Problem statement:To predict how best the data fits,and to predict the accuracy based on the given features of Flight Price pridection

1)Data collection

In [1]:
```
#Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
#Reading the Train_data
train_df=pd.read_csv(r"C:\Users\Jayadeep\Downloads\Copy of Data_Train.CSV")
train_df
```

Out[2]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [3]:
```
#Reading Test_data
test_df=pd.read_csv(r"C:\Users\Jayadeep\Downloads\Copy of Test_set.CSV")
test_df
```

Out[3]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info
...	...	...	...	...	...	...	...	...	...	...
2666	Air India	6/06/2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	20:25 07 Jun	23h 55m	1 stop	No info
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU ? BLR	14:20	16:55	2h 35m	non-stop	No info
2668	Jet Airways	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	04:25 07 Mar	6h 35m	1 stop	No info
2669	Air India	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	04:00	19:15	15h 15m	1 stop	No info
2670	Multiple carriers	15/06/2019	Delhi	Cochin	DEL ? BOM ? COK	04:55	19:15	14h 20m	1 stop	No info

2671 rows × 10 columns

2)Data cleaning and Preprocessing

```
#Exploratory data anlysis
```

In [4]:
```
train_df.head()
```

Out[4]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302

In [5]:
```
test_df.head()
```

Out[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info

In [6]:
```
train_df.shape
```

Out[6]:
```
(10683, 11)
```

In [7]:
```
test_df.shape
```

Out[7]:
```
(2671, 10)
```

In [8]:
```
train_df.describe
```

Out[8]:
```
<bound method NDFrame.describe of           Airline Date_of_Journey    Source Destination  \
0           IndiGo      24/03/2019  Banglore   New Delhi
1         Air India       1/05/2019   Kolkata    Banglore
2       Jet Airways       9/06/2019     Delhi      Cochin
3           IndiGo      12/05/2019   Kolkata    Banglore
4           IndiGo      01/03/2019  Banglore   New Delhi
...            ...             ...       ...         ...
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  \
0                 BLR ? DEL    22:20  01:10 22 Mar   2h 50m    non-stop
1      CCU ? IXR ? BBI ? BLR    05:50         13:15   7h 25m     2 stops
2      DEL ? LKO ? BOM ? COK    09:25  04:25 10 Jun      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
...                      ...      ...           ...      ...         ...
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
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]>
```

In [9]:
```
test_df.describe
```

Out[9]:
```
<bound method NDFrame.describe of                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]>
```

In [10]:
```
train_df.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
```

In [11]:
```
test_df.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
```

In [12]:
```
train_df.isnull().sum()
```

Out[12]:
```
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
```

In [13]:
```
train_df.fillna(method='ffill',inplace=True)
```

In [14]:
```
train_df.isnull().sum()
```

Out[14]:
```
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
```

In [15]:
```
test_df.isnull().sum()
```

Out[15]:
```
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
```

In [16]:
```
train_df['Airline'].value_counts()
```

Out[16]:
```
Jet Airways                          3849
IndiGo                               2053
Air India                            1752
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
```

In [17]:
```
train_df['Source'].value_counts()
```

Out[17]:
```
Delhi       4537
Kolkata     2871
Banglore    2197
Mumbai       697
Chennai      381
Name: Source, dtype: int64
```

In [18]:
```
train_df['Additional_Info'].value_counts()
```

Out[18]:
```
No info                         8345
In-flight meal not included     1982
No check-in baggage included     320
1 Long layover                    19
Change airports                    7
Business class                     4
No Info                            3
1 Short layover                    1
Red-eye flight                     1
2 Long layover                     1
Name: Additional_Info, dtype: int64
```

In [19]:
```
train_df['Destination'].value_counts()
```

Out[19]:
```
Cochin       4537
Banglore     2871
Delhi        1265
New Delhi     932
Hyderabad     697
Kolkata       381
Name: Destination, dtype: int64
```

In [20]:
```
train_df['Total_Stops'].value_counts()
```

Out[20]:
```
1 stop      5625
non-stop    3492
2 stops     1520
3 stops       45
4 stops        1
Name: Total_Stops, dtype: int64
```

In [21]:
```
a={"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}}
train_df=train_df.replace(a)
train_df
```

