Project Report Template

1. INTRODUCTION:

A person who already has reserved a ticket for a flight realizes how powerfully the price of the ticket switches. Airline utilizes progressed techniques considered Revenue Management to accomplish acharacteristic esteeming technique . The most affordable ticket available changes over a course of time.The expense of the booking may be far and wide. This esteeming technique normally alters the costaccording to the different times in a day namely forenoon, evening, or night. Expenses for the flight may

similarly alter according to the different seasons in a year like summers, rainy and winters, also during the period of festivals. The buyers would be looking for the cheapest ticket while the outrageous objective ofthe transporter would be generating more and more revenue. Travelers for most part attempt to buy the ticket ahead of their departure day. The reason would be their belief that the prices might be the highest when they would make a booking much nearer to the day of their flight but conventionally this isn't

verifiable. The buyer might wrap up paying more than they should for a comparable seat. Considering thechallenges faced by the travellers for getting an affordable seat, various strategies are utilized which will

extract a particular day on which the fare will be the least. For this purpose, Machine Learning comes into the picture.

1.1Overview:

we will be analyzing the flight fare prediction using Machine Learning dataset using essential exploratory data analysis techniques then will draw some predictions about the price of the flight based on some features such as what type of airline it is, what is the arrival time, what is the departure time, what is the duration of the flight, source, destination and more.

1.2 Purpose:

With consideration of some features like arrival time,depature time as well as time to purchase the ticket using these factors prices can be predict.due to this factors there may be change in airline fare prices and also detect how factors are related to being change of Flight ticket.

A booking is an arrangement to reserve a certain ticket,accomodataion or a place in advance.Prices can be frozen by booking to pay a more affordable price for travel scheduled in the feature.

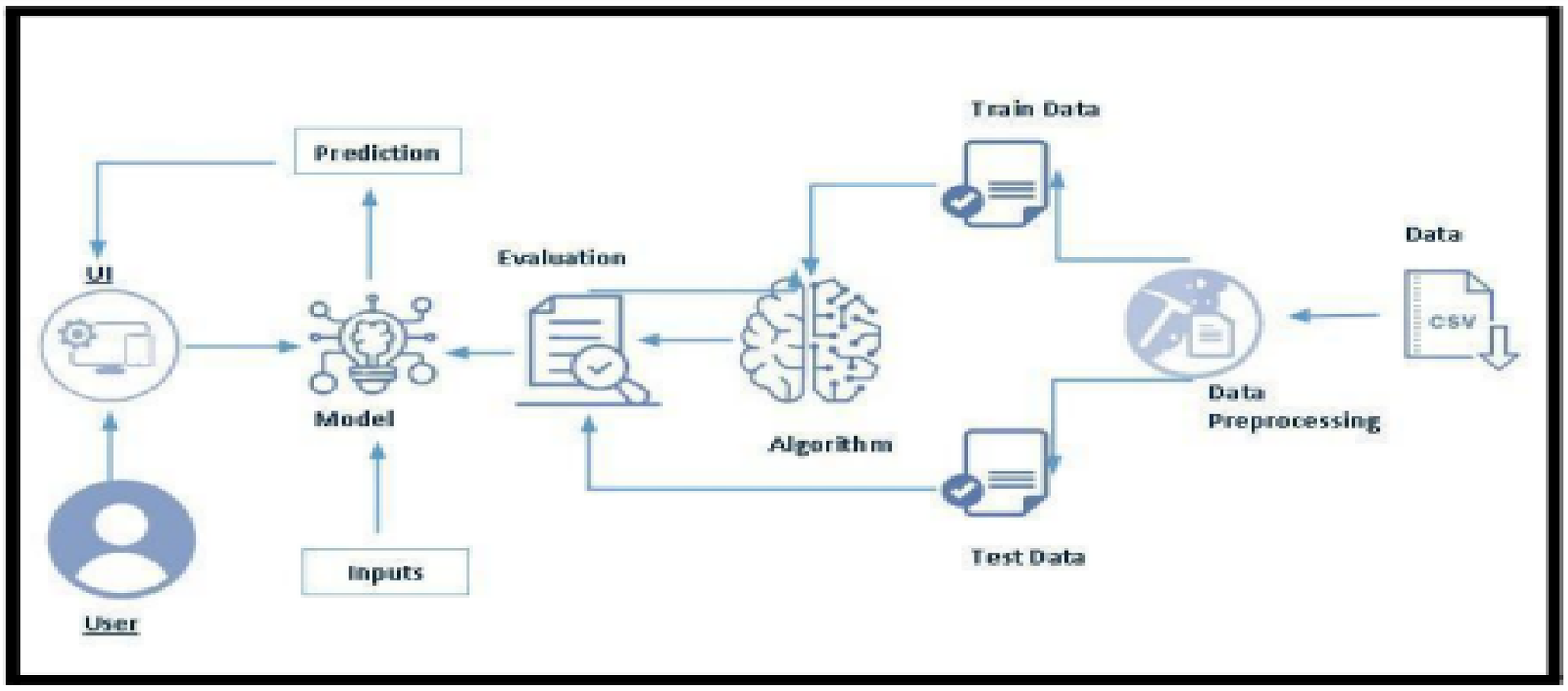
2. Problem Definition & Design Thinking:

Problem Definition:

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. Features are taken from chosen dataset and in the price wherein the airline price ticket costs vary overtime. we have implemented flight price prediction for users by using KNN, decisiontree 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.

Design Thinking:

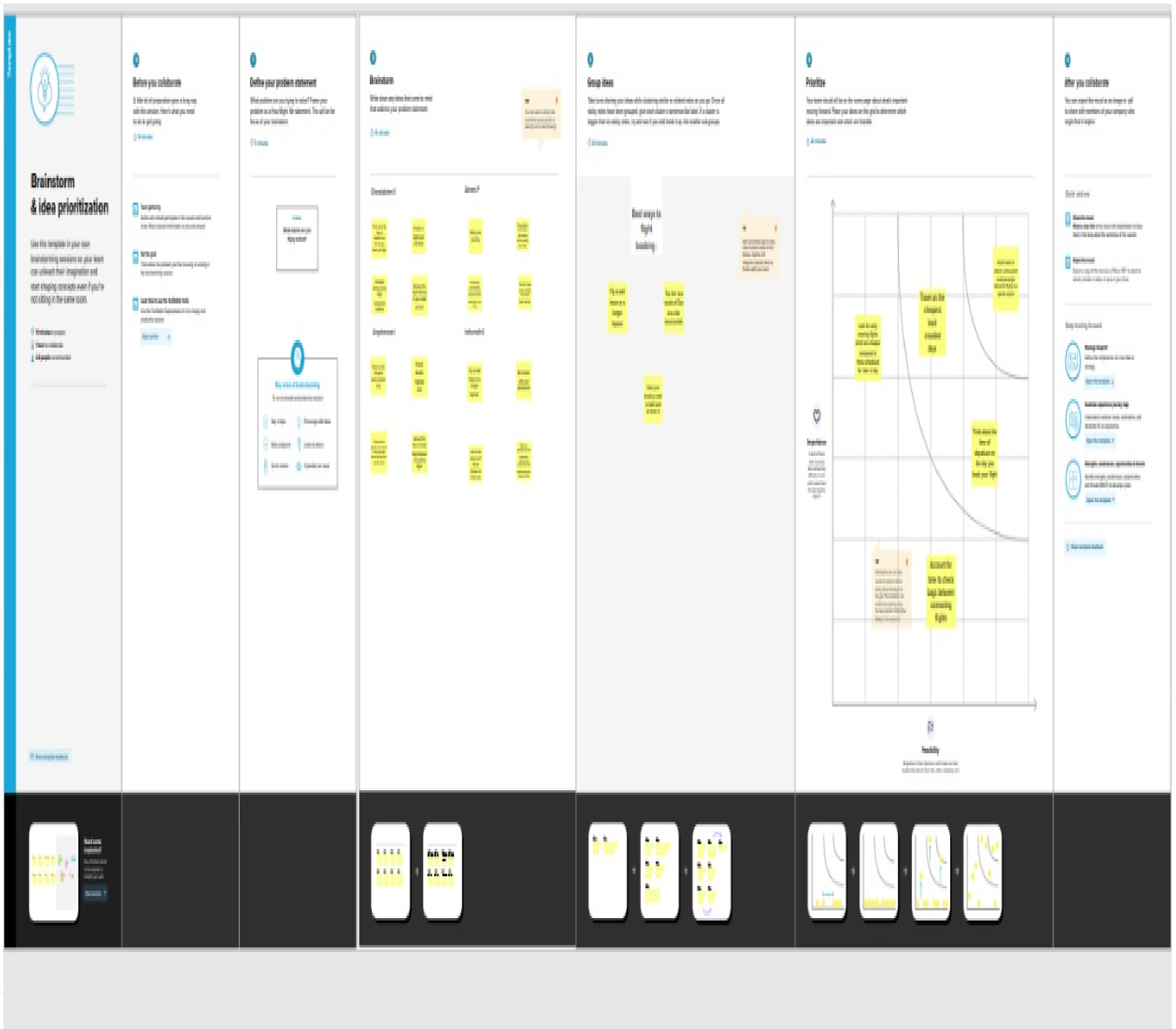


2.1 Empathy Map:

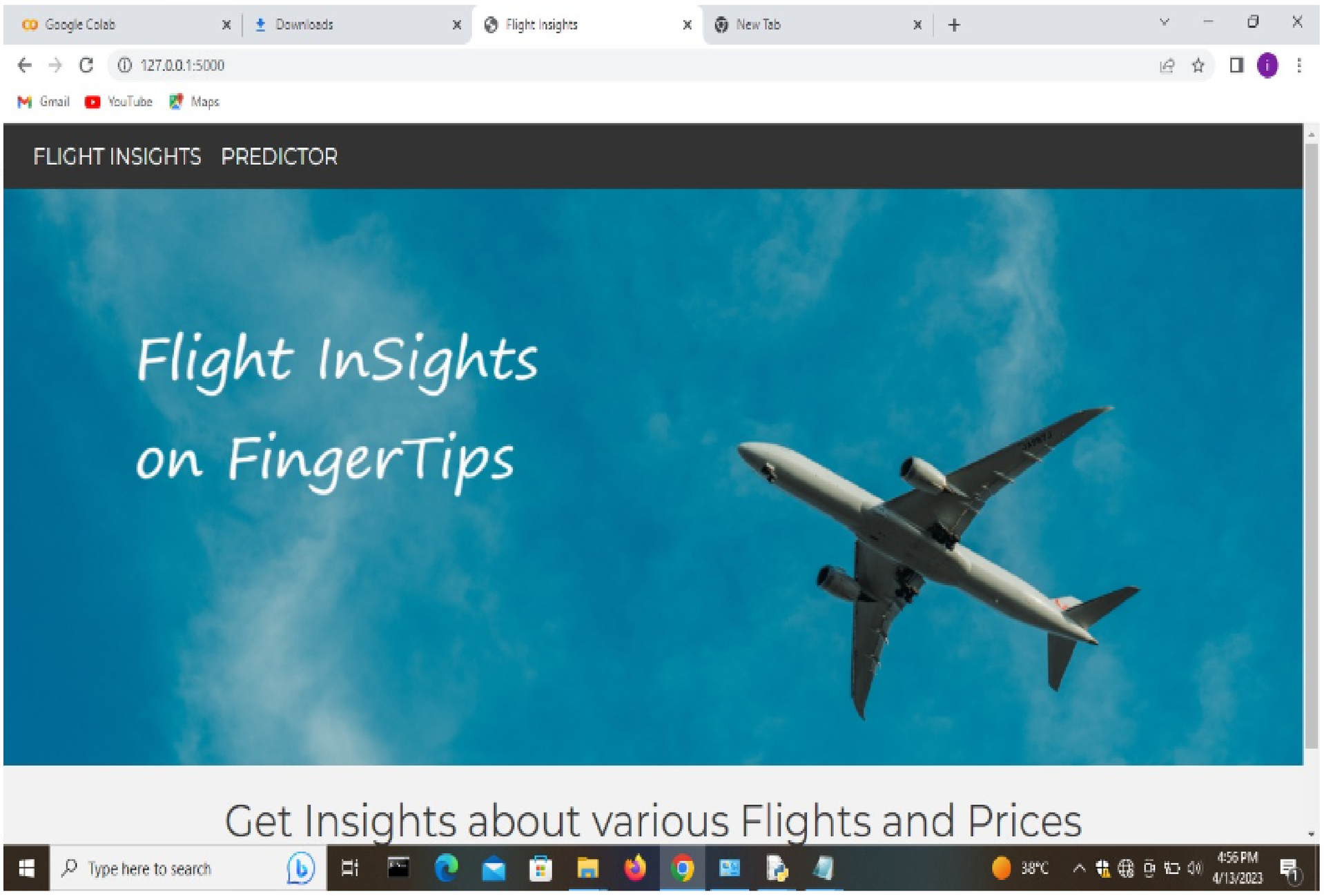
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

