1 Regression - Flight Price Prediction

[102]:
```python
import pandas as pd
import numpy as np

flight_data = pd.read_csv("C:/Users/chait/Desktop/CPS-NEU/Courses/Analytics␣
↪System Technology/Assignments/Week2/flight_price_detection/Data_Train_csv.
↪csv")
```

[97]:
```python
flight_data
```

[97]:
```
            Airline Date_of_Journey    Source Destination  \
0            IndiGo      24/03/2019  Banglore   New Delhi
1         Air India        1/5/2019   Kolkata    Banglore
2       Jet Airways        9/6/2019     Delhi      Cochin
3            IndiGo       12/5/2019   Kolkata    Banglore
4            IndiGo        1/3/2019  Banglore   New Delhi
…               …              …        …         …
10677      Air Asia        9/4/2019   Kolkata    Banglore
10678     Air India      27/04/2019   Kolkata    Banglore
10679   Jet Airways      27/04/2019  Banglore       Delhi
10680       Vistara        1/3/2019  Banglore   New Delhi
10681     Air India        9/5/2019     Delhi      Cochin

                       Route Dep_Time    Arrival_Time Duration Total_Stops  \
0                  BLR → DEL    22:20  3/22/2021 1:10   2h 50m    non-stop
1      CCU → IXR → BBI → BLR     5:50           13:15   7h 25m     2 stops
2      DEL → LKO → BOM → COK     9:25  6/10/2021 4:25      19h     2 stops
3            CCU → NAG → BLR    18:05           23:30   5h 25m      1 stop
4            BLR → NAG → DEL    16:50           21:35   4h 45m      1 stop
…                          …      …              …      …           …
10677              CCU → BLR    19:55           22:25   2h 30m    non-stop
10678              CCU → BLR    20:45           23:20   2h 35m    non-stop
10679              BLR → DEL     8:20           11:20       3h    non-stop
10680              BLR → DEL    11:30           14:10   2h 40m    non-stop
10681  DEL → GOI → BOM → COK    10:55           19:15   8h 20m     2 stops
```

```
      Additional_Info  Price
0             No info   3897
1             No info   7662
2             No info  13882
3             No info   6218
4             No info  13302
…                 …      …
10677         No info   4107
10678         No info   4145
10679         No info   7229
10680         No info  12648
10681         No info  11753

[10682 rows x 11 columns]
```

Flight prices are affected greatly by the demand and supply at the given time. Bookings done at the end moment have the max surge in the prices. I wanted to know the factors affecting the surge in the prices of the flights. My goal is to understand on what factors is the rate of the flight prices dependent on? How do these factors play a role in affecting the prices Hence, a regression technique can be used to predict the factors affecting the prices. The dataset contains 10682 records containing different flight carriers having 11 columns such as the

1. Airline: The name of the airline.
2. Date_of_Journey: The date of the journey
3. Source: The source from which the service begins.
4. Destination: The destination where the service ends.
5. Route: The route taken by the flight to reach the destination.
6. Dep_Time: The time when the journey starts from the source.
7. Arrival_Time: Time of arrival at the destination.
8. Duration: Total duration of the flight.
9. Total_Stops: Total stops between the source and destination.
10. Additional_Info: Additional information about the flight
11. Price: The price of the ticket

[63]:
```
flight_data.shape
flight_data.columns
```

[63]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

[64]:
```
flight_data.describe()
```

[64]:
```
              Price
count  10682.000000
mean    9086.292735
std     4610.885695
```

```
min      1759.000000
25%      5277.000000
50%      8372.000000
75%     12373.000000
max     79512.000000
```

[65]:
```
flight_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10682 entries, 0 to 10681
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10682 non-null  object
 1   Date_of_Journey  10682 non-null  object
 2   Source           10682 non-null  object
 3   Destination      10682 non-null  object
 4   Route            10681 non-null  object
 5   Dep_Time         10682 non-null  object
 6   Arrival_Time     10682 non-null  object
 7   Duration         10682 non-null  object
 8   Total_Stops      10681 non-null  object
 9   Additional_Info  10682 non-null  object
 10  Price            10682 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.1+ KB
```

2 Data Cleaning and Processing

We need to do some data cleaning and processing here before our analysis.We also need to check for nulls. We use a Data Dictionary to map the values in the Total_stops column to integer values.Since we need this as a part of our data analysis in the further steps and also for data visualisation.

[103]:
```
# Using Dictionary to map Total Number of Stops
flight_data = flight_data.dropna(subset=['Total_Stops'])
d = {"non-stop": 0, "1 stop": 1,"2 stops":2,"3 stops":3,"4 stops":4}
flight_data['Total_Stops'] = flight_data['Total_Stops'].map(d)
flight_data['Total_Stops'] = flight_data['Total_Stops'].astype(int)
```

[104]:
```
flight_data.head()
```

[104]:
```
       Airline Date_of_Journey    Source Destination                  Route  \
0       IndiGo      24/03/2019  Banglore   New Delhi              BLR → DEL   
1    Air India        1/5/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR   
2  Jet Airways        9/6/2019     Delhi      Cochin  DEL → LKO → BOM → COK   
3       IndiGo       12/5/2019   Kolkata    Banglore        CCU → NAG → BLR   
4       IndiGo        1/3/2019  Banglore   New Delhi        BLR → NAG → DEL   
```

```
  Dep_Time    Arrival_Time Duration  Total_Stops Additional_Info  Price
0    22:20  3/22/2021 1:10   2h 50m            0         No info   3897
1     5:50           13:15   7h 25m            2         No info   7662
2     9:25  6/10/2021 4:25      19h            2         No info  13882
3    18:05           23:30   5h 25m            1         No info   6218
4    16:50           21:35   4h 45m            1         No info  13302
```

[67]:
```
flight_data.Additional_Info.unique()
```

[67]:
```
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)
```

Route Column is split to get the multiple columns with the cities that the flights travels to.

[105]:
```
flight_data.Route = flight_data.Route.str.split('→')
flight_data['Route_City1_Code'] = flight_data.Route.str[0]
flight_data['Route_City2_Code'] = flight_data.Route.str[1]
flight_data['Route_City3_Code'] = flight_data.Route.str[2]
flight_data['Route_City4_Code'] = flight_data.Route.str[3]
flight_data['Route_City5_Code'] = flight_data.Route.str[4]
flight_data['Route_City6_Code'] = flight_data.Route.str[5]

#check for the flight data
flight_data
```

[105]:
```
           Airline Date_of_Journey    Source Destination  \
0           IndiGo      24/03/2019  Banglore   New Delhi
1        Air India        1/5/2019   Kolkata    Banglore
2      Jet Airways        9/6/2019     Delhi      Cochin
3           IndiGo       12/5/2019   Kolkata    Banglore
4           IndiGo        1/3/2019  Banglore   New Delhi
…              …              …        …         …
10677     Air Asia        9/4/2019   Kolkata    Banglore
10678    Air India      27/04/2019   Kolkata    Banglore
10679  Jet Airways      27/04/2019  Banglore       Delhi
10680      Vistara        1/3/2019  Banglore   New Delhi
10681    Air India        9/5/2019     Delhi      Cochin

                           Route Dep_Time    Arrival_Time Duration  \
0                    [BLR , DEL]    22:20  3/22/2021 1:10   2h 50m
1      [CCU , IXR , BBI , BLR]     5:50           13:15   7h 25m
2      [DEL , LKO , BOM , COK]     9:25  6/10/2021 4:25      19h
3             [CCU , NAG , BLR]    18:05           23:30   5h 25m
4             [BLR , NAG , DEL]    16:50           21:35   4h 45m
…                           …      …              …        …
```

```
10677                [CCU ,  BLR]   19:55          22:25   2h 30m
10678                [CCU ,  BLR]   20:45          23:20   2h 35m
10679                [BLR ,  DEL]    8:20          11:20       3h
10680                [BLR ,  DEL]   11:30          14:10   2h 40m
10681  [DEL ,  GOI ,  BOM ,  COK]   10:55          19:15   8h 20m

       Total_Stops Additional_Info  Price Route_City1_Code Route_City2_Code  \
0                0         No info   3897              BLR              DEL
1                2         No info   7662              CCU              IXR
2                2         No info  13882              DEL              LKO
3                1         No info   6218              CCU              NAG
4                1         No info  13302              BLR              NAG
…              …             …   …               …                …
10677            0         No info   4107              CCU              BLR
10678            0         No info   4145              CCU              BLR
10679            0         No info   7229              BLR              DEL
10680            0         No info  12648              BLR              DEL
10681            2         No info  11753              DEL              GOI

      Route_City3_Code Route_City4_Code Route_City5_Code Route_City6_Code
0                  NaN              NaN              NaN              NaN
1                 BBI               BLR              NaN              NaN
2                 BOM               COK              NaN              NaN
3                  BLR              NaN              NaN              NaN
4                  DEL              NaN              NaN              NaN
…                …              …              …              …
10677              NaN              NaN              NaN              NaN
10678              NaN              NaN              NaN              NaN
10679              NaN              NaN              NaN              NaN
10680              NaN              NaN              NaN              NaN
10681             BOM               COK              NaN              NaN

[10681 rows x 17 columns]
```

[69]:
```
#Checking for nulls
flight_data.isnull().sum()
```

[69]:
```
Airline                0
Date_of_Journey        0
Source                 0
Destination            0
Route                  0
Dep_Time               0
Arrival_Time           0
Duration               0
Total_Stops            0
Additional_Info        0
```

```
Price                   0
Route_City1_Code        0
Route_City2_Code        0
Route_City3_Code     3491
Route_City4_Code     9116
Route_City5_Code    10635
Route_City6_Code    10680
dtype: int64
```

[70]:
```
flight_data.shape
```

[70]:
```
(10681, 17)
```

[71]:
```
flight_data.Route_City3_Code.unique()
```

[71]:
```
array([nan, ' BBI ', ' BOM ', ' BLR', ' DEL', ' COK', ' DEL ', ' AMD ',
       ' HYD', ' JDH ', ' MAA ', ' COK ', ' GOI ', ' NAG ', ' GAU ',
       ' BHO ', ' IXR ', ' IDR ', ' ISK ', ' HYD ', ' VGA ', ' PNQ ',
       ' JAI ', ' TRV ', ' HBX ', ' IMF ', ' CCU ', ' UDR ', ' VTZ ',
       ' IXC '], dtype=object)
```

[72]:
```
flight_data.Route_City4_Code.unique()
```

[72]:
```
array([nan, ' BLR', ' COK', ' DEL', ' BOM ', ' HYD', ' DEL ', ' HYD ',
       ' GWL ', ' TRV ', ' BBI ', ' BHO ', ' AMD ', ' NAG '], dtype=object)
```

[88]:
```
# Strip the additional columns of white spaces before mapping them
flight_data["Route_City1_Code"] = flight_data["Route_City1_Code"].str.strip()
flight_data["Route_City2_Code"] = flight_data["Route_City2_Code"].str.strip()
flight_data["Route_City3_Code"] = flight_data["Route_City3_Code"].str.strip()
flight_data["Route_City4_Code"] = flight_data["Route_City4_Code"].str.strip()
flight_data["Route_City5_Code"] = flight_data["Route_City5_Code"].str.strip()
flight_data["Route_City6_Code"] = flight_data["Route_City6_Code"].str.strip()
```

Furthermore,since the data is now split into City Codes, we use a Data Dictionary to map the city codes with the corresponding cities by using a csv file with the city codes along with the cities.Hence,now we have additional columns with all the city codes.