Out[21]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	2	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	1	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	1	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	2	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	0	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	5	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	2	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [22]:
```
b={"Source":{"Delhi":1,"Kolkata":2,"Banglore":3,"Mumbai":4,"Chennai":5}}
train_df=train_df.replace(b)
train_df
```

Out[22]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	3	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	2	1/05/2019	2	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	9/06/2019	1	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	1	12/05/2019	2	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	1	01/03/2019	3	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	2	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	2	27/04/2019	2	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	0	27/04/2019	3	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	5	01/03/2019	3	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	2	9/05/2019	1	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [23]:
```
c={"Destination":{"Cochin":1,"Banglore":3,"Delhi":4,"New Delhi":5,"Hyderabad":6,"Kolkata":7}}
train_df=train_df.replace(c)
train_df
```

Out[23]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	3	5	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	2	1/05/2019	2	3	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	9/06/2019	1	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	1	12/05/2019	2	3	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	1	01/03/2019	3	5	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	2	3	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	2	27/04/2019	2	3	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	0	27/04/2019	3	4	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	5	01/03/2019	3	5	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	2	9/05/2019	1	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [24]:
```
d={"Total_Stops":{"non-stop":0,"1 stop":1,"2 stops":2,"3 stops":3,"4 stops":4}}
train_df=train_df.replace(d)
train_df
```

Out[24]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	3	5	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	0	No info	3897
1	2	1/05/2019	2	3	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2	No info	7662
2	0	9/06/2019	1	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2	No info	13882
3	1	12/05/2019	2	3	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1	No info	6218
4	1	01/03/2019	3	5	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	2	3	CCU ? BLR	19:55	22:25	2h 30m	0	No info	4107
10679	2	27/04/2019	2	3	CCU ? BLR	20:45	23:20	2h 35m	0	No info	4145
10680	0	27/04/2019	3	4	BLR ? DEL	08:20	11:20	3h	0	No info	7229
10681	5	01/03/2019	3	5	BLR ? DEL	11:30	14:10	2h 40m	0	No info	12648
10682	2	9/05/2019	1	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2	No info	11753

10683 rows × 11 columns

In [25]:
```
train_df
```

Out[25]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	3	5	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	0	No info	3897
1	2	1/05/2019	2	3	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2	No info	7662
2	0	9/06/2019	1	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2	No info	13882
3	1	12/05/2019	2	3	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1	No info	6218
4	1	01/03/2019	3	5	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	2	3	CCU ? BLR	19:55	22:25	2h 30m	0	No info	4107
10679	2	27/04/2019	2	3	CCU ? BLR	20:45	23:20	2h 35m	0	No info	4145
10680	0	27/04/2019	3	4	BLR ? DEL	08:20	11:20	3h	0	No info	7229
10681	5	01/03/2019	3	5	BLR ? DEL	11:30	14:10	2h 40m	0	No info	12648
10682	2	9/05/2019	1	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2	No info	11753

10683 rows × 11 columns

3)Data Visualization

In [26]:
```
df=train_df[['Airline','Source','Destination','Total_Stops','Price']]
sns.heatmap(df.corr(),annot=True)
```

Out[26]: <AxesSubplot:>

In [27]:
```
x=df[['Airline','Source','Destination','Total_Stops']]
y=df['Price']
```

4)Building the Model

In [28]:
```
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=0)
```

In [29]:
```
from sklearn.linear_model import LinearRegression
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.intercept_)
coeff_df=pd.DataFrame(regr.coef_,columns=['coefficient'])
coeff_df
```

```
8298.52853489179
```

Out[29]:

