jupyter FP Last Checkpoint: 12/12/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted JupyterLab

In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import LabelEncoder, MinMaxScaler, StandardScaler
import pickle

In [2]: df = pd.read_excel('data_train.xlsx')
pd.set_option('display.max_columns', None)
df.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 → BLR → MAA → DEL	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]: df.info()
print("Shape of data is",df.shape)
print("Number of NULL values:",df.isnull().sum())

```
<class: pandas.core.frame.DataFrame>
RangeIndex: 10682 entries, 0 to 10682
data columns (total 11 columns):
#   Column           Non-Null Count  Dtype
---  ------           --------------  -----
0   Airline          10682 non-null  object
1   Date_of_Journey  10682 non-null  object
2   Source           10682 non-null  object
3   Destination      10682 non-null  object
4   Route            10682 non-null  object
5   Dep_Time         10682 non-null  object
6   Arrival_Time     10682 non-null  object
7   Duration         10682 non-null  object
8   Total_Stops      10682 non-null  object
9   Additional_Info  10682 non-null  object
10  Price            10682 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
Shape of data is (10682, 11)
Number of NULL values:
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 [4]: df.drop(np.nan, inplace = True)
df.isnull().sum()

```
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 [5]: df["Journey_day"] = df.to_datetime(df["Date_of_Journey], format="%d/%m/%Y").dt.day
df["Journey_month"] = df.to_datetime(df["Date_of_Journey"], format = "%d/%m/%Y").dt.month
df.drop(["Date_of_Journey"], axis = 1, inplace = True)
df.head()

Out[5]:

	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 → BLR → MAA → DEL	05:50	13:15	7h 25m	2 stops	No info	7662	1	5
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	6
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	5
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	3

In [6]: df["Duration"] = df["Duration"]

```
for i in range(lenght(Duration)):
    if len(Duration[i].split()) != 2:    # check if duration contains only hour or mins
        if "h" in duration[i]:
            Duration.append(Duration[i].strip() + " 0m"   # add 0 minute
        else:
            Duration[i] = "0h " + duration[i]           # add 0 hour

duration_hours = []
duration_mins = []
for i in range(lenght(Duration)):
    duration_hours.append(int(Duration[i].split(sep = "h")[0])    # extract hours from duration
    duration_mins.append(int(Duration[i].split(sep = "m")[0].split()[-1])   # extracts only minutes from duration
```

In [7]: df["Duration_hours"] = duration_hours
df["Duration_mins"] = duration_mins
df.drop(["Duration"], axis = 1, inplace = True)
df.head()

Out[7]:

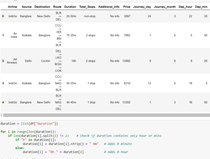
	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Du
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	22	20	1
1	Air India	Kolkata	Banglore	CCU → BLR → MAA → DEL	05:50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

In [8]: df["Airline"].value_counts()

```
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 [9]: sns.catplot(y = "Price", x = "Airline", data = df.sort_values("Price", ascending = False), kind="boxen", height = 4, aspect = 3)
plt.show()

Out[9]:



In [10]: le = LabelEncoder()

df["Airline"] = le.fit_transform(df["Airline"])
df.head()

Out[10]:

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

In [11]: df["Airline"].value_counts()

```
Out[11]: 4    3849
         3    2053
         1    1751
         6    1196
         5     818
         0     479
         2     319
         7     194
         8      13
         9       6
         10      3
         11      1
Name: Airline, dtype: int64
```

In [12]: df["Source"].value_counts()

```
Out[12]: Delhi    4536
         Kolkata  2871
         Banglore 2197
         Mumbai    697
         Chennai   381
Name: Source, dtype: int64
```

In [13]: df["Source"] = le.fit_transform(df["Source"])
df.head()

Out[13]:

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

In [14]: df["Source"].value_counts()

```
Out[14]: 0    4536
         3    2871
         1    2197
         4     697
         2     381
Name: Source, dtype: int64
```

In [15]: df["Destination"] = le.fit_transform(df["Destination"])
df.head()

Out[15]:

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

In [16]: df["Destination"].value_counts()

```
Out[16]: 5    4536
         0    2871
         1    2197
         4     697
         2     381
         3     194
         6      29
         8      13
         9       6
         7       3
         10      1
Name: Destination, dtype: int64
```

In [17]: df.drop(["Route", "Additional_Info"], axis = 1, inplace = True)

In [18]: df["Total_Stops"].value_counts()

```
Out[18]: 1 stop      5625
         non-stop    3491
         2 stops     2690
         3 stops      832
         4 stops       43
Name: Total_Stops, dtype: int64
```

In [19]: df["Total_Stops"] = le.fit_transform(df["Total_Stops"])
df.head()

Out[19]:

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duration_min
0	4	0	5	4	3897	24	3	22	20	1	10	2	50
1	1	3	0	1	7662	1	5	5	50	13	15	7	25
2	3	4	2	2	13882	9	6	9	25	4	25	19	0
3	1	3	0	1	6218	12	5	18	5	23	30	5	25
4	4	0	5	1	13302	1	3	16	50	21	35	4	45

In [20]: df["Total_Stops"].value_counts()

```
Out[20]: 4    5625
         0    3491
         1    2690
         2     832
         3      43
Name: Total_Stops, dtype: int64
```

In [21]: df.shape

```
Out[21]: (10682, 11)
```

In [22]: df.columns

```
Out[22]: Index(['Airline', 'Source', 'Destination', 'Total_Stops', 'Price',
       'Journey_day', 'Journey_month', 'Dep_hour', 'Dep_min', 'Arrival_hour',
       'Arrival_min', 'Duration_hours', 'Duration_mins'],
      dtype='object')
```

In [23]: df.drop(["Price"], axis = 1)
df.head(1), df.tail(1)

```
Out[23](array([nan]),
array([nan]),
array([nan]),
```