[74]:
```
# Create Data Dictionary to map city names with their codes imported from csv␣
 ↪file
import csv


with open('C:/Users/chait/Desktop/CPS-NEU/Courses/Analytics System Technology/
 ↪Assignments/Week2/flight_price_detection/airport_codes.csv', mode='r') as␣
 ↪inp:
    reader = csv.reader(inp)
    mydict = {rows[5]:rows[4] for rows in reader}
    mydict.pop("Code")

print(mydict)
```

```
{'DEL': 'New Delhi', 'BOM': 'Mumbai', 'MAA': 'Chennai', 'BLR': 'Bangalore',
'GOI': 'Vasco da Gama', 'CCU': 'Kolkata', 'COK': 'Kochi', 'HYD': 'Hyderabad',
'AMD': 'Ahmedabad', 'TRV': 'Thiruvananthapuram', 'JAI': 'Jaipur', 'PNQ': 'Pune',
'ATQ': 'Amritsar', 'BDQ': 'Vadodara', 'IXE': 'Mangalore', 'CCJ': 'Calicut',
'VNS': 'Varanasi', 'CJB': 'Coimbatore', 'GAU': 'Guwahati', 'BPM': 'Hyderabad',
'VTZ': 'Visakhapatnam', 'PAT': 'Patna', 'BBI': 'Bhubaneswar', 'IXM': 'Madurai',
'LKO': 'Lucknow', 'IDR': 'Indore', 'JDH': 'Jodhpur', 'IXC': 'Chandigarh', 'IXB':
'Siliguri', 'STV': '', 'IXZ': 'Port Blair', 'NAG': 'Nagpur', 'UDR': 'Udaipur',
'IXR': 'Ranchi', 'RPR': 'Raipur', 'SXR': 'Srinagar', 'RAJ': 'Rajkot', 'IXU':
'Aurangabad', 'BHO': 'Bhopal', 'BHJ': 'Bhuj', 'IXA': 'Agartala', 'VGA':
'Gannavaram', 'JRH': 'Jorhat', 'IXL': 'Leh', 'KUU': 'Bhuntar', 'PUT':
'Puttaparthi', 'DBR': 'Darbhanga, Bihar, India', 'GDB': 'Gondia', 'ISK':
'Nasik', 'IMF': 'Imphal', 'JGA': 'Jamnagar', 'RJA': 'Rajahmundry', 'TIR':
'Tirupati', 'DIB': 'Dibrugarh', 'HJR': 'Khajuraho', 'AGX': 'Agatti', 'IXD':
'Allahabad', 'SLV': 'Jubbarhatti', 'CNN': 'Kannur', 'TRZ': 'Tiruchirappalli',
'IXJ': 'Jammu', 'IXG': 'Belgaum', 'BHU': 'Bhavnagar', 'TEZ': '', 'GAY': '',
'JLR': '', 'DED': 'Dehradun', 'PBD': 'Porbandar', 'IXS': 'Silchar', 'DIU':
'Diu', 'HBX': 'Hubli', 'RRK': '', 'BUP': '', 'AGR': 'Agra', 'BEP': 'Bellary',
'IXX': 'Bidar', 'DHM': 'Gaggal', 'IXW': 'Jamshedpur', 'IXK': 'Keshod', 'KTU':
'Kota', 'IXP': 'Pathankot', 'PNY': 'Puducherry (Pondicherry)', 'KNU': 'Kanpur',
'MYQ': 'Mysore', 'REW': 'Rewa', 'KLH': '', 'TNI': '', 'DMU': 'Dimapur', 'GWL':
'Gwalior', 'IXI': 'Lilabari', 'SHL': 'Shillong', 'IXV': '', 'DBD': '', 'MZU':
'', 'HSS': '', 'RTC': '', 'GUX': '', 'LUH': '', 'JSA': '', 'AKD': 'Akola',
'BEK': 'Bareilly', 'BKB': 'Bikaner', 'PAB': 'Bilaspur', 'NMB': 'Daman', 'GOP':
'Gorakhpur', 'CBD': 'IAF Camp', 'CDP': 'Kadapa', 'IXH': 'Kailashahar', 'IXY':
'Kandla', 'PGH': 'Pantnagar', 'SXV': 'Salem', 'ZER': 'Ziro', 'RUP': '', 'JRG':
'', 'AIP': 'Adampur', 'AJL': 'Aizawl (Lengpui)', 'RGH': 'Balurghat', 'JGB':
'Jagdalpur', 'PYB': 'Jeypore', 'IXN': 'Khowai', 'LTU': 'Latur', 'LDA': 'Malda',
'IXQ': 'Manik Bhandar', 'NDC': 'Nanded', 'NVY': 'Neyveli', 'IXT': 'Pasighat',
'RJI': 'Rajouri', 'SSE': 'Solapur', 'TEI': 'Tezu', 'TJV': 'Thanjavur', 'TCR':
'Vagaikulam', 'WGC': 'Warangal', 'COH': '', 'KQH': 'Ajmer (Kishangarh)', 'DEP':
'Daporijo', 'RDP': 'Durgapur', 'SAG': 'Kakadi', 'KJB': 'Orvakal', 'OMN':
'Osmanabad', 'PYG': 'Pakyong', 'RMD': 'Ramagundam', 'VDY': 'Toranagallu'}
```

[87]:
```
# Mapping the city codes to values
flight_data['Route_City1']= flight_data['Route_City1_Code'].map(mydict)
flight_data['Route_City2']= flight_data['Route_City2_Code'].map(mydict)
flight_data['Route_City3'] = flight_data['Route_City3_Code'].map(mydict)
flight_data['Route_City4']= flight_data['Route_City4_Code'].map(mydict)
flight_data['Route_City5']= flight_data['Route_City5_Code'].map(mydict)
flight_data['Route_City6']= flight_data['Route_City6_Code'].map(mydict)
flight_data.head()
```

[87]:
```
  Airline Date_of_Journey    Source Destination                  Route Dep_Time  \
0  IndiGo      24/03/2019  Banglore       Delhi              [BLR , DEL]    22:20
1  IndiGo        1/3/2019  Banglore       Delhi   [BLR , NAG , DEL]    16:50
2  IndiGo        3/4/2019  Banglore       Delhi              [BLR , DEL]     4:00
3  IndiGo        1/5/2019  Banglore       Delhi              [BLR , DEL]    18:55
4  IndiGo        6/4/2019  Banglore       Delhi              [BLR , DEL]     4:00

    Arrival_Time Duration  Total_Stops Additional_Info  …  Route_City1  \
0  3/22/2021 1:10   2h 50m            0         No info  …    Bangalore
1           21:35   4h 45m            1         No info  …    Bangalore
2            6:50   2h 50m            0         No info  …    Bangalore
3           21:50   2h 55m            0         No info  …    Bangalore
4            6:50   2h 50m            0         No info  …    Bangalore

  Route_City2 Route_City3 Route_City4 Route_City5 Route_City6   Mean price  \
0   New Delhi         NaN         NaN         NaN         NaN  5274.112811
1      Nagpur   New Delhi         NaN         NaN         NaN  5274.112811
2   New Delhi         NaN         NaN         NaN         NaN  5274.112811
3   New Delhi         NaN         NaN         NaN         NaN  5274.112811
4   New Delhi         NaN         NaN         NaN         NaN  5274.112811

  Hours minutes total_travel_time(mins)
0     2      50                     170
1     4      45                     285
2     2      50                     170
3     2      55                     175
4     2      50                     170

[5 rows x 27 columns]
```

[84]:
```
# New Delhi -> Delhi (Since both the cities are same)
flight_data["Destination"].replace({"New Delhi": "Delhi"}, inplace=True)
flight_data.Total_Stops.unique()

#Only one row in total_stops column is null,the corr Route col is also␣
 ↪null,hence dropping the row
flight_data = flight_data.dropna(subset=['Total_Stops'])
flight_data.head()
```

[84]:
```
  Airline Date_of_Journey    Source Destination                  Route Dep_Time  \
0  IndiGo      24/03/2019  Banglore       Delhi              [BLR , DEL]    22:20
1  IndiGo        1/3/2019  Banglore       Delhi   [BLR , NAG , DEL]    16:50
2  IndiGo        3/4/2019  Banglore       Delhi              [BLR , DEL]     4:00
3  IndiGo        1/5/2019  Banglore       Delhi              [BLR , DEL]    18:55
4  IndiGo        6/4/2019  Banglore       Delhi              [BLR , DEL]     4:00

    Arrival_Time Duration  Total_Stops Additional_Info  …  Route_City1  \
0  3/22/2021 1:10   2h 50m            0         No info  …    Bangalore
1           21:35   4h 45m            1         No info  …    Bangalore
2            6:50   2h 50m            0         No info  …    Bangalore
3           21:50   2h 55m            0         No info  …    Bangalore
4            6:50   2h 50m            0         No info  …    Bangalore

  Route_City2 Route_City3 Route_City4 Route_City5 Route_City6   Mean price  \
0   New Delhi         NaN         NaN         NaN         NaN  5274.112811
1      Nagpur   New Delhi         NaN         NaN         NaN  5274.112811
2   New Delhi         NaN         NaN         NaN         NaN  5274.112811
3   New Delhi         NaN         NaN         NaN         NaN  5274.112811
4   New Delhi         NaN         NaN         NaN         NaN  5274.112811

  Hours minutes total_travel_time(mins)
0     2      50                     170
1     4      45                     285
2     2      50                     170
3     2      55                     175
4     2      50                     170

[5 rows x 27 columns]
```

Next, we use Data Dictionary again to map the Airlines by their Destinations. This gives a list of Airlines with the corresponding destinations.

