Flight dataset

August 11, 2024

1 Flight Price Prediction

[109]:
```
# importing necessary libraries
import numpy as np
import pandas as pd
import csv
import seaborn as sns
import matplotlib.pyplot as plt
```

1.1 Loading the data

[110]:
```
Flight_price = pd.read_csv("C:/Users/deeps/OneDrive/Documents/WEBSTER/DATASET/
 ↪Excel/Clean_Dataset_Flight.csv")
```

[111]:
```
Flight_price
```

[111]:

	S.No.	airline	flight	source_city	departure_time	stops	\
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	
3	3	Vistara	UK-995	Delhi	Morning	zero	
4	4	Vistara	UK-963	Delhi	Morning	zero	
…	…	…	…	…	…	…	
300148	300148	Vistara	UK-822	Chennai	Morning	one	
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	
300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	
300152	300152	Vistara	UK-822	Chennai	Morning	one	

	arrival_time	destination_city	class	duration	days_left	\
0	Night	Mumbai	Economy	2.17	1	
1	Morning	Mumbai	Economy	2.33	1	
2	Early_Morning	Mumbai	Economy	2.17	1	
3	Afternoon	Mumbai	Economy	2.25	1	
4	Morning	Mumbai	Economy	2.33	1	
…	…	…	…	…	…	
300148	Evening	Hyderabad	Business	10.08	49	

```
300149          Night        Hyderabad  Business     10.42         49
300150          Night        Hyderabad  Business     13.83         49
300151        Evening        Hyderabad  Business     10.00         49
300152        Evening        Hyderabad  Business     10.08         49

        price (in Rs)
0                5953
1                5953
2                5956
3                5955
4                5955
…                …
300148          69265
300149          77105
300150          79099
300151          81585
300152          81585

[300153 rows x 12 columns]
```

1.2 Data Exploration

[112]:
```
Flight_price.shape
```

[112]:
```
(300153, 12)
```

[113]:
```
Flight_price.head()
```

[113]:
```
   S.No.   airline   flight source_city departure_time stops   arrival_time  \
0      0  SpiceJet  SG-8709       Delhi        Evening  zero          Night
1      1  SpiceJet  SG-8157       Delhi  Early_Morning  zero        Morning
2      2   AirAsia   I5-764       Delhi  Early_Morning  zero  Early_Morning
3      3   Vistara   UK-995       Delhi        Morning  zero      Afternoon
4      4   Vistara   UK-963       Delhi        Morning  zero        Morning

  destination_city    class  duration  days_left  price (in Rs)
0           Mumbai  Economy      2.17          1           5953
1           Mumbai  Economy      2.33          1           5953
2           Mumbai  Economy      2.17          1           5956
3           Mumbai  Economy      2.25          1           5955
4           Mumbai  Economy      2.33          1           5955
```

[114]:
```
Flight_price.columns
```

[114]:
```
Index(['S.No.', 'airline', 'flight', 'source_city', 'departure_time', 'stops',
       'arrival_time', 'destination_city', 'class', 'duration', 'days_left',
       'price (in Rs)'],
```

```
      dtype='object')
```

[115]:
```
Flight_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   S.No.             300153 non-null  int64  
 1   airline           300153 non-null  object 
 2   flight            300153 non-null  object 
 3   source_city       300153 non-null  object 
 4   departure_time    300153 non-null  object 
 5   stops             300153 non-null  object 
 6   arrival_time      300153 non-null  object 
 7   destination_city  300153 non-null  object 
 8   class             300153 non-null  object 
 9   duration          300153 non-null  float64
 10  days_left         300153 non-null  int64  
 11  price (in Rs)     300153 non-null  int64  
dtypes: float64(1), int64(3), object(8)
memory usage: 27.5+ MB
```

[116]:
```
Flight_price.duplicated().sum()
```

[116]:
```
0
```

[117]:
```
Flight_price.isnull()
```

[117]:

	S.No.	airline	flight	source_city	departure_time	stops	\
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
…	…	…	…	…	…	…	
300148	False	False	False	False	False	False	
300149	False	False	False	False	False	False	
300150	False	False	False	False	False	False	
300151	False	False	False	False	False	False	
300152	False	False	False	False	False	False	

	arrival_time	destination_city	class	duration	days_left	\
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	

```
4              False             False  False     False      False
…                …                 …      …         …          …
300148         False             False  False     False      False
300149         False             False  False     False      False
300150         False             False  False     False      False
300151         False             False  False     False      False
300152         False             False  False     False      False

        price (in Rs)
0               False
1               False
2               False
3               False
4               False
…                 …
300148          False
300149          False
300150          False
300151          False
300152          False

[300153 rows x 12 columns]
```

[118]:
```
Flight_price.isnull().sum()
#There are no duplicates and null values present in the given dataset.
#Each record is an unique flight plan of its own without the key as well, maybe␣
 ↪flight codes are making them unqiue. Lets check that.
```

[118]:
```
S.No.               0
airline             0
flight              0
source_city         0
departure_time      0
stops               0
arrival_time        0
destination_city    0
class               0
duration            0
days_left           0
price (in Rs)       0
dtype: int64
```

