In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df1=pd.read_excel(r'Data_Train (1).xlsx')
df1.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]:
```
df1.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 [4]:
```
df1.shape
```

Out[4]:
```
(10683, 11)
```

In [5]:
```
df1.isnull().sum()
```

Out[5]:
```
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 [6]:
```
df1.dropna(inplace=True)
```

In [7]:
```
df1.isnull().sum()
```

Out[7]:
```
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 [8]:
```
df1["Date_of_Journey"]= pd.to_datetime(df1["Date_of_Journey"])
df1['Journey_day']=df1["Date_of_Journey"].dt.day
df1['Journey_month']=df1['Date_of_Journey'].dt.month
df1.drop(['Date_of_Journey'],axis=1,inplace=True)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\2383354031.py:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=Fa
lse (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.
  df1["Date_of_Journey"]= pd.to_datetime(df1["Date_of_Journey"])
```

In [9]:
```
df1.head()
```

Out[9]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	5	1
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	6	9
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	5	12
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	3	1

In [10]:
```
df1['Dep_hour']=pd.to_datetime(df1['Dep_Time']).dt.hour
df1['Dep_minute']=pd.to_datetime(df1['Dep_Time']).dt.minute
df1.head()
```

Out[10]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hou
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	22
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	5	1	5
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	6	9	9
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	5	12	18
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	3	1	16

In [11]:
```
df1.drop(['Dep_Time'],axis=1,inplace=True)
df1.head()
```

Out[11]:

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_minu
0	IndiGo	Banglore	New Delhi	BLR → DEL	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	22	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	13:15	7h 25m	2 stops	No info	7662	5	1	5	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	04:25 10 Jun	19h	2 stops	No info	13882	6	9	9	
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	23:30	5h 25m	1 stop	No info	6218	5	12	18	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	21:35	4h 45m	1 stop	No info	13302	3	1	16	

In [12]:

```
df1['Arrival_hour']=pd.to_datetime(df1['Arrival_Time']).dt.hour
df1['Arrival_minute']=pd.to_datetime(df1['Arrival_Time']).dt.minute
df1.head()
```

Out[12]:

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_minu
0	IndiGo	Banglore	New Delhi	BLR → DEL	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	22	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	13:15	7h 25m	2 stops	No info	7662	5	1	5	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	04:25 10 Jun	19h	2 stops	No info	13882	6	9	9	
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	23:30	5h 25m	1 stop	No info	6218	5	12	18	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	21:35	4h 45m	1 stop	No info	13302	3	1	16	

In [13]:

```
df1.drop(['Arrival_Time'],axis=1,inplace=True)
df1.head()
```

Out[13]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_ho
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	22	20	
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	5	1	5	50	
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2 stops	No info	13882	6	9	9	25	
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	5h 25m	1 stop	No info	6218	5	12	18	5	
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	4h 45m	1 stop	No info	13302	3	1	16	50	

In [14]:
```
df1.drop(['Route','Additional_Info'],axis=1,inplace=True)
df1.head()
```

Out[14]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute
0	IndiGo	Banglore	New Delhi	2h 50m	non-stop	3897	24	3	22	20	1	10
1	Air India	Kolkata	Banglore	7h 25m	2 stops	7662	5	1	5	50	13	15
2	Jet Airways	Delhi	Cochin	19h	2 stops	13882	6	9	9	25	4	25
3	IndiGo	Kolkata	Banglore	5h 25m	1 stop	6218	5	12	18	5	23	30
4	IndiGo	Banglore	New Delhi	4h 45m	1 stop	13302	3	1	16	50	21	35

In [15]:

```python
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df1["Source"] = le.fit_transform(df1["Source"])
df1["Destination"] = le.fit_transform(df1["Destination"])
```

In [16]:

```python
df1['Total_Stops'].value_counts()
```

Out[16]:

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

In [17]:

```python
df1.replace({'non-stop':0,'1 stop':1,'2 stops':2,'3 stops':3,'4 stops':4},inplace=True)
```

In [18]:

```python
df1.head()
```

Out[18]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute
0	IndiGo	0	5	2h 50m	0	3897	24	3	22	20	1	10
1	Air India	3	0	7h 25m	2	7662	5	1	5	50	13	15
2	Jet Airways	2	1	19h	2	13882	6	9	9	25	4	25
3	IndiGo	3	0	5h 25m	1	6218	5	12	18	5	23	30
4	IndiGo	0	5	4h 45m	1	13302	3	1	16	50	21	35

In [19]:
```
duration=list(df1['Duration'])
for i in range(len(duration)):
    if len(duration[i].split())!=2:
        if 'h' in duration[i]:
            duration[i]=duration[i].strip()+ ' 0m'
        else:
            duration[i]='0h '+ duration[i]

duration_hours=[]
duration_minutes=[]
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep='h')[0]))
    duration_minutes.append(int(duration[i].split(sep='m')[0].split()[-1]))
df1['duration_hours']=duration_hours
df1['duration_minutes']=duration_minutes
```

In [20]:
```
df1.head()
```

Out[20]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	durati
0	IndiGo	0	5	2h 50m	0	3897	24	3	22	20	1	10	
1	Air India	3	0	7h 25m	2	7662	5	1	5	50	13	15	
2	Jet Airways	2	1	19h	2	13882	6	9	9	25	4	25	
3	IndiGo	3	0	5h 25m	1	6218	5	12	18	5	23	30	
4	IndiGo	0	5	4h 45m	1	13302	3	1	16	50	21	35	

In [21]:
```
df1.drop(['Duration'],axis=1,inplace=True)
df1.head()
```

Out[21]:

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hours
0	IndiGo	0	5	0	3897	24	3	22	20	1	10	2
1	Air India	3	0	2	7662	5	1	5	50	13	15	7
2	Jet Airways	2	1	2	13882	6	9	9	25	4	25	19
3	IndiGo	3	0	1	6218	5	12	18	5	23	30	5
4	IndiGo	0	5	1	13302	3	1	16	50	21	35	4

In [22]:
```
df1.Airline.value_counts()
```

Out[22]:
```
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: Airline, dtype: int64
```

In [23]:
```
Airlines = {
    "Jet Airways":1,
    "IndiGo":2,
    "Air India":3,
    "Multiple carriers":4,
    "SpiceJet":5 , "Vistara":6 ,"Air Asia":7 , "GoAir":8,
}

df1.loc[: , "Airline"] = df1["Airline"].map(Airlines)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\3778155136.py:9: FutureWarning: In a future version, `df.iloc[:, i] = newvals` wi
ll attempt to set the values inplace instead of always setting a new array. To retain the old behavior, use either `df[df.column
s[i]] = newvals` or, if columns are non-unique, `df.isetitem(i, newvals)`
  df1.loc[: , "Airline"] = df1["Airline"].map(Airlines)
```

In [24]:

```
df1.isna().sum()
```

Out[24]:

```
Airline            23
Source              0
Destination         0
Total_Stops         0
Price               0
Journey_day         0
Journey_month       0
Dep_hour            0
Dep_minute          0
Arrival_hour        0
Arrival_minute      0
duration_hours      0
duration_minutes    0
dtype: int64
```

In [25]:

```
df1 = df1[df1.Airline != 'Trujet']

df1 = df1[df1.Airline != 'Multiple carriers Premium economy']
df1 = df1[df1.Airline != 'Jet Airways Business']
df1 = df1[df1.Airline != 'Vistara Premium economy']
```

In [26]:

```
df1.isna().sum()
```

```
Out[26]: Airline            23
         Source              0
         Destination         0
         Total_Stops         0
         Price               0
         Journey_day         0
         Journey_month       0
         Dep_hour            0
         Dep_minute          0
         Arrival_hour        0
         Arrival_minute      0
         duration_hours      0
         duration_minutes    0
         dtype: int64
```

```
In [27]: df1.head()
```

Out[27]:

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hours
0	2.0	0	5	0	3897	24	3	22	20	1	10	2
1	3.0	3	0	2	7662	5	1	5	50	13	15	7
2	1.0	2	1	2	13882	6	9	9	25	4	25	19
3	2.0	3	0	1	6218	5	12	18	5	23	30	5
4	2.0	0	5	1	13302	3	1	16	50	21	35	4

```
In [28]: df1.dropna(inplace=True)
```

```
In [29]: df1.isnull().sum()
```

Out[29]:
```
Airline            0
Source             0
Destination        0
Total_Stops        0
Price              0
Journey_day        0
Journey_month      0
Dep_hour           0
Dep_minute         0
Arrival_hour       0
Arrival_minute     0
duration_hours     0
duration_minutes   0
dtype: int64
```

In [30]:
```
df2=pd.read_excel(r'Test_set (1).xlsx')
df2.head()
```

Out[30]:

	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 [31]:
```
df2.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 [32]:
```
df2.isnull().sum()
```

Out[32]:
```
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 [33]:
```
df2.shape
```

Out[33]:
```
(2671, 10)
```

In [34]:
```
df2["Date_of_Journey"]= pd.to_datetime(df2["Date_of_Journey"])
df2['Journey_day']=df2["Date_of_Journey"].dt.day
df2['Journey_month']=df2['Date_of_Journey'].dt.month
df2.drop(['Date_of_Journey'],axis=1,inplace=True)
df2.head()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\1340305263.py:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=Fa
lse (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.
  df2["Date_of_Journey"]= pd.to_datetime(df2["Date_of_Journey"])
```

Out[34]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Journey_day	Journey_month
0	Jet Airways	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info	6	6
1	IndiGo	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info	5	12
2	Jet Airways	Delhi	Cochin	DEL → BOM → COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included	21	5
3	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info	21	5
4	Air Asia	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info	24	6

In [35]:

```
df2['Dep_hour']=pd.to_datetime(df2['Dep_Time']).dt.hour
df2['Dep_minute']=pd.to_datetime(df2['Dep_Time']).dt.minute
df2.head()
```

Out[35]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Journey_day	Journey_month	Dep_hour	Dep_
0	Jet Airways	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info	6	6	17	
1	IndiGo	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info	5	12	6	
2	Jet Airways	Delhi	Cochin	DEL → BOM → COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included	21	5	19	
3	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info	21	5	8	
4	Air Asia	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info	24	6	23	

In [36]:

```
df2.drop(['Dep_Time'],axis=1,inplace=True)
df2.head()
```

Out[36]:

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Journey_day	Journey_month	Dep_hour	Dep_minute
0	Jet Airways	Delhi	Cochin	DEL → BOM → COK	04:25 07 Jun	10h 55m	1 stop	No info	6	6	17	30
1	IndiGo	Kolkata	Banglore	CCU → MAA → BLR	10:20	4h	1 stop	No info	5	12	6	20
2	Jet Airways	Delhi	Cochin	DEL → BOM → COK	19:00 22 May	23h 45m	1 stop	In-flight meal not included	21	5	19	15
3	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	21:00	13h	1 stop	No info	21	5	8	0
4	Air Asia	Banglore	Delhi	BLR → DEL	02:45 25 Jun	2h 50m	non-stop	No info	24	6	23	55

In [37]:

```
df2['Arrival_hour']=pd.to_datetime(df2['Arrival_Time']).dt.hour
df2['Arrival_minute']=pd.to_datetime(df2['Arrival_Time']).dt.minute
df2.head()
```

Out[37]:

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Journey_day	Journey_month	Dep_hour	Dep_minute	Ar
0	Jet Airways	Delhi	Cochin	DEL → BOM → COK	04:25 07 Jun	10h 55m	1 stop	No info	6	6	17	30	
1	IndiGo	Kolkata	Banglore	CCU → MAA → BLR	10:20	4h	1 stop	No info	5	12	6	20	
2	Jet Airways	Delhi	Cochin	DEL → BOM → COK	19:00 22 May	23h 45m	1 stop	In-flight meal not included	21	5	19	15	
3	Multiple carriers	Delhi	Cochin	DEL → BOM → COK	21:00	13h	1 stop	No info	21	5	8	0	
4	Air Asia	Banglore	Delhi	BLR → DEL	02:45 25 Jun	2h 50m	non-stop	No info	24	6	23	55	

In [38]:

```
df2.drop(['Arrival_Time','Route','Additional_Info'],axis=1,inplace=True)
df2.head()
```

Out[38]:

	Airline	Source	Destination	Duration	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute
0	Jet Airways	Delhi	Cochin	10h 55m	1 stop	6	6	17	30	4	25
1	IndiGo	Kolkata	Banglore	4h	1 stop	5	12	6	20	10	20
2	Jet Airways	Delhi	Cochin	23h 45m	1 stop	21	5	19	15	19	0
3	Multiple carriers	Delhi	Cochin	13h	1 stop	21	5	8	0	21	0
4	Air Asia	Banglore	Delhi	2h 50m	non-stop	24	6	23	55	2	45

In [39]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df2["Source"] = le.fit_transform(df2["Source"])
df2["Destination"] = le.fit_transform(df2["Destination"])
```

In [40]:
```
df2['Total_Stops'].value_counts()
```

Out[40]:
```
1 stop      1431
non-stop     849
2 stops      379
3 stops       11
4 stops        1
Name: Total_Stops, dtype: int64
```

In [41]:
```
df2.replace({'non-stop':0,'1 stop':1,'2 stops':2,'3 stops':3,'4 stops':4},inplace=True)
```

In [42]:
```
df2.head()
```

Out[42]:

	Airline	Source	Destination	Duration	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute
0	Jet Airways	2	1	10h 55m	1	6	6	17	30	4	25
1	IndiGo	3	0	4h	1	5	12	6	20	10	20
2	Jet Airways	2	1	23h 45m	1	21	5	19	15	19	0
3	Multiple carriers	2	1	13h	1	21	5	8	0	21	0
4	Air Asia	0	2	2h 50m	0	24	6	23	55	2	45

In [43]:
```
duration=list(df2['Duration'])
for i in range(len(duration)):
    if len(duration[i].split())!=2:
        if 'h' in duration[i]:
            duration[i]=duration[i].strip()+ ' 0m'
        else:
            duration[i]='0h '+ duration[i]

test_duration_hours=[]
test_duration_minutes=[]
for i in range(len(duration)):
    test_duration_hours.append(int(duration[i].split(sep='h')[0]))
    test_duration_minutes.append(int(duration[i].split(sep='m')[0].split()[-1]))
df2['duration_hours']=test_duration_hours
df2['duration_minutes']=test_duration_minutes
```

In [44]:
```
df2.head()
```

Out[44]:

	Airline	Source	Destination	Duration	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hou
0	Jet Airways	2	1	10h 55m	1	6	6	17	30	4	25	
1	IndiGo	3	0	4h	1	5	12	6	20	10	20	
2	Jet Airways	2	1	23h 45m	1	21	5	19	15	19	0	
3	Multiple carriers	2	1	13h	1	21	5	8	0	21	0	
4	Air Asia	0	2	2h 50m	0	24	6	23	55	2	45	

In [45]:
```
df2.drop(['Duration'],axis=1,inplace=True)
df2.head()
```

Out[45]:

	Airline	Source	Destination	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hours	duratic
0	Jet Airways	2	1	1	6	6	17	30	4	25	10	
1	IndiGo	3	0	1	5	12	6	20	10	20	4	
2	Jet Airways	2	1	1	21	5	19	15	19	0	23	
3	Multiple carriers	2	1	1	21	5	8	0	21	0	13	
4	Air Asia	0	2	0	24	6	23	55	2	45	2	

In [46]:

```
df2.Airline.value_counts()
```

Out[46]:

```
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: Airline, dtype: int64
```

In [47]:

```
Airlines = {
    "Jet Airways":1,
    "IndiGo":2,
    "Air India":3,
    "Multiple carriers":4,
    "SpiceJet":5 , "Vistara":6 ,"Air Asia":7 , "GoAir":8,
}

df2.loc[: , "Airline"] = df2["Airline"].map(Airlines)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\771253343.py:9: FutureWarning: In a future version, `df.iloc[:, i] = newvals` wil
l attempt to set the values inplace instead of always setting a new array. To retain the old behavior, use either `df[df.columns
[i]] = newvals` or, if columns are non-unique, `df.isetitem(i, newvals)`
  df2.loc[: , "Airline"] = df2["Airline"].map(Airlines)
```

In [48]:
```
df2.isna().sum()
```

Out[48]:
```
Airline             7
Source              0
Destination         0
Total_Stops         0
Journey_day         0
Journey_month       0
Dep_hour            0
Dep_minute          0
Arrival_hour        0
Arrival_minute      0
duration_hours      0
duration_minutes    0
dtype: int64
```

In [49]:
```
df2 = df2[df2.Airline != 'Multiple carriers Premium economy']
df2 = df2[df2.Airline != 'Jet Airways Business']
df2 = df2[df2.Airline != 'Vistara Premium economy']
```

In [50]:
```
df2.isnull().sum()
```

Out[50]:
```
Airline             7
Source              0
Destination         0
Total_Stops         0
Journey_day         0
Journey_month       0
Dep_hour            0
Dep_minute          0
Arrival_hour        0
Arrival_minute      0
duration_hours      0
duration_minutes    0
dtype: int64
```

In [51]:
```
df2.dropna(inplace=True)
```

In [52]:
```
df2.head()
```

Out[52]:

	Airline	Source	Destination	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hours	duratior
0	1.0	2	1	1	6	6	17	30	4	25	10	
1	2.0	3	0	1	5	12	6	20	10	20	4	
2	1.0	2	1	1	21	5	19	15	19	0	23	
3	4.0	2	1	1	21	5	8	0	21	0	13	
4	7.0	0	2	0	24	6	23	55	2	45	2	

In [53]:
```
x=df1.drop(['Price'],axis=1)
x.head()
```

Out[53]:

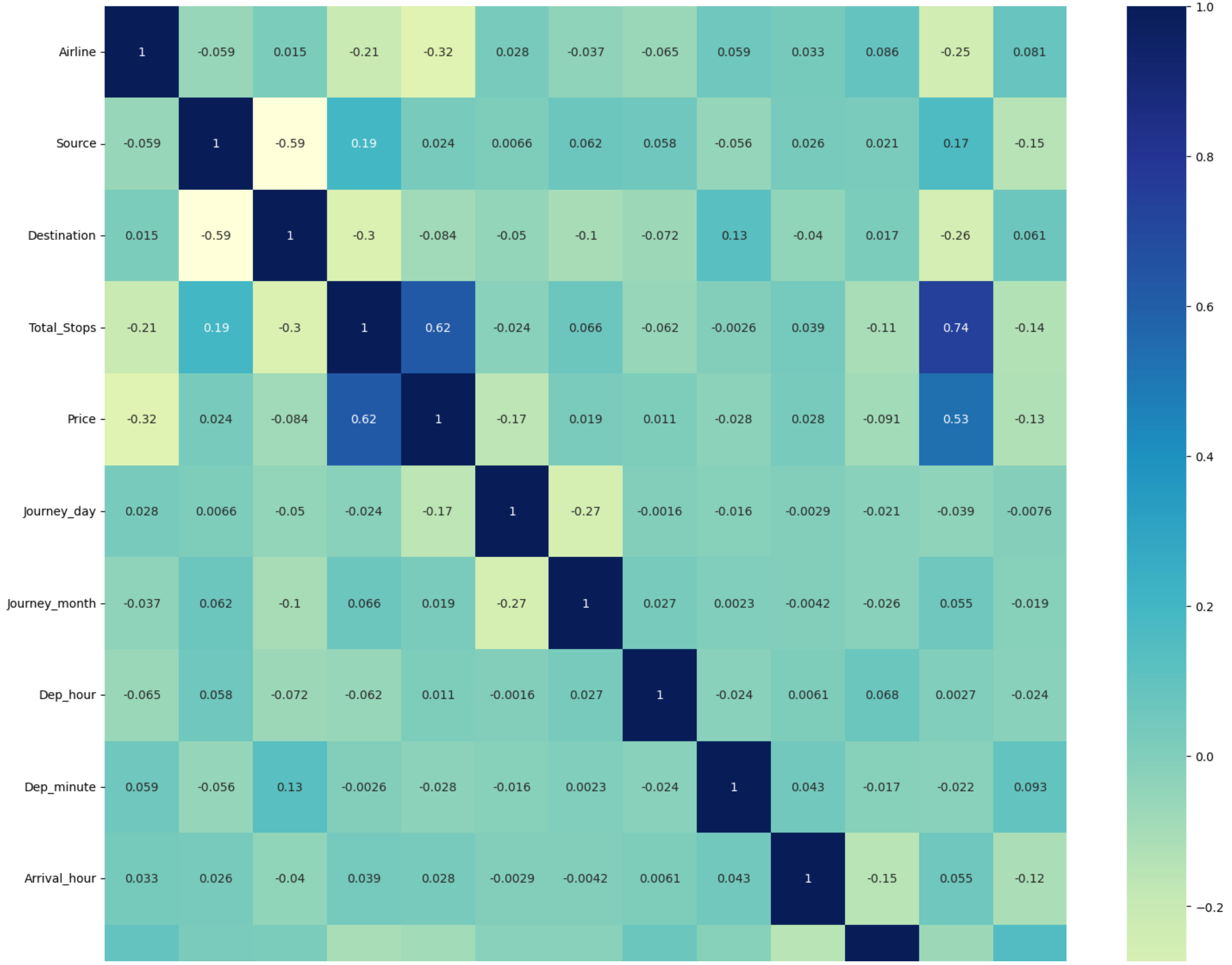
	Airline	Source	Destination	Total_Stops	Journey_day	Journey_month	Dep_hour	Dep_minute	Arrival_hour	Arrival_minute	duration_hours	duratior
0	2.0	0	5	0	24	3	22	20	1	10	2	
1	3.0	3	0	2	5	1	5	50	13	15	7	
2	1.0	2	1	2	6	9	9	25	4	25	19	
3	2.0	3	0	1	5	12	18	5	23	30	5	
4	2.0	0	5	1	3	1	16	50	21	35	4	

In [54]:
```
y=df1['Price']
y.head()
```

Out[54]:
```
0     3897
1     7662
2    13882
3     6218
4    13302
Name: Price, dtype: int64
```

In [55]:
```
plt.figure(figsize=(18,18))
sns.heatmap(df1.corr(),annot=True,cmap='YlGnBu')
```

```
plt.show()
```


Airline 1 -0.059 0.015 -0.21 -0.32 0.028 -0.037 -0.065 0.059 0.033 0.086 -0.25 0.081
Source -0.059 1 -0.59 0.19 0.024 0.0066 0.062 0.058 -0.056 0.026 0.021 0.17 -0.15
Destination 0.015 -0.59 1 -0.3 -0.084 -0.05 -0.1 -0.072 0.13 -0.04 0.017 -0.26 0.061
Total_Stops -0.21 0.19 -0.3 1 0.62 -0.024 0.066 -0.062 -0.0026 0.039 -0.11 0.74 -0.14
Price -0.32 0.024 -0.084 0.62 1 -0.17 0.019 0.011 -0.028 0.028 -0.091 0.53 -0.13
Journey_day 0.028 0.0066 -0.05 -0.024 -0.17 1 -0.27 -0.0016 -0.016 -0.0029 -0.021 -0.039 -0.0076
Journey_month -0.037 0.062 -0.1 0.066 0.019 -0.27 1 0.027 0.0023 -0.0042 -0.026 0.055 -0.019
Dep_hour -0.065 0.058 -0.072 -0.062 0.011 -0.0016 0.027 1 -0.024 0.0061 0.068 0.0027 -0.024
Dep_minute 0.059 -0.056 0.13 -0.0026 -0.028 -0.016 0.0023 -0.024 1 0.043 -0.017 -0.022 0.093
Arrival_hour 0.033 0.026 -0.04 0.039 0.028 -0.0029 -0.0042 0.0061 0.043 1 -0.15 0.055 -0.12
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

In [56]:
```
from sklearn.ensemble import ExtraTreesRegressor
selection=ExtraTreesRegressor()
selection.fit(x,y)
```

Out[56]:
```
▾ ExtraTreesRegressor
ExtraTreesRegressor()
```

In [57]:
```
selection.feature_importances_
```

Out[57]:
```
array([0.15105599, 0.02115595, 0.04331621, 0.30705876, 0.11400644,
       0.08867386, 0.02724947, 0.02552556, 0.03223592, 0.02369507,
       0.1459313 , 0.02009546])
```

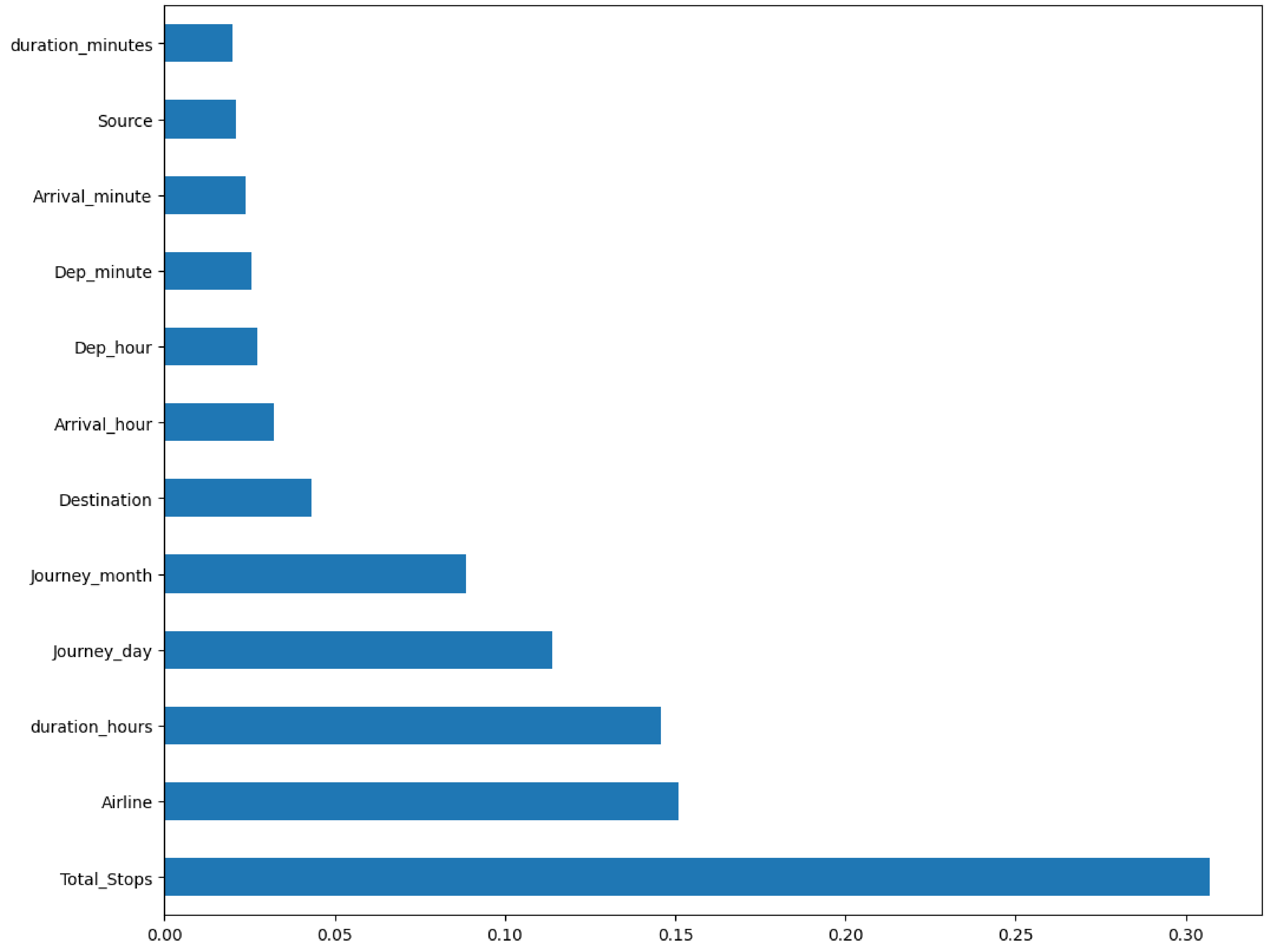
In [58]:
```
feat_importances=pd.Series(selection.feature_importances_,index=x.columns)
feat_importances
```

Out[58]:

```
Airline             0.151056
Source              0.021156
Destination         0.043316
Total_Stops         0.307059
Journey_day         0.114006
Journey_month       0.088674
Dep_hour            0.027249
Dep_minute          0.025526
Arrival_hour        0.032236
Arrival_minute      0.023695
duration_hours      0.145931
duration_minutes    0.020095
dtype: float64
```

In [59]:

```
plt.figure(figsize=(12,10))
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
```


duration_minutes
Source
Arrival_minute
Dep_minute
Dep_hour
Arrival_hour
Destination
Journey_month
Journey_day
duration_hours
Airline
Total_Stops
0.00
0.05
0.10
0.15
0.20
0.25
0.30

In [60]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=100)
```

In [61]:
```python
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
```

In [62]:
```python
lr = LinearRegression()
xgb = XGBRegressor()
rfr = RandomForestRegressor()
dtr = DecisionTreeRegressor()
```

In [63]:
```python
print(lr.fit(x_train , y_train))
print(xgb.fit(x_train , y_train))
print(rfr.fit(x_train , y_train))
print(dtr.fit(x_train , y_train))
```

```
LinearRegression()
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...)
RandomForestRegressor()
DecisionTreeRegressor()
```

In [64]:
```python
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
print(r2_score(lr.predict(x_train) , y_train))
print(r2_score(xgb.predict(x_train) , y_train))
print(r2_score(rfr.predict(x_train) , y_train))
print(r2_score(dtr.predict(x_train) , y_train))
```

```
-0.0723870398904598
0.928049819876814
0.9483308487742239
0.9665767652632145
```

In [65]:
```
y_pred1=lr.predict(x_test)
print(y_pred1)
y_pred2=xgb.predict(x_test)
print(y_pred2)
y_pred3=rfr.predict(x_test)
print(y_pred3)
y_pred4=dtr.predict(x_test)
print(y_pred4)
```

```
[11291.68467341 10421.56000472  8171.03267612 ...  4392.54031734
  7185.81522614 11258.62861641]
[ 9098.864   12736.763    9307.795   ...  2257.6172   3795.127   11242.669 ]
[ 9801.20533333 12490.04333333  9666.06666667 ...  2510.44333333
  3610.05        10474.315     ]
[12681. 12121.  9646. ...  2700.  3597.  6224.]
```

In [66]:
```
print(lr.score(x_train,y_train))
print(lr.score(x_test,y_test))
```

```
0.4825353472838043
0.42954092941134925
```

In [67]:
```
print(xgb.score(x_train,y_train))
print(xgb.score(x_test,y_test))
```

```
0.9353095153307481
0.7954975112007732
```

In [68]:
```
print(rfr.score(x_train,y_train))
print(rfr.score(x_test,y_test))
```

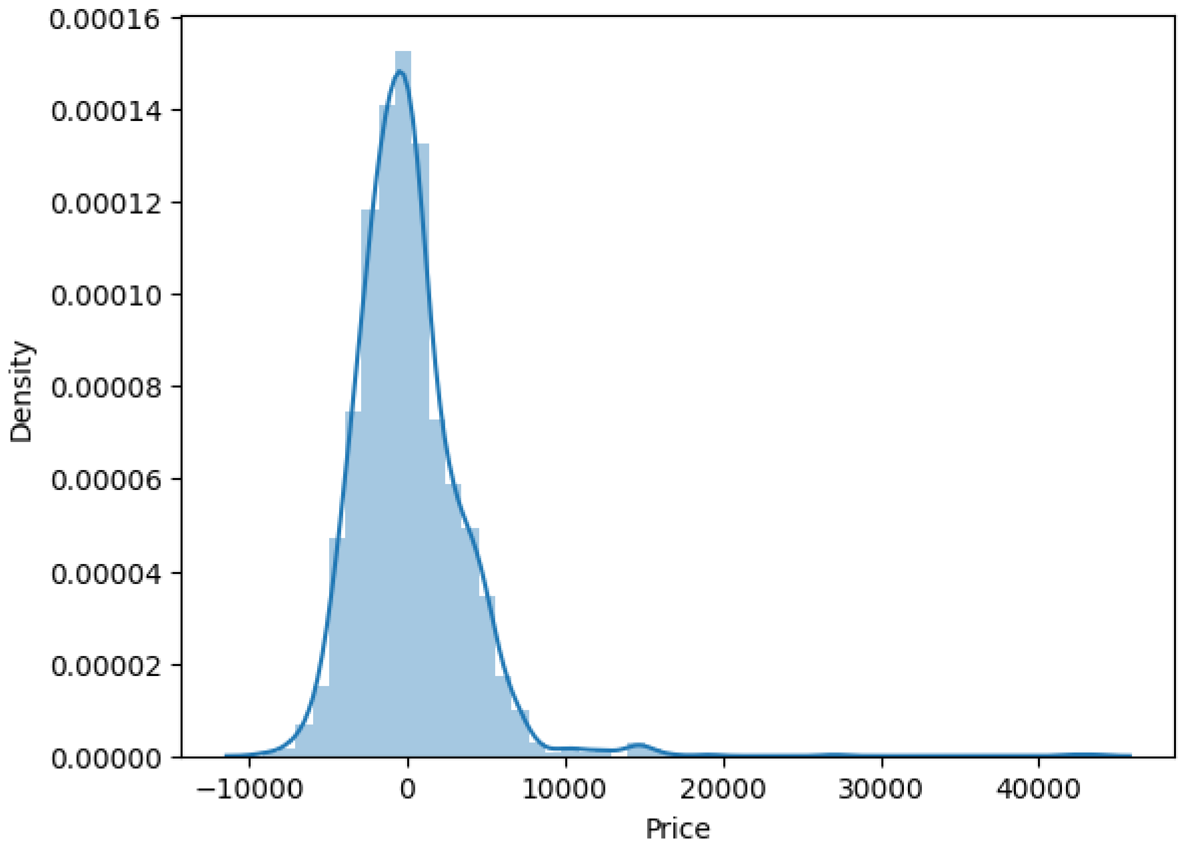
```
0.9532034093624842
0.765507987602049
```

In [69]:
```
print(dtr.score(x_train,y_train))
print(dtr.score(x_test,y_test))
```

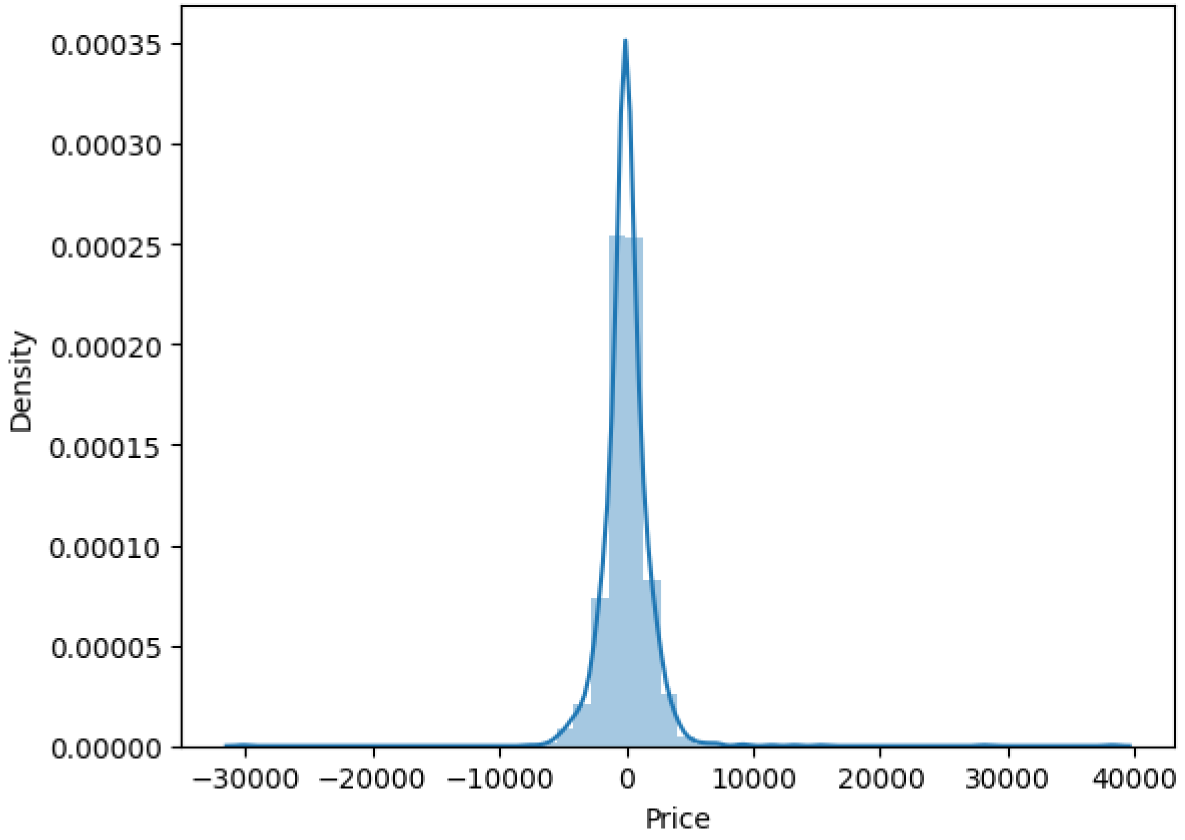
```
0.9676577479455467
0.6290696151127271
```

In [70]:
```
sns.distplot(y_test-y_pred1)
plt.show()
sns.distplot(y_test-y_pred2)
plt.show()
sns.distplot(y_test-y_pred3)
plt.show()
sns.distplot(y_test-y_pred4)
plt.show()
```

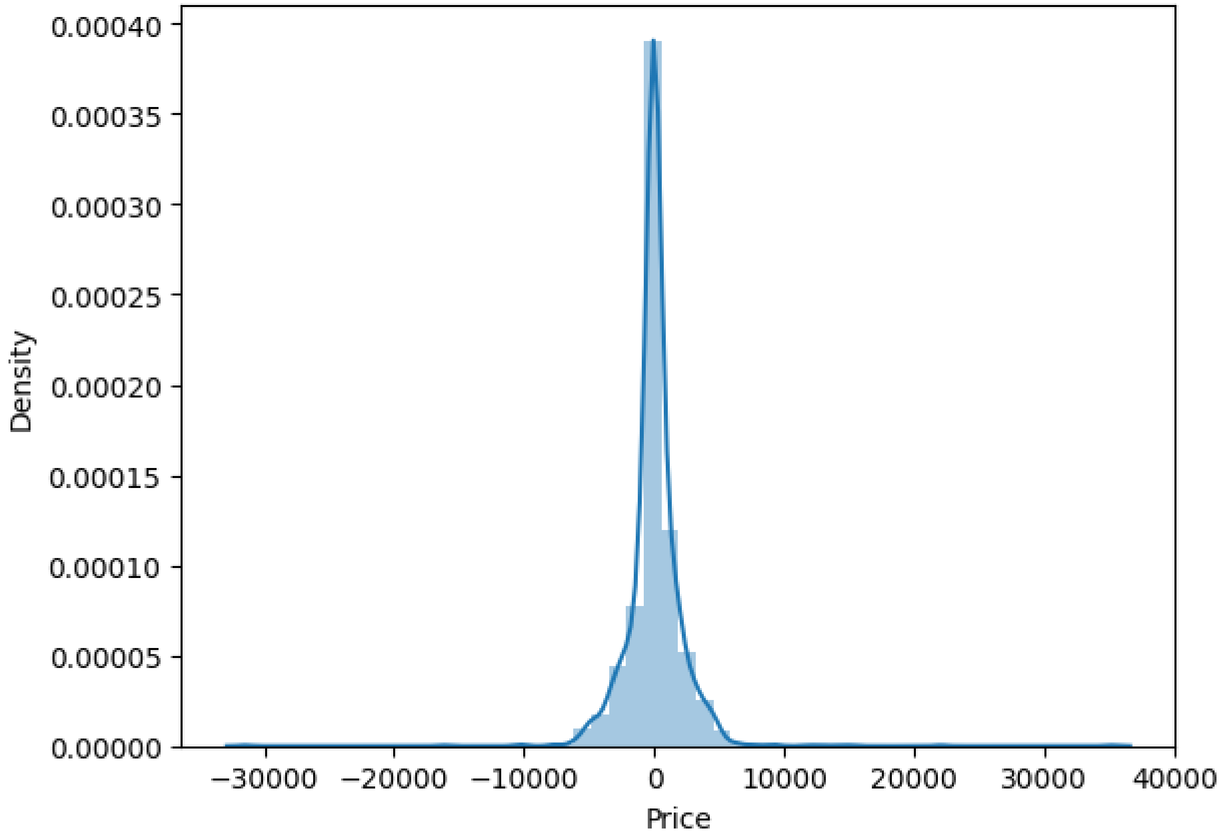
```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\901131556.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred1)
```



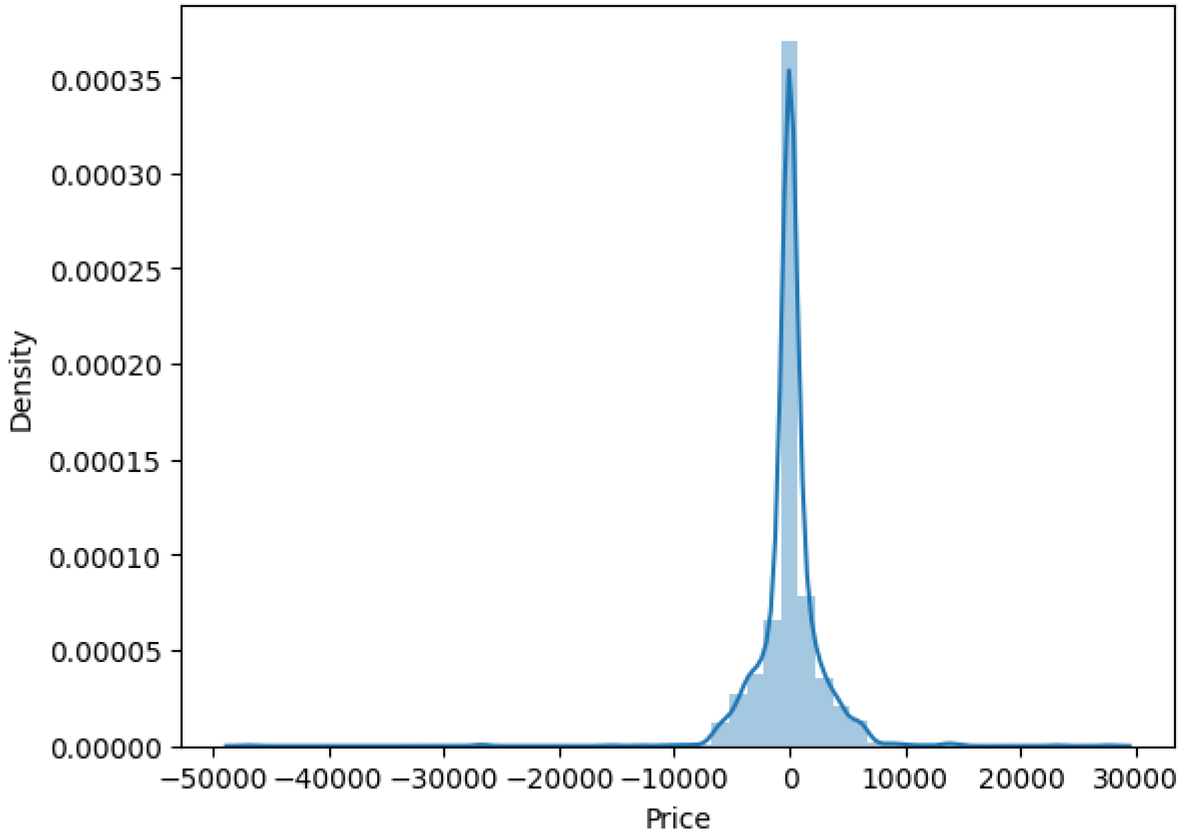
```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\901131556.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred2)
```



```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\901131556.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred3)
```



```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\901131556.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred4)
```



In [71]:
```
from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(y_test,y_pred1))
print('MSE:',metrics.mean_squared_error(y_test,y_pred1))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred1)))
print('MAE:',metrics.mean_absolute_error(y_test,y_pred2))
print('MSE:',metrics.mean_squared_error(y_test,y_pred2))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred2)))
print('MAE:',metrics.mean_absolute_error(y_test,y_pred3))
print('MSE:',metrics.mean_squared_error(y_test,y_pred3))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred3)))
print('MAE:',metrics.mean_absolute_error(y_test,y_pred4))
print('MSE:',metrics.mean_squared_error(y_test,y_pred4))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred4)))
```

```
MAE: 2397.0583733791823
MSE: 11983046.823721472
RMSE: 3461.6537700529025
MAE: 1178.2504311967746
MSE: 4295773.395837148
RMSE: 2072.624760017391
MAE: 1224.002935579534
MSE: 4925732.465703084
RMSE: 2219.39912266881
MAE: 1435.2289477798622
MSE: 7791752.99265309
RMSE: 2791.371167124338
```

In [72]:
```
from sklearn.model_selection import RandomizedSearchCV
n_estimators=[int(x) for x in np.linspace(start=100,stop=1200,num=12)]
max_features=['auto','sqrt']
max_depth=[int(x) for x in np.linspace(start=5,stop=30,num=6)]
min_samples_split=[2,5,10,15,100]
min_samples_leaf=[1,2,5,10]

random_grid={'n_estimators':n_estimators,'max_features':max_features,'max_depth':max_depth,'min_samples_split':min_samples_split
```

In [73]:
```
rscv=RandomizedSearchCV(estimator=rfr,param_distributions=random_grid,scoring='neg_mean_squared_error',n_iter=10,cv=5,random_sta
```

In [74]:
```
rscv.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   4.2s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   4.1s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   3.7s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   4.8s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   6.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   6.0s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   5.8s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   7.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   6.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   5.6s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   2.5s
```

```
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   3.9s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   2.4s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   2.8s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   2.4s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   7.9s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   4.9s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   6.4s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   5.2s
```

```
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   6.4s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   9.1s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   9.3s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   7.7s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   7.5s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   9.1s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=  12.5s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=  10.8s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=  12.1s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=  12.4s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=  13.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   3.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.5s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   3.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   1.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   1.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   2.8s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   1.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   1.4s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   3.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.6s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=  10.7s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   9.5s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=  10.7s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=  10.6s
```

```
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=  12.4s
C:\Users\HP\miniconda3\lib\site-packages\sklearn\ensemble\_forest.py:413: FutureWarning: `max_features='auto'` has been deprecat
ed in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as
it is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

Out[74]:

```
▸        RandomizedSearchCV
▸ estimator: RandomForestRegressor
      ▸ RandomForestRegressor
```

In [75]:

```python
rscv.best_params_
```

Out[75]:

```
{'n_estimators': 700,
 'min_samples_split': 15,
 'min_samples_leaf': 1,
 'max_features': 'auto',
 'max_depth': 20}
```

In [76]:

```python
y_pred5=rscv.predict(x_test)
y_pred5
```

Out[76]:

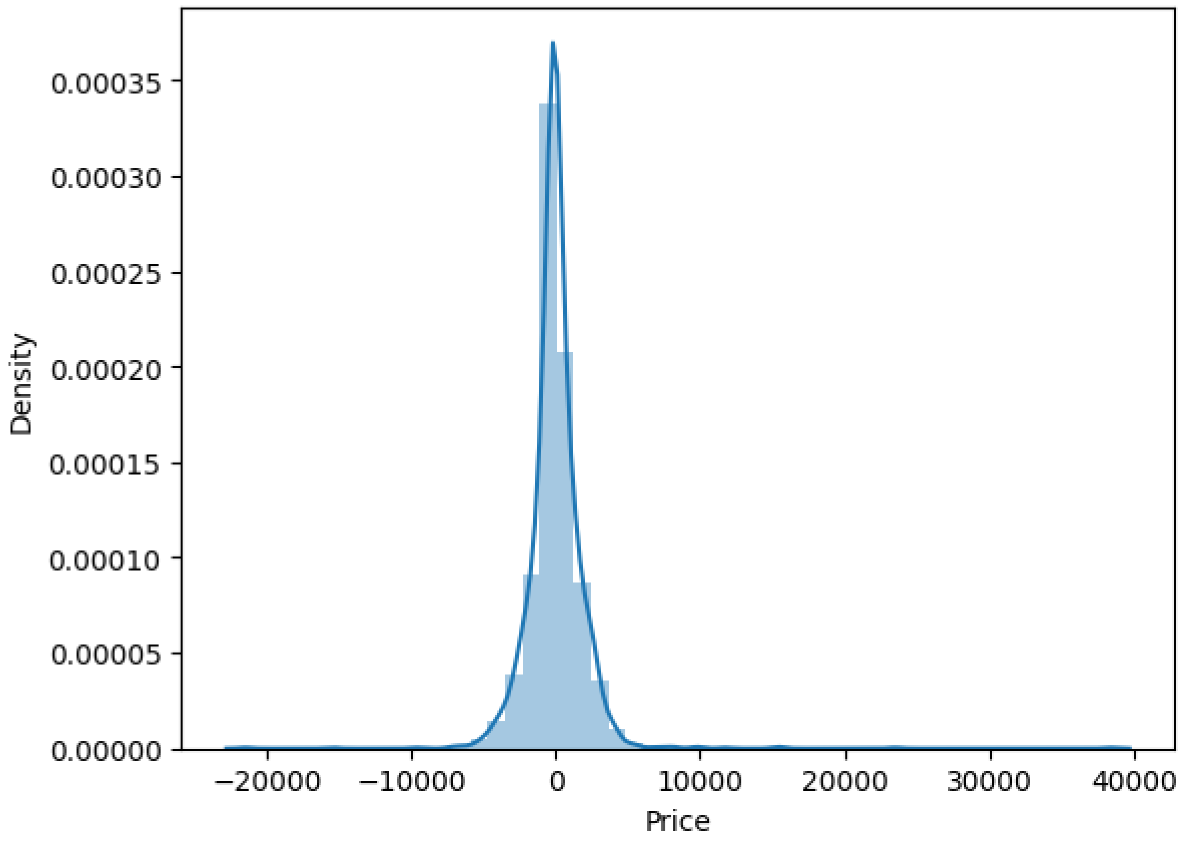
```
array([ 9835.54802006, 13252.37389602,  9714.37428375, ...,
        2376.62928788,  3658.35685666, 10480.51202761])
```

In [77]:

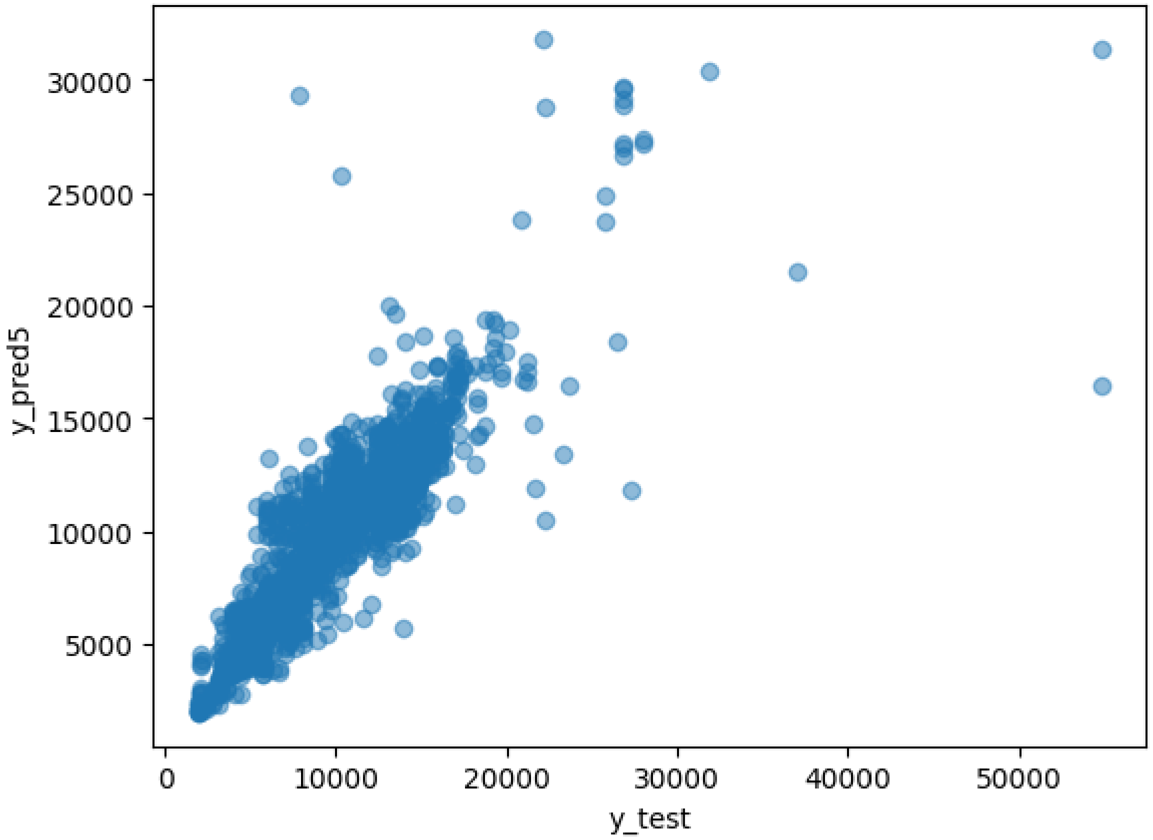
```python
sns.distplot(y_test-y_pred5)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11460\2870952028.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred5)
```

Out[77]:
```
<AxesSubplot: xlabel='Price', ylabel='Density'>
```



In [78]:
```
plt.scatter(y_test,y_pred5,alpha=0.5)
plt.xlabel('y_test')
plt.ylabel('y_pred5')
plt.show()
```



In [79]:
```
print('MAE:',metrics.mean_absolute_error(y_test,y_pred5))
print('MSE:',metrics.mean_squared_error(y_test,y_pred5))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_pred5)))
```

```
MAE: 1169.1174930982406
MSE: 4176492.3430799
RMSE: 2043.6468244488576
```

In [80]:
```
metrics.r2_score(y_test,y_pred5)
```

Out[80]:
```
0.8011759467018374
```

In [81]:
```
metrics.r2_score(y_test,y_pred3)
```

Out[81]:
```
0.765507987602049
```

In [82]:
```
import pickle
file = open(r'flight_price_pred.pkl', "wb")
pickle.dump(rscv,file)
```

In [83]:
```
model = open(r'flight_price_pred.pkl', "rb")
forest = pickle.load(model)
```

In [84]:
```
z = forest.predict(df2.iloc[1:2 , :])
for i in z:
    print(i)
```

```
4308.255675141243
```

In [85]:
```
df1.Source.unique()
```

Out[85]:
```
array([0, 3, 2, 1, 4])
```

In [ ]: