

Prediction of Airline Ticket Fare Using ML

```
In [107...]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [108...]: train_data=pd.read_excel(r'D:\Data Science Project1\Flight_Price_DataSet/Data_Train.xlsx')
```

```
In [109...]: train_data.head(3)
```

```
Out[109]:
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	nonstop
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2

```
In [110...]: train_data.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 [111...]: train_data.dropna(inplace=True)
```

```
In [112...]: train_data.isnull().sum()
```

```
Out[112]: 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 [113... #Doing copy bcz we want to do manipulation in the data and also keep the origina data
```

```
In [114... data=train_data.copy()
```

```
In [115... def convert_into_datetime(col):
           data[col]= pd.to_datetime(data[col])
```

```
In [116... data.columns
```

```
Out[116]: Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
                  'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
                  'Additional_Info', 'Price'],
                  dtype='object')
```

```
In [117... for items in ['Date_of_Journey', 'Dep_Time', 'Arrival_Time']:
           convert_into_datetime(items)
```

```
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '24/03/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '24/06/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '27/05/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '18/04/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '24/04/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '15/04/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '21/03/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '15/05/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '18/06/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.  
  
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:  
  Parsing '15/06/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
```

```
C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '18/05/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '27/06/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '21/05/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '15/03/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '24/05/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '21/04/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '21/06/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '27/03/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '18/03/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.

C:\Users\hp\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py:1047: UserWarning:
  Parsing '27/04/2019' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
```

In [118]: `data.dtypes`

```
Out[118]: Airline          object
Date_of_Journey  datetime64[ns]
Source           object
Destination      object
Route            object
Dep_Time         datetime64[ns]
Arrival_Time     datetime64[ns]
Duration          object
Total_Stops      object
Additional_Info   object
Price            int64
dtype: object
```

In [119]: `data['Date_of_Journey'].max()`

```
Out[119]: Timestamp('2019-12-06 00:00:00')
```

In [120]: `data['Journey_Day']=data['Date_of_Journey'].dt.day`
`data['Journey_Month']=data['Date_of_Journey'].dt.month`
`data['Journey_Year']=data['Date_of_Journey'].dt.year`

In [121]: `data.drop('Date_of_Journey',axis=1,inplace=True)`

In [122]: *#Extraction of hour and minutes from departure time and arrival time.*

```
def extract_hour_min(df,col):
    df[col+'_hour']=df[col].dt.hour
    df[col+'_minute']=df[col].dt.minute
    df.drop(col,axis=1,inplace=True)
    return df.head(2)
```

In [123]: `extract_hour_min(data,'Dep_Time')`

```
Out[123]:
```

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	Banglore	New Delhi	BLR → DEL	2022-03-22 01:10:00	2h 50m	non-stop	No info	3897
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2022-10-17 13:15:00	7h 25m	2 stops	No info	7662

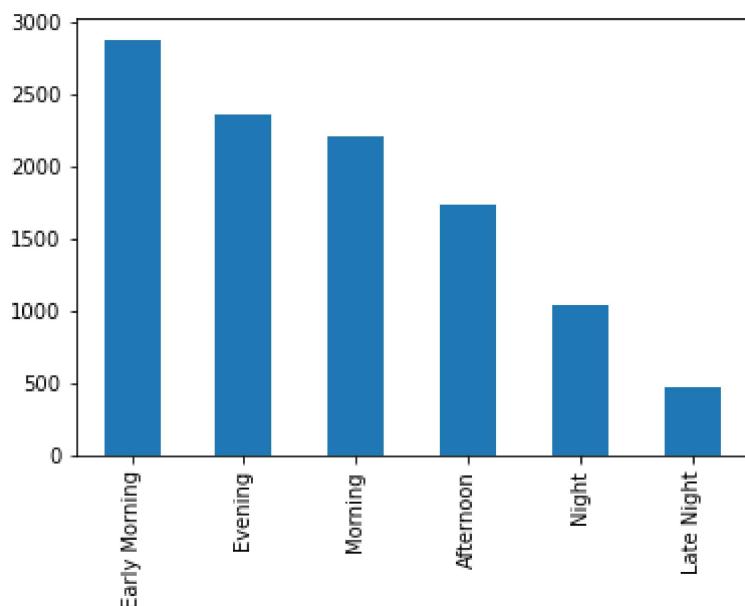
In [124]: `extract_hour_min(data,'Arrival_Time')`

Out[124]:	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_Day
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	5

```
In [125... def flight_departure_time(x):
    if (x>4) and (x<=8):
        return "Early Morning"
    elif(x>8) and (x<=12):
        return "Morning"
    elif(x>12) and (x<=16):
        return "Afternoon"
    elif(x>16) and (x<=20):
        return "Evening"
    elif(x>20) and (x<=24):
        return "Night"
    else:
        return "Late Night"
```

```
In [126... data['Dep_Time_hour'].apply(flight_departure_time).value_counts().plot(kind='bar')
```

Out[126]: <AxesSubplot:>



```
In [127... !pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\hp\anaconda3\lib\site-packages (5.6.0)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from plotly) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from plotly) (8.0.1)
```

In [128...]: `!pip install cufflinks`

