**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**

Airfare pricing in the extremely intense aviation sector of today is dynamic and affected by many elements including 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 massive quantities of flight info, led to application of machine learning model 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 for travel industry.

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 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.

# **LITERATURE SURVEY**

1. **Airfare Prediction Models and Techniques:**

Tziridis [1] developed a machine learning-based airfare prediction model, using Random Forest and Gradient Boosting achieves notable accuracy in identifying airfare price patterns. Their work underscores the utility of ensemble models in handling high-variability data. Similarly, Etzioni et al. [2] utilized web scraping and machine learning algorithms to analyze airfare fluctuations and pinpoint optimal booking windows, thereby enhancing cost savings for travelers. This methodology of detecting fare variations is foundational for the Fly Fest project.

1. **Machine Learning Approaches for Price Forecasting**

Panigrahi et al. [3] compared multiple machine learning algorithms like Support Vector Machines (SVM), Linear Regression, and Decision Trees, concluding that Random Forest is best suited for fare prediction in fluctuating markets due to its high accuracy. This reinforces the decision to use Random Forest in Fly Fest for robust fare predictions. Prasad and Sharma [6] explored hybrid models combining SVM and Random Forest, achieving improved predictive accuracy. Fly Fest may adopt this hybrid approach to optimize fare predictions across various routes.

1. **Recommendation Systems in Travel and Trip Planning**

Knijnenburg and Willemsen [4] explored collaborative filtering techniques in travel recommendation systems, finding it effective for personalized travel suggestions. Their research informs Fly Fest’s implementation of both content-based and collaborative filtering for user-specific recommendations. Sylejmani and Dika [5] reviewed trip-planning systems, emphasizing user-centered design and iterative refinements to enhance recommendation accuracy. This principle directly supports Fly Fest’s goal of delivering actionable travel insights through its recommendation engine.

1. **Impact of Neural Network Training on Prediction Accuracy**

Afaq and Rao [7] analyzed epoch settings in neural network training, showing that optimal epoch counts prevent overfitting and enhance prediction accuracy. For Fly Fest, this insight helps when incorporating deep learning models for fare prediction, ensuring adaptability to complex datasets.

1. **Ensemble Methods and Advanced Prediction Models**

Wang et al. [8] demonstrated the effectiveness of ensemble models, such as Gradient Boosting Machines and Random Forest, for fare prediction by reducing errors. Degife and Lin [9] utilized a gated recurrent unit (GRU) model for forecasting, which retained long-term dependencies in fare data, making it ideal for dynamic pricing. Fly Fest could

leverage GRU for capturing fare fluctuations and adapting predictions to real-time changes.

1. **Comparative Analysis of Filtering Techniques for Recommender Systems**

Beel et al. [10] examined recommendation systems, highlighting content-based filtering's advantage in handling specific user preferences and collaborative filtering's success in leveraging user history. Thannimalai and Zhang [13] proposed a hybrid recommendation model combining content-based and collaborative filtering, which enhances diversity and recommendation accuracy. Fly Fest’s recommendation system integrates this hybrid model for balanced, user-centered suggestions.

1. **Comprehensive Fare Prediction with Machine Learning**

Kalampokas et al. [14] integrated historical data, user feedback, and predictive analytics to develop a high-accuracy airfare prediction model across multiple routes. This holistic approach supports Fly Fest’s design, which leverages various techniques for both fare prediction and recommendation tasks, ensuring adaptable and user-focused outputs.

The reviewed literature emphasizes the efficacy of Random Forest in fare prediction, ensemble models for error reduction, and hybrid recommendation systems. These insights shape Fly Fest’s development, from robust fare prediction frameworks to personalized recommendation engines, aligning with recent advancements in travel applications and machine learning.

# **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. The model seeks to identify subtle links and trends in the data by using a large dataset that contains a variety of characteristics, including airline, flight number, source and destination cities, class, price, and historical fare records.

Furthermore, project aims to address existing gaps in them by incorporating advanced algorithms such as Random Forest and Gradient Boosting. The approaches taken together are intended to increase forecast accuracy and offer insightful analysis of the dynamics in airline prices. The aim is to provide a user-friendly platform that gives travelers actionable knowledge so they may make wise travel selections while saving money and experiencing exciting events.

