

Birla Institute of Technology and Science, Pilani
(Department of Management)



MPBA G513 – Predictive Analytics

**Forecasting Flight Ticket Prices Using Predictive Analytics:
Enhancing Airline Revenue Management**

Submitted by

Group 2

Shreya Gupta	2024H1540840P
Anakari Sravya Hadassa	2024H1540850P
Kratika Garg	2024H1540855P
Tarani Satya Kandula	2024H1540861P
Aamina Nooraiyeen	2024H1540863P

Abstract

Precise flight ticket price forecasting is an increasing requirement for airlines that operate in a competitive and dynamic environment. This project uses predictive analytics to predict airfare prices with the main objective of improving airline revenue management. Based on a structured Kaggle dataset of historical flight data, the project uses machine learning algorithms to reveal pricing trends and facilitate data-driven decision-making.

The key attributes in the dataset are airline, class of flight, source and destination cities, duration, stops, and days left until departure. The predictive models used Linear Regression, Ridge Regression, Random Forest Regressor and XG Boost were evaluated using performance measures like Mean Squared Error (MSE) and R-squared (R^2). Out of these, the Random Forest Regressor showed better performance with an R^2 value of 0.98.

To close the loop between analysis and usage, a Streamlit application accessible to users was created for real-time prediction of fares. It also points out factors of influence in airfare, with airline and departure time coming out as predictors.

By providing timely and accurate forecasts, this project can help airline businesses in maximizing revenue strategies, dynamic pricing, and enhance customer satisfaction through enhanced transparency and fare planning.

Index

Abstract.....	2
Index.....	3
Literature Review.....	4
Problem Statement.....	5
Dataset Description.....	5
Methodology.....	6
1. Data Cleaning.....	6
2. EDA.....	6
3. Feature Encoding.....	7
4. Feature Selection and Scaling.....	7
5. Multicollinearity Check (VIF).....	7
6. Model Selection.....	8
7. Model Training.....	8
8. Model Evaluation.....	10
9. Feature Importance Analysis:.....	11
10. Model Deployment.....	13
Results & Analysis.....	14
Brand Perception Insights.....	17
Conclusion.....	18
References.....	18

Introduction

In the current rapid and competitive air transport sector, pricing has evolved to be more dynamic and data-driven. Airfares change often based on a variety of factors, ranging from operational aspects to demand from consumers and competition in the market. For airlines and consumers alike, being able to predict fare changes is paramount—airlines want to optimize revenue, while travelers want the best value.

With increased accessibility of past flight data and innovations in machine learning, there lies a significant possibility to unravel the pricing behaviors and provide credible fare predictions. The project exploits the potential by seeking to determine whether various data features like airline, route, class of travel, booking horizon, and schedule are useful for accurately predicting ticket fares.

In addition to constructing forecasting models, this research seeks to identify the underlying patterns and drivers that affect airfare prices. In doing so, it not only makes a technical contribution to the price forecasting field but also provides practical insights that can inform strategic decision-making in airline revenue management and improve the user experience for travelers.

Literature Review

Machine learning (ML) usage to forecast airfare has drawn significant momentum due to the fact that it will help enhance revenue management and enlighten consumers with prices. The traditional statistical models are not capable of keeping pace with the complexities and the non-linear relationships that define airfare prices, which makes scholars turn towards sophisticated ML techniques.

Doganis et al. (2006) emphasized the imperative requirement for data-driven approaches to airline revenue management, noting that historical fare data could dramatically improve forecasting performance. Building on this, **Grover and Mehta** (2017) demonstrated that ensemble models like Random Forest and Gradient Boosting outperform traditional linear models in dealing with the complex variables influencing airfare prices.

Bhambri and Ratra (2019) investigated ensemble learning techniques for predicting airline ticket prices and found that Random Forest models yielded lower prediction errors compared to other methods. Their study points out the robustness of ensemble techniques in capturing the intricate patterns in airfare data.

Arya et al. (2021) have highlighted feature engineering, i.e., time-variable variable conversion, as a significant step in enhancing the performance of a model. According to their research, appropriate preprocessing and feature engineering can prove to be significant in enhancing the predictive capability of ML models for forecasting airfares.

