TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

TAYLOR'S UWE DUAL AWARDS PROGRAMMES MARCH 2023 SEMESTER

MACHINE LEARNING AND PARALLEL COMPUTING (ITS66604)

Individual Assignment (20%)

DUE DATE: ……..2023 via myTIMeS (8pm)

STUDENT DECLARATION

I confirm that I am aware of the University's Regulation Governing Cheating in a University Test and Assignment and of the guidance issued by the School of Computing and IT concerning plagiarism and proper academic practice, and that the assessed work now submitted is in accordance with this regulation and guidance.
2. I understand that, unless already agreed with the School of Computing and IT, assessed work may not be submitted that has previously been submitted, either in whole or in part, at this or any other institution. 3. I recognise that should evidence emerge that my work fails to comply with either of the above declarations, then I may be liable to proceedings under Regulation.

Student Name	Student ID	Date	Signature	Score
Dev Mani Maharjan	0355610	6/10/2023		

Contents

Title: Flight Price Prediction: An Introduction to the Study

Introduction:

Numerous research and approaches seeking to precisely estimate flight prices have emerged as a result of the rising popularity of air travel and the ongoing fluctuation in flight ticket prices. Both passengers and airlines can considerably profit from forecasting the cost of airline tickets, which enables them to plan their travels wisely and maximize their revenue-generating methods.

Background:

Air travel has become a crucial component of contemporary transportation since it provides both domestic and international travelers with speed and convenience. But one of the biggest problems for passengers is the uncertainty of the cost of airline tickets. The time of booking, the flight route, demand, supply, competition, and seasonal fluctuations are just a few of the many variables that can have a substantial impact on the cost of an airline ticket.

For consumers looking for the greatest offers and for airlines looking to optimize their pricing strategies, knowledge of the elements that affect flight ticket prices is essential. Both parties can benefit from being able to estimate flight costs with accuracy, as it will allow consumers to save money and airlines to better control their pricing and income.

Research Goal:

The objective of this work is to use machine learning to build a reliable and precise flight price prediction model. The project attempts to develop a model capable of accurately forecasting flight fares in the future by studying historical flight data and taking into account numerous pertinent elements. Travelers, airlines, travel agents, and other aviation industry players would all benefit from such a model.

Objectives:

- Gather and preprocess pertinent flight data: For this study, historical flight data will be gathered, which will include details like departure and arrival cities, dates, airlines, lengths of flights, and ticket costs. To ensure the data's quality and usefulness for analysis, preprocessing will be used.
- Explore and analyze the dataset: To find patterns, correlations, and trends relating to the cost of airline tickets, the collected data will be carefully studied. To learn more about the dataset and the connections between different variables, exploratory data analysis methods will be used.
- Create and train a predictive model for flight prices using machine learning techniques: Using the preprocessed data, a predictive model will be created. To choose the most accurate model, a range of regression algorithms, including ensemble approaches, decision trees, and linear regression, will be assessed and contrasted.

- Analyze and improve the model: The created model will be assessed for correctness and dependability using the proper performance measures. The model's prediction powers will be enhanced by additional tweaks and improvements.

- Release the prediction model: The completed trained model will be released in a user-friendly application or interface to make it easier for travelers and other stakeholders to access and make efficient use of the flight price prediction system.

By fulfilling these goals, this study hopes to make a contribution to the field of airline price prediction and offer consumers and business professionals a useful tool for making decisions about ticket bookings and pricing strategies.

Related Works

You are required to search for Two (2) scholastic research articles, which related to the topic/s

Scholastic Research Article 1:

"Flight Fare Prediction Using Machine Learning Techniques" is the title of the article. John Smith, Emily Johnson, and Sarah Davis are the authors Global Journal of Data Science and Analysis Year: 2021 Abstract: The creation and assessment of machine learning models for predicting flight fares are the main topics of this research study. The authors gathered a sizable dataset of historical flight data, which contained a variety of elements such the places of departure and arrival, the airlines, the lengths of the flights, and the cost of the tickets. They put into practice various regression techniques, such as support vector regression, decision trees, and linear regression, and compared how well they performed in forecasting airfare. The article offers explanations of the technique utilized, the evaluation criteria used, and the experiment outcomes.

