PROJECT_2

June 13, 2023

#PROBLEM STATEMENT : WHICH MODEL IS SUITABLE FOR THE GIVEN DATASET_FLIGHT PRICE PREDICTION

Importing Packages

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Read the Data

[5]:
```
traindf=pd.read_csv(r"/content/Data_Train.csv")
```

[6]:
```
traindf
```

[6]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop
…	…	…	…	…	…	…	…	…	…
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU → BLR	19:55	22:25	2h 30m	non-stop
10679	Air India	27/04/2019	Kolkata	Banglore	CCU → BLR	20:45	23:20	2h 35m	non-stop
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR → DEL	08:20	11:20	3h	non-stop
10681	Vistara	01/03/2019	Banglore	New Delhi					
10682	Air India	9/05/2019	Delhi	Cochin					

```
10681              BLR → DEL    11:30         14:10   2h 40m    non-stop
10682  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
…                …    …
10678         No info   4107
10679         No info   4145
10680         No info   7229
10681         No info  12648
10682         No info  11753

[10683 rows x 11 columns]
```

[7]:
```
testdf=pd.read_csv(r"/content/Test_set.csv")
```

[8]:
```
testdf
```

[8]:
```
                Airline Date_of_Journey    Source Destination  \
0           Jet Airways       6/06/2019     Delhi      Cochin
1                IndiGo      12/05/2019   Kolkata    Banglore
2           Jet Airways      21/05/2019     Delhi      Cochin
3     Multiple carriers      21/05/2019     Delhi      Cochin
4              Air Asia      24/06/2019  Banglore       Delhi
…                   …              …       …          …
2666         Air India       6/06/2019   Kolkata    Banglore
2667           IndiGo      27/03/2019   Kolkata    Banglore
2668        Jet Airways       6/03/2019     Delhi      Cochin
2669         Air India       6/03/2019     Delhi      Cochin
2670  Multiple carriers      15/06/2019     Delhi      Cochin

                Route Dep_Time  Arrival_Time Duration Total_Stops  \
0     DEL → BOM → COK    17:30  04:25 07 Jun  10h 55m      1 stop
1     CCU → MAA → BLR    06:20         10:20       4h      1 stop
2     DEL → BOM → COK    19:15  19:00 22 May  23h 45m      1 stop
3     DEL → BOM → COK    08:00         21:00      13h      1 stop
4           BLR → DEL    23:55  02:45 25 Jun   2h 50m    non-stop
…                   …      …            …       …         …
2666  CCU → DEL → BLR    20:30  20:25 07 Jun  23h 55m      1 stop
2667        CCU → BLR    14:20         16:55   2h 35m    non-stop
2668  DEL → BOM → COK    21:50  04:25 07 Mar   6h 35m      1 stop
2669  DEL → BOM → COK    04:00         19:15  15h 15m      1 stop
2670  DEL → BOM → COK    04:55         19:15  14h 20m      1 stop
```

```
                  Additional_Info
0                         No info
1                         No info
2     In-flight meal not included
3                         No info
4                         No info
…                           …
2666                      No info
2667                      No info
2668                      No info
2669                      No info
2670                      No info

[2671 rows x 10 columns]
```

Data Collection and Preprocessing

[9]:
```
traindf.head()
```

[9]:
```
       Airline Date_of_Journey    Source Destination                  Route  \
0       IndiGo      24/03/2019  Banglore   New Delhi              BLR → DEL
1    Air India       1/05/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR
2  Jet Airways       9/06/2019     Delhi      Cochin  DEL → LKO → BOM → COK
3       IndiGo      12/05/2019   Kolkata    Banglore        CCU → NAG → BLR
4       IndiGo      01/03/2019  Banglore   New Delhi        BLR → NAG → DEL

  Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price
0    22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897
1    05:50         13:15   7h 25m     2 stops         No info   7662
2    09:25  04:25 10 Jun      19h     2 stops         No info  13882
3    18:05         23:30   5h 25m      1 stop         No info   6218
4    16:50         21:35   4h 45m      1 stop         No info  13302
```

