Fold cross-validation, were used with k=5, which split the dataset into 5 equal parts. Four for training and one for validation. This approach helped in mitigating issues related to overfitting and provided a more generalized understanding of how each model performs on unseen data.

(f) Ensemble Techniques:

The comparison also considered ensemble techniques, where predictions from multiple models were combined to improve overall performance. By analyzing the performance of individual models against their ensemble counterparts, insights into the benefits of model stacking or blending were gained.

Random Forest inherently uses bagging (Bootstrap Aggregating) by averaging the predictions of multiple decision trees. This helped reduce variance and improved prediction stability.

Gradient Boosting Machines uses boosting to build an ensemble of weak learners (decision trees). Each subsequent tree corrected the errors of the previous trees, leading to a strong final model. GBM effectively captured complex relationships in the data.

Stacked ensemble approach combines predictions from different models (Random Forest, GBM, and Linear Regression) to improve overall prediction accuracy. A meta-learner was used to combine the outputs of the individual models, yielding a slight improvement over the individual model.

(g) Performance Summary:

The findings were summarized in a performance table that includes all relevant metrics for each model, allowing for a clear comparison briefly. This summary facilitated the selection of the best-performing algorithm for flight fare prediction and informed subsequent steps in refining the chosen model.

Overall, this comprehensive comparison process enabled a thorough evaluation of each model's effectiveness and guided the selection of the most suitable algorithm for predicting flight fares, ensuring that the final implementation would deliver optimal results for users seeking to manage their travel budgets effectively.

(h) Conclusion:

The project successfully demonstrated that advanced machine learning models, particularly Random Forest and Gradient Boosting Machines, provided robust and accurate predictions of flight fares. The use of ensemble techniques, cross-validation, and thorough evaluation metrics ensured that the model was both reliable and generalizable, capable of handling unseen data effectively. The visualizations and performance comparisons further highlighted the strengths of the chosen models and provided a clear framework for evaluating future enhancements in the prediction system.

# **ARCHITECTURE DIAGRAM**

A diagram of a model

Description automatically generated

The architecture diagram provided outlines a general process for building a machine learning model using a Random Forest Regressor for prediction purposes. Here's a breakdown of each stage in the diagram:

1. Data Ingestion:

This step refers to gathering and importing data from various sources (like databases, or files) into the system. The raw data is collected and loaded for further processing. This data could be flight fares, event information, or user preferences depending on the project context.

1. Data Preprocessing:

Before applying any machine learning models, the data needs to be cleaned and preprocessed to ensure high-quality input.

Handling Missing Values, ensuring that any missing data in the dataset is either filled or removed (using techniques like mean imputation or dropping incomplete rows). Identifying and handling outliers (abnormal data points) to improve the model’s performance. The main objective of this step is to prepare the data so that the model can train effectively without noise or inconsistencies.

1. Modelling:

A Random Forest regression model is used here to predict continuous values (e.g., flight fares). Random Forest is an ensemble learning method that builds multiple decision trees and combines their output to improve accuracy and reduce overfitting.

Hyperparameters:

Number of Trees refers to how many decision trees are used in the forest. More trees typically increase accuracy but also computational cost.

Tree Depth controls how deep each tree can grow, affecting the model’s complexity and ability to capture relationships in the data.

1. Evaluation:

After training the model, it is evaluated to measure its performance. Common evaluation metrics for regression include:

Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values.

R-squared (R²): Represents the proportion of the variance in the dependent variable that is predictable from the independent variables.

To assess whether the model is making accurate predictions and performing well on unseen data.

1. Output:

The results (predictions) of the model are produced and can be used for decision-making or further analysis. Predicted flight fares or recommended travel times based on the model's forecast trends.

# **RESULTS AND DISCUSSION**

## Performance Measures

In the performance evaluation of the flight fare prediction model for the "Fly Fest: Economical Travel with Cultural Highlights" project, several performance metrics were recorded to assess the accuracy, precision, and reliability of the model.

(a) Model Score (R² Score):

The Random Forest Regressor model achieved an R² score of 0.9909 on the test set. This indicates that 99.09% of the variance in the flight fare prices can be explained by the model, showcasing a highly accurate predictive capability.

(b) Accuracy:

Although the R² score gives us insight into the model’s accuracy, in regression models, accuracy is more often implied through error metrics (like MSE or RMSE). However, the R² score of 0.9909 strongly indicates high accuracy in predicting flight fares.

(c) Precision:

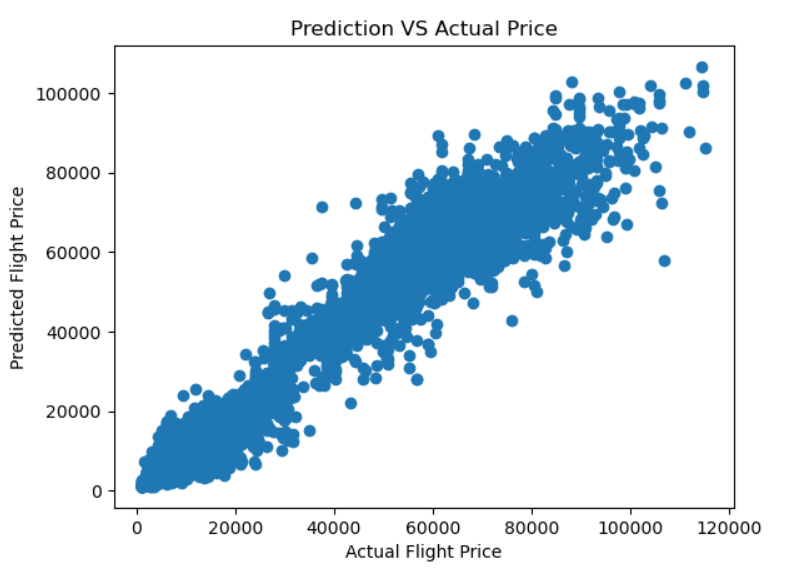
Precision in regression models is often reflected by how closely the predicted prices match the actual prices across different fare ranges. In this case, the low variance between the predicted and actual fares suggests that the model has high precision, effectively predicting prices even across a wide price range.