	coefficient
0	-349.864027
1	-4177.926793
2	2472.364335
3	3589.079598

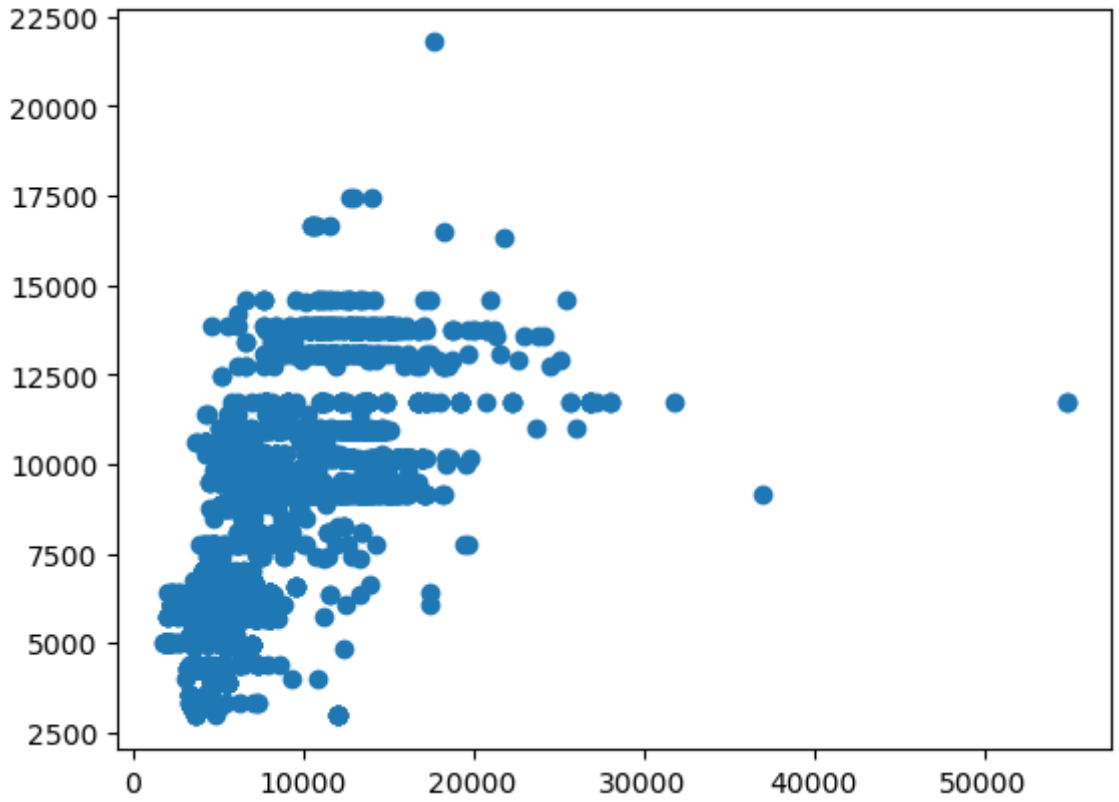
5)Evaluation of model

In [31]:
```
score=regr.score(x_test,y_test)
print(score)
```

```
0.43702395281202056
```

In [34]:
```
predictions=regr.predict(x_test)
plt.scatter(y_test,predictions)
```

Out[34]:
```
<matplotlib.collections.PathCollection at 0x1f8649ff790>
```



In [35]:
```
x=np.array(df['Price']).reshape(-1,1)
y=np.array(df['Total_Stops']).reshape(-1,1)
df.dropna(inplace=True)
```

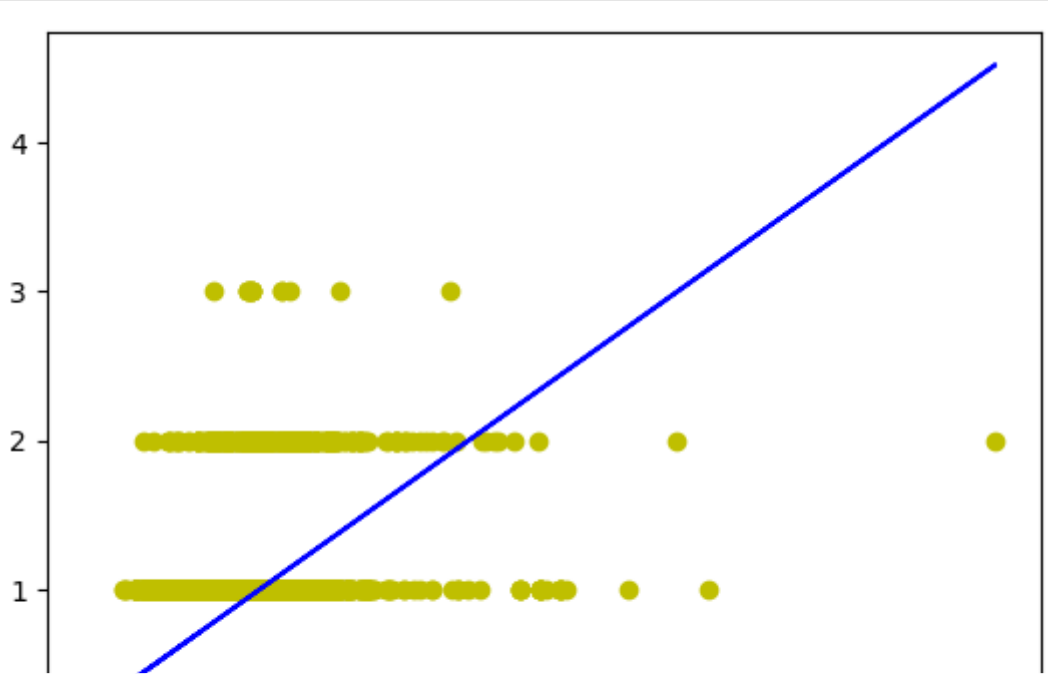
```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_3348\2340303250.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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

In [36]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
regr.fit(x_train,y_train)
regr.fit(x_train,y_train)
```

