

Project Report Template

1 INTRODUCTION

Overview

A brief description about your project

Purpose

The use of this project. What can be achieved using this.

2 Problem Definition & Design Thinking

Empathy Map

Paste the empathy map screenshot

Ideation & Brainstorming Map

Paste the Ideation & brainstorming map screenshot

3 RESULT

Final findings (Output) of the project along with screenshots.

4 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

5 APPLICATIONS

The areas where this solution can be applied

6 CONCLUSION

Conclusion summarizing the entire work and findings.

7 FUTURE SCOPE

Enhancements that can be made in the future.

8 APPENDIX

A. Source Code

Attach the code for the solution built.

1. INTRODUCTION:

1.1 OVERVIEW

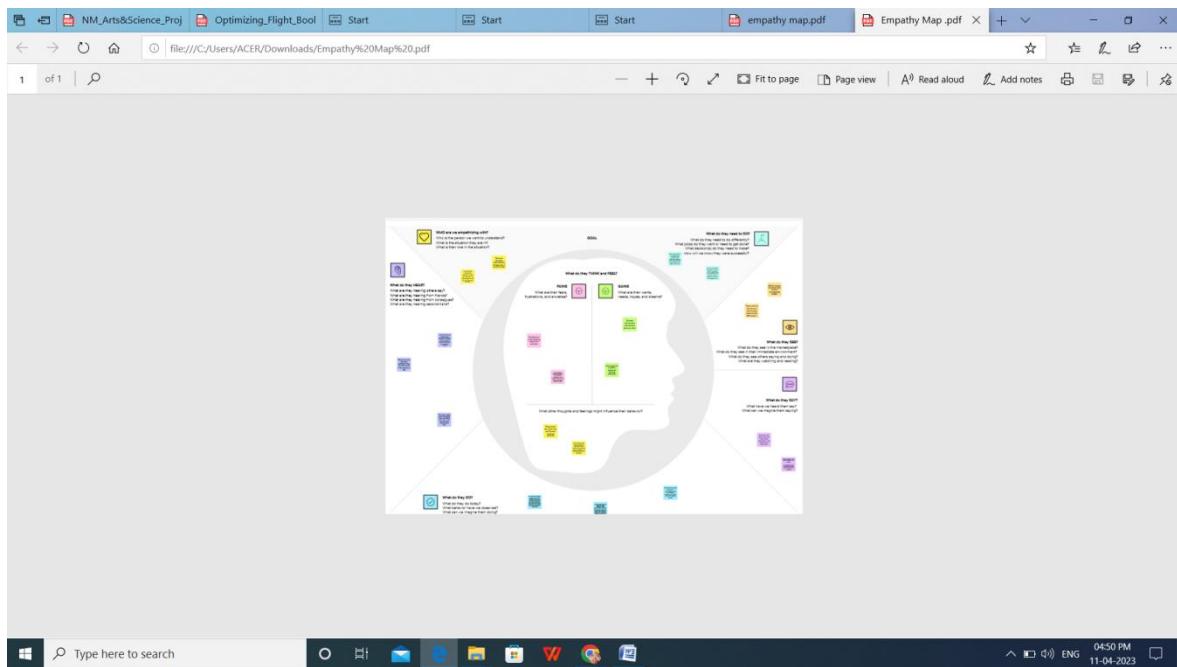
People who work frequently travel through flight will have better knowledge on best discount and right time to buy the ticket. For the business purpose many airline companies change prices according to the seasons or time duration. They will increase the price when people travel more. Estimating the highest prices of the airlines data for the route is collected with features such as Duration, Source, Destination, Arrival and Departure.

1.2 PURPOSE

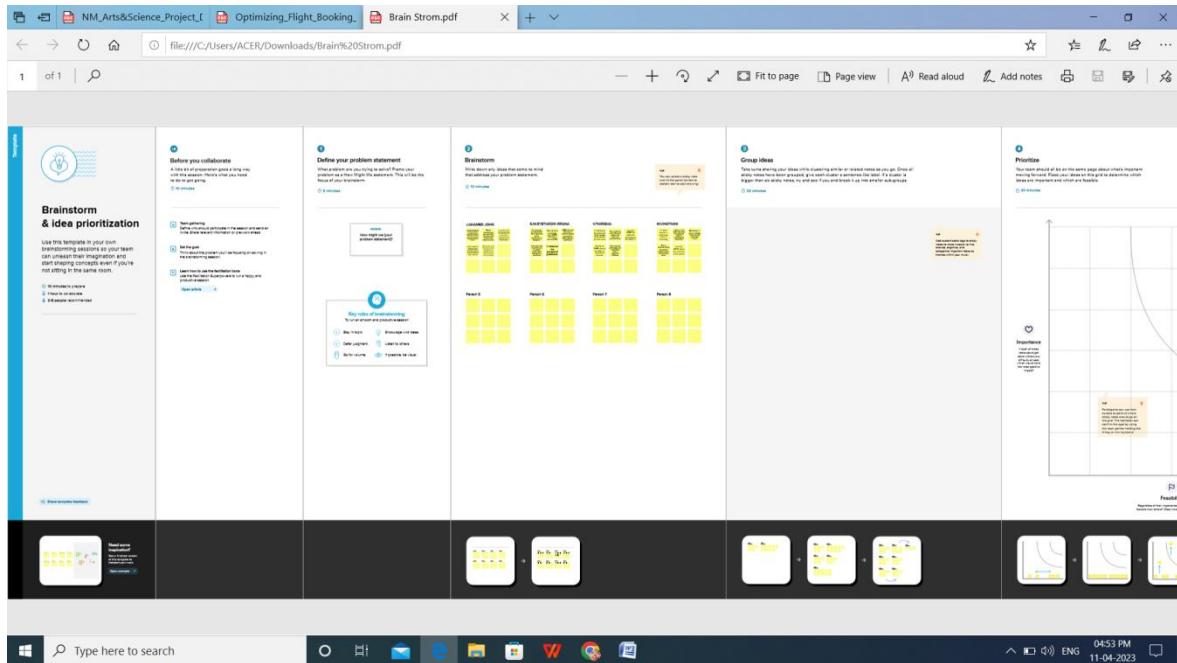
Features are taken from chosen dataset and in the price where in the airline price ticket costs vary overtime. we have implemented flight price prediction for users by using KNN, decision tree and random forest algorithms. Random Forest shows the best accuracy of 80% for predicting the flight price. Also, we have done correlation tests and metrics for the statistical analysis.

2. PROBLEM DEFINITION & DESIGN THINKING

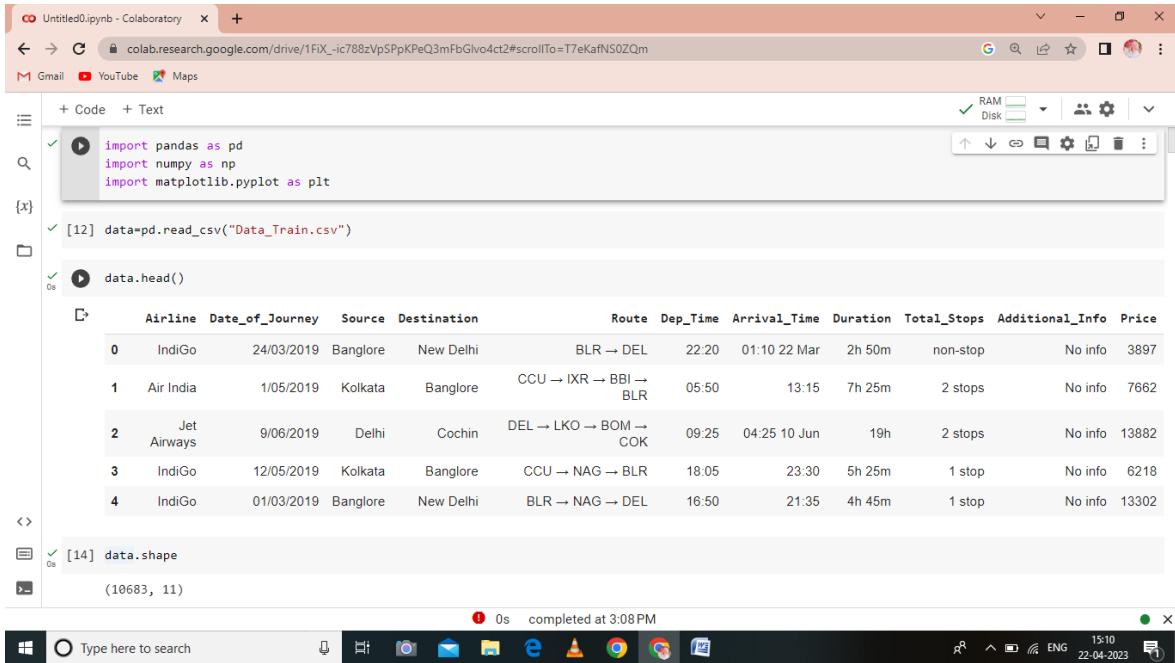
2.1 EMPATHY MAP



2.2 IDEATION & BRAINSTORMING MAP



3. RESULT:



Untitled0.ipynb - Colaboratory

`import pandas as pd
import numpy as np
import matplotlib.pyplot as plt`

[12] `data=pd.read_csv("Data_Train.csv")`

[13] `data.head()`

	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

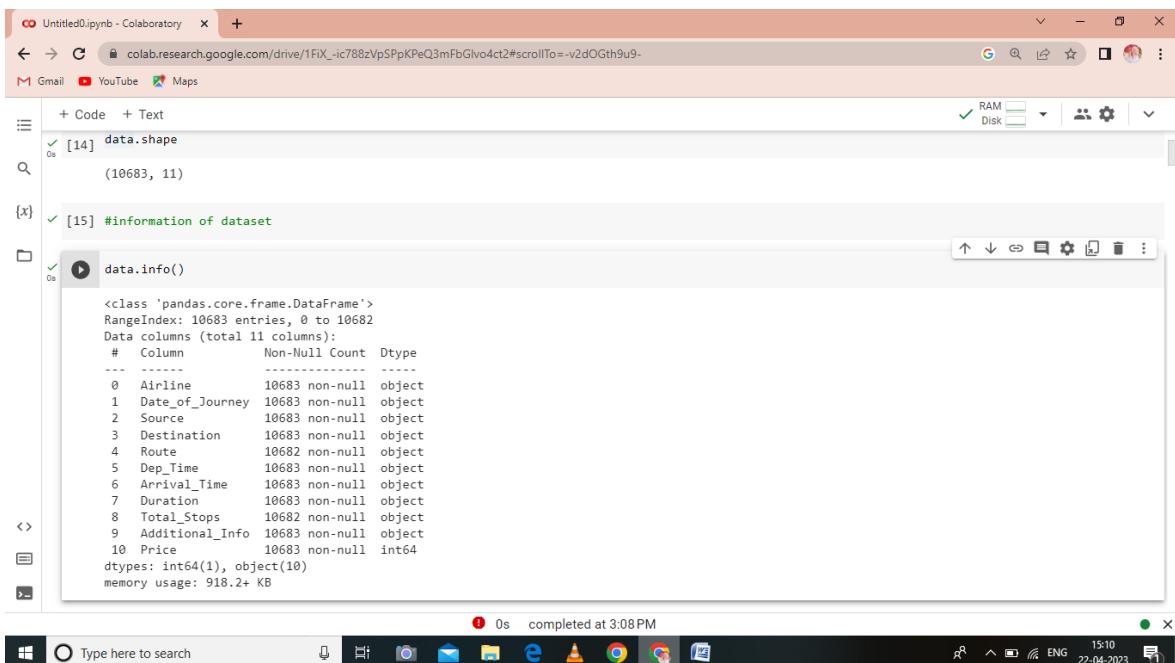
[14] `data.shape`

(10683, 11)

0s completed at 3:08 PM

RAM Disk

15:10 22-04-2023



Untitled0.ipynb - Colaboratory

`data.shape`

(10683, 11)

[15] #information of dataset

`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

```

0s completed at 3:08 PM

RAM Disk

15:10 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=1kkd3GNtfEvI

Gmail YouTube Maps

RAM Disk

```
[16]: 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
```

```
[17]: data.isnull().sum()
```

```
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
```

```
[20]: data.dropna(inplace=True)
```

0s completed at 3:08PM

Type here to search

15:10 ENG 22-04-2023

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=aTnP2m0A15R

Gmail YouTube Maps

RAM Disk

```
[20]: Price      0
      dtype: int64
```

```
[21]: data.dropna(inplace=True)
```

```
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
```

```
[22]: #creating list of category columns
category=['Airline','Source','Destination','Additional_Info']
category
```

```
['Airline', 'Source', 'Destination', 'Additional_Info']
```

0s completed at 3:08PM

Type here to search

15:11 ENG 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=tLrq9AqExJYW

Gmail YouTube Maps

```
+ Code + Text
[22]: category=[ 'Airline','Source','Destination','Additional_Info']
category
['Airline', 'Source', 'Destination', 'Additional_Info']

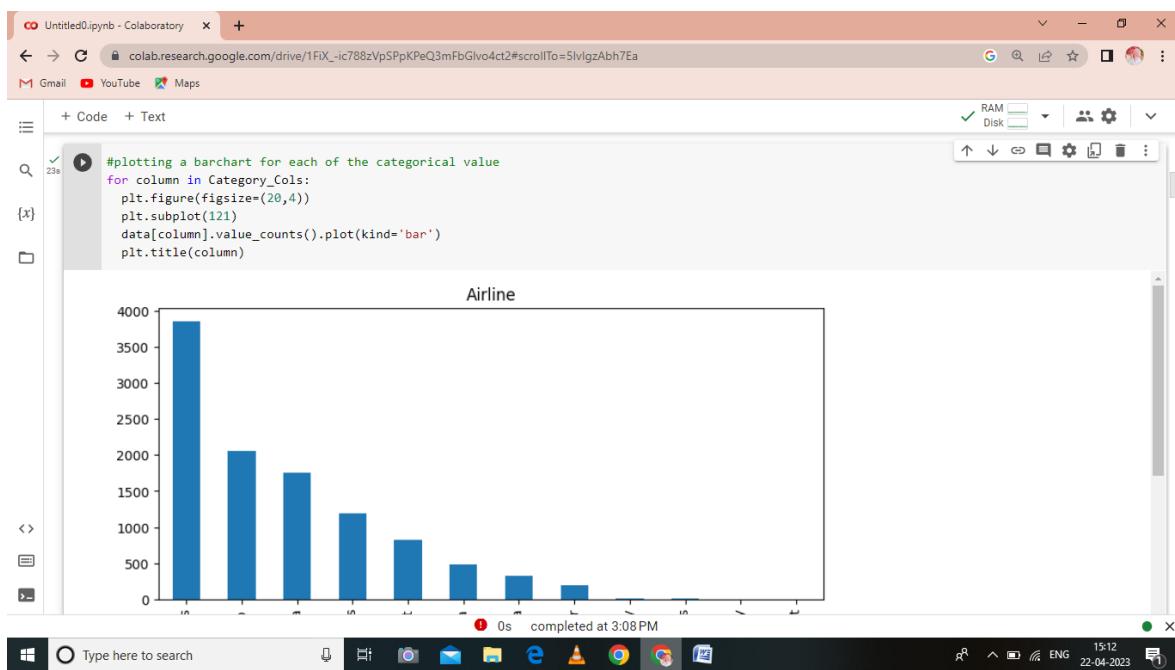
for i in category:
    print(i, data[i].unique())

Airline ['IndiGo' 'Air India' 'Jet Airways' 'SpiceJet' 'Multiple carriers' 'GoAir'
'Vistara' 'Air Asia' 'Vistara Premium economy' 'Jet Airways Business'
'Multiple carriers Premium economy' 'Trujet']
Source ['Banglore' 'Kolkata' 'Delhi' 'Chennai' 'Mumbai']
Destination ['New Delhi' 'Banglore' 'Cochin' 'Kolkata' 'Delhi' 'Hyderabad']
Additional_Info ['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']

[24]: Category_Cols=data.select_dtypes(include=['object']).columns
Category_Cols
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info'],
      dtype='object')
```

0s completed at 3:08 PM

Type here to search



Project Report Template

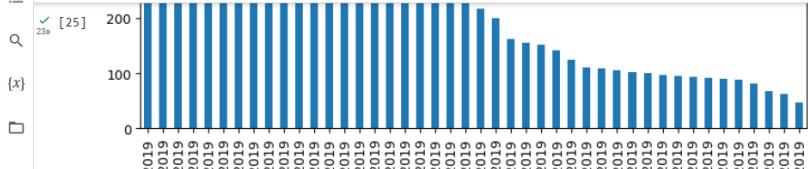
Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=hgMP7y8tBOeu

Gmail YouTube Maps

+ Code + Text

23s [25]



{x}

RAM Disk

```
#Since the maximum number of stops is 4, there should be maximum 6 cities in any particular route. we split the data in route columns.
data.Route=data.Route.str.split('>')
data.Route
```

```
0      [BLR > DEL]
1      [CCU > IXR > BBI > BLR]
2      [DEL > LKO > BOM > COK]
3      [CCU > NAG > BLR]
4      [BLR > NAG > DEL]
...
10678     [CCU > BLR]
10679     [CCU > BLR]
10680     [BLR > DEL]
10681     [BLR > DEL]
10682     [DEL > GOI > BOM > COK]
Name: Route, Length: 10682, dtype: object
```

0s completed at 3:08 PM

Type here to search

15:12 ENG 22-04-2023

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=hgMP7y8tBOeu

Gmail YouTube Maps

+ Code + Text

0s

#Since the maximum number of stops is 4, there should be maximum 6 cities in any particular route. we split the data
data.Route=data.Route.str.split('>')
data.Route

```
0      [BLR > DEL]
1      [CCU > IXR > BBI > BLR]
2      [DEL > LKO > BOM > COK]
3      [CCU > NAG > BLR]
4      [BLR > NAG > DEL]
...
10678     [CCU > BLR]
10679     [CCU > BLR]
10680     [BLR > DEL]
10681     [BLR > DEL]
10682     [DEL > GOI > BOM > COK]
Name: Route, Length: 10682, dtype: object
```

```
[27] data['City1']=data.Route.str[0]
data['City2']=data.Route.str[1]
data['City3']=data.Route.str[2]
data['City4']=data.Route.str[3]
data['City5']=data.Route.str[4]
data['City6']=data.Route.str[5]
```

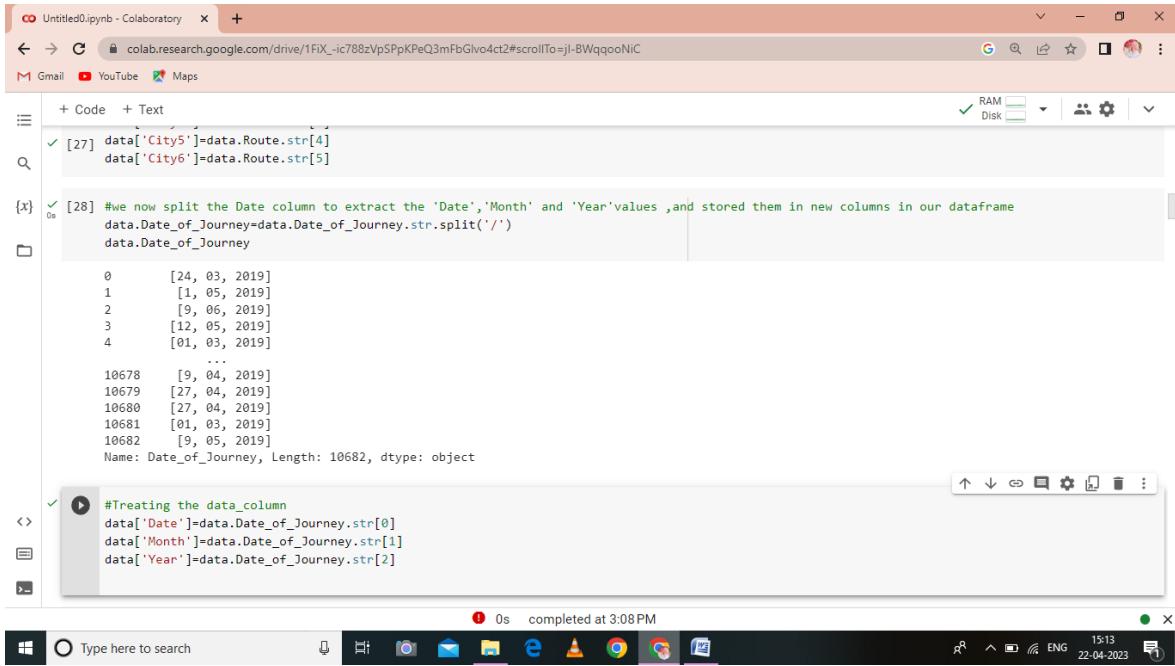
```
[28] #we now split the Date column to extract the 'Date','Month' and 'Year'values ,and stored them in new columns in our dataframe
data.Date_of_Journey=data.Date_of_Journey.str.split('/')
```

0s completed at 3:08 PM

Type here to search

15:12 ENG 22-04-2023

Project Report Template



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=jI-BWqqooNiC
Gmail YouTube Maps
RAM Disk
+ Code + Text
[27] data['City5']=data.Route.str[4]
data['City6']=data.Route.str[5]

{x} [28] #we now split the Date column to extract the 'Date','Month' and 'Year'values ,and stored them in new columns in our dataframe
data.Date_of_Journey=data.Date_of_Journey.str.split('/')
data.Date_of_Journey

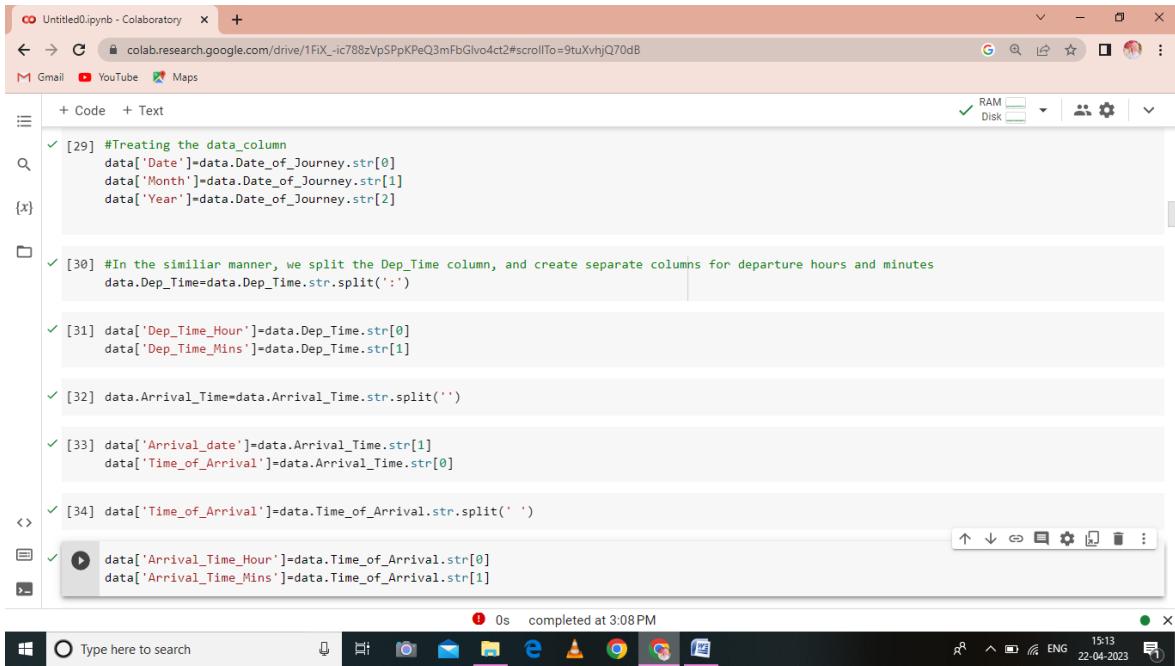
0 [24, 03, 2019]
1 [1, 05, 2019]
2 [9, 06, 2019]
3 [12, 05, 2019]
4 [01, 03, 2019]
...
10678 [9, 04, 2019]
10679 [27, 04, 2019]
10680 [27, 04, 2019]
10681 [01, 03, 2019]
10682 [9, 05, 2019]
Name: Date_of_Journey, Length: 10682, dtype: object

#Treating the data_column
data['Date']=data.Date_of_Journey.str[0]
data['Month']=data.Date_of_Journey.str[1]
data['Year']=data.Date_of_Journey.str[2]

0s completed at 3:08 PM