Further developments are seen in **Liu's (2023)** study, which conducted an extensive feature correlation analysis and compared various ML models for predicting air ticket prices. The study concluded that ensemble learning-based regression algorithms such as Random Forest and Gradient Boosting provide higher prediction accuracy than traditional methods.

Furthermore, **Upadhye et al. (2024)** proposed a hybrid solution that integrated K-Means clustering with decision tree ensembles to overcome computational inefficiencies and overfitting in big flight datasets. Their localized modeling method efficiently segments data, enabling more accurate and efficient fare predictions.

Collectively, these works highlight the evolving role played by machine learning to forecast airfares, ensemble methods importance, feature engineering, and the role of hybrid model-based methodologies in creating powerful prediction systems.

Problem Statement

To predict flight ticket prices using machine learning techniques in order to assist airlines in optimizing revenue management and enable dynamic pricing based on key travel and operational factors.

Dataset Description

Source: To address the problem of forecasting airfare prices for airline revenue optimization, we use a structured dataset titled “Flights_dataset.csv”, comprising 300,153 flight records with 11 predictive features and 1 target variable (price). This is a dataset curated from Ease my trip which is available in Kaggle, structured for fare prediction and modeling purposes.

Key Features:

- **Categorical:** airline, flight, source_city, destination_city, departure_time, arrival_time, stops, class
- **Numerical:** duration, days_left
- **Target Variable:** Price

Methodology

1. Data Cleaning

The dataset was initially loaded into a Pandas DataFrame, and basic data exploration was conducted to understand its structure, checked for missing values, and inspected the data types of each column.

- Dropped unnecessary columns (Unnamed: 0, flight).
- Handled missing values using the dropna() method to ensure that no incomplete rows were used in model training.

2. EDA

Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying patterns, trends, and relationships within the dataset before proceeding to modeling or drawing insights. By visualizing and summarizing the data, EDA helps to uncover distributions, detect anomalies, and identify key features that influence the target variable. In this section, we analyze airfare pricing, booking trends, and airline operations using a combination of visualizations and tabular summaries.

- **Distribution of Airfare Prices:** The airfare price distribution is highly right-skewed, with the majority of tickets priced below ₹20,000 and a noticeable secondary peak around ₹50,000–₹60,000, likely due to premium or business class fares.
- **Most Booked Flights from Source Cities:** Delhi and Mumbai are the leading air travel hubs, each accounting for over 60,000 flights, followed by Bangalore, Kolkata, Hyderabad, and Chennai, reflecting their importance in the domestic flight network.
- **Airfare Price Trend vs Days Left by Class:** Both economy and business class fares decrease as the number of days left before departure increases, with last-minute bookings being significantly more expensive. Business class fares remain substantially higher than economy throughout, but both stabilize when booked well in advance.
- **Flight Count by Airline and City:** Indigo and Vistara dominate the market in terms of flight counts from major cities. Indigo has the highest number of flights from Bangalore, while Vistara leads in Delhi and Mumbai, highlighting their strong operational presence on these routes.

These insights provide a comprehensive overview of airfare dynamics, passenger booking behavior, and airline activity in the dataset, forming a strong foundation for further analysis.

3. Feature Encoding

Several columns in the dataset, such as the airline, departure time, and source city, were categorical. To make them usable by machine learning models, these categorical variables were encoded using Label Encoding.

A Label Encoder was used to transform categorical columns into numeric values, which is necessary for regression models.

4. Feature Selection and Scaling

- **Feature Selection:** The target, price, was isolated from the features (X) to create the input and output datasets used for model training.
- **Scaling:** A StandardScaler was used to scale the features so that all features had a mean of 0 and standard deviation of 1. This enhances the performance and convergence of machine learning algorithms.

