PROJECT ON FLIGHT PREDICTION DATASET

1.PROBLEM STATEMENT:- TO PREDICT THEINSURANCE CHARGES BASED ON VARIOUS FEATURES OF THE DATASET

In [ ]:

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn import metrics
from sklearn.linear_model import Lasso,LassoCV
from sklearn.linear_model import Ridge,RidgeCV
from sklearn.preprocessing import StandardScaler
```

2. Data Collection

Read the Data

In [3]:
```
train_df=pd.read_csv(r"C:\Users\HP\Documents\Data_Train.csv")
train_df
```

Out[3]:

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

10683 rows × 11 columns

In [4]:
```
test_df=pd.read_csv(r"C:\Users\HP\Documents\Test_set.csv")
test_df
```

Out[4]:

	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

3.Data Cleaning and preprocessing

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

Out[5]:

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

In [6]:
```
test_df.head()
```

Out[6]:

	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 [7]:
```
train_df.tail()
```

Out[7]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Inf
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU → BLR	19:55	22:25	2h 30m	non-stop	No inf
10679	Air India	27/04/2019	Kolkata	Banglore	CCU → BLR	20:45	23:20	2h 35m	non-stop	No inf
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR → DEL	08:20	11:20	3h	non-stop	No inf
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR → DEL	11:30	14:10	2h 40m	non-stop	No inf
10682	Air India	9/05/2019	Delhi	Cochin	DEL → GOI → BOM → COK	10:55	19:15	8h 20m	2 stops	No inf

In [8]:
```
test_df.tail()
```

Out[8]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_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

In [9]:
```
train_df.describe()
```

Out[9]:

	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

In [10]:
```
test_df.describe()
```

Out[10]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Inf
count	2671	2671	2671	2671	2671	2671	2671	2671	2671	267
unique	11	44	5	6	100	199	704	320	5	
top	Jet Airways	9/05/2019	Delhi	Cochin	DEL → BOM → COK	10:00	19:00	2h 50m	1 stop	No inf
freq	897	144	1145	1145	624	62	113	122	1431	214

In [11]:
```
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 [12]:
```
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 [13]:
```
train_df.columns
```

Out[13]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

In [14]:
```
test_df.columns
```

Out[14]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info'],
      dtype='object')
```

In [15]:
```
train_df.shape
```

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

In [16]:
```
test_df.shape
```

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

In [17]:
```
train_df.duplicated().sum()
```

Out[17]:
```
220
```

In [18]:
```
test_df.duplicated().sum()
```

Out[18]:
```
26
```

In [19]:
```
train_df.isnull().any()
```

Out[19]:
```
Airline           False
Date_of_Journey   False
Source            False
Destination       False
Route              True
Dep_Time          False
Arrival_Time      False
Duration          False
Total_Stops        True
Additional_Info   False
Price             False
dtype: bool
```

In [20]:
```
train_df.dropna(inplace=True)
```

In [21]:
```
test_df.isnull().any()
```

Out[21]:
```
Airline            False
Date_of_Journey    False
Source             False
Destination        False
Route              False
Dep_Time           False
Arrival_Time       False
Duration           False
Total_Stops        False
Additional_Info    False
dtype: bool
```

In [22]:
```
train_df['Source'].value_counts()
```

Out[22]:
```
Source
Delhi       4536
Kolkata     2871
Banglore    2197
Mumbai       697
Chennai      381
Name: count, dtype: int64
```

In [23]:
```
test_df['Source'].value_counts()
```

Out[23]:
```
Source
Delhi       1145
Kolkata      710
Banglore     555
Mumbai       186
Chennai       75
Name: count, dtype: int64
```

In [24]:
```
train_df['Airline'].value_counts()
```

Out[24]:
```
Airline
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: count, dtype: int64
```

In [25]:
```
test_df['Airline'].value_counts()
```

Out[25]:
```
Airline
Jet Airways                          897
IndiGo                               511
Air India                            440
Multiple carriers                    347
SpiceJet                             208
Vistara                              129
Air Asia                              86
GoAir                                 46
Multiple carriers Premium economy      3
Vistara Premium economy                2
Jet Airways Business                   2
Name: count, dtype: int64
```