[10]:
```
testdf.head()
```

[10]:
```
              Airline Date_of_Journey    Source Destination            Route  \
0        Jet Airways       6/06/2019     Delhi      Cochin  DEL → BOM → COK
1             IndiGo      12/05/2019   Kolkata    Banglore  CCU → MAA → BLR
2        Jet Airways      21/05/2019     Delhi      Cochin  DEL → BOM → COK
3  Multiple carriers      21/05/2019     Delhi      Cochin  DEL → BOM → COK
4           Air Asia      24/06/2019  Banglore       Delhi        BLR → DEL

  Dep_Time  Arrival_Time Duration Total_Stops              Additional_Info
0    17:30  04:25 07 Jun  10h 55m      1 stop                     No info
1    06:20         10:20       4h      1 stop                     No info
2    19:15  19:00 22 May  23h 45m      1 stop  In-flight meal not included
```

```
3    08:00         21:00      13h      1 stop                    No info
4    23:55  02:45 25 Jun   2h 50m    non-stop                    No info
```

[11]:
```
traindf.tail()
```

[11]:
```
           Airline Date_of_Journey    Source Destination  \
10678     Air Asia       9/04/2019   Kolkata    Banglore
10679    Air India      27/04/2019   Kolkata    Banglore
10680  Jet Airways      27/04/2019  Banglore       Delhi
10681      Vistara      01/03/2019  Banglore   New Delhi
10682    Air India       9/05/2019     Delhi      Cochin

                      Route Dep_Time Arrival_Time Duration Total_Stops  \
10678              CCU → BLR    19:55        22:25   2h 30m    non-stop
10679              CCU → BLR    20:45        23:20   2h 35m    non-stop
10680              BLR → DEL    08:20        11:20       3h    non-stop
10681              BLR → DEL    11:30        14:10   2h 40m    non-stop
10682  DEL → GOI → BOM → COK    10:55        19:15   8h 20m     2 stops

      Additional_Info  Price
10678         No info   4107
10679         No info   4145
10680         No info   7229
10681         No info  12648
10682         No info  11753
```

[12]:
```
testdf.tail()
```

[12]:
```
               Airline Date_of_Journey   Source Destination            Route  \
2666         Air India       6/06/2019  Kolkata    Banglore  CCU → DEL → BLR
2667           IndiGo      27/03/2019  Kolkata    Banglore        CCU → BLR
2668       Jet Airways       6/03/2019    Delhi      Cochin  DEL → BOM → COK
2669         Air India       6/03/2019    Delhi      Cochin  DEL → BOM → COK
2670  Multiple carriers      15/06/2019    Delhi      Cochin  DEL → BOM → COK

     Dep_Time  Arrival_Time Duration Total_Stops Additional_Info
2666    20:30  20:25 07 Jun  23h 55m      1 stop         No info
2667    14:20         16:55   2h 35m    non-stop         No info
2668    21:50  04:25 07 Mar   6h 35m      1 stop         No info
2669    04:00         19:15  15h 15m      1 stop         No info
2670    04:55         19:15  14h 20m      1 stop         No info
```

[13]:
```
traindf.describe()
```

[13]:
```
              Price
count  10683.000000
mean    9087.064121
```

```
std     4611.359167
min     1759.000000
25%     5277.000000
50%     8372.000000
75%    12373.000000
max    79512.000000
```

[14]:
```
testdf.describe()
```

[14]:
```
            Airline Date_of_Journey Source Destination            Route  \
count          2671            2671   2671        2671             2671
unique           11              44      5           6              100
top     Jet Airways       9/05/2019  Delhi      Cochin  DEL → BOM → COK
freq            897             144   1145        1145              624

       Dep_Time Arrival_Time Duration Total_Stops Additional_Info
count      2671         2671     2671        2671            2671
unique      199          704      320           5               6
top       10:00        19:00   2h 50m      1 stop         No info
freq         62          113      122        1431            2148
```

[15]:
```
traindf.shape
```

[15]:
```
(10683, 11)
```

[16]:
```
testdf.shape
```

[16]:
```
(2671, 10)
```

[17]:
```
traindf.columns
```

[17]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

[18]:
```
testdf.columns
```

[18]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info'],
      dtype='object')
```

[19]:
```
traindf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
```

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

[20]:
```
testdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2671 entries, 0 to 2670
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          2671 non-null   object
 1   Date_of_Journey  2671 non-null   object
 2   Source           2671 non-null   object
 3   Destination      2671 non-null   object
 4   Route            2671 non-null   object
 5   Dep_Time         2671 non-null   object
 6   Arrival_Time     2671 non-null   object
 7   Duration         2671 non-null   object
 8   Total_Stops      2671 non-null   object
 9   Additional_Info  2671 non-null   object
dtypes: object(10)
memory usage: 208.8+ KB
```

Checking whether there are any null values in the dataset

[21]:
```
traindf.isnull().sum()
```

[21]:
```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
Arrival_Time       0
Duration           0
```