[119]:
```
Flight_price.describe()
```

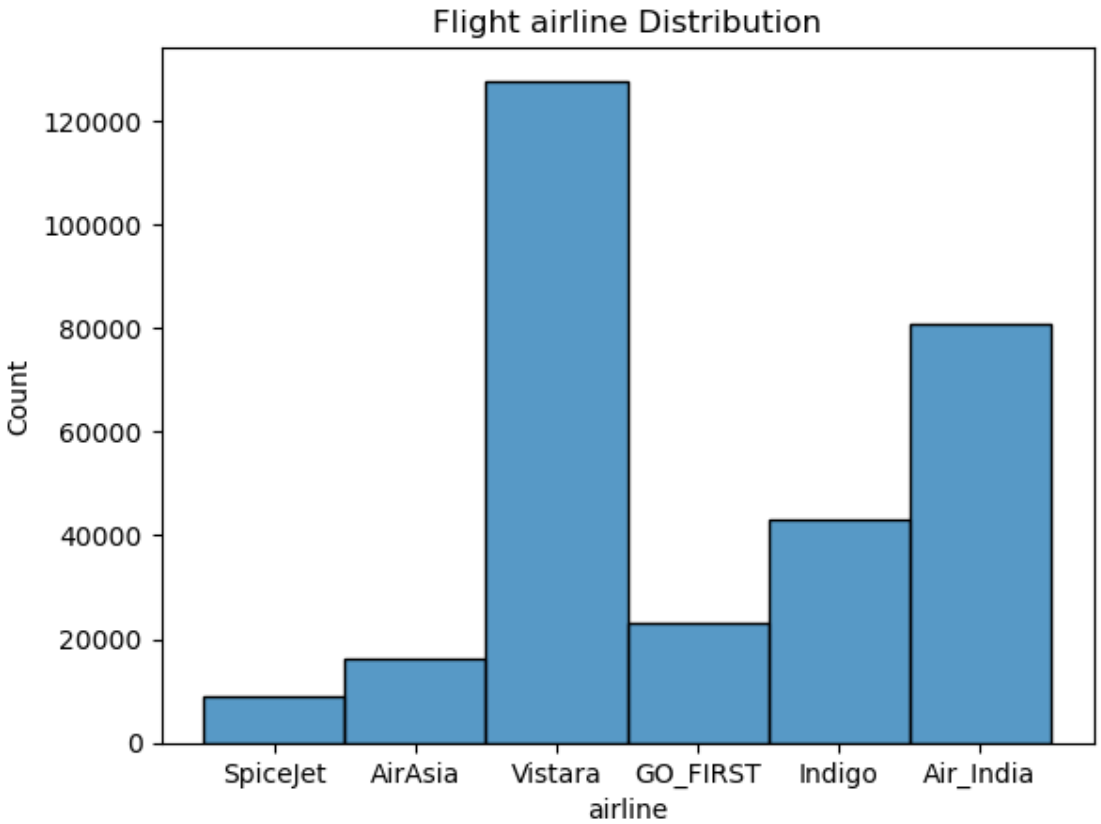
[119]:
```
                S.No.       duration      days_left  price (in Rs)
count  300153.000000  300153.000000  300153.000000  300153.000000
mean   150076.000000      12.221021      26.004751   20889.660523
```

```
std      86646.852011       7.191997      13.561004   22697.767366
min          0.000000       0.830000       1.000000    1105.000000
25%      75038.000000       6.830000      15.000000    4783.000000
50%     150076.000000      11.250000      26.000000    7425.000000
75%     225114.000000      16.170000      38.000000   42521.000000
max     300152.000000      49.830000      49.000000  123071.000000
```

[120]:
```
Flight_price['airline'].value_counts()
```

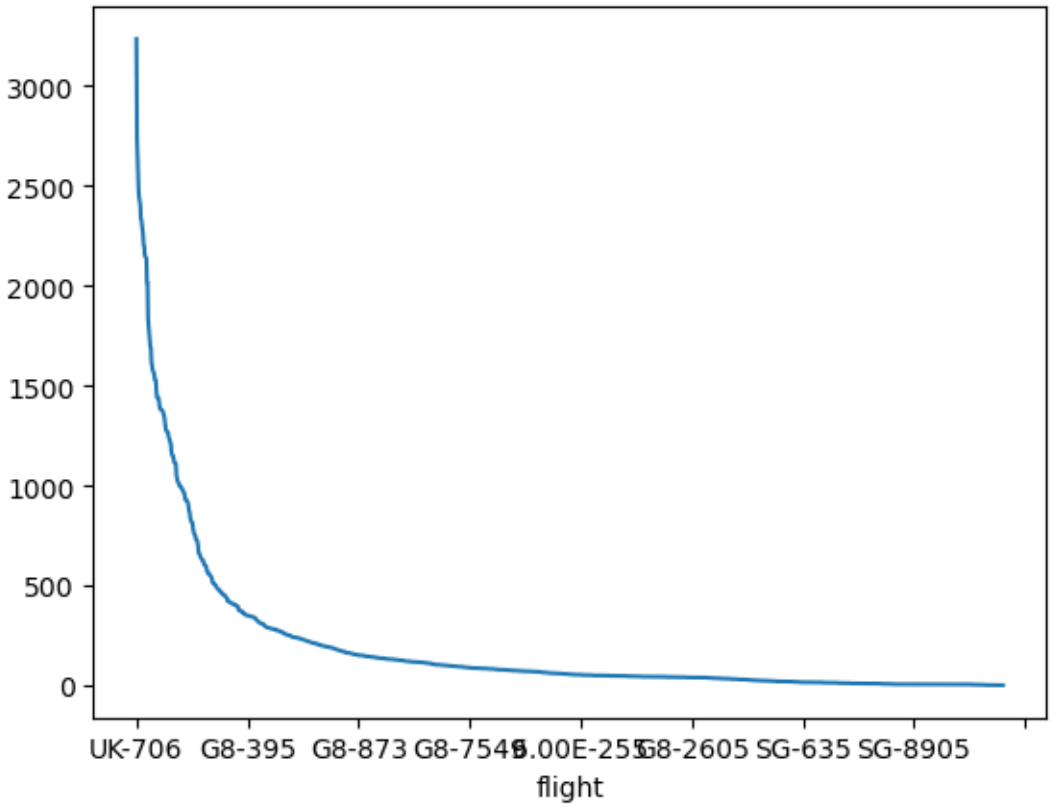
[120]:
```
airline
Vistara      127859
Air_India     80892
Indigo        43120
GO_FIRST      23173
AirAsia       16098
SpiceJet       9011
Name: count, dtype: int64
```

[121]:
```
sns.histplot(Flight_price.airline)
plt.title('Flight airline Distribution')
plt.show()
```

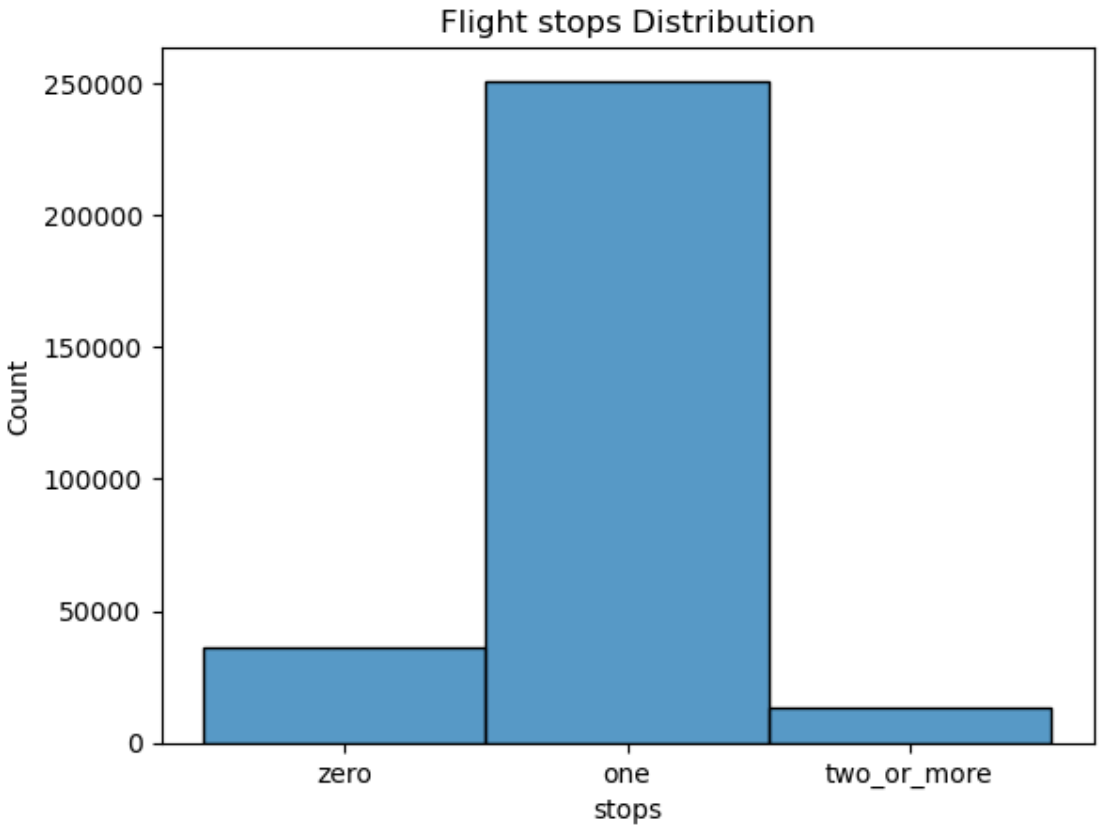


[122]:
```
Flight_price['flight'].value_counts().plot()
```

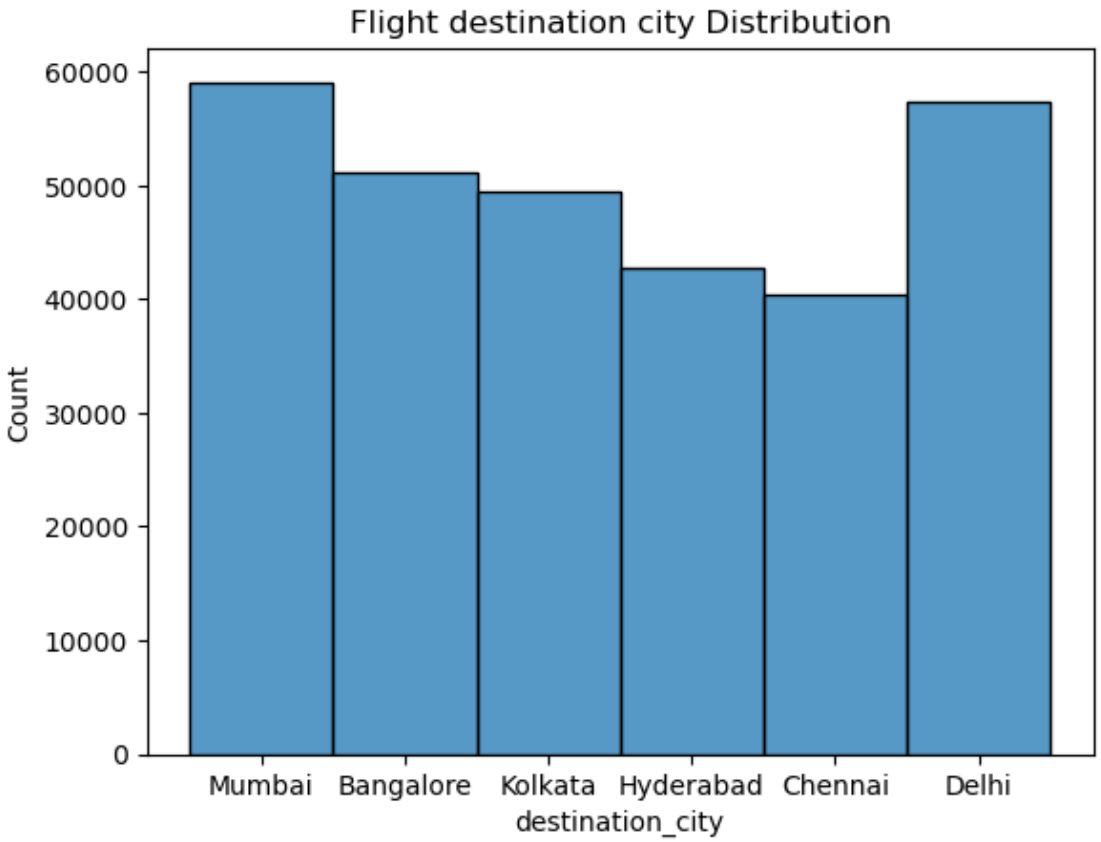
[122]: <Axes: xlabel='flight'>



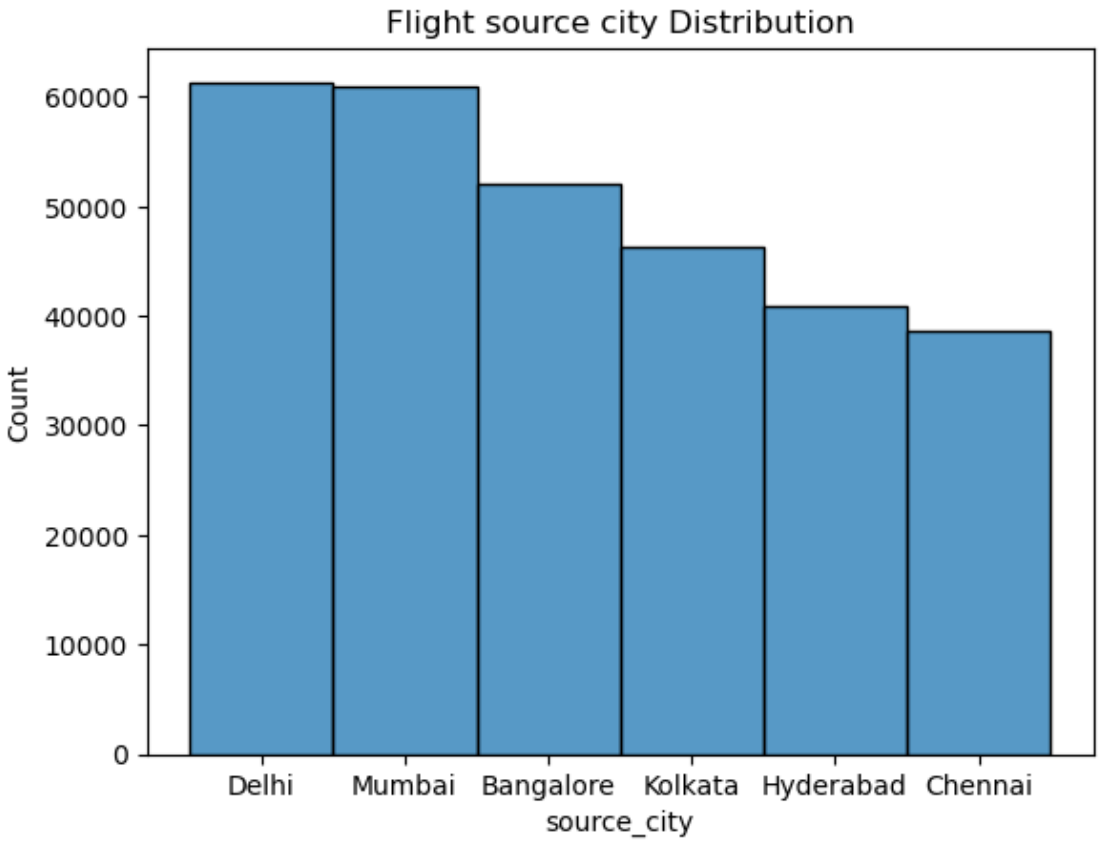
[123]:
```
sns.histplot(Flight_price.stops)
plt.title('Flight stops Distribution')
plt.show()
```



[124]:
```
sns.histplot(Flight_price.destination_city)
plt.title('Flight destination city Distribution')
plt.show()
```



[125]:
```
sns.histplot(Flight_price.source_city)