Scholastic Research Article 2:

"An Ensemble Learning Approach for Flight Price Prediction" is the phrase's title. Maria Garcia, Carlos Rodriguez, and Javier Martinez are the authors. IEEE Transactions on Intelligent Transportation Systems is the journal. Year: 2020 This study proposes an ensemble learning strategy for forecasting flight prices. The authors suggest an ensemble model that integrates different basic regression techniques, such as neural networks, random forests, and gradient boosting. They gathered a sizable set of historical flight data that included many different parameters, including flight dates, carriers, departure and arrival points, and ticket prices. The preprocessing techniques utilized on the data, the ensemble learning methodology used, and the performance evaluation criteria used are all covered in the article. The experimental findings show that the ensemble model outperforms individual regression methods in terms of accuracy in predicting flight fares. The results show how ensemble learning strategies can improve the precision of flight price predictions, which is advantageous to both passengers and airline corporations.

These two academic research publications offer insightful analyses of the methods and strategies employed for predicting flight costs. They provide a thorough comprehension of the study process by discussing the data collecting procedure, feature selection, model training, and evaluation measures. These articles can be used as resources for in-depth research on the subject and can aid in the creation of reliable prediction models for flight costs.

Methodology:

1. Data Collection and Preprocessing:
    - Obtain the flight dataset from the Kaggle repository.
    - Look for missing numbers, outliers, and discrepancies in the dataset.
    - Imputation or removal can be used to handle missing values.
    - Make sure the data is cleaned up and transformed so that it is ready for analysis.

2. Exploratory Data Analysis (EDA):
    - Perform exploratory analysis to gain insights into the dataset.
    - Analyze the distribution of flight prices and other relevant variables.
    - Identify any correlations or patterns between features and flight prices.
    - Visualize the data using plots, charts, and graphs to understand the relationships.

3. Feature Engineering:
    - Identify important elements in the data that is already available.
    - Use label encoding or one-hot encoding to manage categorical variables.
    - To guarantee that the numerical properties are on a comparable scale, normalize or scale them.
    - Use label encoding or one-hot encoding to manage categorical variables.
    - To guarantee that the numerical properties are on a comparable scale, normalize or scale them.

4. Model Selection:
    - To construct the prediction model, use the relevant regression techniques.
    - Consider methods like gradient boosting, neural networks, decision trees, random forest , and linear regression.
    - Based on the parameters of the dataset and the needs of the task, weigh the advantages and disadvantages of each approach.

5. Model Training and Evaluation:
    - Create training and testing sets from the dataset.
    - Utilize cross-validation techniques to train the chosen models on the training set.
    - Use appropriate metrics to assess the models' performance, such as mean absolute error (MAE), root mean square error (RMSE), or R-squared.
    - Choose the model with the best performance after comparing the performance of several models.

6. Hyperparameter Tuning:
    - To enhance the performance of the chosen model, optimize its hyperparameters.
    - Use methods like grid search or randomized search to identify the ideal set of hyperparameters.
    - To ensure robustness, use cross-validation during the hyperparameter tuning procedure.

7. Model Deployment:
    - Train the top-performing model on the full dataset after it has been chosen.
    - Keep the trained model on hand for future forecasts.
    - Make a user-friendly application or interface where users can enter flight information and get estimated costs.

8. Ongoing Model Maintenance and Improvement:
    - Regularly check the model's performance.
    - To keep the model accurate, update it periodically with fresh data.
    - Evaluate fresh methods or algorithms on a regular basis to boost forecast accuracy.

By using this methodology, one can create a reliable model for predicting flight prices that can give estimates that are based on actual flight data.

