

EDA And Feature Engineering Flight Price Prediction

FEATURES The various features of the cleaned dataset are explained below:

1. Airline: The name of the airline company is stored in the airline column. It is a categorical feature having 6 different airlines.
2. Flight: Flight stores information regarding the plane's flight code. It is a categorical feature.
3. Source City: City from which the flight takes off. It is a categorical feature having 6 unique cities.
4. Departure Time: This is a derived categorical feature obtained created by grouping time periods into bins. It stores information about the departure time and have 6 unique time labels.
5. Stops: A categorical feature with 3 distinct values that stores the number of stops between the source and destination cities.
6. Arrival Time: This is a derived categorical feature created by grouping time intervals into bins. It has six distinct time labels and keeps information about the arrival time.
7. Destination City: City where the flight will land. It is a categorical feature having 6 unique cities.
8. Class: A categorical feature that contains information on seat class; it has two distinct values: Business and Economy.
9. Duration: A continuous feature that displays the overall amount of time it takes to travel between cities in hours.
10. Days Left: This is a derived characteristic that is calculated by subtracting the trip date by the booking date.
11. Price: Target variable stores information of the ticket price.

In []:

```
In [316... import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [317... df=pd.read_excel('flight_price.xlsx')
```

```
In [318... df.head(2)
```

Out[318]:

| | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Tota |
|---|--------------|-----------------|----------|-------------|-----------------------------------|----------|--------------|----------|------|
| 0 | IndiGo | 24/03/2019 | Banglore | New Delhi | BLR → DEL | 22:20 | 01:10 22 Mar | 2h 50m | nc |
| 1 | Air India | 1/05/2019 | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 13:15 | 7h 25m | |

In [319... `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
```

In [320... `df.tail(5)`

Out[320]:

| | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration |
|--------------|-------------|-----------------|----------|-------------|---|----------|--------------|----------|
| 10678 | Air Asia | 9/04/2019 | Kolkata | Banglore | CCU → BLR | 19:55 | 22:25 | 2h 30m |
| 10679 | Air India | 27/04/2019 | Kolkata | Banglore | CCU → BLR | 20:45 | 23:20 | 2h 35m |
| 10680 | Jet Airways | 27/04/2019 | Banglore | Delhi | BLR → DEL | 08:20 | 11:20 | 3h |
| 10681 | Vistara | 01/03/2019 | Banglore | New Delhi | BLR → DEL | 11:30 | 14:10 | 2h 40m |
| 10682 | Air India | 9/05/2019 | Delhi | Cochin | DEL → GOI → BOM → COK | 10:55 | 19:15 | 8h 20m |

In [321]:

df.describe()

Out[321]:

| | 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 |

In [322]:

```
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]
```

In [323]:

df.head(2)

Out[323]:

| | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info |
|---|-----------|-----------------|----------|-------------|-----------------------------------|----------|--------------|----------|-------------|-----------------|
| 0 | IndiGo | 24/03/2019 | Banglore | New Delhi | BLR → DEL | 22:20 | 01:10 22 Mar | 2h 50m | non-stop | No info |
| 1 | Air India | 1/05/2019 | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 13:15 | 7h 25m | 2 stops | No info |

In [324... `df.drop('Date_of_Journey',axis=1,inplace=True)`

In [325... `df.head(2)`

Out[325]:

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

In [326... `df['Arrival_hours']=df['Arrival_Time'].str.split(' ').str[0].str.split(':').str[0]`

In [327... `df['Arrival_min']=df['Arrival_Time'].str.split(' ').str[0].str.split(':').str[1]`

In [328... `df.drop('Arrival_Time',axis=1,inplace=True)`

In [329... `df.head(2)`

Out[329]:

| | Airline | Source | Destination | Route | Dep_Time | Duration | Total_Stops | Additional_Info | Price |
|---|-----------|----------|-------------|-----------------------------------|----------|----------|-------------|-----------------|-------|
| 0 | IndiGo | Banglore | New Delhi | BLR → DEL | 22:20 | 2h 50m | non-stop | No info | 3897 |
| 1 | Air India | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 7h 25m | 2 stops | No info | 7662 |

In [330... `#change dtype`
`df['Date']=df['Date'].astype(int)`
`df['Month']=df['Month'].astype(int)`
`df['Year']=df['Year'].astype(int)`

```
df['Arrival_hours']=df['Arrival_hours'].astype(int)
df['Arrival_min']=df['Arrival_min'].astype(int)
```

In [331...

```
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  int64
13  Arrival_min            10683 non-null  int64
dtypes: int64(6), object(8)
memory usage: 1.1+ MB
```

In [332...

```
df.head(2)
```

Out[332]:

| | Airline | Source | Destination | Route | Dep_Time | Duration | Total_Stops | Additional_Info | Price |
|---|-----------|----------|-------------|-----------------------------------|----------|----------|-------------|-----------------|-------|
| 0 | IndiGo | Banglore | New Delhi | BLR → DEL | 22:20 | 2h 50m | non-stop | No info | 3897 |
| 1 | Air India | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 7h 25m | 2 stops | No info | 7662 |

In [333...

```
df['Dep_hour']=df['Dep_Time'].str.split(':').str[0]
df['Dep_minute']=df['Dep_Time'].str.split(':').str[1]
df.drop('Dep_Time',axis=1,inplace=True)
```

In [334...

```
df.head(2)
```

Out[334]:

| | Airline | Source | Destination | Route | Duration | Total_Stops | Additional_Info | Price | Date | Mon |
|---|-----------|----------|-------------|-----------------------------------|----------|-------------|-----------------|-------|------|-----|
| 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 | 1 | |

In [335...

```
df['Dur_Hour']=df['Duration'].str.split(' ').str[0].str.split('h').str[0]
df['Dur_Minu']=df['Duration'].str.split(' ').str[1].str.split('m').str[0]

#df['Duration'].str.split(' ').str[0].str.split('h').str[0]
#df['Duration'].str.split(' ').str[1].str.split('h').str[0]
```

In [336...

```
df.head(2)
```

Out[336]:

| | Airline | Source | Destination | Route | Duration | Total_Stops | Additional_Info | Price | Date | Mon |
|---|-----------|----------|-------------|-----------------------------------|----------|-------------|-----------------|-------|------|-----|
| 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 | 1 | |

In [337...

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 17 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  int64
12  Arrival_min            10683 non-null  int64
13  Dep_hour               10683 non-null  object
14  Dep_minute             10683 non-null  object
15  Dur_Hour               10683 non-null  object
16  Dur_Minu               9651 non-null   object
dtypes: int64(6), object(11)
memory usage: 1.4+ MB

```

In [338...

```
df['Dur_Hour'].value_counts()
```

```
Out[338]: 2      2402
          1      621
          3      501
          7      487
          5      481
          9      445
          12     428
          8      424
          13     407
          11     365
          10     355
          6      340
          14     337
          15     268
          23     265
          26     241
          16     234
          4      222
          22     218
          24     197
          21     196
          25     186
          27     179
          20     162
          18     141
          19     134
          17     129
          28      94
          29      65
          30      49
          38      34
          37      17
          33      13
          32       9
          34       8
          35       7
          36       7
          31       6
          47       2
          42       2
          39       2
          5m       1
          41       1
          40       1
```

