Problem Statement

Finding the best fit for the given model

Data Collection

In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
```

In [2]:

```
trd=pd.read_csv(r"C:\Users\Sushma sree\Downloads\Data_Train.csv")
trd
```

Out[2]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duratior
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	..
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10683 rows × 11 columns

In [3]:

```
tst=pd.read_csv(r"C:\Users\Sushma sree\Downloads\Test_set.csv")
tst
```

Out[3]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m
...	...	...	...	...	...	...	...	...
2666	Air India	6/06/2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	20:25 07 Jun	23h 55m
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU ? BLR	14:20	16:55	2h 35m
2668	Jet Airways	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	04:25 07 Mar	6h 35m
2669	Air India	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	04:00	19:15	15h 15m
2670	Multiple carriers	15/06/2019	Delhi	Cochin	DEL ? BOM ? COK	04:55	19:15	14h 20m

2671 rows × 10 columns

Data Cleaning

In [4]:

```
trd.head()
```

Out[4]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	To
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	

In [5]:

```
trd.tail()
```

Out[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duratior
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

```
In [6]:  trd.describe
```

```
Out[6]: <bound method NDFrame.describe of            Airline Date_of_Journey    Sourc
e 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 [7]:  trd.shape
```

```
Out[7]: (10683, 11)
```

In [8]:

```
trd.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 [9]:

```
tst.head()
```

Out[9]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	To
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m	
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h	
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m	
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h	
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m	

In [10]:

```
tst.tail()
```

Out[10]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
2666	Air India	6/06/2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	20:25 07 Jun	23h 55m
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU ? BLR	14:20	16:55	2h 35m
2668	Jet Airways	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	04:25 07 Mar	6h 35m
2669	Air India	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	04:00	19:15	15h 15m
2670	Multiple carriers	15/06/2019	Delhi	Cochin	DEL ? BOM ? COK	04:55	19:15	14h 20m

In [11]:

```
tst.describe
```

Out[11]:

```
<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 [12]:

```
tst.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]:

```
tst.shape
```

Out[13]:

```
(2671, 10)
```

Data Preprocessing

In [14]:

```
trd.isna().any()
```

Out[14]:

```
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 [15]: trd.isnull().sum()
```

```
Out[15]: 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 [16]: trd.dropna(inplace=True)
```

```
In [17]: trd.isnull().sum()
```

```
Out[17]: 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 [18]: tst.isna().any()
```

```
Out[18]: 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 [19]:

```python
tst.isnull().sum()
```

Out[19]:

```
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 [20]:

```python
trd.duplicated().sum()
```

Out[20]:

```
220
```

In [21]:

```python
tst.duplicated().sum()
```

Out[21]:

```
26
```

In [22]:

```python
trd['Source'].value_counts()
```

Out[22]:

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

In [23]:

```python
trd['Airline'].value_counts()
```

Out[23]:

```
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 [24]:

```
trd['Destination'].value_counts()
```

Out[24]:

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

In [25]:

```
trd['Total_Stops'].value_counts()
```

Out[25]:

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

In [26]:

```python
t={'Total_Stops':{'1 stop':0,'non-stop':1,'2 stops':2,'3 stops':3,'4 stop
trd=trd.replace(t)
trd
```

Out[26]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duratior
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	..
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [27]:

```
s={'Source':{'Delhi':0,'Kolkata':1,'Banglore':2,'Mumbai':3,'Chennai':4}}
trd=trd.replace(s)
trd
```

Out[27]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	IndiGo	24/03/2019	2	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	Air India	1/05/2019	1	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	Jet Airways	9/06/2019	0	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	IndiGo	12/05/2019	1	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	IndiGo	01/03/2019	2	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	1	Banglore	CCU ? BLR	19:55	22:25	2h 30m
10679	Air India	27/04/2019	1	Banglore	CCU ? BLR	20:45	23:20	2h 35m
10680	Jet Airways	27/04/2019	2	Delhi	BLR ? DEL	08:20	11:20	3h
10681	Vistara	01/03/2019	2	New Delhi	BLR ? DEL	11:30	14:10	2h 40m
10682	Air India	9/05/2019	0	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [28]:

```
d={'Destination':{'Cochin':0,'Banglore':1,'Delhi':2,'New Delhi':3,'Hyderal
trd=trd.replace(d)
trd
```

Out[28]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	IndiGo	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	Air India	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	Jet Airways	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	IndiGo	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	IndiGo	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	Air India	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	Jet Airways	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	Vistara	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	Air India	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [29]:

```python
a={'Airline':{'Jet Airways':0,'IndiGo':1,'Air India':2,'Multiple carriers
              'Multiple carriers Premium economy':8,'Jet Airways Business
trd=trd.replace(a)
trd
```

Out[29]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	2	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	0	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	1	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	2	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	0	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	2	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [30]:

```python
tst['Source'].value_counts()
```

Out[30]:

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

In [31]:

```
tst['Airline'].value_counts()
```

Out[31]:

```
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 [32]:

```
tst['Destination'].value_counts()
```

Out[32]:

```
Destination
Cochin       1145
Banglore      710
Delhi         317
New Delhi     238
Hyderabad     186
Kolkata        75
Name: count, dtype: int64
```

In [33]:

```
tst['Total_Stops'].value_counts()
```

Out[33]:

```
Total_Stops
1 stop      1431
non-stop     849
2 stops      379
3 stops       11
4 stops        1
Name: count, dtype: int64
```

In [34]:

```
t1={'Total_Stops':{'1 stop':0,'non-stop':1,'2 stops':2,'3 stops':3,'4 sto
tst=trd.replace(t1)
tst
```

Out[34]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	2	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	0	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	1	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	2	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	0	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	2	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [35]:

```python
s1={'Source':{'Delhi':0,'Kolkata':1,'Banglore':2,'Mumbai':3,'Chennai':4}}
tst=tst.replace(s1)
tst
```

Out[35]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	2	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	0	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	1	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	2	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	0	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	2	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [36]:

```
d1={'Destination':{'Cochin':0,'Banglore':1,'Delhi':2,'New Delhi':3,'Hydera
tst=tst.replace(d1)
tst
```

Out[36]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	2	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	0	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	1	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	2	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	0	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	2	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [37]:

```
a1={'Airline':{'Jet Airways':0,'IndiGo':1,'Air India':2,'Multiple carrier
              'Multiple carriers Premium economy':8,'Jet Airways Business
tst=tst.replace(a1)
tst
```

Out[37]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m
1	2	1/05/2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m
2	0	9/06/2019	0	0	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h
3	1	12/05/2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m
...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	1	CCU ? BLR	19:55	22:25	2h 30m
10679	2	27/04/2019	1	1	CCU ? BLR	20:45	23:20	2h 35m
10680	0	27/04/2019	2	2	BLR ? DEL	08:20	11:20	3h
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m
10682	2	9/05/2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m

10682 rows × 11 columns

In [38]:

```
trd['Destination'].value_counts()
```

Out[38]:

```
Destination
0    4536
1    2871
2    1265
3     932
4     697
5     381
Name: count, dtype: int64
```

In [39]:

```
trd['Source'].value_counts()
```

Out[39]:

```
Source
0    4536
1    2871
2    2197
3     697
4     381
Name: count, dtype: int64
```

In [40]:

```
tst['Destination'].value_counts()
```

Out[40]:

```
Destination
0    4536
1    2871
2    1265
3     932
4     697
5     381
Name: count, dtype: int64
```

In [41]:

```
tst['Source'].value_counts()
```

Out[41]:

```
Source
0    4536
1    2871
2    2197
3     697
4     381
Name: count, dtype: int64
```

Data Visualisation

In [42]:

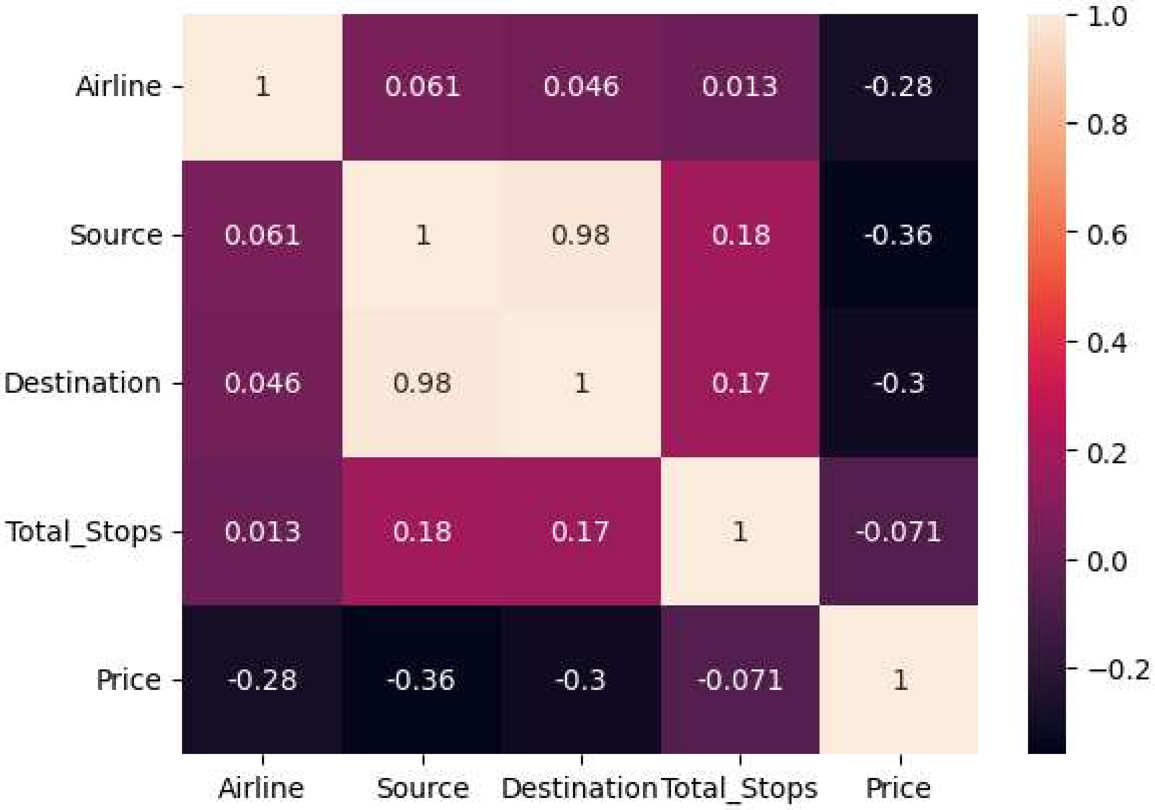
```
import seaborn as sns
```

In [43]:

```python
ed=trd[['Airline','Source','Destination','Total_Stops','Price']]
sns.heatmap(ed.corr(),annot=True)
```

Out[43]:

```
<Axes: >
```



In [44]:

```python
x=ed[['Airline','Source','Destination','Total_Stops']]
y=ed['Price']
```

Data Modeling

Linear Regression

In [45]:

```python
x=trd[['Destination']]
y=trd['Price']
```

In [46]:

```python
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [47]:

```python
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [48]:

```
lr.fit(x_train,y_train)
```

Out[48]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [49]:

```
lr.score(x_test,y_test)
```

Out[49]:

```
0.11551601253498878
```

In [50]:

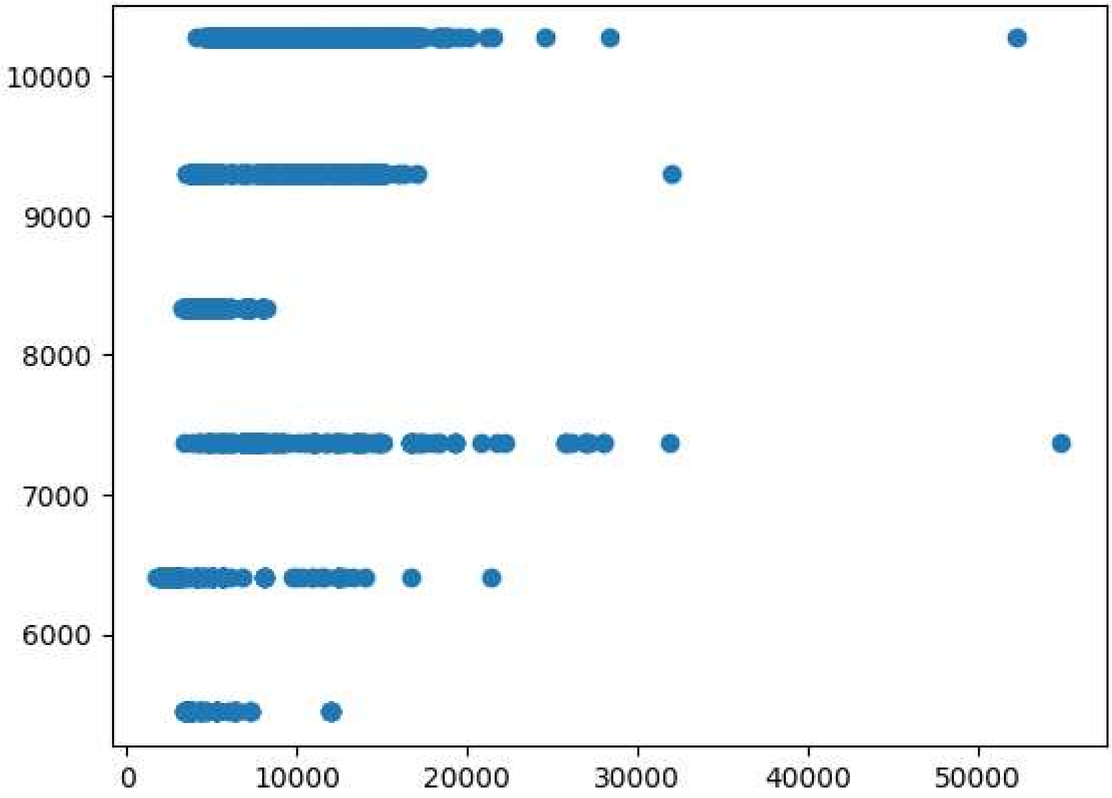
```
y_pred=lr.predict(x_test)
```

In [51]:

```
plt.scatter(y_test,y_pred)
```

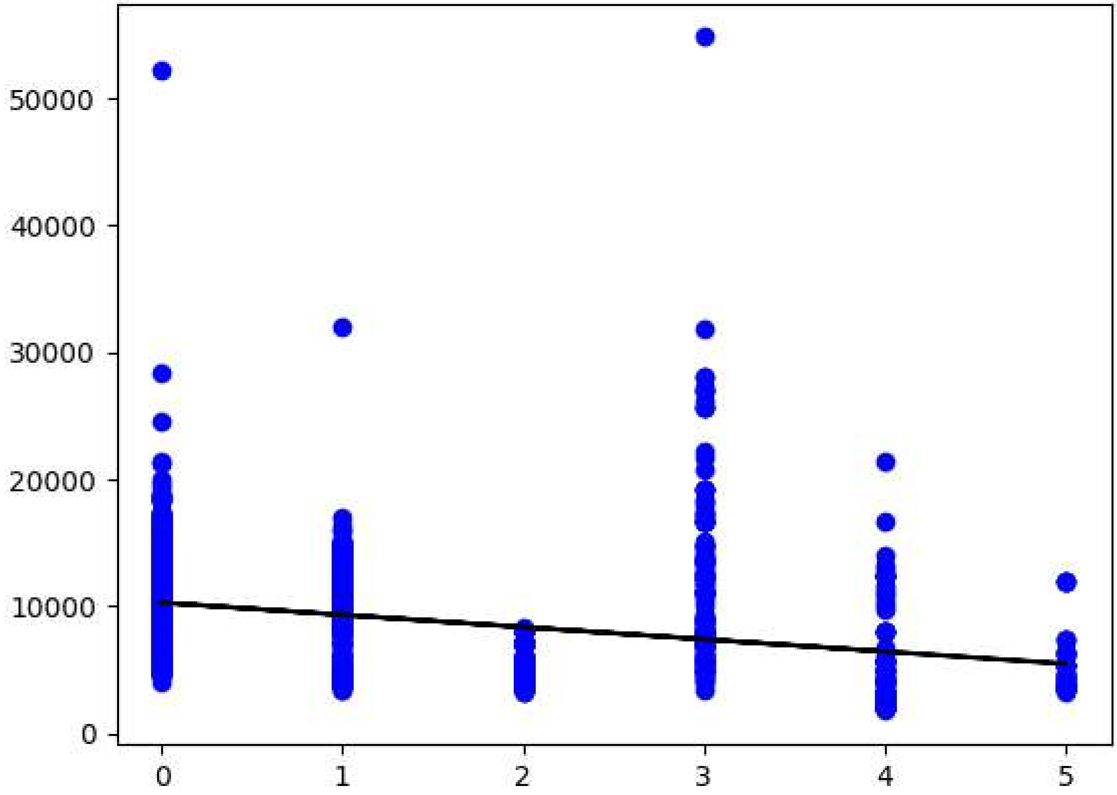
Out[51]:

```
<matplotlib.collections.PathCollection at 0x1fb27c31150>
```



In [52]:

```
y_pred=lr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



Logistic Regression

In [53]:

```
import numpy as np
```

In [54]:

```
x=trd[['Price']]
y=trd['Total_Stops']
```

In [55]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_s
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression(max_iter=10000)
```

In [56]:

```
lg.fit(x_train,y_train)
```

Out[56]:

```
LogisticRegression(max_iter=10000)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [57]:

```
lg.score(x_test,y_test)
```

Out[57]:

```
0.7160686427457098
```

Decision Tree

In [58]:

```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

Out[58]:

```
DecisionTreeClassifier(random_state=0)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [59]:

```
score=clf.score(x_test,y_test)
print(score)
```

```
0.9369734789391576
```

Randam Forest

In [60]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[60]:

```
RandomForestClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [61]:

```
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 [62]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="ac
```

In [63]:

```
grid_search.fit(x_train,y_train)
```

```
C:\Users\Sushma sree\AppData\Local\Programs\Python\Python310\lib\site-package
s\sklearn\model_selection\_split.py:700: UserWarning: The least populated cla
ss in y has only 1 members, which is less than n_splits=2.
  warnings.warn(
```

Out[63]:

```
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')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [64]:

```
grid_search.best_score_
```

Out[64]:

```
0.8736118904019652
```

In [65]:

```
rf_best=grid_search.best_estimator_
rf_best
```

Out[65]:

```
RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=30)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [66]:

```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.9369734789391576
```

Conclusion

By performing all the models to the given datasets we conclude that Decision Tree has the highest accuracy.

for our model Decision Tree is the best fit.

In [ ]:

```

```