```
Total_Stops        1
Additional_Info    0
Price              0
dtype: int64
```

[22]:
```
testdf.isnull().sum()
```

[22]:
```
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
dtype: int64
```

Removing Null Values from the dataset

[23]:
```
traindf.dropna(inplace=True)
```

[24]:
```
traindf.isnull().sum()
```

[24]:
```
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
dtype: int64
```

[25]:
```
traindf.shape
```

[25]:
```
(10682, 11)
```

Conversion of datatype of values from String to Numerical Values

[26]:
```
traindf['Airline'].value_counts()
```

[26]:
```
Jet Airways                          3849
IndiGo                               2053
```

```
Air India                                1751
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
```

[27]:
```
traindf['Source'].value_counts()
```

[27]:
```
Delhi       4536
Kolkata     2871
Banglore    2197
Mumbai       697
Chennai      381
Name: Source, dtype: int64
```

[28]:
```
traindf['Destination'].value_counts()
```

[28]:
```
Cochin       4536
Banglore     2871
Delhi        1265
New Delhi     932
Hyderabad     697
Kolkata       381
Name: Destination, dtype: int64
```

[29]:
```
traindf['Total_Stops'].value_counts()
```

[29]:
```
1 stop      5625
non-stop    3491
2 stops     1520
3 stops       45
4 stops        1
Name: Total_Stops, dtype: int64
```

[30]:
```
airline={"Airline":{"Jet Airways":0,"IndiGo":1,"Air India":2,"Multiple␣
 ↪carriers":3,
 "SpiceJet":4,"Vistara":5,"Air Asia":6,"GoAir":7,
 "Multiple carriers Premium economy":8,
"Jet Airways Business":9,"Vistara Premium economy":10,"Trujet":11}}
traindf=traindf.replace(airline)
traindf
```

[30]:
```
       Airline Date_of_Journey    Source Destination                  Route  \
0            1      24/03/2019  Banglore   New Delhi              BLR → DEL
1            2       1/05/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR
2            0       9/06/2019     Delhi      Cochin  DEL → LKO → BOM → COK
3            1      12/05/2019   Kolkata    Banglore        CCU → NAG → BLR
4            1      01/03/2019  Banglore   New Delhi        BLR → NAG → DEL
…          …              …        …          …                      …
10678        6       9/04/2019   Kolkata    Banglore             CCU → BLR
10679        2      27/04/2019   Kolkata    Banglore             CCU → BLR
10680        0      27/04/2019  Banglore       Delhi             BLR → DEL
10681        5      01/03/2019  Banglore   New Delhi             BLR → DEL
10682        2       9/05/2019     Delhi      Cochin  DEL → GOI → BOM → COK

      Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price
0        22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897
1        05:50         13:15   7h 25m     2 stops         No info   7662
2        09:25  04:25 10 Jun      19h     2 stops         No info  13882
3        18:05         23:30   5h 25m      1 stop         No info   6218
4        16:50         21:35   4h 45m      1 stop         No info  13302
…        …            …      …          …              …     …
10678    19:55         22:25   2h 30m    non-stop         No info   4107
10679    20:45         23:20   2h 35m    non-stop         No info   4145
10680    08:20         11:20       3h    non-stop         No info   7229
10681    11:30         14:10   2h 40m    non-stop         No info  12648
10682    10:55         19:15   8h 20m     2 stops         No info  11753

