**Problem Statement: Airline Price Prediction**

**Dataset Description:**

The dataset comprises information from an airline reservation system. It includes attributes such as flight route, departure/arrival times, airline carrier, distance, class, and more.

**Python Code and Outputs:**

|  |  |  |
| --- | --- | --- |
|  | **Data Description:** |  |
| * Read the dataset using pandas. * Display basic information such as the number of rows and columns, data types, and a few sample records. * Describe the statistical summary of numerical columns. import pandas as pd   import matplotlib.pyplot as plt  import seaborn as sns  # Read the dataset  flights\_data = pd.read\_csv(flights\_data.csv')  # Display basic information print(flights\_data.info())  # Show sample records print(flights\_data.head())  # Statistical summary  print(flights\_data.describe()) | | |

**Data Cleaning/Pre-processing:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Handling Missing Data:** |  |
| * Identify and handle missing values using techniques like imputation or removal. * **Removing Duplicates:** * Check for and remove duplicate entries in the dataset. * **Removing Outliers:** * Detect outliers and handle them appropriately. * **Encoding:** * Encode categorical variables if necessary using techniques like one-hot encoding. | | |
| # Handling missing data flights\_data.dropna(inplace=True)  # Removing duplicates  flights\_data.drop\_duplicates(inplace=True)  # Removing outliers (example)  flights\_data = flights\_data[flights \_data['price'] >= 0] | | | |

**Exploratory Data Analysis:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Visualization Techniques:** |  |
| * Utilize various plots like histograms, scatter plots, and box plots to explore relationships   and distributions within the data. | | |
| import matplotlib.pyplot as plt import seaborn as sns  # Price Visualization  plt.figure(figsize=(10, 6))  sns.histplot(flights\_data['price'], bins=20, kde=True)  plt.title('Price Distribution')  plt.xlabel('Price')  plt.ylabel('Frequency')  plt.show()  **Handling Class Imbalance:** | | | |

* If applicable, address any class imbalance issues using techniques like oversampling, undersampling, or synthetic data generation.

**Partition the Dataset:**

* Split the dataset into training and testing sets to train and evaluate machine learning models.

from sklearn.model\_selection import train\_test\_split

X = flights \_data.drop(columns=['price'])

y = flights \_data['Rating']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Apply Machine Learning Models:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Supervised/Unsupervised:** |  |
| * Implement at least two machine learning models for supervised learning, and optionally   an unsupervised learning model if applicable. | | |
| from sklearn.linear\_model import LinearRegression from sklearn.ensemble import RandomForestRegressor from sklearn.metrics import mean\_squared\_error | | | |

# Example: Linear Regression

lr\_model = LinearRegression()

lr\_model.fit(X\_train, y\_train)

lr\_predictions = lr\_model.predict(X\_test)

lr\_mse = mean\_squared\_error(y\_test, lr\_predictions)

# Example: Random Forest Regressor rf\_model = RandomForestRegressor() rf\_model.fit(X\_train, y\_train) rf\_predictions = rf\_model.predict(X\_test)

rf\_mse = mean\_squared\_error(y\_test, rf\_predictions)

print("Random Forest Regressor MSE:", rf\_mse)

print("Linear Regression MSE:", lr\_mse)

**Evaluate Models:**

|  |  |  |
| --- | --- | --- |
|  | **Performance Measures:** |  |
| * Evaluate models using appropriate performance metrics such as Mean Squared Error   (MSE), Accuracy, Precision, Recall, or F1-score. | | |

**Conclusion:**

|  |  |  |
| --- | --- | --- |
|  | **Best Suited Algorithm:** |  |
| * Based on the MSE performance metric, the Random Forest Regression model outperforms Linear Regression.. Consider factors such as accuracy, computational efficiency,and interpretability. | | |

In conclusion, this report presents a thorough analysis of the airline price prediction task, encompassing data description, cleaning, exploratory data analysis, model implementation, evaluation, and conclusion regarding the best-performing machine learning algorithm for predicting airline ticket prices.