```

Type here to search 15:13 22-04-2023



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=9tuXvhjQ70dB
Gmail YouTube Maps
RAM Disk
+ Code + Text
[29] #Treating the data_column
data['Date']=data.Date_of_Journey.str[0]
data['Month']=data.Date_of_Journey.str[1]
data['Year']=data.Date_of_Journey.str[2]

{x} [30] #In the similiar manner, we split the Dep_Time column, and create separate columns for departure hours and minutes
data.Dep_Time=data.Dep_Time.str.split(':')

[31] data['Dep_Time_Hour']=data.Dep_Time.str[0]
data['Dep_Time_Mins']=data.Dep_Time.str[1]

[32] data.Arrival_Time=data.Arrival_Time.str.split('')

[33] data['Arrival_date']=data.Arrival_Time.str[1]
data['Time_of_Arrival']=data.Arrival_Time.str[0]

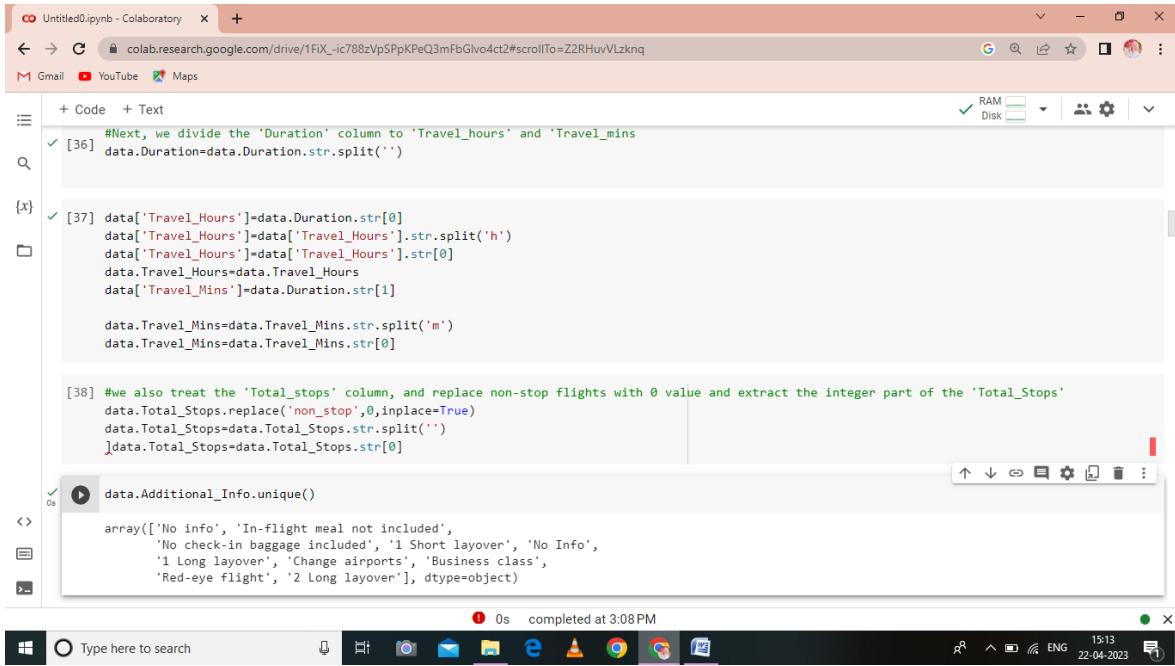
[34] data['Time_of_Arrival']=data.Time_of_Arrival.str.split(' ')
data['Arrival_Time_Hour']=data.Time_of_Arrival.str[0]
data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]

0s completed at 3:08 PM

```

Type here to search 15:13 22-04-2023

Project Report Template



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=Z2RHuvVLzkng
Gmail YouTube Maps
RAM Disk
+ Code + Text
[36] #Next, we divide the 'Duration' column to 'Travel_hours' and 'Travel_mins'
data.Duration=data.Duration.str.split('')

[37] data['Travel_Hours']=data.Duration.str[0]
data['Travel_Hours']=data['Travel_Hours'].str.split('h')
data['Travel_Hours']=data['Travel_Hours'].str[0]
data.Travel_Hours=data.Travel_Hours
data['Travel_Mins']=data.Duration.str[1]

data.Travel_Mins=data.Travel_Mins.str.split('m')
data.Travel_Mins=data.Travel_Mins.str[0]

[38] #we also treat the 'Total_stops' column, and replace non-stop flights with 0 value and extract the integer part of the 'Total_Stops'
data.Total_Stops.replace('non_stop',0,inplace=True)
data.Total_Stops=data.Total_Stops.str.split('')
data.Total_Stops=data.Total_Stops.str[0]

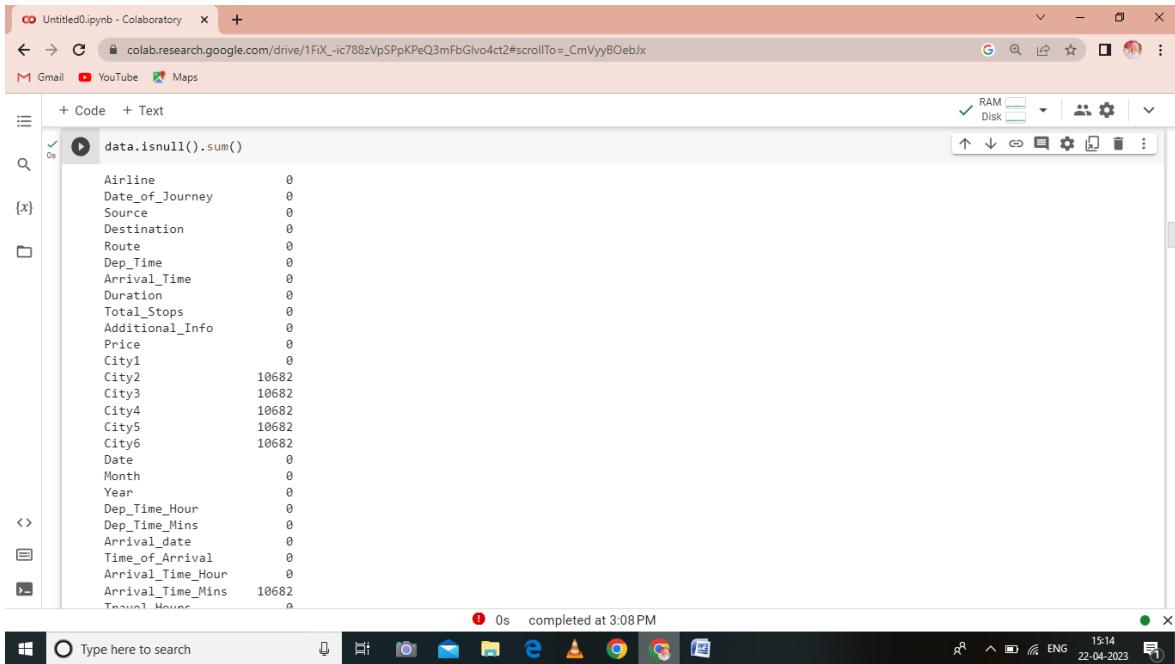
data.Additional_Info.unique()
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)

```

0s completed at 3:08 PM

Type here to search

15:13 22-04-2023



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=_CmVyy8OebJx
Gmail YouTube Maps
RAM Disk
+ Code + Text
[39] data.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
City1            0
City2            10682
City3            10682
City4            10682
City5            10682
City6            10682
Date              0
Month             0
Year              0
Dep_Time_Hour    0
Dep_Time_Mins    0
Arrival_date      0
Time_of_Arrival   0
Arrival_Time_Hour 0
Arrival_Time_Mins 10682
Travel_Hours      0

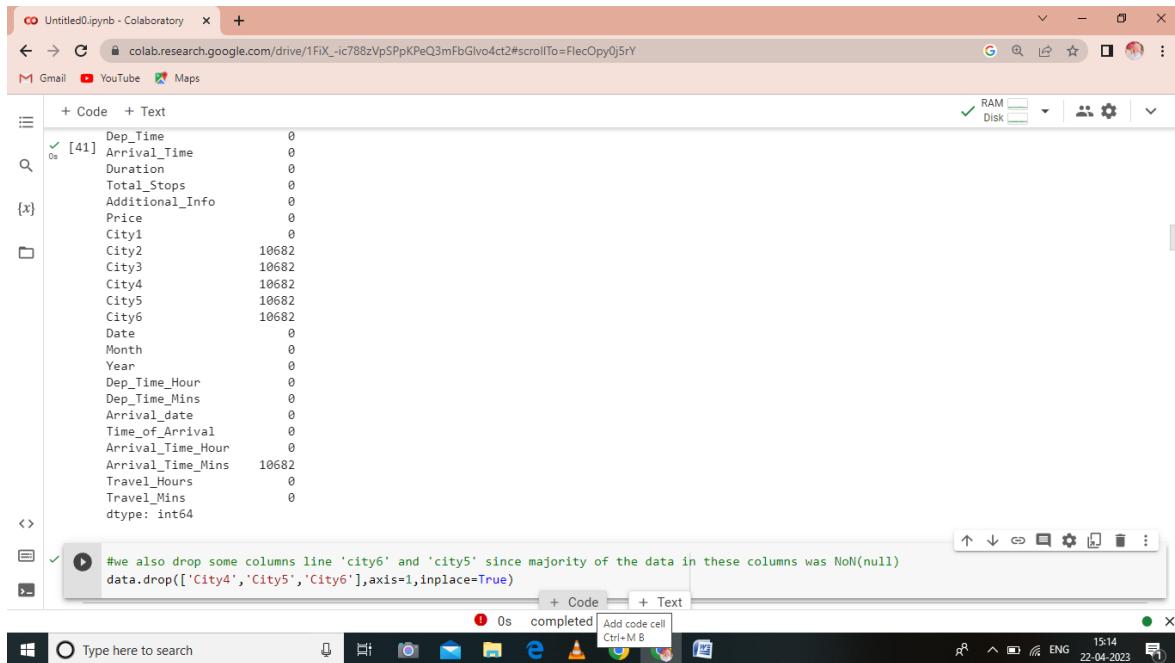
```

0s completed at 3:08 PM

Type here to search

15:14 22-04-2023

Project Report Template



Untitled0.ipynb - Colaboratory

`[41]`

```

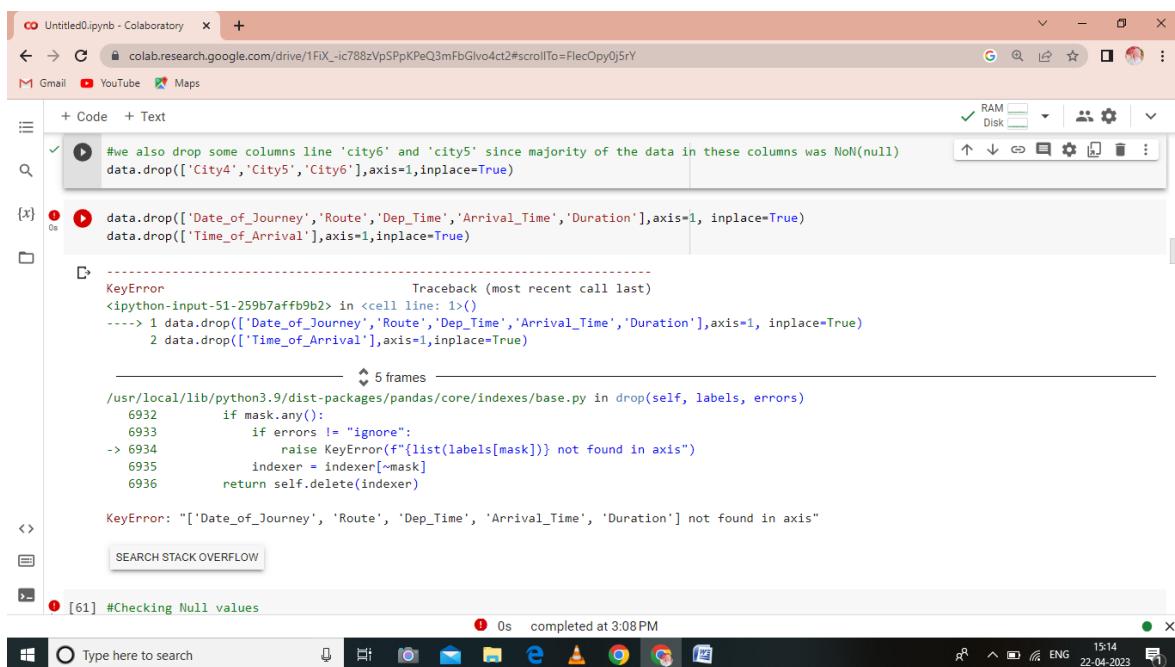
Dep_Time          0
Arrival_Time      0
Duration          0
Total_Stops        0
Additional_Info    0
Price              0
City1              0
City2            10682
City3            10682
City4            10682
City5            10682
City6            10682
Date                0
Month               0
Year                0
Dep_Time_Hour     0
Dep_Time_Mins      0
Arrival_date        0
Time_of_Arrival      0
Arrival_Time_Hour    0
Arrival_Time_Mins    10682
Travel_Hours        0
Travel_Mins         0
dtype: int64

```

`#we also drop some columns like 'city6' and 'city5' since majority of the data in these columns was NoN(null)`

`data.drop(['City4','City5','City6'],axis=1,inplace=True)`

0s completed



Untitled0.ipynb - Colaboratory

`[42]`

```

#we also drop some columns like 'city6' and 'city5' since majority of the data in these columns was NoN(null)
data.drop(['City4','City5','City6'],axis=1,inplace=True)

```

`{x}`

`[43]`

```

data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1, inplace=True)
data.drop(['Time_of_Arrival'],axis=1,inplace=True)

```

`KeyError`

```

Traceback (most recent call last)
<ipython-input-51-259b7affb9b2> in <cell line: 1>()
----> 1 data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1, inplace=True)
      2 data.drop(['Time_of_Arrival'],axis=1,inplace=True)

/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
       6932         if mask.any():
       6933             if errors != "ignore":
-> 6934                 raise KeyError(f"list({labels[mask]}) not found in axis")
       6935             indexer = indexer[~mask]
       6936             return self.delete(indexer)

KeyError: "['Date_of_Journey', 'Route', 'Dep_Time', 'Arrival_Time', 'Duration'] not found in axis"

```

`SEARCH STACK OVERFLOW`

`[61]` #Checking Null values

0s completed at 3:08PM

Project Report Template

Untitled0.ipynb - Colaboratory

```
[51]: [6934] raise KeyError(f"list({labels[mask]}) not found in axis")
       6935     indexer = indexer[~mask]
       6936     return self._delete(indexer)

{x} KeyError: "[‘Date_of_Journey’, ‘Route’, ‘Dep_Time’, ‘Arrival_Time’, ‘Duration’] not found in axis"

[61]: #Checking Null values
      data.isnull.sum()

-----  

AttributeError: Traceback (most recent call last)
<ipython-input-61-9394018b16a3> in <cell line: 2>()
      1 #Checking Null values
      2 data.isnull.sum()

      AttributeError: 'function' object has no attribute 'sum'

-----  

#filling City3 as None ,the missing values are less
data[‘City3’].fillna(‘None’,inplace=True)
```

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

Type here to search

Untitled0.ipynb - Colaboratory

```
[59]: #filling Arrival_Date as departure Date
      data[‘Arrival_date’].fillna(data[‘Date’],inplace=True)

{x} [60]: #filling Travel_Mins as zero(0)
      data[‘Travel_Mins’].fillna(0,inplace=True)

[65]: #Changing the numerical columns from object to int
      #data.Total_Stops=data.Total_Stops.astype(‘int64’)
      data.Date=data.Date.astype(‘int64’)
      data.Month=data.Month.astype(‘int64’)
      data.Year=data.Year.astype(‘int64’)
      data.Dep_Time_Hour=data.Dep_Time_Hour.astype(‘int64’)
      data.Dep_Time_Hour=data.Dep_Time_Hour.astype(‘int64’)
      data.Dep_Time_Mins=data.Dep_Time_Mins.astype(‘int64’)
      data.Arrival_date=data.Arrival_date.astype(‘int64’)
      data.Arrival_Time_Hour=data.Arrival_Time_Hour.astype(‘int64’)
      data.Arrival_Time_Mins=data.Arrival_Time_Mins.astype(‘int64’)
      #data.Travel_Hours=data.Travel_Hours.astype(‘int64’)
      data.Travel_Mins=data.Travel_Mins.astype(‘int64’)

-----  

0s completed at 3:08 PM
```

Type here to search

Project Report Template

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=wupl7s2Aye2W

Gmail YouTube Maps

```
+ Code + Text
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 10682
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Airline          10682 non-null   object 
 1   Source           10682 non-null   object 
 2   Destination      10682 non-null   object 
 3   Total_Stops      10682 non-null   object 
 4   Additional_Info  10682 non-null   object 
 5   Price            10682 non-null   int64  
 6   City1            10682 non-null   object 
 7   City2            0    non-null     float64
 8   City3            10682 non-null   object 
 9   Date             10682 non-null   int64  
 10  Month            10682 non-null   int64  
 11  Year             10682 non-null   int64  
 12  Dep_Time_Hour   10682 non-null   int64  
 13  Dep_Time_Mins   10682 non-null   int64  
 14  Arrival_date    10682 non-null   int64  
 15  Time_of_Arrival 10682 non-null   object 
 16  Arrival_Time_Hour 10682 non-null   object 
 17  Arrival_Time_Mins 0    non-null     float64
 18  Travel_Hours    10682 non-null   object 
 19  Travel_Mins     10682 non-null   object 
dtypes: float64(2), int64(7), object(11)
memory usage: 1.7+ MB
```

0s completed at 3:08 PM

Type here to search

RAM Disk

15:15 22-04-2023

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=4-8g7-2zDuo

Gmail YouTube Maps

```
+ Code + Text
```

```
[66] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 10682
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Total_Stops      10682 non-null   object 
 1   Additional_Info  10682 non-null   object 
 2   Price            10682 non-null   int64  
 3   City1            10682 non-null   object 
 4   City2            0    non-null     float64
 5   City3            10682 non-null   object 
 6   Date             10682 non-null   int64  
 7   Month            10682 non-null   int64  
 8   Year             10682 non-null   int64  
 9   Dep_Time_Hour   10682 non-null   int64  
 10  Dep_Time_Mins   10682 non-null   int64  
 11  Arrival_date    10682 non-null   int64  
 12  Time_of_Arrival 10682 non-null   object 
 13  Arrival_Time_Hour 10682 non-null   object 
 14  Arrival_Time_Mins 0    non-null     float64
 15  Travel_Hours    10682 non-null   object 
 16  Travel_Mins     10682 non-null   object 
dtypes: float64(2), int64(7), object(11)
memory usage: 1.7+ MB
```