```
Requirement already satisfied: cufflinks in c:\users\hp\anaconda3\lib\site-packages (0.17.3)
Requirement already satisfied: pandas>=0.19.2 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (1.4.2)
Requirement already satisfied: numpy>=1.9.2 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (1.21.5)
Requirement already satisfied: colorlover>=0.2.1 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (0.3.0)
Requirement already satisfied: ipython>=5.3.0 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (8.2.0)
Requirement already satisfied: plotly>=4.1.1 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (5.6.0)
Requirement already satisfied: ipywidgets>=7.0.0 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (7.6.5)
Requirement already satisfied: six>=1.9.0 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (1.16.0)
Requirement already satisfied: setuptools>=34.4.1 in c:\users\hp\anaconda3\lib\site-packages (from cufflinks) (61.2.0)
Requirement already satisfied: decorator in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (5.1.1)
Requirement already satisfied: pickleshare in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.7.5)
Requirement already satisfied: colorama in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.4.4)
Requirement already satisfied: stack-data in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (3.0.20)
Requirement already satisfied: backcall in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.2.0)
Requirement already satisfied: pygments>=2.4.0 in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (2.11.2)
Requirement already satisfied: matplotlib-inline in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.1.2)
Requirement already satisfied: traitlets>=5 in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\hp\anaconda3\lib\site-packages (from ipython>=5.3.0->cufflinks) (0.18.1)
Requirement already satisfied: ipython-genutils~0.2.0 in c:\users\hp\anaconda3\lib\site-packages (from ipywidgets>=7.0.0->cufflinks) (0.2.0)
Requirement already satisfied: ipykernel>=4.5.1 in c:\users\hp\anaconda3\lib\site-packages (from ipywidgets>=7.0.0->cufflinks) (6.9.1)
Requirement already satisfied: widgetsnbextension~3.5.0 in c:\users\hp\anaconda3\lib\site-packages (from ipywidgets>=7.0.0->cufflinks) (3.5.2)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in c:\users\hp\anaconda3\lib\site-packages (from ipywidgets>=7.0.0->cufflinks) (1.0.0)
Requirement already satisfied: nbformat>=4.2.0 in c:\users\hp\anaconda3\lib\site-packages (from ipywidgets>=7.0.0->cufflinks) (5.3.0)
Requirement already satisfied: tornado<7.0,>=4.2 in c:\users\hp\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (6.1)
Requirement already satisfied: jupyter-client<8.0 in c:\users\hp\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (6.1.12)
Requirement already satisfied: debugpy<2.0,>=1.0.0 in c:\users\hp\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (1.5.1)
Requirement already satisfied: nest-asyncio in c:\users\hp\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (1.5.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\hp\anaconda3\lib\site-packages (from jedi>=0.16->ipython>=5.3.0->cufflinks) (0.8.3)
Requirement already satisfied: jupyter-core>=4.6.0 in c:\users\hp\anaconda3\lib\site-packages (from jupyter-client<8.0->ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks)
```

(4.9.2)

Requirement already satisfied: python-dateutil>=2.1 in c:\users\hp\anaconda3\lib\site-packages (from jupyter-client<8.0->ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (2.8.2)

Requirement already satisfied: pyzmq>=13 in c:\users\hp\anaconda3\lib\site-packages (from jupyter-client<8.0->ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (22.3.0)

Requirement already satisfied: pywin32>=1.0 in c:\users\hp\anaconda3\lib\site-packages (from jupyter-core>=4.6.0->jupyter-client<8.0->ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (302)

Requirement already satisfied: fastjsonschema in c:\users\hp\anaconda3\lib\site-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (2.15.1)

Requirement already satisfied: jsonschema>=2.6 in c:\users\hp\anaconda3\lib\site-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (4.4.0)

Requirement already satisfied: attrs>=17.4.0 in c:\users\hp\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (21.4.0)

Requirement already satisfied: pyrsistent!=0.17.0,!0.17.1,!0.17.2,>=0.14.0 in c:\users\hp\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (0.18.0)

Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.19.2->cufflinks) (2021.3)

Requirement already satisfied: tenacity>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from plotly>=4.1.1->cufflinks) (8.0.1)

Requirement already satisfied: wcwidth in c:\users\hp\anaconda3\lib\site-packages (from prompt-toolkit!=3.0.0,!3.0.1,<3.1.0,>=2.0.0->ipython>=5.3.0->cufflinks) (0.2.5)

Requirement already satisfied: notebook>=4.4.1 in c:\users\hp\anaconda3\lib\site-packages (from widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (6.4.8)

Requirement already satisfied: terminado>=0.8.3 in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (0.13.1)

Requirement already satisfied: nbconvert in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (6.4.4)

Requirement already satisfied: Send2Trash>=1.8.0 in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (1.8.0)

Requirement already satisfied: argon2-cffi in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (21.3.0)

Requirement already satisfied: prometheus-client in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (0.13.1)

Requirement already satisfied: jinja2 in c:\users\hp\anaconda3\lib\site-packages (from notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (2.11.3)

Requirement already satisfied: pywinpty>=1.1.0 in c:\users\hp\anaconda3\lib\site-packages (from terminado>=0.8.3->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (2.0.2)

Requirement already satisfied: argon2-cffi-bindings in c:\users\hp\anaconda3\lib\site-packages (from argon2-cffi->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (21.2.0)

Requirement already satisfied: cffi>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (1.15.0)

Requirement already satisfied: pycparser in c:\users\hp\anaconda3\lib\site-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (2.21)

Requirement already satisfied: MarkupSafe>=0.23 in c:\users\hp\anaconda3\lib\site-packages (from jinja2->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.0->cufflinks) (2.0.1)

Requirement already satisfied: entrypoints>=0.2.2 in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~3.5.0->ipywidgets>=7.0.

