EDA and Feature Engineering

Flight Price Prediction Dataset

```
# Importing required libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
%matplotlib inline
```

```
train_df = pd.read_excel("Data_Train.xlsx")
train_df.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Durat
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 5
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI	05:50	13:15	7h 2

```
test_df = pd.read_excel("Test_set.xlsx")
test_df.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Durat
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 5
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU → MAA	06:20	10:20	

```
df = train_df.append(test_df)
df.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Durat
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 5
					CCU → IXR			

```
1 df.tail()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Dura
2666	Air India	6/06/2019	Kolkata	Banglore	CCU → DEL → BLR	20:30	20:25 07 Jun	23
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU → BLR	14:20	16:55	2

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Date_of_Journey  13354 non-null  object 
 2   Source           13354 non-null  object 
 3   Destination      13354 non-null  object 
 4   Route            13353 non-null  object 
 5   Dep_Time         13354 non-null  object 
 6   Arrival_Time     13354 non-null  object 
 7   Duration         13354 non-null  object 
 8   Total_Stops      13353 non-null  object 
 9   Additional_Info  13354 non-null  object 
 10  Price            10683 non-null  float64
dtypes: float64(1), object(10)
memory usage: 1.2+ MB
```

```
1 ## Feature Engineering Process
2 ## Derive nea feature from Data_of_Journey
3 # df['Date'] , df['Month'] , df['Year']
4
5 # df['Date'] = df['Date_of_Journey'].str.split('/').str[0]
6 # df['Month'] = df['Date_of_Journey'].str.split('/').str[1]
7 # df['Year'] = df['Date_of_Journey'].str.split('/').str[2]
8
9 df['Date'] = df['Date_of_Journey'].apply(lambda x:x.split('/')[0])
```

```
df['Month'] = df['Date_of_Journey'].apply(lambda x:x.split('/')[1])
df['Year'] = df['Date_of_Journey'].apply(lambda x:x.split('/')[2])

```

```
df.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Durat
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 5
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI	05:50	13:15	7h 2

```
## convert string type into integer
df['Date'] = df['Date'].astype(int)
df['Month'] = df['Month'].astype(int)
df['Year'] = df['Year'].astype(int)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Date_of_Journey  13354 non-null  object 
 2   Source           13354 non-null  object 
 3   Destination      13354 non-null  object 
 4   Route            13353 non-null  object 
 5   Dep_Time         13354 non-null  object 
 6   Arrival_Time     13354 non-null  object 
 7   Duration         13354 non-null  object 
 8   Total_Stops      13353 non-null  object 
 9   Additional_Info  13354 non-null  object 
 10  Price            10683 non-null  float64
 11  Date             13354 non-null  int64  
 12  Month            13354 non-null  int64  
 13  Year             13354 non-null  int64  
dtypes: float64(1), int64(3), object(10)
memory usage: 1.5+ MB
```

```
## drop date_of_jounrey

df.drop('Date_of_Journey', axis=1, inplace=True)
df.head(1)
```

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops
0	IndiGo	Banglore	New Delhi	BLR →	22:20	01:10 22 Mar	2h 50m	non-stop

```
## remooving month from Arrival_Time - 01:10 22 Mar  -> 01:10
df['Arrival_Time'] = df['Arrival_Time'].apply(lambda x : x.split(' ')[0])
```

```
# Creating Arrival_Hour column from arrival time
df['Arrival_Hour'] = df['Arrival_Time'].apply(lambda x : x.split(':')[0])

# Creating Arrival_Min column from arrival time
df['Arrival_Min'] = df['Arrival_Time'].apply(lambda x : x.split(':')[1])
```

```
df.head()
```

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10	2h 50m	non-stop
1	Air India	Kolkata	Banglore	CCU → IXR → BBI	05:50	13:15	7h 25m	2 stops