```
data[data['Travel_Hours']=='5m']
```

```
Airline  Source  Destination  Total_Stops  Additional_Info  Price  City1  City2  City3  Date  Month  Year  Dep_Time_Hour  Dep_Time_Mins  Arrival_
```

```
[ ] data.drop(index=6474,inplace=True,axis=0)
```

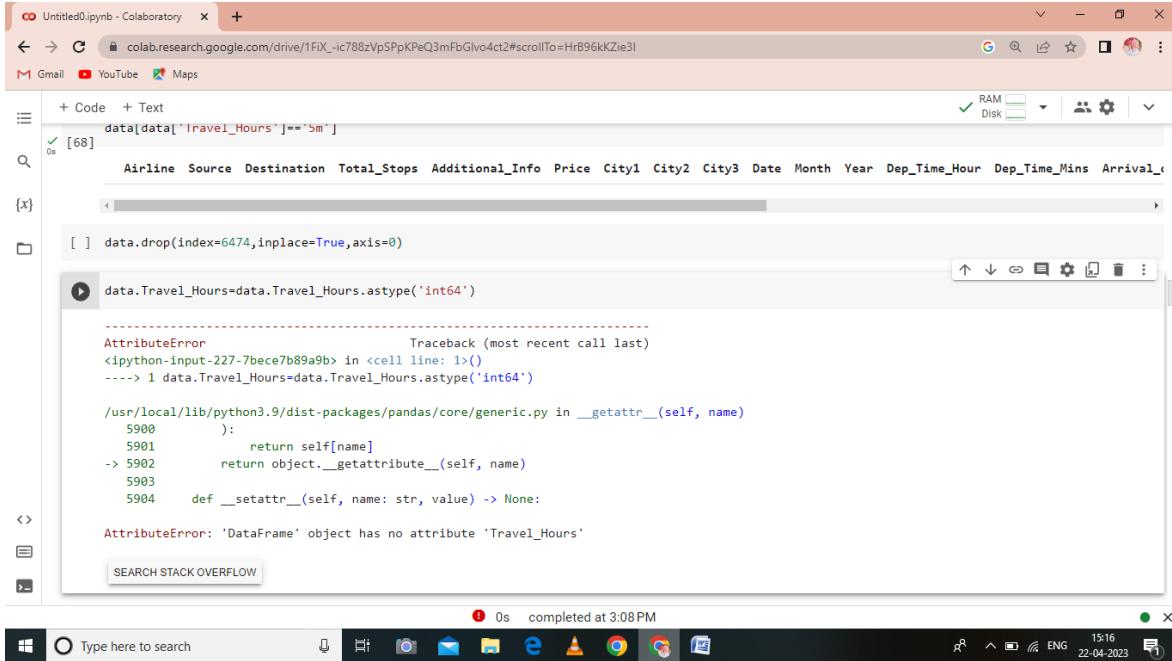
0s completed at 3:08 PM

Type here to search

RAM Disk

15:16 22-04-2023

Project Report Template



Untitled0.ipynb - Colaboratory

```
[68] In [ ] Out [ ]
```

```
data[data['Travel_Hours']=='5m']
```

```
[ ] data.drop(index=6474,inplace=True,axis=0)
```

```
data.Travel_Hours=data.Travel_Hours.astype('int64')
```

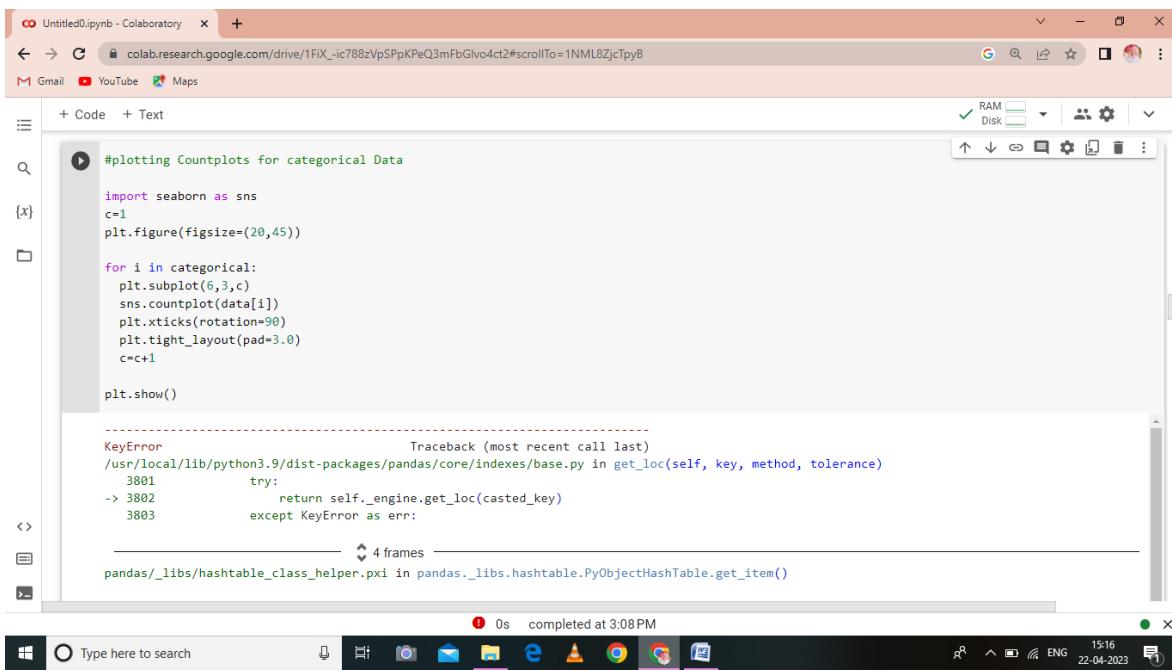
```
AttributeError: 'DataFrame' object has no attribute 'Travel_Hours'
```

```
SEARCH STACK OVERFLOW
```

0s completed at 3:08 PM

Type here to search

Windows taskbar: Type here to search, File, Camera, Mail, Edge, Chrome, File Explorer, Task View, 15:16, ENG, 22-04-2023



Untitled0.ipynb - Colaboratory

```
+ Code + Text
```

```
#plotting Countplots for categorical Data
```

```
import seaborn as sns
```

```
c=1
```

```
plt.figure(figsize=(20,45))
```

```
for i in categorical:
```

```
    plt.subplot(6,3,c)
```

```
    sns.countplot(data[i])
```

```
    plt.xticks(rotation=90)
```

```
    plt.tight_layout(pad=3.0)
```

```
c=c+1
```

```
plt.show()
```

```
KeyError: Traceback (most recent call last)
```

```
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
```

```
    try:
```

```
-> 3802         return self._engine.get_loc(casted_key)
```

```
    except KeyError as err:
```

```
        raise KeyError(f"None of {key} are in the index") from err
```

```
4 frames
```

```
pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

0s completed at 3:08 PM

Type here to search

Windows taskbar: Type here to search, File, Camera, Mail, Edge, Chrome, File Explorer, Task View, 15:16, ENG, 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory

```
#Distribution of 'PRICE' column
plt.figure(figsize=(15,8))
sns.distplot(data.price)

-----  

NameError: name 'sns' is not defined
SEARCH STACK OVERFLOW
<Figure size 1500x800 with 0 Axes>
```

0s completed at 3:08PM

Type here to search

Untitled0.ipynb - Colaboratory

```
data.columns
Index(['Airline', 'Source', 'Destination', 'Date', 'Month', 'Year',
       'Dep_Time_Hour', 'Dep_Time_Mins', 'Arrival_date', 'Arrival_Time_Hour',
       'Arrival_Time_Mins', 'Price'],
      dtype='object')

[73] #Checking the relation of price with categorical data
import seaborn as sns
c=1

for i in categorical:
    plt.figure(figsize = (10,20))

    plt.subplot(6,3,c)

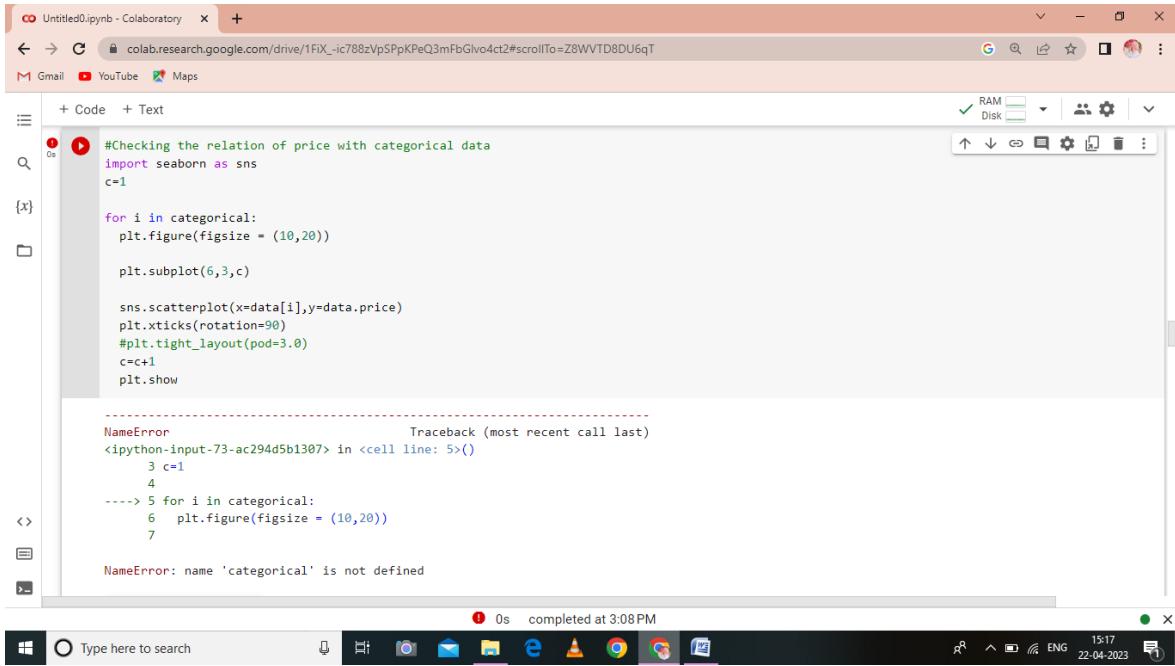
    sns.scatterplot(x=data[i],y=data.Price)
    plt.xticks(rotation=90)
    #plt.tight_layout(pad=3.0)
    c=c+1
    plt.show
```

NameError: name 'Price' is not defined
Traceback (most recent call last)
<ipython-input-73-ac294d5b1307> in <cell line: 5>()

0s completed at 3:08PM

Type here to search

Project Report Template



```
#Checking the relation of price with categorical data
import seaborn as sns
c=1

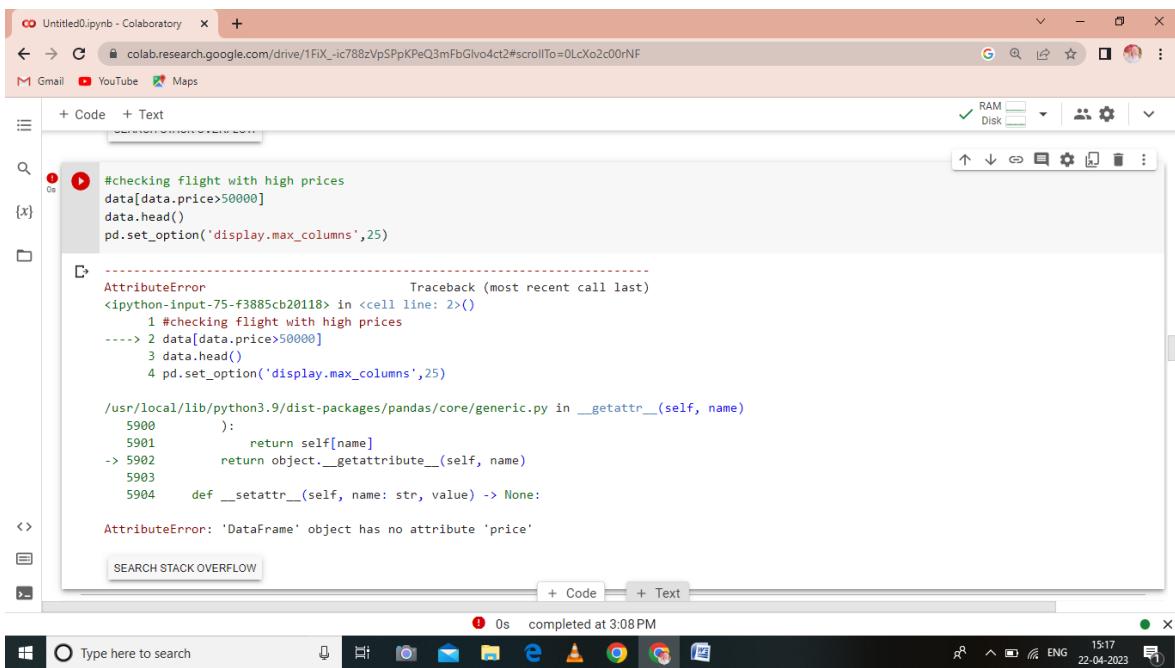
for i in categorical:
    plt.figure(figsize = (10,20))

    plt.subplot(6,3,c)

    sns.scatterplot(x=data[i],y=data.price)
    plt.xticks(rotation=90)
    #plt.tight_layout(pad=3.0)
    c+=1
    plt.show

-----
NameError: name 'categorical' is not defined
```

0s completed at 3:08 PM



```
#checking flight with high prices
data[data.price>50000]
data.head()
pd.set_option('display.max_columns',25)

-----
AttributeError: Traceback (most recent call last)
<ipython-input-75-f3885cb20118> in <cell line: 2>()
      1 #checking flight with high prices
----> 2 data[data.price>50000]
      3 data.head()
      4 pd.set_option('display.max_columns',25)

/usr/local/lib/python3.9/dist-packages/pandas/core/generic.py in __getattr__(self, name)
    5900     ):
    5901         return self[name]
-> 5902         return object.__getattribute__(self, name)
    5903
    5904     def __setattr__(self, name: str, value) -> None:
```

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

Project Report Template

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=CCQXdSS0197V

Gmail YouTube Maps

+ Code + Text

```
data.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arriva
0	IndiGo	Banglore	New Delhi		No info	3897	BLR → DEL	NaN	None	24	3	2019	22	20	
1	Air India	Kolkata	Banglore		No info	7662	CCU → IXR → BBI → BLR	NaN	None	1	5	2019	5	50	
2	Jet Airways	Delhi	Cochin		No info	13882	DEL → LKO → BOM → COK	NaN	None	9	6	2019	9	25	
3	IndiGo	Kolkata	Banglore		No info	6218	NAG → CCU	NaN	None	12	5	2019	18	5	

① 0s completed at 3:08 PM

Type here to search

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=rr_7HaFm1OLb

Gmail YouTube Maps

+ Code + Text

```
[76]
```

2	Jet Airways	Delhi	Cochin		No info	13882	BLR → DEL → LKO → BOM → COK	NaN	None	9	6	2019	9	25	
3	IndiGo	Kolkata	Banglore		No info	6218	CCU → NAG → BLR	NaN	None	12	5	2019	18	5	
4	IndiGo	Banglore	New Delhi		No info	13302	BLR → NAG → DEL	NaN	None	1	3	2019	16	50	

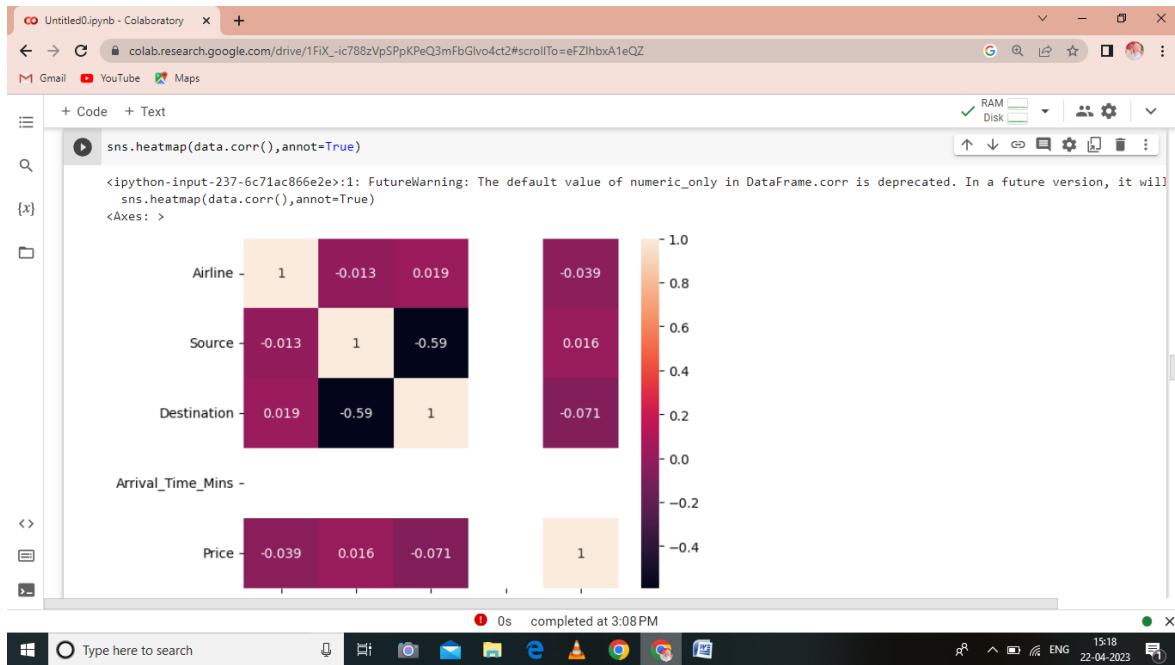
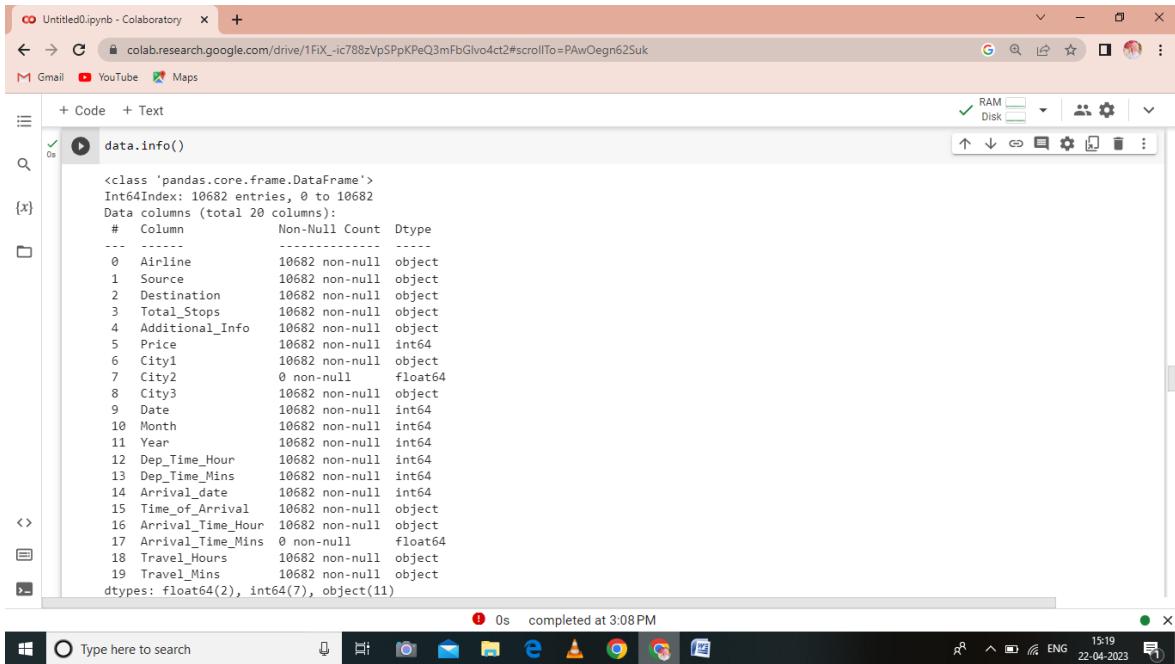
```
data['Year'].max()
```

```
'2019'
```

① 0s completed at 3:08 PM

Type here to search

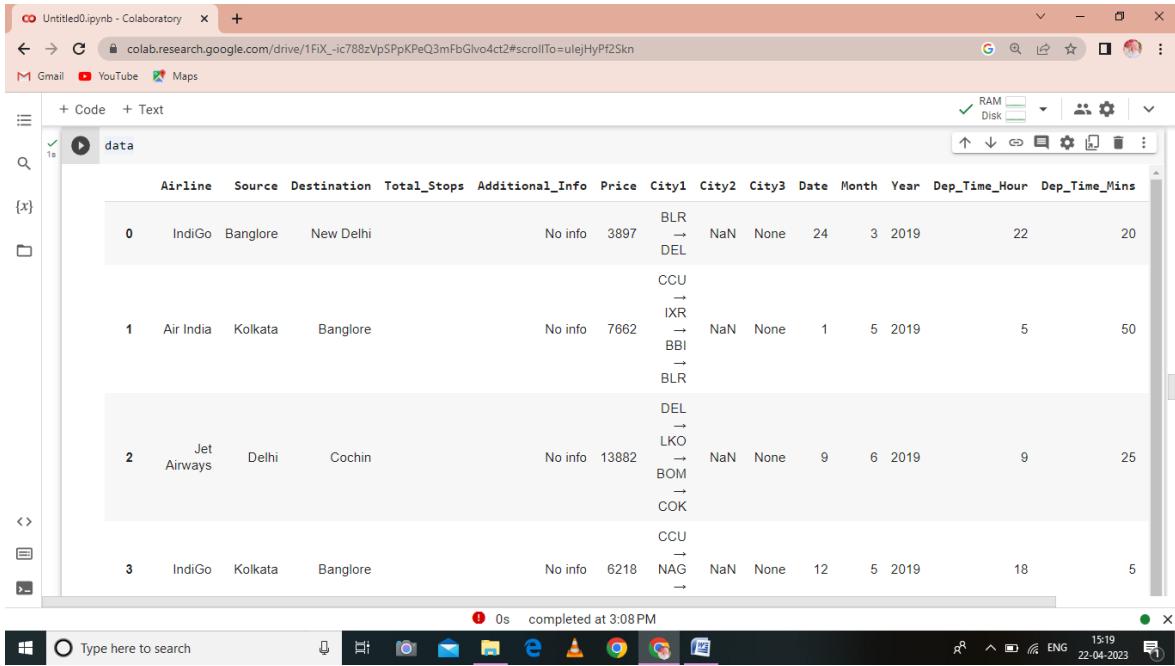
Project Report Template