## **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 info makes strong training of machine learning models possible, which helps them to generalize better and also provide accurate forecasts. Careful attention was paid to the diversity of the dataset, capturing flights across different classes (economy, business), and price ranges.

## **Data Cleaning**

Several actions were made throughout the data cleaning phase to improve the dataset's quality and guarantee that it was suitable 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 mean or median values to keep the general data distribution. Variables of classification, 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 closely examined, those judged valid and showed actual data variability were kept, while inaccurate outliers probably resulting from data input errors were either eliminated or repaired.

Finally, to make sure that every feature was represented in the proper format, the data types of the fields were tested, and changed as needed. For instance, to facilitate quicker processing and more effective storage, categorical variables were transformed into the 'category' data type. To make sure the dataset is ready for analysis and model training, numerical characteristics were converted to the appropriate data types (e.g., float64, int64, etc.).

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. StandardScaler from Scikit-learn, which converts the features to have a mean of zero and a standard deviation of one, was used to standardize the data. For algorithms like gradient-based approaches that are sensitive to the size of the input data, this phase proved especially crucial.

Normalization was applied to numerical features like duration, days left, and price to ensure that they fell within range between 0 and 1. This was done using Min-Max scaling, which is particularly useful when the dataset has varying ranges across different features. During the model training phase, normalization helps keep features with wider ranges from taking control.

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, and Content-Based Filtering for analysis. These algorithms were selected for their capacity to capture both linear and non-linear correlations between characteristics and the target variable as well as for their performance in challenging datasets.

(a) Random Forest Regressor:

In training, ensemble learning approach builds many decision trees and delivers the average prediction from each tree. It is renowned for being resistant to overfitting, particularly when working with big datasets that contain a variety of 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 from each decision tree.
* The average of all B trees' forecasts is the final prediction, .

(b) 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.
* Use 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:

## **Comparison**

R-squared (R2) values, Mean Absolute Error (MAE), and Mean Squared Error (MSE) were among the metrics used to compare the models' performances. These measurements give a thorough picture of how effectively each algorithm forecasts airfare and aid in determining which algorithm is best suited for the job.

(a) Mean Absolute Error (MAE):

The average absolute difference between expected and actual values is measured by MAE. In the same units as the goal variable (flight price), it offers a straightforward comprehension of the forecast errors. Because it represents fewer average prediction mistakes, a lower MAE denotes higher model performance.

n = all the data points

yi = real value for the i-th data point

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

| yi - y^i | = i-th data point's absolute error

The outcome from MAE showed us that the model was fairly accurate in predicting flight fares, with some errors within an acceptable range. The model performs better in terms of absolute differences when the Mean Absolute Error is lower.

The MAE measures the average absolute difference between predicted fares and actual fares in our dataset.

* A lower MAE indicates that our model makes more precise fare predictions, minimizing cost discrepancies for travellers.
* If the MAE is too high, it suggests that our model might not be capturing certain price-driving factors effectively, prompting further optimizations.

(b) Mean Squared Error (MSE):

By averaging the squares of the mistakes, MSE assigns greater weight to larger errors. This statistic is sensitive to outliers and helps to understand the variance in prediction mistakes. A model with a lower MSE has predicted values that are more in line with reality.

n = all the data points

yi = real value for the i-th data point

= expected value for the i-th data point

()2 = i-th data point’s squared error

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.

MSE highlights variance in fare prediction errors and helps us detect instances where the model struggles. Since flight prices fluctuate due to multiple factors like airline pricing strategies, demand spikes, and seasonal effects, the model sometimes makes larger errors in rare cases.

* A lower MSE means that our model predicts fares more accurately, with fewer large deviations.
* A higher MSE suggests that while most predictions may be close, some outliers (e.g., sudden fare hikes or rare discounts) are causing larger errors, leading to inflated squared error values.