Name: Dur_Hour, dtype: int64

```
In [339... df['Dur_Hour'].str.isnumeric().sum()
```

```
Out[339]: 10682
```

```
In [340... df['Dur_Hour']=df['Dur_Hour'].str.split('m').str[0]
```

```
In [341... df['Dur_Hour'].value_counts()
```



```
Out[341]: 2      2402
          1      621
          3      501
          7      487
          5      482
          9      445
          12     428
          8      424
          13     407
          11     365
          10     355
          6      340
          14     337
          15     268
          23     265
          26     241
          16     234
          4      222
          22     218
          24     197
          21     196
          25     186
          27     179
          20     162
          18     141
          19     134
          17     129
          28      94
          29      65
          30      49
          38      34
          37      17
          33      13
          32       9
          34       8
          36       7
          35       7
          31       6
          47       2
          42       2
          39       2
          41       1
          40       1
          Name: Dur_Hour, dtype: int64
```

```
In [342... df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 17 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  int64
12   Arrival_min            10683 non-null  int64
13   Dep_hour               10683 non-null  object
14   Dep_minute             10683 non-null  object
15   Dur_Hour               10683 non-null  object
16   Dur_Minu               9651 non-null   object
dtypes: int64(6), object(11)
memory usage: 1.4+ MB
```

```
In [343... df['Dur_Hour']=df['Dur_Hour'].astype(int)
#df['Dur_Minu']=df['Dur_Minu'].astype(int)
```

```
In [344... df['Dur_Minu'].value_counts
```

```
Out[344]: <bound method IndexOpsMixin.value_counts of 0          50
1           25
2          NaN
3           25
4           45
...
10678       30
10679       35
10680       NaN
10681       40
10682       20
Name: Dur_Minu, Length: 10683, dtype: object>
```

```
In [345... # replace missing values
df['Dur_Minu']=df['Dur_Minu'].replace(np.nan,0)
```

```
In [346... df['Dur_Minu'].value_counts
```

```
Out[346]: <bound method IndexOpsMixin.value_counts of 0      50
1         25
2         0
3         25
4         45
..
10678     30
10679     35
10680      0
10681     40
10682     20
Name: Dur_Minu, Length: 10683, dtype: object>
```

```
In [347... df['Dur_Minu']=df['Dur_Minu'].astype(int)
```

```
In [348... df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 17 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  int64
12  Arrival_min            10683 non-null  int64
13  Dep_hour               10683 non-null  object
14  Dep_minute             10683 non-null  object
15  Dur_Hour               10683 non-null  int64
16  Dur_Minu               10683 non-null  int64
dtypes: int64(8), object(9)
memory usage: 1.4+ MB
```

```
In [349... df['Dep_hour']=df['Dep_hour'].astype(int)
df['Dep_minute']=df['Dep_minute'].astype(int)
df.drop('Duration',axis=1,inplace=True)
```

```
In [350... df.head(2)
```

Out[350]:

| | Airline | Source | Destination | Route | Total_Stops | Additional_Info | Price | Date | Month | Year |
|---|-----------|----------|-------------|-----------------------------------|-------------|-----------------|-------|------|-------|------|
| 0 | IndiGo | Banglore | New Delhi | BLR → DEL | non-stop | No info | 3897 | 24 | 3 | 2019 |
| 1 | Air India | Kolkata | Banglore | CCU → IXR → BBI → BLR | 2 stops | No info | 7662 | 1 | 5 | 2019 |

In [351... `df['Total_Stops'].unique()`

Out[351]: array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
dtype=object)

In [352... *#convert category to numerical*
`df['Total_Stops']=df['Total_Stops'].map({'non-stop':0, '1 stop':1, '2 stops':2, '3 sto`

In [353... `df['Total_Stops'].isnull().sum()`

Out[353]: 0

In [355... `df.head(2)`

Out[355]:

| | Airline | Source | Destination | Route | Total_Stops | Additional_Info | Price | Date | Month | Year |
|---|-----------|----------|-------------|-----------------------------------|-------------|-----------------|-------|------|-------|------|
| 0 | IndiGo | Banglore | New Delhi | BLR → DEL | 0 | No info | 3897 | 24 | 3 | 2019 |
| 1 | Air India | Kolkata | Banglore | CCU → IXR → BBI → BLR | 2 | No info | 7662 | 1 | 5 | 2019 |

In [356... `df['Additional_Info'].unique()`

Out[356]: array(['No info', 'In-flight meal not included',
'No check-in baggage included', '1 Short layover', 'No Info',
'1 Long layover', 'Change airports', 'Business class',
'Red-eye flight', '2 Long layover'], dtype=object)

In [357... `df['Source'].unique()`

Out[357]: array(['Banglore', 'Kolkata', 'Delhi', 'Chennai', 'Mumbai'], dtype=object)

