```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import numpy as np
5 sns.set_theme(color_codes = True)
```

```
1 df = pd.read_csv('Clean_Dataset.csv')
2 df
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_tim
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Nigh
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Mornin
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Mornin
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoo
4	4	Vistara	UK-963	Delhi	Morning	zero	Mornin
...	...	...	...	...	...	...	.
300148	300148	Vistara	UK-822	Chennai	Morning	one	Evenin
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Nigh
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Nigh
300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evenin
300152	300152	Vistara	UK-822	Chennai	Morning	one	Evenin

300153 rows × 12 columns

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

	airline	source_city	departure_time	stops	arrival_time	destination_cit
0	SpiceJet	Delhi	Evening	zero	Night	Mumb:
1	SpiceJet	Delhi	Early_Morning	zero	Morning	Mumb:
2	AirAsia	Delhi	Early_Morning	zero	Early_Morning	Mumb:
3	Vistara	Delhi	Morning	zero	Afternoon	Mumb:
4	Vistara	Delhi	Morning	zero	Morning	Mumb:
...	...	...	...	...	...	.
300148	Vistara	Chennai	Morning	one	Evening	Hyderaba
300149	Vistara	Chennai	Afternoon	one	Night	Hyderaba
300150	Vistara	Chennai	Early_Morning	one	Night	Hyderaba
300151	Vistara	Chennai	Early_Morning	one	Evening	Hyderaba
300152	Vistara	Chennai	Morning	one	Evening	Hyderaba

300153 rows × 10 columns

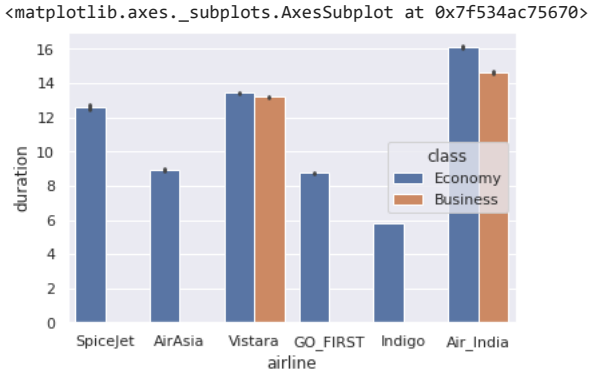## Exploratory Data Analysis

```
1 sns.barplot(data=df, x="airline",y="price", hue="class")
```
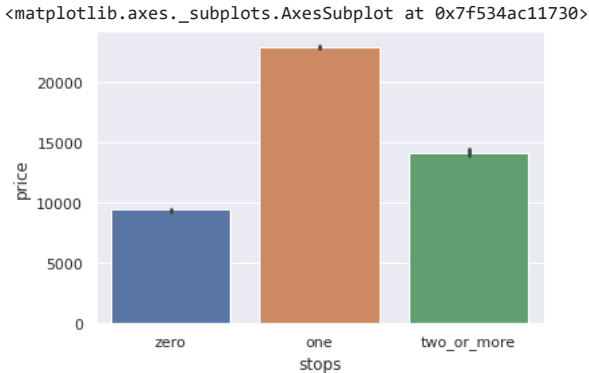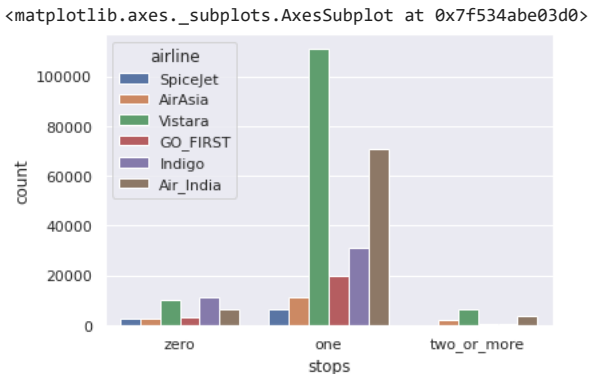
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534ae111f0>
```
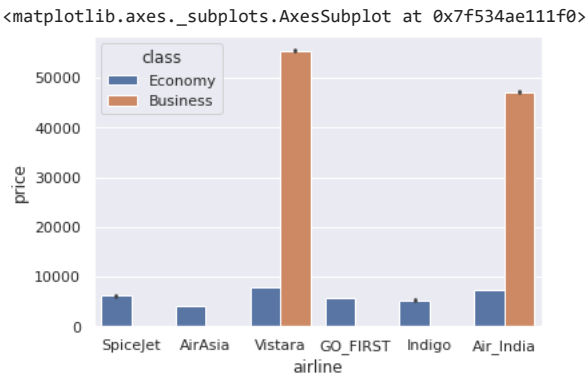