In [26]:
```
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}}
train_df=train_df.replace(airline)
train_df
```

Out[26]:

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

10682 rows × 11 columns

In [27]:
```
city={"Source":{"Delhi":0,"Kolkata":1,"Banglore":2,
"Mumbai":3,"Chennai":4}}
train_df=train_df.replace(city)
train_df
```

Out[27]:

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

10682 rows × 11 columns

In [28]:
```
destination={"Destination":{"Cochin":0,"Banglore":1,"Delhi":2,
"New Delhi":3,"Hyderabad":4,"Kolkata":5}}
train_df=train_df.replace(destination)
train_df
```

Out[28]:

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

10682 rows × 11 columns

In [29]:
```
stops={"Total_Stops":{"non-stop":0,"1 stop":1,"2 stops":2,"3 stops":3,"4 stops":4}}
train_df=train_df.replace(stops)
train_df
```

Out[29]:

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

10682 rows × 11 columns

In [30]:
```
train_df
```

Out[30]:

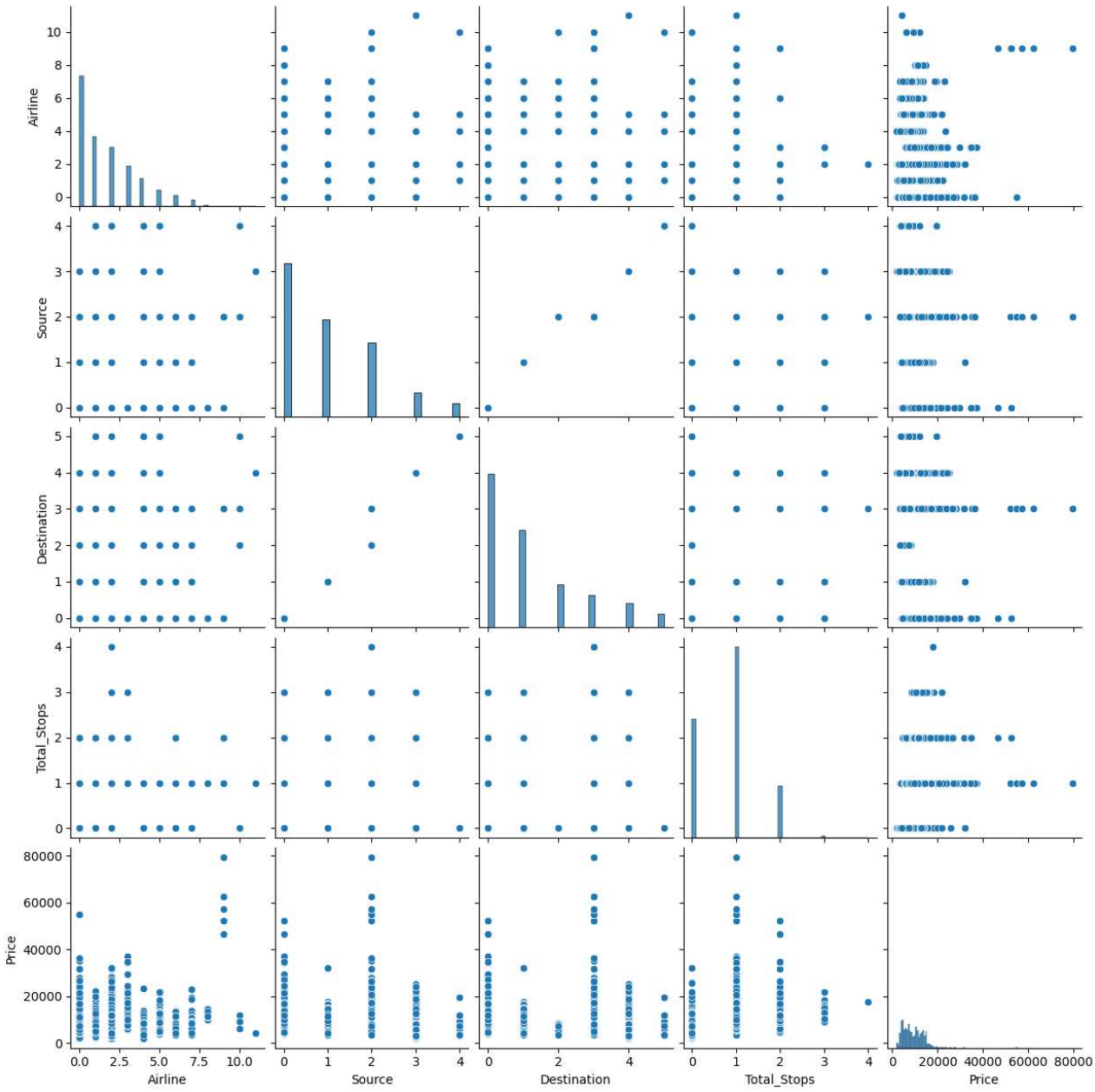
	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	1	24/03/2019	2	3	BLR → DEL	22:20	01:10 22 Mar	2h 50m	0	No info
1	2	1/05/2019	1	1	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2	No info
2	0	9/06/2019	0	0	DEL → LKO → BOM → COK	09:25	04:25 10 Jun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