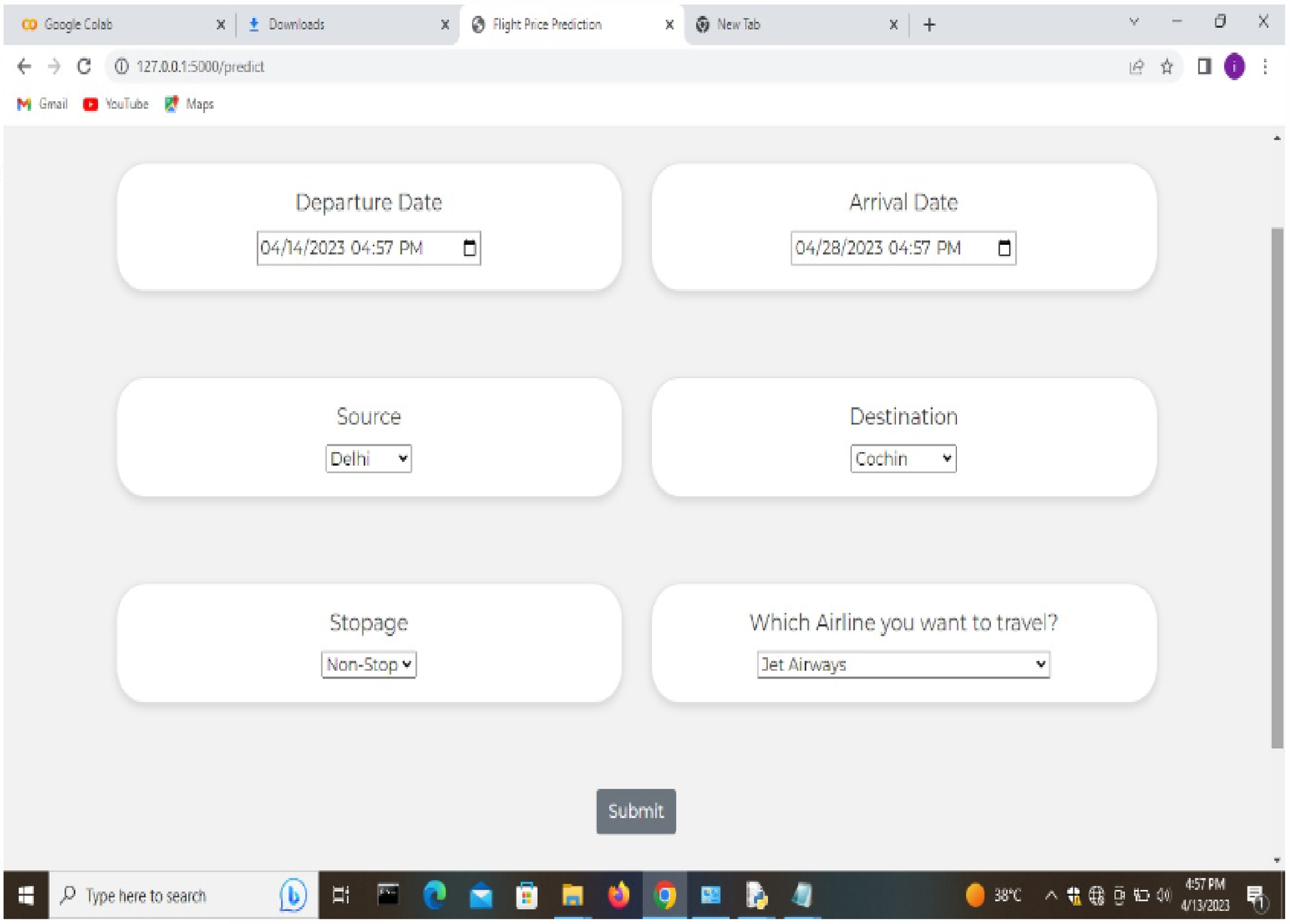
2.2 Ideation & Brainstroming Map:

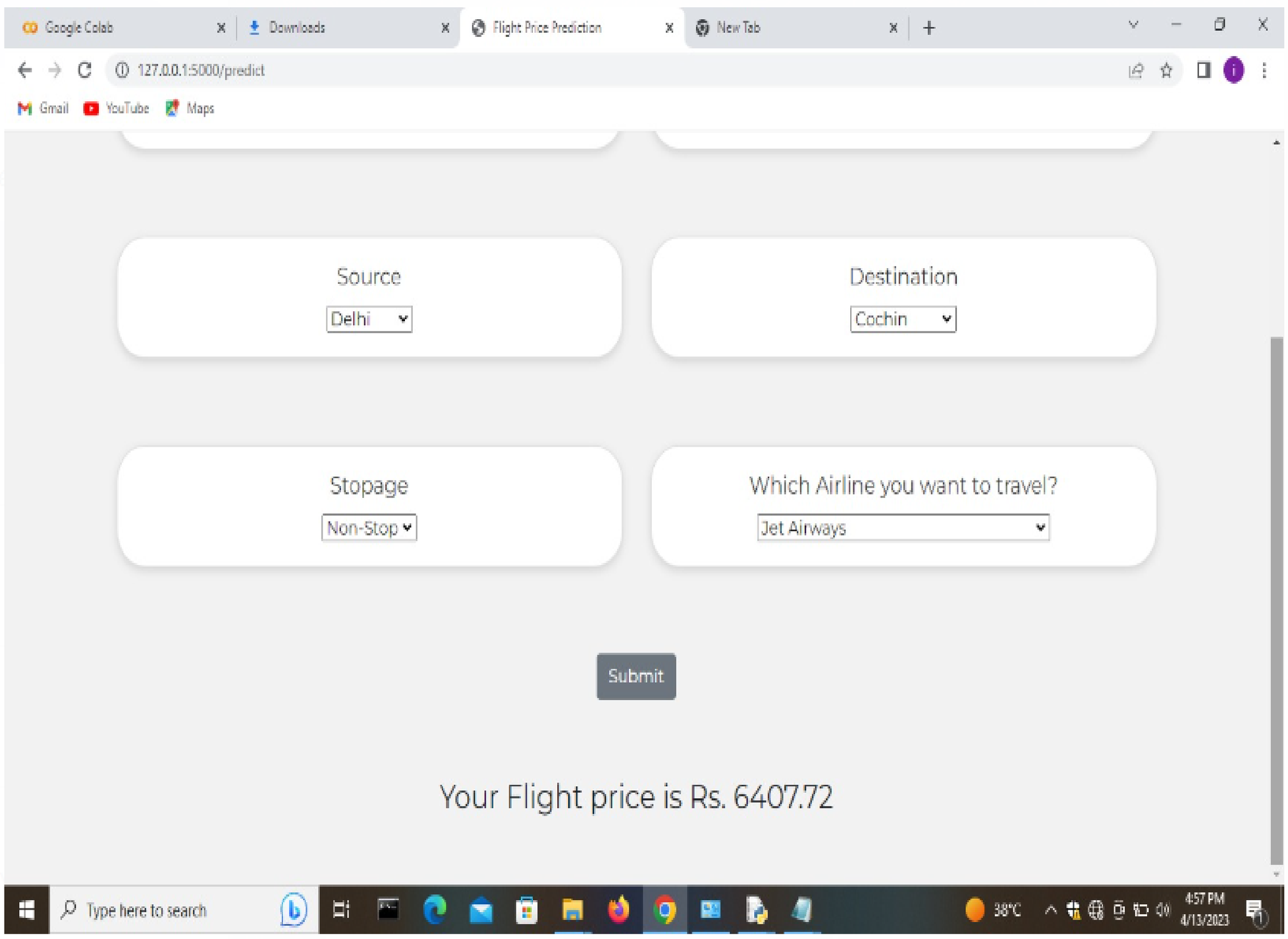
Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.


Brainstorm
& idea prioritization
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.
Before you collaborate
Define your problem statement
Brainstorm
Group ideas
Prioritize
After you collaborate
Best ways to flight booking
Travel on the cheapest and shortest day
Account for time to check bags between connecting flights
Possibility

3.RESULT:


Google Colab
Downloads
Flight Insights
New Tab
127.0.0.1:5000
Gmail
YouTube
Maps
FLIGHT INSIGHTS PREDICTOR
Flight InSights
on FingerTips
Get Insights about various Flights and Prices
Type here to search
38°C
4:56 PM
4/13/2023


Google Colab
Downloads
Flight Price Prediction
New Tab
127.0.0.1:5000/predict
Gmail
YouTube
Maps
Departure Date
04/14/2023 04:57 PM
Arrival Date
04/28/2023 04:57 PM
Source
Delhi
Destination
Cochin
Stopage
Non-Stop
Which Airline you want to travel?
Jet Airways
Submit
Type here to search
38°C
4:57 PM
4/13/2023


Google Colab
Downloads
Flight Price Prediction
New Tab
127.0.0.1:5000/predict
Gmail
YouTube
Maps
Source
Delhi
Destination
Cochin
Stopage
Non-Stop
Which Airline you want to travel?
Jet Airways
Submit
Your Flight price is Rs. 6407.72
Type here to search
38°C
4:57 PM
4/13/2023

4. ADVANTAGES

- **Convenience**

  The greatest benefit of booking a flight online is the convenience.