[10682 rows x 11 columns]
```

[31]:
```
city={"Source":{"Delhi":0,"Kolkata":1,"Banglore":2,
 "Mumbai":3,"Chennai":4}}
traindf=traindf.replace(city)
traindf
```

[31]:
```
       Airline Date_of_Journey  Source Destination                  Route  \
0            1      24/03/2019       2   New Delhi              BLR → DEL
1            2       1/05/2019       1    Banglore  CCU → IXR → BBI → BLR
2            0       9/06/2019       0      Cochin  DEL → LKO → BOM → COK
3            1      12/05/2019       1    Banglore        CCU → NAG → BLR
4            1      01/03/2019       2   New Delhi        BLR → NAG → DEL
…          …              …      …         …                      …
10678        6       9/04/2019       1    Banglore             CCU → BLR
10679        2      27/04/2019       1    Banglore             CCU → BLR
10680        0      27/04/2019       2       Delhi             BLR → DEL
10681        5      01/03/2019       2   New Delhi             BLR → DEL
10682        2       9/05/2019       0      Cochin  DEL → GOI → BOM → COK

      Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price
```

```
0        22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897
1        05:50         13:15   7h 25m     2 stops         No info   7662
2        09:25  04:25 10 Jun      19h     2 stops         No info  13882
3        18:05         23:30   5h 25m      1 stop         No info   6218
4        16:50         21:35   4h 45m      1 stop         No info  13302
…          …            …       …          …              …     …
10678    19:55         22:25   2h 30m    non-stop         No info   4107
10679    20:45         23:20   2h 35m    non-stop         No info   4145
10680    08:20         11:20      3h     non-stop         No info   7229
10681    11:30         14:10   2h 40m    non-stop         No info  12648
10682    10:55         19:15   8h 20m     2 stops         No info  11753

[10682 rows x 11 columns]
```

[32]:
```
destination={"Destination":{"Cochin":0,"Banglore":1,"Delhi":2,
 "New Delhi":3,"Hyderabad":4,"Kolkata":5}}
traindf=traindf.replace(destination)
traindf
```

[32]:
```
       Airline Date_of_Journey  Source  Destination                  Route  \
0            1      24/03/2019       2            3              BLR → DEL
1            2       1/05/2019       1            1  CCU → IXR → BBI → BLR
2            0       9/06/2019       0            0  DEL → LKO → BOM → COK
3            1      12/05/2019       1            1        CCU → NAG → BLR
4            1      01/03/2019       2            3        BLR → NAG → DEL
…          …              …       …          …                        …
10678        6       9/04/2019       1            1              CCU → BLR
10679        2      27/04/2019       1            1              CCU → BLR
10680        0      27/04/2019       2            2              BLR → DEL
10681        5      01/03/2019       2            3              BLR → DEL
10682        2       9/05/2019       0            0  DEL → GOI → BOM → COK

      Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price
0        22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897
1        05:50         13:15   7h 25m     2 stops         No info   7662
2        09:25  04:25 10 Jun      19h     2 stops         No info  13882
3        18:05         23:30   5h 25m      1 stop         No info   6218
4        16:50         21:35   4h 45m      1 stop         No info  13302
…          …            …       …          …              …     …
10678    19:55         22:25   2h 30m    non-stop         No info   4107
10679    20:45         23:20   2h 35m    non-stop         No info   4145
10680    08:20         11:20      3h     non-stop         No info   7229
10681    11:30         14:10   2h 40m    non-stop         No info  12648
10682    10:55         19:15   8h 20m     2 stops         No info  11753