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Source           13354 non-null  object 
 2   Destination      13354 non-null  object 
 3   Route            13353 non-null  object 
 4   Dep_Time         13354 non-null  object 
 5   Arrival_Time     13354 non-null  object 
 6   Duration         13354 non-null  object 
 7   Total_Stops      13353 non-null  object 
 8   Additional_Info  13354 non-null  object 
 9   Price            10683 non-null  float64
 10  Date             13354 non-null  int64  
 11  Month            13354 non-null  int64  
 12  Year             13354 non-null  int64  
 13  Arrival_Hour     13354 non-null  object 
 14  Arrival_Min      13354 non-null  object 
dtypes: float64(1), int64(3), object(11)
memory usage: 1.6+ MB
```

```
## Conver Arrival Hour and Arrival Minutes columns into integer
df['Arrival_Hour'] = df['Arrival_Hour'].astype(int)
df['Arrival_Min'] = df['Arrival_Min'].astype(int)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Source           13354 non-null  object 
 2   Destination      13354 non-null  object 
 3   Route            13353 non-null  object 
 4   Dep_Time         13354 non-null  object 
 5   Arrival_Time     13354 non-null  object 
 6   Duration         13354 non-null  object 
 7   Total_Stops      13353 non-null  object 
 8   Additional_Info  13354 non-null  object 
 9   Price            10683 non-null  float64
 10  Date             13354 non-null  int64  
 11  Month            13354 non-null  int64  
 12  Year             13354 non-null  int64  
 13  Arrival_Hour     13354 non-null  int64  
 14  Arrival_Min      13354 non-null  int64  
dtypes: float64(1), int64(5), object(9)
memory usage: 1.6+ MB
```

```
## Drop th Arrival_Time

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Airline          13354 non-null  object 
 1   Source           13354 non-null  object 
 2   Destination      13354 non-null  object 
 3   Route            13353 non-null  object 
 4   Dep_Time         13354 non-null  object 
 5   Duration         13354 non-null  object 
 6   Total_Stops      13353 non-null  object 
 7   Additional_Info  13354 non-null  object 
 8   Price            10683 non-null  float64
 9   Date             13354 non-null  int64  
 10  Month            13354 non-null  int64  
```

```
 11  Year             13354 non-null  int64
 12  Arrival_Hour     13354 non-null  int64
 13  Arrival_Min      13354 non-null  int64
dtypes: float64(1), int64(5), object(8)
memory usage: 1.5+ MB
```

```
# Creating Depature Hour and Depature min from from depature time
df['Dep_Hour'] = df['Dep_Time'].apply(lambda x : x.split(':')[0])
df['Dep_Min'] = df['Dep_Time'].apply(lambda x : x.split(':')[1])

#convert into int type
df['Dep_Hour'] = df['Dep_Hour'].astype(int)
df['Dep_Min'] = df['Dep_Min'].astype(int)
```

```
# drop Dep_time column

df.drop('Dep_Time', axis=1, inplace=True)
```

```
df.head()
```

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897.0
1	Air India	Kolkata	Banglore	CCU → IXR → BBI	7h 25m	2 stops	No info	7662.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13354 entries, 0 to 2670
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          13354 non-null  object
 1   Source           13354 non-null  object
 2   Destination      13354 non-null  object
 3   Route            13353 non-null  object
 4   Duration         13354 non-null  object
 5   Total_Stops      13353 non-null  object
 6   Additional_Info  13354 non-null  object
 7   Price            10683 non-null  float64
 8   Date             13354 non-null  int64
 9   Month            13354 non-null  int64
 10  Year             13354 non-null  int64
 11  Arrival_Hour     13354 non-null  int64
 12  Arrival_Min      13354 non-null  int64
```

```
 13  Dep_Hour         13354 non-null  int64
 14  Dep_Min          13354 non-null  int64
dtypes: float64(1), int64(7), object(7)
memory usage: 1.6+ MB
```

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

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

```
# convert Total_Stops into numeric

df['Total_Stops'] = df['Total_Stops'].map({'non-stop':0,'1 stop':1, '2 stops':2, '3 stops
```

```
df.head()
```

