ML 17 - EDA And Feature Engineering Flight Price Prediction By Virat Tiwari

December 12, 2023

1 ML 17 - EDA And Feature Engineering Flight Price Prediction By Virat Tiwari

[135]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[136]:
```
df=pd.read_excel("Data_Train.xlsx")
df.head()
```

[136]:

	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

[137]:
```
df.tail()
```

[137]:

	Airline	Date_of_Journey	Source	Destination
10678	Air Asia	9/04/2019	Kolkata	Banglore
10679	Air India	27/04/2019	Kolkata	Banglore
10680	Jet Airways	27/04/2019	Banglore	Delhi
10681	Vistara	01/03/2019	Banglore	New Delhi
10682	Air India	9/05/2019	Delhi	Cochin

```
                        Route Dep_Time Arrival_Time Duration Total_Stops  \
10678              CCU → BLR    19:55        22:25   2h 30m    non-stop
10679              CCU → BLR    20:45        23:20   2h 35m    non-stop
10680              BLR → DEL    08:20        11:20       3h    non-stop
10681              BLR → DEL    11:30        14:10   2h 40m    non-stop
10682  DEL → GOI → BOM → COK    10:55        19:15   8h 20m     2 stops

      Additional_Info  Price
10678         No info   4107
10679         No info   4145
10680         No info   7229
10681         No info  12648
10682         No info  11753
```

[138]:
```
df.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
```

[139]:
```
df.describe()
```

[139]:
```
              Price
count  10683.000000
mean    9087.064121
std     4611.359167
min     1759.000000
25%     5277.000000
50%     8372.000000
75%    12373.000000
max    79512.000000
```

[140]:
```
df.head(2)
```

[140]:
```
      Airline Date_of_Journey    Source Destination                  Route  \
0      IndiGo      24/03/2019  Banglore   New Delhi              BLR → DEL
1   Air India       1/05/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR

  Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price
0    22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897
1    05:50         13:15   7h 25m     2 stops         No info   7662
```

[141]:
```
df["Date"]=df["Date_of_Journey"].str.split("/").str[0]
df["Month"]=df["Date_of_Journey"].str.split("/").str[1]
df["Year"]=df["Date_of_Journey"].str.split("/").str[2]
```

[142]:
```
df.head(2)
```

[142]:
```
      Airline Date_of_Journey    Source Destination                  Route  \
0      IndiGo      24/03/2019  Banglore   New Delhi              BLR → DEL
1   Air India       1/05/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR

  Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price Date  \
0    22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897   24
1    05:50         13:15   7h 25m     2 stops         No info   7662    1

  Month  Year
0    03  2019
1    05  2019
```

[143]:
```
df["Date"]=df["Date"].astype(int)
df["Month"]=df["Month"].astype(int)
df["Year"]=df["Year"].astype(int)
```

[144]:
```
df.head(2)
```

[144]:
```
      Airline Date_of_Journey    Source Destination                  Route  \
0      IndiGo      24/03/2019  Banglore   New Delhi              BLR → DEL
1   Air India       1/05/2019   Kolkata    Banglore  CCU → IXR → BBI → BLR

  Dep_Time  Arrival_Time Duration Total_Stops Additional_Info  Price  Date  \
0    22:20  01:10 22 Mar   2h 50m    non-stop         No info   3897    24
1    05:50         13:15   7h 25m     2 stops         No info   7662     1

   Month  Year
0      3  2019
1      5  2019
```

[145]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 14 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 
 11  Date             10683 non-null  int64 
 12  Month            10683 non-null  int64 
 13  Year             10683 non-null  int64 
dtypes: int64(4), object(10)
memory usage: 1.1+ MB
```

[146]:
```
df.drop("Date_of_Journey",axis=1,inplace=True)
```

[147]:
```
df.head(2)
```

[147]:

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	2019
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2019

[148]:
```
df["Arrival_Time"].str.split(" ").str[0]
```

[148]:
```
0        01:10
1        13:15
2        04:25
3        23:30
4        21:35
         …
10678    22:25
10679    23:20
10680    11:20
10681    14:10
10682    19:15
```

```
Name: Arrival_Time, Length: 10683, dtype: object
```

[149]:
```
df["Arrival Hours"]=df["Arrival_Time"].str.split(" ").str[0].str.split(":").
  ↪str[0]
```

[150]:
```
df["Arrival_Minutes"]=df["Arrival_Time"].str.split(" ").str[0].str.split(":").
  ↪str[1]
```

[151]:
```
df.drop("Arrival_Time",axis=1,inplace=True)
```

[152]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Source           10683 non-null  object
 2   Destination      10683 non-null  object
 3   Route            10682 non-null  object
 4   Dep_Time         10683 non-null  object
 5   Duration         10683 non-null  object
 6   Total_Stops      10682 non-null  object
 7   Additional_Info  10683 non-null  object
 8   Price            10683 non-null  int64 
 9   Date             10683 non-null  int64 
 10  Month            10683 non-null  int64 
 11  Year             10683 non-null  int64 
 12  Arrival Hours    10683 non-null  object
 13  Arrival_Minutes  10683 non-null  object
dtypes: int64(4), object(10)
memory usage: 1.1+ MB
```