plt.title('Flight source city Distribution')
plt.show()
```



1.3 Exploratory Data Analysis

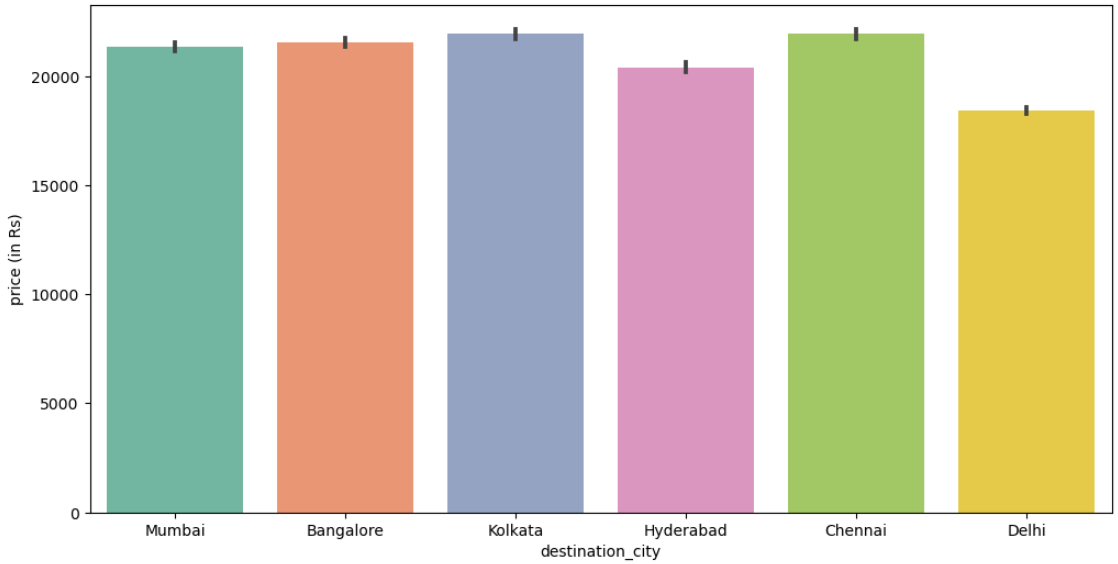
[126]:
```
compare_airline_price = pd.DataFrame(Flight_price.
 ↪groupby(['airline','class'])['price (in Rs)'].mean())
compare_airline_price
```

[126]:

airline	class	price (in Rs)
AirAsia	Economy	4091.072742
Air_India	Business	47131.039212
	Economy	7313.682169
GO_FIRST	Economy	5652.007595
Indigo	Economy	5324.216303
SpiceJet	Economy	6179.278881
Vistara	Business	55477.027777
	Economy	7806.943645