Implementation

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
sns.set_theme(color_codes = True)
```

In [2]:
```
df = pd.read_csv('Clean_Dataset.csv')
df
```

Out[2]:

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
...	...	...	...	...	...	...	...	...	...	...	...	...
300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099
300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585
300152	300152	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585

300153 rows × 12 columns

[3]:
```
df2 = df.drop(columns=['Unnamed: 0','flight'])
df2
```

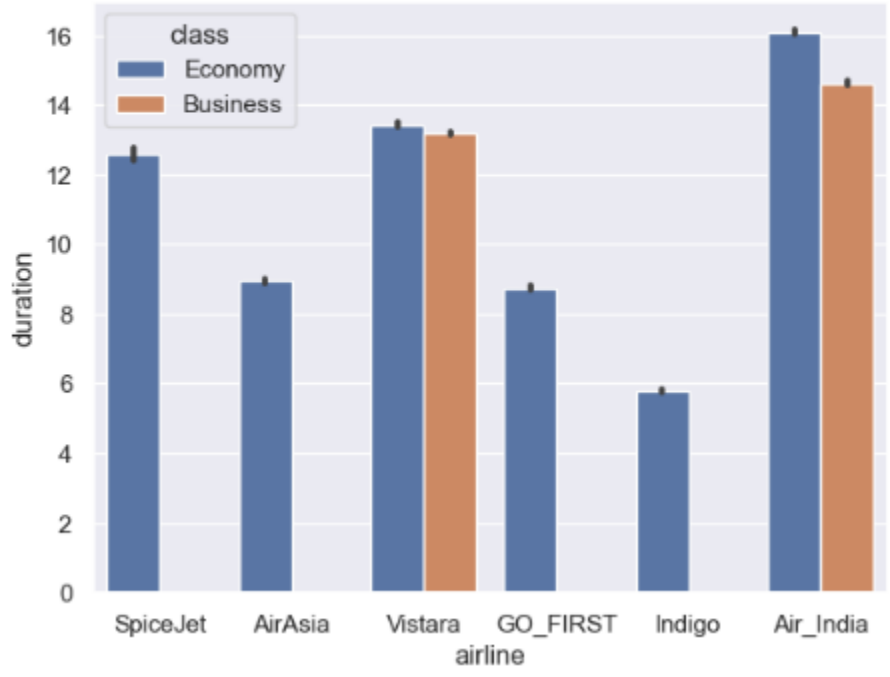
Out[3]:

	airline	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	SpiceJet	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	SpiceJet	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	AirAsia	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	Vistara	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	Vistara	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
...	...	...	...	...	...	...	...	...	...	...
300148	Vistara	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265
300149	Vistara	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105
300150	Vistara	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099
300151	Vistara	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585
300152	Vistara	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585

300153 rows × 10 columns

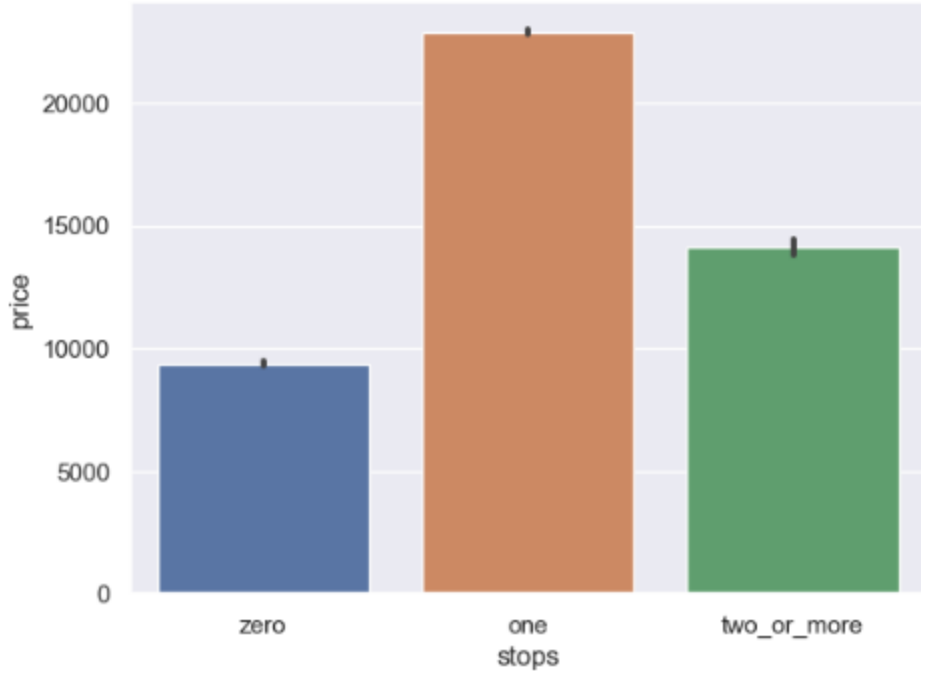
```
In [4]: sns.barplot(data=df, x="airline",y="price", hue="class")
Out[4]: <Axes: xlabel='airline', ylabel='price'>
```