```
data.info()

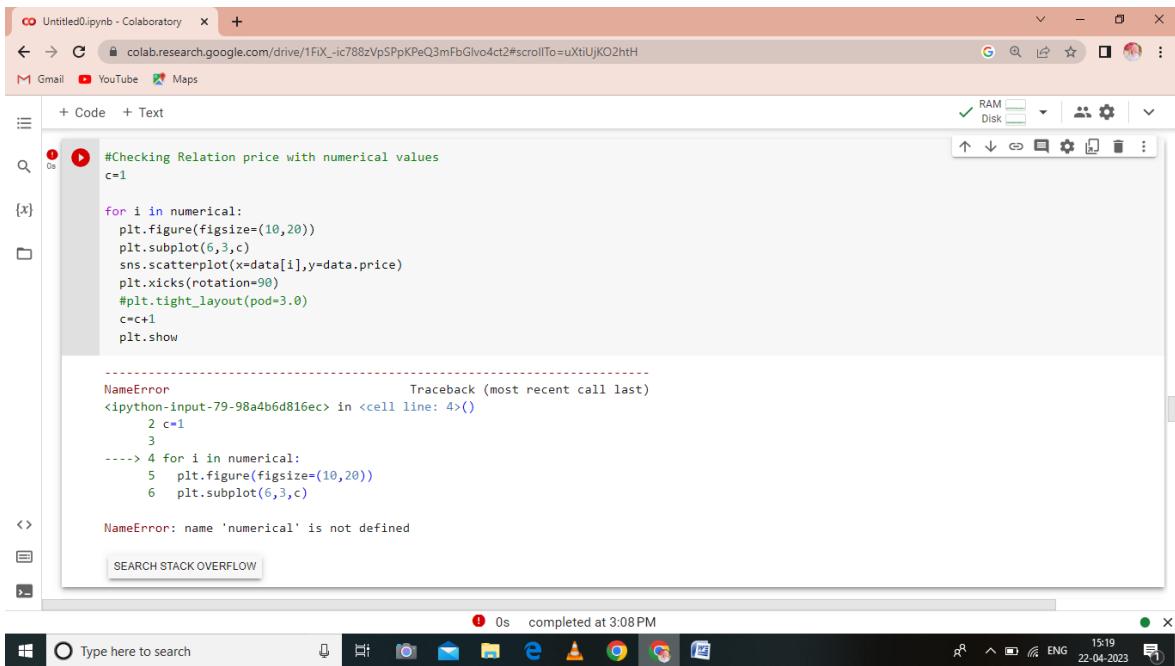
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 10682
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Airline          10682 non-null   object  
 1   Source           10682 non-null   object  
 2   Destination      10682 non-null   object  
 3   Total_Stops      10682 non-null   object  
 4   Additional_Info   10682 non-null   object  
 5   Price            10682 non-null   int64  
 6   City1            10682 non-null   object  
 7   City2            0 non-null       float64 
 8   City3            10682 non-null   object  
 9   Date             10682 non-null   int64  
 10  Month            10682 non-null   int64  
 11  Year             10682 non-null   int64  
 12  Dep_Time_Hour    10682 non-null   int64  
 13  Dep_Time_Mins    10682 non-null   int64  
 14  Arrival_date     10682 non-null   int64  
 15  Time_of_Arrival  10682 non-null   object  
 16  Arrival_Time_Hour 10682 non-null   object  
 17  Arrival_Time_Mins 0 non-null       float64 
 18  Travel_Hours     10682 non-null   object  
 19  Travel_Mins       10682 non-null   object  
dtypes: float64(2), int64(7), object(11)
```

Project Report Template



The screenshot shows a Jupyter Notebook interface in Google Colab. The code cell displays a Pandas DataFrame named 'data' containing flight information. The columns include Airline, Source, Destination, Total_Stops, Additional_Info, Price, City1, City2, City3, Date, Month, Year, Dep_Time_Hour, and Dep_Time_Mins. The data shows four flights from IndiGo, Air India, Jet Airways, and IndiGo, with various flight routes and prices.

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins
0	IndiGo	Banglore	New Delhi		No info	3897	BLR → DEL	NaN	None	24	3	2019	22	20
1	Air India	Kolkata	Banglore		No info	7662	CCU → IXR → BBI → BLR	NaN	None	1	5	2019	5	50
2	Jet Airways	Delhi	Cochin		No info	13882	DEL → LKO → BOM → COK	NaN	None	9	6	2019	9	25
3	IndiGo	Kolkata	Banglore		No info	6218	NAG →	NaN	None	12	5	2019	18	5



The screenshot shows a Jupyter Notebook interface in Google Colab. The code cell contains a script to plot a scatter plot of price versus numerical values. It uses plt.figure(figsize=(10,20)) and sns.scatterplot(x=data[1],y=data.price). The script encounters a NameError for 'numerical' because it was not defined earlier in the code. The error message is displayed below the code cell.

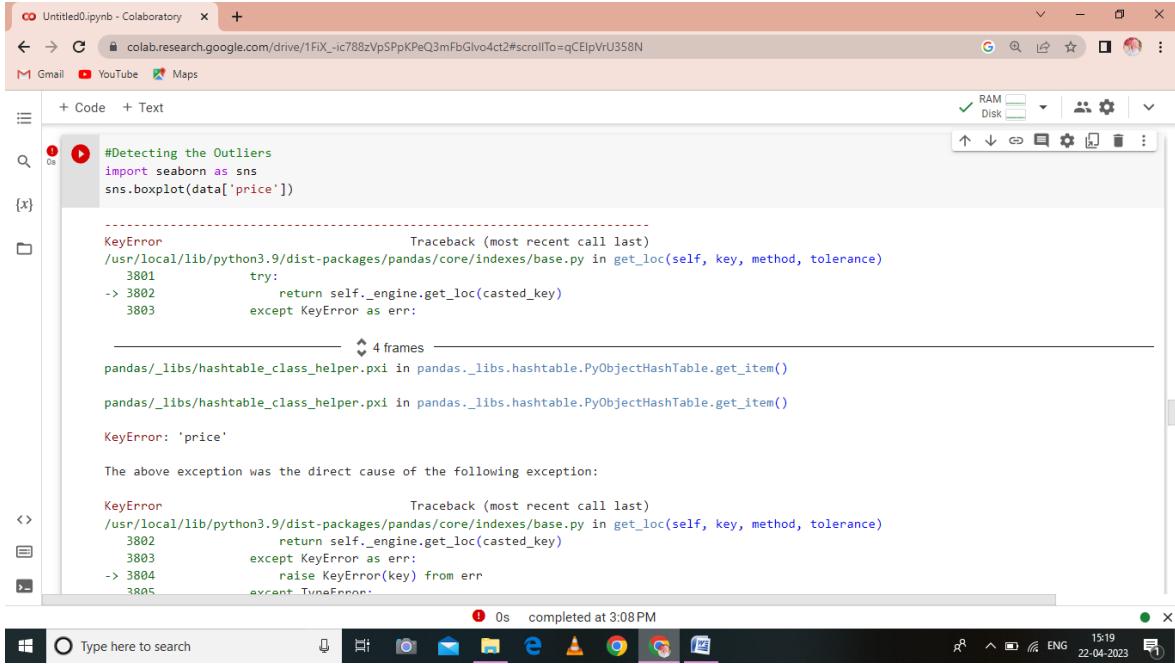
```
#Checking Relation price with numerical values
c=1

for i in numerical:
    plt.figure(figsize=(10,20))
    plt.subplot(6,3,c)
    sns.scatterplot(x=data[1],y=data.price)
    plt.xticks(rotation=90)
    #plt.tight_layout(pad=3.0)
    c+=1
    plt.show

-----
NameError: name 'numerical' is not defined

```

Project Report Template



```
#Detecting the Outliers
import seaborn as sns
sns.boxplot(data['price'])

-----
KeyError                                     Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3801         try:
    3802             return self._engine.get_loc(casted_key)
    3803         except KeyError as err:
    3804             raise KeyError(key) from err
    3805         except TypeErorr:
    3806             # If we have a listlike key, _check_indexing_error will raise

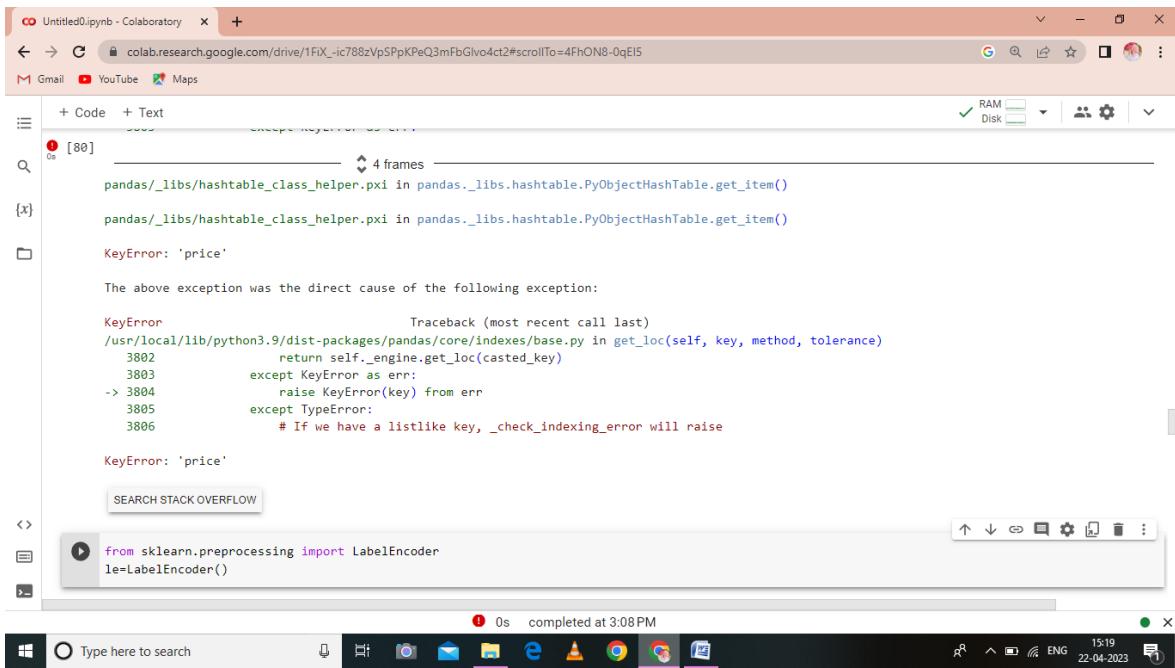
KeyError: 'price'

The above exception was the direct cause of the following exception:

KeyError                                     Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3802             return self._engine.get_loc(casted_key)
    3803         except KeyError as err:
    3804             raise KeyError(key) from err
    3805         except TypeErorr:
    3806             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'price'
```

① 0s completed at 3:08 PM



```
[80] _____
-----
KeyError                                     Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3802             return self._engine.get_loc(casted_key)
    3803         except KeyError as err:
    3804             raise KeyError(key) from err
    3805         except TypeErorr:
    3806             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'price'

SEARCH STACK OVERFLOW

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

① 0s completed at 3:08 PM

Project Report Template

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=YJfV-MqZqaYc

Gmail YouTube Maps

+ Code + Text

[80] **KeyError: 'price'**

SEARCH STACK OVERFLOW

{x}

```
[ ] from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

[ 0 ] data.Airline=le.fit_transform(data.Airline)
data.Source=le.fit_transform(data.Source)
data.Destination=le.fit_transform(data.Destination)
data.Total_Stops=le.fit_transform(data.Total_Stops)
data.City1=>le.fit_transform(data.City1->)
data.City2=>le.fit_transform(data.City2)
data.City3=>le.fit_transform(data.City3)
data.Additional_Info=>le.fit_transform(data.Additional_Info)
data.head()

File "<ipython-input-82-5aa9df7c1860>", line 5
    data.City1=>le.fit_transform(data.City1->
                                ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

Type here to search

RAM Disk

15:20 22-04-2023

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=D9jClweFLgyC

Gmail YouTube Maps

+ Code + Text

[0] data.head()

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arriva
0	IndiGo	Banglore	New Delhi		No info	3897	BLR → DEL	NaN	None	24	3	2019	22	20	
1	Air India	Kolkata	Banglore		No info	7662	CCU → IXR → BBI → BLR	NaN	None	1	5	2019	5	50	
2	Jet Airways	Delhi	Cochin		No info	13882	DEL → LKO → BOM → COK	NaN	None	9	6	2019	9	25	
3	IndiGo	Kolkata	Banglore		No info	6218	CCU → NAG →	NaN	None	12	5	2019	18	5	

0s completed at 3:08 PM

Type here to search

RAM Disk

15:20 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory + colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=wesqVnGCNUFe

RAM Disk ✓

```
[83]: 3 IndiGo Kolkata Banglore No info 6218 NAG →
      → BLR
{x} 4 IndiGo Banglore New Delhi No info 13302 NAG →
      → BLR → DEL
```

```
data=data[['Airline','Source','Destination','Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Hour','Arrival_Time_Mins','Price']
data.head()
```

	Airline	Source	Destination	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_Time_Mins	Price
0	3	0	5	24	03	2019	22	20	0			NaN 3897
1	1	3	0	1	05	2019	05	50	1			NaN 7662
2	4	2	1	9	06	2019	09	25	0			NaN 13882
3	3	3	0	12	05	2019	18	05	2			NaN 6218
4	3	0	5	01	03	2019	16	50	2			NaN 13302

0s completed at 3:08PM

Type here to search

15:20 ENG 22-04-2023

Untitled0.ipynb - Colaboratory + colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=5_GbiAZ76b1m

RAM Disk ✓

```
[ ]
```

	Airline	Source	Destination	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_Time_Mins	Price
0	3	0	5	24	03	2019	22	20	0			NaN 3897
1	1	3	0	1	05	2019	05	50	1			NaN 7662
2	4	2	1	9	06	2019	09	25	0			NaN 13882
3	3	3	0	12	05	2019	18	05	2			NaN 6218
4	3	0	5	01	03	2019	16	50	2			NaN 13302

```
y=data1['Price']
x=data1.drop(columns=['Price'],axis=1)
```

```
NameError: Traceback (most recent call last)
<ipython-input-88-4bf852500dd5> in <cell line: 1>()
----> 1 y=data1['Price']
      2 x=data1.drop(columns=['Price'],axis=1)

NameError: name 'data1' is not defined
```

SEARCH STACK OVERFLOW

0s completed at 3:08PM

Type here to search

15:20 ENG 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=JAga3SVJ79LP

Gmail YouTube Maps

+ Code + Text SEARCH STACK OVERFLOW

```
[89]: from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

File "<ipython-input-89-2363079cfedc>", line 2
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random-state=42)
                                         ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

SEARCH STACK OVERFLOW

```
[89]: x_train.head()

-----
NameError Traceback (most recent call last)
<ipython-input-90-3c4ecc6cd86a> in <cell line: 1>()
----> 1 x_train.head()

NameError: name 'x_train' is not defined
```

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

Type here to search

Windows Taskbar: Type here to search, File Explorer, Camera, Mail, Edge, File Explorer, Google Chrome, File Explorer

15:21 ENG 22-04-2023

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=Jh4YEoS8yjx

Gmail YouTube Maps

+ Code + Text

```
[89]: from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
for i in [rfr,gb,ad]:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    test_score=r2_score(y_test,y_pred)
    train_score=r2_score(y_train,i.predict(x_train))
    if abs(train_score-test_score)<=0.2:
        print(i)
        print("R2 score is",r2_score(y_test,y_pred))
        print("R2 for train data",r2_score(y_train,i.predict(x_train)))
        print("Mean Absolute Error is",mean_absolute_error(y_pred,y_test))
        print("Mean Squared Error is",mean_squared_error(y_pred,y_test))
        print("Root Mean Squared Error is",mean_absolute_error(y_pred,y_test,squared=False))
```

File "<ipython-input-93-614a9ab08cf1>", line 11
 print("Mean Absolute Error is",mean_absolute_error(y_pred,y_test))
 ^
SyntaxError: invalid syntax

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

Type here to search

Windows Taskbar: Type here to search, File Explorer, Camera, Mail, Edge, File Explorer, Google Chrome, File Explorer

15:21 ENG 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory

```

model=keras.Sequential()
model.add(Dense(7,activation='relu',input_dim=11))

model.add(Dense(7,activation='relu'))

model.add(Dense(1,activation='linear'))

model.summary()

Model: "sequential_1"
-----  

Layer (type)      Output Shape       Param #
-----  

dense_1 (Dense)   (None, 7)          84  

dense_2 (Dense)   (None, 7)          56  

dense_3 (Dense)   (None, 1)          8  

-----  

Total params: 148
Trainable params: 148
Non-trainable params: 0

```

0s completed at 3:08 PM

Untitled0.ipynb - Colaboratory

```

total params: 148
Trainable params: 148
Non-trainable params: 0

[107] model.compile(loss='mse',optimizer='rmsprop',metrics=['mae'])

File "<ipython-input-107-39d4500bbb91>", line 1
    model.compile(loss='mse',optimizer='rmsprop',metrics=['mae'])
                                         ^
SyntaxError: unexpected EOF while parsing

SEARCH STACK OVERFLOW

0s completed at 3:08 PM

```

```

model.fit(x_train,y_train,batch_size=20, epochs=10)

NameError: Traceback (most recent call last)
<ipython-input-108-04fc75443784> in <cell line: 1>()
----> 1 model.fit(x_train,y_train,batch_size=20, epochs=10)

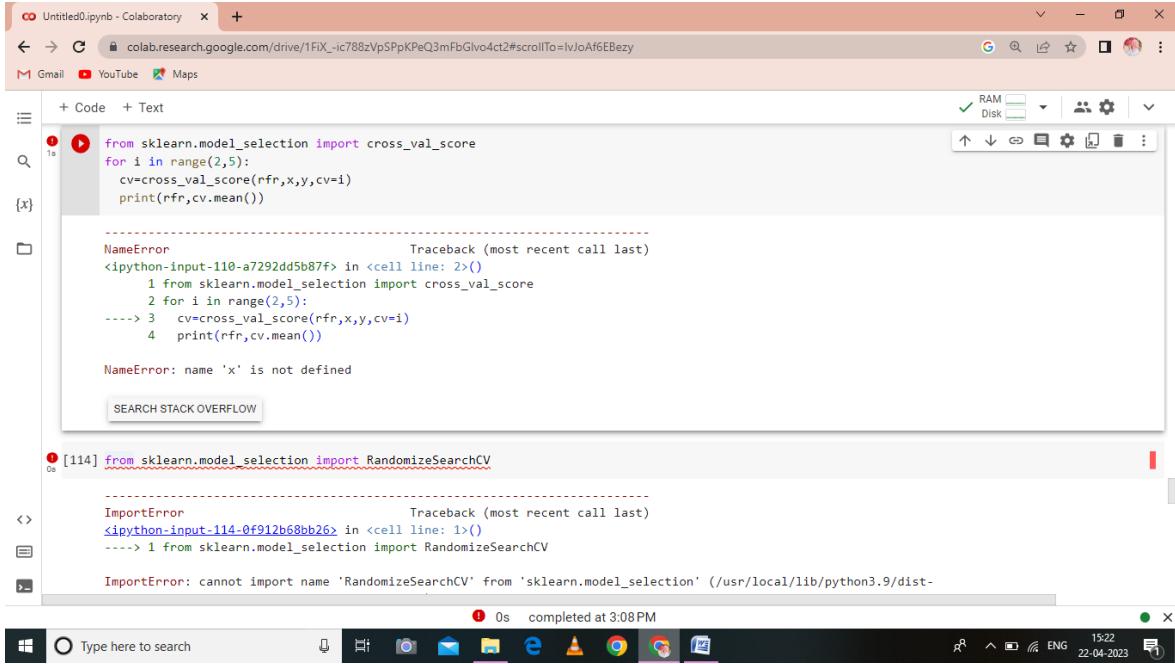
NameError: name 'x_train' is not defined

SEARCH STACK OVERFLOW

```

0s completed at 3:08 PM

Project Report Template



```

from sklearn.model_selection import cross_val_score
for i in range(2,5):
    cv=cross_val_score(rfr,x,y,cv=i)
    print(rfr, cv.mean())