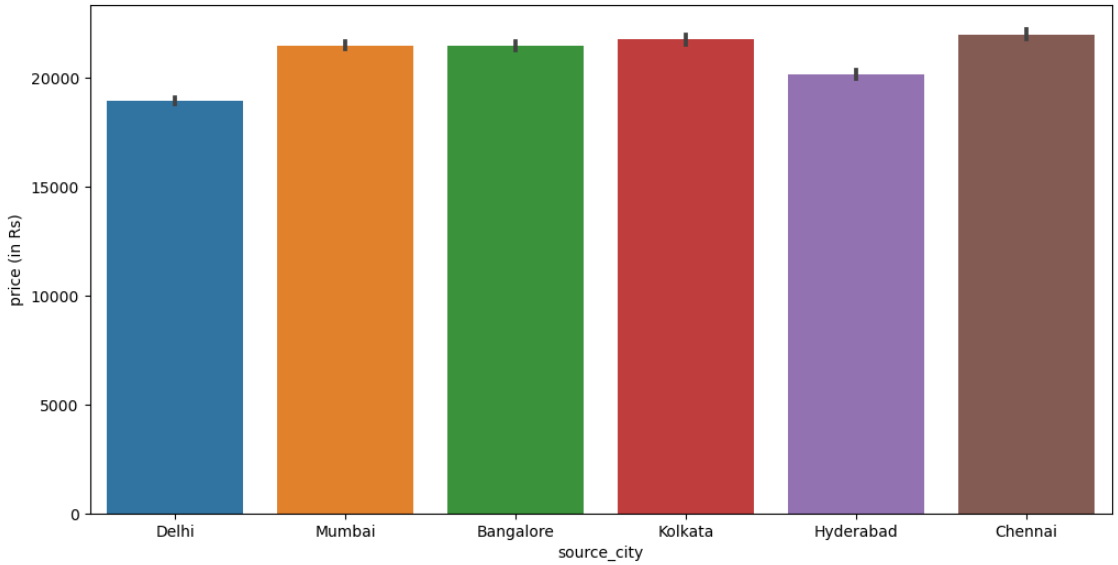
[127]:
```
plt.figure(figsize = (12,6))
sns.barplot(x = 'destination_city', y ='price (in Rs)', data = Flight_price,␣
 ↪palette = 'Set2')
```

[127]: <Axes: xlabel='destination_city', ylabel='price (in Rs)'>



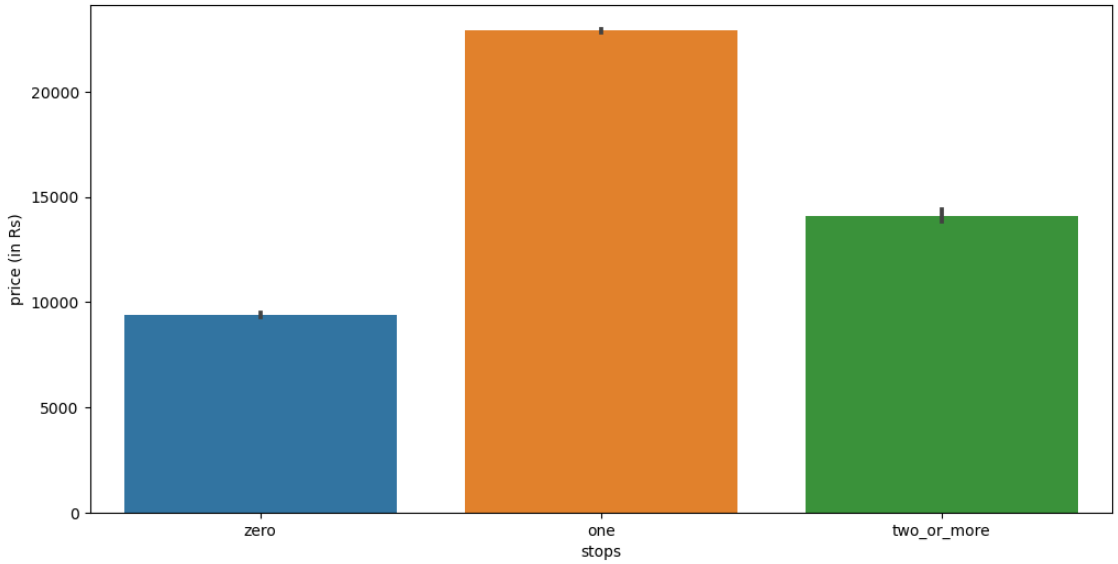
[128]:
```
plt.figure(figsize = (12,6))
sns.barplot(x = 'source_city', y ='price (in Rs)', data = Flight_price)
```

[128]: <Axes: xlabel='source_city', ylabel='price (in Rs)'>



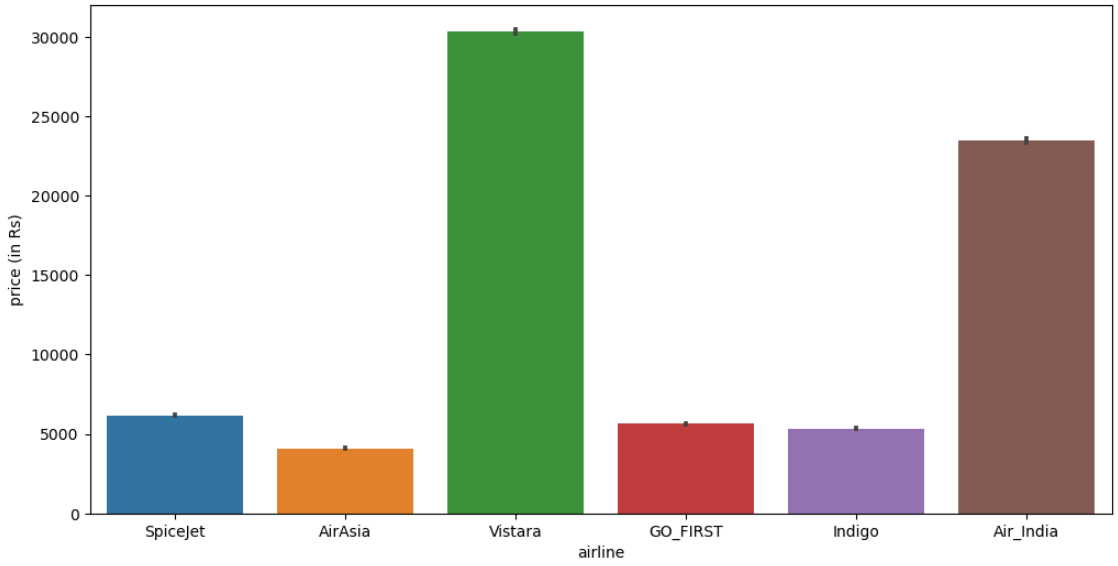
[129]:
```
plt.figure(figsize = (12,6))
sns.barplot(x = 'stops', y ='price (in Rs)', data = Flight_price)
```