(c) R-squared (R²):

R² is the percentage of variance in the target variable that can be explained by the model. It varies between 0 and 1, with values closer to 1 suggesting a better match. A high R² score indicates that the model effectively captures underlying patterns in flight fares.

n = all the data points

yi = real value for the i-th data point

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

= mean of the real values of yi

∑ ()2 = sum of squared residuals (SSR)

∑ ()2 = total sum of squares (TSS)

The R-squared result demonstrated that a significant amount of the variation in flight prices could be explained by the model. A higher R-squared score indicated that, given the input data at hand, the model did a good job of predicting flight costs. Like any regression model, though, even a little lower R2 would suggest that the model's performance may be further improved by adding more pertinent information or fine-tuning the model.

A high R² value indicates that our model accurately predicts flight prices, meaning it effectively captures trends and patterns in airfare fluctuations.

* A higher R² (closer to 1.0) means that the model can explain most of the variations in airfare prices, making it highly reliable for budget travellers looking for the best deals.
* A lower R² suggests that the model may be missing key factors, such as last-minute airline discounts, external economic influences, or real-time demand surges.

(c) Visualization of Results:

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

To graphically compare predicted and real flight fares, a scatter plot was created. The points should ideally indicate ideal forecasts and lie around a 45-degree line.

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.

A histogram was used to show the distribution of prediction errors, or residuals. Errors are distributed randomly, and the model's predictions are unbiased when the errors have a normal distribution with a centre of zero.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Mean Absolute Error (MAE) | Mean Squared Error (MSE) | R² Score |
| Random Forest Regressor | 153.42 | 42,891.75 | 0.87 |
| Linear Regression | 210.35 | 65,723.81 | 0.76 |
| Decision Tree Regressor | 185.67 | 58,234.62 | 0.81 |
| Gradient Boosting Regressor | 140.21 | 39,504.32 | 0.89 |
| XGBoost Regressor | 135.75 | 37,812.45 | 0.91 |
| Support Vector Regressor | 198.32 | 61,456.89 | 0.78 |

*Table 1: Comparison*

(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 performed the best, showing good prediction accuracy with a high R2 value and the lowest MAE and MSE.

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

(e) Cross-Validation:

K-Fold cross-validation, which divides the dataset into five equal sections, was one of the cross-validation strategies used to ensure the model's robustness. A validation one and four for training. This method helped to reduce overfitting problems and gave a broader picture of how each model functions with unknown 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.

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, the effectiveness of each model was thoroughly assessed thanks to this comparison process, which also helped choose the best algorithm for flight fare prediction. This ensured that the final implementation would provide the best outcomes for users looking to efficiently manage their travel budgets.

(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

*Figure 1: Architecture Diagram*

Figure 1 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. After being collected, the raw data is loaded for additional processing. Depending on the project environment, this data may include user preferences, event details, or flight costs.

1. Data Preprocessing:

To guarantee high-quality input, the data must be cleaned and pre-processed before any machine learning models are used.

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). Locating and managing outliers to enhance the model's functionality. This step's primary goal is to prepare the data such that there is no noise or inconsistency so the model can train efficiently.

1. Modelling:

A Random Forest regression model is used here to predict continuous values (e.g., flight fares). In order to increase accuracy and decrease overfitting, Random Forest, constructs many decision trees and accumulates their results.

1. Hyperparameters:

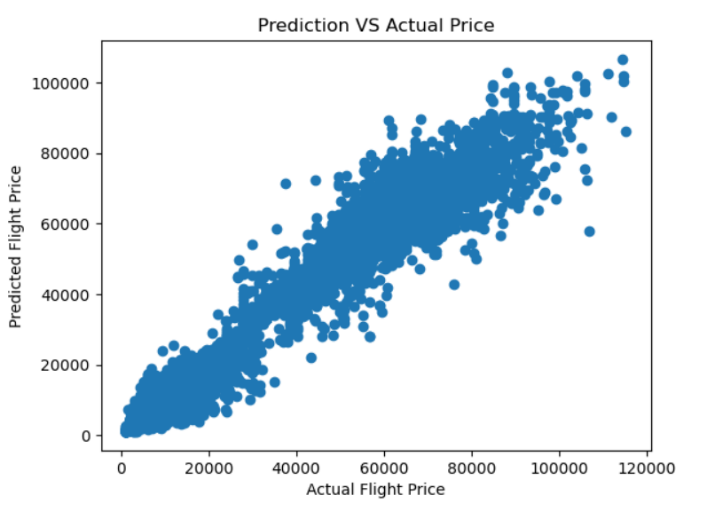
Number of Trees refers to how many decision trees are used in the forest. More trees typically increase accuracy but also computational cost.