[77]:
```
# Using Dictionary to map Airlines and the destinations
from collections import defaultdict
AirlineDestinations = defaultdict(list)
for i, j in zip(flight_data.Airline,flight_data.Destination):
    if j not in AirlineDestinations[i]:
        AirlineDestinations[i].append(j)
```

[78]:
```
AirlineDestinations
```

[78]:
```
defaultdict(list,
            {'IndiGo': ['Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Hyderabad'],
             'Air India': ['Banglore',
              'Cochin',
              'Kolkata',
              'Delhi',
              'Hyderabad'],
             'Jet Airways': ['Cochin', 'Delhi', 'Banglore', 'Hyderabad'],
             'SpiceJet': ['Banglore',
              'Cochin',
              'Delhi',
              'Kolkata',
              'Hyderabad'],
             'Multiple carriers': ['Cochin'],
             'GoAir': ['Cochin', 'Delhi', 'Banglore'],
             'Vistara': ['Delhi',
              'Kolkata',
              'Hyderabad',
              'Banglore',
              'Cochin'],
             'Air Asia': ['Delhi', 'Banglore', 'Cochin'],
             'Vistara Premium economy': ['Delhi', 'Kolkata'],
             'Jet Airways Business': ['Delhi', 'Cochin'],
             'Multiple carriers Premium economy': ['Cochin'],
             'Trujet': ['Hyderabad']})
```

3 The Pandas Profiling Report for our dataset is as follows

[79]:
```
import pandas as pd
from pandas_profiling import ProfileReport

profile = ProfileReport(flight_data, title="Pandas Profiling Report")
profile
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[79]:

We group the dataset by Airline Source and Destination to find the Mean Price for each Airline as per the corresponding Source and Destination. We split the Duration column into two columns- Hour and Minutes.

[80]:
```
mean_price = flight_data.groupby(['Airline','Source','Destination'])['Price'].
 ↪mean().rename("Mean price").reset_index()
flight_data = flight_data.merge(mean_price)
flight_data
```

[80]:
```
                                 Airline Date_of_Journey    Source  \
0                                 IndiGo      24/03/2019  Banglore
1                                 IndiGo        1/3/2019  Banglore
2                                 IndiGo        3/4/2019  Banglore
3                                 IndiGo        1/5/2019  Banglore
4                                 IndiGo        6/4/2019  Banglore
…                                    …             …        …
10676  Multiple carriers Premium economy      21/03/2019     Delhi
10677                             Trujet        6/3/2019    Mumbai
10678               Jet Airways Business        3/3/2019     Delhi
```

```
10679              Jet Airways Business        6/3/2019     Delhi
10680           Vistara Premium economy        1/3/2019   Chennai

      Destination                       Route Dep_Time    Arrival_Time  \
0           Delhi                 [BLR ,  DEL]    22:20  3/22/2021 1:10
1           Delhi          [BLR ,  NAG ,  DEL]    16:50           21:35
2           Delhi                 [BLR ,  DEL]     4:00            6:50
3           Delhi                 [BLR ,  DEL]    18:55           21:50
4           Delhi                 [BLR ,  DEL]     4:00            6:50
…           …                               …      …             …
10676      Cochin          [DEL ,  BOM ,  COK]     7:30           15:30
10677   Hyderabad          [BOM ,  NDC ,  HYD]    13:05           16:20
10678      Cochin   [DEL ,  ATQ ,  BOM ,  COK]    20:05   3/4/2021 4:25
10679      Cochin   [DEL ,  ATQ ,  BOM ,  COK]    20:05   3/7/2021 4:25
10680     Kolkata                 [MAA ,  CCU]     7:05            9:20

      Duration  Total_Stops Additional_Info  …  Route_City4_Code  \
0       2h 50m            0         No info  …               NaN
1       4h 45m            1         No info  …               NaN
2       2h 50m            0         No info  …               NaN
3       2h 55m            0         No info  …               NaN
4       2h 50m            0         No info  …               NaN
…          …          …               …  …               …
10676       8h            1         No info  …               NaN
10677   3h 15m            1         No info  …               NaN
10678   8h 20m            2         No info  …               COK
10679   8h 20m            2         No info  …               COK
10680   2h 15m            0         No info  …               NaN

      Route_City5_Code Route_City6_Code Route_City1 Route_City2 Route_City3  \
0                  NaN              NaN   Bangalore   New Delhi         NaN
1                  NaN              NaN   Bangalore      Nagpur   New Delhi
2                  NaN              NaN   Bangalore   New Delhi         NaN
3                  NaN              NaN   Bangalore   New Delhi         NaN
4                  NaN              NaN   Bangalore   New Delhi         NaN
…                …                …          …          …          …
10676              NaN              NaN   New Delhi      Mumbai       Kochi
10677              NaN              NaN      Mumbai      Nanded   Hyderabad
10678              NaN              NaN   New Delhi    Amritsar      Mumbai
10679              NaN              NaN   New Delhi    Amritsar      Mumbai
10680              NaN              NaN     Chennai     Kolkata         NaN

      Route_City4 Route_City5 Route_City6    Mean price
0             NaN         NaN         NaN   5274.112811
1             NaN         NaN         NaN   5274.112811
2             NaN         NaN         NaN   5274.112811
3             NaN         NaN         NaN   5274.112811
```

```
4              NaN         NaN         NaN   5274.112811
…            …          …          …           …
10676          NaN         NaN         NaN  11418.846154
10677          NaN         NaN         NaN   4140.000000
10678        Kochi         NaN         NaN  49387.500000
10679        Kochi         NaN         NaN  49387.500000
10680          NaN         NaN         NaN   9125.000000

[10681 rows x 24 columns]
```

[139]:
```
# Split the Duration Column into Hours and Minutes
h = flight_data.Duration.str.split(' ')
hour = h.str[0]
hour  = hour.str.split('h').str[0]
#print(hour)
flight_data['Hours'] = hour
#print(h)
m = h.str[1]
minutes  = m.str.split('m').str[0]
#print(minutes)
flight_data['minutes'] = minutes
flight_data
```

[139]:
```
           Airline Date_of_Journey    Source Destination  \
0           IndiGo      24/03/2019  Banglore   New Delhi
1        Air India       1/5/2019   Kolkata    Banglore
2      Jet Airways       9/6/2019     Delhi      Cochin
3           IndiGo      12/5/2019   Kolkata    Banglore
4           IndiGo       1/3/2019  Banglore   New Delhi
…            …            …         …        …
10677     Air Asia       9/4/2019   Kolkata    Banglore
10678    Air India      27/04/2019  Kolkata    Banglore
10679  Jet Airways      27/04/2019 Banglore       Delhi
10680      Vistara       1/3/2019  Banglore   New Delhi
10681    Air India       9/5/2019     Delhi      Cochin

                           Route Dep_Time    Arrival_Time Duration  \
0                    [BLR ,  DEL]    22:20  3/22/2021 1:10   2h 50m
1      [CCU ,  IXR ,  BBI ,  BLR]     5:50           13:15   7h 25m
2      [DEL ,  LKO ,  BOM ,  COK]     9:25  6/10/2021 4:25      19h
3             [CCU ,  NAG ,  BLR]    18:05           23:30   5h 25m
4             [BLR ,  NAG ,  DEL]    16:50           21:35   4h 45m
…                               …       …              …        …
10677                [CCU ,  BLR]    19:55           22:25   2h 30m
10678                [CCU ,  BLR]    20:45           23:20   2h 35m
10679                [BLR ,  DEL]     8:20           11:20       3h
10680                [BLR ,  DEL]    11:30           14:10   2h 40m
```

```
10681  [DEL ,  GOI ,  BOM ,  COK]    10:55          19:15   8h 20m

       Total_Stops Additional_Info  Price Route_City1_Code Route_City2_Code  \
0                0         No info   3897             BLR              DEL
1                2         No info   7662             CCU             IXR
2                2         No info  13882             DEL             LKO
3                1         No info   6218             CCU             NAG
4                1         No info  13302             BLR             NAG
…             …               …    …              …               …
10677            0         No info   4107             CCU              BLR
10678            0         No info   4145             CCU              BLR
10679            0         No info   7229             BLR              DEL
10680            0         No info  12648             BLR              DEL
10681            2         No info  11753             DEL             GOI

      Route_City3_Code Route_City4_Code Route_City5_Code Route_City6_Code  \
0                  NaN              NaN              NaN              NaN
1                 BBI              BLR              NaN              NaN
2                 BOM              COK              NaN              NaN
3                  BLR              NaN              NaN              NaN
4                  DEL              NaN              NaN              NaN
…                 …              …              …              …
10677              NaN              NaN              NaN              NaN
10678              NaN              NaN              NaN              NaN
10679              NaN              NaN              NaN              NaN
10680              NaN              NaN              NaN              NaN
10681             BOM              COK              NaN              NaN

      Hours minutes
0         2      50
1         7      25
2        19     NaN
3         5      25
4         4      45
…      …     …
10677     2      30
10678     2      35
10679     3     NaN
10680     2      40
10681     8      20