[129]: <Axes: xlabel='stops', ylabel='price (in Rs)'>



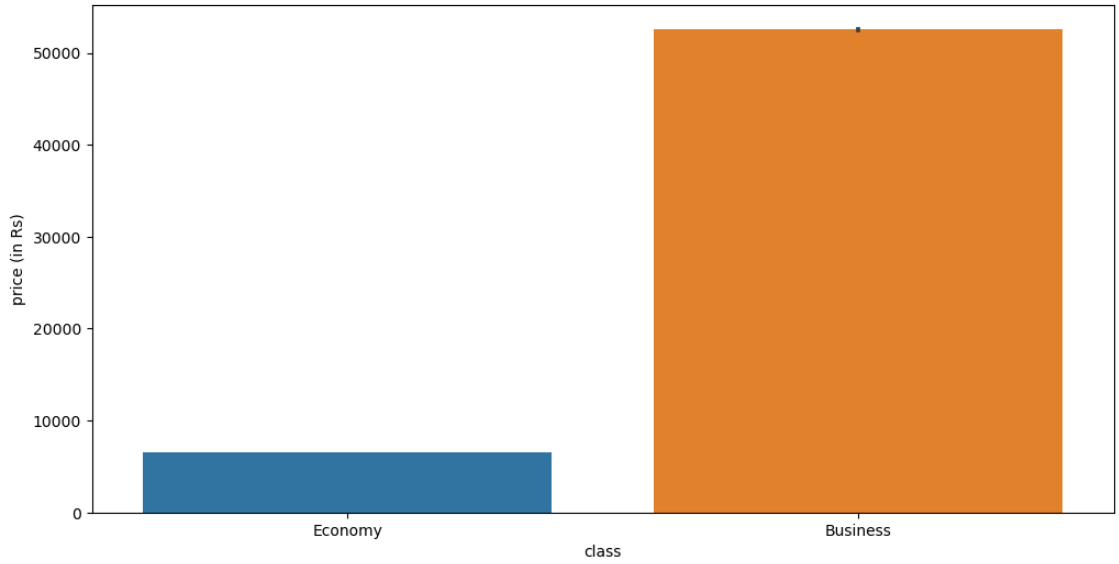
[130]:
```
plt.figure(figsize = (12,6))
sns.barplot(x = 'airline', y ='price (in Rs)', data = Flight_price)
```

[130]: <Axes: xlabel='airline', ylabel='price (in Rs)'>



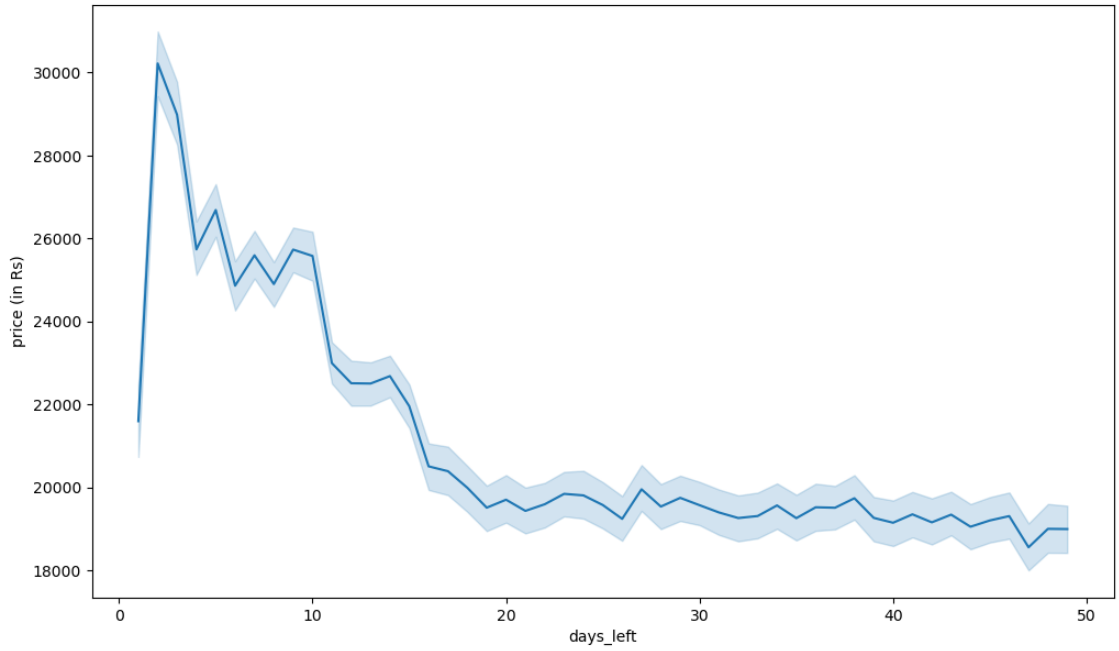
[131]:
```
plt.figure(figsize = (12,6))
sns.barplot(x = 'class', y ='price (in Rs)', data = Flight_price)
```

[131]: <Axes: xlabel='class', ylabel='price (in Rs)'>



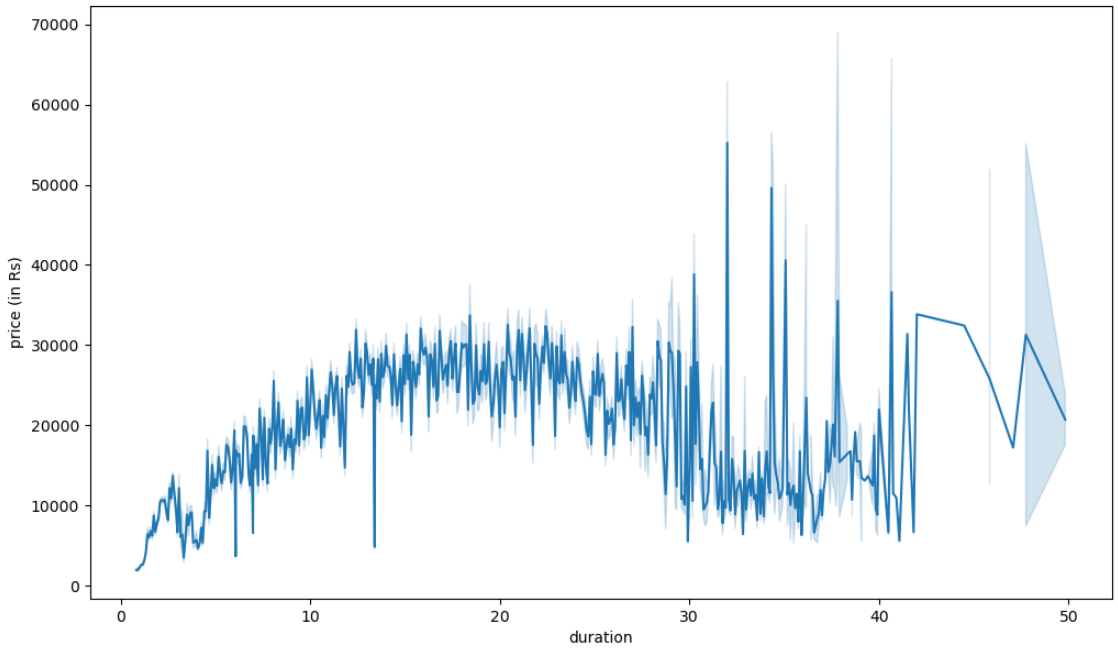
[132]:
```
plt.figure(figsize=(12,7))
sns.lineplot(x = 'days_left',y='price (in Rs)',data = Flight_price)
```

[132]: <Axes: xlabel='days_left', ylabel='price (in Rs)'>



[133]:
```
plt.figure(figsize=(12,7))
sns.lineplot(x = 'duration',y='price (in Rs)',data = Flight_price)
```

[133]:
```
<Axes: xlabel='duration', ylabel='price (in Rs)'>
```



1.4 Data Preparation

[134]:
```
# Convert class column into binary column
Flight_price['class'] = Flight_price['class'].apply(lambda x: 1 if␣
 ↪x=='Business' else 0)
Flight_price.head()
```

[134]:

	S.No.	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price (in Rs)
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	0	2.17	1	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	0	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	0	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	0	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	0	2.33	1	5955

[135]:
```
# Convert stops columns into categorical column
Flight_price['stops'] = Flight_price['stops'].apply(lambda x: 0 if x=='zero'␣
 ↪else 1 if x=='one' else 2)
Flight_price
```

[135]:

	S.No.	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price (in Rs)
0	0	SpiceJet	SG-8709	Delhi	Evening	0	Night	Mumbai	0	2.17	1	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	0	Morning	Mumbai	0	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	0	Early_Morning	Mumbai	0	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	0	Afternoon	Mumbai	0	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	0	Morning	Mumbai	0	2.33	1	5955
…	…	…	…	…	…	…	…	…	…	…	…	…
300148	300148	Vistara	UK-822	Chennai	Morning	1	Evening	Hyderabad	1	10.08	49	69265
300149	300149	Vistara	UK-826	Chennai	Afternoon	1	Night	Hyderabad	1	10.42	49	77105
300150	300150	Vistara	UK-832	Chennai	Early_Morning	1	Night	Hyderabad	1	13.83	49	79099
300151	300151	Vistara	UK-828	Chennai	Early_Morning	1	Evening	Hyderabad	1	10.00	49	81585
300152	300152	Vistara	UK-822	Chennai	Morning	1	Evening	Hyderabad	1	10.08	49	81585

[300153 rows x 12 columns]

[136]:
```
# Convert stops columns into categorical column
Flight_price.dummies = pd.get_dummies(Flight_price['airline'])
Flight_price.dummies
```

```
C:\Users\deeps\AppData\Local\Temp\ipykernel_46472\3180061013.py:2: UserWarning:
Pandas doesn't allow columns to be created via a new attribute name - see
https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  Flight_price.dummies = pd.get_dummies(Flight_price['airline'])
```

[136]:

	AirAsia	Air_India	GO_FIRST	Indigo	SpiceJet	Vistara
0	False	False	False	False	True	False
1	False	False	False	False	True	False
2	True	False	False	False	False	False
3	False	False	False	False	False	True
4	False	False	False	False	False	True
…	…	…	…	…	…	…
300148	False	False	False	False	False	True
300149	False	False	False	False	False	True
300150	False	False	False	False	False	True
300151	False	False	False	False	False	True
300152	False	False	False	False	False	True

[300153 rows x 6 columns]

[137]:
```
Flight_price.dummies = Flight_price.dummies.astype(int)
Flight_price.dummies
```

[137]:

	AirAsia	Air_India	GO_FIRST	Indigo	SpiceJet	Vistara
0	0	0	0	0	1	0
1	0	0	0	0	1	0
2	1	0	0	0	0	0
3	0	0	0	0	0	1
4	0	0	0	0	0	1
…	…	…	…	…	…	…
300148	0	0	0	0	0	1
300149	0	0	0	0	0	1
300150	0	0	0	0	0	1
300151	0	0	0	0	0	1
300152	0	0	0	0	0	1

[300153 rows x 6 columns]

[138]:
```
combined = pd.concat([Flight_price,Flight_price.dummies],axis='columns')
combined
```

[138]:

	S.No.	airline	flight	source_city	departure_time	stops	\
0	0	SpiceJet	SG-8709	Delhi	Evening	0	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	0	

```
2           2   AirAsia   I5-764       Delhi  Early_Morning      0
3           3   Vistara   UK-995       Delhi        Morning      0
4           4   Vistara   UK-963       Delhi        Morning      0
…         …       …       …          …              …    …
300148  300148   Vistara   UK-822     Chennai        Morning      1
300149  300149   Vistara   UK-826     Chennai      Afternoon      1
300150  300150   Vistara   UK-832     Chennai  Early_Morning      1
300151  300151   Vistara   UK-828     Chennai  Early_Morning      1
300152  300152   Vistara   UK-822     Chennai        Morning      1

         arrival_time destination_city  class  duration  days_left  \
0               Night           Mumbai      0      2.17          1
1             Morning           Mumbai      0      2.33          1
2       Early_Morning           Mumbai      0      2.17          1
3           Afternoon           Mumbai      0      2.25          1
4             Morning           Mumbai      0      2.33          1
…                 …                …    …       …          …
300148        Evening        Hyderabad      1     10.08         49
300149          Night        Hyderabad      1     10.42         49
300150          Night        Hyderabad      1     13.83         49
300151        Evening        Hyderabad      1     10.00         49
300152        Evening        Hyderabad      1     10.08         49

        price (in Rs)  AirAsia  Air_India  GO_FIRST  Indigo  SpiceJet  Vistara
0                5953        0          0         0       0         1        0
1                5953        0          0         0       0         1        0
2                5956        1          0         0       0         0        0
3                5955        0          0         0       0         0        1
4                5955        0          0         0       0         0        1
…                …       …          …         …     …         …      …
300148          69265        0          0         0       0         0        1
300149          77105        0          0         0       0         0        1
300150          79099        0          0         0       0         0        1
300151          81585        0          0         0       0         0        1
300152          81585        0          0         0       0         0        1

[300153 rows x 18 columns]
```

[139]:
```
final_flight = combined.drop(['airline'], axis='columns')
final_flight
```

[139]:
```
         S.No.   flight source_city departure_time  stops   arrival_time  \
0            0  SG-8709       Delhi        Evening      0          Night
1            1  SG-8157       Delhi  Early_Morning      0        Morning
2            2   I5-764       Delhi  Early_Morning      0  Early_Morning
3            3   UK-995       Delhi        Morning      0      Afternoon
4            4   UK-963       Delhi        Morning      0        Morning
```

```
…         …       …          …              …     …              …
300148  300148   UK-822     Chennai        Morning      1        Evening
300149  300149   UK-826     Chennai      Afternoon      1          Night
300150  300150   UK-832     Chennai  Early_Morning      1          Night
300151  300151   UK-828     Chennai  Early_Morning      1        Evening
300152  300152   UK-822     Chennai        Morning      1        Evening

       destination_city  class  duration  days_left  price (in Rs)  AirAsia  \
0                Mumbai      0      2.17          1           5953        0
1                Mumbai      0      2.33          1           5953        0
2                Mumbai      0      2.17          1           5956        1
3                Mumbai      0      2.25          1           5955        0
4                Mumbai      0      2.33          1           5955        0
…                   …    …        …         …              …       …
300148        Hyderabad      1     10.08         49          69265        0
300149        Hyderabad      1     10.42         49          77105        0
300150        Hyderabad      1     13.83         49          79099        0
300151        Hyderabad      1     10.00         49          81585        0
300152        Hyderabad      1     10.08         49          81585        0

        Air_India  GO_FIRST  Indigo  SpiceJet  Vistara
0               0         0       0         1        0
1               0         0       0         1        0
2               0         0       0         0        0
3               0         0       0         0        1
4               0         0       0         0        1
…             …       …     …       …       …
300148          0         0       0         0        1
300149          0         0       0         0        1
300150          0         0       0         0        1
300151          0         0       0         0        1
300152          0         0       0         0        1

[300153 rows x 17 columns]
```

1.5 Model Developmemnt for Flight Price Predicton

[140]:
```
from sklearn.linear_model import LinearRegression
from sklearn import datasets
from sklearn.model_selection import train_test_split
x =␣
 ↪final_flight[['days_left','class','stops','duration','AirAsia','Air_India','GO_FIRST','Indi
y = final_flight['price (in Rs)']

#DATA PARTITIONING
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=32)
```

[141]:
```
#model fitting
linearmodel = LinearRegression()
linearmodel.fit(X_train, y_train)
```

[141]:
```
LinearRegression()
```

[142]:
```
linearmodel.predict(X_test)
```

[142]:
```
array([ 8761.21026385,  8774.48079844, 60868.42542522, …,
       18674.53124244,  2046.91986758, 11389.04591774])
```

1.5.1 Model accuracy

[143]:
```
linearmodel.score(X_test,y_test)
```

[143]:
```
0.9064216075901539
```

[144]:
```
X_train.head()
```

[144]:

	days_left	class	stops	duration	AirAsia	Air_India	GO_FIRST	\
22688	16	0	0	2.08	0	0	0	
181611	12	0	0	2.83	0	0	0	
46295	19	0	1	8.17	0	0	0	
186736	44	0	1	27.17	0	0	0	
45765	16	0	2	7.83	0	0	1	

	Indigo	SpiceJet	Vistara
22688	0	0	1
181611	1	0	0
46295	1	0	0
186736	0	1	0
45765	0	0	0

2 Prediction

2.0.1 1. What is the price of economy class Indigo flight such that there is only 1 stop, flight duration should be 5 hours and flight after 20days.

[145]:
```
linearmodel.predict([[20,0,1,5,0,0,0,1,0,0]])
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with feature
names
  warnings.warn(
```

[145]:
```
array([7526.61198599])
```

2.0.2 2. Predict the price of business class Vistara flight such that there is no stop and flight is after 3days with 3 hours duration flight.

[146]:
```
linearmodel.predict([[3,1,0,3,0,0,0,0,0,1]])
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with feature
names
  warnings.warn(
```

[146]:
```
array([50380.95981305])
```

[ ]:
```

```