5. Multicollinearity Check (VIF)

The Variance Inflation Factor (VIF) was computed to identify potential multicollinearity between the features. Multicollinearity occurs when two or more predictors are highly correlated, which can distort the model's estimates. The VIF values were very close to 1, which showed no significant multicollinearity, ensuring that the features were independent of each other.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd

# Assume X is your feature DataFrame (already scaled/encoded)
# If you use NumPy arrays, convert back to DataFrame for VIF:
X_scaled_df = pd.DataFrame(X_train_scaled, columns=X.columns)

# Calculate VIF
vif_data = pd.DataFrame({})
vif_data["Feature"] = X_scaled_df.columns
vif_data["VIF"] = [variance_inflation_factor(X_scaled_df.values, i) for i in range(X_scaled_df.shape[1])]

print(vif_data.sort_values(by="VIF", ascending=False))
```

	Feature	VIF
7	duration	1.319315
3	stops	1.297214
6	class	1.056260
1	source_city	1.055840
5	destination_city	1.055364
0	airline	1.041026
2	departure_time	1.015717
4	arrival_time	1.007887
8	days left	1.002596

Figure 1: Code snippet for VIF values of different columns

6. Model Selection

Three supervised regression algorithms were implemented to compare performance:

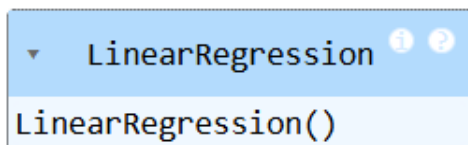
- **Linear Regression:** A baseline model to assess linear relationships.
- **Ridge Regression:** A regularized linear model to reduce multicollinearity and overfitting.
- **Random Forest Regressor:** An ensemble-based model capable of capturing non-linear patterns in complex datasets.
- **XGBoost** (Extreme Gradient Boosting): is a powerful and efficient machine learning algorithm based on gradient boosting that excels in both speed and predictive accuracy for structured/tabular data.

7. Model Training

The python code snippets for each of the 3 models is given below:

- **Linear Regression:**

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

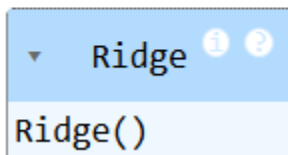


A screenshot of a Jupyter Notebook cell. The top part shows a dropdown menu for the class 'LinearRegression' with an information icon and a help icon. Below the dropdown, the constructor 'LinearRegression()' is displayed.

Figure 2: Model fitted for Linear Regression

- **Ridge Regression:**

```
from sklearn.linear_model import Ridge
ridge = Ridge(alpha=1.0)
ridge.fit(X_train_scaled, y_train)
```



A screenshot of a Jupyter Notebook cell. The top part shows a dropdown menu for the class 'Ridge' with an information icon and a help icon. Below the dropdown, the constructor 'Ridge()' is displayed.

Figure 3: Model fitted for Ridge Regression

- **Random Forest Regressor:**

```
: from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

y_pred = model.predict(X_test_scaled)
```

Figure 4: Model fitted for Random Forest Regressor

- **XGBoost:**

```
import xgboost as xgb

model = xgb.XGBRegressor(n_estimators=100, learning_rate=0.05, max_depth=6, random_state=42)

model.fit(X_train, y_train)