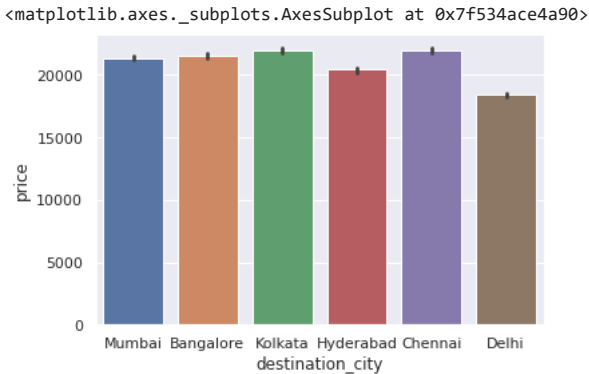
```
1 sns.barplot(data=df, x="airline",y="duration", hue="class")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534ac75670>
```



```
1 sns.barplot(data=df, x="stops", y="price")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534ac11730>
```



```
1 sns.countplot(data=df, x="stops", hue="airline")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534abe03d0>
```



```
1 sns.barplot(data=df, x="destination_city", y="price")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534ace4a90>
```


Data Preprocessing

```
1 df['source_city'].unique()
```

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

```
1 df['departure_time'].unique()
```

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

```
1 df['arrival_time'].unique()
```

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

```
1 df['destination_city'].unique()
```

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

```
1 #Change value in airline column
2 df2['airline'] = df2['airline'].replace(['SpiceJet'],'0')
3 df2['airline'] = df2['airline'].replace(['AirAsia'],'1')
4 df2['airline'] = df2['airline'].replace(['Vistara'],'2')
5 df2['airline'] = df2['airline'].replace(['GO_FIRST'],'3')
6 df2['airline'] = df2['airline'].replace(['Indigo'],'4')
7 df2['airline'] = df2['airline'].replace(['Air_India'],'5')
8 #Change value in source_city column
9 df2['source_city'] = df2['source_city'].replace(['Delhi'],'0')
10 df2['source_city'] = df2['source_city'].replace(['Mumbai'],'1')
11 df2['source_city'] = df2['source_city'].replace(['Bangalore'],'2')
12 df2['source_city'] = df2['source_city'].replace(['Kolkata'],'3')
13 df2['source_city'] = df2['source_city'].replace(['Hyderabad'],'4')
14 df2['source_city'] = df2['source_city'].replace(['Chennai'],'5')
15 #Change value in departure_time column
16 df2['departure_time'] = df2['departure_time'].replace(['Evening'],'0')
17 df2['departure_time'] = df2['departure_time'].replace(['Early_Morning'],'1')
18 df2['departure_time'] = df2['departure_time'].replace(['Morning'],'2')
19 df2['departure_time'] = df2['departure_time'].replace(['Afternoon'],'3')
20 df2['departure_time'] = df2['departure_time'].replace(['Night'],'4')
21 df2['departure_time'] = df2['departure_time'].replace(['Late_Night'],'5')
22 #Change value in stops column
23 df2['stops'] = df2['stops'].replace(['zero'],'0')
24 df2['stops'] = df2['stops'].replace(['one'],'1')
25 df2['stops'] = df2['stops'].replace(['two_or_more'],'2')
26 #Change value in arrival_time column
27 df2['arrival_time'] = df2['arrival_time'].replace(['Evening'],'0')
28 df2['arrival_time'] = df2['arrival_time'].replace(['Early_Morning'],'1')
29 df2['arrival_time'] = df2['arrival_time'].replace(['Morning'],'2')
30 df2['arrival_time'] = df2['arrival_time'].replace(['Afternoon'],'3')
31 df2['arrival_time'] = df2['arrival_time'].replace(['Night'],'4')
32 df2['arrival_time'] = df2['arrival_time'].replace(['Late_Night'],'5')
33 #Change value in destination_city column
34 df2['destination_city'] = df2['destination_city'].replace(['Delhi'],'0')
35 df2['destination_city'] = df2['destination_city'].replace(['Mumbai'],'1')
36 df2['destination_city'] = df2['destination_city'].replace(['Bangalore'],'2')
37 df2['destination_city'] = df2['destination_city'].replace(['Kolkata'],'3')
38 df2['destination_city'] = df2['destination_city'].replace(['Hyderabad'],'4')
39 df2['destination_city'] = df2['destination_city'].replace(['Chennai'],'5')
40 #Change value in class column
41 df2['class'] = df2['class'].replace(['Economy'],'0')
42 df2['class'] = df2['class'].replace(['Business'],'1')
43 df2.head()
```