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	0.0	No info	3897.0
1	Air India	Kolkata	Banglore	CCU → IXR → BBI	7h 25m	2.0	No info	7662.0

```
df.drop('Route', axis=1, inplace=True)
```

```
df.head()
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date
0	IndiGo	Banglore	New Delhi	2h 50m	0.0	No info	3897.0	24
1	Air India	Kolkata	Banglore	7h 25m	2.0	No info	7662.0	1
2	Jet Airways	Delhi	Cochin	19h	2.0	No info	13882.0	9
3	IndiGo	Kolkata	Banglore	5h 25m	1.0	No info	6218.0	12
4	IndiGo	Banglore	New Delhi	4h 45m	1.0	No info	13302.0	1

```
#convert Duration columns into minutes

df['Duration_hour'] = df["Duration"].str.split(' ').str[0].str.split('h').str[0]
```

```
df.head()
```

```
df.head()
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date
0	IndiGo	Banglore	New Delhi	2h 50m	0.0	No info	3897.0	24
1	Air India	Kolkata	Banglore	7h 25m	2.0	No info	7662.0	1
2	Jet Airways	Delhi	Cochin	19h	2.0	No info	13882.0	9
3	IndiGo	Kolkata	Banglore	5h 25m	1.0	No info	6218.0	12
4	IndiGo	Banglore	New Delhi	4h 45m	1.0	No info	13302.0	1

```
#There could be a change if any of df['Duration_hour'] column has less than 1 hour so it
df[df['Duration_hour'].str.contains('m')]
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Da
6474	Air India	Mumbai	Hyderabad	5m	2.0	No info	17327.0	
2660	Air India	Mumbai	Hyderabad	5m	2.0	No info	NaN	

```
#3 This is bad data so we can drop this column
df.drop(df[df['Duration_hour'].str.contains('m')].index, inplace=True)
```

```
df[df['Duration_hour'].str.contains('m')]
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date	Mon

```
df['Duration_hour'] = df['Duration_hour'].astype(int)
```

```
df.head()
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date
0	IndiGo	Banglore	New Delhi	2h 50m	0.0	No info	3897.0	24
1	Air India	Kolkata	Banglore	7h 25m	2.0	No info	7662.0	1
2	Jet Airways	Delhi	Cochin	19h	2.0	No info	13882.0	9
3	IndiGo	Kolkata	Banglore	5h 25m	1.0	No info	6218.0	12
4	IndiGo	Banglore	New Delhi	4h 45m	1.0	No info	13302.0	1

```
## convert hours into minutes
df['Duration_hour'] = df['Duration_hour'] * 60
```

```
df.head()
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date
0	IndiGo	Banglore	New Delhi	2h 50m	0.0	No info	3897.0	24
1	Air India	Kolkata	Banglore	7h 25m	2.0	No info	7662.0	1
2	Jet Airways	Delhi	Cochin	19h	2.0	No info	13882.0	9
3	IndiGo	Kolkata	Banglore	5h 25m	1.0	No info	6218.0	12
4	IndiGo	Banglore	New Delhi	4h 45m	1.0	No info	13302.0	1

```
# convers minues into numeric and create a new column called duration
df["Duration"].str.split(' ').str[1].str.split('m').str[0]
```

```
0        50
1        25
2       NaN
3        25
4        45
       ...
2666     55
2667     35
2668     35
2669     15
2670     20
Name: Duration, Length: 13351, dtype: object
```

```
df['Duration_min'] = df["Duration"].str.split(' ').str[1].str.split('m').str[0]
```

```
# Check null values because if any trip has only hours so minutes will be 0 that will be
df['Duration_min'].isnull().sum()
```

```
1283
```

```
df['Duration_min'].unique()
```

```
array(['50', '25', nan, '45', '30', '5', '15', '35', '10', '20', '55',
       '40'], dtype=object)
```

```
df['Duration_min'] = df['Duration_min'].fillna(0)
```

```
df['Duration_min'].unique()
```

```
array(['50', '25', 0, '45', '30', '5', '15', '35', '10', '20', '55', '40'],
      dtype=object)
```

```
#convert df['Duration_min'] into minutes

df['Duration_min'] = df['Duration_min'].astype(int)
```

```

df.head()
```