y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)
```

Figure 5: Model fitted for XG Boost

8. Model Evaluation

It was performed using parameters like MSE (To quantify average squared errors) and R^2 (To measure the proportion of variance explained)

- Linear Regression: Mean Squared Error (MSE): 49200540.29, R-squared (R^2): 0.9046
- Ridge Regression: Mean Squared Error: 49200538.32, R-squared (R^2): 0.90
- Random Forest Regressor: Mean Squared Error: 7770420.238232188, R-squared (R^2): 0.9849259215366715
- XG Boost: R-squared (R^2): 0.9437, MSE: 29012602.00

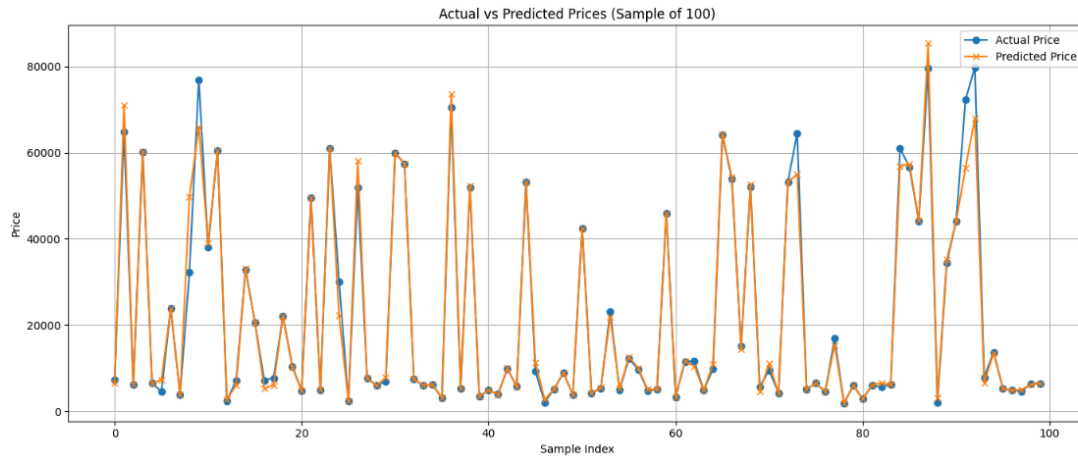


Figure 5: Actual vs Predicted Prices for a sample of 100 values using Random Forest Regressor

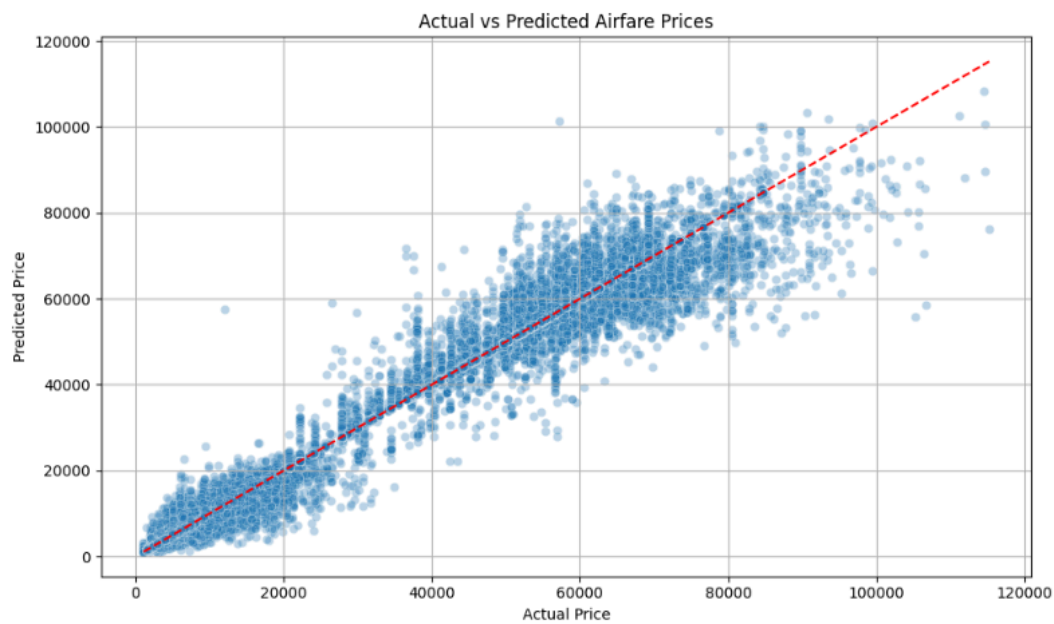


Figure 6: Actual vs Predicted Prices using Random Forest Regressor

9. Feature Importance Analysis:

- **Correlation Analysis:** Calculated correlation between the features

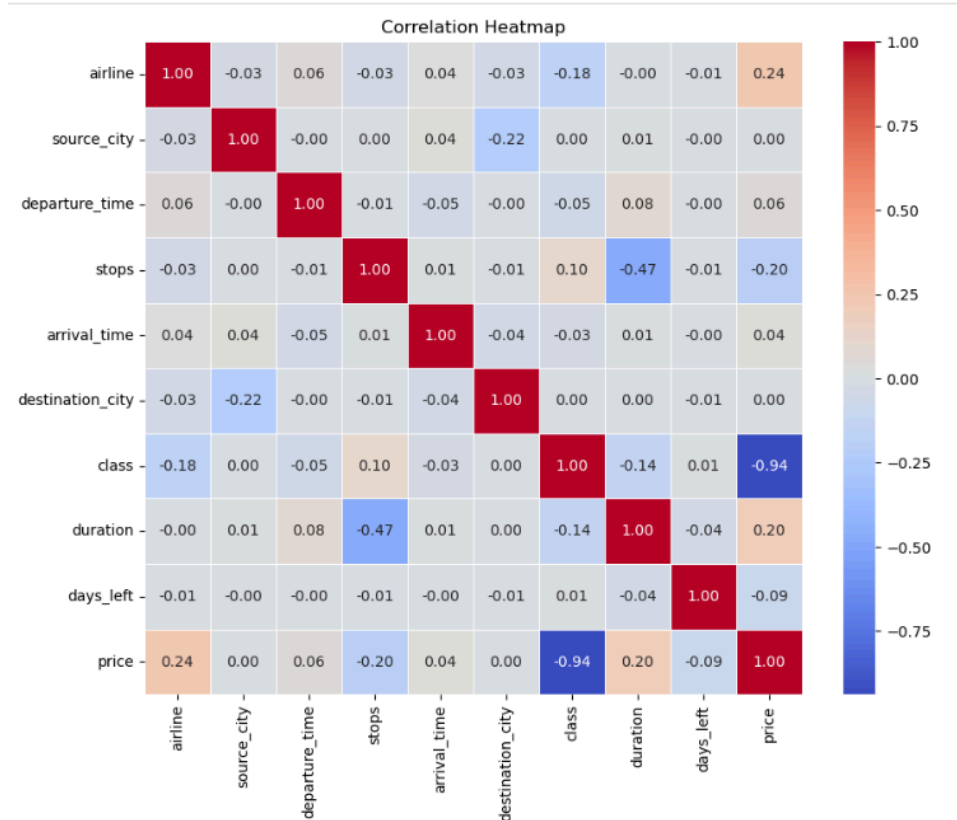


Figure 7: Correlation Heatmap for all features

Key Observations:

- Class shows a strong negative correlation with price (-0.94), indicating that economy class tickets are significantly cheaper than business class tickets.
- Duration has a moderate positive correlation with price (0.20), suggesting that longer flights generally cost more.
- Airline displays a mild positive correlation with price (0.24), implying that pricing may vary slightly depending on the airline.
- Stops has a slight negative correlation with price (-0.20), showing that flights with more stops are generally less expensive.

• Feature Importance Chart:

The feature importance charts show that "class" (economy or business) is by far the most significant factor in predicting flight prices, contributing nearly 90% to the model's