Out[36]:
```
LinearRegression()
```

In [38]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='y')
plt.plot(x_test,y_pred,color='b')
plt.show()
```



The accuracy of the Linear Regression is 0.43702

Logistic Regression

In [39]:
```
#Importing Libraries
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [40]:
```
#Logistic Regression
x=np.array(df['Price']).reshape(-1,1)
y=np.array(df['Total_Stops']).reshape(-1,1)
```

In [41]:
```
df.dropna(inplace=True)
```

```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_3348\1379821321.py:1: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df.dropna(inplace=True)
```

In [42]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [43]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(max_iter=10000)
```

In [44]:
```
lr.fit(x_train,y_train)
```

```
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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)
```

Out[44]:
```
LogisticRegression(max_iter=10000)
```

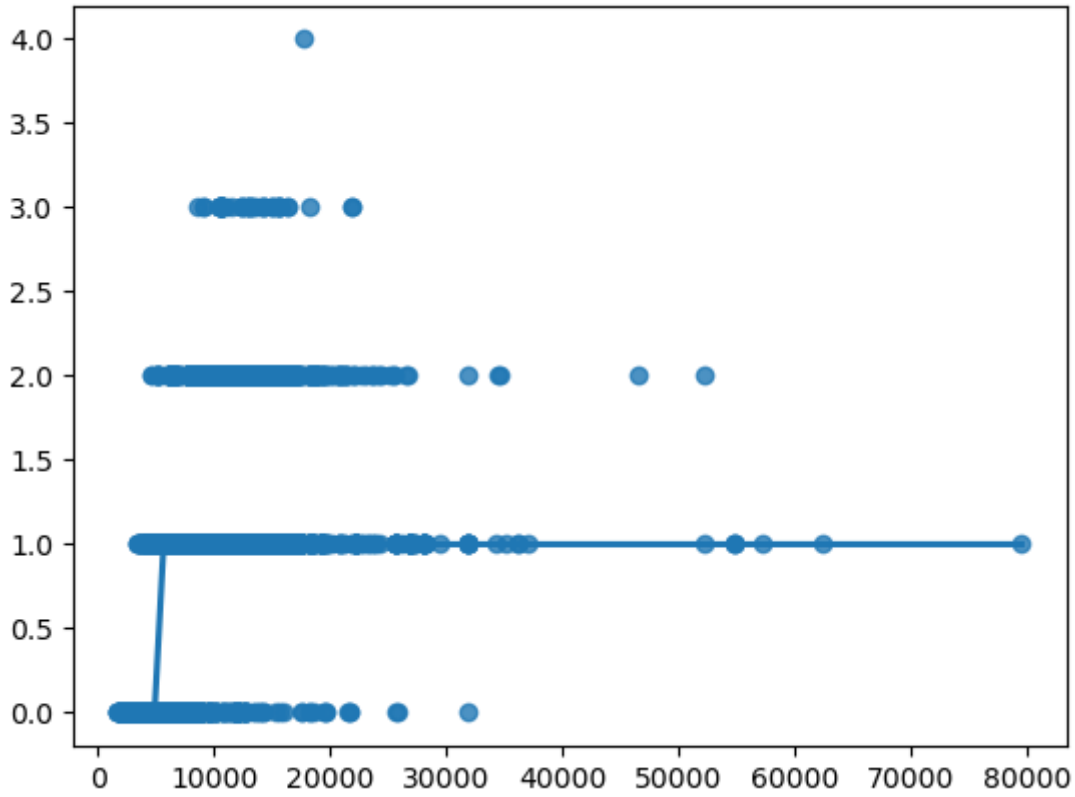
In [45]:
```
score=lr.score(x_test,y_test)
print(score)
```