```

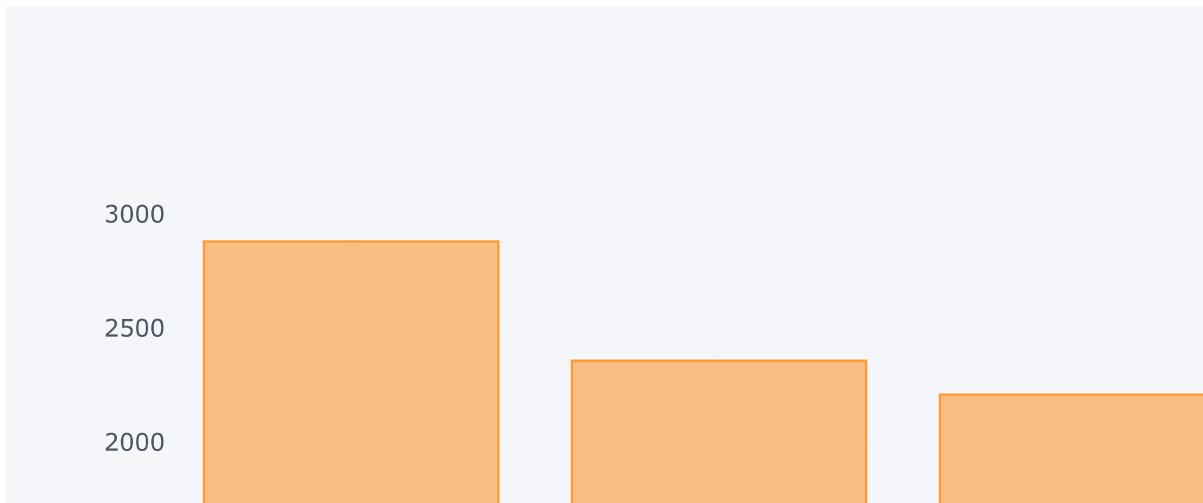
0->cufflinks) (0.4)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.5.13)
Requirement already satisfied: bleach in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (4.1.0)
Requirement already satisfied: beautifulsoup4 in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (4.11.1)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.8.4)
Requirement already satisfied: jupyterlab-pygments in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.1.2)
Requirement already satisfied: testpath in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.5.0)
Requirement already satisfied: defusedxml in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.7.1)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\hp\anaconda3\lib\site-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (1.5.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (2.3.1)
Requirement already satisfied: packaging in c:\users\hp\anaconda3\lib\site-packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (21.3)
Requirement already satisfied: webencodings in c:\users\hp\anaconda3\lib\site-packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.5.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\hp\anaconda3\lib\site-packages (from packaging->bleach->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (3.0.4)
Requirement already satisfied: pure-eval in c:\users\hp\anaconda3\lib\site-packages (from stack-data->ipython>=5.3.0->cufflinks) (0.2.2)
Requirement already satisfied: asttokens in c:\users\hp\anaconda3\lib\site-packages (from stack-data->ipython>=5.3.0->cufflinks) (2.0.5)
Requirement already satisfied: executing in c:\users\hp\anaconda3\lib\site-packages (from stack-data->ipython>=5.3.0->cufflinks) (0.8.3)

```

```
In [129...]: import plotly
          import cufflinks as cf
          from cufflinks.offline import go_offline
          from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

```
In [130...]: cf.go_offline()
```

```
In [131...]: data['Dep_Time_hour'].apply(flight_departure_time).value_counts().iplot(kind='bar')
```



```
In [132]: def preprocess_duration(x):
    if 'h' not in x:
        x='0h '+x
    elif 'm' not in x:
        x=x+' 0m'
    return x
```

```
In [133]: data['Duration']=data['Duration'].apply(preprocess_duration)
```

```
In [134]: data['Duration']
```

```
Out[134]: 0      2h 50m
1      7h 25m
2      19h 0m
3      5h 25m
4      4h 45m
      ...
10678    2h 30m
10679    2h 35m
10680    3h 0m
10681    2h 40m
10682    8h 20m
Name: Duration, Length: 10682, dtype: object
```

```
In [135]: int(data['Duration'][0].split(' ')[0][0:-1])
```

Out[135]: 2

In [136]:

```
int(data['Duration'][0].split(' ')[1][0:-1])
```

Out[136]: 50

In [137]:

```
data['Duration_Hours']=data['Duration'].apply(lambda x:int(x.split(' ')[0][0:-1]))
```

In [138]:

```
data['Duration_Minutes']=data['Duration'].apply(lambda x:int(x.split(' ')[1][0:-1]))
```

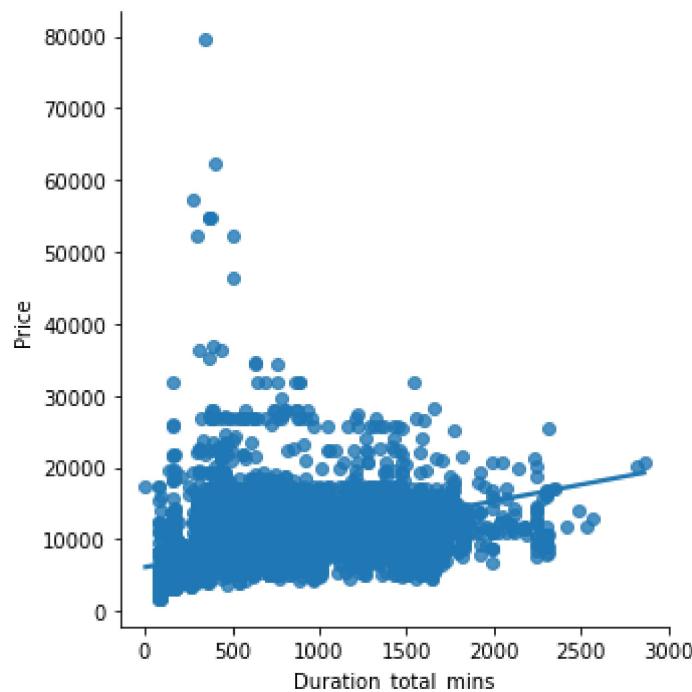
In [139]:

```
data['Duration_total_mins']=data['Duration'].str.replace('h','*60').str.replace(' ',',').str.replace('m','')
```

In [140]:

```
sns.lmplot(x='Duration_total_mins', y='Price', data=data)
```

Out[140]: <seaborn.axisgrid.FacetGrid at 0x2481ada55e0>

In [141]:

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

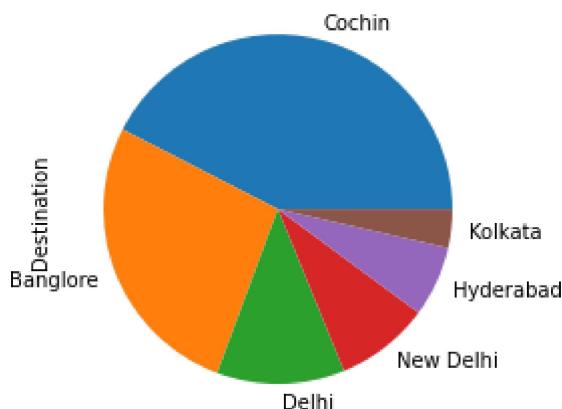
Out[141]:

Cochin	4536
Banglore	2871
Delhi	1265
New Delhi	932
Hyderabad	697
Kolkata	381

Name: Destination, dtype: int64In [142]:

```
data['Destination'].value_counts().plot(kind='pie')
```

Out[142]: <AxesSubplot:ylabel='Destination'>



```
In [143]: data['Route'].value_counts()
```

```
Out[143]:
```