decisions. "Duration" and "days_left" have minor influence, while other features like airline, city, and stops contribute very little. This suggests that ticket class overwhelmingly determines price in this dataset. Other commonly assumed factors, such as airline or number of stops, have minimal impact on the model's predictions.

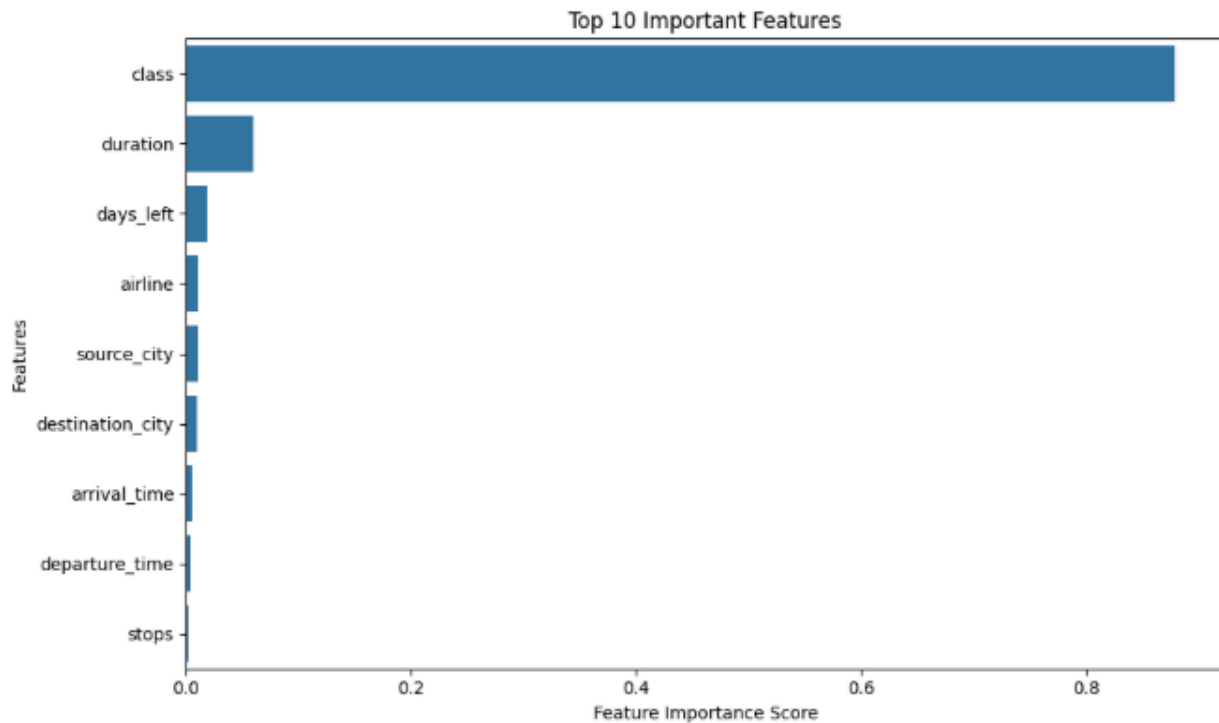


Figure 8: Feature Importance Chart

10. Model Deployment

An interactive **Streamlit** application was developed to enable real-time fare prediction. Users can input flight parameters and instantly receive a predicted price, making the tool practical for consumer decision-making and airline pricing strategy testing.



The image shows a Streamlit web application titled "Airfare Price Predictor". It features a series of input fields for flight details: "Airline" (set to "Air_India"), "Source City" (set to "Chennai"), "Destination City" (set to "Hyderabad"), "Departure Time" (set to "Early_Morning"), "Arrival Time" (set to "Morning"), "Number of Stops" (set to "zero"), and "Ticket Class" (set to "Economy"). Below these is a "Duration of Flight (in hours)" field set to "2.00" with minus and plus buttons. At the bottom is a "Days Left for Journey" slider ranging from 0 to 60, currently set at 5. A "Predict Fare" button is located at the very bottom.

Figure 9: Snippet of the Streamlit app for predicting flight prices based on the model

Results & Analysis

To analyze how well different machine learning models can predict flight ticket prices, we applied several regression techniques. We then compared their performance using key evaluation metrics like Mean Squared Error (MSE) and R^2 score. These metrics helped us understand both the accuracy of the predictions and how well each model could generalize to new data.

- **Linear Regression:**