```
0.7101404056162246
```

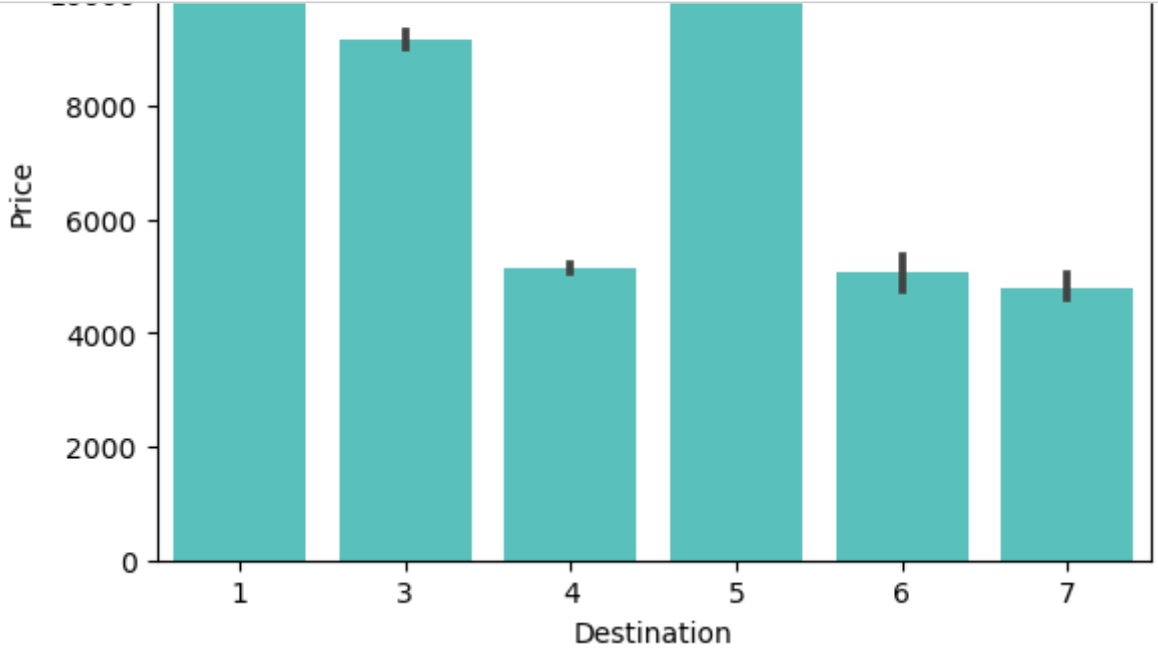
In [46]:
```
sns.regplot(x=x,y=y,data=df,logistic=True,ci=None)
```

```
C:\Users\Jayadeep\anaconda3\lib\site-packages\statsmodels\genmod\families\links.py:187: RuntimeWarning: overflow enc
ountered in exp
  t = np.exp(-z)
```

Out[46]: <AxesSubplot:>



In [48]:
```
sns.barplot(x='Destination', y='Price', data=df, color="mediumturquoise")
plt.show()
```



The accuracy of the Logistic Regression is 0.71014

Decision Tree

In [49]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [50]:
```
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

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

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

```
0.9351014040561623
```

The accuracy of the Decision Tree is 0.9351

Random Classifier

In [52]:
```
import pandas as pd
import numpy as  np
import matplotlib.pyplot as plt ,seaborn as sns
```

In [55]:
```
#Random forest classifier
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_3348\2206399760.py:4: DataConversionWarning: A column-vector y was pa
ssed 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)
```

Out[55]:
```
RandomForestClassifier()
```

In [56]:
```
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]}
```

In [57]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
```

In [60]:
```
grid_search.fit(x_train,y_train)
```