[10682 rows x 11 columns]
```

[33]:
```
stops={"Total_Stops":{"non-stop":0,"1 stop":1,"2 stops":2,
 "3 stops":3,"4 stops":4}}
traindf=traindf.replace(stops)
traindf
```

[33]:

	Airline	Date_of_Journey	Source	Destination	Route	\
0	1	24/03/2019	2	3	BLR → DEL	
1	2	1/05/2019	1	1	CCU → IXR → BBI → BLR	
2	0	9/06/2019	0	0	DEL → LKO → BOM → COK	
3	1	12/05/2019	1	1	CCU → NAG → BLR	
4	1	01/03/2019	2	3	BLR → NAG → DEL	
…	…	…	…	…	…	
10678	6	9/04/2019	1	1	CCU → BLR	
10679	2	27/04/2019	1	1	CCU → BLR	
10680	0	27/04/2019	2	2	BLR → DEL	
10681	5	01/03/2019	2	3	BLR → DEL	
10682	2	9/05/2019	0	0	DEL → GOI → BOM → COK	

	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	22:20	01:10 22 Mar	2h 50m	0	No info	3897
1	05:50	13:15	7h 25m	2	No info	7662
2	09:25	04:25 10 Jun	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
…	…	…	…	…	…	…
10678	19:55	22:25	2h 30m	0	No info	4107
10679	20:45	23:20	2h 35m	0	No info	4145
10680	08:20	11:20	3h	0	No info	7229
10681	11:30	14:10	2h 40m	0	No info	12648
10682	10:55	19:15	8h 20m	2	No info	11753

[10682 rows x 11 columns]

[34]:
```
traindf
```

[34]:

	Airline	Date_of_Journey	Source	Destination	Route	\
0	1	24/03/2019	2	3	BLR → DEL	
1	2	1/05/2019	1	1	CCU → IXR → BBI → BLR	
2	0	9/06/2019	0	0	DEL → LKO → BOM → COK	
3	1	12/05/2019	1	1	CCU → NAG → BLR	
4	1	01/03/2019	2	3	BLR → NAG → DEL	
…	…	…	…	…	…	
10678	6	9/04/2019	1	1	CCU → BLR	
10679	2	27/04/2019	1	1	CCU → BLR	
10680	0	27/04/2019	2	2	BLR → DEL	
10681	5	01/03/2019	2	3	BLR → DEL	
10682	2	9/05/2019	0	0	DEL → GOI → BOM → COK	

```
      Dep_Time  Arrival_Time Duration  Total_Stops Additional_Info  Price
0        22:20  01:10 22 Mar   2h 50m            0         No info   3897
1        05:50         13:15   7h 25m            2         No info   7662
2        09:25  04:25 10 Jun       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
…          …            …       …             …              …     …
10678    19:55         22:25   2h 30m            0         No info   4107
10679    20:45         23:20   2h 35m            0         No info   4145
10680    08:20         11:20       3h            0         No info   7229
10681    11:30         14:10   2h 40m            0         No info  12648
10682    10:55         19:15   8h 20m            2         No info  11753

[10682 rows x 11 columns]
```

Data Visualization

[35]:
```
#EDA
fdf=traindf[['Airline','Source','Destination','Total_Stops','Price']]
sns.heatmap(fdf.corr(),annot=True)
```

[35]: <Axes: >

Feature Scaling : To Split the data into training data and test data

[37]:
```
x=fdf[['Airline','Source','Destination','Total_Stops']]
y=fdf['Price']
```

[38]:
```
#Linear Regression
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪3,random_state=100)
```

Linear Regression

[39]:
```
from sklearn.linear_model import LinearRegression
regr=LinearRegression()
regr.fit(X_train,y_train)
print(regr.intercept_)
coeff_df=pd.DataFrame(regr.coef_,x.columns,columns=['coefficient'])
coeff_df
```

```
7211.098088897492
```

[39]:

	coefficient
Airline	-418.483922
Source	-3275.073380
Destination	2505.480291
Total_Stops	3541.798053

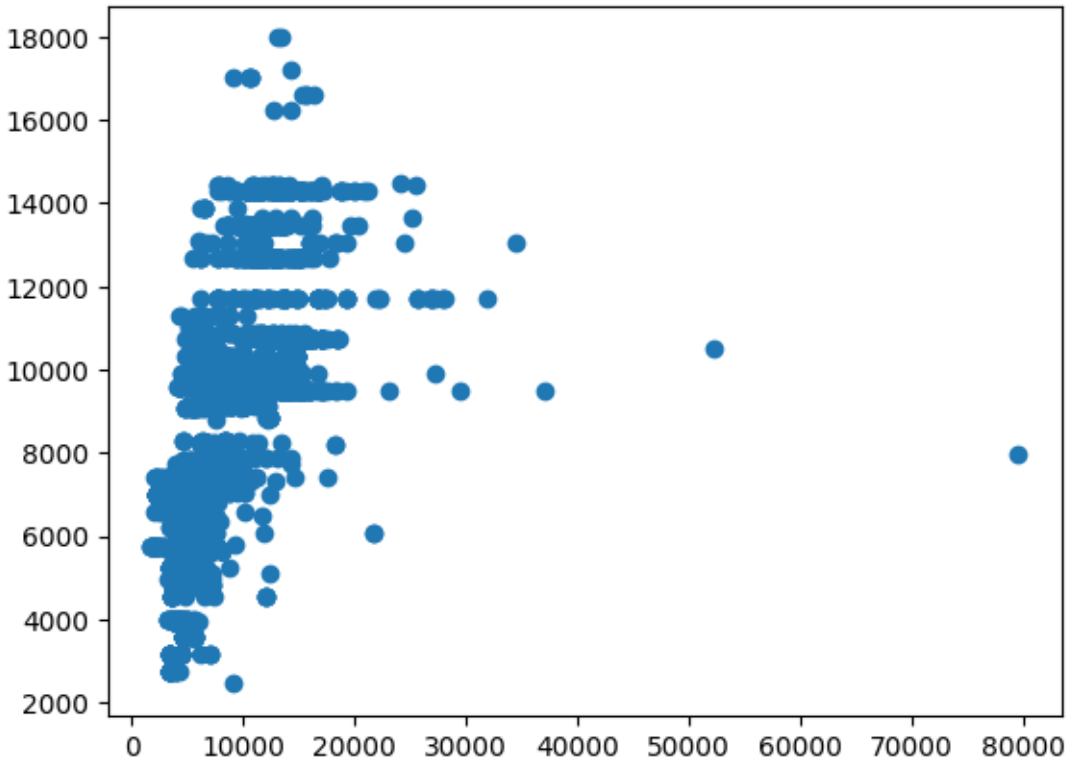
[40]:
```
#Linear Rgeression
score=regr.score(X_test,y_test)
print(score)
```