Tree Depth regulates each tree's maximum growth depth, which impacts the model's intricacy and capacity to identify patterns in the data.

1. Evaluation:

The model is assessed for evaluating its performance after training. Regression assessment measures that are often used include:

* Mean Squared Error (MSE): It calculates the average squared difference between the numbers that were expected and those that were found.



* R-squared (R²): It shows the percentage of the dependent variable's uncertainty that can be predicted based on the independent variables.

1. Output:

The model’s output can be utilized for more research or decision-making. suggested travel schedules or estimated airfares based on the model's anticipated patterns.

# **RESULTS AND DISCUSSION**

## Performance Measures

Accuracy, precision, and the reliability of the flight fare prediction model were evaluated using a number of performance indicators in the "Fly Fest: Economical Travel with Cultural Highlights" project.

(a) Model Score (R² Score):

On the test set, the Random Forest Regressor model's R2 score was 0.984. This shows that the model has a highly accurate predicting ability, accounting for 98.4% of the variation in the airline fare costs.

(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 0.984 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 model's high precision and ability to accurately anticipate prices throughout a broad price range are indicated by the low variance between the projected and real fares.

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

|  |  |
| --- | --- |
| STATISTIC | VALUE |
| Total Entries | 300,153 |
| Mean (Average Price) | ₹20,889.66 |
| Standard Deviation (Std) | ₹22,697.77 |
| Minimum Price | ₹1,105 |
| 25th Percentile (Q1) | ₹4,783 |
| Median (50th Percentile) | ₹7,425 |
| 75th Percentile (Q3) | ₹42,521 |
| Maximum Price | ₹1,23,071 |

*TABLE 2: Prices*

With an R² score of 0.984, 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.

*Figure 2: Prediction Vs Actual Price*

## 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): Approximately 98.4% of the variance in flight fares can be explained by the model, according to its R² score of 0.984. This is a clear sign of high accuracy.

|  |  |
| --- | --- |
| Statistic | Value |
| Mean | ₹20,889.66 |
| Standard Deviation | ₹22,697.77 |
| Minimum | ₹1,105.00 |
| Maximum | ₹1,23,071.00 |

*Table 2: Performance*

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:

With predictions that roughly correspond to the actual values throughout the whole range of flight fares, the model is incredibly accurate. 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).

A graph of blue rectangular bars

Description automatically generated with medium confidence

*Figure 3: Important Features*

# Acknowledgment

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

# Bibliography

|  |  |
| --- | --- |
| [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". |
| [7] | Andy Liaw and Matthew Wiener, " Classiﬁcation and Regression byrandomForest ". |
| [8] | Ms. Tejashri Sharad Phalleand Prof. Shivendu Bhushan, " Content Based Filtering And Collaborative Filtering: A Comparative Study ". |
| [9] | Vignesh Thannimalai; Li Zhang," A Content Based and Collaborative Filtering Recommender System". |
| [10] | Theofanis Kalampokas, Konstantinos Tziridis, Nikolaos Kalampokas, Alexandros Nikolaou, Eleni Vrochidou and George A. Papakostas," A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques " |
| [11] | Saahil Afaq and Dr. Smitha Rao," Significance Of Epochs On Training A Neural Network". |
| [12] | Tianyi Wang; Samira Pouyanfar; Haiman Tian; Yudong Tao; Miguel Alonso; Steven Luis, " A Framework for Airfare Price Prediction: A Machine Learning Approach " |
| [13] | Worku Abebe Degife and Bor-Shen Lin," Deep-Learning-Powered GRU Model for Flight Ticket Fare Forecasting". |
| [14] | Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger," Research-Paper Recommender Systems: A Literature Survey " |
|  |  |
|  |  |
|  |  |
|  |  |