```
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:680: 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)
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:680: 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)
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:680: 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)
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:680: 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)
C:\Users\Jayadeep\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:680: 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().
```

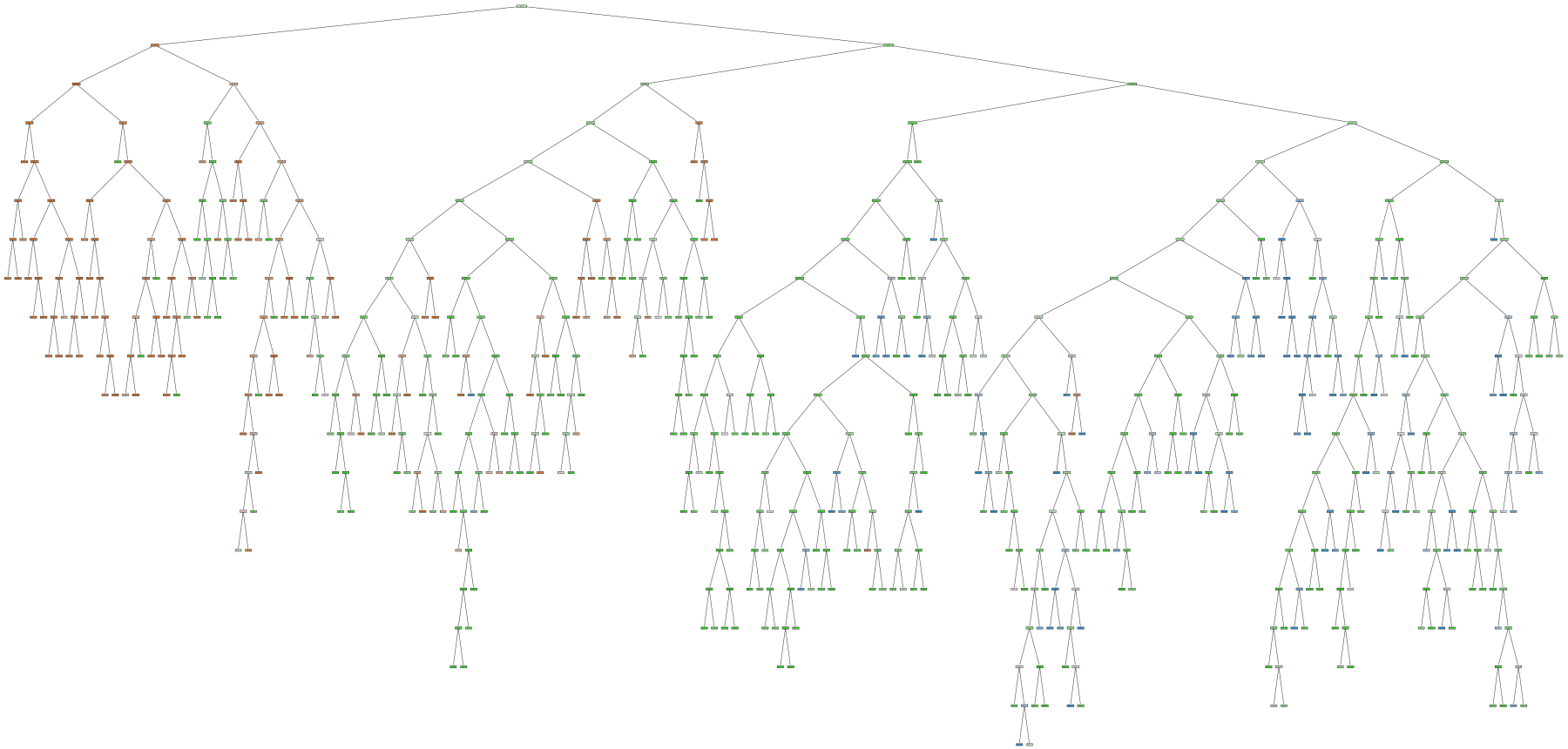
In [61]:
```
grid_search.best_score_
```

Out[61]:
```
0.868815191227601
```

In [62]:
```
rf_best=grid_search.best_estimator_
rf_best
```

Out[62]:
```
RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=30)
```

In [63]:
```
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);
```



In [64]:
```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.9344773790951638
```

The accuracy of the Random Forest is 0.9344

conclusion:

The given dataset is "Flight Price Pridiction",here we need to find the bestfit Model. As per the data set ,we have used different types of models,that different models got different types of accuracyies.In this process, in Decision tree and Random forest I got same accuracy of 0.9351 and 0.09344 , but with point difference Decision tree has highest accyracy.I have done so many visuvalization graps as per the given Features,to predict the output.For the highest accuracy,I have done so many models among those DecisionTree got highest accuracy.

Therefore Decision Tree Model is the Bestfit for this DataSet.