- **Numerous flight**

  With booking flight online you will have a vast selection of flights and hotels to choose from. Using online booking services is usually the best way through which travelers can get the best possible deals.

- **Price**

  When making your reservations online, you can look around and choose the best price for you.

- **Changes and cancellations**

  It is easier for travelers to make changes or cancellations to their online reservations. You will not have to call the airline or hotel and

wait for a customer service representative to assist you with the process.

- **Loyalty points**

  Booking flights online enable the travelers to gain loyalty points on credit cards that they can redeem for a souvenir, gifts or may be free flights accommodation.

- **Total cost of your flight**

  Booking your flight will display the full price description to customers. This includes the taxes and other charges incurred. This also means that the price that is displayed on your screen is the actual price you will pay. This way, customers are fully informed and are not surprised by any secret charges.

DISADVANTAGES:

Need a good internet connection.

Reliable internet access is required to check reservations and add bookings that are made over the phone.

You need to be ready for an influx of new customers.

Not all online booking systems are created equal.

Enter the wrong information and book wrong flight.

High competition on the market.

Booking agencies on the high street may close down due to lack of customers.

5.APPLICATIONS:

Flight booking applications helps the airline industry automate the booking process.Users worldwide can book flights on the go using the

sample apps,which include feature such as quick flight search,download tickets,check and modify book details,one tap,check in,and many more.

The Hopper app has helped over 70 million travelers find and secure the best price on flights, each and every time they book their trips - saving its users trip compared to other travel booking sites or apps.

Hooper is one of the best flight booking apps.It allows you to book flights uo to a year in advance and with 95% accuracy.so if you are planning for something book using hooper.

6.CONCLUSION:

We collected airfare data from a particular airline organization from internet and showed that it is realizable to foresee costs for flights based on recorded fare data. From the experiments we concluded which reviews impacts airfare prediction at most.

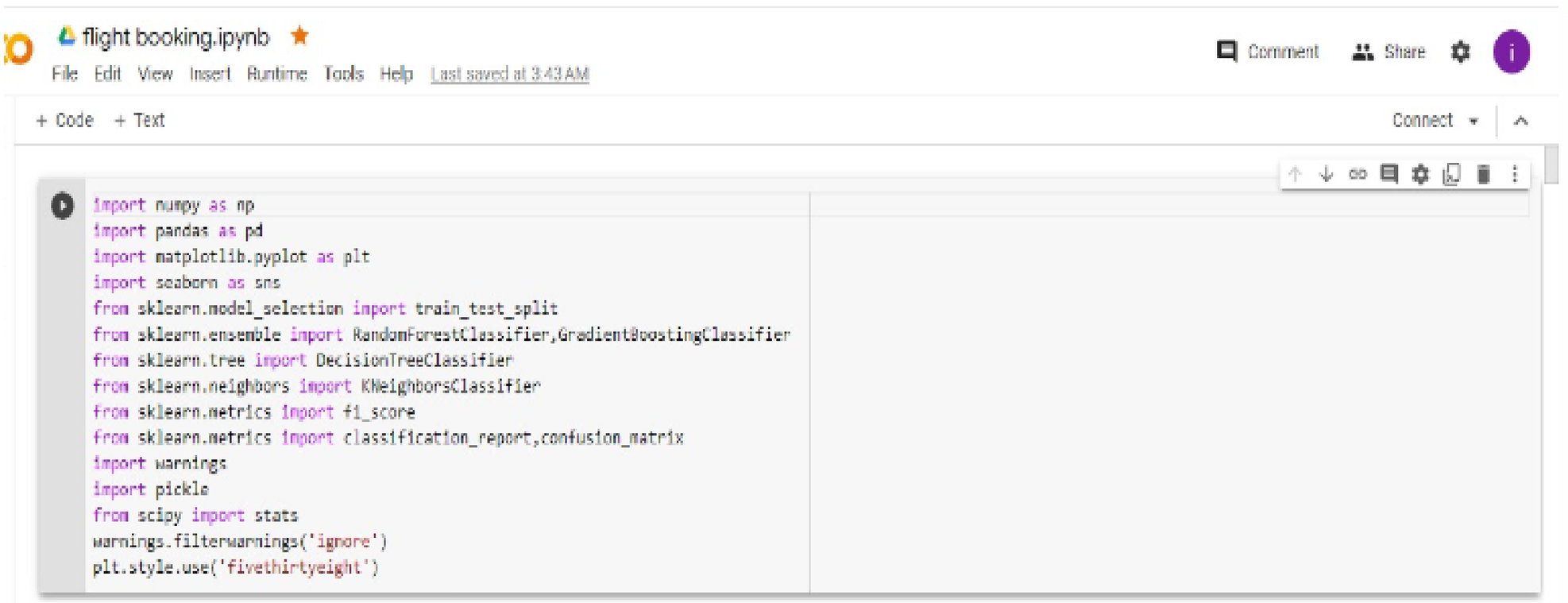
A part from the features selected, there are other features that could improve the prediction accuracy. In the future, this work could be extended to predict the airfare prices for the entire flight map of the airline. Additional experiments on larger airfare data sets are essential, but this initial pilot study highlights the potential of Machine Learning models to guide consumers to make an airfare purchase in the best market period.

7.FUTURE SCOPE:

In Upcoming days when huge amount of information is accessed as in detailed information in the dataset, the expected results in future are highly correct. For further research anyone desire to expand upon it ought to request different sources of historical data or be a lot of organized in collection knowledge manually over amount of your time to boot, a lot of different combination of plane are going to be traversed. There is whole possibility that planes differ their execution ideas consisting characteristics of the plane. At last, it is curious to match our model accuracy with that of the business models accuracy offered nowadays.

8.APPENDIX:

A. Source code:



flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

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

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

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

```
[ ] for i in data["Airline"]:
      print(data.Airline.unique())
```

```
Streaming output truncated to the last 5000 lines.
 'Vistara' 'Air Asia' 'Vistara Premium economy' 'Jet Airways Business'
 'Multiple carriers Premium economy' 'Trujet']
['IndiGo' 'Air India' 'Jet Airways' 'SpiceJet' 'Multiple carriers' 'GoAir'
 'Vistara' 'Air Asia' 'Vistara Premium economy' 'Jet Airways Business'
 'Multiple carriers Premium economy' 'Trujet']
['IndiGo' 'Air India' 'Jet Airways' 'SpiceJet' 'Multiple carriers' 'GoAir'
```

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] 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: 10683, dtype: object
```

```
[ ] data['Date']=data.Date_of_Journey.str[0]
    data['Month']=data.Date_of_Journey.str[1]
    data['Year']=data.Date_of_Journey.str[2]
```

```
[ ] data.Total_Stops.unique()
```

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

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] data.Route=data.Route.str.split('->')
```

```
[ ] 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: 10683, dtype: object
```

```
[ ] 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]
```

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] 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)
```

```
[ ] data.Additional_Info.replace('No Info','No Info',inplace=True)
```

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

```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
```

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

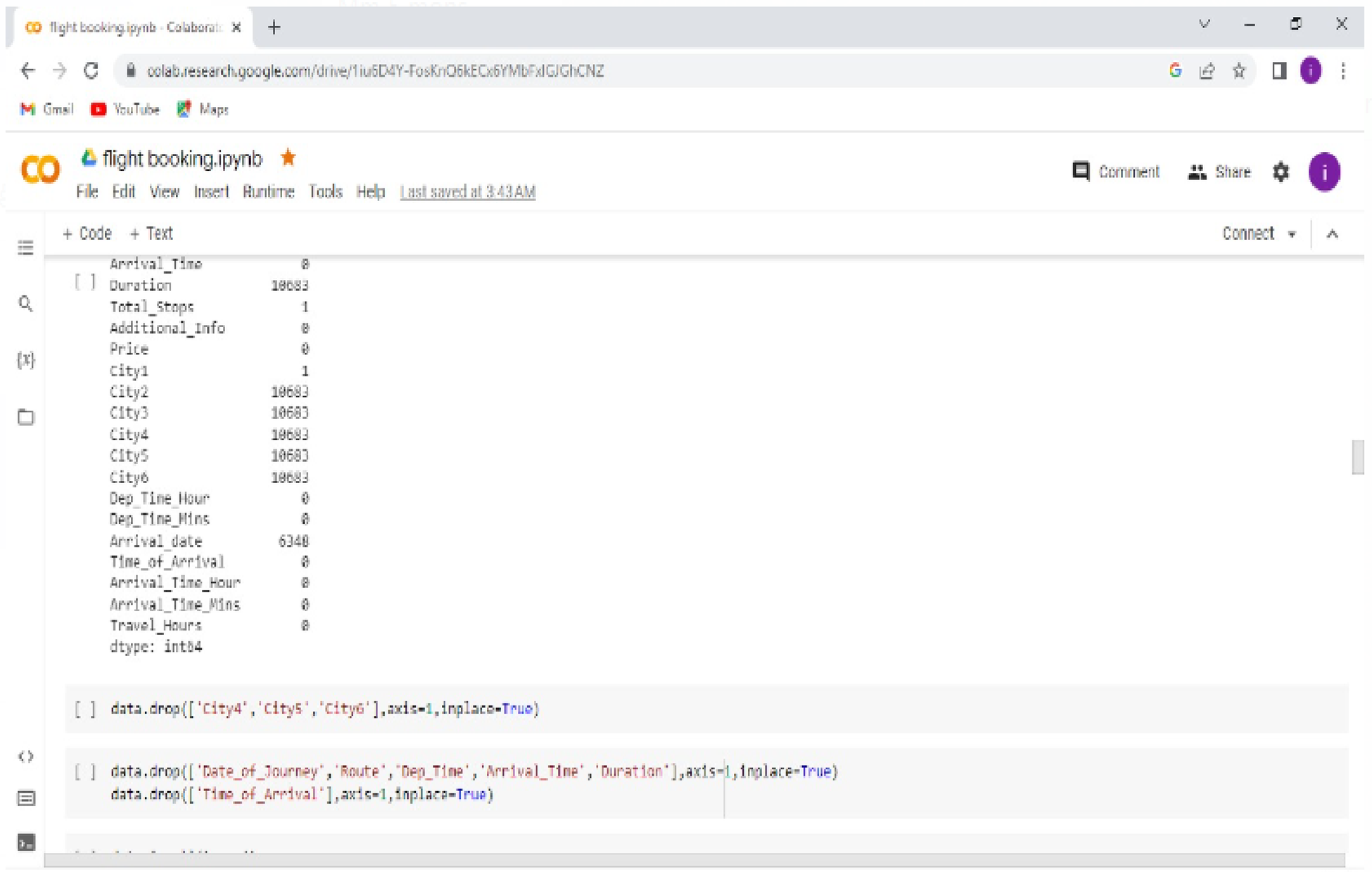
```
[ ] data.Dep_Time=data.Dep_Time.str.split(':')
```

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

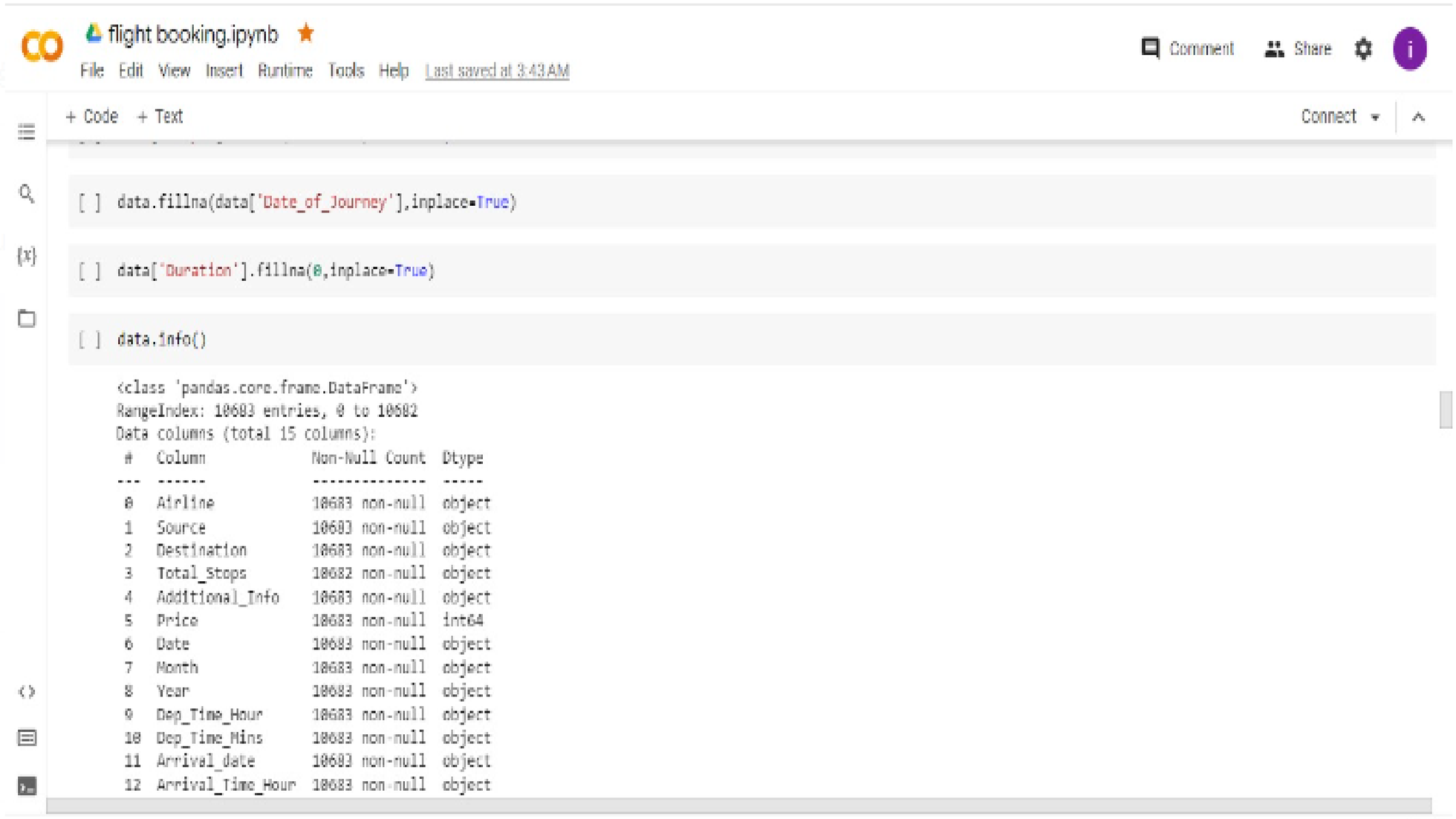
```
[ ] data.Arrival_Time=data.Arrival_Time.str.split(' ')
    data['Arrival_date']=data.Arrival_Time.str[1]
    data['Time_of_Arrival']=data.Arrival_Time.str[0]
    data['Time_of_Arrival']=data.Time_of_Arrival.str.split(':')
    data['Arrival_Time_Hour']=data.Time_of_Arrival.str[0]
    data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]
```

```
[ ] data.Duration=data.Duration.str.split('')
```

```
[ ] data.Duration=data.Duration.str[0]
    data.Duration=data.Duration.str.split('h')
    data.Duration=data.Duration.str[0]
    data.Duration=data.Duration
    data.Travel_Mins=data.Duration.str[1]
    #data.Travel_Mins=data.Travel_Mins.str.split('m')
    #data.Travel_Mins=data.Travel_Mins.str[0]
```


flight booking.ipynb - Colaborat
colab.research.google.com/drive/1iu6D4Y-FosKnQ6kECx6YMbFxIGJGhCNZ
Gmail
YouTube
Maps
flight booking.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:43 AM
Comment
Share
+ Code + Text
Connect
Arrival_Time 0
Duration 10683
Total_Stops 1
Additional_Info 0
Price 0
City1 1
City2 10683
City3 10683
City4 10683
City5 10683
City6 10683
Dep_Time_Hour 0
Dep_Time_Mins 0
Arrival_date 6348
Time_of_Arrival 0
Arrival_Time_Hour 0
Arrival_Time_Mins 0
Travel_Hours 0
dtype: int64
data.drop(['City4','City5','City6'],axis=1,inplace=True)
data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1,inplace=True)
data.drop(['Time_of_Arrival'],axis=1,inplace=True)


flight booking.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:43 AM
Comment
Share
+ Code + Text
Connect
data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1,inplace=True)
data.drop(['Time_of_Arrival'],axis=1,inplace=True)
data.isnull().sum()
Airline 0
Source 0
Destination 0
Total_Stops 1
Additional_Info 0
Price 0
City1 1
City2 10683
City3 10683
Dep_Time_Hour 0
Dep_Time_Mins 0
Arrival_date 6348
Arrival_Time_Hour 0
Arrival_Time_Mins 0
Travel_Hours 0
dtype: int64
data['City3'].fillna('None',inplace=True)


flight booking.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:43 AM
Comment
Share
+ Code + Text
Connect
[ ] data.fillna(data['Date_of_Journey'],inplace=True)
[ ] data['Duration'].fillna(0,inplace=True)
[ ] data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 15 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Airline 10683 non-null object
1 Source 10683 non-null object
2 Destination 10683 non-null object
3 Total_Stops 10682 non-null object
4 Additional_Info 10683 non-null object
5 Price 10683 non-null int64
6 Date 10683 non-null object
7 Month 10683 non-null object
8 Year 10683 non-null object
9 Dep_Time_Hour 10683 non-null object
10 Dep_Time_Mins 10683 non-null object
11 Arrival_Date 10683 non-null object
12 Arrival_Time_Hour 10683 non-null object

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] data[data['Travel_Hours']=='150m']
```

Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_Ti

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

```
[ ] categorical=['Airline','Source','Destination','Additional_Info','City1']
    numerical=['Total_Stops','Date','Month','Year','Dep_time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Hour','Arrival_Time_mins','Travel_Hours','Travel_Mins
```

```
[ ] 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()
```

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

[ ]

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_T
0	3	0	5	4	8	3897	18	0	0	22	20	22	01	
1	1	3	0	1	8	7662	84	0	0	05	50	NaN	13	
2	4	2	1	1	8	13882	118	0	0	09	25	10	04	
3	3	3	0	0	8	6218	91	0	0	18	05	NaN	23	
4	3	0	5	0	8	13302	29	0	0	16	50	NaN	21	

```
[ ] data.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_Tin
0	3	0	5	4	No info	3897	24	03	2019	22	20	22	01	
1	1	3	0	1	No info	7662	1	05	2019	05	50	1	13	
2	4	2	1	1	No info	13882	9	06	2019	09	25	10	04	
3	3	3	0	0	No info	6218	12	05	2019	18	05	12	23	
4	3	0	5	0	No info	13302	01	03	2019	16	50	01	21	

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share Connect

+ Code + Text

[ ]

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3
count	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.0	10682.0
mean	3.966205	1.952059	1.435967	1.458622	7.392998	9086.292735	74.827748	0.0	0.0
std	2.352090	1.177110	1.474773	1.806640	1.214254	4610.685695	36.731657	0.0	0.0
min	0.000000	0.000000	0.000000	0.000000	0.000000	1759.000000	0.000000	0.0	0.0
25%	3.000000	2.000000	0.000000	0.000000	8.000000	5277.000000	48.000000	0.0	0.0
50%	4.000000	2.000000	1.000000	0.000000	8.000000	8372.000000	73.000000	0.0	0.0
75%	4.000000	3.000000	2.000000	4.000000	8.000000	12373.000000	104.000000	0.0	0.0
max	11.000000	4.000000	5.000000	5.000000	9.000000	79512.000000	128.000000	0.0	0.0

```
[ ] 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
```

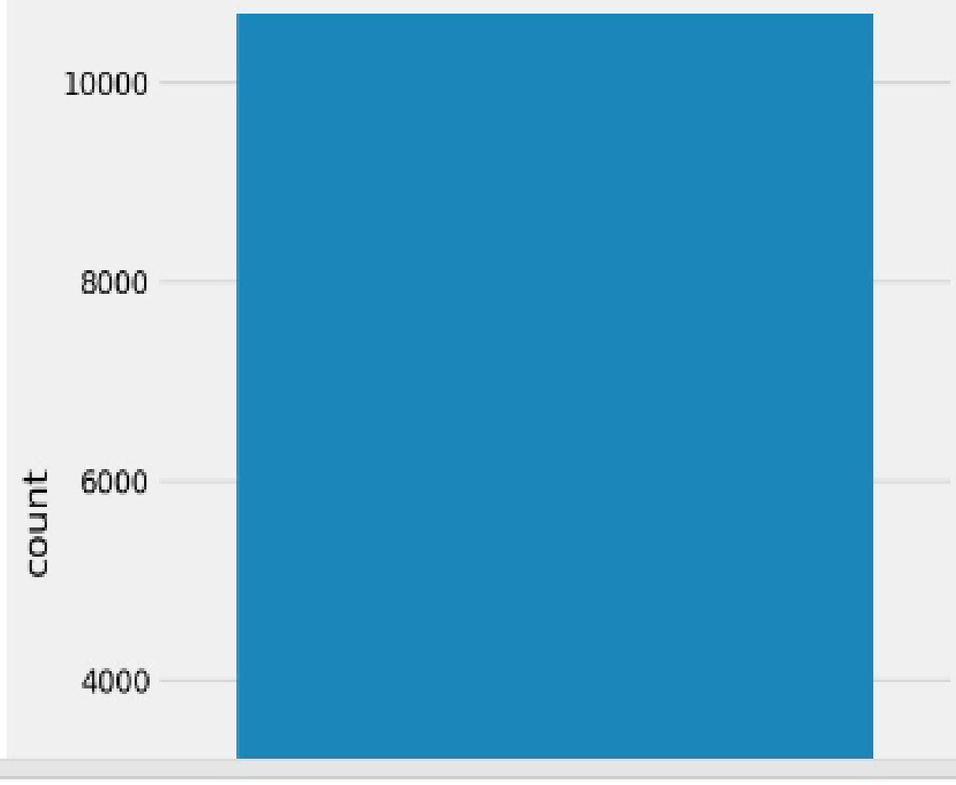
flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share Connect

+ Code + Text

[ ]



flight booking.ipynb

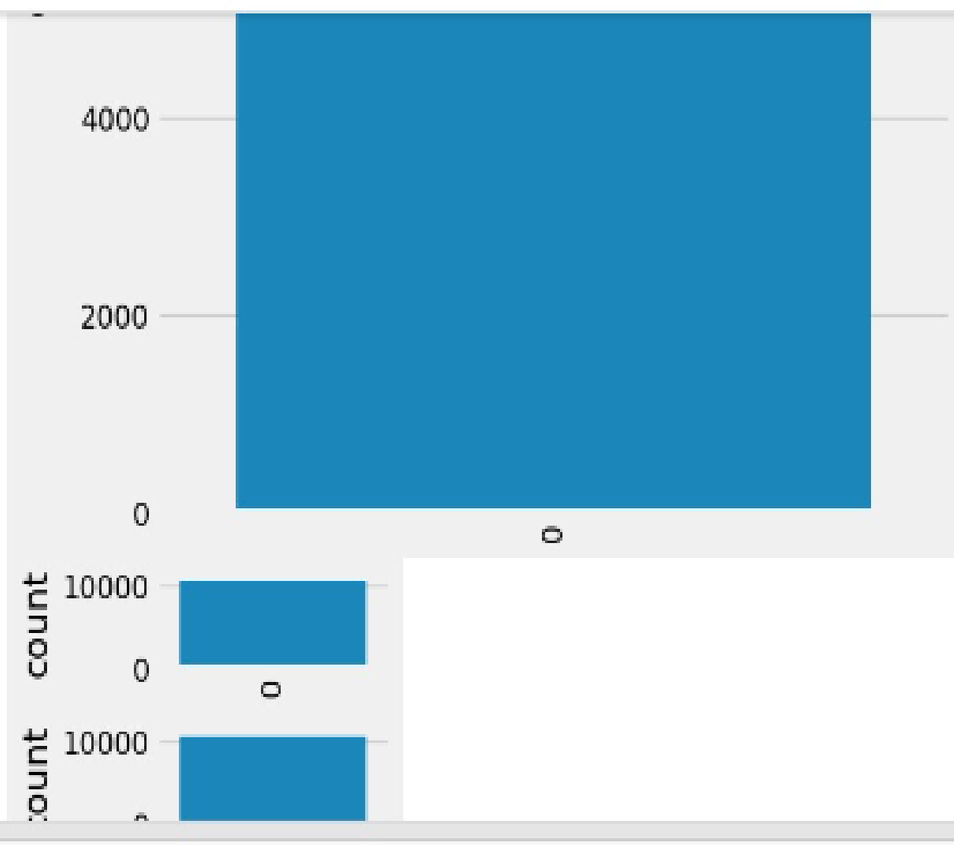
File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

[ ]



flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] plt.figure(figsize=(15,8))
    sns.displot(data.Price)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f40fd14d708>
<Figure size 1500x800 with 0 Axes>
```

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

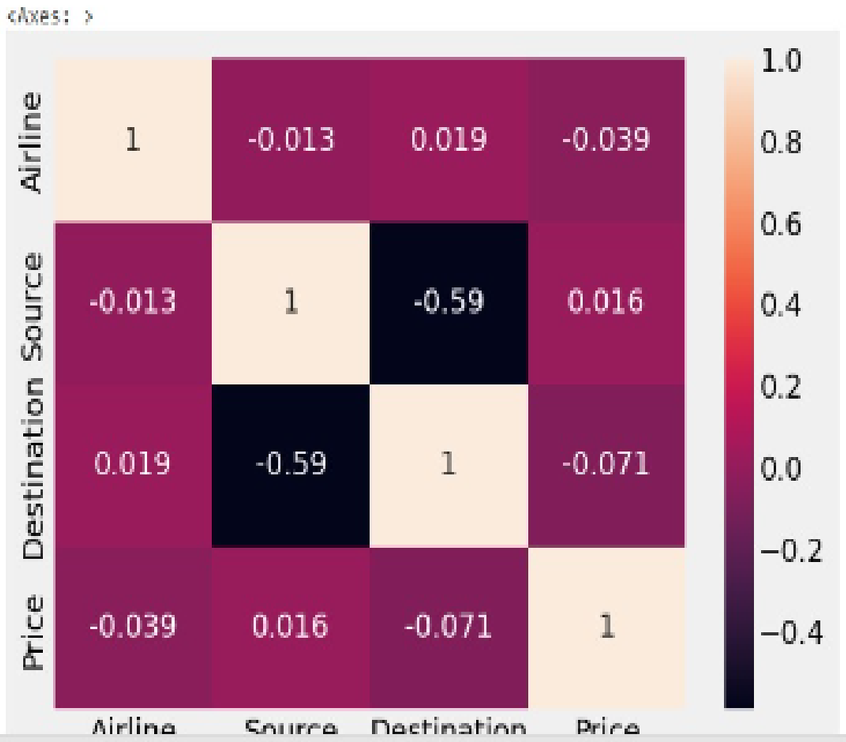
Comment Share

+ Code + Text

Connect

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

```
<Axes: >
```



flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

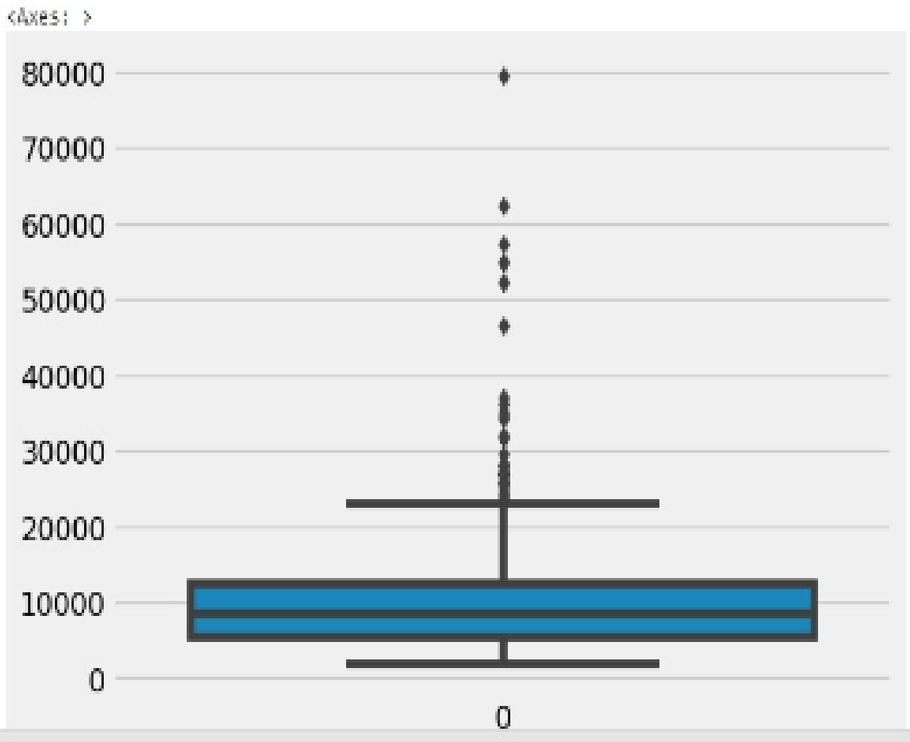
Comment Share

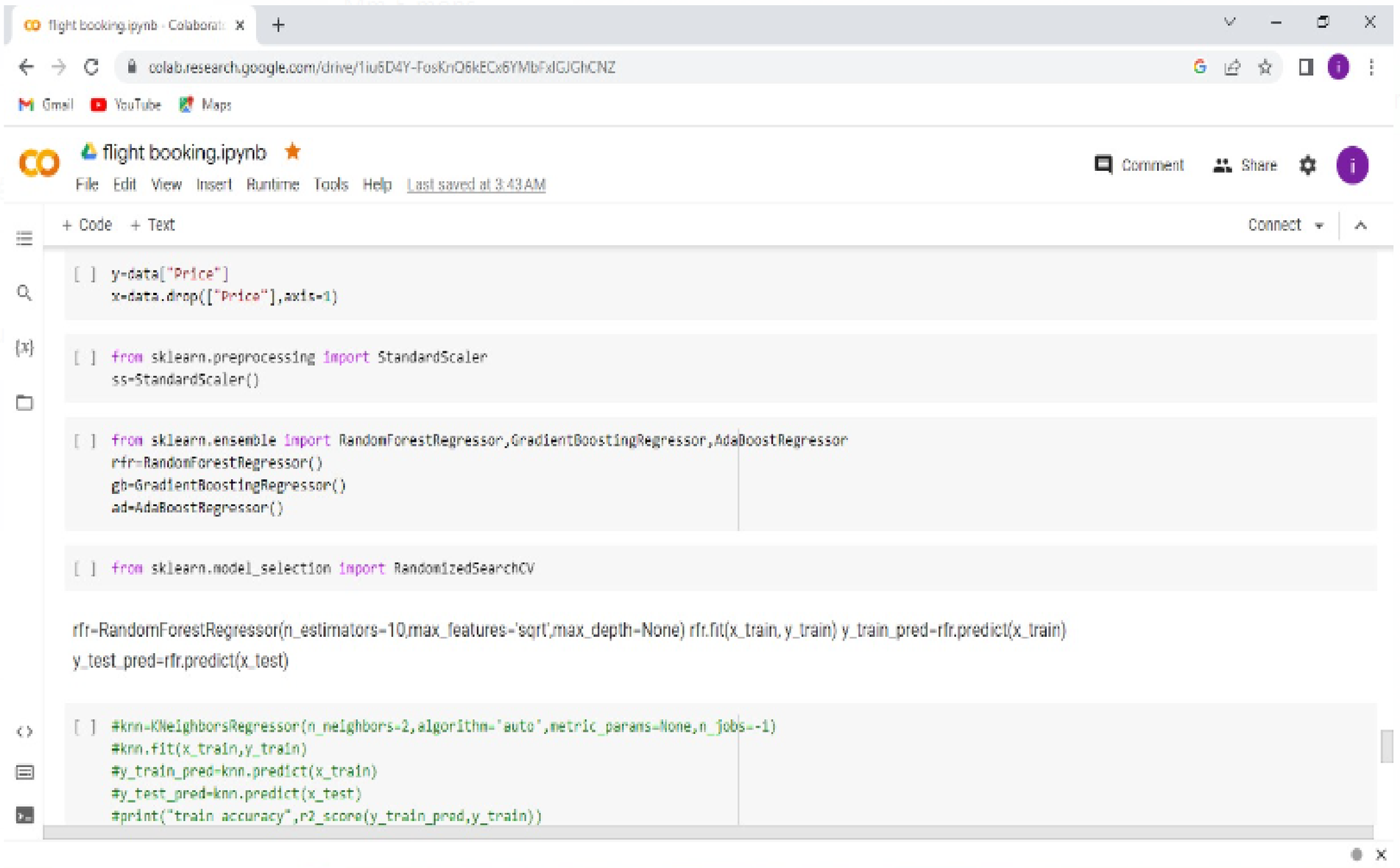
+ Code + Text

Connect

```
import seaborn as sns
[ ] sns.boxplot(data['Price'])
```

```
<Axes: >
```




flight booking.ipynb - Colaborat
colab.research.google.com/drive/1iu6D4Y-FosKnO6kECx6YMbFxIGJGhCNZ
Gmail
YouTube
Maps
flight booking.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:43 AM
Comment
Share
+ Code + Text
Connect
[ ] y=data["Price"]
x=data.drop(["Price"],axis=1)
[ ] from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
[ ] from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
rfr=RandomForestRegressor()
gb=GradientBoostingRegressor()
ad=AdaBoostRegressor()
[ ] from sklearn.model_selection import RandomizedSearchCV
rfr=RandomForestRegressor(n_estimators=10,max_features='sqrt',max_depth=None) rfr.fit(x_train, y_train) y_train_pred=rfr.predict(x_train)
y_test_pred=rfr.predict(x_test)
[ ] #knn=KNeighborsRegressor(n_neighbors=2,algorithm='auto',metric_params=None,n_jobs=-1)
#knn.fit(x_train,y_train)
#y_train_pred=knn.predict(x_train)
#y_test_pred=knn.predict(x_test)
#print("train accuracy",r2_score(y_train_pred,y_train))

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

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

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

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

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

flight booking.ipynb

File Edit View Insert Runtime Tools Help Last saved at 3:43 AM

Comment Share

+ Code + Text

Connect

```
[ ] data=data[['Airline','Source','Destination','Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Mins','Price']]
```

```
[ ] data.head()
```

	Airline	Source	Destination	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Mins	Price
0	3	0	5	24	03	2019	22	20	22	10	3897
1	1	3	0	1	05	2019	05	50	1	15	7662
2	4	2	1	9	06	2019	09	25	10	25	13882
3	3	3	0	12	05	2019	18	05	12	30	6218
4	3	0	5	01	03	2019	16	50	01	35	13302

```
[ ] data.describe()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3
count	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.000000	10682.0	10682.0
mean	3.966205	1.952069	1.435967	1.458622	7.392998	9086.292735	74.827748	0.0	0.0
std	2.352090	1.177110	1.474773	1.806640	1.214254	4610.685695	36.731657	0.0	0.0