	Airline	Source	Destination	Duration	Total_Stops	Additional_Info	Price	Date
0	IndiGo	Banglore	New Delhi	2h 50m	0.0	No info	3897.0	24
1	Air India	Kolkata	Banglore	7h 25m	2.0	No info	7662.0	1
2	Jet Airways	Delhi	Cochin	19h	2.0	No info	13882.0	9
3	IndiGo	Kolkata	Banglore	5h 25m	1.0	No info	6218.0	12
4	IndiGo	Banglore	New Delhi	4h 45m	1.0	No info	13302.0	1

```
# drop duration column
df.drop('Duration', axis=1, inplace=True)
```

```
df.head()

```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Ye
0	IndiGo	Banglore	New Delhi	0.0	No info	3897.0	24	3	20
1	Air India	Kolkata	Banglore	2.0	No info	7662.0	1	5	20
2	Jet Airways	Delhi	Cochin	2.0	No info	13882.0	9	6	20
3	IndiGo	Kolkata	Banglore	1.0	No info	6218.0	12	5	20
4	IndiGo	Banglore	New Delhi	1.0	No info	13302.0	1	3	20

```
#create a new column by adding duration hour and minuts

df['Duration_in_minutes'] = df['Duration_hour'] + df['Duration_min']
```

```
df.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Ye
0	IndiGo	Banglore	New Delhi	0.0	No info	3897.0	24	3	20
1	Air India	Kolkata	Banglore	2.0	No info	7662.0	1	5	20
2	Jet Airways	Delhi	Cochin	2.0	No info	13882.0	9	6	20

```
# now drop duration hour and duration_min column

df.drop('Duration_hour', axis=1, inplace=True)
df.drop('Duration_min', axis=1, inplace=True)
```

```
df.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Ye
0	IndiGo	Banglore	New Delhi	0.0	No info	3897.0	24	3	20
1	Air India	Kolkata	Banglore	2.0	No info	7662.0	1	5	20
2	Jet Airways	Delhi	Cochin	2.0	No info	13882.0	9	6	20
3	IndiGo	Kolkata	Banglore	1.0	No info	6218.0	12	5	20
4	IndiGo	Banglore	New Delhi	1.0	No info	13302.0	1	3	20

```
## Working with categorical features
df['Airline'].unique()
```

```
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)
```

```
#Lebel encoding
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
```

```
df['Airline'] = labelencoder.fit_transform(df['Airline'])
df['Source'] = labelencoder.fit_transform(df['Source'])
df['Destination'] = labelencoder.fit_transform(df['Destination'])
df['Additional_Info'] = labelencoder.fit_transform(df['Additional_Info'])
```

```
df.shape
```

```
(13351, 14)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13351 entries, 0 to 2670
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Airline              13351 non-null  int64  
 1   Source               13351 non-null  int64  
 2   Destination          13351 non-null  int64  
 3   Total_Stops          13350 non-null  float64
 4   Additional_Info      13351 non-null  int64  
 5   Price                10681 non-null  float64
 6   Date                 13351 non-null  int64  
 7   Month                13351 non-null  int64  
 8   Year                 13351 non-null  int64  
 9   Arrival_Hour         13351 non-null  int64  
 10  Arrival_Min          13351 non-null  int64  
 11  Dep_Hour             13351 non-null  int64  
 12  Dep_Min              13351 non-null  int64  
 13  Duration_in_minutes  13351 non-null  int64  
dtypes: float64(2), int64(12)
memory usage: 1.5 MB
```

```
df.describe()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Pr
count	13351.000000	13351.000000	13351.000000	13350.000000	13351.000000	10681.000
mean	3.977530	1.953786	1.435248	0.825768	7.407610	9085.898
std	2.363982	1.178474	1.473404	0.674478	1.198494	4610.921
min	0.000000	0.000000	0.000000	0.000000	0.000000	1759.000
25%	3.000000	2.000000	0.000000	0.000000	8.000000	5277.000
50%	4.000000	2.000000	1.000000	1.000000	8.000000	8372.000
75%	4.000000	3.000000	2.000000	1.000000	8.000000	12373.000
max	11.000000	4.000000	5.000000	4.000000	9.000000	79512.000