[10681 rows x 19 columns]
```

We now compute the Total Travel Time by calculating the values in the Hours and Minutes columns into Minutes.We also split the Date_of_Journey column into Day,month and year inorder to further deepen our analysis.

[140]:
```
# Convert the Duration into Minutes
flight_data['Hours'].replace('None', np.nan, inplace=True)
flight_data['minutes'].replace('None', np.nan, inplace=True)
flight_data['Hours'] = flight_data['Hours'].fillna(0)
flight_data['minutes'] = flight_data['minutes'].fillna(0)

flight_data['Hours'] = pd.to_numeric(flight_data['Hours'])
flight_data['minutes'] = pd.to_numeric(flight_data['minutes'])

flight_data['total_travel_time(mins)'] =flight_data['Hours'] *60 +␣
 ↪flight_data['minutes']

flight_data
```

[140]:
```
            Airline Date_of_Journey    Source Destination  \
0            IndiGo      24/03/2019  Banglore   New Delhi
1         Air India        1/5/2019   Kolkata    Banglore
2       Jet Airways        9/6/2019     Delhi      Cochin
3            IndiGo       12/5/2019   Kolkata    Banglore
4            IndiGo        1/3/2019  Banglore   New Delhi
…              …              …         …         …
10677      Air Asia        9/4/2019   Kolkata    Banglore
10678     Air India      27/04/2019   Kolkata    Banglore
10679   Jet Airways      27/04/2019  Banglore       Delhi
10680       Vistara        1/3/2019  Banglore   New Delhi
10681     Air India        9/5/2019     Delhi      Cochin

                           Route Dep_Time    Arrival_Time Duration  \
0                   [BLR ,  DEL]    22:20  3/22/2021 1:10   2h 50m
1      [CCU ,  IXR ,  BBI ,  BLR]     5:50           13:15   7h 25m
2      [DEL ,  LKO ,  BOM ,  COK]     9:25  6/10/2021 4:25      19h
3             [CCU ,  NAG ,  BLR]    18:05           23:30   5h 25m
4             [BLR ,  NAG ,  DEL]    16:50           21:35   4h 45m
…                            …       …              …       …
10677               [CCU ,  BLR]    19:55           22:25   2h 30m
10678               [CCU ,  BLR]    20:45           23:20   2h 35m
10679               [BLR ,  DEL]     8:20           11:20       3h
10680               [BLR ,  DEL]    11:30           14:10   2h 40m
10681  [DEL ,  GOI ,  BOM ,  COK]    10:55           19:15   8h 20m

       Total_Stops Additional_Info  Price Route_City1_Code Route_City2_Code  \
0                0         No info   3897              BLR              DEL
1                2         No info   7662              CCU              IXR
2                2         No info  13882              DEL              LKO
3                1         No info   6218              CCU              NAG
4                1         No info  13302              BLR              NAG
…             …              …     …              …               …
```

```
10677           0         No info   4107             CCU                BLR
10678           0         No info   4145             CCU                BLR
10679           0         No info   7229             BLR                DEL
10680           0         No info  12648             BLR                DEL
10681           2         No info  11753             DEL               GOI

      Route_City3_Code Route_City4_Code Route_City5_Code Route_City6_Code  \
0                  NaN              NaN              NaN              NaN
1                 BBI              BLR              NaN              NaN
2                 BOM              COK              NaN              NaN
3                  BLR              NaN              NaN              NaN
4                  DEL              NaN              NaN              NaN
…                …             …              …              …
10677              NaN              NaN              NaN              NaN
10678              NaN              NaN              NaN              NaN
10679              NaN              NaN              NaN              NaN
10680              NaN              NaN              NaN              NaN
10681             BOM              COK              NaN              NaN

       Hours  minutes  total_travel_time(mins)
0          2       50                      170
1          7       25                      445
2         19        0                     1140
3          5       25                      325
4          4       45                      285
…        …       …                      …
10677      2       30                      150
10678      2       35                      155
10679      3        0                      180
10680      2       40                      160
10681      8       20                      500

[10681 rows x 20 columns]
```

[ ]:
```python
#Split the Date into Month,Day and Year columns
rslt_df1['Date']=rslt_df1['Date_of_Journey'].str.split('/').str[0]
rslt_df1['Month']=rslt_df1['Date_of_Journey'].str.split('/').str[1]
rslt_df1['Year']=rslt_df1['Date_of_Journey'].str.split('/').str[2]
rslt_df1
```

[29]:
```python
flight_data.groupby(['Source','Destination']).count()
```

[29]:
```
                      Airline  Date_of_Journey  Route  Dep_Time  Arrival_Time  \
Source   Destination
Banglore Delhi           1265             1265   1265      1265          1265
         New Delhi        932              932    932       932           932
Chennai  Kolkata          381              381    381       381           381
```

```
Delhi    Cochin          4536             4536   4536       4536           4536
Kolkata  Banglore        2871             2871   2871       2871           2871
Mumbai   Hyderabad        696              696    696        696            696

                      Duration  Total_Stops  Additional_Info  Price  \
Source   Destination
Banglore Delhi            1265         1265             1265   1265
         New Delhi         932          932              932    932
Chennai  Kolkata           381          381              381    381
Delhi    Cochin           4536         4536             4536   4536
Kolkata  Banglore         2871         2871             2871   2871
Mumbai   Hyderabad         696          696              696    696

                      Route_City1_Code  Route_City2_Code  Route_City3_Code  \
Source   Destination
Banglore Delhi                    1265              1265                 0
         New Delhi                 932               932               645
Chennai  Kolkata                   381               381                 0
Delhi    Cochin                   4536              4536              4323
Kolkata  Banglore                 2871              2871              2147
Mumbai   Hyderabad                 696               696                75

                      Route_City4_Code  Route_City5_Code  Route_City6_Code  \
Source   Destination
Banglore Delhi                       0                 0                 0
         New Delhi                  83                 8                 1
Chennai  Kolkata                     0                 0                 0
Delhi    Cochin                   1138                25                 0
Kolkata  Banglore                  313                11                 0
Mumbai   Hyderabad                  31                 2                 0

                      Mean price  Hours  minutes  total_travel_time(mins)
Source   Destination
Banglore Delhi              1265   1265     1265                     1265
         New Delhi           932    932      932                      932
Chennai  Kolkata             381    381      381                      381
Delhi    Cochin             4536   4536     4536                     4536
Kolkata  Banglore           2871   2871     2871                     2871
Mumbai   Hyderabad           696    696      696                      696
```

We add an additional column called Distance to compute the total distance between the Source and Destination,which will be useful for us. We also find the Mean Time Column to find the Mean Time between the Source and Destination when we group the data by Source and Destination.

[30]:
```
#compute the distance
```

```
dist = [['Banglore','Delhi', 1709.97], ['Chennai', 'Kolkata', 1385.64],␣
↪['Delhi','Cochin', 2048.81],['Kolkata','Banglore', 1546.77␣
↪],['Mumbai','Hyderabad', 622.81 ]]

# Create the pandas DataFrame
dist_df = pd.DataFrame(dist, columns = ['Source',␣
↪'Destination','Distance(kms)'])

flight_data = pd.merge(flight_data, dist_df, on=["Source", "Destination"])

flight_data
```

[30]:

```
        Airline Date_of_Journey   Source Destination  \
0     Air India        1/5/2019  Kolkata    Banglore
1     Air India        1/5/2019  Kolkata    Banglore
2     Air India      18/05/2019  Kolkata    Banglore
3     Air India      15/05/2019  Kolkata    Banglore
4     Air India        6/6/2019  Kolkata    Banglore
…           …               …        …           …
9744   SpiceJet      15/06/2019   Mumbai   Hyderabad
9745   SpiceJet      18/05/2019   Mumbai   Hyderabad
9746   SpiceJet      18/03/2019   Mumbai   Hyderabad
9747   SpiceJet      27/03/2019   Mumbai   Hyderabad
9748     Trujet        6/3/2019   Mumbai   Hyderabad

                          Route Dep_Time     Arrival_Time Duration  \
0     [CCU , IXR , BBI , BLR]     5:50            13:15   7h 25m
1     [CCU , GAU , DEL , BLR]     9:50            23:15  13h 25m
2                 [CCU , BLR]    14:15            16:45   2h 30m
3           [CCU , HYD , BLR]    19:00  5/16/2021 11:05   16h 5m
4           [CCU , BOM , BLR]     9:25            21:50  12h 25m
…                         …      …                …        …
9744              [BOM , HYD]    13:15            14:45   1h 30m
9745              [BOM , HYD]     5:45             7:15   1h 30m
9746              [BOM , HYD]    22:45   3/19/2021 0:10   1h 25m
9747              [BOM , HYD]     5:45             7:05   1h 20m
9748        [BOM , NDC , HYD]    13:05            16:20   3h 15m

      Total_Stops              Additional_Info  … Route_City2_Code  \
0               2                      No info  …              IXR
1               2                      No info  …              GAU
2               0                      No info  …              BLR
3               1                      No info  …              HYD
4               1                      No info  …              BOM
…             …                          …   …           …
9744            0                      No info  …              HYD
9745            0  No check-in baggage included  …              HYD
```

```
9746          0                        No info  …              HYD
9747          0                        No info  …              HYD
9748          1                        No info  …              NDC

     Route_City3_Code Route_City4_Code Route_City5_Code Route_City6_Code  \
0                 BBI              BLR              NaN              NaN
1                 DEL              BLR              NaN              NaN
2                 NaN              NaN              NaN              NaN
3                 BLR              NaN              NaN              NaN
4                 BLR              NaN              NaN              NaN
…                  …               …               …               …
9744              NaN              NaN              NaN              NaN
9745              NaN              NaN              NaN              NaN
9746              NaN              NaN              NaN              NaN
9747              NaN              NaN              NaN              NaN
9748              HYD              NaN              NaN              NaN

        Mean price Hours  minutes  total_travel_time(mins)  Distance(kms)
0     10357.324219     7       25                      445        1546.77
1     10357.324219    13       25                      805        1546.77
2     10357.324219     2       30                      150        1546.77
3     10357.324219    16        5                      965        1546.77
4     10357.324219    12       25                      745        1546.77
…              …    …     …                        …              …
9744   2511.106557     1       30                       90         622.81
9745   2511.106557     1       30                       90         622.81
9746   2511.106557     1       25                       85         622.81
9747   2511.106557     1       20                       80         622.81
9748   4140.000000     3       15                      195         622.81