In [358... `df['Airline'].unique()`

```
Out[358]: array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
                'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
                'Vistara Premium economy', 'Jet Airways Business',
                'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

```
In [359... df['Destination'].unique()
```

```
Out[359]: array(['New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi', 'Hyderabad'],
                dtype=object)
```

```
In [360... from sklearn.preprocessing import OneHotEncoder
```

```
In [361... #Encode categorical features as a one-hot numeric array.
encoder=OneHotEncoder()
```

```
In [362... encoder.fit_transform(df[['Source', 'Airline', 'Destination']]).toarray()
```

```
Out[362]: array([[1., 0., 0., ..., 0., 0., 1.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 1., ..., 0., 0., 0.],
                ...,
                [1., 0., 0., ..., 0., 0., 0.],
                [1., 0., 0., ..., 0., 0., 1.],
                [0., 0., 1., ..., 0., 0., 0.]])
```


```
In [363... df1=pd.DataFrame(encoder.fit_transform(df[['Source', 'Airline', 'Destination']]).toar
```

```
In [373... df1.head(5)
```

```
Out[373]:
```

| | Source_Banglore | Source_Chennai | Source_Delhi | Source_Kolkata | Source_Mumbai | Airline_Air Asia | Ai |
|---|-----------------|----------------|--------------|----------------|---------------|------------------|----|
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | |
| 2 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| 3 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | |
| 4 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

5 rows × 23 columns

◀  ▶

```
In [375... df
```

Out[375]:

| | Route | Total_Stops | Additional_Info | Price | Date | Month | Year | Arrival_hours | Arrival_min |
|--------------|---|-------------|-----------------|-------|------|-------|------|---------------|-------------|
| 0 | BLR → DEL | 0 | No info | 3897 | 24 | 3 | 2019 | 1 | 10 |
| 1 | CCU → IXR → BBI → BLR | 2 | No info | 7662 | 1 | 5 | 2019 | 13 | 15 |
| 2 | DEL → LKO → BOM → COK | 2 | No info | 13882 | 9 | 6 | 2019 | 4 | 25 |
| 3 | CCU → NAG → BLR | 1 | No info | 6218 | 12 | 5 | 2019 | 23 | 30 |
| 4 | BLR → NAG → DEL | 1 | No info | 13302 | 1 | 3 | 2019 | 21 | 35 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 10678 | CCU → BLR | 0 | No info | 4107 | 9 | 4 | 2019 | 22 | 25 |
| 10679 | CCU → BLR | 0 | No info | 4145 | 27 | 4 | 2019 | 23 | 20 |
| 10680 | BLR → DEL | 0 | No info | 7229 | 27 | 4 | 2019 | 11 | 20 |
| 10681 | BLR → DEL | 0 | No info | 12648 | 1 | 3 | 2019 | 14 | 10 |
| 10682 | DEL → GOI → BOM → COK | 2 | No info | 11753 | 9 | 5 | 2019 | 19 | 15 |

10683 rows × 13 columns



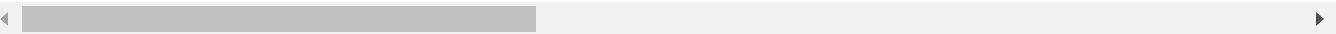
```
In [376... df3=(pd.concat([df1,df]))
```

```
In [377... df3
```

Out[377]:

| | Source_Banglore | Source_Chennai | Source_Delhi | Source_Kolkata | Source_Mumbai | Airline_Ai Asi |
|-------|-----------------|----------------|--------------|----------------|---------------|-------------------|
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 4 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... |
| 10678 | NaN | NaN | NaN | NaN | NaN | NaN |
| 10679 | NaN | NaN | NaN | NaN | NaN | NaN |
| 10680 | NaN | NaN | NaN | NaN | NaN | NaN |
| 10681 | NaN | NaN | NaN | NaN | NaN | NaN |
| 10682 | NaN | NaN | NaN | NaN | NaN | NaN |

21366 rows × 36 columns



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```