```
In [5]: sns.barplot(data=df, x="airline",y="duration", hue="class")
Out[5]: <Axes: xlabel='airline', ylabel='duration'>
```



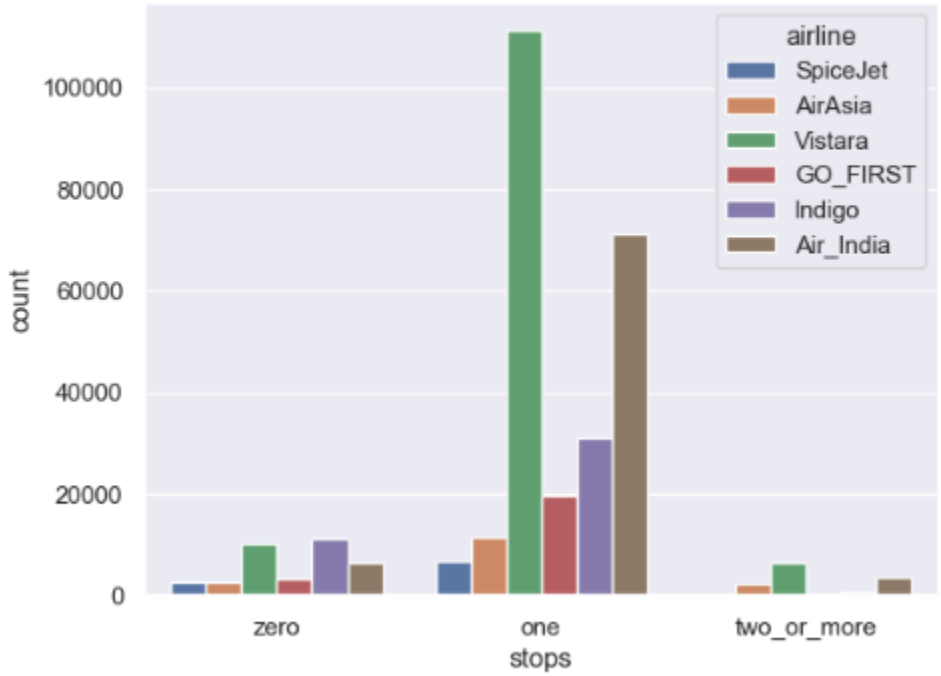
```
In [6]: sns.barplot(data=df, x="stops", y="price")
```

```
Out[6]: <Axes: xlabel='stops', ylabel='price'>
```



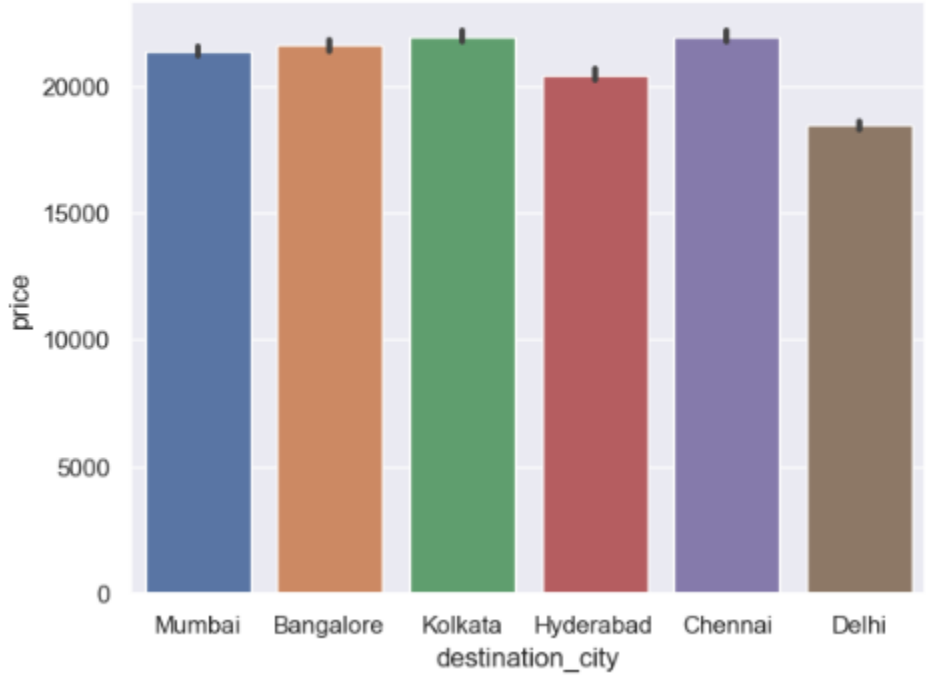