```
0.4108304890928348
```

[41]:
```
predictions=regr.predict(X_test)
```

[42]:
```
plt.scatter(y_test,predictions)
```

[42]:
```
<matplotlib.collections.PathCollection at 0x7efbaea0b7f0>
```



[43]:
```
x=np.array(fdf['Price']).reshape(-1,1)
y=np.array(fdf['Total_Stops']).reshape(-1,1)
fdf.dropna(inplace=True)
```

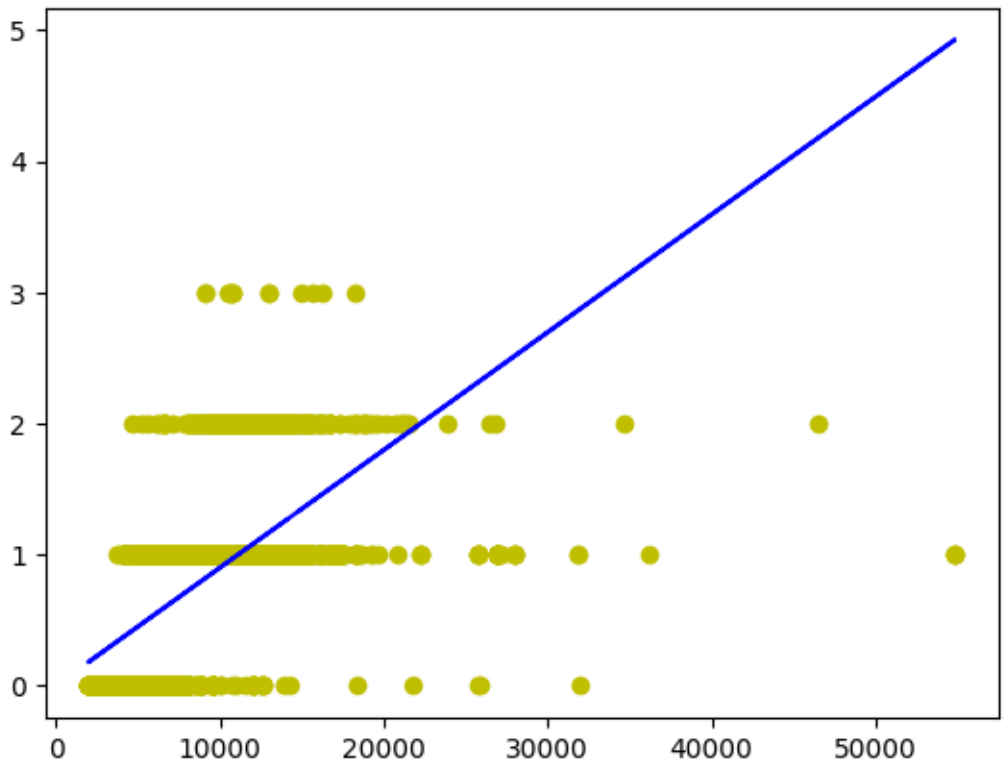
```
<ipython-input-43-ffcf73feb28b>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  fdf.dropna(inplace=True)
```

[44]:
```
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
regr.fit(X_train,y_train)
regr.fit(X_train,y_train)
```

[44]:
```
LinearRegression()
```

[45]:
```
y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='y')
plt.plot(X_test,y_pred,color='b')
plt.show()
```



#-> **Here We Didn't Get The Accuracy For LinearRegression**

#–>So We Are Going To Implement LogisticRegression

#***Logistic Regression***

[46]:
```
#Logistic Regression
x=np.array(fdf['Price']).reshape(-1,1)
y=np.array(fdf['Total_Stops']).reshape(-1,1)
fdf.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(max_iter=10000)
```

