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