

Problem statement:- finding the solution which model is suitable for Flight Price Prediction

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: traindf=pd.read_csv(r"C:\Users\sweet\Downloads\Data_train.csv")
traindf
```

Out[3]:

	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	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	Air India	01-05-2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	09-06-2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	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	09-04-2019	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	09-05-2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

```
In [4]: testdf=pd.read_csv(r"C:\Users\sweet\Downloads\test_data.csv")
testdf
```

Out[4]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	06-06-2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	07-06-2023 04:25	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	22-05-2023 19:00	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	25-06-2023 02:45	2h 50m	non-stop	No info
...
2666	Air India	06-06-2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	07-06-2023 20:25	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	06-03-2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	07-03-2023 04:25	6h 35m	1 stop	No info
2669	Air India	06-03-2019	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

In [5]: traindf

Out[5]:

	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	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	Air India	01-05-2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	09-06-2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	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	09-04-2019	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	09-05-2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [6]: testdf

Out[6]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	06-06-2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	07-06-2023 04:25	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	22-05-2023 19:00	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	25-06-2023 02:45	2h 50m	non-stop	No info
...
2666	Air India	06-06-2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	07-06-2023 20:25	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	06-03-2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	07-03-2023 04:25	6h 35m	1 stop	No info
2669	Air India	06-03-2019	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

In [7]: `traindf.head()`

Out[7]:

	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	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	Air India	01-05-2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	09-06-2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	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 [8]: `testdf.head()`

Out[8]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	06-06-2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	07-06-2023 04:25	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	22-05-2023 19:00	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	25-06-2023 02:45	2h 50m	non-stop	No info

In [9]: `traindf.tail()`

Out[9]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
10678	Air Asia	09-04-2019	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	09-05-2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

In [10]: `testdf.tail()`

Out[10]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
2666	Air India	06-06-2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	07-06-2023 20:25	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	06-03-2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	07-03-2023 04:25	6h 35m	1 stop	No info
2669	Air India	06-03-2019	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 [11]: traindf.describe()
```

```
Out[11]:
```

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

```
Out[12]:
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
count	2671	2671	2671	2671	2671	2671	2671	2671	2671	2671
unique	11	40	5	6	100	199	704	320	5	6
top	Jet Airways	09-05-2019	Delhi	Cochin	DEL ? BOM ? COK	10:00	19:00	2h 50m	1 stop	No info
freq	897	144	1145	1145	624	62	113	122	1431	2148

```
In [13]: traindf.shape
```

```
Out[13]: (10683, 11)
```

```
In [14]: testdf.shape
```

```
Out[14]: (2671, 10)
```



```
In [15]: traindf.columns
```

```
Out[15]: Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',  
              'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',  
              'Additional_Info', 'Price'],  
              dtype='object')
```

```
In [16]: testdf.columns
```

```
Out[16]: Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',  
              'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',  
              'Additional_Info'],  
              dtype='object')
```

```
In [17]: 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
```

In [18]: 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
```

In [19]: traindf.isnull().sum()

```
Out[19]: 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 [20]: testdf.isnull().sum()
```

```
Out[20]: 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 [21]: traindf.dropna(inplace=True)
```

```
In [22]: traindf.isnull().sum()
```

```
Out[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  
Price              0  
dtype: int64
```

```
In [23]: traindf.shape
```

```
Out[23]: (10682, 11)
```

```
In [24]: traindf['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]: traindf['Source'].value_counts()
```

```
Out[25]: Source
Delhi      4536
Kolkata    2871
Banglore   2197
Mumbai     697
Chennai    381
Name: count, dtype: int64
```

```
In [26]: traindf['Destination'].value_counts()
```

```
Out[26]: Destination
Cochin      4536
Banglore    2871
Delhi       1265
New Delhi   932
Hyderabad   697
Kolkata     381
Name: count, dtype: int64
```

```
In [27]: traaindf['Total_Stops'].value_counts()
```

```
Out[27]: Total_Stops
1 stop      5625
non-stop    3491
2 stops     1520
3 stops       45
4 stops        1
Name: count, dtype: int64
```

```
In [28]: 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
```

Out[28]:

	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	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	2	01-05-2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	09-06-2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	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	09-04-2019	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	09-05-2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10682 rows × 11 columns

```
In [29]: city={"Source":{"Delhi":0,"Kolkata":1,"Banglore":2,
"Mumbai":3,"Chennai":4}}
traindf=traindf.replace(city)
traindf
```

Out[29]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24-03-2019	2	New Delhi	BLR ? DEL	22:20	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	2	01-05-2019	1	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	09-06-2019	0	Cochin	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	19h	2 stops	No info	13882
3	1	12-05-2019	1	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	1	01-03-2019	2	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...
10678	6	09-04-2019	1	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	2	27-04-2019	1	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	0	27-04-2019	2	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	5	01-03-2019	2	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	2	09-05-2019	0	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10682 rows × 11 columns

```
In [30]: destination={"Destination":{"Cochin":0,"Banglore":1,"Delhi":2,
    "New Delhi":3,"Hyderabad":4,"Kolkata":5}}
traindf=traindf.replace(destination)
traindf
```

Out[30]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24-03-2019	2	3	BLR ? DEL	22:20	22-03-2023 01:10	2h 50m	non-stop	No info	3897
1	2	01-05-2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	0	09-06-2019	0	0	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	19h	2 stops	No info	13882
3	1	12-05-2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	1	01-03-2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...
10678	6	09-04-2019	1	1	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	2	27-04-2019	1	1	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	0	27-04-2019	2	2	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	5	01-03-2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	2	09-05-2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10682 rows × 11 columns


```
In [31]: stops={"Total_Stops":{"non-stop":0,"1 stop":1,"2 stops":2,
"3 stops":3,"4 stops":4}}
traindf=traindf.replace(stops)
traindf
```

Out[31]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24-03-2019	2	3	BLR ? DEL	22:20	22-03-2023 01:10	2h 50m	0	No info	3897
1	2	01-05-2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2	No info	7662
2	0	09-06-2019	0	0	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	19h	2	No info	13882
3	1	12-05-2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1	No info	6218
4	1	01-03-2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1	No info	13302
...
10678	6	09-04-2019	1	1	CCU ? BLR	19:55	22:25	2h 30m	0	No info	4107
10679	2	27-04-2019	1	1	CCU ? BLR	20:45	23:20	2h 35m	0	No info	4145
10680	0	27-04-2019	2	2	BLR ? DEL	08:20	11:20	3h	0	No info	7229
10681	5	01-03-2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	0	No info	12648
10682	2	09-05-2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2	No info	11753

10682 rows × 11 columns

In [32]: traindf

Out[32]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24-03-2019	2	3	BLR ? DEL	22:20	22-03-2023 01:10	2h 50m	0	No info	3897
1	2	01-05-2019	1	1	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2	No info	7662
2	0	09-06-2019	0	0	DEL ? LKO ? BOM ? COK	09:25	10-06-2023 04:25	19h	2	No info	13882
3	1	12-05-2019	1	1	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1	No info	6218
4	1	01-03-2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1	No info	13302
...
10678	6	09-04-2019	1	1	CCU ? BLR	19:55	22:25	2h 30m	0	No info	4107
10679	2	27-04-2019	1	1	CCU ? BLR	20:45	23:20	2h 35m	0	No info	4145
10680	0	27-04-2019	2	2	BLR ? DEL	08:20	11:20	3h	0	No info	7229
10681	5	01-03-2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	0	No info	12648
10682	2	09-05-2019	0	0	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2	No info	11753

10682 rows × 11 columns

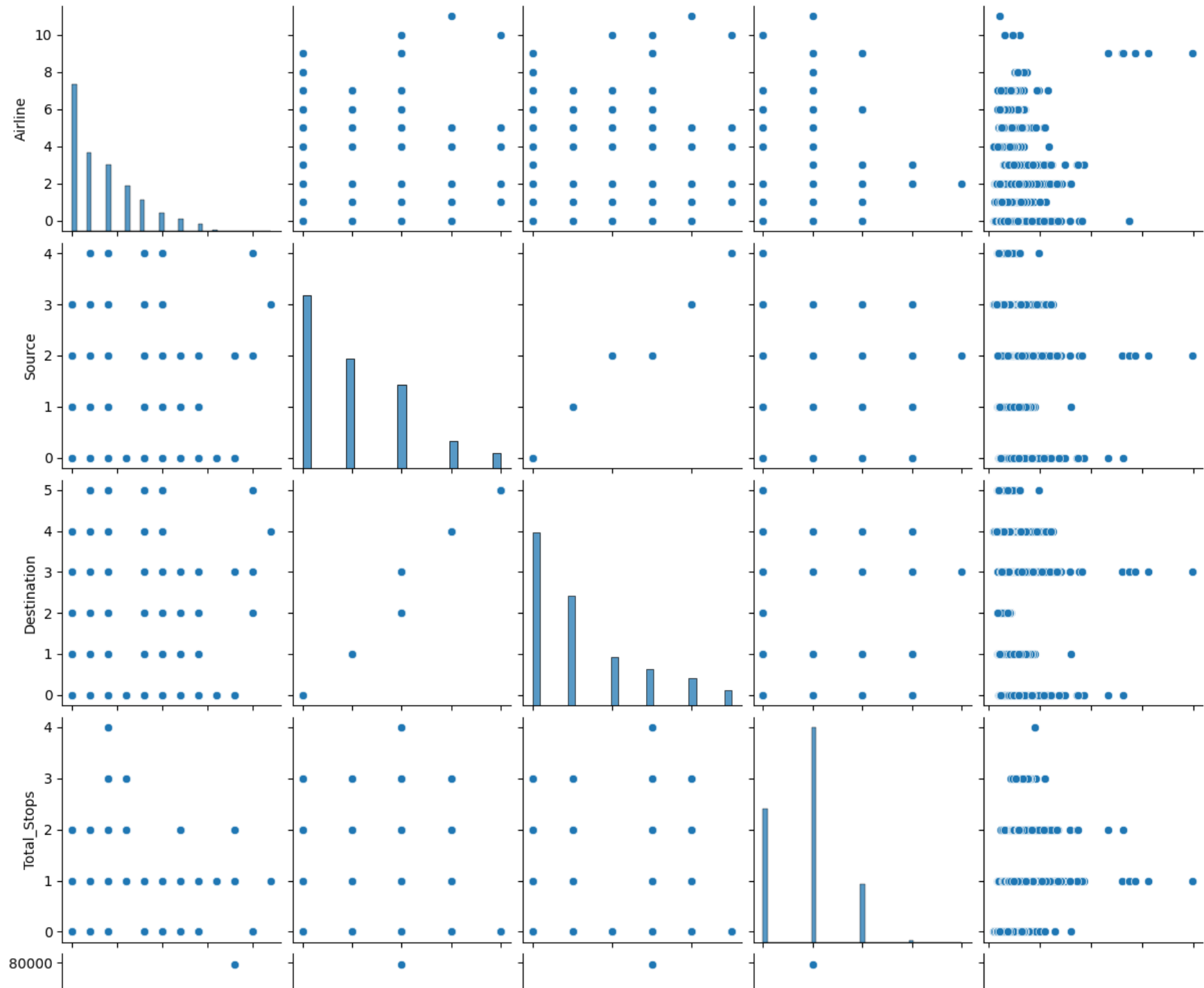
```
In [33]: import seaborn as sns  
df=traindf[['Airline', 'Source', 'Destination', 'Total_Stops', 'Price']]  
sns.heatmap(df.corr(), annot=True)
```

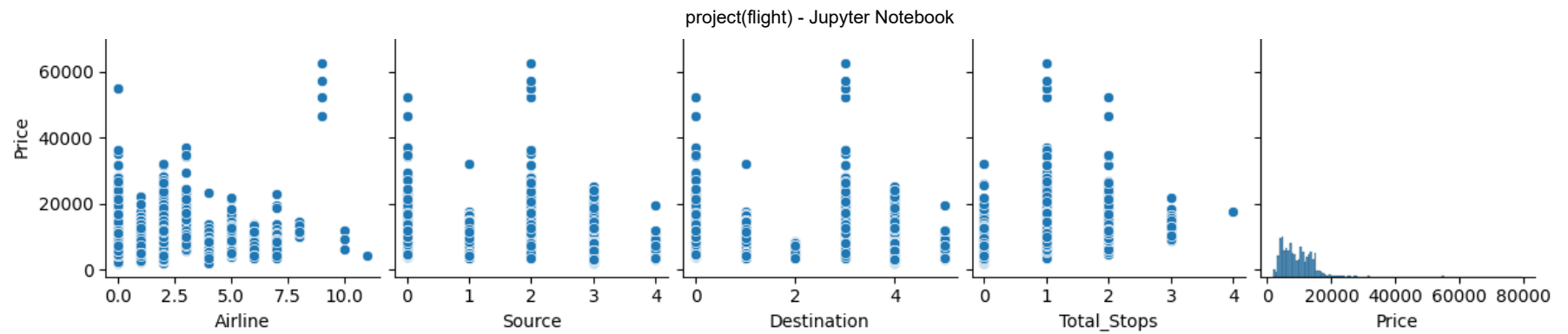
Out[33]: <Axes: >



```
In [34]: sns.pairplot(df)
```

```
Out[34]: <seaborn.axisgrid.PairGrid at 0x12eb28d2210>
```



Linear Regression

```
In [35]: import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [36]: x=df[['Airline', 'Source', 'Destination', "Total_Stops"]]
y=df['Price']
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)
```

```
In [37]: #Data prediction and Evaluation
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
```

7211.098088897471

Out[37]:

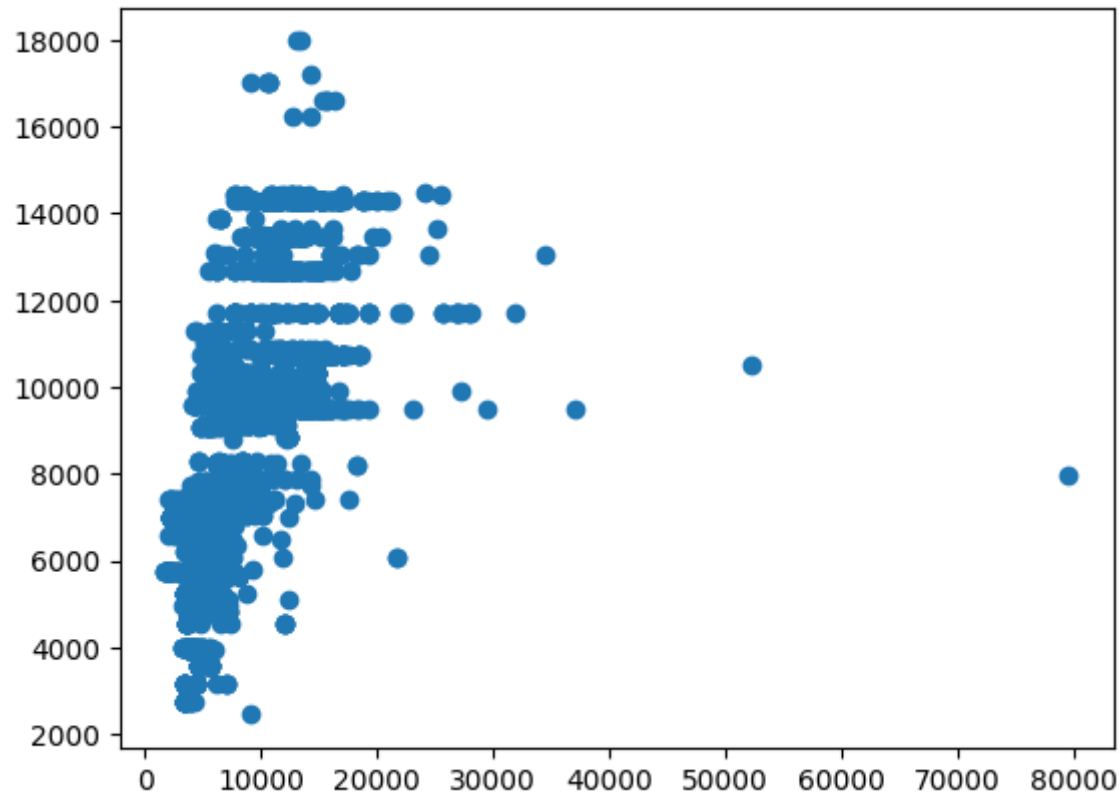
	coefficient
Airline	-418.483922
Source	-3275.073380
Destination	2505.480291
Total_Stops	3541.798053

```
In [38]: score=regr.score(X_test,y_test)
print(score)
```

0.41083048909283415


```
In [39]: predictions=regr.predict(X_test)  
plt.scatter(y_test,predictions)
```

```
Out[39]: <matplotlib.collections.PathCollection at 0x12eb75b9090>
```



```
In [40]: x=np.array(df['Price']).reshape(-1,1)
y=np.array(df['Total_Stops']).reshape(-1,1)
df.dropna(inplace=True)
```

C:\Users\sweet\AppData\Local\Temp\ipykernel_17272\3039801757.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

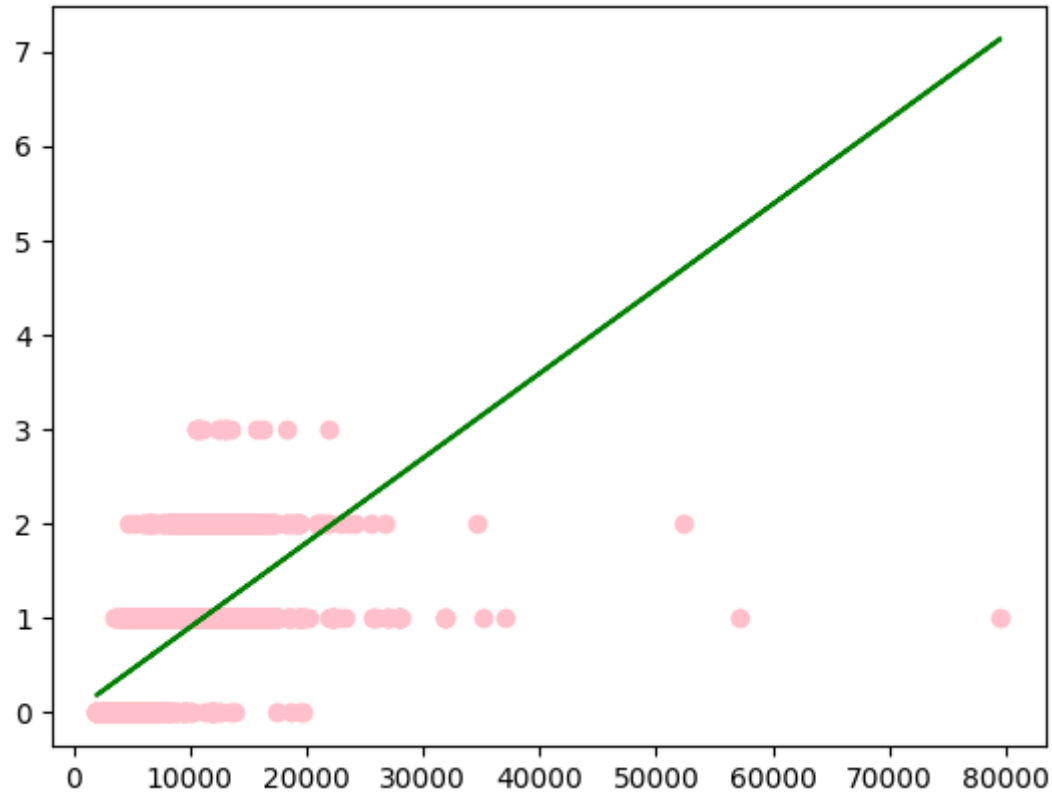
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df.dropna(inplace=True)
```

```
In [41]: 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[41]: ▾ LinearRegression
LinearRegression()
```

```
In [42]: y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='pink')
plt.plot(X_test,y_pred,color='g')
plt.show()
```



Logistic Regression

```
In [43]: x=np.array(df['Price']).reshape(-1,1)
y=np.array(df['Destination']).reshape(-1,1)
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)
import warnings
warnings.simplefilter(action='ignore')
```

C:\Users\sweet\AppData\Local\Temp\ipykernel_17272\1131727007.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 [44]: lr.fit(x_train,y_train)
```

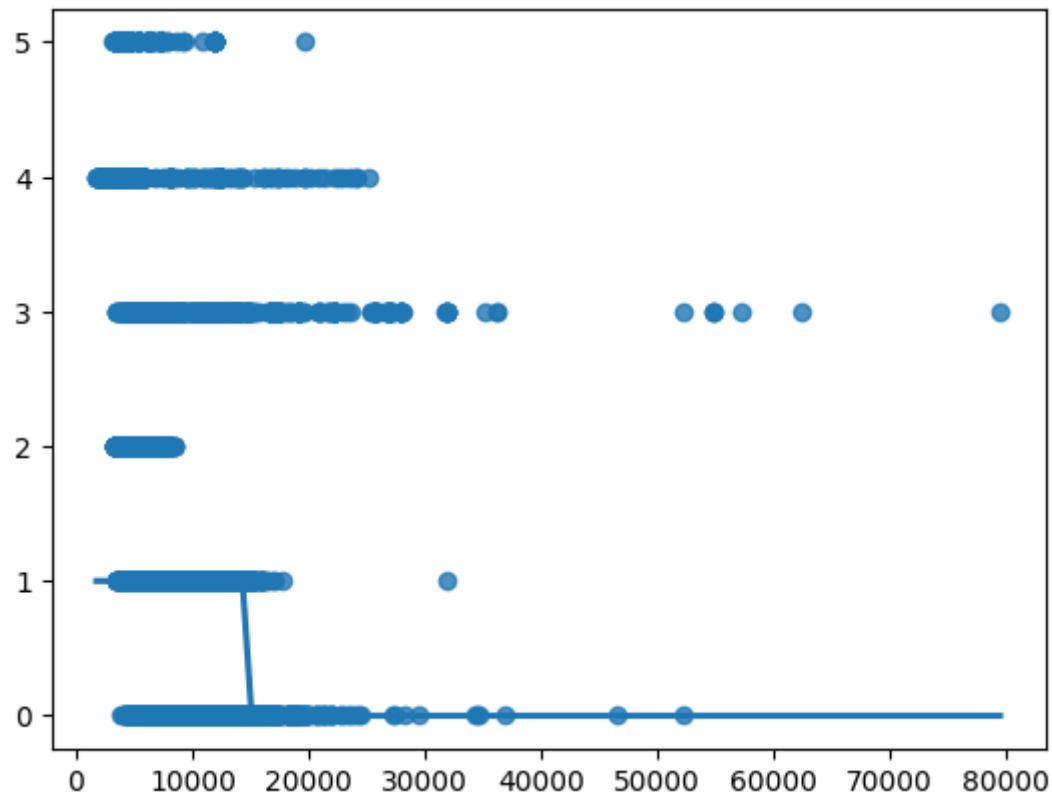
```
Out[44]: LogisticRegression
LogisticRegression(max_iter=10000)
```

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

0.431201248049922

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

```
Out[46]: <Axes: >
```



Decision Tree

```
In [47]: from sklearn.tree import DecisionTreeClassifier  
clf=DecisionTreeClassifier(random_state=0)  
clf.fit(x_train,y_train)
```

```
Out[47]: ▾ DecisionTreeClassifier  
DecisionTreeClassifier(random_state=0)
```

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

0.921996879875195

Random Forest

```
In [49]: from sklearn.ensemble import RandomForestClassifier  
rfc=RandomForestClassifier()  
rfc.fit(X_train,y_train)
```

```
Out[49]: ▾ RandomForestClassifier  
RandomForestClassifier()
```

```
In [50]: 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 [51]: from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(X_train,y_train)
```

```
Out[51]: GridSearchCV
  ▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

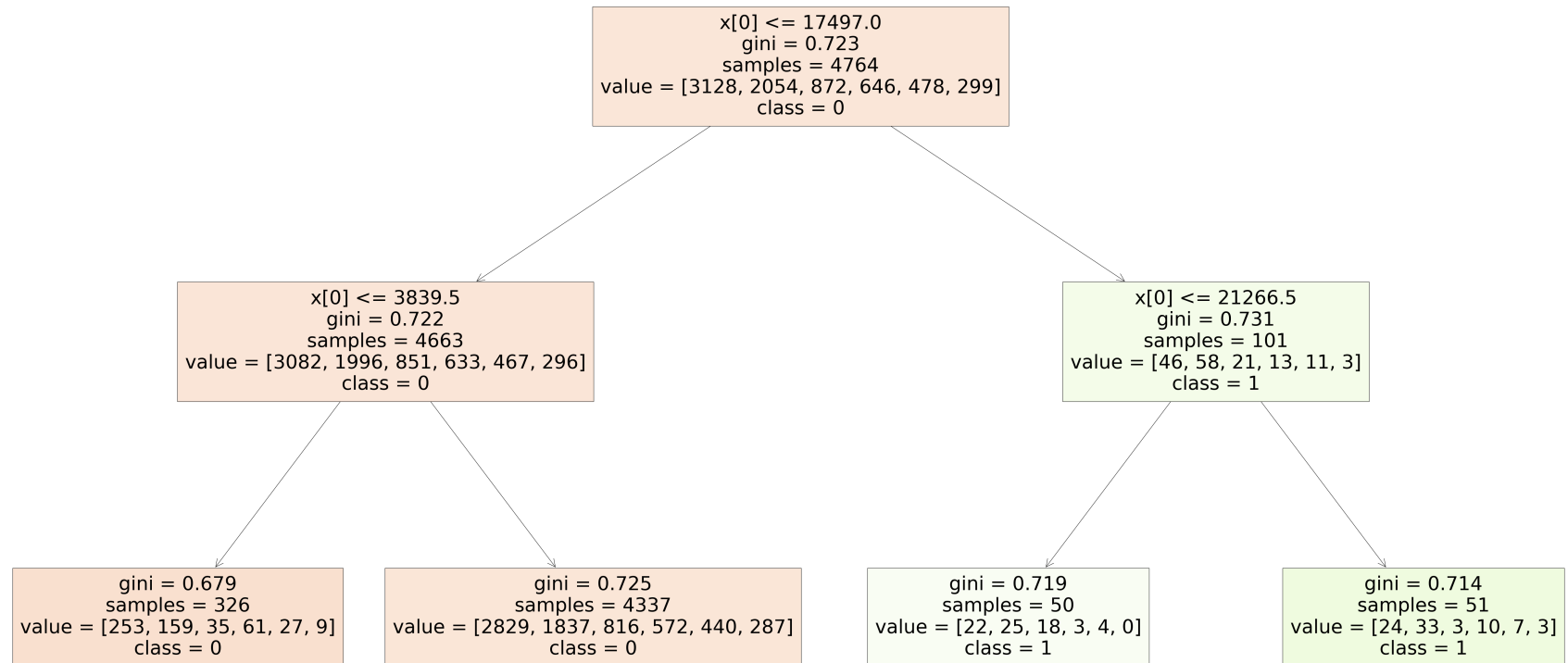
```
In [52]: grid_search.best_score_
```

```
Out[52]: 0.42182694348222594
```

```
In [53]: rf_best=grid_search.best_estimator_
rf_best
```

```
Out[53]: RandomForestClassifier
RandomForestClassifier(max_depth=2, min_samples_leaf=50, n_estimators=10)
```

```
In [54]: 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 [55]: score=rfc.score(x_test,y_test)  
print(score)
```

0.31138845553822153

Conclusion

In the Flight Price Prediction dataset project ,we have to find the best model after applying linear regression,logistic regression,decision tree and random forest.By observing all the order model,i conclude that The DECISION TREE is the best ,model for Flight Price Prediction dataset,because it got high accuracy by comparing all otherws models.