DEL → BOM → COK	2376
BLR → DEL	1552
CCU → BOM → BLR	979
CCU → BLR	724
BOM → HYD	621
...	
CCU → VTZ → BLR	1
CCU → IXZ → MAA → BLR	1
BOM → COK → MAA → HYD	1
BOM → CCU → HYD	1
BOM → BBI → HYD	1

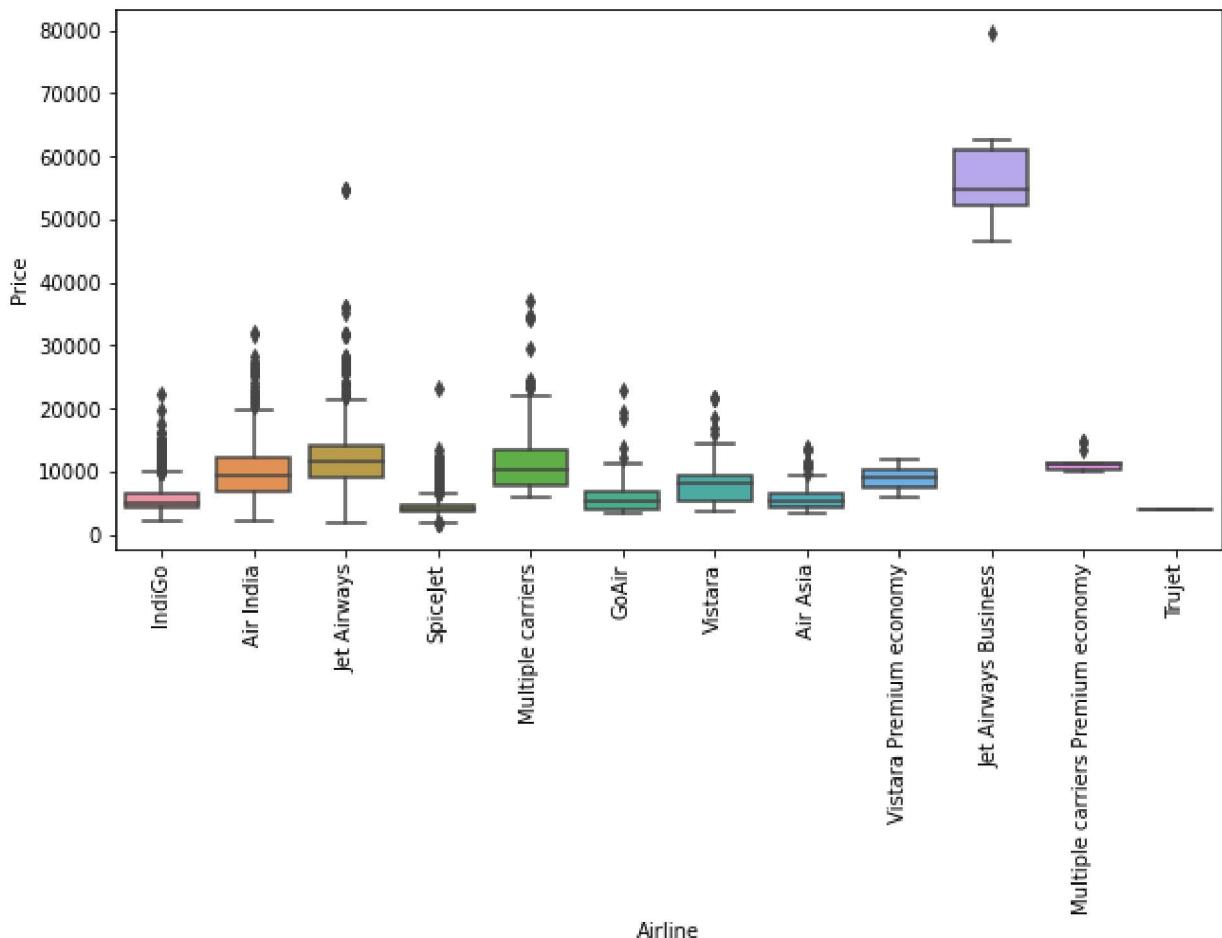
Name: Route, Length: 128, dtype: int64

```
In [144]: data[data['Airline']=='Jet Airways'].groupby('Route').size().sort_values(ascending=False)
```

```
Out[144]: Route
CCU → BOM → BLR          930
DEL → BOM → COK          875
BLR → BOM → DEL          385
BLR → DEL                382
CCU → DEL → BLR          300
BOM → HYD                207
DEL → JAI → BOM → COK   207
DEL → AMD → BOM → COK   141
DEL → IDR → BOM → COK   86
DEL → NAG → BOM → COK   61
DEL → ATQ → BOM → COK   38
DEL → COK                34
DEL → BHO → BOM → COK   29
DEL → BDQ → BOM → COK   28
DEL → LKO → BOM → COK   25
DEL → JDH → BOM → COK   23
CCU → GAU → BLR          22
DEL → MAA → BOM → COK   16
DEL → IXC → BOM → COK   13
BLR → MAA → DEL          10
BLR → BDQ → DEL          8
DEL → UDR → BOM → COK   7
BOM → DEL → HYD          5
CCU → BOM → PNQ → BLR   4
BLR → BOM → JDH → DEL   3
DEL → DED → BOM → COK   2
BOM → BDQ → DEL → HYD   2
DEL → CCU → BOM → COK   1
BOM → VNS → DEL → HYD   1
BOM → UDR → DEL → HYD   1
BOM → JDH → DEL → HYD   1
BOM → IDR → DEL → HYD   1
BOM → DED → DEL → HYD   1
dtype: int64
```