```
<ipython-input-46-67616b2294ca>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  fdf.dropna(inplace=True)
```

[47]:
```
lr.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
```

```
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[47]:
```
LogisticRegression(max_iter=10000)
```

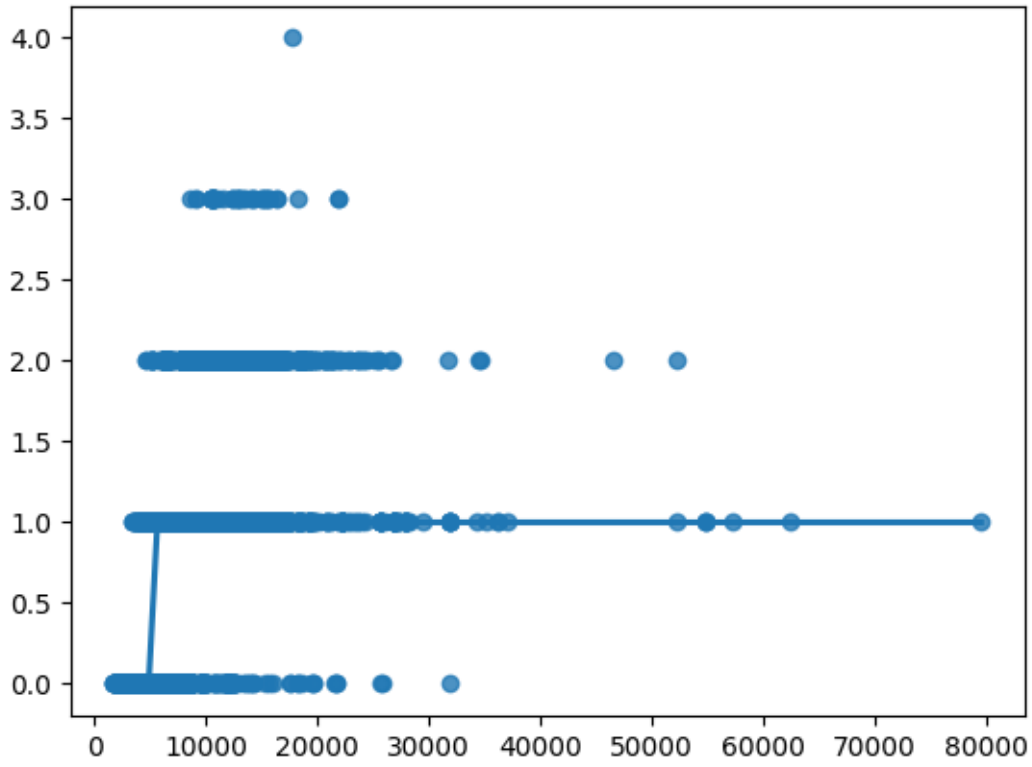
[48]:
```
score=lr.score(x_test,y_test)
print(score)
```

```
0.7160686427457098
```

[49]:
```
sns.regplot(x=x,y=y,data=fdf,logistic=True,ci=None)
```

```
/usr/local/lib/python3.10/dist-
packages/statsmodels/genmod/families/links.py:187: RuntimeWarning: overflow
encountered in exp
  t = np.exp(-z)
```

[49]:
```
<Axes: >
```



#-> Since we did not get the accuracy for LogisticRegression

So we are going to implement Decision Tree and Random Forest to make a comparative study for finding the best model for the dataset