```

Traceback (most recent call last)

```

<ipython-input-110-a7292dd5b87f> in <cell line: 2>()
      1 from sklearn.model_selection import cross_val_score
      2 for i in range(2,5):
--> 3     cv=cross_val_score(rfr,x,y,cv=i)
      4     print(rfr, cv.mean())

NameError: name 'x' is not defined

```

SEARCH STACK OVERFLOW

```

[114] from sklearn.model_selection import RandomizeSearchCV

```

ImportError Traceback (most recent call last)

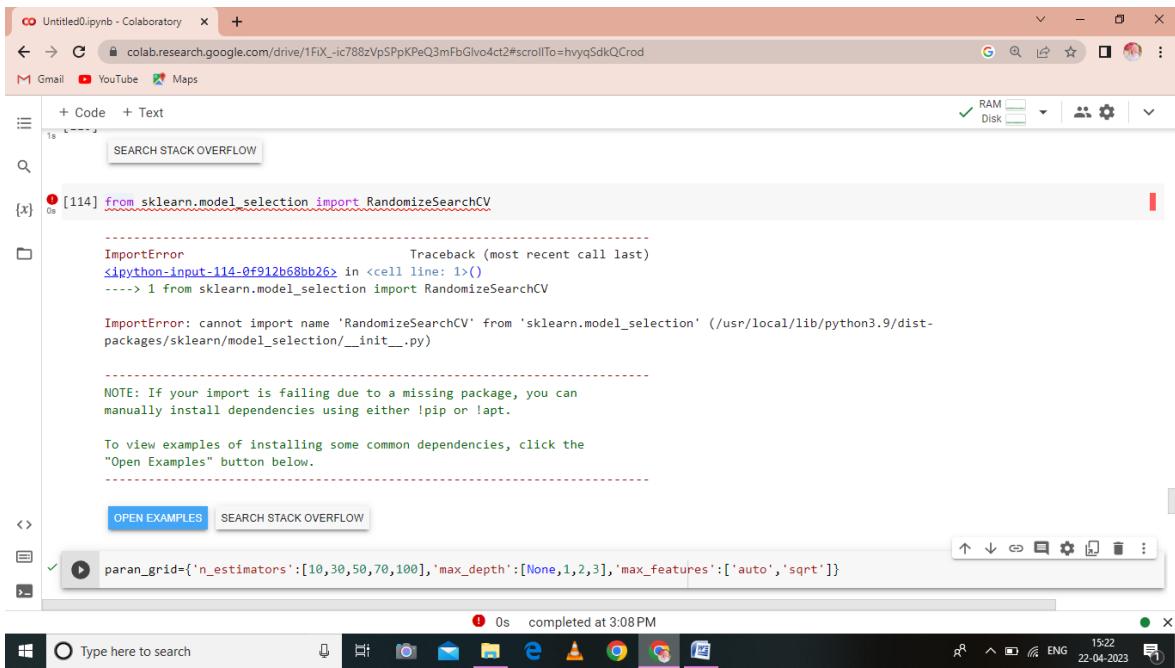
```

<ipython-input-114-0f912b68bb26> in <cell line: 1>()
--> 1 from sklearn.model_selection import RandomizeSearchCV

ImportError: cannot import name 'RandomizeSearchCV' from 'sklearn.model_selection' (/usr/local/lib/python3.9/dist-

```

0s completed at 3:08 PM



```

from sklearn.model_selection import RandomizeSearchCV

```

ImportError Traceback (most recent call last)

```

<ipython-input-114-0f912b68bb26> in <cell line: 1>()
--> 1 from sklearn.model_selection import RandomizeSearchCV

ImportError: cannot import name 'RandomizeSearchCV' from 'sklearn.model_selection' (/usr/local/lib/python3.9/dist-
packages/sklearn/model_selection/_init__.py)

NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.

```

OPEN EXAMPLES **SEARCH STACK OVERFLOW**

```

param_grid={'n_estimators':[10,30,50,70,100], 'max_depth':[None,1,2,3], 'max_features':['auto','sqrt']}

```

0s completed at 3:08 PM

Project Report Template

Untitled0.ipynb - Colaboratory

```

rfr=RandomForestRegressor()
rf_res=RandomizedSearchCV(estimator=rfr,param_distributions=param_grid,cv=3,verbose=2,n_jobs=-1)

rf_res.fit(x_train,y_train)

-----
NameError: Traceback (most recent call last)


0s completed at 3:08 PM


```

Untitled0.ipynb - Colaboratory

```

2 gb_res=RandomizedSearchCV(estimator=gb,param_distribution=param_grid,cv=3,verbose=2,n_jobs=-1)
3
4 gb_res.fit(x_train,y_train)

NameError: name 'gb_OradientBoostingRegressor' is not defined

SEARCH STACK OVERFLOW

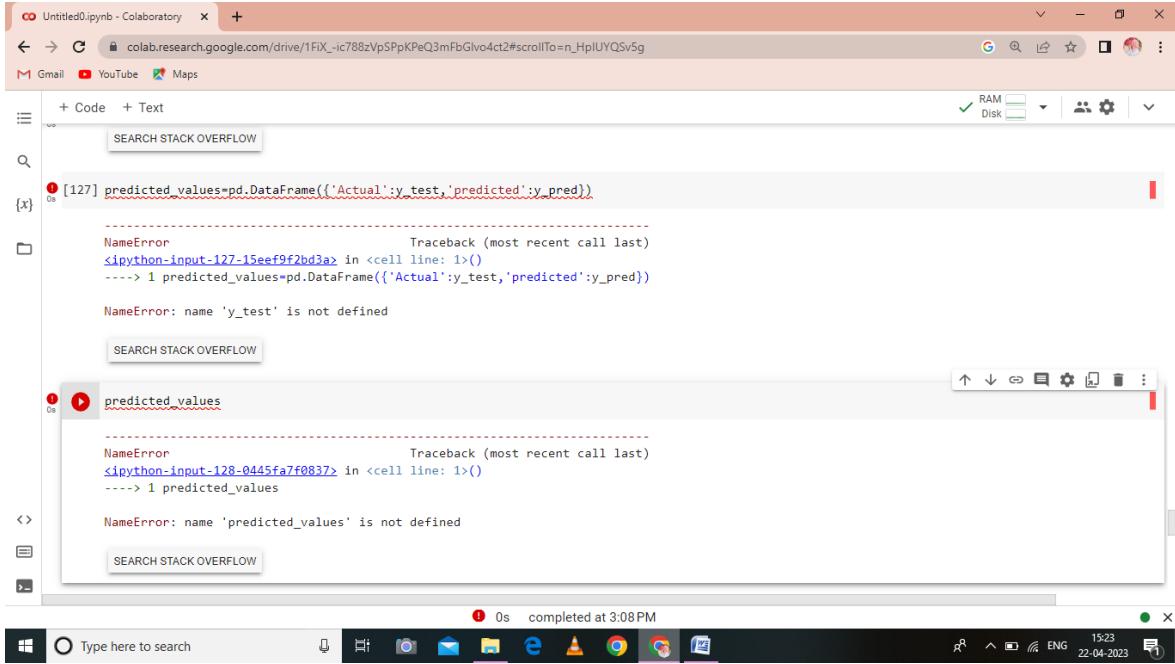
[124]
from sklearn.model_selection import cross_val_score
for i in range(2,5):
    cv=cross_val_score(gb,x,y,cv=i)
    print(rfr.cv.mean())

-----
NameError: Traceback (most recent call last)


0s completed at 3:08 PM


```

Project Report Template



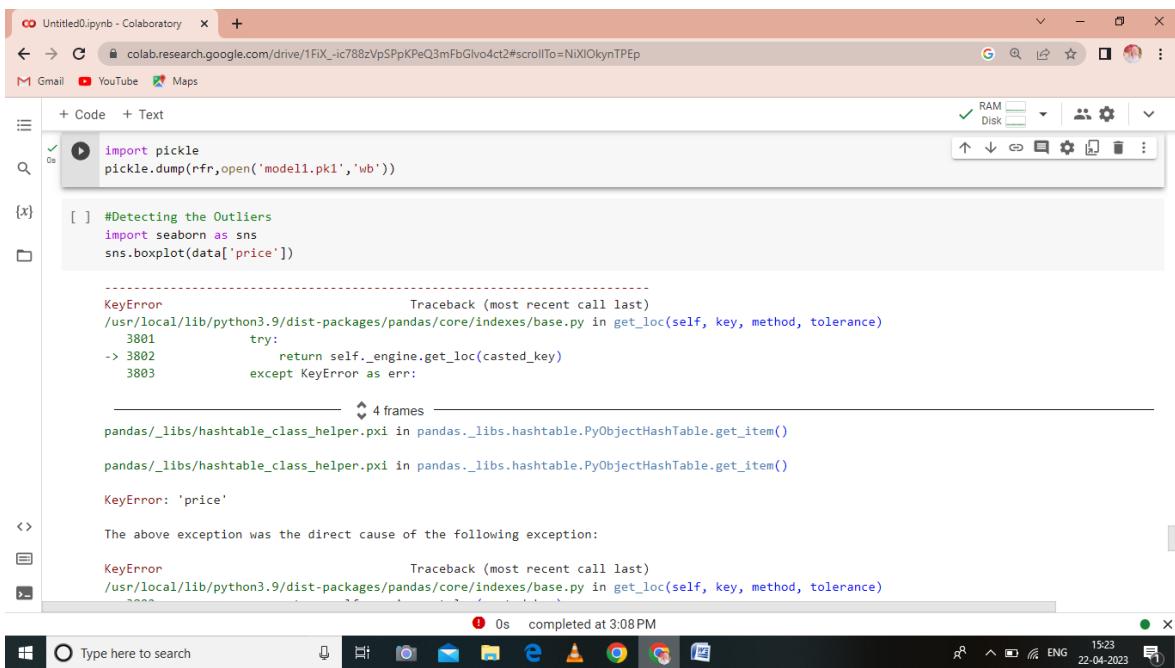
```
[127] predicted_values=pd.DataFrame({'Actual':y_test,'predicted':y_pred})

-----  

NameError          Traceback (most recent call last)
<ipython-input-127-15eef9f2bd3a> in <cell line: 1>()
----> 1 predicted_values=pd.DataFrame({'Actual':y_test,'predicted':y_pred})

NameError: name 'y_test' is not defined
```

The screenshot shows a Jupyter Notebook cell with the code `predicted_values=pd.DataFrame({'Actual':y_test,'predicted':y_pred})` and a resulting NameError. The error message indicates that the variable `y_test` is not defined. The notebook interface includes a search bar for Stack Overflow and a toolbar with various icons.



```
import pickle
pickle.dump(rfr,open('model1.pkl','wb'))
```

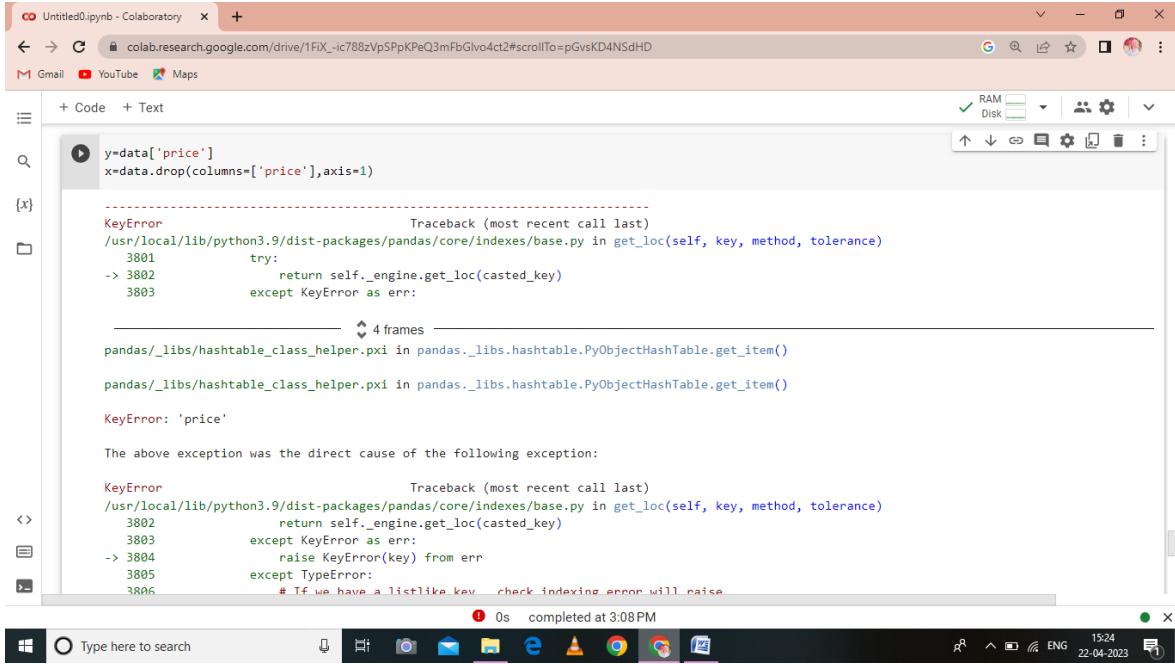
[] #Detecting the Outliers
import seaborn as sns
sns.boxplot(data['price'])

```
-----  

KeyError          Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3801      try:
    3802          return self._engine.get_loc(casted_key)
    3803      except KeyError as err:
    3804          raise KeyError(
    3805              f"KeyError: '{key}'\n\nThe above exception was the direct cause of the following exception:
```

The screenshot shows a Jupyter Notebook cell with the code `import pickle` and `pickle.dump(rfr,open('model1.pkl','wb'))`. It then attempts to run a cell containing `#Detecting the Outliers` and `sns.boxplot(data['price'])`. A KeyError is raised for the key 'price'. The error message provides a detailed traceback from the pandas library. The notebook interface includes a search bar for Stack Overflow and a toolbar with various icons.

Project Report Template



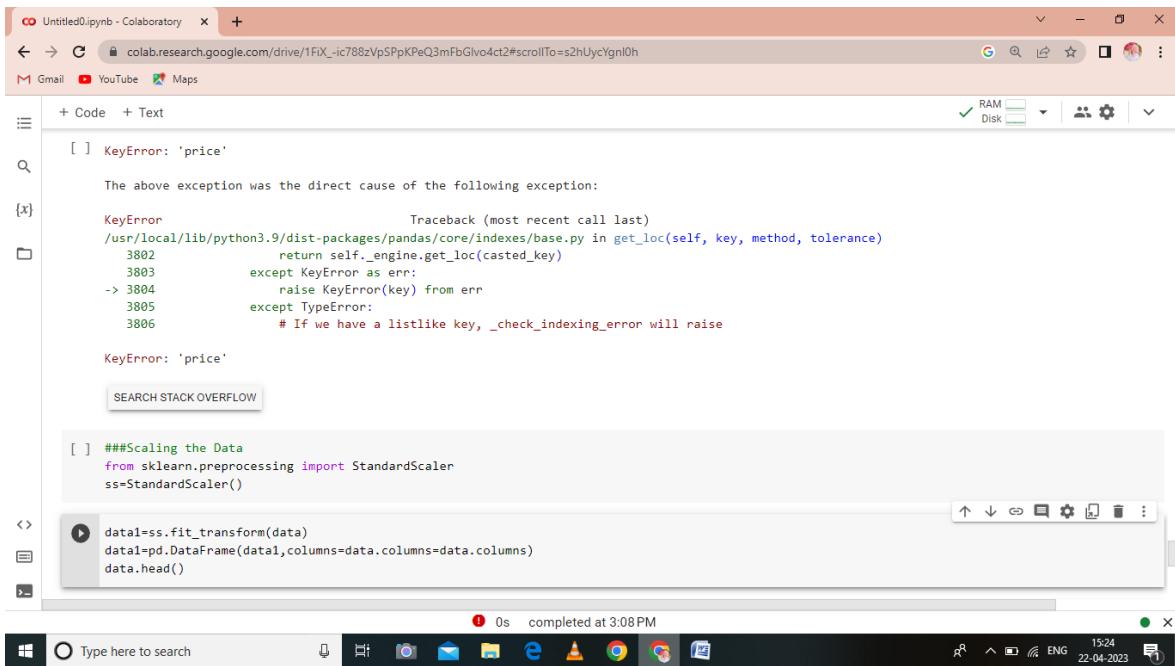
```

y=data['price']
x=data.drop(columns=['price'],axis=1)

-----
KeyError                                Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3801         try:
    3802             return self._engine.get_loc(casted_key)
    3803         except KeyError as err:
    3804             raise KeyError(key) from err
    3805         except TypeError:
    3806             # If we have a listlike key, _check_indexing_error will raise

```

0s completed at 3:08 PM



```

[ ] KeyError: 'price'

The above exception was the direct cause of the following exception:

KeyError                                Traceback (most recent call last)
/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
    3802         return self._engine.get_loc(casted_key)
    3803     except KeyError as err:
    3804         raise KeyError(key) from err
    3805     except TypeError:
    3806         # If we have a listlike key, _check_indexing_error will raise

KeyError: 'price'

SEARCH STACK OVERFLOW

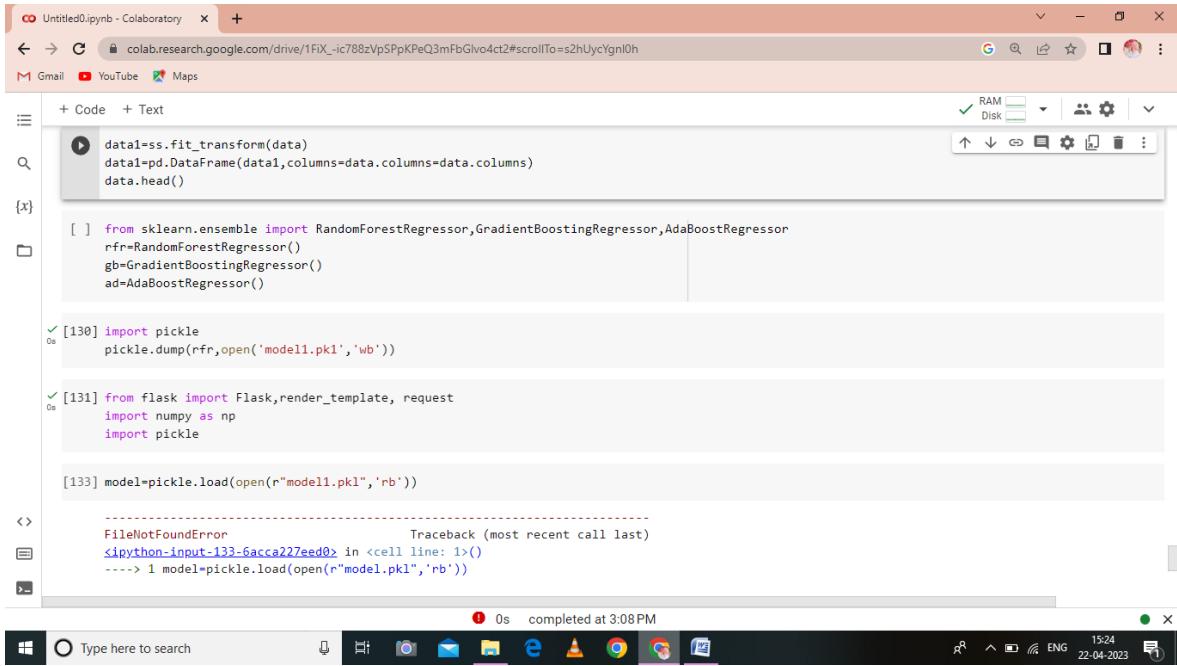
[ ] ##Scaling the Data
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()

[ ] data1=ss.fit_transform(data)
data1=pd.DataFrame(data1,columns=data.columns)
data1.head()

```

0s completed at 3:08 PM

Project Report Template



Untitled0.ipynb - Colaboratory

```

data=ss.fit_transform(data)
data=pd.DataFrame(data1,columns=data.columns)
data.head()

[ ] from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
rfr=RandomForestRegressor()
gb=GradientBoostingRegressor()
ad=AdaBoostRegressor()

[130] import pickle
pickle.dump(rfr,open('model1.pkl','wb'))

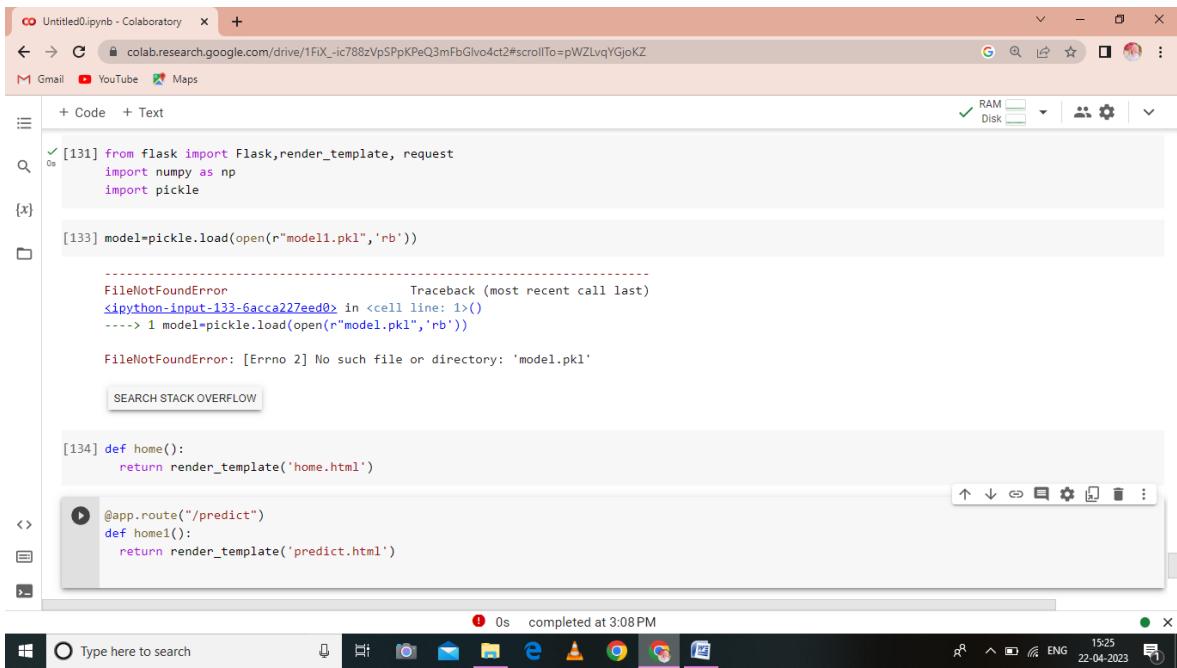
[131] from flask import Flask,render_template, request
import numpy as np
import pickle

[133] model=pickle.load(open(r"model1.pkl",'rb'))