10682 rows × 11 columns

DATA VISUALIZATION

In [31]:
```
sns.pairplot(train_df)
```

Out[31]: <seaborn.axisgrid.PairGrid at 0x2184afaf8b0>

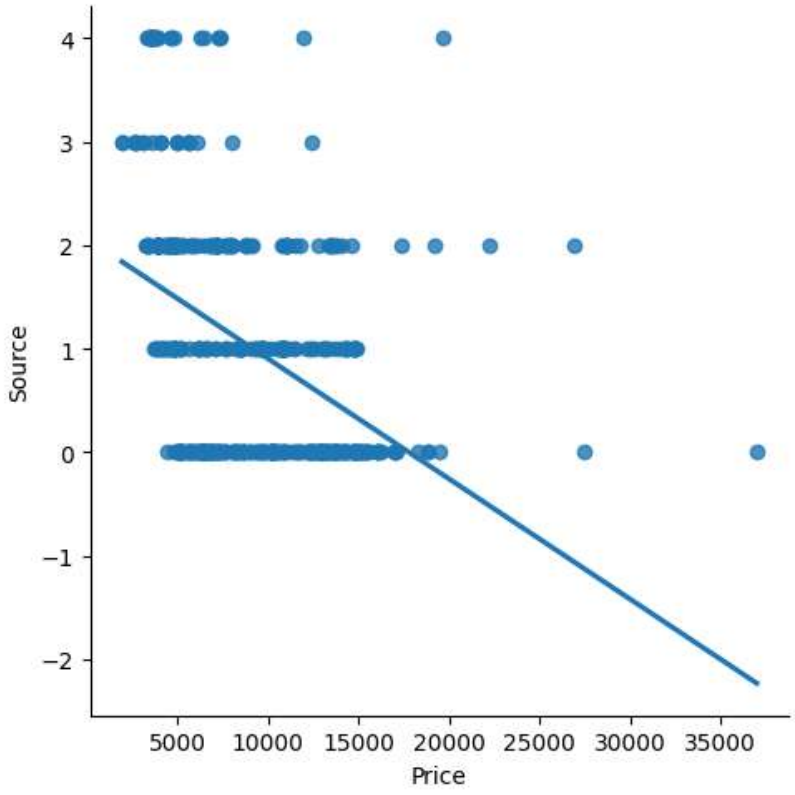


In [32]:
```
train_df=train_df[['Airline','Source','Destination','Total_Stops','Price']]
sns.heatmap(train_df.corr(),annot=True)
```

Out[32]: <Axes: >

In [33]:
```
train_df500=train_df[:][:500]
sns.lmplot(x="Price",y="Source",data=train_df500,order=1,ci=None)
```

Out[33]: <seaborn.axisgrid.FacetGrid at 0x21850f53a60>



LINEAR REGRESSION

In [34]:
```
X=train_df[['Airline','Source','Destination','Total_Stops']]
y=train_df['Price']
```

In [35]:
```
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=30)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
regr=LinearRegression()
regr.fit(X_train,y_train)
```

Out[36]:
```
▾ LinearRegression

LinearRegression()
```

In [37]:
```
score=regr.score(X_test,y_test)
print(score)
```

```
0.43788984175329504
```

In [38]:
```
predictions=regr.predict(X_test)
```

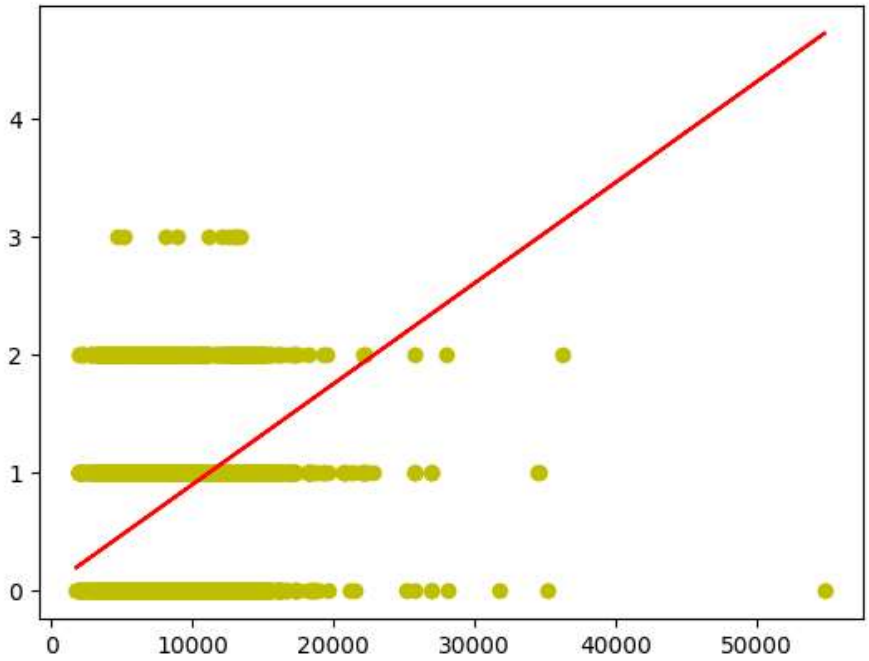
In [39]:
```
X=np.array(train_df['Price']).reshape(-1,1)
y=np.array(train_df['Total_Stops']).reshape(-1,1)
train_df.dropna(inplace=True)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_29452\586166043.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)
  train_df.dropna(inplace=True)
```

In [40]:
```
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)
```

Out[40]:
```
▾ LinearRegression

LinearRegression()
```

In [68]:
```
y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='y')
plt.plot(X_test,y_pred,color='r')
plt.show()
```



Logistic Regression

In [42]:
```
#Logistic Regression
x=np.array(train_df['Price']).reshape(-1,1)
y=np.array(train_df['Total_Stops']).reshape(-1,1)
train_df.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)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_29452\601506973.py: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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  train_df.dropna(inplace=True)
```

In [43]:
```
lr.fit(x_train,y_train)
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-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)
```

Out[43]:
```
LogisticRegression
LogisticRegression(max_iter=10000)
```

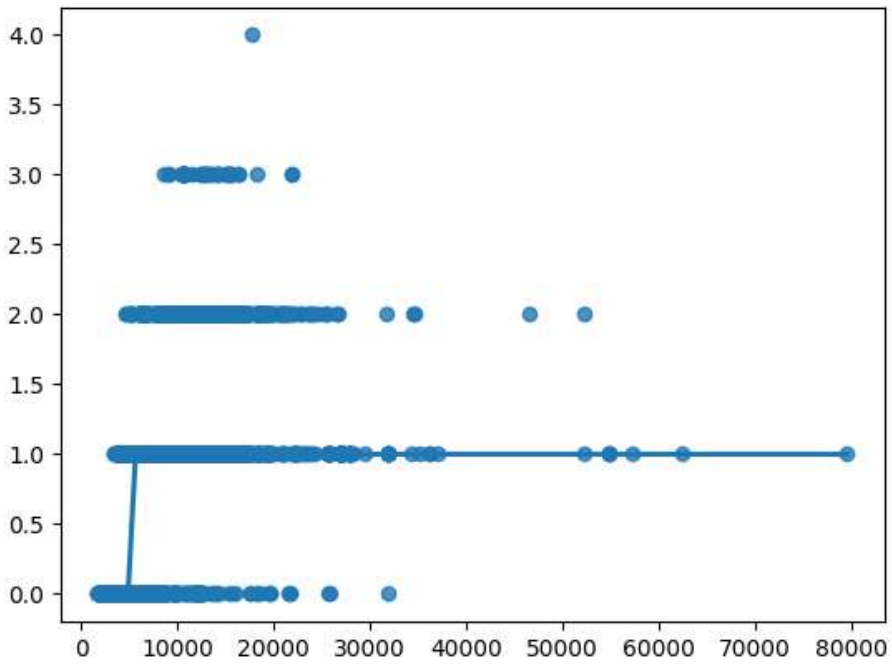
In [44]:
```
score=lr.score(x_test,y_test)
print(score)
```

```
0.7160686427457098
```

In [45]:
```
sns.regplot(x=x,y=y,data=train_df,logistic=True,ci=None)
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\genmod\families\l
inks.py:198: RuntimeWarning: overflow encountered in exp
  t = np.exp(-z)
```

Out[45]: <Axes: >



Decision Tree

In [46]:
```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

Out[46]:
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [47]:
```
score=clf.score(x_test,y_test)
print(score)
```

```
0.9369734789391576
```

Random Forest

In [48]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_29452\4104924521.py:3: DataConversionWarning: A column-vec
tor y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for ex
ample using ravel().
  rfc.fit(X_train,y_train)
```

Out[48]:
```
RandomForestClassifier
RandomForestClassifier()
```

In [49]:
```
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 [50]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
```

In [51]:
```
grid_search.fit(X_train,y_train)
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_s
plit.py:700: UserWarning: The least populated class in y has only 1 members, which is less than
n_splits=2.
  warnings.warn(
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
```

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

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_v
alidation.py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples,), for example using ravel().
```

In [64]:
```
grid_search.best_score_
```

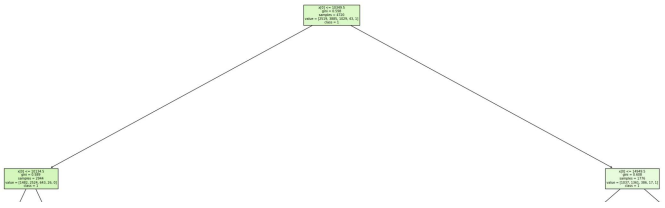
Out[64]:
```
0.5238731668896858
```

In [65]:
```
rf_best=grid_search.best_estimator_
rf_best
```

Out[65]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=5, min_samples_leaf=5, n_estimators=50)
```

In [67]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(50,40))
plot_tree(rf_best.estimators_[5],class_names=['0','1','2','3','4'],filled=True)
```

```
[Text(0.8409090909090909, 0.08333333333333333, 'gini = 0.63\nsamples = 42\nvalue = [23, 31, 11,
1, 0]\nclass = 1'),
 Text(0.8863636363636364, 0.08333333333333333, 'gini = 0.602\nsamples = 28\nvalue = [5, 16, 18,
0, 0]\nclass = 2'),
 Text(0.9545454545454546, 0.25, 'x[0] <= 22015.0\ngini = 0.61\nsamples = 120\nvalue = [74, 79, 2
5, 0, 0]\nclass = 1'),
 Text(0.9318181818181818, 0.08333333333333333, 'gini = 0.605\nsamples = 59\nvalue = [47, 33, 14,
0, 0]\nclass = 0'),
 Text(0.9772727272727273, 0.08333333333333333, 'gini = 0.58\nsamples = 61\nvalue = [27, 46, 11,
0, 0]\nclass = 1')]
```



Conclusion

In [ ]:
```
I have got accuaracy for different models are;

linear-43%
logistic-71%
decision tree-93%
random forest-52%
from the above it is concluded that decision tree model is best fit for given flight price predictio
```