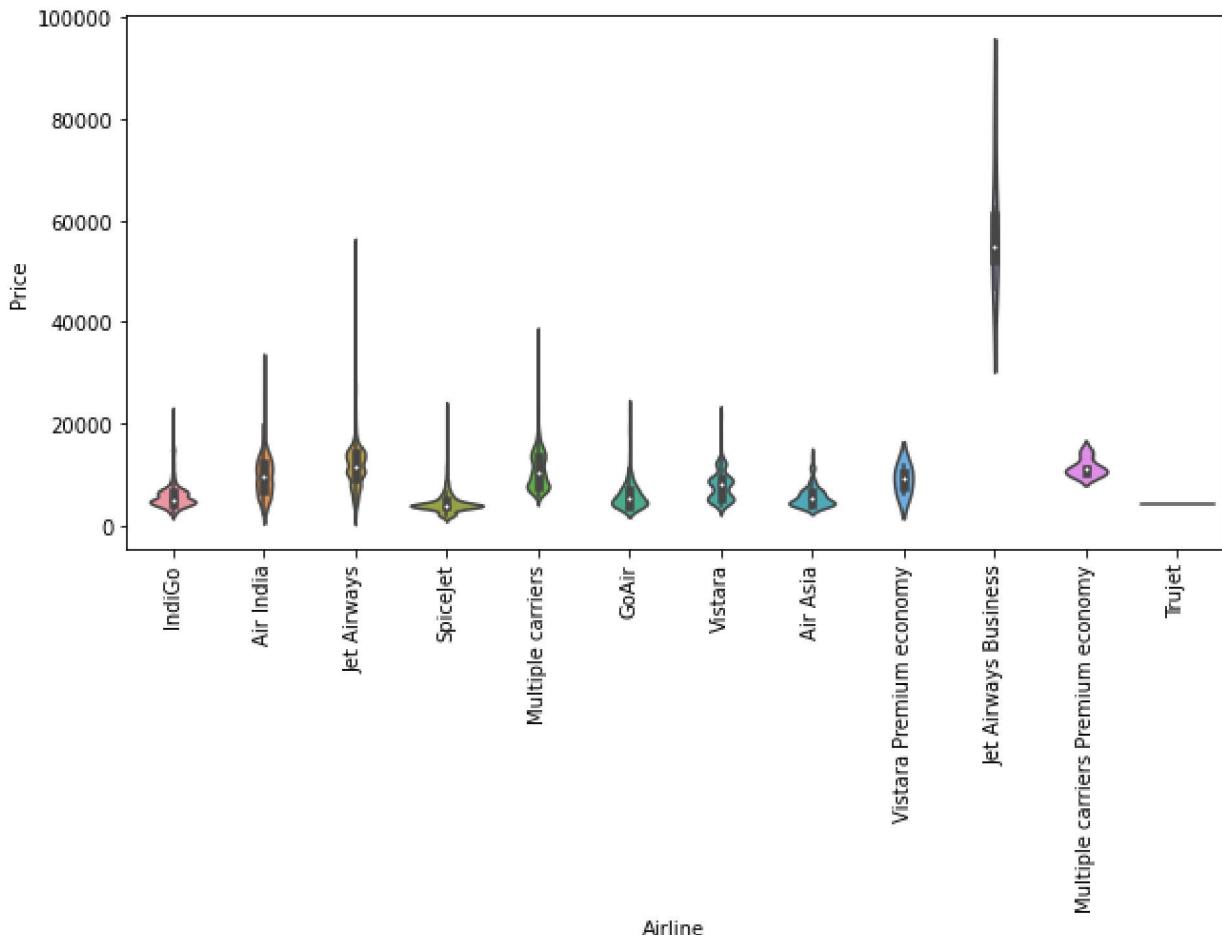
```
In [145]: plt.figure(figsize=(10,5))
sns.boxplot(x='Airline',y='Price',data=data)
plt.xticks(rotation ='vertical')
```

```
Out[145]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),  
 [Text(0, 0, 'IndiGo'),  
  Text(1, 0, 'Air India'),  
  Text(2, 0, 'Jet Airways'),  
  Text(3, 0, 'SpiceJet'),  
  Text(4, 0, 'Multiple carriers'),  
  Text(5, 0, 'GoAir'),  
  Text(6, 0, 'Vistara'),  
  Text(7, 0, 'Air Asia'),  
  Text(8, 0, 'Vistara Premium economy'),  
  Text(9, 0, 'Jet Airways Business'),  
  Text(10, 0, 'Multiple carriers Premium economy'),  
  Text(11, 0, 'Trujet')])
```



```
In [146]: plt.figure(figsize=(10,5))
sns.violinplot(x='Airline',y='Price',data=data)
plt.xticks(rotation = 'vertical')
```

```
Out[146]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),  
 [Text(0, 0, 'IndiGo'),  
  Text(1, 0, 'Air India'),  
  Text(2, 0, 'Jet Airways'),  
  Text(3, 0, 'SpiceJet'),  
  Text(4, 0, 'Multiple carriers'),  
  Text(5, 0, 'GoAir'),  
  Text(6, 0, 'Vistara'),  
  Text(7, 0, 'Air Asia'),  
  Text(8, 0, 'Vistara Premium economy'),  
  Text(9, 0, 'Jet Airways Business'),  
  Text(10, 0, 'Multiple carriers Premium economy'),  
  Text(11, 0, 'Trujet')])
```



In [147]: `data.head(5)`

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

```
In [148...]: data.drop(columns=['Additional_Info', 'Route', 'Duration_total_mins', 'Journey_Year'], axis=1)
```

```
In [149...]: data.head(5)
```

Out[149]:

	Airline	Source	Destination	Duration	Total_Stops	Price	Journey_Day	Journey_Month	Dep_1
0	IndiGo	Banglore	New Delhi	2h 50m	non-stop	3897	24		3
1	Air India	Kolkata	Banglore	7h 25m	2 stops	7662	5		1
2	Jet Airways	Delhi	Cochin	19h 0m	2 stops	13882	6		9
3	IndiGo	Kolkata	Banglore	5h 25m	1 stop	6218	5		12
4	IndiGo	Banglore	New Delhi	4h 45m	1 stop	13302	3		1

```
In [150...]: categorical_col=[col for col in data.columns if data[col].dtype=='object']
```

```
numerical_col=[col for col in data.columns if data[col].dtype!='object']
```

```
In [151...]: categorical_col
```

Out[151]:

```
['Airline', 'Source', 'Destination', 'Duration', 'Total_Stops']
```

```
In [152...]: numerical_col
```

Out[152]:

```
['Price',
 'Journey_Day',
 'Journey_Month',
 'Dep_Time_hour',
 'Dep_Time_minute',
 'Arrival_Time_hour',
 'Arrival_Time_minute',
 'Duration_Hours',
 'Duration_Minutes']
```

```
In [153...]: data['Source'].unique()
```

Out[153]:

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

```
In [154...]: # building up of hot-one encoding from scratch using simple for loop :
data['Source'].apply(lambda x:1 if x=='Banglore' else 0)
```