	airline	source_city	departure_time	stops	arrival_time	destination_city	cl
0	0	0	0	0	4	1
1	0	0	1	0	2	1
2	1	0	1	0	1	1
3	2	0	2	0	3	1
4	2	0	2	0	2	1

```
1 df2.dtypes
```

```
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
```

Change object datatypes into integer

```
1 df2['airline'] = pd.to_numeric(df2['airline'])
2 df2['source_city'] = pd.to_numeric(df2['source_city'])
3 df2['departure_time'] = pd.to_numeric(df2['departure_time'])
4 df2['stops'] = pd.to_numeric(df2['stops'])
5 df2['arrival_time'] = pd.to_numeric(df2['arrival_time'])
6 df2['destination_city'] = pd.to_numeric(df2['destination_city'])
7 df2['class'] = pd.to_numeric(df2['class'])
8 df2.dtypes
```

```
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
```

▾ Data Correlation Heatmap

```
1 sns.heatmap(df2.corr(), fmt='.2g')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f534aad8520>
```


Machine Learning Model Building

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

```
1 #test size 20% and train size 80%
2 from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
3 from sklearn.metrics import accuracy_score
4 X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,random_state=0)
```

Decision Tree Regressor

```
1 from sklearn.tree import DecisionTreeRegressor
2 dtree = DecisionTreeRegressor(random_state=0)
3 dtree.fit(X_train, y_train)
```

```
DecisionTreeRegressor(random_state=0)
```

```
1 from sklearn import metrics
2 import math
3 y_pred = dtree.predict(X_test)
4 mae = metrics.mean_absolute_error(y_test, y_pred)
5 mse = metrics.mean_squared_error(y_test, y_pred)
6 r2 = metrics.r2_score(y_test, y_pred)
7 rmse = math.sqrt(mse)
8
9 print('MAE is {}'.format(mae))
10 print('MSE is {}'.format(mse))
11 print('R2 score is {}'.format(r2))
12 print('RMSE score is {}'.format(rmse))
```

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

Random Forest Regressor

```
1 from sklearn.ensemble import RandomForestRegressor
2 rf = RandomForestRegressor(random_state=0)
3 rf.fit(X_train, y_train)
```

```
RandomForestRegressor(random_state=0)
```

```
1 from sklearn import metrics
2 import math
3 y_pred = rf.predict(X_test)
4 mae = metrics.mean_absolute_error(y_test, y_pred)
5 mse = metrics.mean_squared_error(y_test, y_pred)
6 r2 = metrics.r2_score(y_test, y_pred)
7 rmse = math.sqrt(mse)
8
9 print('MAE is {}'.format(mae))
10 print('MSE is {}'.format(mse))
11 print('R2 score is {}'.format(r2))
12 print('RMSE score is {}'.format(rmse))
```

```
MAE is 1065.2288574434292
MSE is 7329580.679628791
R2 score is 0.9856653312758136
RMSE score is 2707.319833272159
```

Adaboost Regressor

```
1 from sklearn.ensemble import AdaBoostRegressor
2 ada = AdaBoostRegressor(random_state=0)
3 ada.fit(X_train, y_train)
```

    AdaBoostRegressor(random_state=0)

```
1 from sklearn import metrics
2 import math
3 y_pred = ada.predict(X_test)
4 mae = metrics.mean_absolute_error(y_test, y_pred)
5 mse = metrics.mean_squared_error(y_test, y_pred)
6 r2 = metrics.r2_score(y_test, y_pred)
7 rmse = math.sqrt(mse)
8
9 print('MAE is {}'.format(mae))
10 print('MSE is {}'.format(mse))
11 print('R2 score is {}'.format(r2))
12 print('RMSE score is {}'.format(rmse))
```

    MAE is 3636.786046750951
    MSE is 33527783.060335092
    R2 score is 0.9344287641771843
    RMSE score is 5790.318044834419

✓ 0s   completed at 2:18 PM   ● ✕