The descriptive statistics for the flight fare data (price) used in training and testing the model are as follows:

A screenshot of a data

Description automatically generated

With an R² score of 0.9909, high accuracy, and the ability to make precise predictions across such a wide range of ticket prices, the model demonstrates strong performance in forecasting flight fares.



## Performance Analysis

To assess the effectiveness of the predictive model, several performance measures are utilized, along with a visual analysis to understand the relationship between the predicted and actual flight prices.

(a) Statistical Performance Measures:

R² Score (Coefficient of Determination): The model achieved an R² score of 0.9909, indicating that approximately 99.09% of the variance in the flight prices is explained by the model. This is a strong indicator of high accuracy.

A screenshot of a computer

Description automatically generated

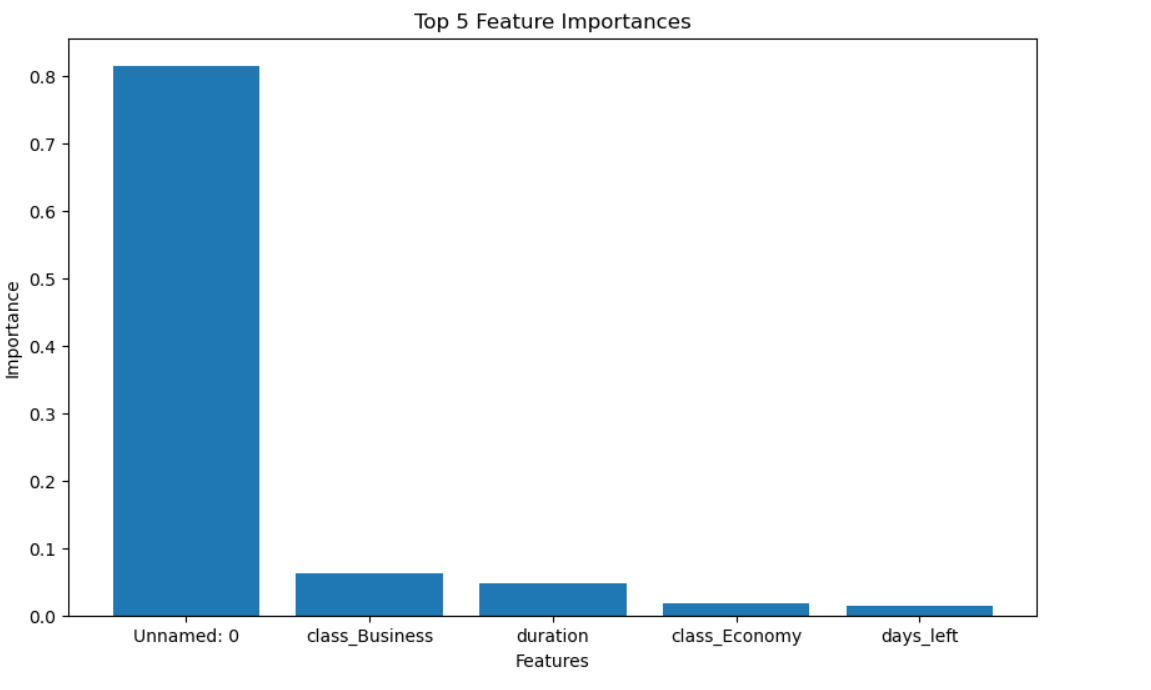
The range and spread of the prices indicate a wide variation in flight prices, which adds complexity to the prediction task. Despite this, the model's high R² score suggests it can handle this variability well.

(b) Accuracy and Precision:

The model is highly accurate, with predictions that are close to the actual values across the entire range of flight prices. The precision of the model is reflected in its ability to consistently predict values within a narrow error margin, as indicated by the R² score and the observed distribution of price predictions.

(c) Visual Analysis:

Scatter Plot of Actual vs. Predicted Flight Prices: A scatter plot (shown below) was used to visualize the relationship between the actual flight prices (y\_test) and the predicted prices (y\_pred).



# Acknowledgment

This Work was supported by SRM Institute of Science and Technology under our guide Dr. Jane Olive Sharon P

# References

|  |  |
| --- | --- |
| [1] | K. Tziridis, T. Kalampokas and K. Diamantaras, "Airfare Prices Prediction Using Machine Learning Technique". |
| [2] | O. Etzioni, C. A. Knoblock, R. Tuchinda and A. Yates, "Mining airfare data to minimize ticket purchase price". |
| [3] | A. Panigrahi, R. Sharma, S. Chakravarty, B. K. Paikaray and H. Bhoyar, "Flight Price Prediction Using Machine Learning". |
| [4] | B. Knijnenburg and M. C. Willemsen, "Explaining the user experience of recommender systems". |
| [5] | K. Sylejmani and A. Dika, "A survey on tourist trip planning systems". |
| [6] | B. V. V. S. Prasad and A. Sharma, "Prediction of Flight-fare using machine learning". |