```
y_pred = model.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Mean Absolute Error:", mean_absolute_error(y_test, y_pred))
print(f"R-squared (R²): {r2:.4f}")

Mean Squared Error (MSE): 49200540.29
Mean Absolute Error: 4624.994868016881
R-squared (R²): 0.9046
```

Figure 10: Snippet of the code and metrics for Linear Regression

With the Linear Regression model, It produced a **Mean Squared Error (MSE) of 49,200,540.9** and an **R² score of 0.91**, which shows a strong linear relationship between the input features and the flight ticket prices. However, the model doesn't capture the more complex, non-linear patterns in the data. As a result, its overall accuracy was low compared to more advanced models.

- **Ridge Regression:**

```
from sklearn.metrics import mean_squared_error

y_pred = ridge.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"Mean Absolute Error:", mean_absolute_error(y_test, y_pred))
print(f"R-squared: {r2:.2f}")

Mean Squared Error: 49200538.32
Mean Absolute Error: 4625.001447540464
R-squared: 0.90
```

Figure 11: Snippet of the code and metrics for Ridge Regression

To reduce the risk of overfitting, we applied Ridge Regression. The results were similar to the Linear Regression model, with an **MSE of 49,200,538.32** and an **R² score of 0.9113**. This is because the dataset had low multicollinearity and followed a mostly linear regression pattern.

- **Random Forest Regressor:**

```
: print("Mean Absolute Error:", mean_absolute_error(y_test, y_pred))
  print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
  print("R2 Score:", r2_score(y_test, y_pred))
```