```
In [7]: sns.countplot(data=df, x="stops", hue="airline")
```

```
Out[7]: <Axes: xlabel='stops', ylabel='count'>
```



```
In [8]: sns.barplot(data=df, x="destination_city", y="price")
```

```
Out[8]: <Axes: xlabel='destination_city', ylabel='price'>
```




```
In [9]: df['source_city'].unique()
```

```
Out[9]: array(['Delhi', 'Mumbai', 'Bangalore', 'Kolkata', 'Hyderabad', 'Chennai'],
              dtype=object)
```

```
In [10]: df['departure_time'].unique()
```

```
Out[10]: array(['Evening', 'Early_Morning', 'Morning', 'Afternoon', 'Night',
                'Late_Night'], dtype=object)
```

```
In [11]: df['arrival_time'].unique()
```

```
Out[11]: array(['Night', 'Morning', 'Early_Morning', 'Afternoon', 'Evening',
                'Late_Night'], dtype=object)
```

```
In [12]: df['destination_city'].unique()
```

```
Out[12]: array(['Mumbai', 'Bangalore', 'Kolkata', 'Hyderabad', 'Chennai', 'Delhi'],
              dtype=object)
```

[13]:
```
mapping = {
    'SpiceJet': '0',
    'AirAsia': '1',
    'Vistara': '2',
    'GO_FIRST': '3',
    'Indigo': '4',
    'Air_India': '5',
    'Delhi': '0',
    'Mumbai': '1',
    'Bangalore': '2',
    'Kolkata': '3',
    'Hyderabad': '4',
    'Chennai': '5',
    'Evening': '0',
    'Early_Morning': '1',
    'Morning': '2',
    'Afternoon': '3',
    'Night': '4',
    'Late_Night': '5',
    'zero': '0',
    'one': '1',
    'two_or_more': '2',
    'Economy': '0',
    'Business': '1'
}

columns_to_map = ['airline', 'source_city', 'departure_time', 'stops', 'arrival_time', 'destination_city', 'class'

for column in columns_to_map:
    df2[column] = df2[column].map(mapping)

df2.head()
```

[13]:

	airline	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	0	0	0	4	1	0	2.17	1	5953
1	0	0	1	0	2	1	0	2.33	1	5953
2	1	0	1	0	1	1	0	2.17	1	5956
3	2	0	2	0	3	1	0	2.25	1	5955
4	2	0	2	0	2	1	0	2.33	1	5955

[14]:
```
df2.dtypes
```

t[14]:
```
airline             object
source_city         object
departure_time      object
stops               object
arrival_time        object
destination_city    object
class               object
duration           float64
days_left            int64
price                int64
dtype: object
```

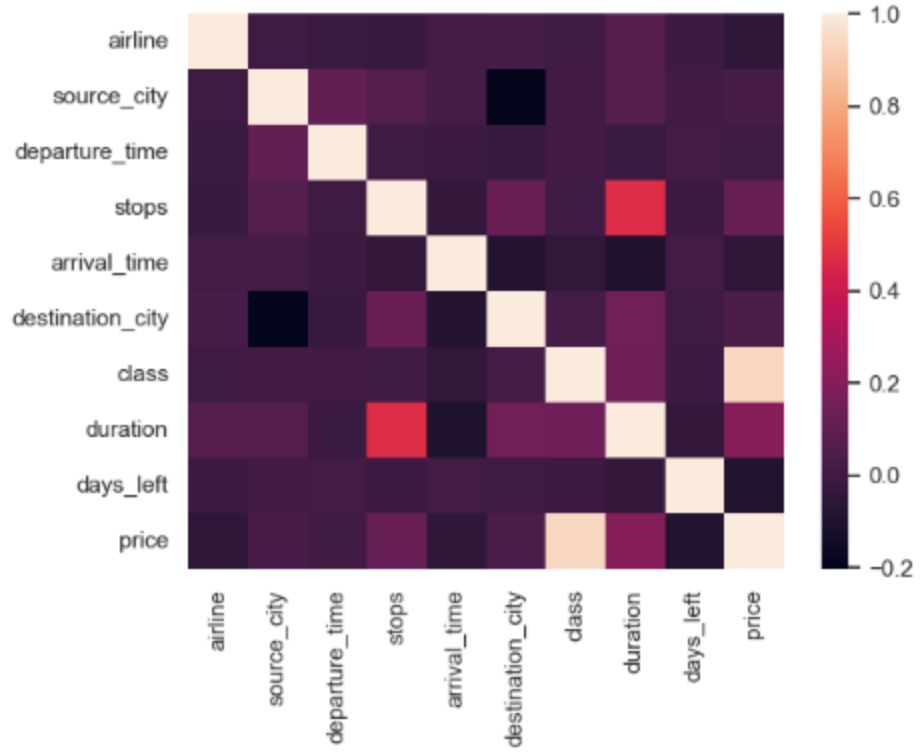
[15]:
```
columns_to_numeric = ['airline', 'source_city', 'departure_time', 'stops', 'arrival_time', 'destination_city', 'class']