[9749 rows x 22 columns]
```

[31]:
```
flight_data
flight_data['mean_time_between_these_destination(mins)'] = flight_data.
↪groupby(['Source','Destination'])['total_travel_time(mins)'].
↪transform('mean')
flight_data
```

[31]:
```
        Airline Date_of_Journey   Source Destination  \
0     Air India        1/5/2019  Kolkata    Banglore
1     Air India        1/5/2019  Kolkata    Banglore
2     Air India      18/05/2019  Kolkata    Banglore
3     Air India      15/05/2019  Kolkata    Banglore
4     Air India        6/6/2019  Kolkata    Banglore
…          …              …       …          …
9744   SpiceJet      15/06/2019   Mumbai   Hyderabad
9745   SpiceJet      18/05/2019   Mumbai   Hyderabad
```

```
9746   SpiceJet      18/03/2019   Mumbai   Hyderabad
9747   SpiceJet      27/03/2019   Mumbai   Hyderabad
9748     Trujet        6/3/2019   Mumbai   Hyderabad

                          Route Dep_Time     Arrival_Time Duration  \
0     [CCU ,  IXR ,  BBI ,  BLR]     5:50            13:15   7h 25m
1     [CCU ,  GAU ,  DEL ,  BLR]     9:50            23:15  13h 25m
2                  [CCU ,  BLR]    14:15            16:45   2h 30m
3            [CCU ,  HYD ,  BLR]    19:00  5/16/2021 11:05   16h 5m
4            [CCU ,  BOM ,  BLR]     9:25            21:50  12h 25m
…                           …        …               …       …
9744               [BOM ,  HYD]    13:15            14:45   1h 30m
9745               [BOM ,  HYD]     5:45             7:15   1h 30m
9746               [BOM ,  HYD]    22:45   3/19/2021 0:10   1h 25m
9747               [BOM ,  HYD]     5:45             7:05   1h 20m
9748         [BOM ,  NDC ,  HYD]    13:05            16:20   3h 15m

      Total_Stops              Additional_Info  …  Route_City3_Code  \
0               2                      No info  …               BBI
1               2                      No info  …               DEL
2               0                      No info  …               NaN
3               1                      No info  …               BLR
4               1                      No info  …               BLR
…             …                          …  …               …
9744            0                      No info  …               NaN
9745            0  No check-in baggage included  …               NaN
9746            0                      No info  …               NaN
9747            0                      No info  …               NaN
9748            1                      No info  …               HYD

     Route_City4_Code Route_City5_Code Route_City6_Code    Mean price Hours  \
0                 BLR              NaN              NaN  10357.324219     7
1                 BLR              NaN              NaN  10357.324219    13
2                 NaN              NaN              NaN  10357.324219     2
3                 NaN              NaN              NaN  10357.324219    16
4                 NaN              NaN              NaN  10357.324219    12
…                 …               …               …            …   …
9744              NaN              NaN              NaN   2511.106557     1
9745              NaN              NaN              NaN   2511.106557     1
9746              NaN              NaN              NaN   2511.106557     1
9747              NaN              NaN              NaN   2511.106557     1
9748              NaN              NaN              NaN   4140.000000     3

     minutes  total_travel_time(mins)  Distance(kms)  \
0         25                      445        1546.77
1         25                      805        1546.77
2         30                      150        1546.77
```

```
3          5                      965        1546.77
4         25                      745        1546.77
…       …                      …          …
9744      30                       90         622.81
9745      30                       90         622.81
9746      25                       85         622.81
9747      20                       80         622.81
9748      15                      195         622.81

      mean_time_between_these_destination(mins)
0                                   747.248346
1                                   747.248346
2                                   747.248346
3                                   747.248346
4                                   747.248346
…                                          …
9744                                191.982759
9745                                191.982759
9746                                191.982759
9747                                191.982759
9748                                191.982759

[9749 rows x 23 columns]
```

4 Data Visualisation

[133]:
```
import plotly
import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
%matplotlib inline

#count
df = flight_data.Airline.value_counts()
print(df)
```

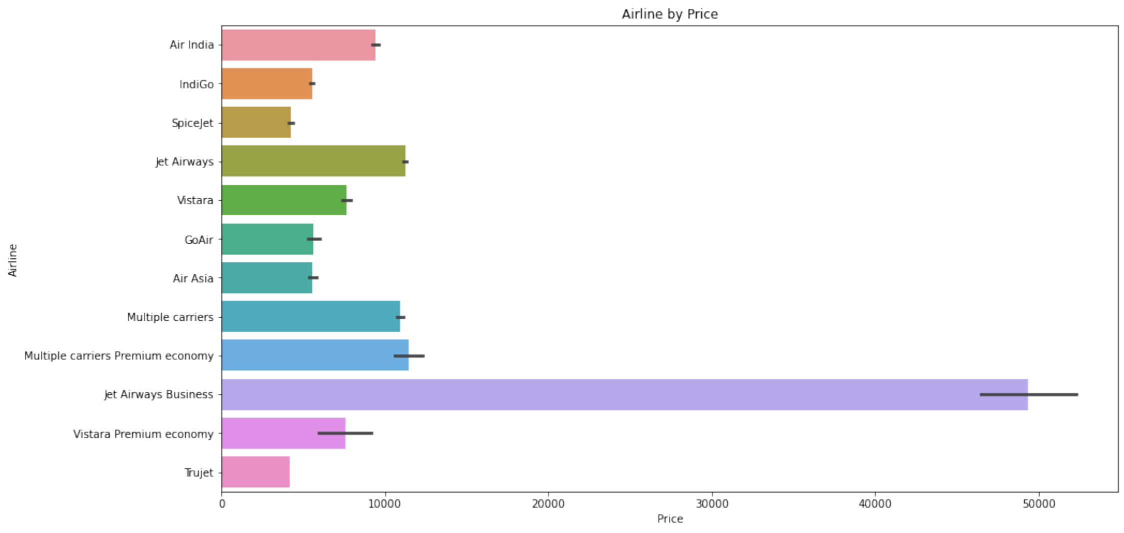
```
Jet Airways                          3849
IndiGo                               2053
Air India                            1750
Multiple carriers                    1196
SpiceJet                              818
Vistara                               479
Air Asia                              319
GoAir                                 194
```

```
Multiple carriers Premium economy      13
Jet Airways Business                    6
Vistara Premium economy                 3
Trujet                                  1
Name: Airline, dtype: int64
```

Airline by Price

[33]:
```
plt.figure(figsize = (15,8))

sns.barplot(
    x="Price",
    y="Airline",data = flight_data, orient = "h"
).set_title("Airline by Price")
```

[33]:
```
Text(0.5, 1.0, 'Airline by Price')
```



[134]:
```
plt.figure(figsize = (15,8))

px.bar(df,
    x=df.index,
    y=df.values,title = "Bar plot for the Airlines",labels=dict(x="Airlines",␣
 ↪y="Count")
)
```

```
<Figure size 1080x576 with 0 Axes>
```

[142]:
```
# selecting the top flights where count exceeds 800
rslt_df =  df[df>800]
rslt_df = rslt_df.index
```

```
rslt_df
rslt_df1 = flight_data[flight_data['Airline'].isin(rslt_df)]
rslt_df1
```

```
[142]:                  Airline Date_of_Journey    Source Destination  \
0                  IndiGo      24/03/2019  Banglore   New Delhi
1               Air India        1/5/2019   Kolkata    Banglore
2             Jet Airways        9/6/2019     Delhi      Cochin
3                  IndiGo       12/5/2019   Kolkata    Banglore
4                  IndiGo        1/3/2019  Banglore   New Delhi
…                     …             …        …         …
10675  Multiple carriers        1/5/2019     Delhi      Cochin
10676            SpiceJet      21/05/2019  Banglore       Delhi
10678           Air India      27/04/2019   Kolkata    Banglore
10679         Jet Airways      27/04/2019  Banglore       Delhi
10681           Air India        9/5/2019     Delhi      Cochin

                             Route Dep_Time    Arrival_Time Duration  \
0                     [BLR , DEL]    22:20  3/22/2021 1:10   2h 50m
1      [CCU , IXR , BBI , BLR]     5:50           13:15   7h 25m
2      [DEL , LKO , BOM , COK]     9:25  6/10/2021 4:25      19h
3             [CCU , NAG , BLR]    18:05           23:30   5h 25m
4             [BLR , NAG , DEL]    16:50           21:35   4h 45m
…                              …      …              …        …
10675         [DEL , BOM , COK]    10:20           19:00   8h 40m
10676               [BLR , DEL]     5:55            8:35   2h 40m
10678               [CCU , BLR]    20:45           23:20   2h 35m
10679               [BLR , DEL]     8:20           11:20       3h
10681  [DEL , GOI , BOM , COK]    10:55           19:15   8h 20m

       Total_Stops              Additional_Info  Price Route_City1_Code  \
0                0                      No info   3897              BLR
1                2                      No info   7662              CCU
2                2                      No info  13882              DEL
3                1                      No info   6218              CCU
4                1                      No info  13302              BLR
…              …                            …    …              …
10675            1                      No info   9794              DEL
10676            0  No check-in baggage included   3257              BLR
10678            0                      No info   4145              CCU
10679            0                      No info   7229              BLR
10681            2                      No info  11753              DEL

      Route_City2_Code Route_City3_Code Route_City4_Code Route_City5_Code  \
0                  DEL              NaN              NaN              NaN
1                  IXR              BBI              BLR              NaN
2                  LKO              BOM              COK              NaN
```

```
3                NAG              BLR              NaN              NaN
4                NAG              DEL              NaN              NaN
…                …                …                …                …
10675            BOM              COK              NaN              NaN
10676             DEL             NaN              NaN              NaN
10678             BLR             NaN              NaN              NaN
10679             DEL             NaN              NaN              NaN
10681            GOI             BOM               COK              NaN

      Route_City6_Code  Hours  minutes  total_travel_time(mins)
0                  NaN      2       50                      170
1                  NaN      7       25                      445
2                  NaN     19        0                     1140
3                  NaN      5       25                      325
4                  NaN      4       45                      285
…                …    …      …                    …
10675              NaN      8       40                      520
10676              NaN      2       40                      160
10678              NaN      2       35                      155
10679              NaN      3        0                      180
10681              NaN      8       20                      500