[153]:
```
df["Arrival_Hour"]=df["Arrival Hours"].astype(int)
df["Arrival_Minutes"]=df["Arrival_Minutes"].astype(int)
```

[154]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Source           10683 non-null  object
 2   Destination      10683 non-null  object
```

```
 3   Route            10682 non-null  object
 4   Dep_Time         10683 non-null  object
 5   Duration         10683 non-null  object
 6   Total_Stops      10682 non-null  object
 7   Additional_Info  10683 non-null  object
 8   Price            10683 non-null  int64
 9   Date             10683 non-null  int64
 10  Month            10683 non-null  int64
 11  Year             10683 non-null  int64
 12  Arrival Hours    10683 non-null  object
 13  Arrival_Minutes  10683 non-null  int64
 14  Arrival_Hour     10683 non-null  int64
dtypes: int64(6), object(9)
memory usage: 1.2+ MB
```

[155]:
```
df.head(2)
```

[155]:

	Airline	Source	Destination	Route	Dep_Time	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival Hours	Arrival_Minutes	Arrival_Hour
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	2h 50m	non-stop	No info	3897	24	3	2019	01	10	1
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	7h 25m	2 stops	No info	7662	1	5	2019	13	15	13

[156]:
```
df["Dep_Hour"]=df["Dep_Time"].str.split(":").str[0]
df["Dep_Min"]=df["Dep_Time"].str.split(":").str[1]
```

[157]:
```
df["Dep_Hour"]=df["Dep_Hour"].astype(int)
df["Dep_Min"]=df["Dep_Min"].astype(int)
```

[158]:
```
df.drop("Dep_Time",axis=1,inplace =True)
```

[159]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          10683 non-null  object
 1   Source           10683 non-null  object
 2   Destination      10683 non-null  object
 3   Route            10682 non-null  object
```

```
 4   Duration         10683 non-null  object
 5   Total_Stops      10682 non-null  object
 6   Additional_Info  10683 non-null  object
 7   Price            10683 non-null  int64
 8   Date             10683 non-null  int64
 9   Month            10683 non-null  int64
 10  Year             10683 non-null  int64
 11  Arrival Hours    10683 non-null  object
 12  Arrival_Minutes  10683 non-null  int64
 13  Arrival_Hour     10683 non-null  int64
 14  Dep_Hour         10683 non-null  int64
 15  Dep_Min          10683 non-null  int64
dtypes: int64(8), object(8)
memory usage: 1.3+ MB
```

[160]:
```
df.head(3)
```

[160]:

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival Hours	Arrival_Minutes	Arrival_Hour	Dep_Hour	Dep_Min
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	2019	01	10	1	22	20
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	1	5	2019	13	15	13	5	50
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2 stops	No info	13882	9	6	2019	04	25	4	9	25

[161]:
```
df.drop("Route",axis=1,inplace=True)
```

[162]:
```
df.head()
```

[162]:

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival Hours	Arrival_Minutes	Arrival_Hour	Dep_Hour	Dep_Min
0	IndiGo	Banglore	New Delhi	2h 50m	non-stop	No info	3897	24	3	2019	01	10	1	22	20
1	Air India	Kolkata	Banglore	7h 25m	2 stops	No info	7662	1	5	2019	13	15	13	5	50
2	Jet Airways	Delhi	Cochin	19h	2 stops	No info	13882	9	6	2019	04	25	4	9	25
3	IndiGo	Kolkata	Banglore	5h 25m	1 stop	No info									
4	IndiGo	Banglore	New Delhi	4h 45m	1 stop	No info									

```
3   6218     12       5  2019            23                30             23
4  13302      1       3  2019            21                35             21

   Dep_Hour  Dep_Min
0        22       20
1         5       50
2         9       25
3        18        5
4        16       50
```

[163]:
```
df["Duration_Hour"]=df["Duration"].str.split(" ").str[0].str.split("h").str[0]
```