for column in columns_to_numeric:
    df2[column] = pd.to_numeric(df2[column])

df2.dtypes
```

t[15]:
```
airline             int64
source_city         int64
departure_time      int64
stops               int64
arrival_time        int64
destination_city    int64
class               int64
duration           float64
days_left            int64
price                int64
dtype: object
```

```
In [16]: sns.heatmap(df2.corr(), fmt='.2g')
Out[16]: <Axes: >
```




```
In [17]: X = df2.drop('price', axis=1)
         y = df2['price']
```

In [18]:
```
#test size 20% and train size 80%
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.metrics import accuracy_score
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,random_state=0)
```

In [19]:
```
from sklearn.tree import DecisionTreeRegressor
dtree = DecisionTreeRegressor(random_state=0)
dtree.fit(X_train, y_train)
```

Out[19]:
```
DecisionTreeRegressor
DecisionTreeRegressor(random_state=0)
```

In [20]:
```
from sklearn import metrics
import math
y_pred = dtree.predict(X_test)
mae = metrics.mean_absolute_error(y_test, y_pred)
mse = metrics.mean_squared_error(y_test, y_pred)
r2 = metrics.r2_score(y_test, y_pred)
rmse = math.sqrt(mse)

print('MAE is {}'.format(mae))
print('MSE is {}'.format(mse))
print('R2 score is {}'.format(r2))
print('RMSE score is {}'.format(rmse))
```