```

FileNotFoundError Traceback (most recent call last)
`<ipython-input-133-6acca227eed0>` in <cell line: 1>()
----> 1 model=pickle.load(open(r"model.pkl",'rb'))

0s completed at 3:08 PM



Untitled0.ipynb - Colaboratory

```

[131] from flask import Flask,render_template, request
import numpy as np
import pickle

[133] model=pickle.load(open(r"model1.pkl",'rb'))

FileNotFoundError Traceback (most recent call last)
<ipython-input-133-6acca227eed0> in <cell line: 1>()
----> 1 model=pickle.load(open(r"model.pkl",'rb'))

FileNotFoundException: [Errno 2] No such file or directory: 'model.pkl'

SEARCH STACK OVERFLOW

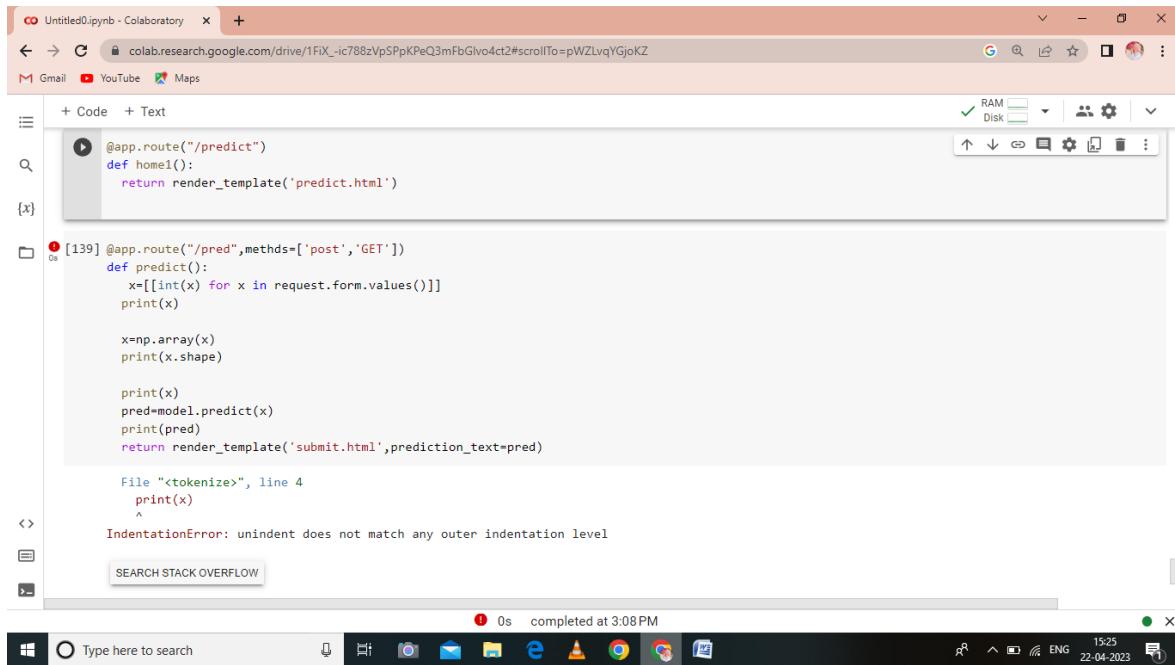
[134] def home():
    return render_template('home.html')

@app.route("/predict")
def home1():
    return render_template('predict.html')

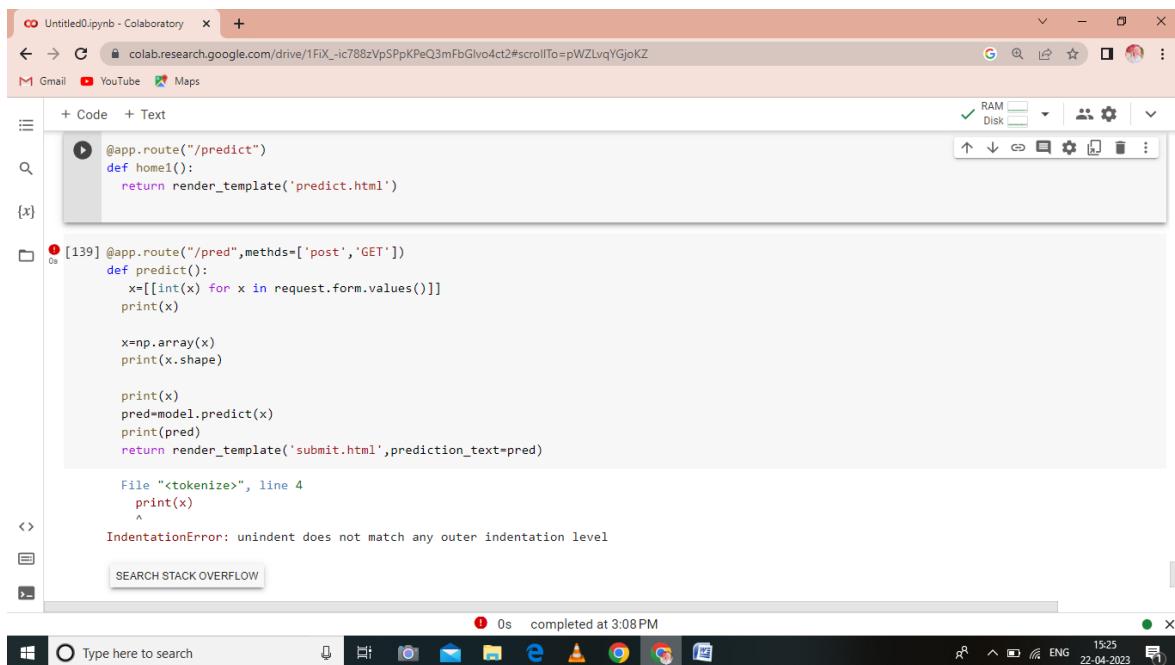
```

0s completed at 3:08 PM

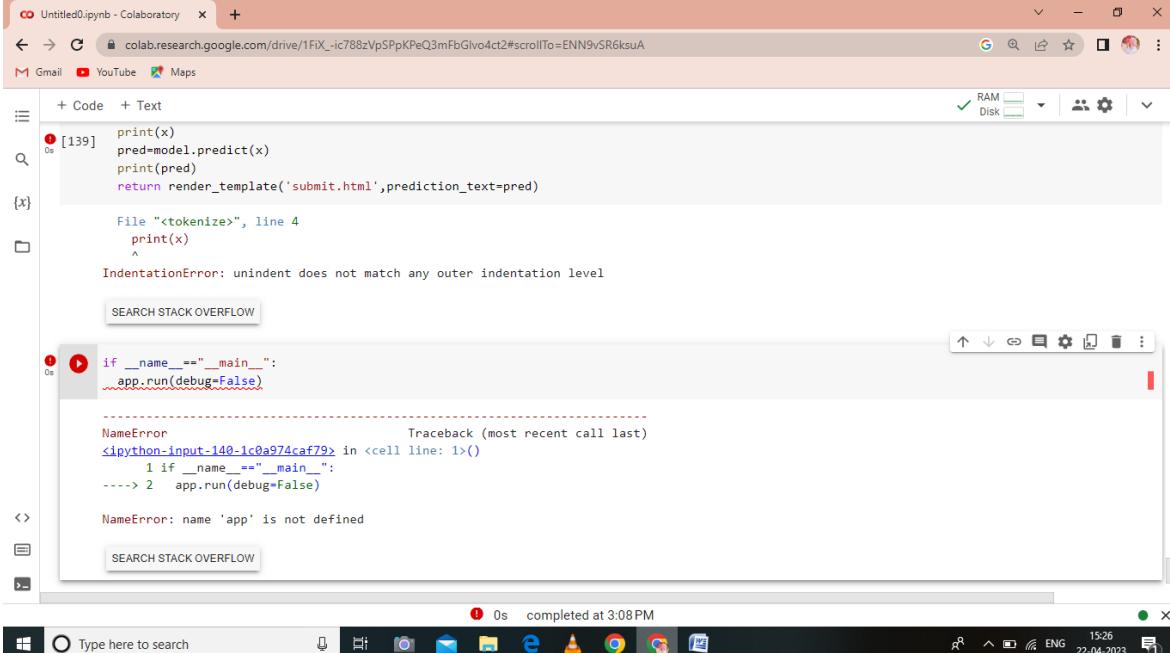
Project Report Template



```
Untitled0.ipynb - Colaboratory +  
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=pWZLqvYGjoKZ  
Gmail YouTube Maps  
+ Code + Text  
Q {x}  
@app.route("/predict")  
def home():  
    return render_template('predict.html')  
  
[139] @app.route("/pred",methods=['post','GET'])  
def predict():  
    x=[[int(x) for x in request.form.values()]]  
    print(x)  
  
    x=np.array(x)  
    print(x.shape)  
  
    print(x)  
    pred=model.predict(x)  
    print(pred)  
    return render_template('submit.html',prediction_text=pred)  
  
File "<tokenize>", line 4  
    print(x)  
          ^  
IndentationError: unindent does not match any outer indentation level  
SEARCH STACK OVERFLOW  
0s completed at 3:08 PM  
Type here to search 15:25 22-04-2023
```



```
Untitled0.ipynb - Colaboratory +  
colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGivo4ct2#scrollTo=pWZLqvYGjoKZ  
Gmail YouTube Maps  
+ Code + Text  
Q {x}  
@app.route("/predict")  
def home():  
    return render_template('predict.html')  
  
[139] @app.route("/pred",methods=['post','GET'])  
def predict():  
    x=[[int(x) for x in request.form.values()]]  
    print(x)  
  
    x=np.array(x)  
    print(x.shape)  
  
    print(x)  
    pred=model.predict(x)  
    print(pred)  
    return render_template('submit.html',prediction_text=pred)  
  
File "<tokenize>", line 4  
    print(x)  
          ^  
IndentationError: unindent does not match any outer indentation level  
SEARCH STACK OVERFLOW  
0s completed at 3:08 PM  
Type here to search 15:25 22-04-2023
```



The screenshot shows a Google Colab notebook titled "Untitled0.ipynb - Colaboratory". The code in cell [139] is as follows:

```
print(x)
pred=model.predict(x)
print(pred)
return render_template('submit.html',prediction_text=pred)

File "<tokenize>", line 4
    print(x)
        ^
IndentationError: unindent does not match any outer indentation level
```

The error message indicates an indentation issue at line 4. Below the code, a search bar says "SEARCH STACK OVERFLOW". The stack trace shows:

```
NameError: name 'app' is not defined
Traceback (most recent call last)
<ipython-input-140-1c0a974caf79> in <cell line: 1>()
      1 if __name__=="__main__":
      2     app.run(debug=False)

NameError: name 'app' is not defined
```

At the bottom of the window, there is a taskbar with icons for file operations and system status.

ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

- ❖ **Cancelation and Changes of Travel**
- ❖ **Early Check-Ins**
- ❖ **Maximize Efficiency**
- ❖ **Special Amenities or Deals**
- ❖ **Time Saver and Less Hectic**

1. Cancelation and Changes of Travel

It is normal for travelers to cancel or change their online reservations. They will not have to call their airlines and wait for a service representative to give them a hand.

Booking a flight online means you may do everything provided have internet access. Some airlines offer bookings without cancellation fees, while others provide online cancellations and changes for free, as long as you do it within 24 hours of departure.

2. Early Check-Ins

If you want to fly within the country, you might check-in in advance and as late as one hour before taking your flight. On the other hand, **international flight booking** allows travelers to check in between three and 24 hours before taking their flights. All you need to do is visit the site of your preferred airline and find your reservation so that you can enter your details. Main airlines also provide mobile check-in options, where travelers will receive a bar code on their phones.

3. Maximize Efficiency

There are a lot of aspects of booking online, which have been automated. Time is valuable, and all the time you spend unnecessarily to put information into booking flights can be spent assisting customers in the queue.

4. Special Amenities or Deals

From time to time, airlines provide loyalty bonuses and other additional benefits for online booking. These may range from free Wi-Fi, beverage credits, and food to additional miles and special packages.

These benefits and deals will motivate you to book directly since they might not be available through third parties.

5. Time Saver and Less Hectic

The internet has everything, so you may easily create packages from scratch based on your personal preferences, dates, and budget. You will also find affordable flights ready tailored to concert tickets, booking hours, and museums. Hence, there will be no need to reach out to a representative and wait to receive feedback. Online booking allows you to reserve a seat any time of the day.

DISADVANTAGES

- If you make a mistake, while you're booking, for example spell your name wrong, you're done, your ticket is not valid and you are charged.
- There is usually nobody you can turn to, in case of a problem with the flight.
- If you want to claim, after the trip, or something went wrong with the trip, your claim is kind of in the air. Nobody is really responsible. The airline itself: Where did you buy the ticket? Ok. claim there. But there is nobody to address to. Yeah, that's YOUR problem.
- Where you bought your ticket? There is nobody at the airport representing the company who sold you the ticket.
- There are about 4000 booking sites and engines. Hardly any of them inform about travel conditions, for example visa and other restrictions. Most call centers and all agents do that.

5. APPLICATION

In this section, we will be building a web application that is integrated to the model we built.

A UI is provided for the user where he has to enter the values for predictions.

The entered values are given to the saved model and prediction is displayed on the UI.

This section has the following tasks:

- Building server side script
- Building HTML pages
- Run the web application

6. CONCLUSION

Justify the creation of a national research program focused on the needs of airport operators;

Reveal how such a program can play a role in helping airport operators meet the many demands of federal agencies, state governments, local communities, and airport users;

Provide guidance on governing, funding, and administering an airport research program.

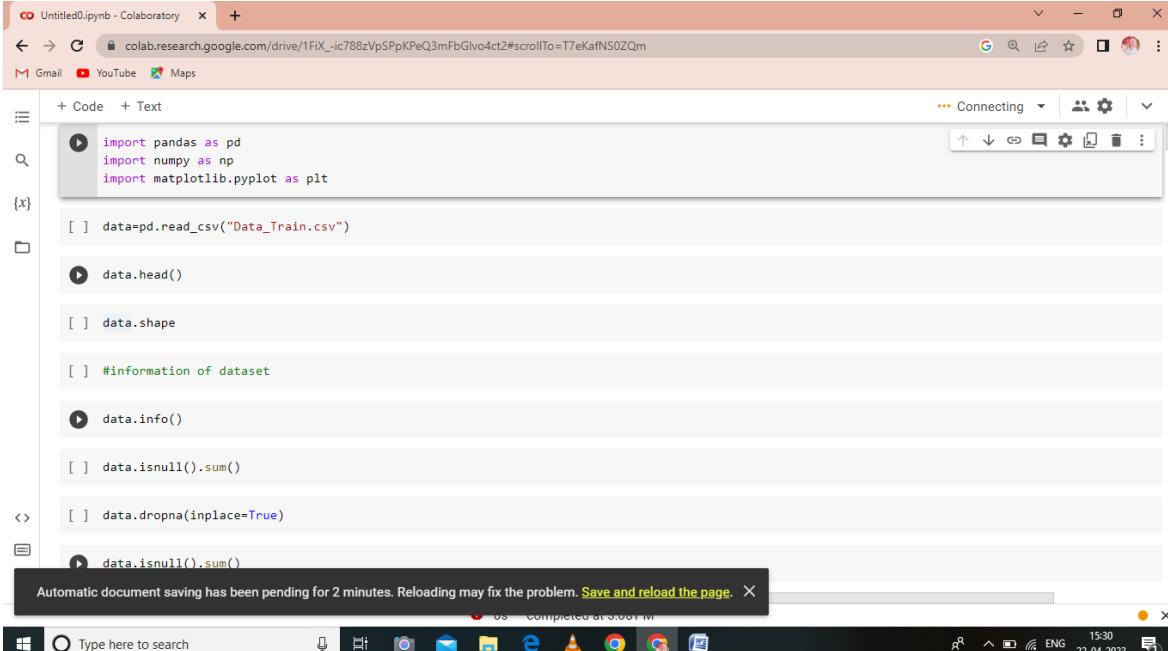
7. FUTURE SCOPE

Emerging technologies are reshaping with robotics, artificial intelligence, the internet of things, unmanned aircraft systems and the push for hybrid and electric airplanes just to name a few.

Alternative fuels can significantly change the current scenario of aviation in support of the environmental protection.

8. APPENDIX

A. Source Code



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data=pd.read_csv("Data_Train.csv")

data.head()

data.shape

#information of dataset

data.info()

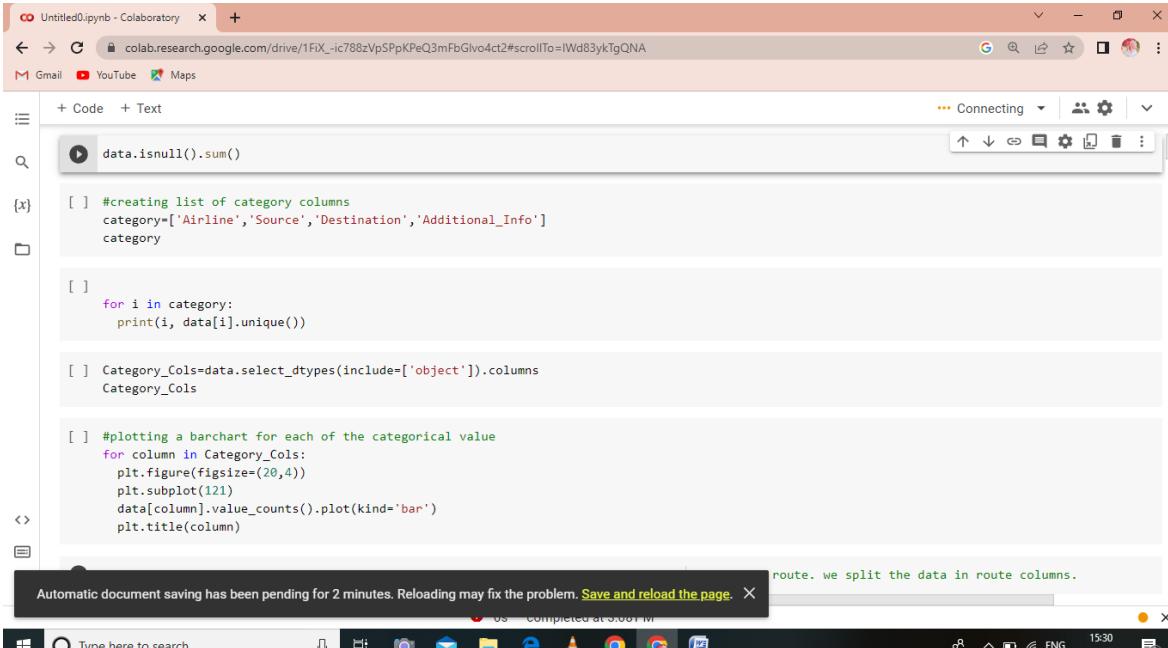
data.isnull().sum()

data.dropna(inplace=True)

data.isnull().sum()

```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#)



```

data.isnull().sum()

#creating list of category columns
category=['Airline','Source','Destination','Additional_Info']
category

for i in category:
    print(i, data[i].unique())

Category_Cols=data.select_dtypes(include=['object']).columns
Category_Cols

#plotting a barchart for each of the categorical value
for column in Category_Cols:
    plt.figure(figsize=(20,4))
    plt.subplot(121)
    data[column].value_counts().plot(kind='bar')
    plt.title(column)

```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#)

Project Report Template

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=tLrq9AqExjYW

Gmail YouTube Maps

Connecting

```

for i in category:
    print(i, data[i].unique())

Category_Cols=data.select_dtypes(include=['object']).columns
Category_Cols

#plotting a barchart for each of the categorical value
for column in Category_Cols:
    plt.figure(figsize=(20,4))
    plt.subplot(121)
    data[column].value_counts().plot(kind='bar')
    plt.title(column)

#Since the maximum number of stops is 4, there should be maximum 6 cities in any particular route. we split the data in route columns.
data.Route=data.Route.str.split('->')
data.Route

[ ] data['City1']=data.Route.str[0]
data['City2']=data.Route.str[1]
data['City3']=data.Route.str[2]
data['City4']=data.Route.str[3]

```

Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. [Save and reload the page.](#) X

0s completed at 3:00 PM

Type here to search

15:30 ENG 22-04-2023

Untitled0.ipynb - Colaboratory +

https://colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=yTb04s2xpKnj

Gmail YouTube Maps

Connecting

```

data['City1']=data.Route.str[0]
data['City2']=data.Route.str[1]
data['City3']=data.Route.str[2]
data['City4']=data.Route.str[3]
data['City5']=data.Route.str[4]
data['City6']=data.Route.str[5]