[9666 rows x 20 columns]
```

[143]:
```
#Using Plotly
fig = px.scatter(
    data_frame=rslt_df1,
    x="total_travel_time(mins)",
    y="Price",  trendline="ols",
    #color="Airline",
    hover_name="Airline",
    size_max=60,labels=dict(Price="Price in Rupees")
)
fig.show()
```

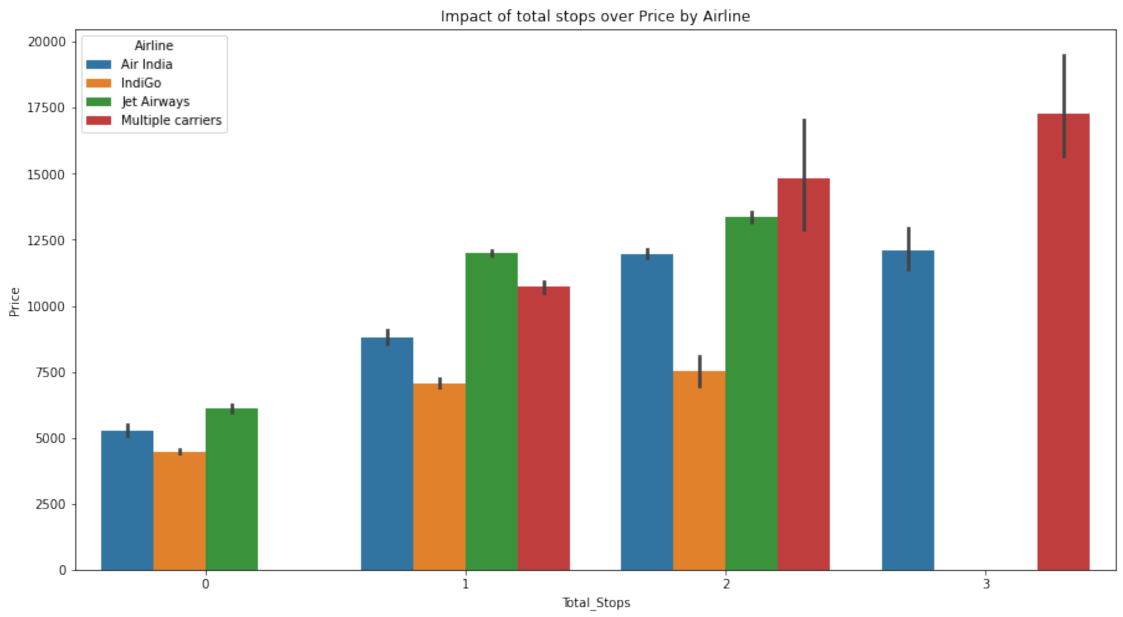
[37]:
```
#distance on price
fig = px.scatter(
    data_frame=rslt_df1,
    x="Distance(kms)",
    y="Price",
    color="Airline",
    hover_name="Airline",
    size_max=60,labels=dict(Price="Price in Rupees")
)
fig.show()
```

```
[38]: #using plotly
      plt.figure(figsize = (15,8))
      px.box(rslt_df1,
          x='Total_Stops',
          y='Price',
      )
```

```
<Figure size 1080x576 with 0 Axes>
```

```
[39]: plt.figure(figsize = (15,8))

      sns.barplot(
          x="Total_Stops",
          y="Price",hue = 'Airline',data = rslt_df1
      ).set_title("Impact of total stops over Price by Airline")
```

```
[39]: Text(0.5, 1.0, 'Impact of total stops over Price by Airline')
```



The above Visualisations show that - Price and Total travel time relationship is linear.As the Total travel time increases,the Price of the flight increases. - As the Number of Stops increases,the Price of the flight increases. - Jet Airways has the highest number of flights. - Jet Airways Business caters the flights in the high price range,whereas Trujet offers low prices.

```
[144]: #group Airline by the Destination
       # rslt_df1["row_count"] = 1
       # group_df =rslt_df1.groupby(['Airline','Destination'])['row_count'].agg('sum').
        ↪reset_index()
       # group_df
```

```
group_df = rslt_df1.groupby(['Airline','Destination']).count().reset_index()
group_df
```

```
[144]:               Airline Destination  Date_of_Journey  Source  Route  Dep_Time  \
       0            Air India    Banglore              512     512    512       512
       1            Air India      Cochin              746     746    746       746
       2            Air India       Delhi              120     120    120       120
       3            Air India   Hyderabad              135     135    135       135
       4            Air India     Kolkata               25      25     25        25
       5            Air India   New Delhi              212     212    212       212
       6               IndiGo    Banglore              445     445    445       445
       7               IndiGo      Cochin              705     705    705       705
       8               IndiGo       Delhi              366     366    366       366
       9               IndiGo   Hyderabad              196     196    196       196
       10              IndiGo     Kolkata              184     184    184       184
       11              IndiGo   New Delhi              157     157    157       157
       12         Jet Airways    Banglore             1256    1256   1256      1256
       13         Jet Airways      Cochin             1586    1586   1586      1586
       14         Jet Airways       Delhi              370     370    370       370
       15         Jet Airways   Hyderabad              219     219    219       219
       16         Jet Airways   New Delhi              418     418    418       418
       17   Multiple carriers      Cochin             1196    1196   1196      1196
       18            SpiceJet    Banglore              300     300    300       300
       19            SpiceJet      Cochin               87      87     87        87
       20            SpiceJet       Delhi              137     137    137       137
       21            SpiceJet   Hyderabad              122     122    122       122
       22            SpiceJet     Kolkata              128     128    128       128
       23            SpiceJet   New Delhi               44      44     44        44

           Arrival_Time  Duration  Total_Stops  Additional_Info  Price  \
       0            512       512          512              512    512
       1            746       746          746              746    746
       2            120       120          120              120    120
       3            135       135          135              135    135
       4             25        25           25               25     25
       5            212       212          212              212    212
       6            445       445          445              445    445
       7            705       705          705              705    705
       8            366       366          366              366    366
       9            196       196          196              196    196
       10           184       184          184              184    184
       11           157       157          157              157    157
       12          1256      1256         1256             1256   1256
       13          1586      1586         1586             1586   1586
       14           370       370          370              370    370
       15           219       219          219              219    219
```

```
16          418       418          418               418   418
17         1196      1196         1196              1196  1196
18          300       300          300               300   300
19           87        87           87                87    87
20          137       137          137               137   137
21          122       122          122               122   122
22          128       128          128               128   128
23           44        44           44                44    44

    Route_City1_Code  Route_City2_Code  Route_City3_Code  Route_City4_Code  \
0                512               512               451               299
1                746               746               671               390
2                120               120                 0                 0
3                135               135                37                24
4                 25                25                 0                 0
5                212               212               174                80
6                445               445               125                 1
7                705               705               646                18
8                366               366                 0                 0
9                196               196                 1                 0
10               184               184                 0                 0
11               157               157                40                 0
12              1256              1256              1256                 4
13              1586              1586              1552               677
14               370               370                 0                 0
15               219               219                12                 7
16               418               418               406                 3
17              1196              1196              1196                51
18               300               300                52                 0
19                87                87                87                 0
20               137               137                 0                 0
21               122               122                 1                 0
22               128               128                 0                 0
23                44                44                 8                 0

    Route_City5_Code  Route_City6_Code  Hours  minutes  \
0                 11                 0    512      512
1                 17                 0    746      746
2                  0                 0    120      120
3                  2                 0    135      135
4                  0                 0     25       25
5                  8                 1    212      212
6                  0                 0    445      445
7                  0                 0    705      705
8                  0                 0    366      366
9                  0                 0    196      196
10                 0                 0    184      184
```

```
11                  0                  0   157    157
12                  0                  0  1256   1256
13                  0                  0  1586   1586
14                  0                  0   370    370
15                  0                  0   219    219
16                  0                  0   418    418
17                  8                  0  1196   1196
18                  0                  0   300    300
19                  0                  0    87     87
20                  0                  0   137    137
21                  0                  0   122    122
22                  0                  0   128    128
23                  0                  0    44     44

    total_travel_time(mins)
0                       512
1                       746
2                       120
3                       135
4                        25
5                       212
6                       445
7                       705
8                       366
9                       196
10                      184
11                      157
12                     1256
13                     1586
14                      370
15                      219
16                      418
17                     1196
18                      300
19                       87
20                      137
21                      122
22                      128
23                       44
```

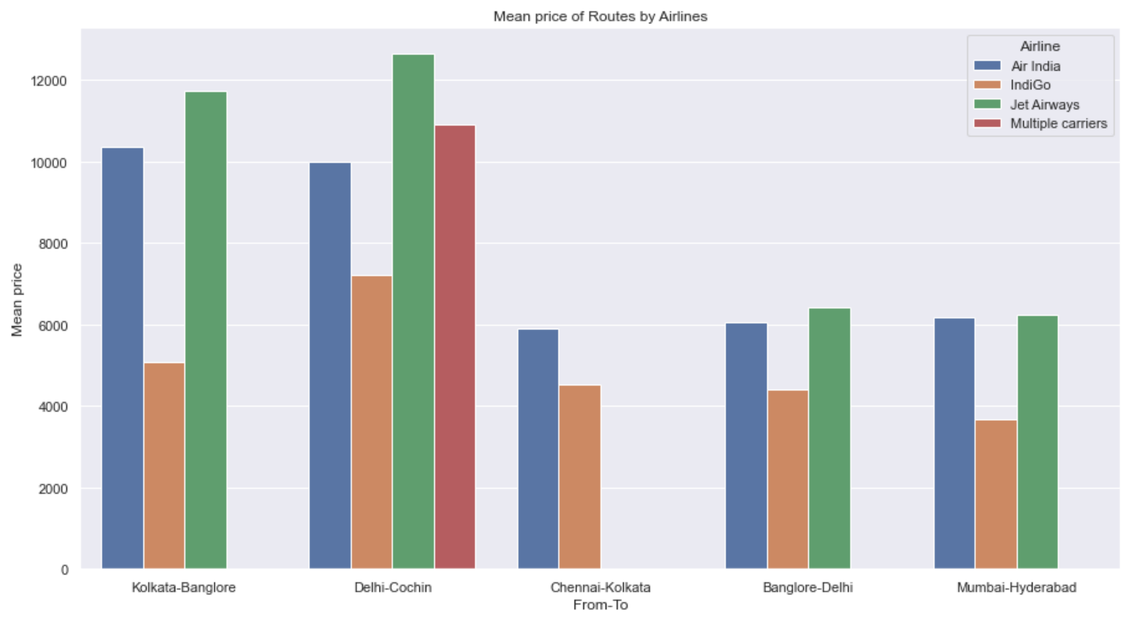
[47]:
```
#Using plotly to plot the Destinations as per Airlines
plt.figure(figsize = (15,8))

px.bar(group_df,
    x='Airline',
    y='row_count',color = 'Destination',barmode = 'group',title = "Airline by␣
 ↪their Destinations"
)
```