```
df.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Yea
0	3	0	5	0.0	8	3897.0	24	3	201
1	1	3	0	2.0	8	7662.0	1	5	201

```
# # we can perform on hot encoding as well
df = pd.get_dummies(df, columns=["Airline","Source","Destination","Additional_Info"], dr

```

4	3	0	5	1.0	8	13302.0	1	3	201

```
df.head()
```

	Total_Stops	Price	Date	Month	Year	Arrival_Hour	Arrival_Min	Dep_Hour	Dep_Min
0	0.0	3897.0	24	3	2019	1	10	22	20
1	2.0	7662.0	1	5	2019	13	15	5	50
2	2.0	13882.0	9	6	2019	4	25	9	25
3	1.0	6218.0	12	5	2019	23	30	18	5
4	1.0	13302.0	1	3	2019	21	35	16	50

5 rows × 39 columns

```
#! pip install https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
```

```
#checkomg th distibution of the data
from pandas_profiling import ProfileReport
profile = ProfileReport(df, title='Pandas Profiling Report', html={'style':{'full_width':
profile.to_notebook_iframe()
```

Summarize dataset: 100% 120/120 [00:34<00:00, 3.49it/s, Com

Generate report structure: 100% 1/1 [00:13<00:00, 13.92s/it]

Render HTML: 100% 1/1 [00:02<00:00, 2.87s/it]

Overview

Dataset statistics

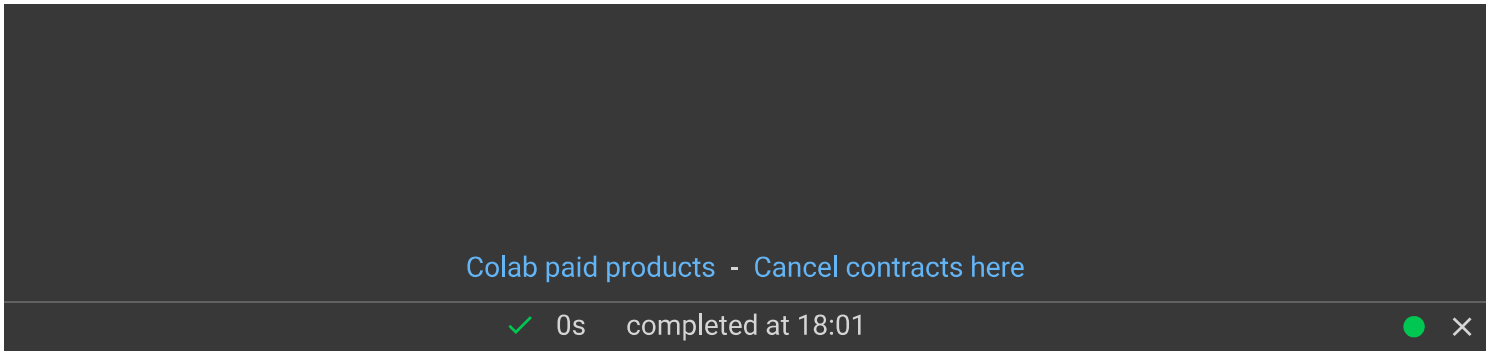
Number of variables	40
Number of observations	13351
Missing cells	2671
Missing cells (%)	0.5%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	1.5 MiB
Average record size in memory	117.0 B

Variable types

Numeric	8
Categorical	32

```
1
```

`Year` has constant value "2019"	Consta
`df_index` is highly correlated with `Year`	High co
`Price` is highly correlated with `Airline_5` and 1 other fields (Airline_5, Additional_Info_3)	High co
`Date` is highly correlated with `Year`	High co
`Arrival_Hour` is highly correlated with `Arrival_Min` and 2 other fields (Arrival_Min, Dep_Hour, Airline_4)	High co
`Arrival_Min` is highly correlated with `Arrival_Hour`	High co


Colab paid products - Cancel contracts here
0s completed at 18:01