Out[154]:

```
0      1
1      0
2      0
3      0
4      1
      ..
10678    0
10679    0
10680    1
10681    1
10682    0
Name: Source, Length: 10682, dtype: int64
```

```
In [155... for item in data['Source'].unique():
    data['Source'+ item]=data['Source'].apply(lambda x:1 if x==item else 0)

##But above method is infeasible when the data is too big say 10000 cities. so next fe
```

```
In [156... #airlines= data.groupby(data['Airline'])['Price'].mean().sort_values()
```

```
In [157... airlines= data.groupby(['Airline'])['Price'].mean().sort_values().index
```

```
In [158... airlines
```

```
Out[158]: Index(['Trujet', 'SpiceJet', 'Air Asia', 'IndiGo', 'GoAir', 'Vistara',
       'Vistara Premium economy', 'Air India', 'Multiple carriers',
       'Multiple carriers Premium economy', 'Jet Airways',
       'Jet Airways Business'],
      dtype='object', name='Airline')
```

```
In [159... #Converting this index in dictionary
```

```
In [ ]:
```

```
In [160... dict1={key:index for index,key in enumerate (airlines,0)}
```

```
In [161... data['Airline'].map(dict1)
```

```
Out[161]: 0      3
1      7
2     10
3      3
4      3
 ..
10678    2
10679    7
10680    10
10681    5
10682    7
Name: Airline, Length: 10682, dtype: int64
```

```
In [162... data['Airline']=data['Airline'].map(dict1)
data['Airline']
```

```
Out[162]: 0      3
1      7
2     10
3      3
4      3
 ..
10678    2
10679    7
10680    10
10681    5
10682    7
Name: Airline, Length: 10682, dtype: int64
```

```
In [163... data['Destination'].unique()
```

```
Out[163]: array(['New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi', 'Hyderabad'],
      dtype=object)
```

```
In [164]: data['Destination'].replace('New Delhi', 'Delhi', inplace=True)
```

```
In [165]: data['Destination'].unique()
```

```
Out[165]: array(['Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Hyderabad'],
   dtype=object)
```

```
In [166]: Destins=data.groupby(['Destination'])['Price'].mean().sort_values().index
Destins
```

```
Out[166]: Index(['Kolkata', 'Hyderabad', 'Delhi', 'Banglore', 'Cochin'], dtype='object', name='Destination')
```

```
In [167]: dict2={key:index for index,key in enumerate (Destins,0)}
```

```
In [168]: dict2
```

```
Out[168]: {'Kolkata': 0, 'Hyderabad': 1, 'Delhi': 2, 'Banglore': 3, 'Cochin': 4}
```

```
In [169]: data['Destination']=data['Destination'].map(dict2)
data['Destination']
```

```
Out[169]: 0      2
1      3
2      4
3      3
4      2
..
10678  3
10679  3
10680  2
10681  2
10682  4
Name: Destination, Length: 10682, dtype: int64
```

```
In [170]: data.head(2)
```

	Airline	Source	Destination	Duration	Total_Stops	Price	Journey_Day	Journey_Month	Dep_Time
0	3	Banglore		2	2h 50m	non-stop	3897	24	3
1	7	Kolkata		3	7h 25m	2 stops	7662	5	1

```
In [171]: data['Total_Stops'].unique()
```

```
Out[171]: array(['non-stop', '2 stops', '1 stop', '3 stops', '4 stops'],
   dtype=object)
```

```
In [172]: stops={'non-stop':0, '2 stops':2, '1 stop':1, '3 stops':3, '4 stops':4}
```

```
In [173]: data['Total_Stops']=data['Total_Stops'].map(stops)
```

```
In [ ]:
```

```
In [ ]:
```

Outlier detection and how to deal with it?

methods available : 1.distribution plot 2.histogram 3.Box plot- 5 point summary data.

In [174...]

```
def plot(df,col):
    fig,(ax1,ax2,ax3)=plt.subplots(3,1)
    sns.distplot(df[col],ax=ax1) #distribution plot
    sns.boxplot(df[col],ax=ax2) #Boxplot
    sns.distplot(df[col],ax=ax3,kde=False) #histogram when kde=False
```

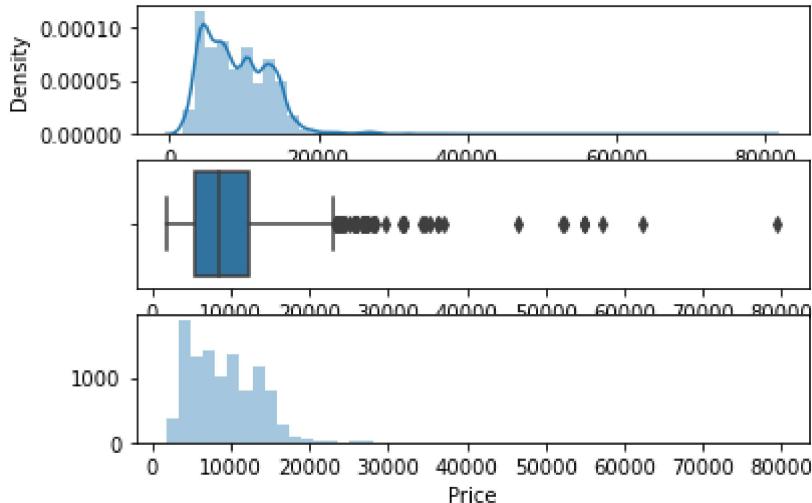
In [175...]

```
plot(data,'Price')
```

C:\Users\hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

C:\Users\hp\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

C:\Users\hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).



