In [1]:
```
#importing basic libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

'''This ipynb is basically for converting data in to specific format to give it to model and then predict the price of the flight, but for predicting you need data in specific format ,for model to understand and thats where feature engineering comes in to play, here we have change the data in such a way that model can understand the data'''

In [2]:
```
train_data=pd.read_excel(r"C:\Users\hemil\OneDrive\Desktop\Data Analyst\EDA PYTHON\Krish Naik Flight Prediction\Data_Train.xlsx")
train_data.head()
```

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

In [3]:
```
test_data=pd.read_excel(r"C:\Users\hemil\OneDrive\Desktop\Data Analyst\EDA PYTHON\Krish Naik Flight Prediction\Test_set.xlsx")
test_data.head()
```

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

In [4]:
```
final_data=pd.concat([train_data,test_data], ignore_index=True)
final_data
```

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	3897.0
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662.0
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882.0
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218.0
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302.0
...	...	...	...	...	...	...	...	...	...	...	...
13349	Air India	6/06/2019	Kolkata	Banglore	CCU → DEL → BLR	20:30	20:25 07 Jun	23h 55m	1 stop	No info	NaN
13350	IndiGo	27/03/2019	Kolkata	Banglore	CCU → BLR	14:20	16:55	2h 35m	non-stop	No info	NaN
13351	Jet Airways	6/03/2019	Delhi	Cochin	DEL → BOM → COK	21:50	04:25 07 Mar	6h 35m	1 stop	No info	NaN
13352	Air India	6/03/2019	Delhi	Cochin	DEL → BOM → COK	04:00	19:15	15h 15m	1 stop	No info	NaN
13353	Multiple carriers	15/06/2019	Delhi	Cochin	DEL → BOM → COK	04:55	19:15	14h 20m	1 stop	No info	NaN

13354 rows × 11 columns

In [5]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Date_of_Journey  13354 non-null  object 
 2   Source           13354 non-null  object 
 3   Destination      13354 non-null  object 
 4   Route            13353 non-null  object 
 5   Dep_Time         13354 non-null  object 
 6   Arrival_Time     13354 non-null  object 
 7   Duration         13354 non-null  object 
 8   Total_Stops      13353 non-null  object 
 9   Additional_Info  13354 non-null  object 
 10  Price            10683 non-null  float64
dtypes: float64(1), object(10)
memory usage: 1.1+ MB
```

In [6]:
```
#We have converted 'Date_of_Journey'column in to 3 columns namely date,month,year.
final_data['Date_of_Journey'] = pd.to_datetime(final_data['Date_of_Journey'], format='%d/%m/%Y', errors='coerce')
final_data['Day']=final_data['Date_of_Journey'].dt.day
final_data['Month']=final_data['Date_of_Journey'].dt.month
final_data['Year']=final_data['Date_of_Journey'].dt.year
```

In [7]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13354 non-null  object
 1   Date_of_Journey  13354 non-null  datetime64[ns]
 2   Source           13354 non-null  object
 3   Destination      13354 non-null  object
 4   Route            13353 non-null  object
 5   Dep_Time         13354 non-null  object
 6   Arrival_Time     13354 non-null  object
 7   Duration         13354 non-null  object
 8   Total_Stops      13353 non-null  object
 9   Additional_Info  13354 non-null  object
 10  Price            10683 non-null  float64
 11  Day              13354 non-null  int32
 12  Month            13354 non-null  int32
 13  Year             13354 non-null  int32
dtypes: datetime64[ns](1), float64(1), int32(3), object(9)
memory usage: 1.3+ MB
```

In [8]:
```
# Safely dropping the 'Date_of_Journey' column
final_data.drop('Date_of_Journey', axis=1 , inplace=True)
```

In [9]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13354 non-null  object
 1   Source           13354 non-null  object
 2   Destination      13354 non-null  object
 3   Route            13353 non-null  object
 4   Dep_Time         13354 non-null  object
 5   Arrival_Time     13354 non-null  object
 6   Duration         13354 non-null  object
 7   Total_Stops      13353 non-null  object
 8   Additional_Info  13354 non-null  object
 9   Price            10683 non-null  float64
 10  Day              13354 non-null  int32
 11  Month            13354 non-null  int32
 12  Year             13354 non-null  int32
dtypes: float64(1), int32(3), object(9)
memory usage: 1.2+ MB
```

In [10]:
```
#Removing Date from Arrival Time,As we needonly time,but not date
# what we have done here is splitted the string where the 'space' comes and then from all that strings, we need the first string.
final_data['Arrival_Time'] = final_data['Arrival_Time'].str.split(' ').str[0]
final_data['Arrival_Hour'] = final_data['Arrival_Time'].str.split(':').str[0]
final_data['Arrival_Minute'] = final_data['Arrival_Time'].str.split(':').str[1]
```

In [11]:
```
# Safely dropping the 'Arrival_Time' column
final_data.drop('Arrival_Time', axis=1 , inplace=True)
final_data
```

Out[11]:

	Airline	Source	Destination	Route	Dep_Time	Duration	Total_Stops	Additional_Info	Price	Day	Month	Year	Arrival_Hour	Arrival_Minute
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	2h 50m	non-stop	No info	3897.0	24	3	2019	01	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	7h 25m	2 stops	No info	7662.0	1	5	2019	13	15
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	19h	2 stops	No info	13882.0	9	6	2019	04	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	5h 25m	1 stop	No info	6218.0	12	5	2019	23	30
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	4h 45m	1 stop	No info	13302.0	1	3	2019	21	35
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13349	Air India	Kolkata	Banglore	CCU → DEL → BLR	20:30	23h 55m	1 stop	No info	NaN	6	6	2019	20	25
13350	IndiGo	Kolkata	Banglore	CCU → BLR	14:20	2h 35m	non-stop	No info	NaN	27	3	2019	16	55
13351	Jet Airways	Delhi	Cochin	DEL → BOM → COK	21:50	6h 35m	1 stop	No info	NaN	6	3	2019	04	25
13352	Air India	Delhi	Cochin	DEL → BOM → COK	04:00	15h 15m	1 stop	No info	NaN	6	3	2019	19	15
13353	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	04:55	14h 20m	1 stop	No info	NaN	15	6	2019	19	15

13354 rows × 14 columns

In [12]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13354 non-null  object
 1   Source           13354 non-null  object
 2   Destination      13354 non-null  object
 3   Route            13353 non-null  object
 4   Dep_Time         13354 non-null  object
 5   Duration         13354 non-null  object
 6   Total_Stops      13353 non-null  object
 7   Additional_Info  13354 non-null  object
 8   Price            10683 non-null  float64
 9   Day              13354 non-null  int32
 10  Month            13354 non-null  int32
 11  Year             13354 non-null  int32
 12  Arrival_Hour     13354 non-null  object
 13  Arrival_Minute   13354 non-null  object
dtypes: float64(1), int32(3), object(10)
memory usage: 1.3+ MB
```

In [13]:
```
#converting data type from object to int
final_data['Arrival_Hour'] = final_data['Arrival_Hour'].astype(int)
final_data['Arrival_Minute'] = final_data['Arrival_Minute'].astype(int)
```

In [14]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          13354 non-null  object
 1   Source           13354 non-null  object
 2   Destination      13354 non-null  object
 3   Route            13353 non-null  object
 4   Dep_Time         13354 non-null  object
 5   Duration         13354 non-null  object
 6   Total_Stops      13353 non-null  object
 7   Additional_Info  13354 non-null  object
 8   Price            10683 non-null  float64
 9   Day              13354 non-null  int32 
 10  Month            13354 non-null  int32 
 11  Year             13354 non-null  int32 
 12  Arrival_Hour     13354 non-null  int32 
 13  Arrival_Minute   13354 non-null  int32 
dtypes: float64(1), int32(5), object(8)
memory usage: 1.2+ MB
```

In [15]:
```
#departure time
#splitting data by ':' in departure time and then converting it in to int
final_data['Dep_Hour'] = final_data['Dep_Time'].str.split(':').str[0]
final_data['Dep_Minute'] = final_data['Dep_Time'].str.split(':').str[1]

#converting data type from object to int
final_data['Dep_Hour'] = final_data['Dep_Hour'].astype(int)
final_data['Dep_Minute'] = final_data['Dep_Minute'].astype(int)

# Safely dropping the 'Arrival_Time' column
final_data.drop('Dep_Time', axis=1 , inplace=True)
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13354 entries, 0 to 13353
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          13354 non-null  object
 1   Source           13354 non-null  object
 2   Destination      13354 non-null  object
 3   Route            13353 non-null  object
 4   Duration         13354 non-null  object
 5   Total_Stops      13353 non-null  object
 6   Additional_Info  13354 non-null  object
 7   Price            10683 non-null  float64
 8   Day              13354 non-null  int32 
 9   Month            13354 non-null  int32 
 10  Year             13354 non-null  int32 
 11  Arrival_Hour     13354 non-null  int32 
 12  Arrival_Minute   13354 non-null  int32 
 13  Dep_Hour         13354 non-null  int32 
 14  Dep_Minute       13354 non-null  int32 
dtypes: float64(1), int32(7), object(7)
memory usage: 1.2+ MB
```

In [16]:
```
final_data = final_data.dropna(subset=['Total_Stops'])
```

In [17]:
```
#total stops
#replacing non-stop with 0
final_data['Total_Stops'] = final_data['Total_Stops'].replace({'non-stop': '0'})
#splitting data by ' ' in departure time and then converting it in to int
final_data['Total_Stops'] = final_data['Total_Stops'].str.split(' ').str[0]

#converting data type from object to int
final_data['Total_Stops'] = final_data['Total_Stops'].astype(int)
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13353 entries, 0 to 13353
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13353 non-null  object
 1   Source           13353 non-null  object
 2   Destination      13353 non-null  object
 3   Route            13353 non-null  object
 4   Duration         13353 non-null  object
 5   Total_Stops      13353 non-null  int32
 6   Additional_Info  13353 non-null  object
 7   Price            10682 non-null  float64
 8   Day              13353 non-null  int32
 9   Month            13353 non-null  int32
 10  Year             13353 non-null  int32
 11  Arrival_Hour     13353 non-null  int32
 12  Arrival_Minute   13353 non-null  int32
 13  Dep_Hour         13353 non-null  int32
 14  Dep_Minute       13353 non-null  int32
dtypes: float64(1), int32(8), object(6)
memory usage: 1.2+ MB

C:\Users\hemil\AppData\Local\Temp\ipykernel_7600\2627101253.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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)
  final_data['Total_Stops'] = final_data['Total_Stops'].replace({'non-stop': '0'})
C:\Users\hemil\AppData\Local\Temp\ipykernel_7600\2627101253.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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)
  final_data['Total_Stops'] = final_data['Total_Stops'].str.split(' ').str[0]
C:\Users\hemil\AppData\Local\Temp\ipykernel_7600\2627101253.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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)
  final_data['Total_Stops'] = final_data['Total_Stops'].astype(int)
```

In [18]:
```
final_data['Total_Stops'].unique()
```

Out[18]:
```
array([0, 2, 1, 3, 4])
```

In [19]:
```
#We will drop Route Column as a whole,as bcz ,irrespective of our stops,price will not get changed.
#Also we have column name total_stops and so we know how much stops are coming in between.
#Their is no need for us to have information about Route to predict the price,Total_Stops is Sufficient.
final_data.drop('Route',axis=1,inplace=True)
```

```
C:\Users\hemil\AppData\Local\Temp\ipykernel_7600\3904364340.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)
  final_data.drop('Route',axis=1,inplace=True)
```

In [20]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13353 entries, 0 to 13353
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13353 non-null  object
 1   Source           13353 non-null  object
 2   Destination      13353 non-null  object
 3   Duration         13353 non-null  object
 4   Total_Stops      13353 non-null  int32
 5   Additional_Info  13353 non-null  object
 6   Price            10682 non-null  float64
 7   Day              13353 non-null  int32
 8   Month            13353 non-null  int32
 9   Year             13353 non-null  int32
 10  Arrival_Hour     13353 non-null  int32
 11  Arrival_Minute   13353 non-null  int32
 12  Dep_Hour         13353 non-null  int32
 13  Dep_Minute       13353 non-null  int32
dtypes: float64(1), int32(8), object(5)
memory usage: 1.1+ MB
```

In [21]:
```
final_data['Additional_Info'].unique()
```

Out[21]:
```
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)
```

In [22]:
```
#This is one method by converting each unique value inside that column as column header and then if that thing is true than 1 is written ,otherwise false=0
# get dummies of 'Additional_Info' column
final_data = pd.get_dummies(final_data, columns=['Additional_Info'],dtype=int)

final_data.head(10)
```

Out[22]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_1 Long layover	Additional_Info_1 Short layover	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additior flig
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	
5	SpiceJet	Kolkata	Banglore	2h 25m	0	3873.0	24	6	2019	11	...	0	0	0	0	0	
6	Jet Airways	Banglore	New Delhi	15h 30m	1	11087.0	12	3	2019	10	...	0	0	0	0	0	
7	Jet Airways	Banglore	New Delhi	21h 5m	1	22270.0	1	3	2019	5	...	0	0	0	0	0	
8	Jet Airways	Banglore	New Delhi	25h 30m	1	11087.0	12	3	2019	10	...	0	0	0	0	0	
9	Multiple carriers	Delhi	Cochin	7h 50m	1	8625.0	27	5	2019	19	...	0	0	0	0	0	

10 rows × 23 columns

In [23]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13353 entries, 0 to 13353
Data columns (total 23 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Airline                                     13353 non-null  object
 1   Source                                      13353 non-null  object
 2   Destination                                 13353 non-null  object
 3   Duration                                    13353 non-null  object
 4   Total_Stops                                 13353 non-null  int32
 5   Price                                       10682 non-null  float64
 6   Day                                         13353 non-null  int32
 7   Month                                       13353 non-null  int32
 8   Year                                        13353 non-null  int32
 9   Arrival_Hour                                13353 non-null  int32
 10  Arrival_Minute                              13353 non-null  int32
 11  Dep_Hour                                    13353 non-null  int32
 12  Dep_Minute                                  13353 non-null  int32
 13  Additional_Info_1 Long layover              13353 non-null  int32
 14  Additional_Info_1 Short layover             13353 non-null  int32
 15  Additional_Info_2 Long layover              13353 non-null  int32
 16  Additional_Info_Business class              13353 non-null  int32
 17  Additional_Info_Change airports             13353 non-null  int32
 18  Additional_Info_In-flight meal not included 13353 non-null  int32
 19  Additional_Info_No Info                     13353 non-null  int32
 20  Additional_Info_No check-in baggage included 13353 non-null  int32
 21  Additional_Info_No info                     13353 non-null  int32
 22  Additional_Info_Red-eye flight              13353 non-null  int32
dtypes: float64(1), int32(18), object(4)
memory usage: 1.5+ MB
```

In [24]:
```
#below code is for transforming data of duration(object) in to minutes(int)
final_data['Hours'] = final_data['Duration'].str.split('h').str[0]
final_data['Minutes'] = final_data['Duration'].str.split(' ').str[1]

final_data.head()
```

Out[24]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Additional_Info_No Info	Addit che
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	

5 rows × 25 columns

In [25]:
```
final_data['Minutes'] = final_data['Minutes'].str.replace('m',' ',regex=False)
final_data.head()
```

Out[25]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Additional_Info_No Info	Addit chec
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	

5 rows × 25 columns

In [26]:
```
#If flight is exactly of 19hrs then we are getting NAN in minutes column and that why making it 0 is necessary for further calculation.
final_data = final_data.fillna(0)
final_data
```

Out[26]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Additional_Info_No Info	A
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
13349	Air India	Kolkata	Banglore	23h 55m	1	0.0	6	6	2019	20	...	0	0	0	0	0	
13350	IndiGo	Kolkata	Banglore	2h 35m	0	0.0	27	3	2019	16	...	0	0	0	0	0	
13351	Jet Airways	Delhi	Cochin	6h 35m	1	0.0	6	3	2019	4	...	0	0	0	0	0	
13352	Air India	Delhi	Cochin	15h 15m	1	0.0	6	3	2019	19	...	0	0	0	0	0	
13353	Multiple carriers	Delhi	Cochin	14h 20m	1	0.0	15	6	2019	19	...	0	0	0	0	0	

13353 rows × 25 columns

In [27]:
```
final_data['Hours'].unique()
```

Out[27]:
```
array(['2', '7', '19', '5', '4', '15', '21', '25', '13', '12', '26', '22',
       '23', '20', '10', '6', '11', '8', '16', '3', '27', '1', '14', '9',
       '18', '17', '24', '30', '28', '29', '37', '34', '38', '35', '36',
       '47', '33', '32', '31', '42', '39', '5m', '41', '40'], dtype=object)
```

In [28]:
```
#we need to change this 5m with 5 ,otherwise their will be an error in changing it to int.
final_data['Hours'] = final_data['Hours'].str.replace('m','',regex=False)
final_data.head()
```

Out[28]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Additional_Info_No Info	Addit chec
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	

5 rows × 25 columns

In [29]:
```
final_data['Hours']=final_data['Hours'].astype(int)
final_data['Minutes']=final_data['Minutes'].astype(int)

final_data['Duration_Minutes'] = final_data['Hours']*60 + final_data['Minutes']
final_data.head()
```

Out[29]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Day	Month	Year	Arrival_Hour	...	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Additional_Info_No Info	Additional_Info_No check-in baggage included	Add
0	IndiGo	Banglore	New Delhi	2h 50m	0	3897.0	24	3	2019	1	...	0	0	0	0	0	
1	Air India	Kolkata	Banglore	7h 25m	2	7662.0	1	5	2019	13	...	0	0	0	0	0	
2	Jet Airways	Delhi	Cochin	19h	2	13882.0	9	6	2019	4	...	0	0	0	0	0	
3	IndiGo	Kolkata	Banglore	5h 25m	1	6218.0	12	5	2019	23	...	0	0	0	0	0	
4	IndiGo	Banglore	New Delhi	4h 45m	1	13302.0	1	3	2019	21	...	0	0	0	0	0	

5 rows × 26 columns

In [30]:
```
# Safely dropping the columns
final_data.drop('Hours', axis=1 , inplace=True)
final_data.drop('Minutes', axis=1 , inplace=True)
final_data.drop('Duration', axis=1 , inplace=True)
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13353 entries, 0 to 13353
Data columns (total 23 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Airline                                     13353 non-null  object
 1   Source                                      13353 non-null  object
 2   Destination                                 13353 non-null  object
 3   Total_Stops                                 13353 non-null  int32
 4   Price                                       13353 non-null  float64
 5   Day                                         13353 non-null  int32
 6   Month                                       13353 non-null  int32
 7   Year                                        13353 non-null  int32
 8   Arrival_Hour                                13353 non-null  int32
 9   Arrival_Minute                              13353 non-null  int32
 10  Dep_Hour                                    13353 non-null  int32
 11  Dep_Minute                                  13353 non-null  int32
 12  Additional_Info_1 Long layover              13353 non-null  int32
 13  Additional_Info_1 Short layover             13353 non-null  int32
 14  Additional_Info_2 Long layover              13353 non-null  int32
 15  Additional_Info_Business class              13353 non-null  int32
 16  Additional_Info_Change airports             13353 non-null  int32
 17  Additional_Info_In-flight meal not included 13353 non-null  int32
 18  Additional_Info_No Info                     13353 non-null  int32
 19  Additional_Info_No check-in baggage included 13353 non-null  int32
 20  Additional_Info_No info                     13353 non-null  int32
 21  Additional_Info_Red-eye flight              13353 non-null  int32
 22  Duration_Minutes                            13353 non-null  int32
dtypes: float64(1), int32(19), object(3)
memory usage: 1.5+ MB
```

In [31]:
```
final_data['Airline'].unique()
```

Out[31]:
```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

In [32]:
```
final_data['Source'].unique()
```

Out[32]:
```
array(['Banglore', 'Kolkata', 'Delhi', 'Chennai', 'Mumbai'], dtype=object)
```

In [33]:
```
final_data['Destination'].unique()
```

Out[33]:
```
array(['New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi', 'Hyderabad'],
      dtype=object)
```

In [34]:
```
#By using Label encoder we have assingned a unique number to every unique string in Airline ,Destination, Source.
#That means New Delhi=1,Banglore=2,Cochin=3...
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
final_data['Airline']=labelencoder.fit_transform(final_data['Airline'])
final_data['Destination']=labelencoder.fit_transform(final_data['Destination'])
final_data['Source']=labelencoder.fit_transform(final_data['Source'])


final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13353 entries, 0 to 13353
Data columns (total 23 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Airline                                     13353 non-null  int32
 1   Source                                      13353 non-null  int32
 2   Destination                                 13353 non-null  int32
 3   Total_Stops                                 13353 non-null  int32
 4   Price                                       13353 non-null  float64
 5   Day                                         13353 non-null  int32
 6   Month                                       13353 non-null  int32
 7   Year                                        13353 non-null  int32
 8   Arrival_Hour                                13353 non-null  int32
 9   Arrival_Minute                              13353 non-null  int32
 10  Dep_Hour                                    13353 non-null  int32
 11  Dep_Minute                                  13353 non-null  int32
 12  Additional_Info_1 Long layover              13353 non-null  int32
 13  Additional_Info_1 Short layover             13353 non-null  int32
 14  Additional_Info_2 Long layover              13353 non-null  int32
 15  Additional_Info_Business class              13353 non-null  int32
 16  Additional_Info_Change airports             13353 non-null  int32
 17  Additional_Info_In-flight meal not included 13353 non-null  int32
 18  Additional_Info_No Info                     13353 non-null  int32
 19  Additional_Info_No check-in baggage included 13353 non-null  int32
 20  Additional_Info_No info                     13353 non-null  int32
 21  Additional_Info_Red-eye flight              13353 non-null  int32
 22  Duration_Minutes                            13353 non-null  int32
dtypes: float64(1), int32(22)
memory usage: 1.3 MB
```

In [35]:
```
final_data.head()
#Now we can split data back in to test and train and then give train dataset to machine learning model and so on.
```

Out[35]:

	Airline	Source	Destination	Total_Stops	Price	Day	Month	Year	Arrival_Hour	Arrival_Minute	...	Additional_Info_1 Short layover	Additional_Info_2 Long layover	Additional_Info_Business class	Additional_Info_Change airports	Additional_Info_In-flight meal not included	Add
0	3	0	5	0	3897.0	24	3	2019	1	10	...	0	0	0	0	0	
1	1	3	0	2	7662.0	1	5	2019	13	15	...	0	0	0	0	0	
2	4	2	1	2	13882.0	9	6	2019	4	25	...	0	0	0	0	0	
3	3	3	0	1	6218.0	12	5	2019	23	30	...	0	0	0	0	0	
4	3	0	5	1	13302.0	1	3	2019	21	35	...	0	0	0	0	0	

5 rows × 23 columns