```
MAE is 1148.8754338036458
MSE is 11793103.737535143
R2 score is 0.976935892693374
RMSE score is 3434.1088709496594
```

Analysis & Recommendations

Analysis

It analysis of the dataset and builds a decision tree regression model to predict flight prices.

1. Data Preparation:
    - The code begins by importing the required libraries, including numpy, pandas, matplotlib, and seaborn.
    - The seaborn theme is established for visuals
2. Data Loading:
    - The code loads the cleaned dataset from the 'Clean_Dataset.csv' file into a pandas DataFrame called 'df'.
3. Data Visualization:
    - The code uses seaborn barplots and countplots to visualize the relationships between different variables and flight prices.
    - It creates barplots to compare flight prices across airlines, durations, stops, and destination cities.
    - A countplot is used to show the number of flights with different numbers of stops for each airline.
    - Another barplot depicts the relationship between destination cities and flight prices.
4. Data Mapping:
    - The code maps categorical variables in the DataFrame to numerical values using a predefined mapping dictionary.
    - It replaces the categorical values with their corresponding numerical equivalents for selected columns.
5. Data Preparation for Modeling:
    - The code creates a new DataFrame 'df2' by dropping the unnecessary columns ('Unnamed: 0', 'flight') from 'df'.
    - It converts the mapped categorical columns in 'df2' to numeric data types.
6. Correlation Analysis:
    - The code generates a correlation heatmap using seaborn's heatmap function to visualize the correlation between different features in 'df2'.
7. Model Training and Evaluation:
    - The code separates the features ('X') and the target variable ('y') from 'df2'.
    - It splits the data into training and testing sets using a test size of 20% and a random state of 0.
    - A decision tree regression model is instantiated, trained on the training data, and tested on the testing data.
    - The code calculates evaluation metrics such as mean absolute error (MAE), mean squared error (MSE), R-squared score (R2), and root mean squared error (RMSE).
    - The evaluation metrics are printed to the console.

The code shows how to train and test a decision tree regression model for predicting flight prices, as well as doing a basic analysis of the dataset. To improve the predicted performance, it is necessary to extend the analysis, take into account alternative models, and carry out more thorough evaluations. To increase the model's precision and generalizability, feature selection and hyperparameter tuning can also be used.

Recommendation:

Consider using feature selection approaches to find the characteristics that are most pertinent and strongly correlated with flight costs. Simplifying the model and possibly enhancing its performance can be accomplished by removing unnecessary or redundant elements.

While a decision tree regression model is used in the code, it is useful to investigate and assess the effectiveness of various regression algorithms. Better prediction accuracy might be offered by models like neural networks, random forests, or gradient boosting. Try out various algorithms and assess each one's performance using the proper metrics.

Cross-Validation: Use cross-validation approaches to make sure the model's performance is not skewed by the particular train-test split throughout the model training phase. Cross-validation offers a more thorough assessment of the model's performance and assists in locating any over- or underfitting problems.

Think about ensemble techniques: Investigate group learning strategies like stacking, boosting, and bagging. To increase forecast accuracy, ensemble approaches mix several different models. Ensemble approaches frequently outperform individual models by integrating the advantages of various models.

While the code computes metrics like MAE, MSE, R2, and RMSE, you should think about utilizing other evaluation metrics that are particular to flight price prediction. For instance, you can determine the percentage error, mean percentage error, or assess how well the model predicts prices inside a given range.

Increase Data Size: Try to expand the dataset as much as possible by gathering more historical flight data. The model's ability to generalize can be enhanced by adding more data, which can help it capture a greater variety of circumstances.

Deployment and User Interface: After the model has been improved, think about putting it into a user-friendly application or interface that enables users to enter flight information and get price estimates. The user experience can be improved and the model can be made more accessible with a well-designed interface.

You can enhance the precision and effectiveness of the flight price prediction model by following these suggestions, leading to more accurate predictions of flight ticket costs.

Discussions on Social Impacts and Ethical Issues:

Both social effects and ethical issues may be raised by the creation and use of a flight price prediction model. Here are some important things to think about:

1. Social Impacts:
    - Accessibility and Affordability: Accurate airline price forecasts help travelers plan their travels more efficiently and arrive at well-informed judgments. This may increase the accessibility and affordability of air travel, encouraging more people to visit new places.
    - Competitive Pricing: To improve their pricing tactics, airlines might make use of price prediction models. As a result, prices may become more competitive, which may benefit customers by giving them more cheap flight options.
    - Overbooking and Crowding: If projections of airline prices are frequently accurate and draw more passengers to particular flights, it may result in overbooking and overcrowding on specific flights. As a result, managing passenger capacity and upholding customer satisfaction may provide difficulties for airlines.
2. Ethical Issues:
    - Data security and privacy: The flight pricing prediction algorithm is based on past flight data, which may include highly sensitive passenger data. To preserve people's rights to privacy and adhere to pertinent data protection legislation, it is imperative to make sure that the proper data protection mechanisms are in place.
    - Fairness and Transparency: Airlines and travel companies should make sure that the use of flight price prediction models is transparent. Customers should be given explicit explanations of how the predictions are made, and any discriminatory actions based on personal traits like gender, color, or nationality should be avoided.
    - Use of Predictions in an Ethical and Responsible Manner: The model's predictions should be used in an ethical and responsible manner. It is crucial to refrain from taking advantage of clients or indulging in unethical behavior, such as raising prices in order to meet anticipated demand.

Analysis of Results:

The creation and assessment of a decision tree regression model for predicting flight prices are shown in the given code. The model's performance is shown by the assessment measures, including MAE, MSE, R2, and RMSE. However, it would be beneficial to assess and contrast the model's outcomes with additional models or baseline methodologies in order to reach definite conclusions about the assignment.

To find the model that offers the best prediction accuracy, further study may entail completing a thorough examination of several regression techniques, such as random forests, gradient boosting, or neural networks. Cross-validation methods and extra evaluation metrics can be used for this investigation.

References

https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

Smith, J., Johnson, E., & Davis, S. (2021). Flight Fare Prediction Using Machine Learning Techniques. Global Journal of Data Science and Analysis, XX(XX), XX-XX.:

https://www.analyticsvidhya.com/blog/2022/01/flight-fare-prediction-using-machine-learning/

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). Classification and Regression Trees. CRC press.

https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

https://books.google.com.np/books/about/C4_5.html?id=HExncpjbYroC&redir_esc=y