```
<Figure size 1080x576 with 0 Axes>
```

```
[ ]: #rslt_df1["row_count"] = 1

     rslt_df1['From-To'] = rslt_df1['Source'].str.
      ↪cat(rslt_df1['Destination'],sep="-")
     rslt_df1
```

```
[53]: #Using seaborn
      sns.set(rc = {'figure.figsize':(15,8)})
      sns.barplot(data=rslt_df1,
          x='From-To',
          y='Mean price',hue = 'Airline').set_title('Mean price of Routes by␣
       ↪Airlines')
```

```
[53]: Text(0.5, 1.0, 'Mean price of Routes by Airlines')
```



The above Visualisations show that

- Jet Airways offers maximum flights to destination Cochin,followed by Bangalore and Delhi.
- The most travelled Destination among all the flight carriers is Cochin
- Delhi-Cochin has the highest mean price compared to other locations.
- Jet Airways has the highest Mean Price to Delhi-Cochin location.

5 Regression Techniques

[56]:
```
#import pandas
import sklearn
from sklearn import linear_model
```

We encode the string data in this dataset, to convert it to integer type, since the regression model does not work on string data. We use 'Label Encoder' to achieve the desired results.Thus our transformed data looks like this.

[115]:
```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
rslt_df1["Airline"]=encoder.fit_transform(rslt_df1['Airline'])
rslt_df1["Source"]=encoder.fit_transform(rslt_df1['Source'])
rslt_df1["Destination"]=encoder.fit_transform(rslt_df1['Destination'])
rslt_df1["Additional_Info"]=encoder.fit_transform(rslt_df1['Additional_Info'])
rslt_df1["Route_City1_Code"]=encoder.fit_transform(rslt_df1['Route_City1_Code'])
rslt_df1["Route_City2_Code"]=encoder.fit_transform(rslt_df1['Route_City2_Code'])
rslt_df1["Route_City3_Code"]=encoder.fit_transform(rslt_df1['Route_City3_Code'])
rslt_df1["Route_City4_Code"]=encoder.fit_transform(rslt_df1['Route_City4_Code'])
rslt_df1["Route_City5_Code"]=encoder.fit_transform(rslt_df1['Route_City5_Code'])
rslt_df1["Route_City6_Code"]=encoder.fit_transform(rslt_df1['Route_City6_Code'])


rslt_df1['Date']=rslt_df1['Date'].astype(int)
rslt_df1['Month']=rslt_df1['Month'].astype(int)
rslt_df1['Year']=rslt_df1['Year'].astype(int)
```

[145]:
```
rslt_df1
```

[145]:
```
                  Airline Date_of_Journey    Source Destination  \
0                  IndiGo      24/03/2019  Banglore   New Delhi
1               Air India        1/5/2019   Kolkata    Banglore
2             Jet Airways        9/6/2019     Delhi      Cochin
3                  IndiGo       12/5/2019   Kolkata    Banglore
4                  IndiGo        1/3/2019  Banglore   New Delhi
…                       …               …         …         …
10675  Multiple carriers        1/5/2019     Delhi      Cochin
10676            SpiceJet      21/05/2019  Banglore       Delhi
10678           Air India      27/04/2019   Kolkata    Banglore
10679         Jet Airways      27/04/2019  Banglore       Delhi
10681           Air India        9/5/2019     Delhi      Cochin

                           Route Dep_Time    Arrival_Time Duration  \
0                    [BLR ,  DEL]    22:20  3/22/2021 1:10   2h 50m
1      [CCU ,  IXR ,  BBI ,  BLR]     5:50           13:15   7h 25m
2      [DEL ,  LKO ,  BOM ,  COK]     9:25  6/10/2021 4:25      19h
3             [CCU ,  NAG ,  BLR]    18:05           23:30   5h 25m
```

```
4             [BLR ,  NAG ,  DEL]    16:50          21:35   4h 45m
…                            …      …           …      …
10675         [DEL ,  BOM ,  COK]    10:20          19:00   8h 40m
10676                [BLR ,  DEL]     5:55           8:35   2h 40m
10678                [CCU ,  BLR]    20:45          23:20   2h 35m
10679                [BLR ,  DEL]     8:20          11:20       3h
10681  [DEL ,  GOI ,  BOM ,  COK]    10:55          19:15   8h 20m

       Total_Stops               Additional_Info  Price Route_City1_Code  \
0                0                       No info   3897              BLR
1                2                       No info   7662              CCU
2                2                       No info  13882              DEL
3                1                       No info   6218              CCU
4                1                       No info  13302              BLR
…            …                           …    …              …
10675            1                       No info   9794              DEL
10676            0  No check-in baggage included   3257              BLR
10678            0                       No info   4145              CCU
10679            0                       No info   7229              BLR
10681            2                       No info  11753              DEL

      Route_City2_Code Route_City3_Code Route_City4_Code Route_City5_Code  \
0                  DEL              NaN              NaN              NaN
1                 IXR              BBI              BLR              NaN
2                 LKO              BOM              COK              NaN
3                 NAG              BLR              NaN              NaN
4                 NAG              DEL              NaN              NaN
…                 …              …              …              …
10675             BOM              COK              NaN              NaN
10676              DEL              NaN              NaN              NaN
10678              BLR              NaN              NaN              NaN
10679              DEL              NaN              NaN              NaN
10681             GOI              BOM              COK              NaN

      Route_City6_Code  Hours  minutes  total_travel_time(mins)
0                  NaN      2       50                      170
1                  NaN      7       25                      445
2                  NaN     19        0                     1140
3                  NaN      5       25                      325
4                  NaN      4       45                      285
…                 …    …      …                      …
10675              NaN      8       40                      520
10676              NaN      2       40                      160
10678              NaN      2       35                      155
10679              NaN      3        0                      180
10681              NaN      8       20                      500
```

```
[9666 rows x 20 columns]
```

[123]:
```
# We drop the other columns and keep only those columns which are relevant to␣
↪us.
rslt_df1=rslt_df1.drop(['Route'],axis = 1)
rslt_df1=rslt_df1.drop(['Date_of_Journey'],axis = 1)
rslt_df1=rslt_df1.drop('Arrival_Time',axis=1)
rslt_df1=rslt_df1.drop('Dep_Time',axis=1)
rslt_df1=rslt_df1.drop('Duration',axis=1)
rslt_df1=rslt_df1.drop('Route_City1',axis=1)
rslt_df1=rslt_df1.drop('Route_City2',axis=1)
rslt_df1=rslt_df1.drop('Route_City3',axis=1)
rslt_df1=rslt_df1.drop('Route_City4',axis=1)
rslt_df1=rslt_df1.drop('Route_City5',axis=1)
rslt_df1=rslt_df1.drop('Route_City6',axis=1)
rslt_df1=rslt_df1.drop('From-To',axis=1)
rslt_df1.head()

rslt_df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8061 entries, 0 to 9625
Data columns (total 22 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Airline                                 8061 non-null   int64
 1   Source                                  8061 non-null   int64
 2   Destination                             8061 non-null   int64
 3   Total_Stops                             8061 non-null   int32
 4   Additional_Info                         8061 non-null   int64
 5   Price                                   8061 non-null   int64
 6   Route_City1_Code                        8061 non-null   int64
 7   Route_City2_Code                        8061 non-null   int64
 8   Route_City3_Code                        8061 non-null   int64
 9   Route_City4_Code                        8061 non-null   int64
 10  Route_City5_Code                        8061 non-null   int64
 11  Route_City6_Code                        8061 non-null   int64
 12  Mean price                              8061 non-null   float64
 13  Hours                                   8061 non-null   int64
 14  minutes                                 8061 non-null   int64
 15  total_travel_time(mins)                 8061 non-null   int64
 16  Distance(kms)                           8061 non-null   float64
 17  mean_time_between_these_destination(mins)  8061 non-null   float64
 18  row_count                               8061 non-null   int64
 19  Date                                    8061 non-null   int32
 20  Month                                   8061 non-null   int32
 21  Year                                    8061 non-null   int32
dtypes: float64(3), int32(4), int64(15)
```

```
memory usage: 1.5 MB
```

Since the Target variable in this dataset is Price, we remove this column and separate into X and y variables.

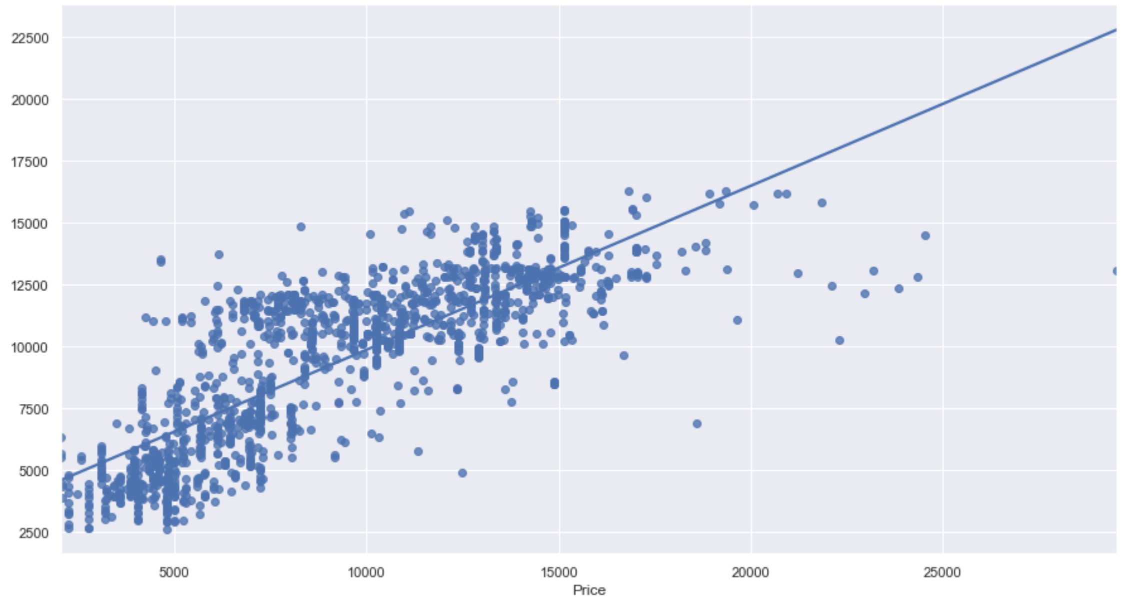
[146]:
```
X=rslt_df1.drop('Price',axis=1)
y=rslt_df1['Price']
rslt_df1.head()
```

[146]:
```
       Airline Date_of_Journey    Source Destination  \
0       IndiGo      24/03/2019  Banglore   New Delhi
1    Air India        1/5/2019   Kolkata    Banglore
2  Jet Airways        9/6/2019     Delhi      Cochin
3       IndiGo       12/5/2019   Kolkata    Banglore
4       IndiGo        1/3/2019  Banglore   New Delhi

                       Route Dep_Time    Arrival_Time Duration  Total_Stops  \
0              [BLR , DEL]    22:20  3/22/2021 1:10   2h 50m            0
1  [CCU , IXR , BBI , BLR]     5:50           13:15   7h 25m            2
2  [DEL , LKO , BOM , COK]     9:25  6/10/2021 4:25      19h            2
3        [CCU , NAG , BLR]    18:05           23:30   5h 25m            1
4        [BLR , NAG , DEL]    16:50           21:35   4h 45m            1

  Additional_Info  Price Route_City1_Code Route_City2_Code Route_City3_Code  \
0         No info   3897              BLR              DEL              NaN
1         No info   7662              CCU              IXR              BBI
2         No info  13882              DEL              LKO              BOM
3         No info   6218              CCU              NAG              BLR
4         No info  13302              BLR              NAG              DEL

  Route_City4_Code Route_City5_Code Route_City6_Code  Hours  minutes  \
0              NaN              NaN              NaN      2       50
1              BLR              NaN              NaN      7       25
2              COK              NaN              NaN     19        0
3              NaN              NaN              NaN      5       25
4              NaN              NaN              NaN      4       45

   total_travel_time(mins)
0                      170
1                      445
2                     1140
3                      325
4                      285
```

We then split our dataset into training and test sets using the train_test_split from the sklearn package.

[126]:
```
    from sklearn.linear_model import LinearRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import r2_score,mean_squared_error
    import seaborn as sns

    lr = LinearRegression()


    x_train,x_test,y_train,y_test = train_test_split(X,y,test_size =0.
 ↪2,random_state=42)

    lr.fit(x_train,y_train)
    pred = lr.predict(x_test)
    test_score = r2_score(y_test,pred)
    train_score = r2_score(y_train,lr.predict(x_train))
    #if(abs(train_score - test_score)) <= 0.1:
      # print(i)
    print('R2 for train data',r2_score(y_train,lr.predict(x_train)))
    print('R2 score for test set is',r2_score(y_test,pred))
    print('root mean square␣
 ↪error(RMSE)',(mean_squared_error(y_test,pred,squared=False)))
    sns.set(rc = {'figure.figsize':(15,8)})
    sns.regplot(x=y_test ,y = pred, ci=None )
    plt.figure(figsize = (15,8))
        #plt.scatter(y_test,pred)
    plt.show()
```

```
R2 for train data 0.6590856688600759
R2 score for test set is 0.6615824384934548
root mean square error(RMSE) 2356.2256725740563
```

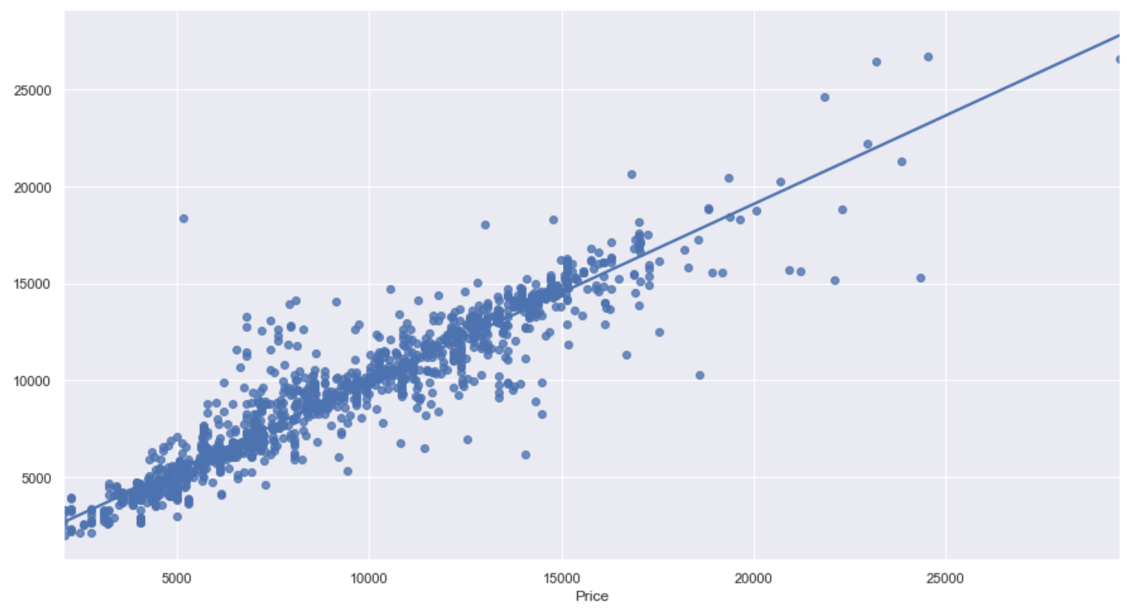


```
<Figure size 1080x576 with 0 Axes>
```

As we can see the R-square value for train and test set is 0.66 and 0.66 respectively through Linear Regression. We can see if the performance can be further enhanced using the ensemble models.

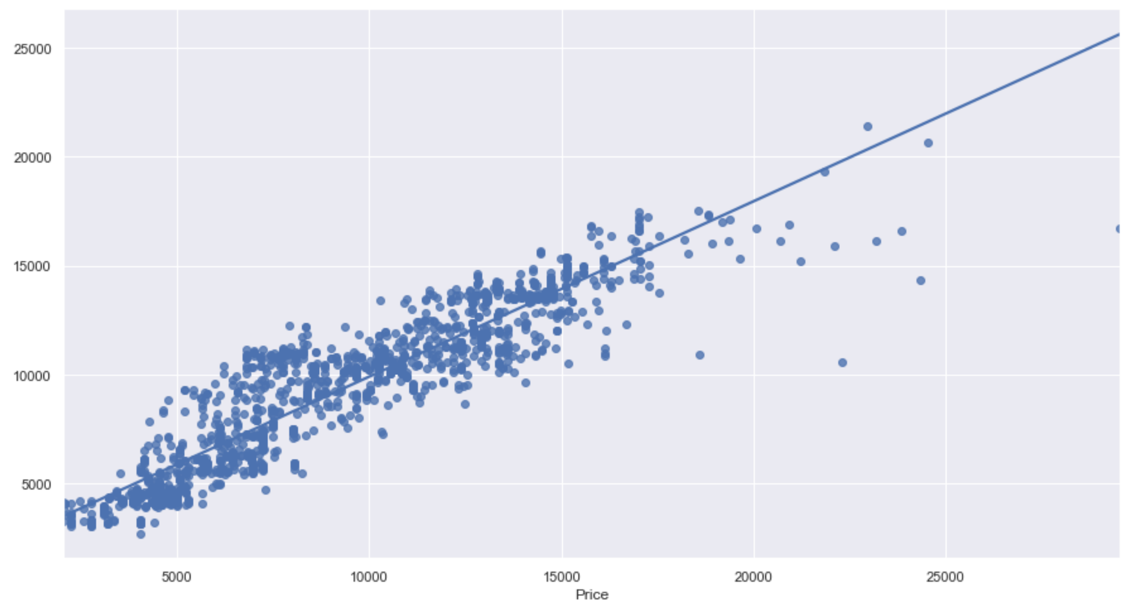
```
[127]: from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_squared_error
from sklearn.ensemble import␣
 ↪AdaBoostRegressor,RandomForestRegressor,GradientBoostingRegressor
import seaborn as sns


rf = RandomForestRegressor()
gd = GradientBoostingRegressor()

x_train,x_test,y_train,y_test = train_test_split(X,y,test_size =0.
 ↪2,random_state=42)
for i in [ rf,gd]:
    i.fit(x_train,y_train)
    pred = i.predict(x_test)
    test_score = r2_score(y_test,pred)
    train_score = r2_score(y_train,i.predict(x_train))
    if(abs(train_score - test_score)) <= 0.1:
       print(i)
       print('R2 for train data',r2_score(y_train,i.predict(x_train)))
       print('R2 score for test set is',r2_score(y_test,pred))
       print('root mean square␣
 ↪error(RMSE)',(mean_squared_error(y_test,pred,squared=False)))
       sns.set(rc = {'figure.figsize':(15,8)})
       sns.regplot(x=y_test ,y = pred, ci=None )
       plt.figure(figsize = (15,8))
       #plt.scatter(y_test,pred)
       plt.show()
```

```
RandomForestRegressor()
R2 for train data 0.9731055440673636
R2 score for test set is 0.8997376469718773
root mean square error(RMSE) 1282.5059364836204
```



```
<Figure size 1080x576 with 0 Axes>
```

```
GradientBoostingRegressor()
R2 for train data 0.8397071011101482
R2 score for test set is 0.8499740586844826
root mean square error(RMSE) 1568.8218083574734
```



```
<Figure size 1080x576 with 0 Axes>
```

As we can see,the R square score improves significantly using the ensemble models. # RandomForestRegressor() model gives the R2 score of about 90% .The corresponding RMSE score also improves to 1647.