```
Mean Absolute Error: 1091.2635908437387
Mean Squared Error: 7770420.238232188
R2 Score: 0.9849259215366715
```

Figure 12: Snippet of the code and metrics for Random Forest Regressor

Random Forest Regression performed better than the other previous models. It brought the **MSE down to around 7770420.23** and an impressive **R² score of 0.9848**. It seems to be overfitted ,to check the overfitting condition, performed R² both training and test sets. The R² scores were 0.9975 for training and 0.9844 for testing — pretty close, which suggests **it's not overfitting**. It captured all non-linear relationships and complex interactions between features which helped to boost the accuracy.

- **XG Boost:**

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

r2_test = r2_score(y_test, y_test_pred)

mae_test = mean_absolute_error(y_test, y_test_pred)

mse_test = (mean_squared_error(y_test, y_test_pred))

print(f"R^2: {r2_test:.4f}")
print(f"MAE: {mae_test:.2f}")
print(f"MSE: {mse_test:.2f}")
```

```
R^2: 0.9437
MAE: 3718.45
MSE: 29012602.00
```

Figure 13: Snippet of the code and metrics for XG Boost

XGBoost demonstrated good performance in predicting airfare prices. It achieved an R² score of **0.9437** on the test set, indicating strong generalization without significant overfitting. The Mean Absolute Error (MAE) values were **3718.45**, while the Mean Squared Error (MSE) stood at **29012602**. These results highlight XGBoost's ability to balance bias and variance effectively,

leveraging its boosting strategy to handle feature interactions and improve predictive accuracy over simpler models.

Below is a table comparing the performance of the 4 models

Model	Mean Squared Error (MSE)	R-squared (R ²)
Linear Regression	49,200,540.29	0.9046
Ridge Regression	49,200,538.32	0.9000
Random Forest Regressor	7,770,420.24	0.9849
XGBoost Regressor	29,012,602	0.9437

Table 1: Comparing performance of various models

Brand Perception Insights

Passengers view airlines operating on the Delhi–Mumbai route as very competitive, with all the large carriers (Indigo, SpiceJet, Vistara, Air India, GO_FIRST, AirAsia) having similar direct flight times (2.0–2.5 hours) and similarly matched fares, particularly for last-minute purchases (generally ₹5953–₹5956 for economy). This gives a perception of a commoditized market where convenience, schedule, and price are the primary differentiators, not huge variations in service or experience.

Multiple visualizations were created to gain better insights into the correlations in the data:

- Correlation Heatmap: A heat map of the correlation matrix to see how the numerical features correlate among themselves and with the target variable (price)..
- Actual vs. Predicted Prices: A line plot comparing actual and predicted prices for a random sample of 100 flights, showing the accuracy of the model.
- Price Trend by Class and Days Left: A line graph representing how prices fluctuate as the remaining days to the flight vary, divided by classes.

Conclusion

This project is a success in proving how machine learning methods, particularly the Random Forest Regressor, can be used to forecast airline ticket prices with great accuracy. Through examining important variables like airline, departure time, stops, and days remaining until departure, the model provides useful information on the price dynamics of air travel.

The supporting Streamlit application improves user experience through real-time and intuitive fare forecasting based on chosen parameters. The application can be particularly beneficial for frugal travelers, travel agencies, and aggregators with a focus on optimizing booking choices and enhancing customer satisfaction.

In the future, the project can be extended by adding more real-world variables like fluctuation in demand, weather, and holidays. The model being integrated into travel websites or apps can further enhance its impact by providing dynamic prices and more intelligent travel planning.

References

1. Doganis, R., et al. (2006). *Airline Revenue Management: A Data-Driven Approach*.
2. Grover, P., & Mehta, A. (2017). *Airfare Prediction Using Machine Learning Techniques*.
3. Bhambri, P., & Ratra, S. (2019). *Ensemble Learning Techniques for Airline Ticket Price Prediction*.
4. Arya, K., Mamania, P., & Chuneekar, V. (2021). *Analysis of Flight Fare Detection using Machine Learning*. IJERT
5. Liu, J. (2023). *Feature Correlation Analysis and Comparison of Machine Learning Models for Air Ticket Price Prediction*. Research Gate
6. Upadhye, M., et al. (2024). *Localized Modeling for Airline Price Prediction Using K-Means and Decision Tree Ensemble*. IJCNIS
7. Kaggle Flights Dataset:
[<https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction>]
8. Scikit-learn Documentation: [<https://scikit-learn.org/stable/>]
9. Streamlit Documentation: [<https://streamlit.io/>]