#we now split the Date column to extract the 'Date','Month' and 'Year'values ,and stored them in new columns in our dataframe
data.Date_of_Journey=data.Date_of_Journey.str.split('/')
data.Date_of_Journey

#Treating the data_column
data['Date']=data.Date_of_Journey.str[0]
data['Month']=data.Date_of_Journey.str[1]
data['Year']=data.Date_of_Journey.str[2]

#In the similiar manner, we split the Dep_Time column, and create separate columns for departure hours and minutes
data.Dep_Time=data.Dep_Time.str.split(':')

data['Dep_Time_Hour']=data.Dep_Time.str[0]
data['Dep_Time_Mins']=data.Dep_Time.str[1]

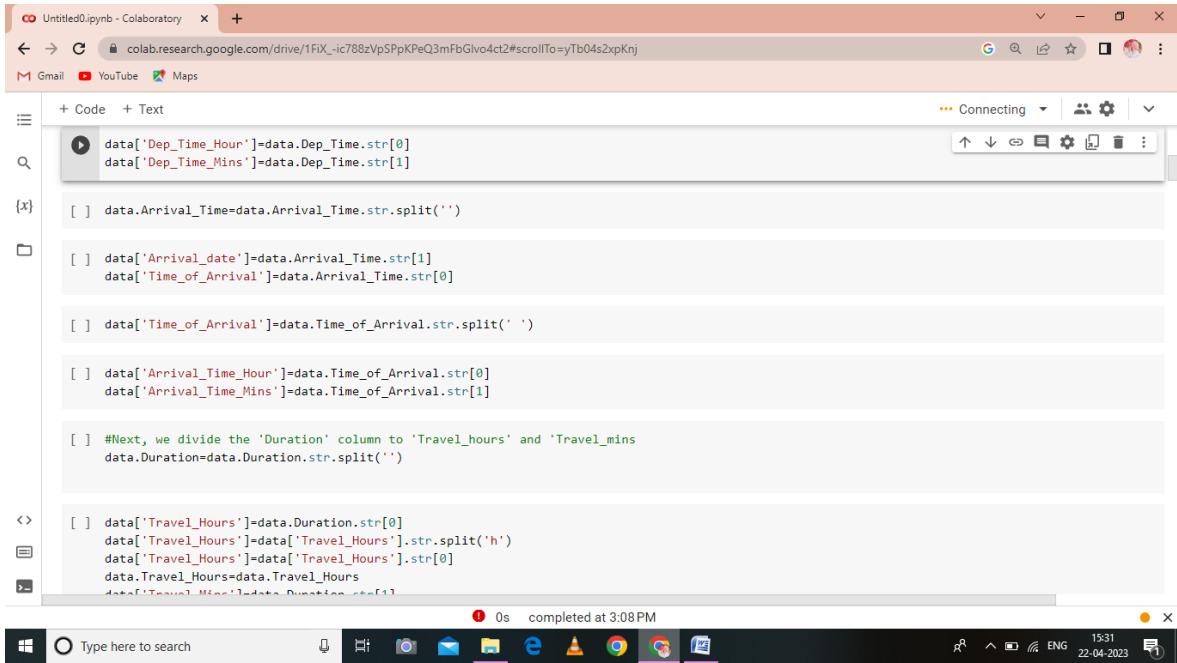
```

0s completed at 3:08PM

Type here to search

15:31 ENG 22-04-2023

Project Report Template



```

data['Dep_Time_Hour']=data.Dep_Time.str[0]
data['Dep_Time_Mins']=data.Dep_Time.str[1]

data.Arrival_Time=data.Arrival_Time.str.split('')

data['Arrival_date']=data.Arrival_Time.str[1]
data['Time_of_Arrival']=data.Arrival_Time.str[0]

data['Time_of_Arrival']=data.Time_of_Arrival.str.split(' ')

data['Arrival_Time_Hour']=data.Time_of_Arrival.str[0]
data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]

#Next, we divide the 'Duration' column to 'Travel_hours' and 'Travel_mins'
data.Duration=data.Duration.str.split('')

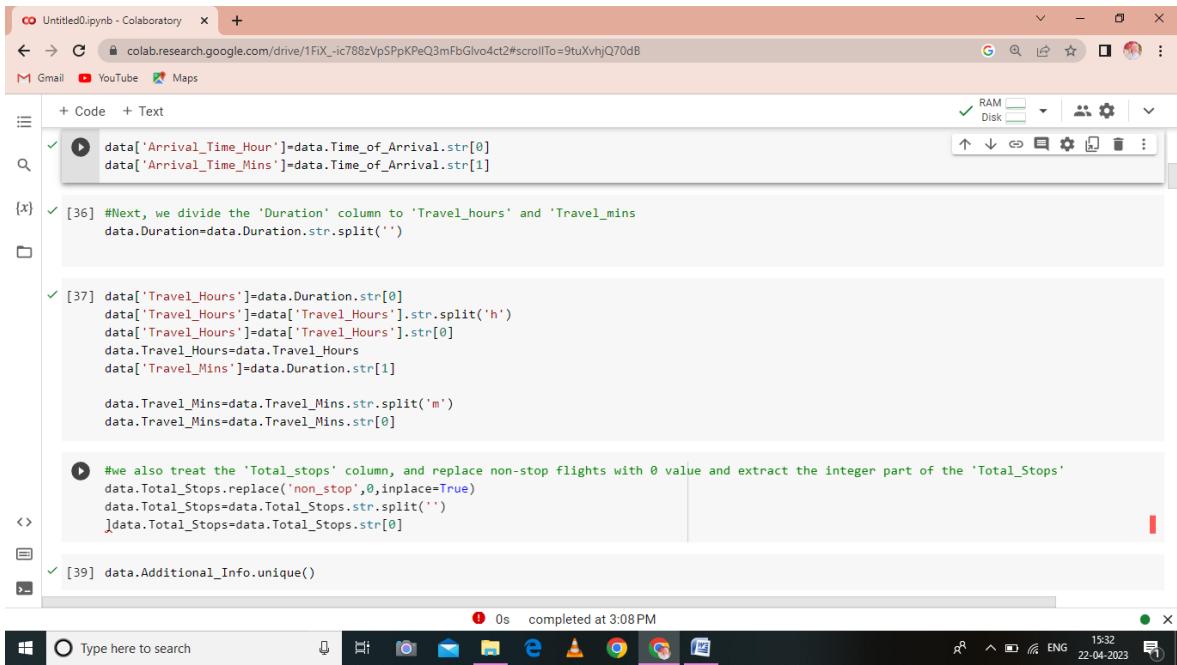
data['Travel_Hours']=data.Duration.str[0]
data['Travel_Hours']=data['Travel_Hours'].str.split('h')
data['Travel_Hours']=data['Travel_Hours'].str[0]
data.Travel_Hours=data.Travel_Hours
data['Travel_Mins']=data.Duration.str[1]

```

0s completed at 3:08 PM

Type here to search

RAM Disk ENG 22-04-2023



```

data['Arrival_Time_Hour']=data.Time_of_Arrival.str[0]
data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]

#Next, we divide the 'Duration' column to 'Travel_hours' and 'Travel_mins'
data.Duration=data.Duration.str.split('')

data['Travel_Hours']=data.Duration.str[0]
data['Travel_Hours']=data['Travel_Hours'].str.split('h')
data['Travel_Hours']=data['Travel_Hours'].str[0]
data.Travel_Hours=data.Travel_Hours
data['Travel_Mins']=data.Duration.str[1]

data.Travel_Mins=data.Travel_Mins.str.split('m')
data.Travel_Mins=data.Travel_Mins.str[0]

#we also treat the 'Total_stops' column, and replace non-stop flights with 0 value and extract the integer part of the 'Total_Stops'
data.Total_Stops.replace('non_stop',0,inplace=True)
data.Total_Stops=data.Total_Stops.str.split('')
data.Total_Stops=data.Total_Stops.str[0]

data.Additional_Info.unique()

```

0s completed at 3:08 PM

Type here to search

RAM Disk ENG 22-04-2023

Project Report Template

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=V6Wc-2VynOCU

Gmail YouTube Maps

+ Code + Text

RAM Disk

```

[41] data.isnull().sum()

{x}
[42] #we also drop some columns like 'city6' and 'city5' since majority of the data in these columns was NaN(null)
    data.drop(['City4','City5','City6'],axis=1,inplace=True)

[43] data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1, inplace=True)
    data.drop(['Time_of_Arrival'],axis=1,inplace=True)

[44] #Checking Null values
    data.isnull.sum()

[45] #filling City3 as None ,the missing values are less
    data['City3'].fillna('None',inplace=True)

[46] #filling Arrival_Date as departure Date
    data['Arrival_date'].fillna(data['Date'],inplace=True)

[47] #filling Travel_Mins as zero(0)
    data['Travel_Mins'].fillna(0,inplace=True)

```

0s completed at 3:08PM

Type here to search

15:33 22-04-2023

Untitled0.ipynb - Colaboratory +

colab.research.google.com/drive/1FiX_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=kRJOEkvtwxE

Gmail YouTube Maps

+ Code + Text

RAM Disk

```

[48] #filling Travel_Mins as zero(0)
    data['Travel_Mins'].fillna(0,inplace=True)

{x}
[49] #Changing the numerical columns from object to int
    #data.Total_Stops=data.Total_Stops.astype('int64')
    data.Date=data.Date.astype('int64')
    data.Month=data.Month.astype('int64')
    data.Year=data.Year.astype('int64')
    data.Dep_Time_Hour=data.Dep_Time_Hour.astype('int64')
    data.Dep_Time_Hour=data.Dep_Time_Hour.astype('int64')
    data.Dep_Time_Mins=data.Dep_Time_Mins.astype('int64')
    data.Arrival_date=data.Arrival_date.astype('int64')
    data.Arrival_Time_Hour=data.Arrival_Time_Hour.astype('int64')
    data.Arrival_Time_Mins=data.Arrival_Time_Mins.astype('int64')
    #data.Travel_Hours=data.Travel_Hours.astype('int64')
    data.Travel_Mins=data.Travel_Mins.astype('int64')

[50] data.info()
Run cell (Ctrl+Enter)
cell executed since last change
executed by Irfana Marulvan
12:40 PM (2 hours ago)
executed in 0.734s
data[data['Travel_Hours']=='5m']

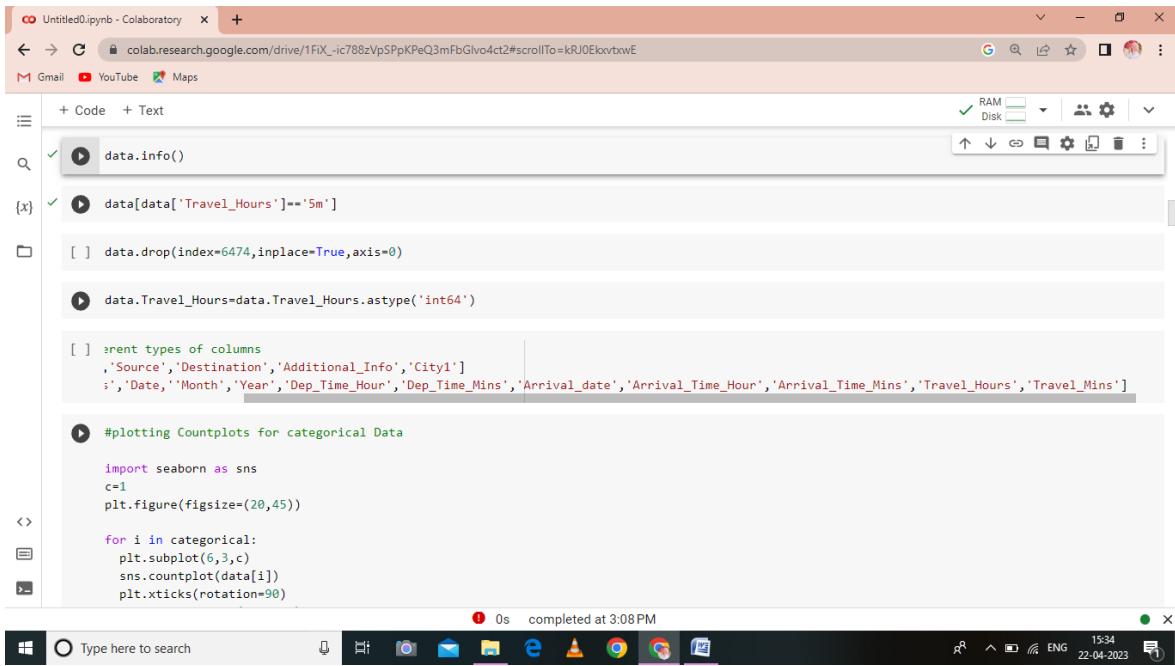
```

0s completed at 3:08PM

Type here to search

15:33 22-04-2023

Project Report Template



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=kRJ0EovtxwE
Gmail YouTube Maps

+ Code + Text
✓ RAM Disk
RAM Disk
data.info()

data[data['Travel_Hours']=='5m']

data.drop(index=6474,inplace=True,axis=0)

data.Travel_Hours=data.Travel_Hours.astype('int64')

#rent types of columns
,'Source','Destination','Additional_Info','City1']
;,'Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Hour','Arrival_Time_Mins','Travel_Hours','Travel_Mins']

#plotting Countplots for categorical Data

import seaborn as sns
c=1
plt.figure(figsize=(20,45))

for i in categorical:
    plt.subplot(6,3,c)
    sns.countplot(data[i])
    plt.xticks(rotation=90)
    c=c+1

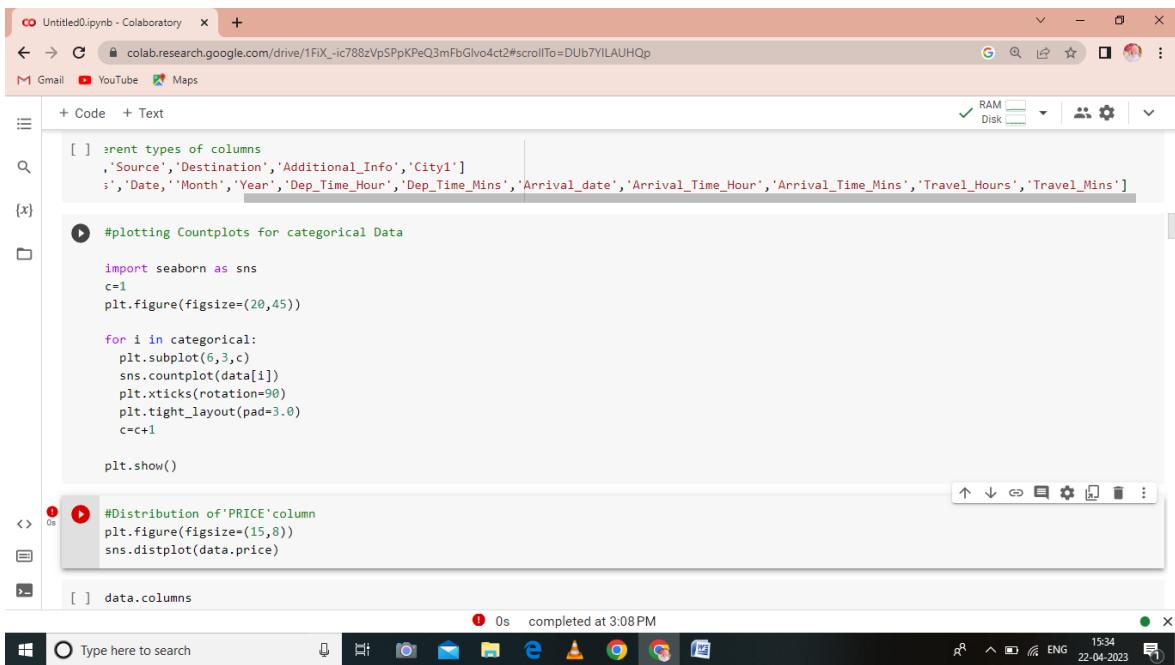
plt.show()

```

0s completed at 3:08 PM

Type here to search

15:34 22-04-2023



```

Untitled0.ipynb - Colaboratory + 
colab.research.google.com/drive/1FIx_-ic788zVpSPpKPeQ3mFbGlvo4ct2#scrollTo=DUb7YLAUHQp
Gmail YouTube Maps

+ Code + Text
✓ RAM Disk
RAM Disk
#rent types of columns
,'Source','Destination','Additional_Info','City1']
;,'Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Hour','Arrival_Time_Mins','Travel_Hours','Travel_Mins']

#plotting Countplots for categorical Data

import seaborn as sns
c=1
plt.figure(figsize=(20,45))

for i in categorical:
    plt.subplot(6,3,c)
    sns.countplot(data[i])
    plt.xticks(rotation=90)
    plt.tight_layout(pad=3.0)
    c=c+1

plt.show()

#Distribution of 'PRICE' column
plt.figure(figsize=(15,8))
sns.distplot(data.price)

data.columns

```

0s completed at 3:08 PM

Type here to search

15:34 22-04-2023



The screenshot shows a Jupyter Notebook interface in Google Colab. The notebook has two cells:

```
#Checking the relation of price with categorical data
import seaborn as sns
c=1

for i in categorical:
    plt.figure(figsize = (10,20))

    plt.subplot(6,3,c)

    sns.scatterplot(x=data[i],y=data.price)
    plt.xticks(rotation=90)
    #plt.tight_layout(pad=3.0)
    c+=1
    plt.show

#checking flight with high prices
data[data.price>50000]
data.head()
pd.set_option('display.max_columns',25)
```

The first cell contains code for visualizing the relationship between categorical variables and price. The second cell contains code for filtering flights with high prices and displaying the top few rows of the dataset.

Below the notebook, the terminal shows the command `data['Year'].max()`.

The screenshot shows a Jupyter Notebook interface in Google Colab. The notebook has several cells:

```
[ ] data['Year'].max()

[ ] sns.heatmap(data.corr(),annot=True)

[✓] [77] data.info()

[✓] [78] data

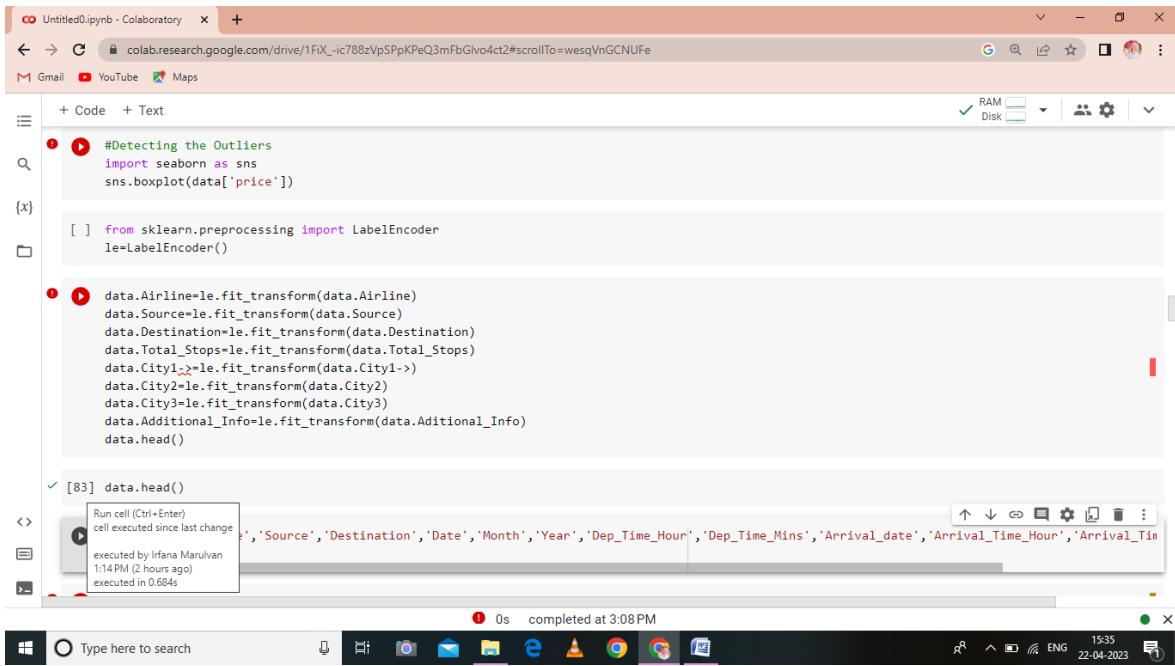
#Checking Relation price with numerical values
c=1

for i in numerical:
    plt.figure(figsize=(10,20))
    plt.subplot(6,3,c)
    sns.scatterplot(x=data[i],y=data.price)
    plt.xticks(rotation=90)
    #plt.tight_layout(pad=3.0)
    c+=1
    plt.show
```

The first cell shows the maximum value in the 'Year' column. The second cell generates a heatmap of the correlation matrix. The third cell displays the dataset's information. The fourth cell contains code for visualizing the relationship between numerical variables and price.

Below the notebook, the terminal shows the command `[80] #Detecting the Outliers`.

Project Report Template



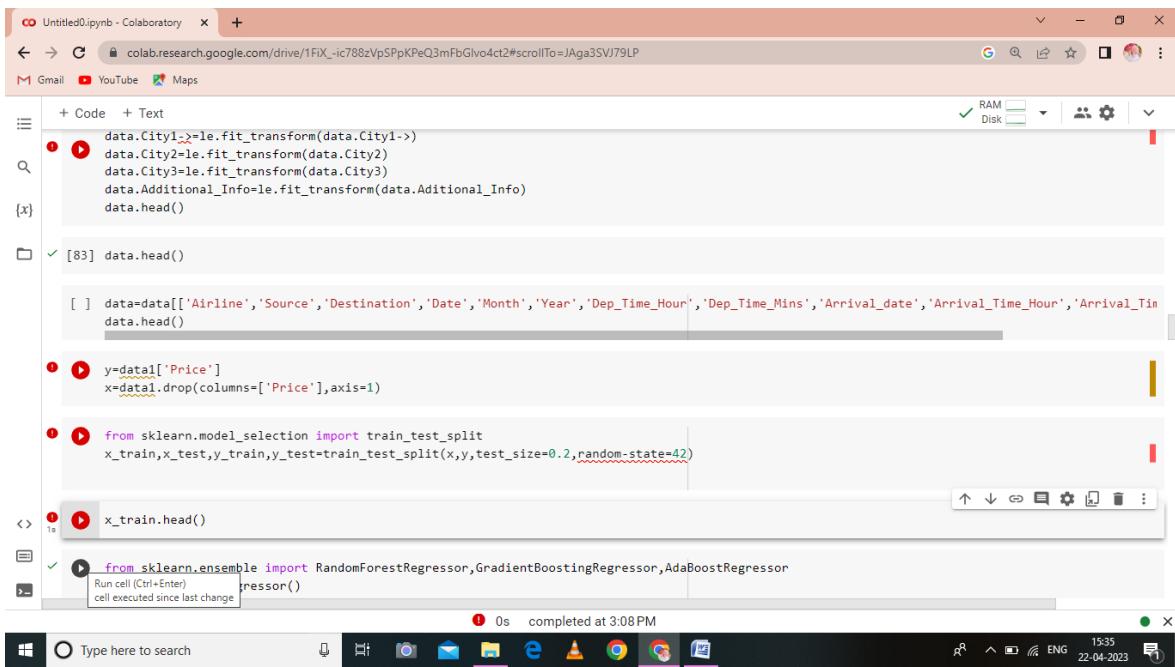
```
#Detecting the Outliers
import seaborn as sns
sns.boxplot(data['price'])

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

data.Airline=le.fit_transform(data.Airline)
data.Source=le.fit_transform(data.Source)
data.Destination=le.fit_transform(data.Destination)
data.Total_Stops=le.fit_transform(data.Total_Stops)
data.City1=>le.fit_transform(data.City1->)
data.City2=le.fit_transform(data.City2)
data.City3=le.fit_transform(data.City3)
data.Additional_Info=le.fit_transform(data.Additional_Info)
data.head()

[83] data.head()
Run cell (Ctrl+Enter)
cell executed since last change
executed by Irfana Marulvan
1:14 PM (2 hours ago)
executed in 0.684s
```

0s completed at 3:08 PM



```
data.City1=>le.fit_transform(data.City1->)
data.City2=le.fit_transform(data.City2)
data.City3=le.fit_transform(data.City3)
data.Additional_Info=le.fit_transform(data.Additional_Info)
data.head()

[83] data.head()
data=data[['Airline','Source','Destination','Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Hour','Arrival_Tin
data.head()

y=data['Price']
x=data.drop(columns=['Price'],axis=1)

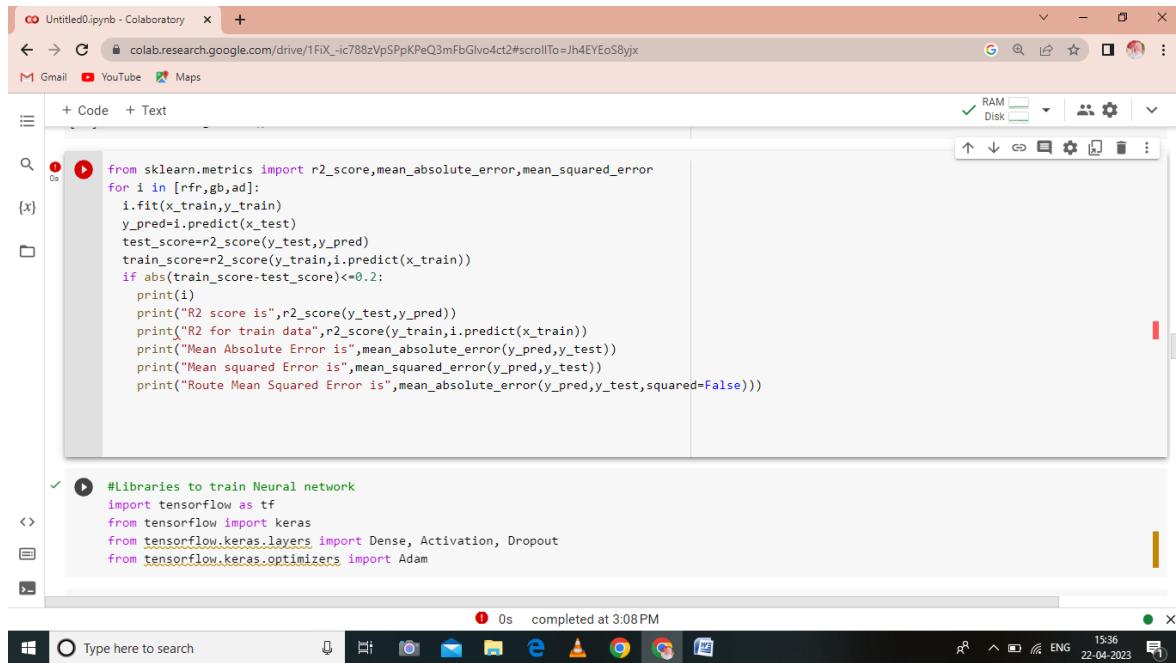
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

x_train.head()

from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
Run cell (Ctrl+Enter)
cell executed since last change
```

0s completed at 3:08 PM

Project Report Template



```

from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
for i in [rfr,gb,ad]:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    test_score=r2_score(y_test,y_pred)
    train_score=r2_score(y_train,i.predict(x_train))
    if abs(train_score-test_score)<=0.2:
        print(i)
        print("R2 score is",r2_score(y_test,y_pred))
        print("R2 for train data",r2_score(y_train,i.predict(x_train)))
        print("Mean Absolute Error is",mean_absolute_error(y_pred,y_test))
        print("Mean squared Error is",mean_squared_error(y_pred,y_test))
        print("Root Mean Squared Error is",mean_absolute_error(y_pred,y_test,squared=False)))

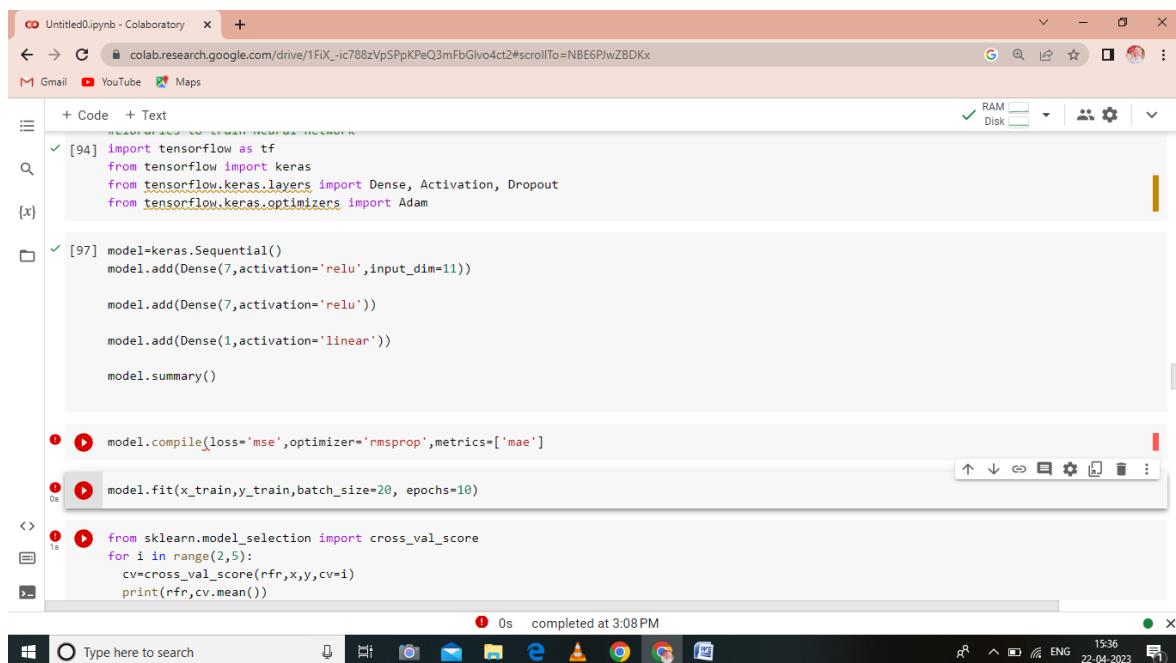
```

```

#Libraries to train Neural network
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam

```

0s completed at 3:08PM



```

[94]: import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam

[97]: model=keras.Sequential()
model.add(Dense(7,activation='relu',input_dim=11))

model.add(Dense(7,activation='relu'))

model.add(Dense(1,activation='linear'))

model.summary()

[98]: model.compile(loss='mse',optimizer='rmsprop',metrics=['mae'])

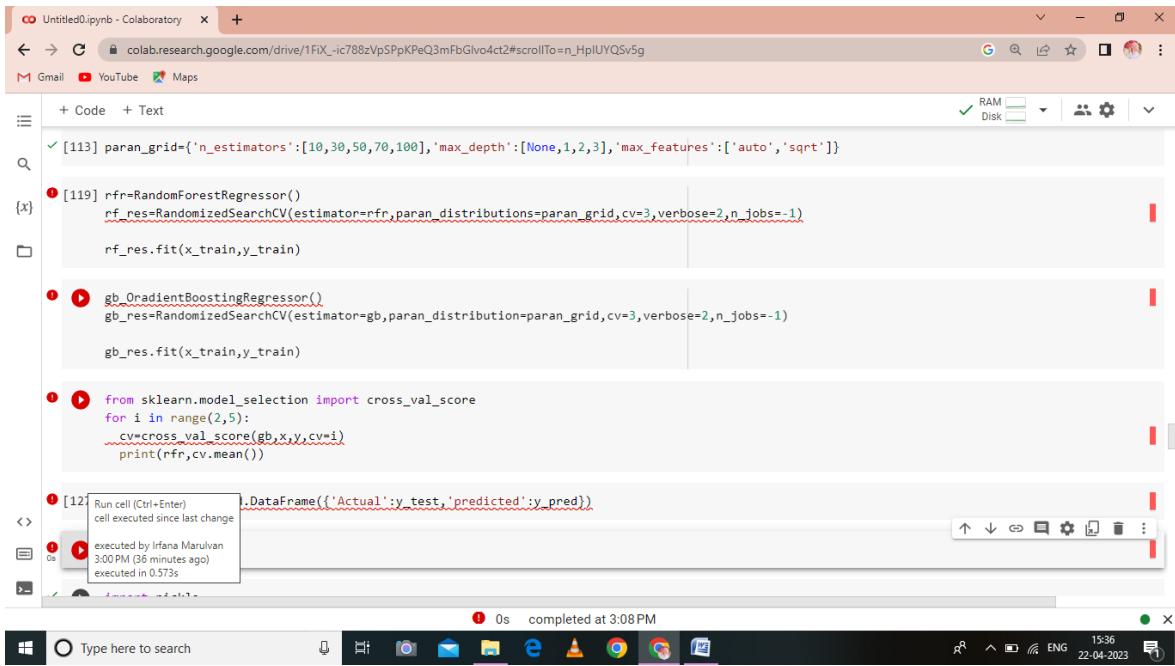
[99]: model.fit(x_train,y_train,batch_size=20, epochs=10)

[100]: from sklearn.model_selection import cross_val_score
for i in range(2,5):
    cv=cross_val_score(rfr,x,y,cv=i)
    print(rfr.cv.mean())

```

0s completed at 3:08PM

Project Report Template



```

[113] param_grid={'n_estimators':[10,30,50,70,100],'max_depth':[None,1,2,3],'max_features':['auto','sqrt']}
[119] rfr=RandomForestRegressor()
rf_res=RandomizedSearchCV(estimator=rfr,param_distributions=param_grid, cv=3, verbose=2, n_jobs=-1)
rf_res.fit(x_train,y_train)

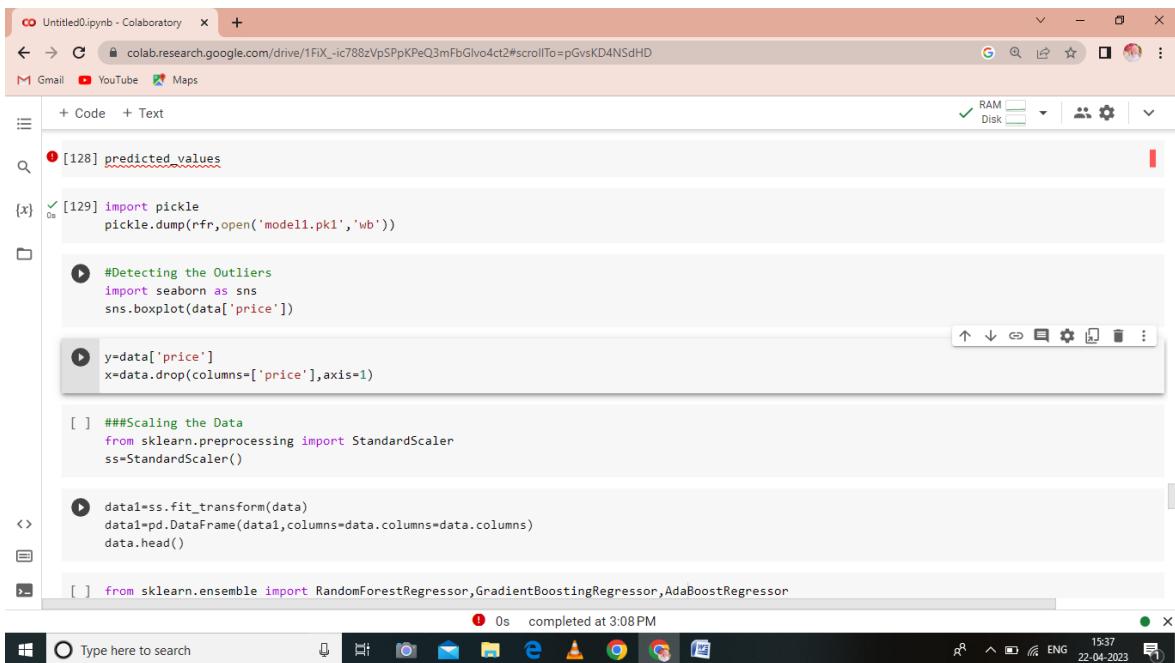
[120] gb_GradientBoostingRegressor()
gb_res=RandomizedSearchCV(estimator=gb,param_distributions=param_grid, cv=3, verbose=2, n_jobs=-1)
gb_res.fit(x_train,y_train)

[121] from sklearn.model_selection import cross_val_score
for i in range(2,5):
    cv=cross_val_score(gb,x,y, cv=i)
    print(rfr, cv.mean())

```

[122] Run cell (Ctrl+Enter)
cell executed since last change
executed by Irina Muruvan
3:00 PM (36 minutes ago)
executed in 0.573s

0s completed at 3:08PM



```

[128] predicted_values
[129] import pickle
pickle.dump(rfr,open('model1.pkl','wb'))

#Detecting the Outliers
import seaborn as sns
sns.boxplot(data['price'])

y=data['price']
x=data.drop(columns=['price'],axis=1)

###Scaling the Data
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()

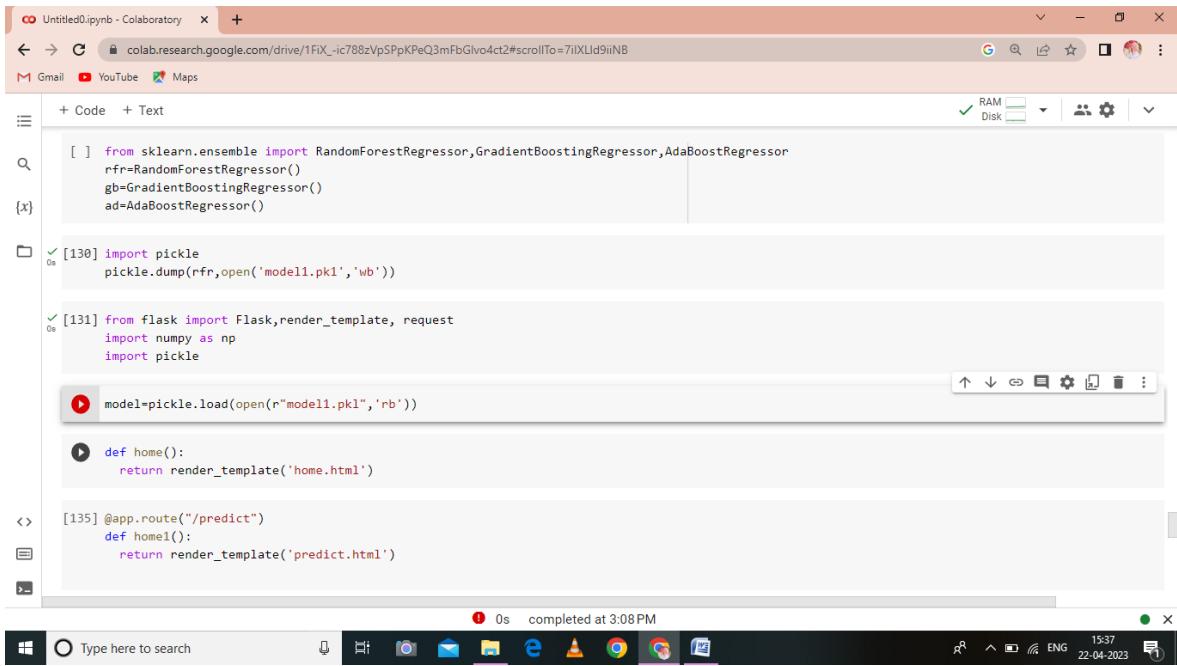
data1=ss.fit_transform(data)
data1=pd.DataFrame(data1,columns=data.columns)
data1.head()

[ ] from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor

```

0s completed at 3:08PM

Project Report Template



```

[ ] from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
rfr=RandomForestRegressor()
gb=GradientBoostingRegressor()
ad=AdaBoostRegressor()

[130] import pickle
pickle.dump(rfr,open('model1.pkl','wb'))

[131] from flask import Flask,render_template, request
import numpy as np
import pickle

model=pickle.load(open(r"model1.pkl",'rb'))

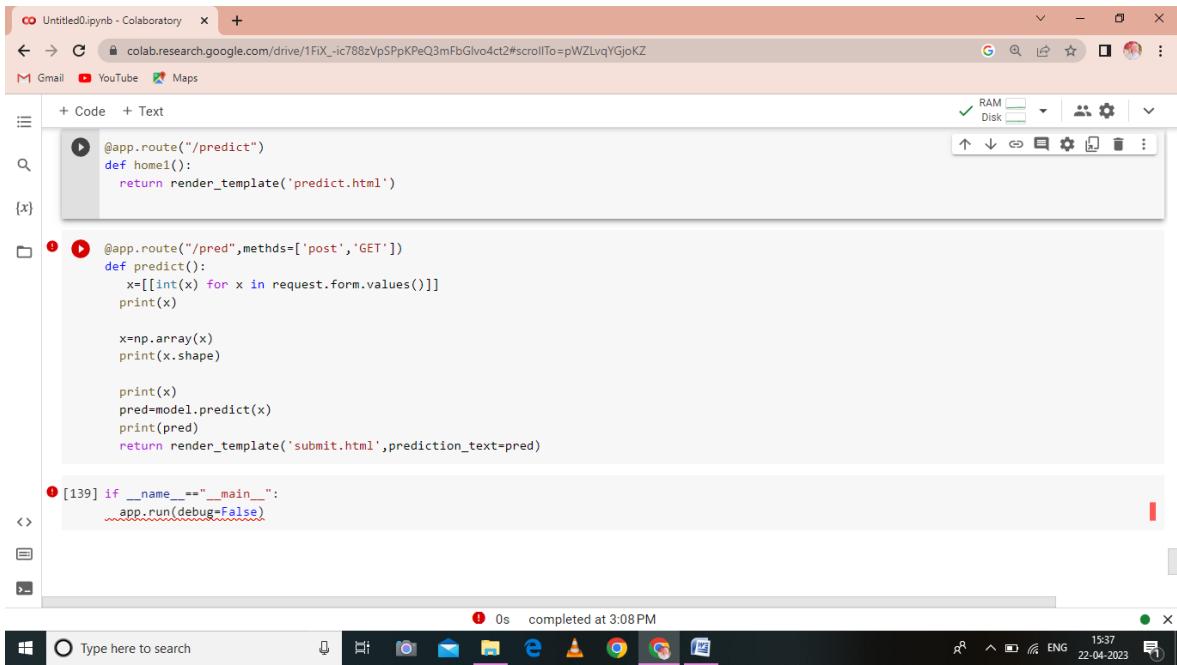
def home():
    return render_template('home.html')

[135] @app.route("/predict")
def home1():
    return render_template('predict.html')

```

0s completed at 3:08 PM

Type here to search



```

@app.route("/predict")
def home1():
    return render_template('predict.html')

[136] @app.route("/pred",methods=['post','GET'])
def predict():
    x=[[int(x) for x in request.form.values()]]
    print(x)

    x=np.array(x)
    print(x.shape)

    print(x)
    pred=model.predict(x)
    print(pred)
    return render_template('submit.html',prediction_text=pred)

[139] if __name__=="__main__":
    app.run(debug=False)

```

0s completed at 3:08 PM

Type here to search