[164]:
```
df["Duration_Min"]=df["Duration"].str.split(" ").str[1].str.split("h","m").
 ↪str[0]
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[164], line 1
----> 1␣
 ↪df["Duration_Min"]=df["Duration"].str.split(" ").str[1].str.split("h","m").st[0]

File /opt/conda/lib/python3.10/site-packages/pandas/core/generic.py:5902, in␣
 ↪NDFrame.__getattr__(self, name)
   5895 if (
   5896     name not in self._internal_names_set
   5897     and name not in self._metadata
   5898     and name not in self._accessors
   5899     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5900 ):
   5901     return self[name]
-> 5902 return object.__getattribute__(self, name)

File /opt/conda/lib/python3.10/site-packages/pandas/core/accessor.py:182, in␣
 ↪CachedAccessor.__get__(self, obj, cls)
    179 if obj is None:
    180     # we're accessing the attribute of the class, i.e., Dataset.geo
    181     return self._accessor
--> 182 accessor_obj = self._accessor(obj)
    183 # Replace the property with the accessor object. Inspired by:
    184 # https://www.pydanny.com/cached-property.html
    185 # We need to use object.__setattr__ because we overwrite __setattr__ on
    186 # NDFrame
    187 object.__setattr__(obj, self._name, accessor_obj)

File /opt/conda/lib/python3.10/site-packages/pandas/core/strings/accessor.py:
 ↪181, in StringMethods.__init__(self, data)
    178 def __init__(self, data) -> None:
```

```
    179     from pandas.core.arrays.string_ import StringDtype
--> 181     self._inferred_dtype = self._validate(data)
    182     self._is_categorical = is_categorical_dtype(data.dtype)
    183     self._is_string = isinstance(data.dtype, StringDtype)

File /opt/conda/lib/python3.10/site-packages/pandas/core/strings/accessor.py:
 ↪235, in StringMethods._validate(data)
    232 inferred_dtype = lib.infer_dtype(values, skipna=True)
    234 if inferred_dtype not in allowed_types:
--> 235     raise AttributeError("Can only use .str accessor with string values
 ↪")
    236 return inferred_dtype

AttributeError: Can only use .str accessor with string values!
```

[165]:
```
df.drop("Duration",axis=1,inplace=True)
```

[166]:
```
df.head(2)
```

[166]:
```
      Airline    Source Destination Total_Stops Additional_Info  Price  Date  \
0      IndiGo  Banglore   New Delhi    non-stop         No info   3897    24
1  Air India   Kolkata    Banglore     2 stops         No info   7662     1

   Month  Year Arrival Hours  Arrival_Minutes  Arrival_Hour  Dep_Hour  \
0      3  2019            01               10             1        22
1      5  2019            13               15            13         5

   Dep_Min Duration_Hour
0       20             2
1       50             7
```

[167]:
```
df["Total_Stops"].unique()
```

[167]:
```
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

[168]:
```
df["Total_Stops"].mode()
```

[168]:
```
0    1 stop
Name: Total_Stops, dtype: object
```

[175]:
```
df["Total_Stops"]=df["Total_Stops"].map({"non-stop":0,"1 stop":1,"2 stops":2,"3␣
 ↪stops":3,"4 stops":4,np.nan:1})
```

[178]:
```
df["Total_Stops"]
```

[178]:
```
0        1.0
1        1.0
2        1.0
3        NaN
4        NaN
        …
10678    1.0
10679    1.0
10680    1.0
10681    1.0
10682    1.0
Name: Total_Stops, Length: 10683, dtype: float64
```

[177]:
```
df.head()
```

[177]:

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	Arrival Hours	Arrival_Minutes	Arrival_Hour	Dep_Hour	Dep_Min	Duration_Hour
0	IndiGo	Banglore	New Delhi	1.0	No info	3897	24	3	2019	01	10	1	22	20	2
1	Air India	Kolkata	Banglore	1.0	No info	7662	1	5	2019	13	15	13	5	50	7
2	Jet Airways	Delhi	Cochin	1.0	No info	13882	9	6	2019	04	25	4	9	25	19
3	IndiGo	Kolkata	Banglore	NaN	No info	6218	12	5	2019	23	30	23	18	5	5
4	IndiGo	Banglore	New Delhi	NaN	No info	13302	1	3	2019	21	35	21	16	50	4

[179]:
```
df["Airline"].unique()
```

[179]:
```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

[180]:
```
df["Source"].unique()
```

[180]:
```
array(['Banglore', 'Kolkata', 'Delhi', 'Chennai', 'Mumbai'], dtype=object)
```

[181]:
```
df["Destination"].unique()
```

[181]:
```
array(['New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi', 'Hyderabad'],
      dtype=object)
```

[182]:
```
from sklearn.preprocessing import OneHotEncoder
```

[183]:
```
encoder=OneHotEncoder()
```

[186]:
```
encoder.fit_transform(df[["Airline","Source","Destination"]]).toarray()
```

[186]:
```
array([[0., 0., 0., …, 0., 0., 1.],
       [0., 1., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 1.],
       [0., 1., 0., …, 0., 0., 0.]])
```

[191]:
```
df1=pd.DataFrame(encoder.fit_transform(df[["Airline","Source","Destination"]]).
↪toarray(),columns=encoder.get_feature_names_out())
```

[194]:
```
Flight_price_prediction_cleaned=pd.concat([df,df1],axis=1)
```

[201]:
```
df.to_csv("Flight_price_prediction_cleaned.csv")
```

THANK YOU SO MUCH !!

YOURS VIRAT TIWARI :)