In [176...]

```
data['Price']=np.where(data['Price']>=35000,data['Price'].median(),data['Price'])
```

In [177...]

```
data['Price']
```

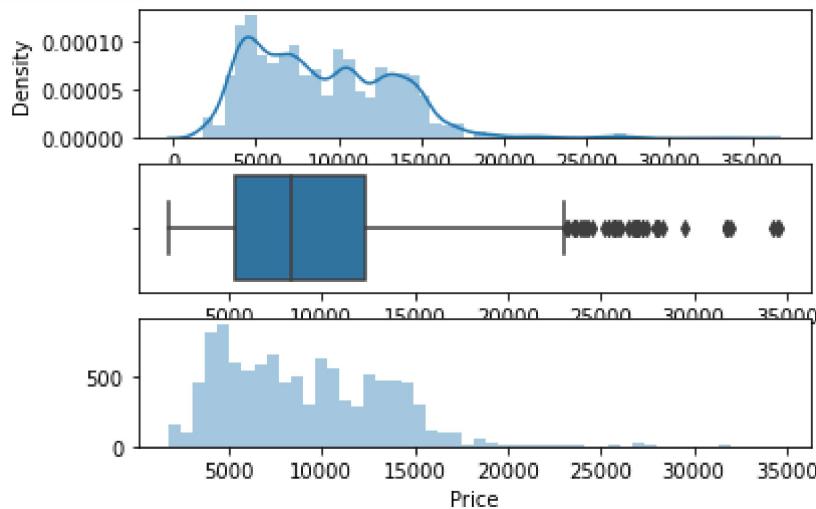
```
Out[177]: 0      3897.0
1      7662.0
2      13882.0
3      6218.0
4      13302.0
...
10678    4107.0
10679    4145.0
10680    7229.0
10681    12648.0
10682    11753.0
Name: Price, Length: 10682, dtype: float64
```

```
In [178]: plot(data, 'Price')
```

C:\Users\hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

C:\Users\hp\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

C:\Users\hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).



```
In [179]: data.drop(columns=['Source', 'Duration'], axis=1, inplace=True)
```

```
In [180]: data.dtypes
```

```
Out[180]: Airline          int64
Destination      int64
Total_Stops      int64
Price            float64
Journey_Day      int64
Journey_Month    int64
Dep_Time_hour    int64
Dep_Time_minute  int64
Arrival_Time_hour int64
Arrival_Time_minute int64
Duration_Hours   int64
Duration_Minutes int64
SourceBanglore   int64
SourceKolkata    int64
SourceDelhi      int64
SourceChennai    int64
SourceMumbai     int64
dtype: object
```

```
In [181... data.dropna(inplace=True)
```

```
In [182... data.isnull().sum()
```

```
Out[182]: Airline          0
Destination      0
Total_Stops      0
Price            0
Journey_Day      0
Journey_Month    0
Dep_Time_hour    0
Dep_Time_minute  0
Arrival_Time_hour 0
Arrival_Time_minute 0
Duration_Hours   0
Duration_Minutes 0
SourceBanglore   0
SourceKolkata    0
SourceDelhi      0
SourceChennai    0
SourceMumbai     0
dtype: int64
```

```
In [ ]:
```

```
In [ ]:
```

Feature selection= features which are most contributing to objective decided (Here obj is to predict fare prices)

1.f reg (correlation) 2.mutual info reg (chi square, p value ...many other techniques)

```
In [183... from sklearn.feature_selection import mutual_info_regression
```

```
In [184... #Price is dependant feature=y here in usecase.
#rest are indepe. = X
X=data.drop(['Price'],axis=1)
X
```

Out[184]:

	Airline	Destination	Total_Stops	Journey_Day	Journey_Month	Dep_Time_hour	Dep_Time_mi
0	3	2	0	24	3	22	
1	7	3	2	5	1	5	
2	10	4	2	6	9	9	
3	3	3	1	5	12	18	
4	3	2	1	3	1	16	
...
10678	2	3	0	4	9	19	
10679	7	3	0	27	4	20	
10680	10	2	0	27	4	8	
10681	5	2	0	3	1	11	
10682	7	4	2	5	9	10	

10682 rows × 16 columns

In [185... y=data['Price']

In [186... y

Out[186]:

0	3897.0
1	7662.0
2	13882.0
3	6218.0
4	13302.0
	...
10678	4107.0
10679	4145.0
10680	7229.0
10681	12648.0
10682	11753.0

Name: Price, Length: 10682, dtype: float64

In [187... mutual_info_regression(X,y)

Out[187]:

```
array([0.98113398, 1.00246478, 0.78089043, 0.1971285 , 0.249526 ,
       0.33343897, 0.25077892, 0.40713936, 0.35394209, 0.47700832,
       0.33783217, 0.39007161, 0.46630269, 0.52136228, 0.13101803,
       0.20399479])
```

In [188... imp=pd.DataFrame(mutual_info_regression(X,y),index=X.columns)
imp

Out[188]:

0

Airline	0.974760
Destination	0.995886
Total_Stops	0.786838
Journey_Day	0.193423
Journey_Month	0.239751
Dep_Time_hour	0.334378
Dep_Time_minute	0.255079
Arrival_Time_hour	0.401861
Arrival_Time_minute	0.348443
Duration_Hours	0.463997
Duration_Minutes	0.344684
SourceBanglore	0.399759
SourceKolkata	0.451232
SourceDelhi	0.518184
SourceChennai	0.133943
SourceMumbai	0.189771

In [189...]

```
imp.columns=[ 'Importance' ]  
imp
```

Out[189]:

	Importance
Airline	0.974760
Destination	0.995886
Total_Stops	0.786838
Journey_Day	0.193423
Journey_Month	0.239751
Dep_Time_hour	0.334378
Dep_Time_minute	0.255079
Arrival_Time_hour	0.401861
Arrival_Time_minute	0.348443
Duration_Hours	0.463997
Duration_Minutes	0.344684
SourceBanglore	0.399759
SourceKolkata	0.451232
SourceDelhi	0.518184
SourceChennai	0.133943
SourceMumbai	0.189771

In [190...]

```
imp.sort_values(by='Importance', ascending=False)
```

Out[190]:

	Importance
Destination	0.995886
Airline	0.974760
Total_Stops	0.786838
SourceDelhi	0.518184
Duration_Hours	0.463997
SourceKolkata	0.451232
Arrival_Time_hour	0.401861
SourceBanglore	0.399759
Arrival_Time_minute	0.348443
Duration_Minutes	0.344684
Dep_Time_hour	0.334378
Dep_Time_minute	0.255079
Journey_Month	0.239751
Journey_Day	0.193423
SourceMumbai	0.189771
SourceChennai	0.133943

Baisc ML model

split vailable data to create basic ML training data(75%of data) and testing data(25% ofthe data)