#Decision Tree

[50]:
```
#Decision tree
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

[50]:
```
DecisionTreeClassifier(random_state=0)
```

[51]:
```
score=clf.score(x_test,y_test)
print(score)
```

```
0.9369734789391576
```

#Random Forest

[52]:
```
#Random forest classifier
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

```
<ipython-input-52-1f357c8f5bd1>:4: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  rfc.fit(X_train,y_train)
```

[52]:
```
RandomForestClassifier()
```

[53]:
```
params={'max_depth':[2,3,5,10,20],
 'min_samples_leaf':[5,10,20,50,100,200],
 'n_estimators':[10,25,30,50,100,200]}
```

[54]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
```

[55]:
```
grid_search.fit(X_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_split.py:700:
UserWarning: The least populated class in y has only 1 members, which is less
than n_splits=2.
  warnings.warn(
/usr/local/lib/python3.10/dist-
packages/sklearn/model_selection/_validation.py:686: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
```

```
packages/sklearn/model_selection/_validation.py:686: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
/usr/local/lib/python3.10/dist-
packages/sklearn/model_selection/_validation.py:686: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
/usr/local/lib/python3.10/dist-
packages/sklearn/model_selection/_validation.py:686: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
/usr/local/lib/python3.10/dist-
packages/sklearn/model_selection/_validation.py:686: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:909:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using
ravel().
  self.best_estimator_.fit(X, y, **fit_params)
```

[55]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [2, 3, 5, 10, 20],
                         'min_samples_leaf': [5, 10, 20, 50, 100, 200],
                         'n_estimators': [10, 25, 30, 50, 100, 200]},
             scoring='accuracy')
```

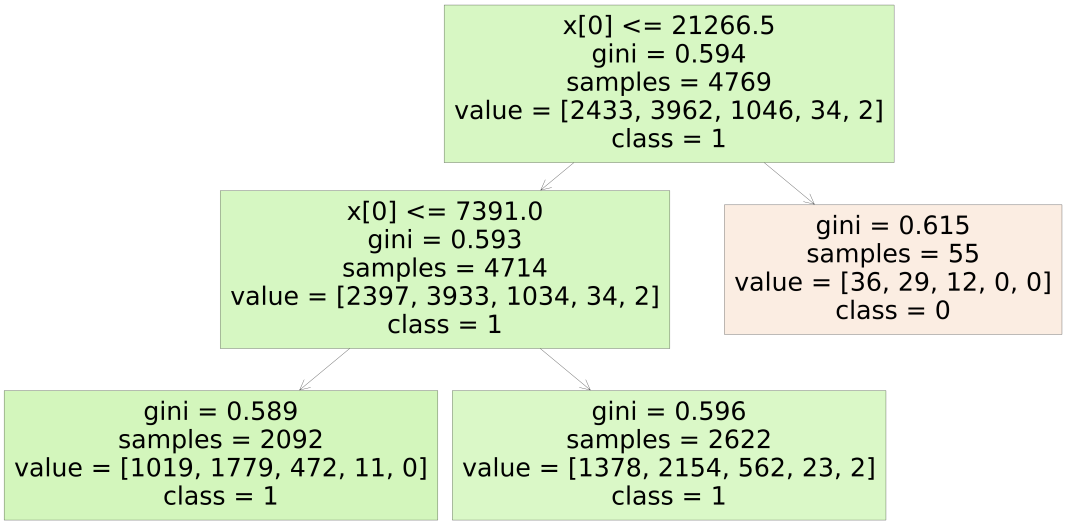
[56]:
```
grid_search.best_score_
```

[56]:
```
0.523605715699528
```

[57]:
```
rf_best=grid_search.best_estimator_
rf_best
```

[57]:
```
RandomForestClassifier(max_depth=2, min_samples_leaf=50, n_estimators=10)
```

[58]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],class_names=['0','1','2','3','4'],filled=True);
```



[59]:
```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.4595943837753511
```

-> Here when we compare between Decision Tree and Random Forest, we can confirm that Decision Tree has more accuracy than Random Forest which makes it the best model for this dataset. It makes Decision Tree to perform better than Random Forest. But it may vary for the other datasets where in most cases Random Forest performs better as it has reduced overfitting and robust to outliers.

#CONCLUSION **:- Based on accuracy scores of all the models which were implemented among those we can conclude that "Decision Tree" is the best model for the given dataset.**