In [191]: `from sklearn.model_selection import train_test_split`In [192]: `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)`In [193]: `from sklearn.ensemble import RandomForestRegressor`In [194]: `ml_model=RandomForestRegressor()`In [195]: `model=ml_model.fit(X_train,y_train)`In [196]: `y_pred=model.predict(X_test)`In [197]: `y_pred`Out[197]: `array([16693.49, 6163.12, 8814.67, ..., 3488.45, 6472.86, 6907.29])`In [198]: `y_pred.shape`Out[198]: `(2671,)`

```
In [199]: #!pip install pickle
In [200]: import pickle
In [201]: file=open(r'D:\Data Science Project1\Flight_Price_DataSet/rf_random.pkl','wb')
In [202]: pickle.dump(model,file) # this is dumpling coefficient of ML L=model not entire model
In [203]: model=open(r'D:\Data Science Project1\Flight_Price_DataSet/rf_random.pkl','rb')
In [204]: forest=pickle.load(model)
In [205]: forest.predict(X_test)
Out[205]: array([16693.49, 6163.12, 8814.67, ..., 3488.45, 6472.86, 6907.29])

In [ ]:
In [206]: def mape(y_true,y_pred):
    y_true,y_pred=np.array(y_true),np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100
In [ ]:
In [207]: mape(y_test,forest.predict(X_test))
Out[207]: 13.22963261419405

In [208]: def predict(ml_model):
    model=ml_model.fit(X_train,y_train)
    print('Training_score: {}'.format(model.score(X_train,y_train)))
    y_prediction=model.predict(X_test)
    print('Predictions are : {}'.format(y_prediction))
    print('\n')

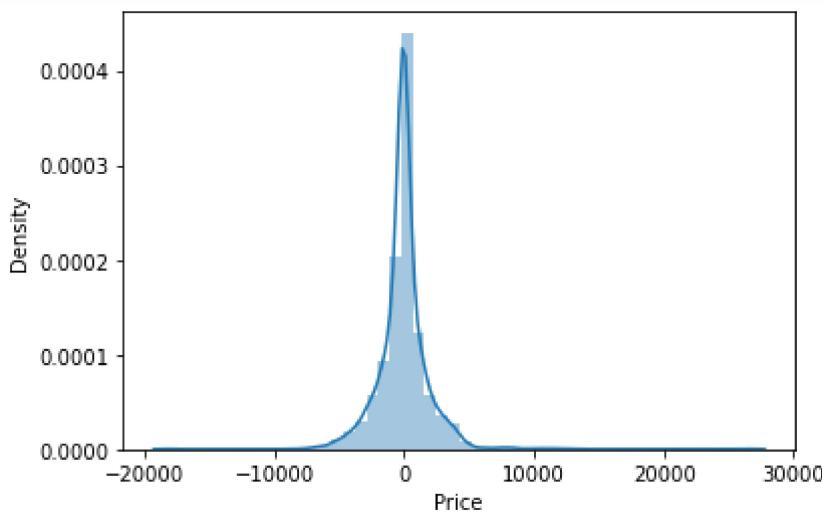
    from sklearn import metrics
    r2_score=metrics.r2_score(y_test,y_prediction)
    print('r2_score: {}'.format(r2_score))
    print('MSE : ', metrics.mean_squared_error(y_test,y_prediction))
    print('MAE : ', metrics.mean_absolute_error(y_test,y_prediction))
    print('RMSE : ', np.sqrt(metrics.mean_squared_error(y_test,y_prediction)))
    print('MAPE : ', mape(y_test,y_prediction))
    sns.distplot(y_test-y_prediction)
In [209]: predict(RandomForestRegressor())

```

```
Training_score: 0.9514108251964829
Predictions are : [16762.51  6351.74  8823.78 ...  3501.62  6382.89  6874.47]
```

```
r2_score: 0.8056865426151383
MSE : 3782816.597974108
MAE : 1182.2291459283156
RMSE : 1944.9464254765755
MAPE : 13.216556887366856
```

```
C:\Users\hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```



```
In [210]: from sklearn.model_selection import RandomizedSearchCV
```

```
In [211]: reg_rf=RandomForestRegressor()
```

```
In [212]: np.linspace(start=1000,stop=1200,num=6)
```

```
Out[212]: array([1000., 1040., 1080., 1120., 1160., 1200.])
```

```
# Number of trees in random forest
n_estimators=[int(x) for x in np.linspace(start=1000,stop=1200,num=6)]

# Number of features to consider at every split
max_features=["auto", "sqrt"]

# Maximum number of levels in tree
max_depth=[int(x) for x in np.linspace(start=5,stop=30,num=4)]

# Minimum number of samples required to split a node
min_samples_split=[5,10,15,100]
```

```
# Create the grid or hyper-parameter space
random_grid={

    'n_estimators':n_estimators,
    'max_features':max_features,
    'max_depth':max_depth,
```

```
'min_samples_split':min_samples_split  
}
```

In [218]: random_grid

```
Out[218]: {'n_estimators': [1000, 1040, 1080, 1120, 1160, 1200],  
           'max_features': ['auto', 'sqrt'],  
           'max_depth': [5, 13, 21, 30],  
           'min_samples_split': [5, 10, 15, 100]}
```

In [219]: rf_Random=RandomizedSearchCV(reg_rf,param_distributions=random_grid, cv=3, verbose=2, n_

In [220]: rf_Random.fit(X_train,y_train)

```
Fitting 3 folds for each of 10 candidates, totalling 30 fits  
Out[220]: RandomizedSearchCV(cv=3, estimator=RandomForestRegressor(), n_jobs=-1,  
                           param_distributions={'max_depth': [5, 13, 21, 30],  
                           'max_features': ['auto', 'sqrt'],  
                           'min_samples_split': [5, 10, 15, 100],  
                           'n_estimators': [1000, 1040, 1080, 1120,  
                           1160, 1200]},  
                           verbose=2)
```

In [221]: rf_Random.best_params_

```
Out[221]: {'n_estimators': 1160,  
           'min_samples_split': 5,  
           'max_features': 'auto',  
           'max_depth': 13}
```

In [222]: pred2=rf_Random.predict(X_test)

In [223]: `from sklearn import metrics
metrics.r2_score(y_test,pred2)`

```
Out[223]: 0.8291923992549644
```

In []: