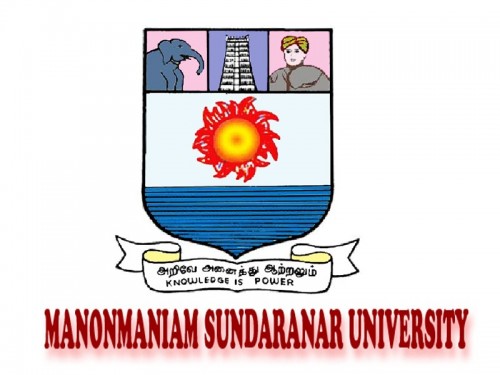
**Kamarajar Government Arts College,**

**Surandai**

**Tenkasi (Dt.) - 627859.**



**APRIL - 2023**

**Optimizing Flight Booking Decisions through Machine Learning Price Prediction**

**Team ID :** NM2023TMID22068

**Team Leader :** SELVARAJ. N

**Reg. No :** 20201061506138

**Team Member :** SELVAMOORTHI. C

**Reg. No :** 20201061506136

**Team Member :** VENKATESH. M

**Reg. No :** 20201061506147

**Team Member :** KARTHICK KUMAR. V

**Reg. No :** 20201061506109

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No:** | **TITLE** | **Pg.No:** |
| 01 | Introduction | 04 |
| 02 | Problem Definition & Design Thinking | 07 |
| 03 | Result | 09 |
| 04 | Advantages &Disadvantages | 10 |
| 05 | Application | 12 |
| 06 | Conclusion | 13 |
| 07 | Future Scope | 14 |
| 08 | Appendix | 16 |

**INTRODUCTION:**

Welcome to the world of flight ticket booking optimization through machine learning.In this project, we will be exploring how we can leverage machine learning techniques to predict flight ticket prices and help customers save money on their air travel expenses.

Airline ticket prices are highly dynamic and can fluctuate rapidly based on various factors such as seasonality, demand, competition, and availability. By using machine learning algorithms, we can analyze historical data and predict future prices with a high degree of accuracy.

Our goal is to build a predictive model that can accurately forecast future ticket prices and provide recommendations to users on the optimal time to book their flights. This project will involve data collection, data cleaning, feature engineering, model selection, and performance evaluation.

Through this project, we aim to help travelers make informed decisions about their flight bookings and save money by booking at the right time. We look forward to embarking on this exciting journey with you!learning models will become even more accurate and effective in predicting flight delays, enabling the industry to operate more efficiently and reliably.

Optimizing flight ticket booking through machine learning price prediction is a project that aims to improve the process of purchasing airline tickets by leveraging machine learning algorithms to predict ticket prices. The goal is to enable travelers to make informed decisions about when to book their tickets and how much to pay for them.

The project will involve collecting and analyzing historical data on flight prices, as well as information about various factors that affect ticket prices, such as time of year, airline, and route. Using this data, the machine learning model will be trained to accurately predict future ticket prices.

The benefits of this project are numerous. For travelers, it can help them save money by identifying the optimal time to book tickets. For airlines, it can improve their revenue management by allowing them to better predict demand and adjust prices accordingly. Overall, this project has the potential to revolutionize the way people purchase airline tickets, making it more convenient and cost-effective.

**Overview:**

Flight booking decisions can be difficult, with travelers often struggling to balance their budget constraints with their preferred travel dates and airline preferences. Machine learning (ML) can help travelers make more informed decisions by predicting flight prices based on historical data and current market trends.

ML algorithms can analyze large amounts of data from various sources, such as airline websites, travel agencies, and social media platforms, to identify patterns and trends that influence flight prices. By using these patterns, algorithms can make accurate predictions on how prices will change in the future, enabling travelers to make more informed decisions on when and where to book their flights.

With the help of ML-powered price prediction tools, travelers can optimize their flight booking decisions and save money by booking at the right time and choosing the best airline and travel dates for their needs. Additionally, airlines and travel agencies can benefit from these tools by improving their pricing strategies and increasing customer satisfaction.

Overall, the use of ML in flight booking can greatly improve the travel experience for both travelers and businesses, making it easier and more efficient to book flights and travel to new destinations**.**

**Purpose:**

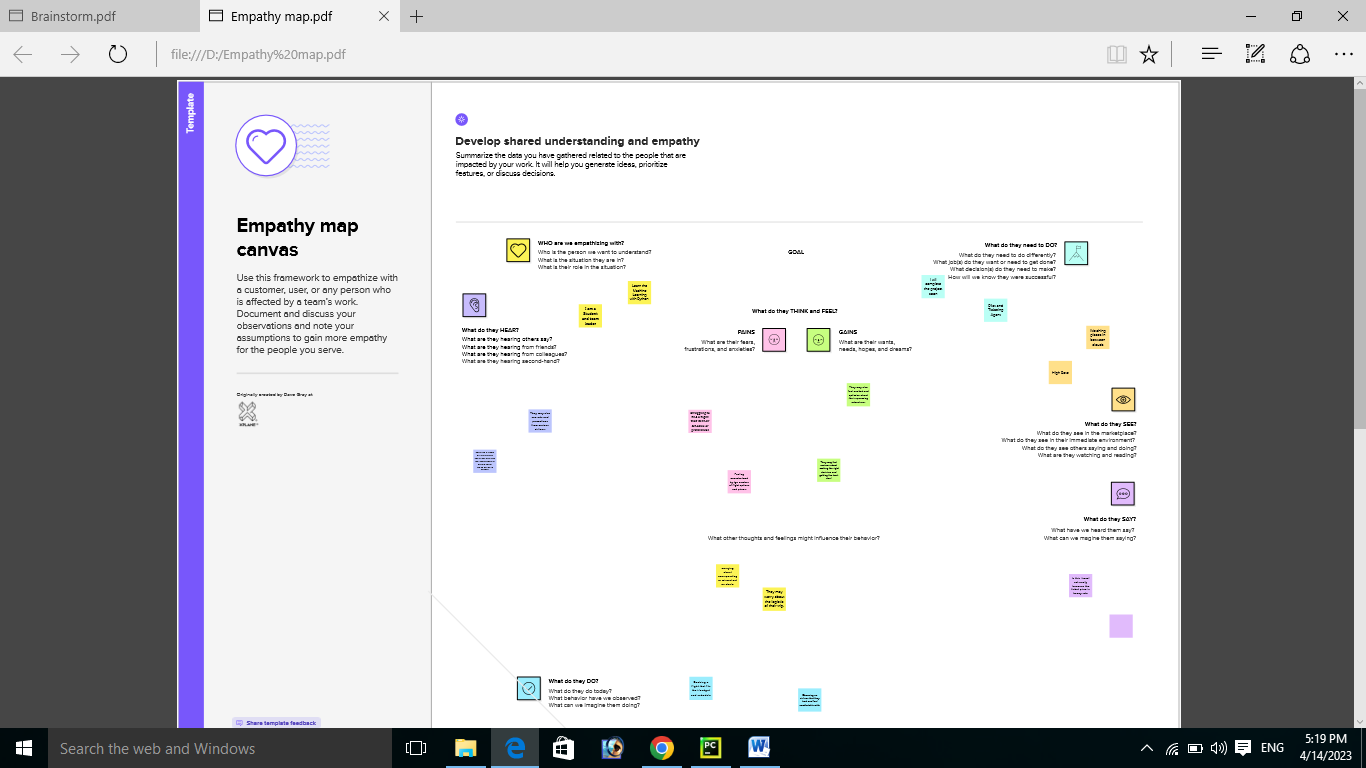
The purpose of using machine learning for optimizing flight booking decisions through price predictions is to help travelers make more informed decisions about their flights, leading to cost savings and a better travel experience. The project aims to provide accurate predictions of flight prices based on historical data and current market trends, which can help travelers determine the best time to book their flights and choose the most cost-effective travel dates and airlines.

The use of machine learning in this project also benefits airlines and travel agencies by enabling them to optimize their pricing strategies, increase customer satisfaction, and gain a competitive edge in the industry.

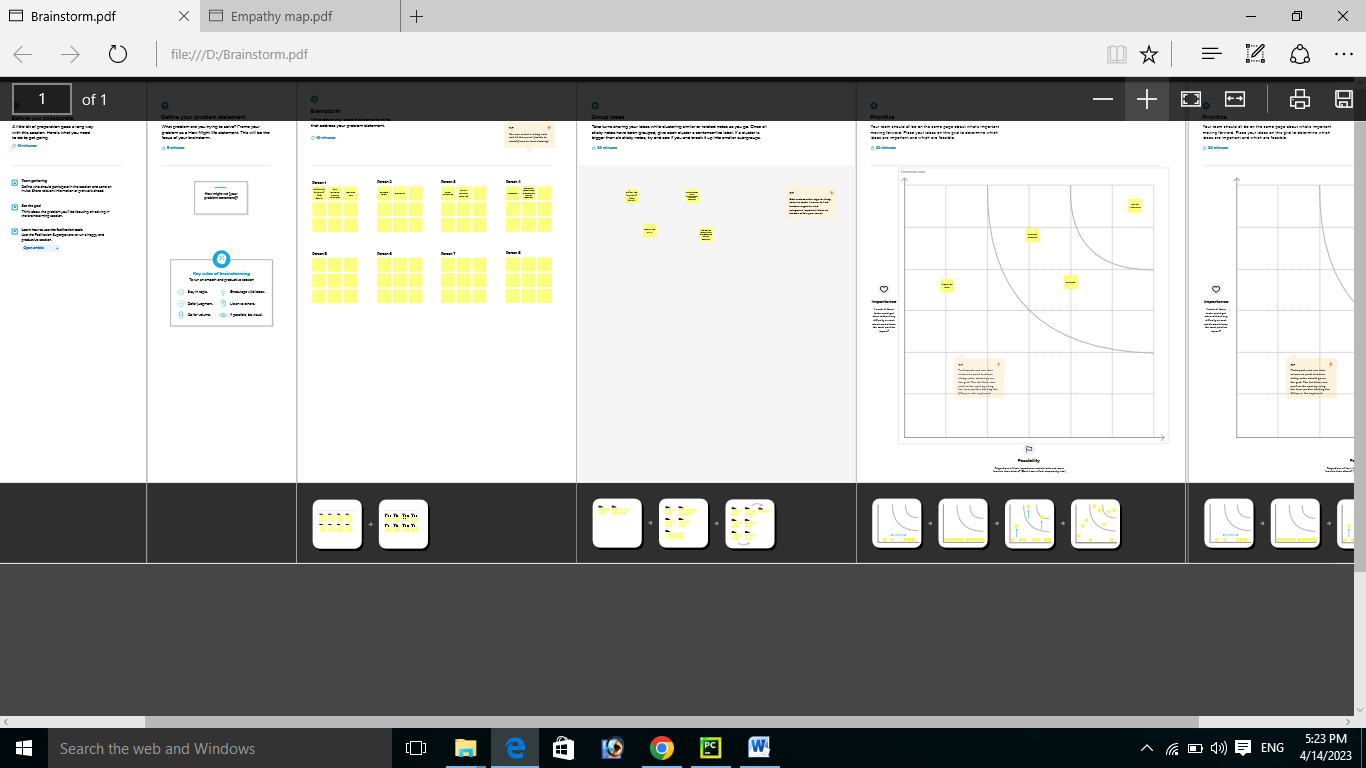
Overall, the purpose of this project is to leverage the power of machine learning to improve the travel experience for both travelers and businesses, making it easier and more efficient to book flights and travel to new destinations.

**Problem definition & designing thinking:**

***Empathy map:***



***Brainstroming Map:***



**Result:**

The use of machine learning for optimizing f;light booking decisoons through price prediction has the potential is the revolitionize the travel industry. By analysing large amounts of data from various sources, machine learning algorithms can make accures price predictions and help travellers make more informed decisions about their flights.

**Output: (final output)**



**Advantages:**

**Cost savings:**

Machine learning price prediction models can analyze a vast amount of data and predict the best time to buy tickets, which can help customers save money on their flight bookings.

**Improved customer satisfaction:**

If customers can book flights at lower prices, they are more likely to be satisfied with the booking experience and the airline brand.

**Competitive advantage:**

Airlines that offer accurate price predictions through machine learning can gain a competitive advantage in the market, as customers will be more likely to choose their services over others.

**Efficient use of resources:**

By predicting the demand for flights, airlines can optimize their resources, including planes and crew, reducing the risk of flying with empty seats**.**

**Disadvantages:**

**Limited accuracy:**

Machine learning algorithms are only as good as the data they are trained on, and there is always a risk of inaccurate predictions. This can lead to customer frustration and dissatisfaction if they miss out on cheaper flights.

**Limited data availability:**

Access to real-time data on flight bookings and cancellations can be limited, which can impact the accuracy of machine learning models.

**Ethical concerns:**

There is a risk that airlines may use the information they gather through machine learning algorithms to engage in price discrimination or other unethical practices.

**Technical challenges:**

Implementing a machine learning algorithm can be technically challenging, requiring significant investment in data analysis, infrastructure, and expertise.

**Applications:**

The application of this project is in the travel industry, particularly in the areas of online travel agencies, airline websites, and travel search engines. By implementing machine learning algorithms for flight booking decisions, companies can provide their customers with more accurate and personalized recommendations for flights, resulting in cost savings and an enhanced travel experience.

This also extends to airlines and travel agencies, as they can use machine learning algorithms to optimize their pricing strategies and improve their revenue management. By analyzing data on customer behavior and market trends, companies can adjust their prices in real-time to attract more customers and increase profitability.

By using machine learning algorithms to analyze data from multiple sources, companies can provide their customers with more comprehensive travel recommendations and an overall better travel experience.

In travel industry, where machine learning algorithms can be used to optimize flight booking decisions, improve revenue management, and enhance the travel experience for both businesses and customers.

**Conclusion:**

By analyzing large amounts of data from various sources, machine learning algorithms can make accurate price predictions and help travelers make more informed decisions about their flights.

The benefits of using machine learning for flight booking decisions include increased accuracy, time and cost savings, customization, and improved pricing strategies for airlines and travel agencies. However, there are potential drawbacks, such as the need for accurate and up-to-date data, technical expertise, unforeseen events, and privacy concerns.

Despite these challenges, the application of machine learning for flight booking decisions is a promising development for the travel industry. Companies that adopt these technologies are likely to gain a competitive advantage by providing better pricing strategies and personalized recommendations to their customers, resulting in increased customer satisfaction and loyalty.

**Future Scope:**

The future scope of this project is vast and promising, with the potential to further improve the travel experience for both travelers and businesses.

**Integration with augmented reality:**

The integration of machine learning algorithms with augmented reality could help travelers to visualize and interact with their travel plans in real-time, providing a more immersive and personalized travel experience.

**Expansion to other modes of transportation:**

The application of machine learning for optimizing travel decisions can be extended beyond flight bookings to include other modes of transportation such as trains, buses, and taxis. This would allow travelers to make more informed decisions about their travel plans and choose the most cost-effective and efficient modes of transportation.

**Integration with smart cities**:

The integration of machine learning algorithms with smart city infrastructure could help travelers to navigate their destinations more efficiently, providing real-time traffic information, parking recommendations, and other helpful information.

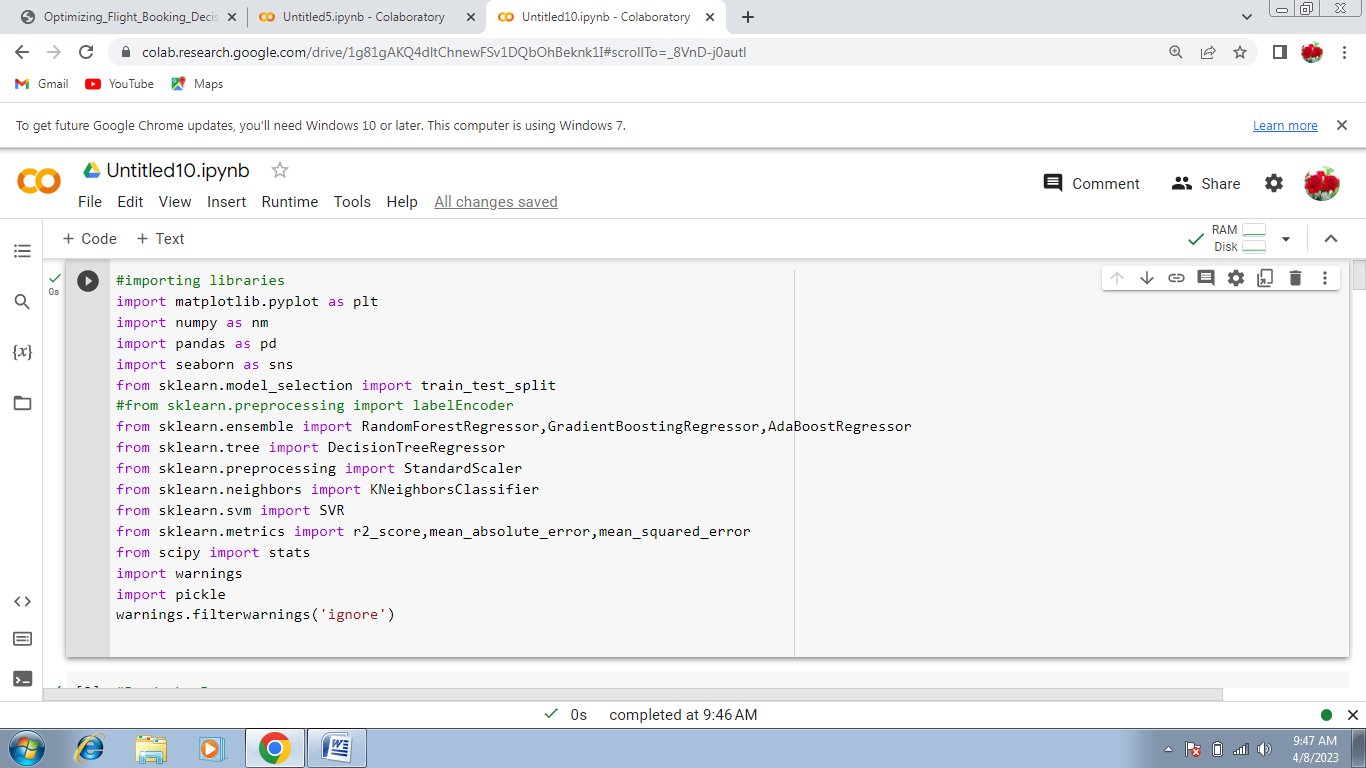
**Personalized travel recommendations:**

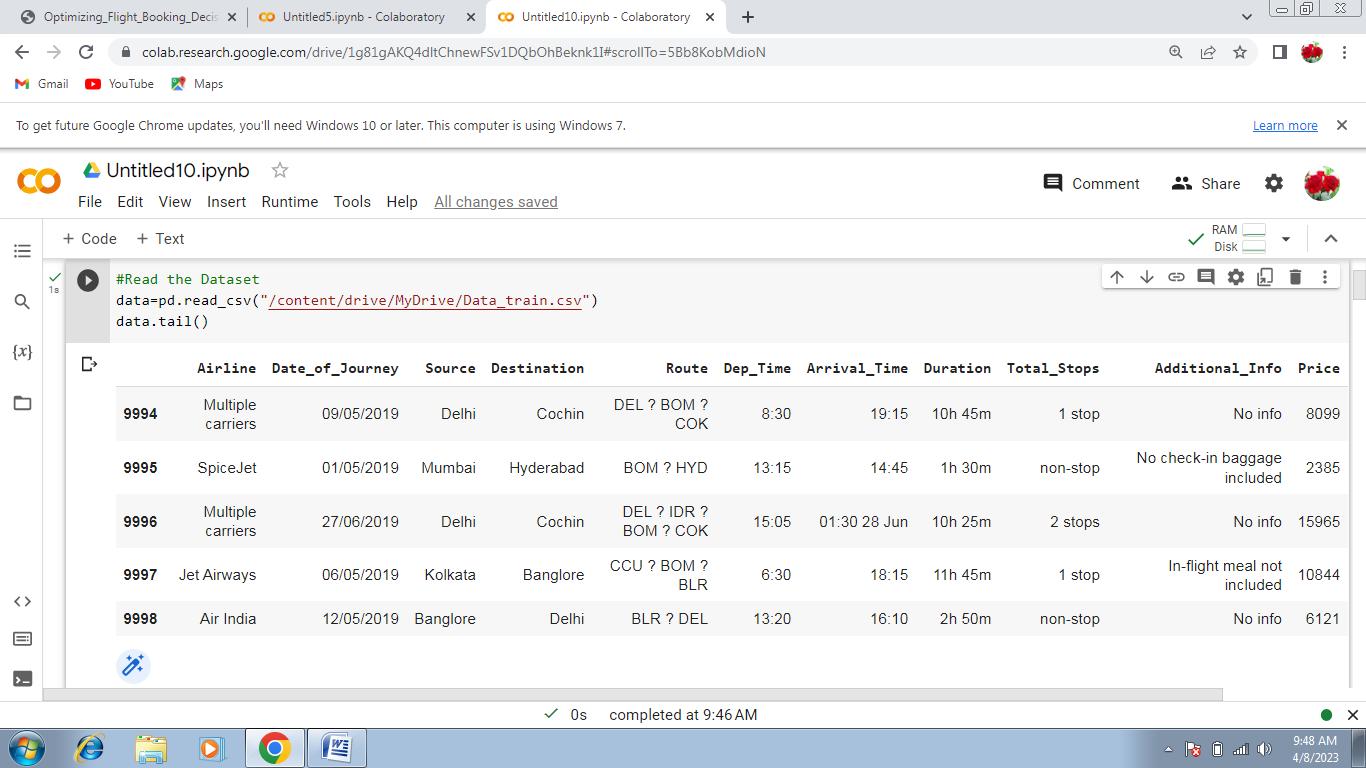
As machine learning algorithms become more advanced, they will be able to provide more personalized travel recommendations based on a traveler's preferences and past travel history. This would allow companies to offer customized travel packages and improve customer loyalty.

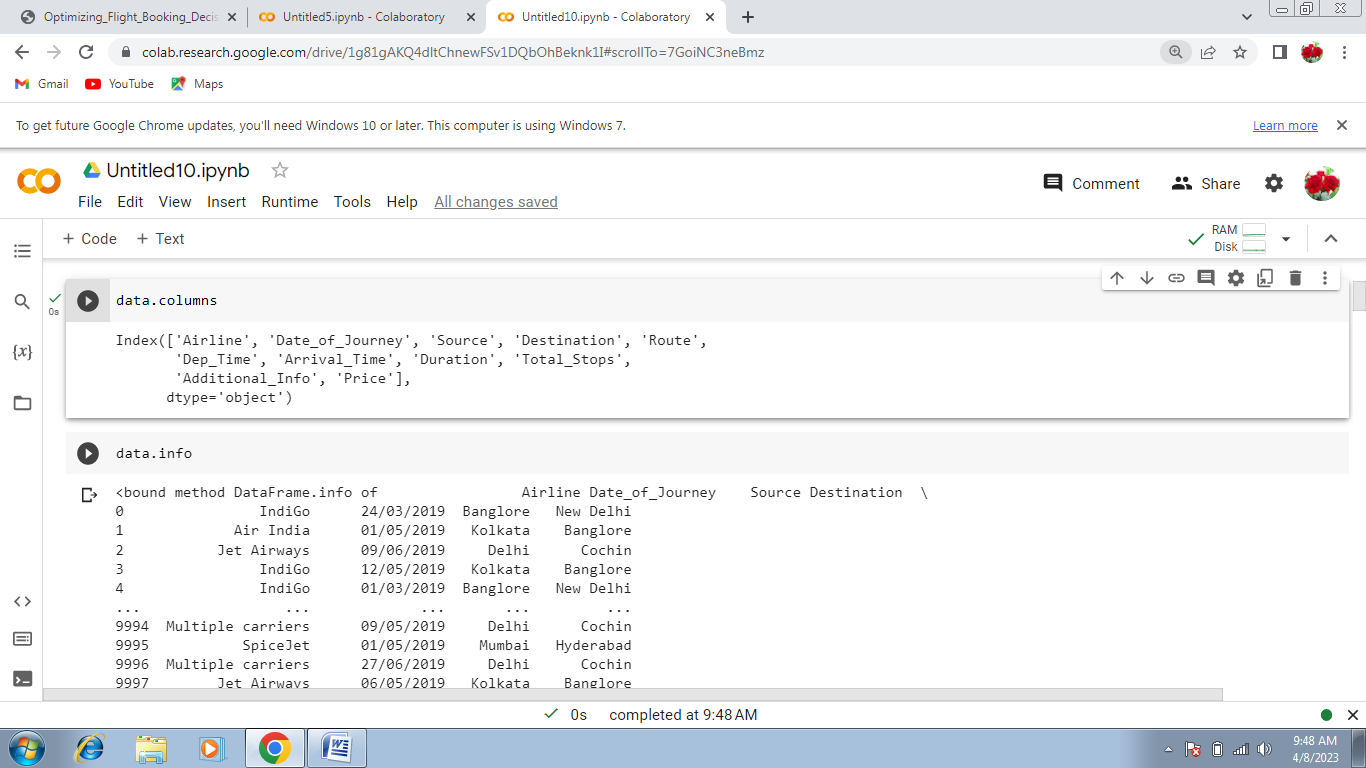
**Increased use of natural language processing:** The use of natural language processing in conjunction with machine learning algorithms could enable travelers to interact with virtual travel assistants using natural language, providing a more intuitive and user-friendly experience.

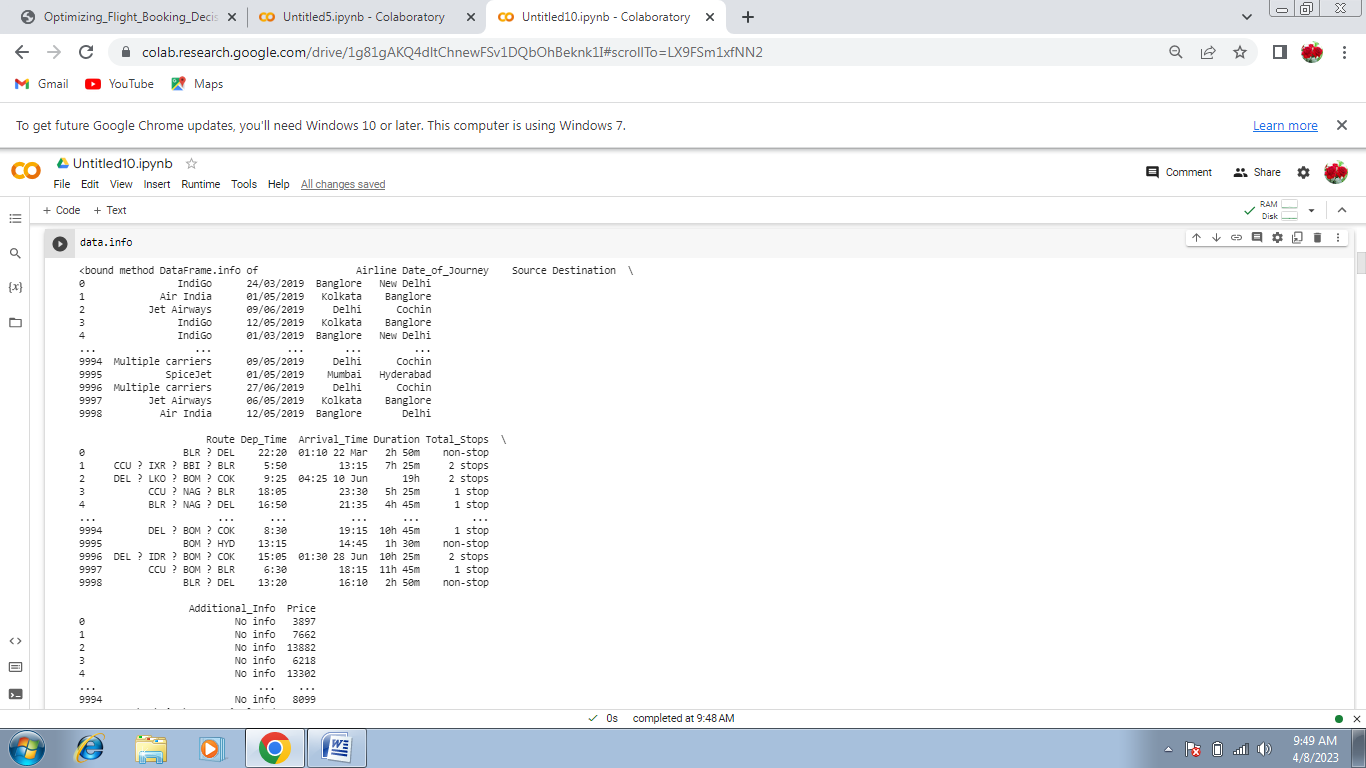
**Appendix :**

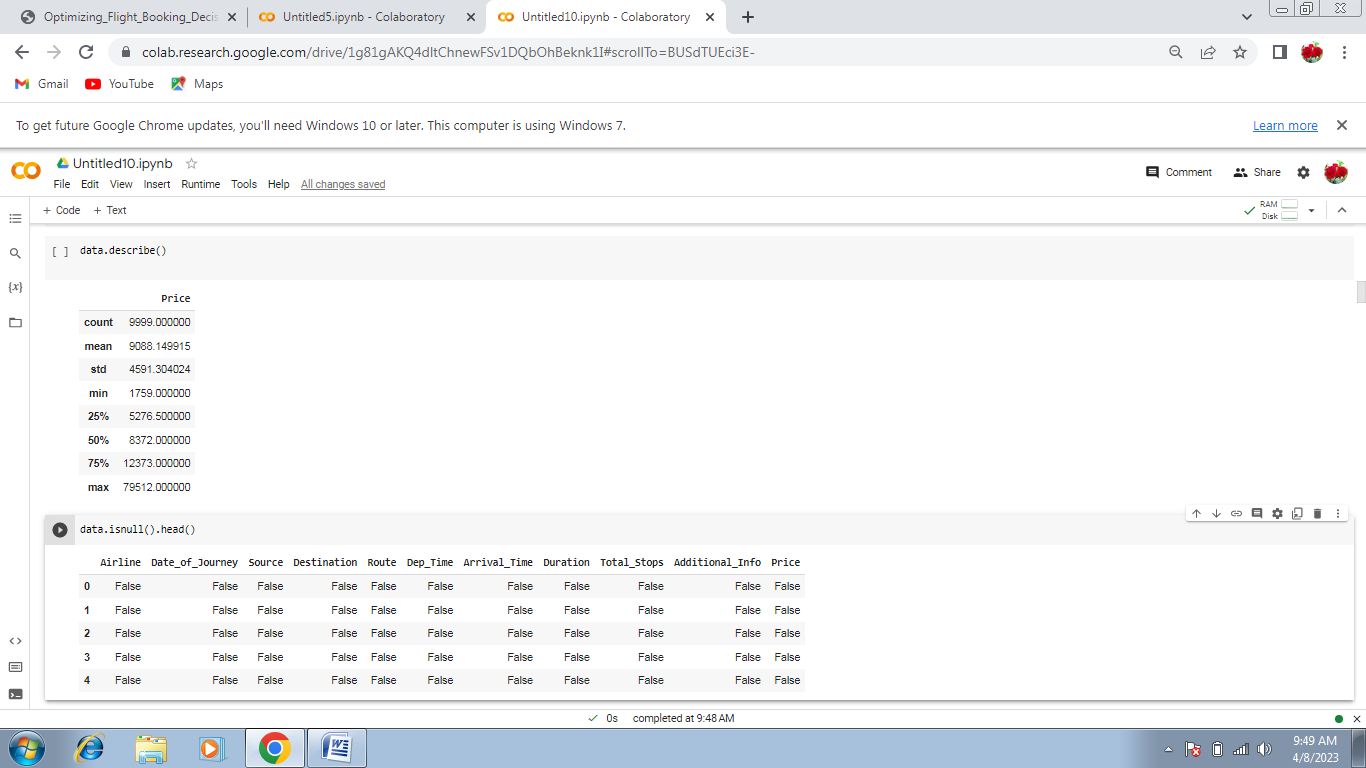
Source Code :

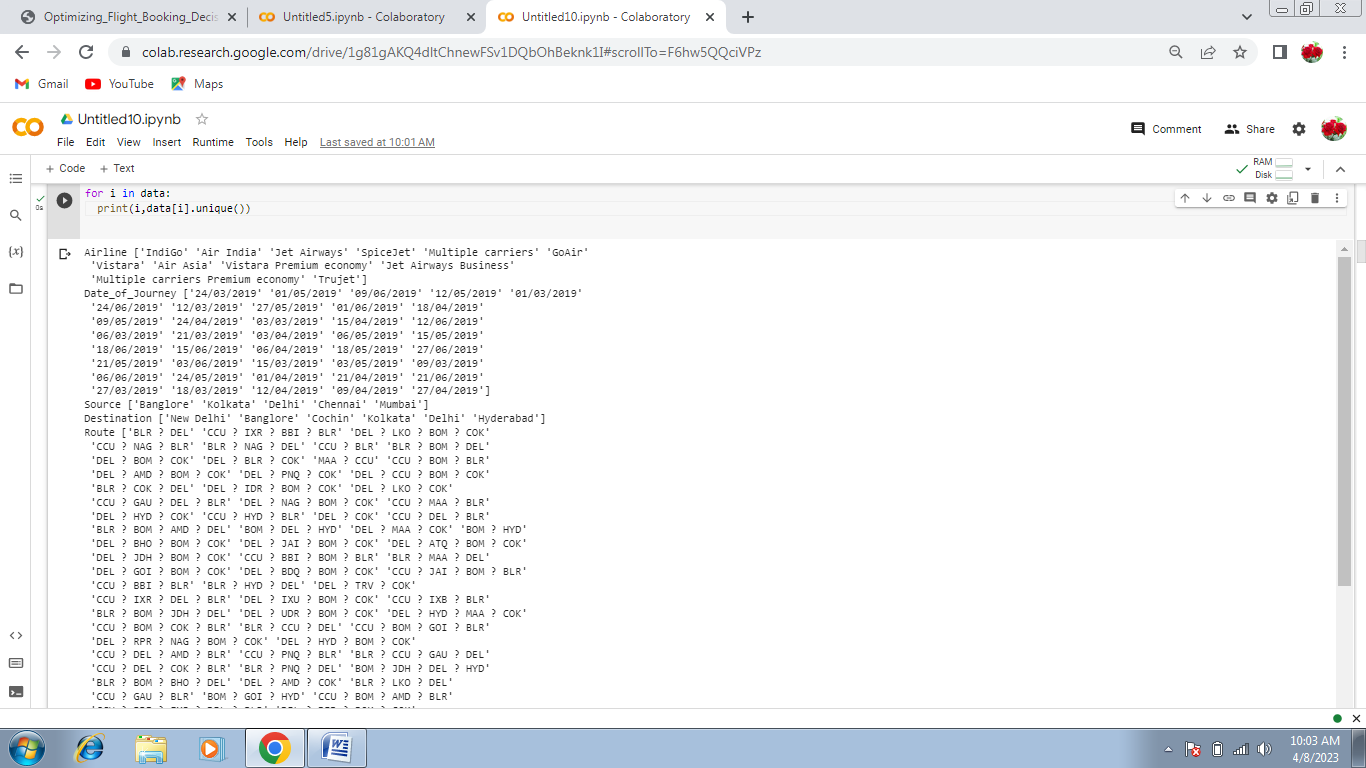


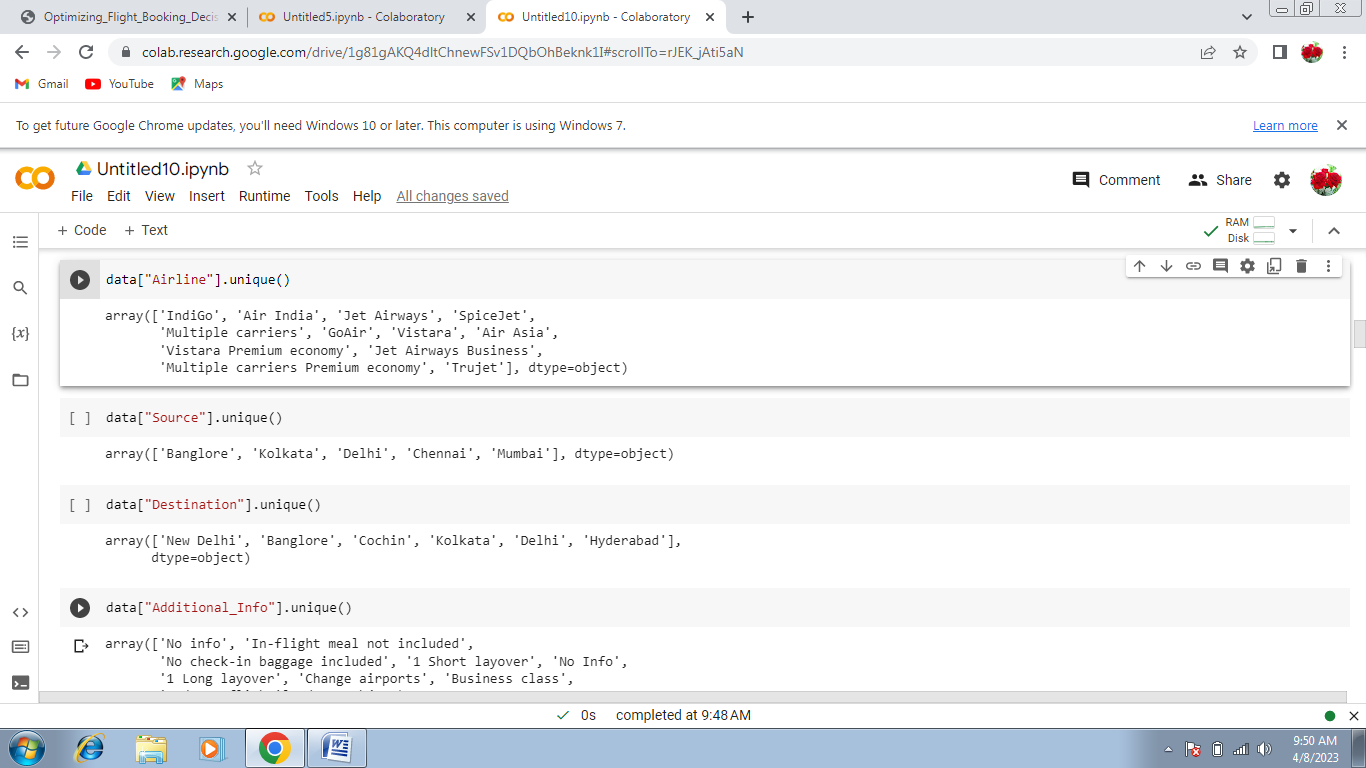


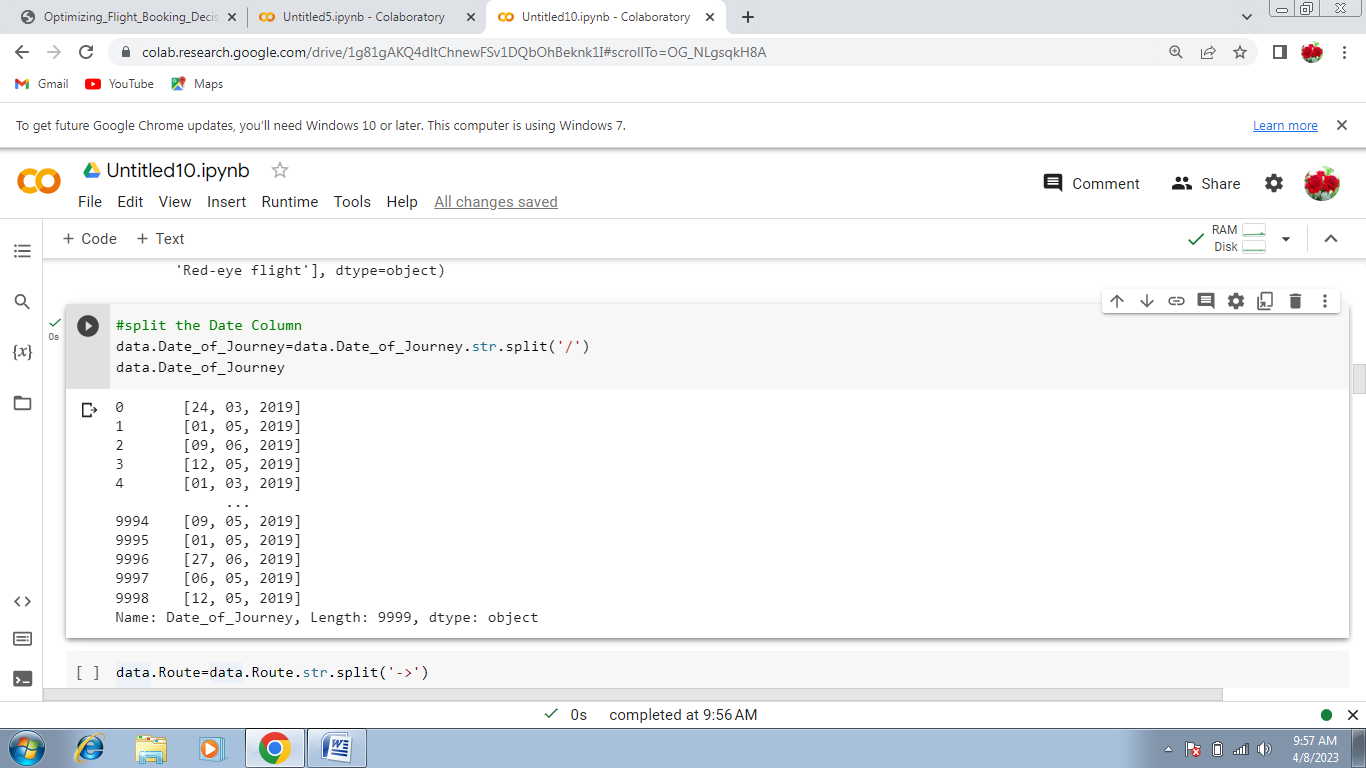


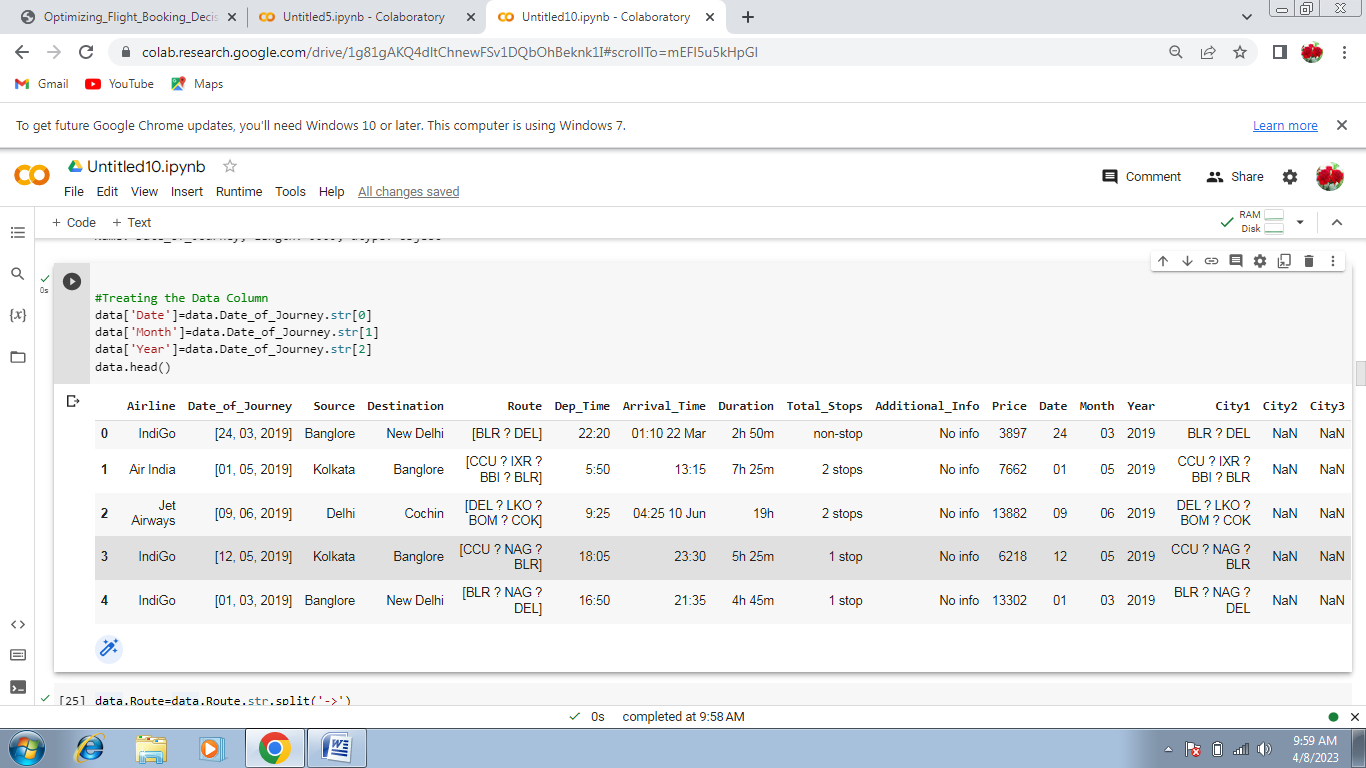


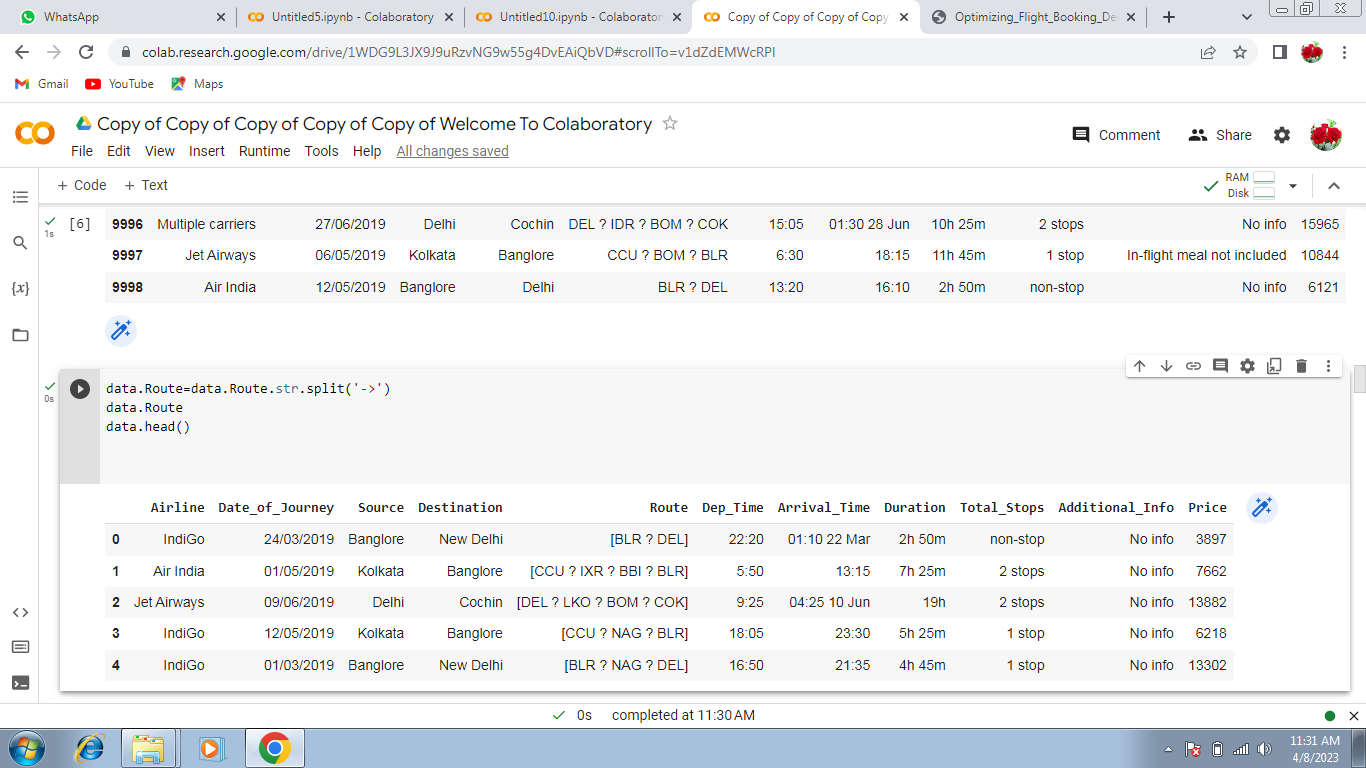


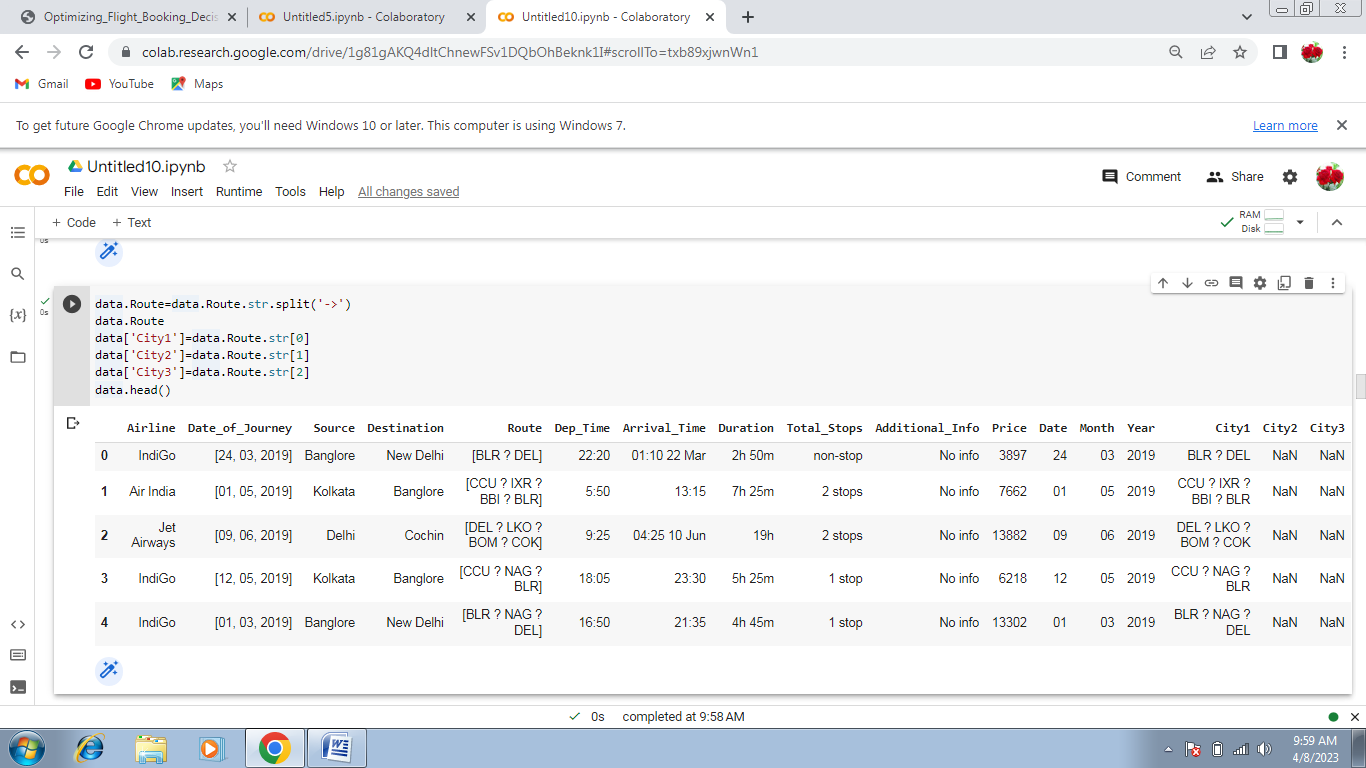


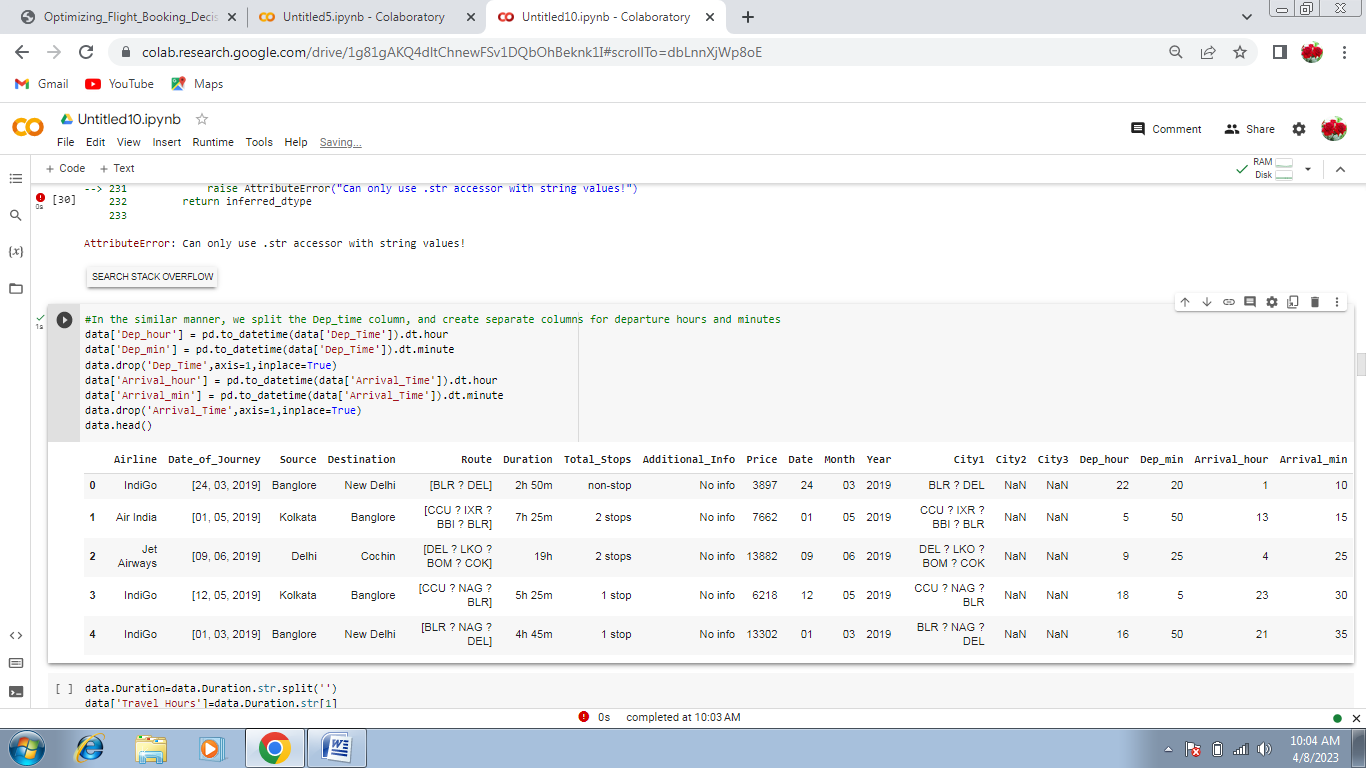


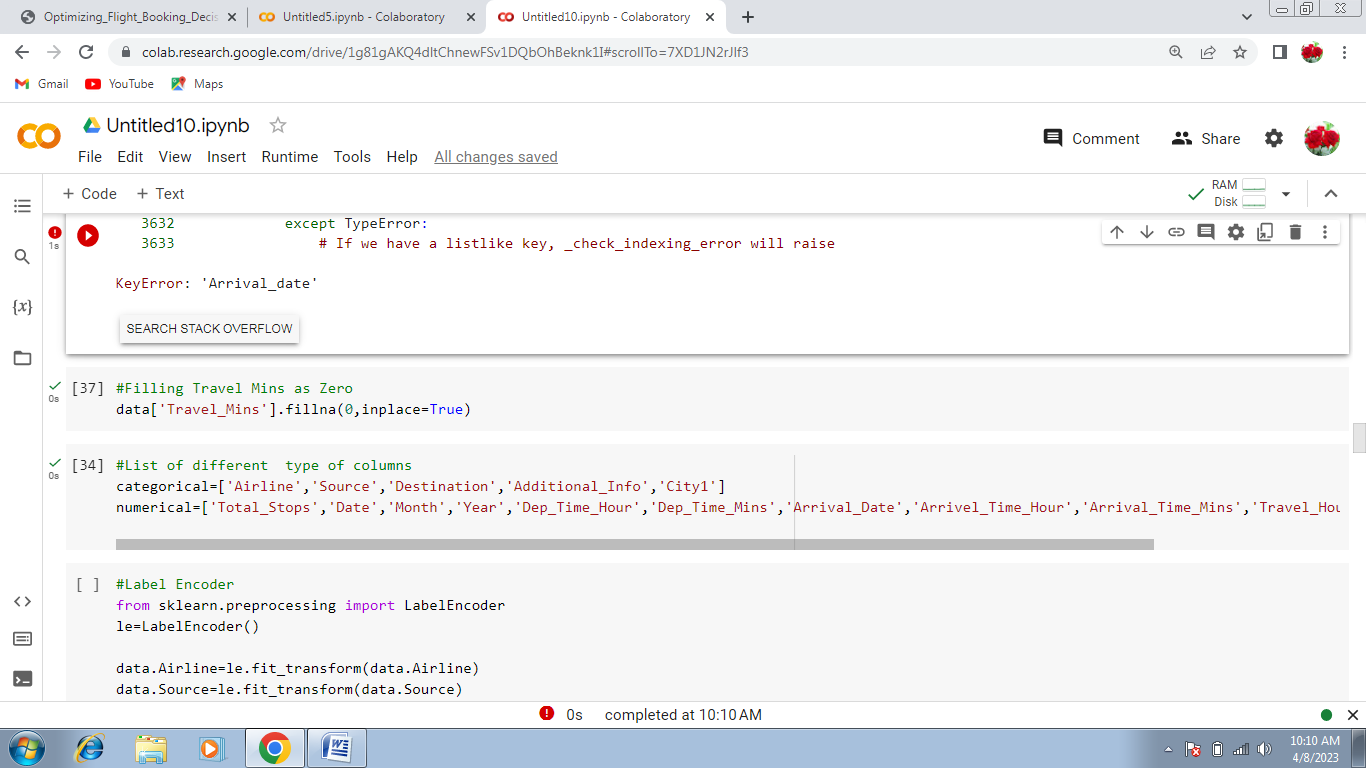


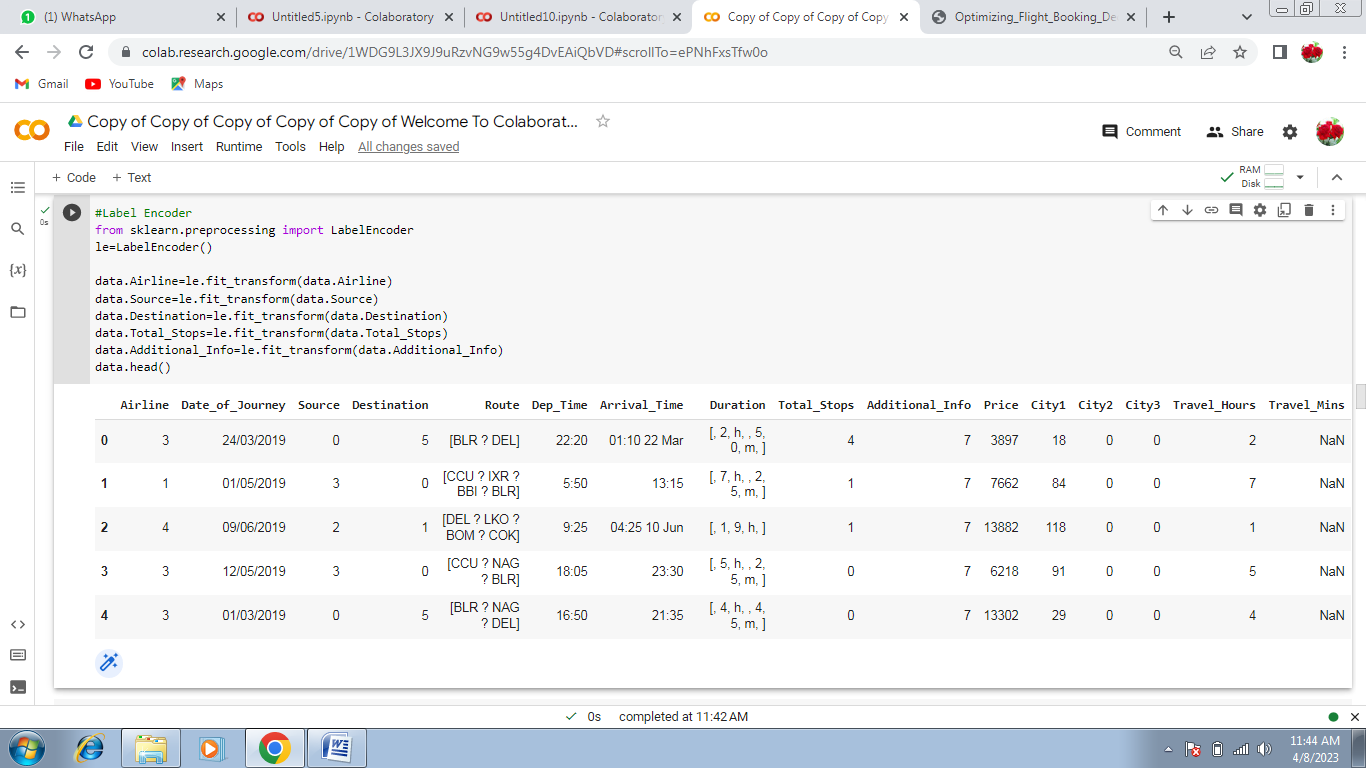


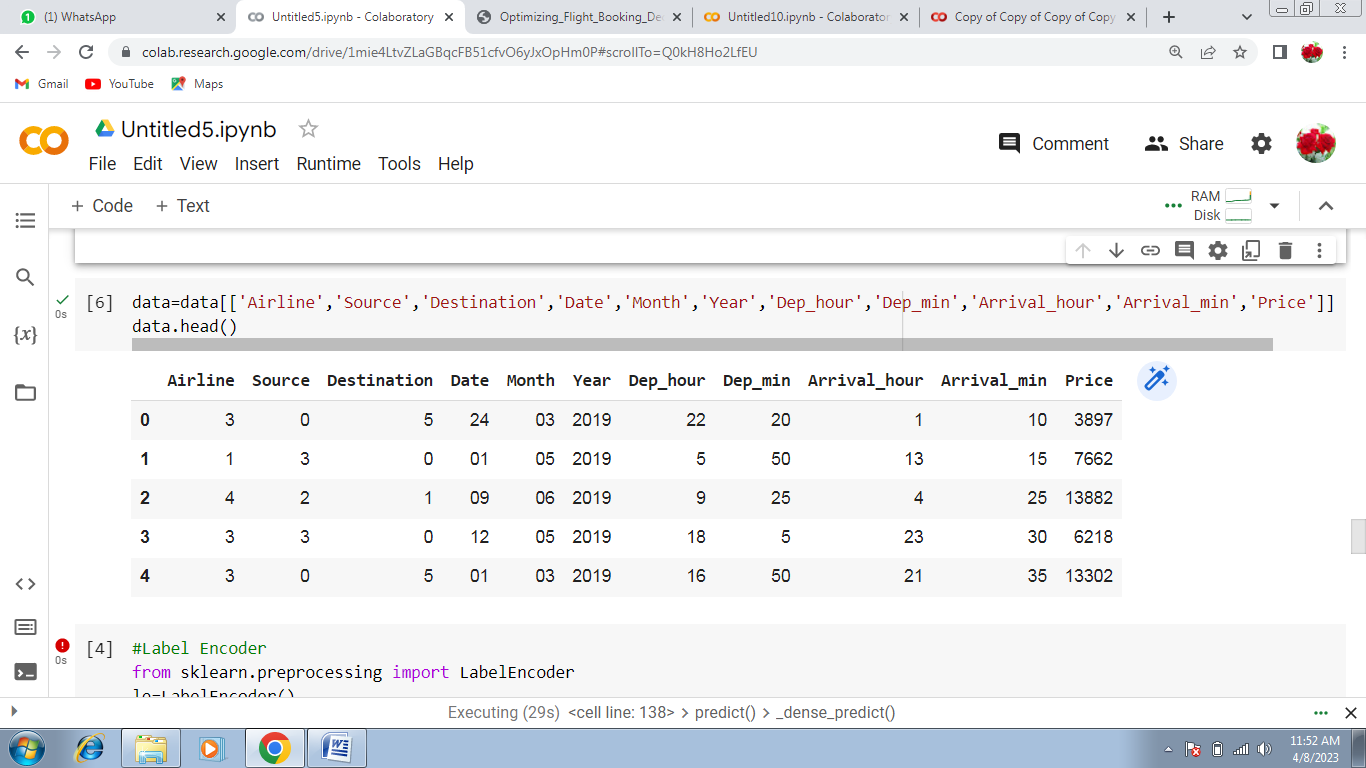


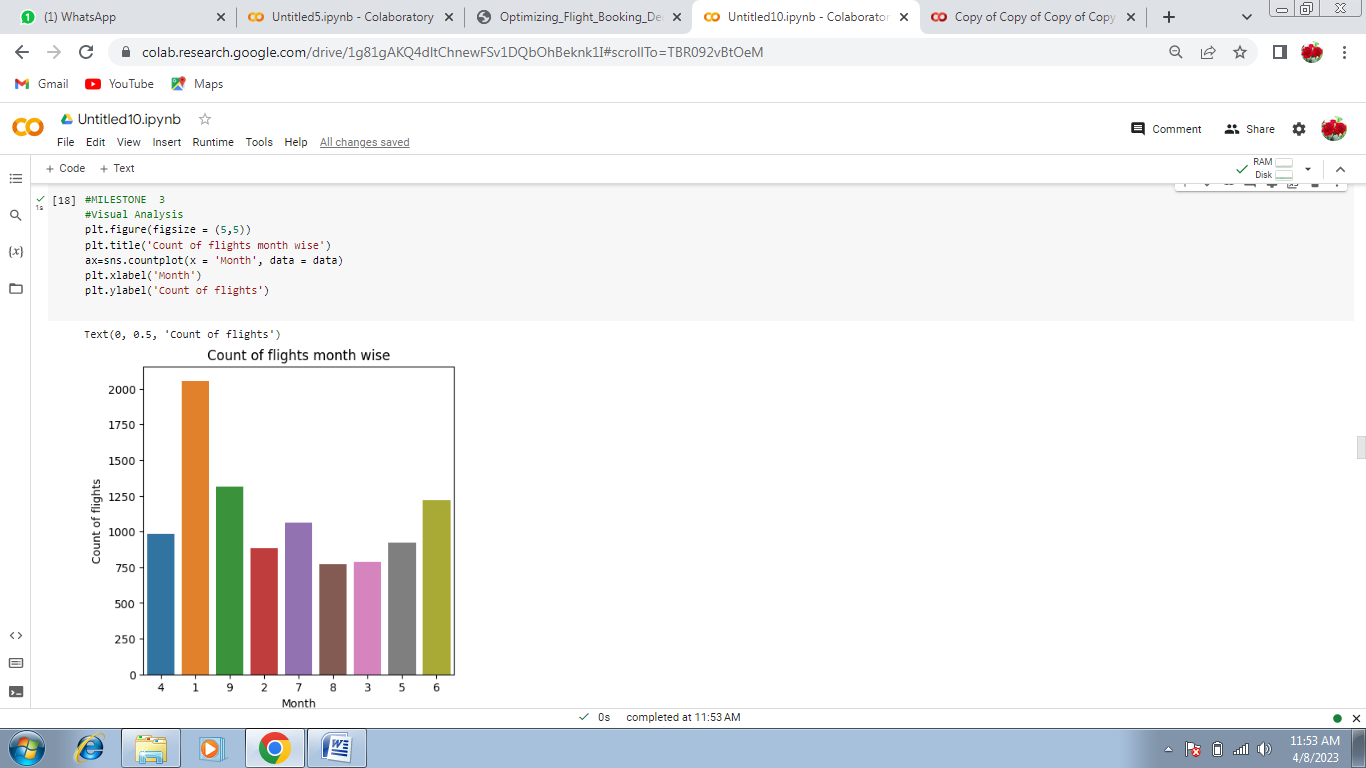


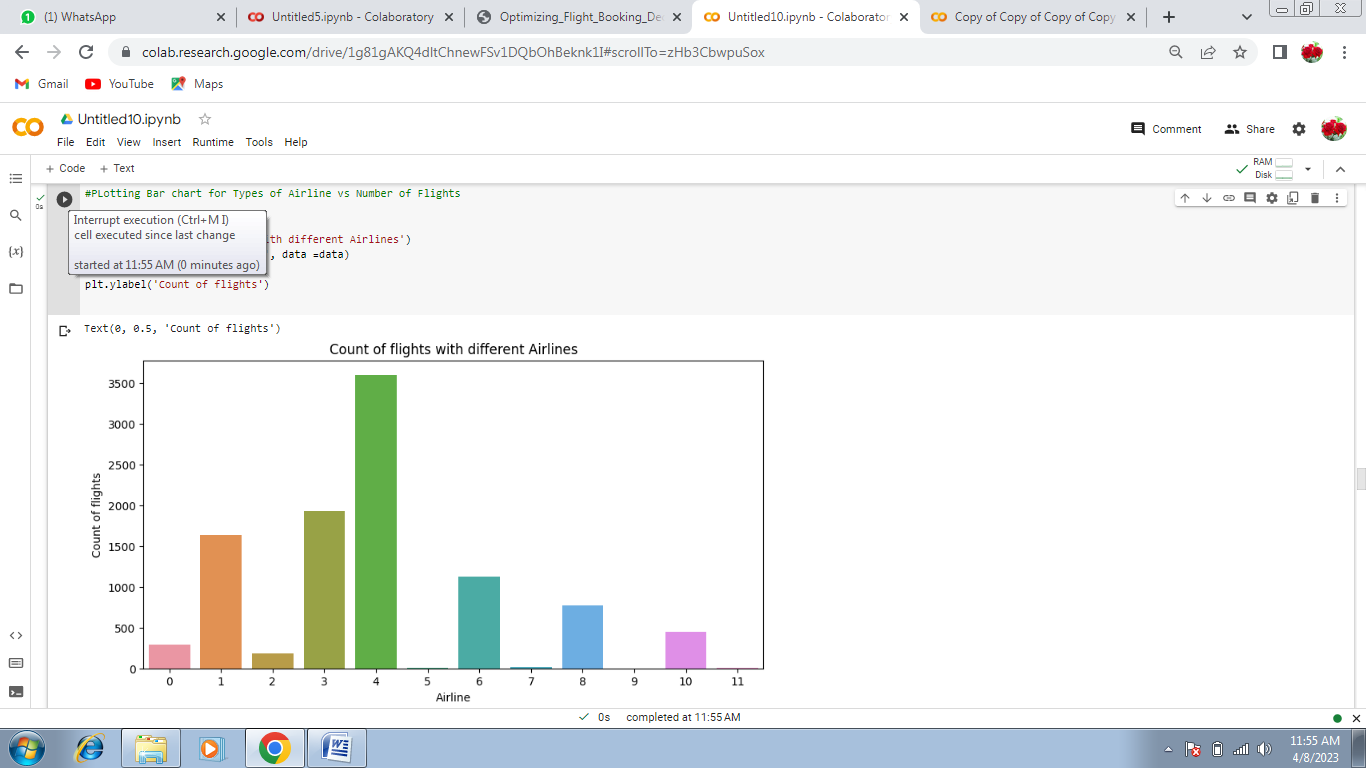


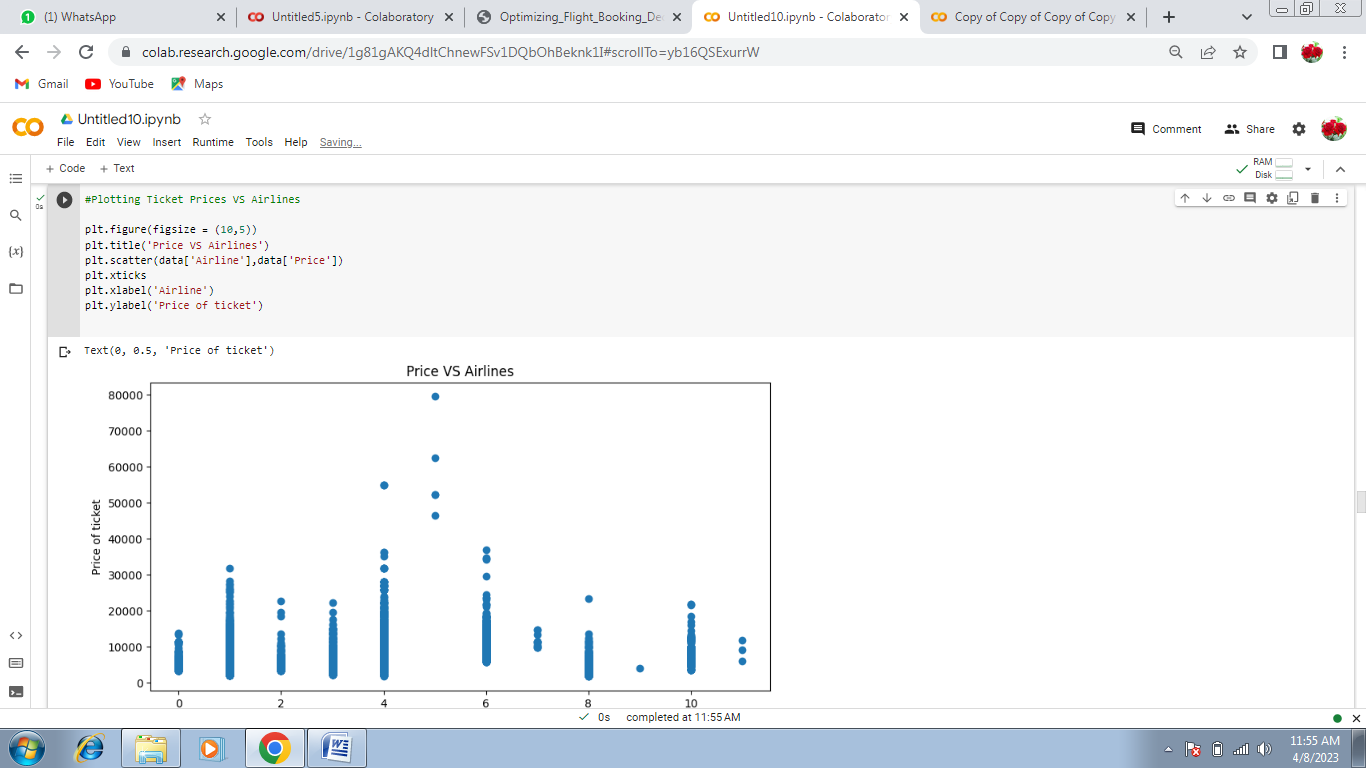


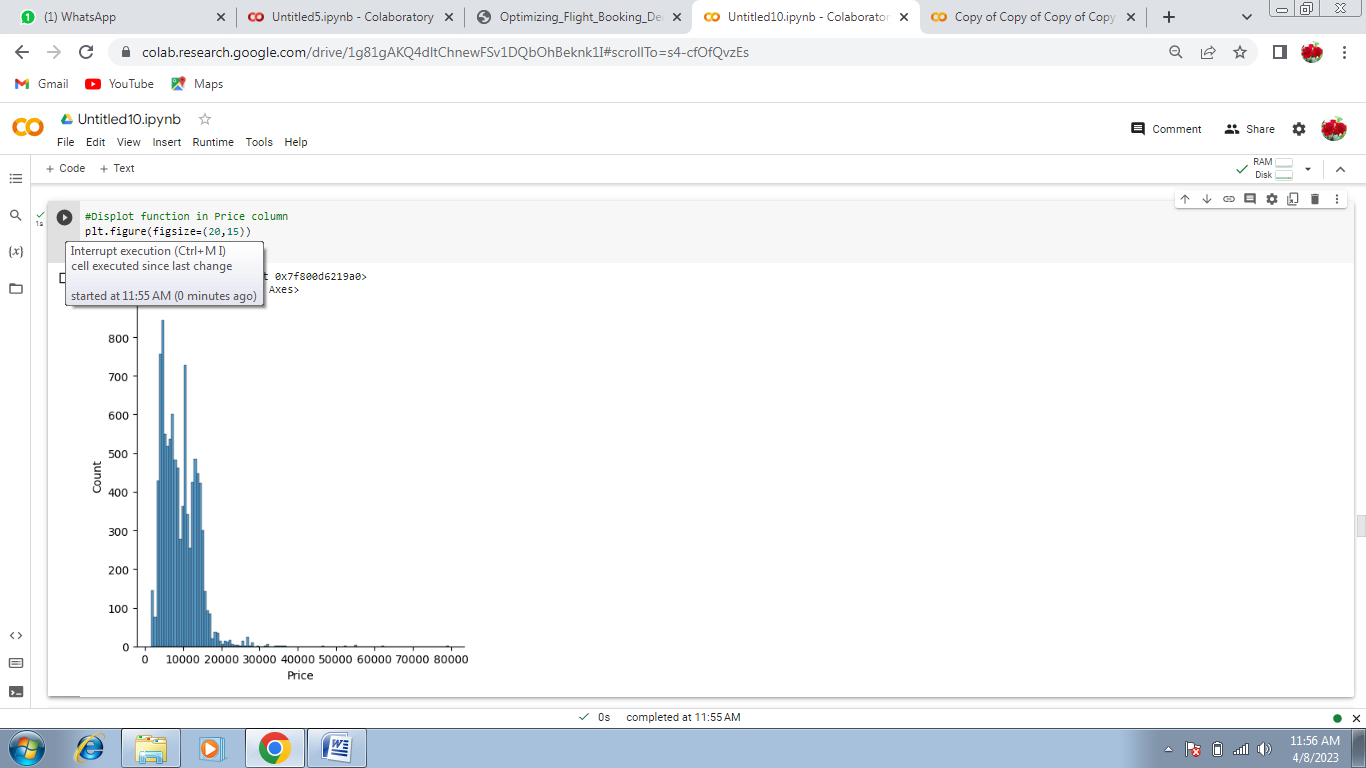




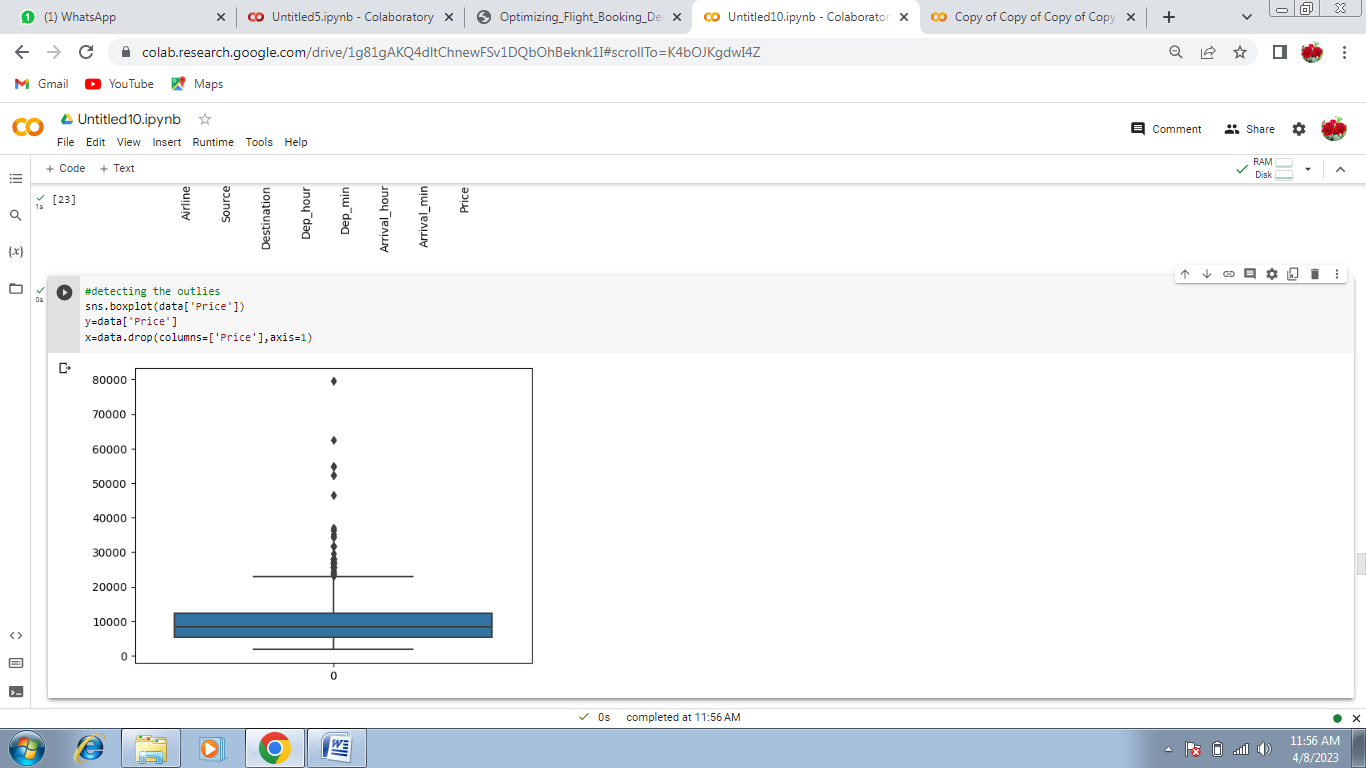


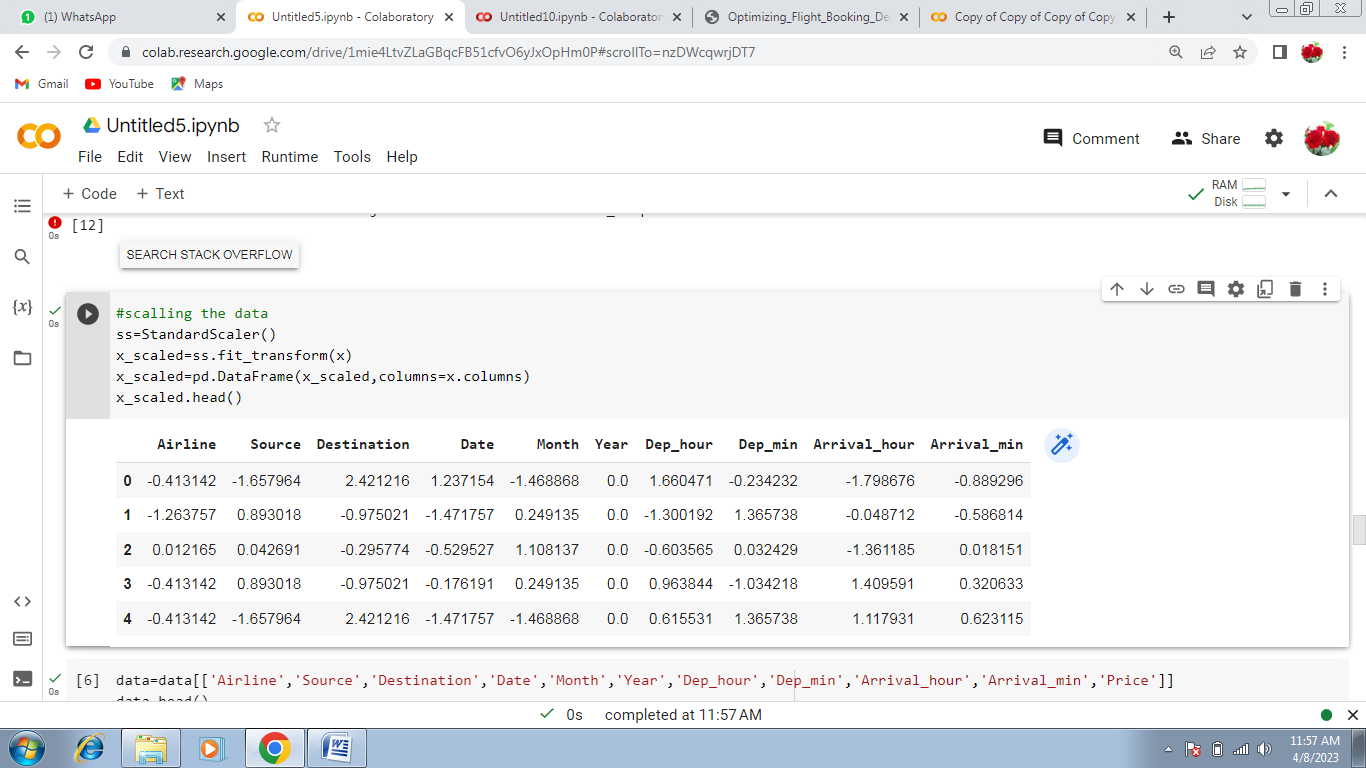


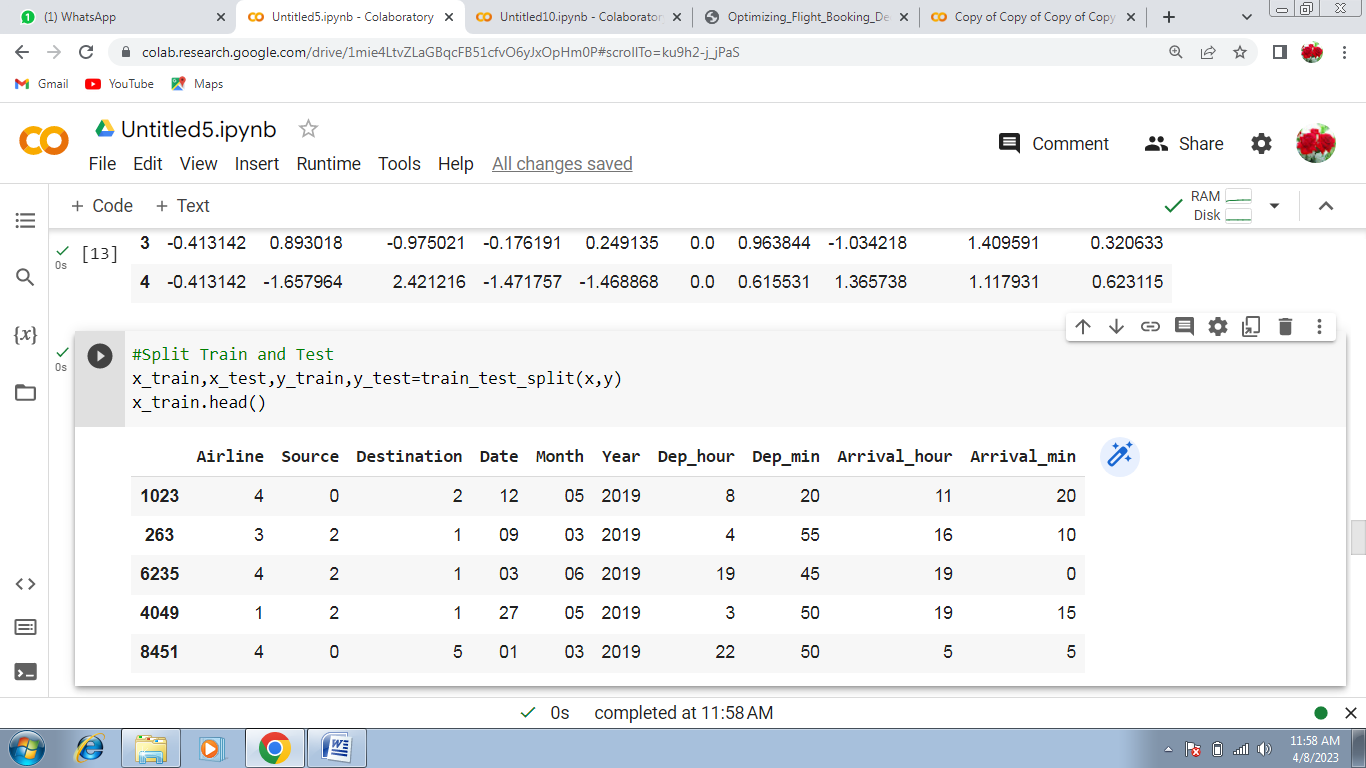


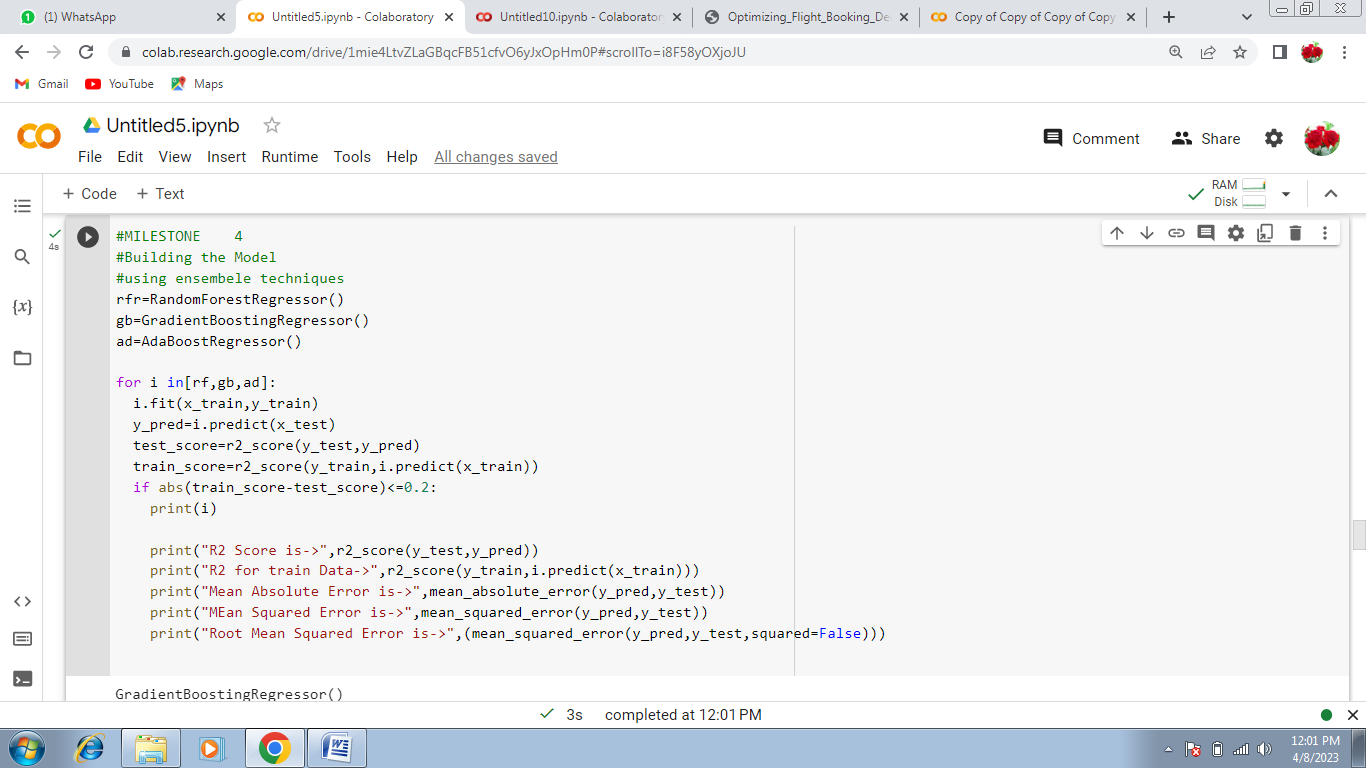


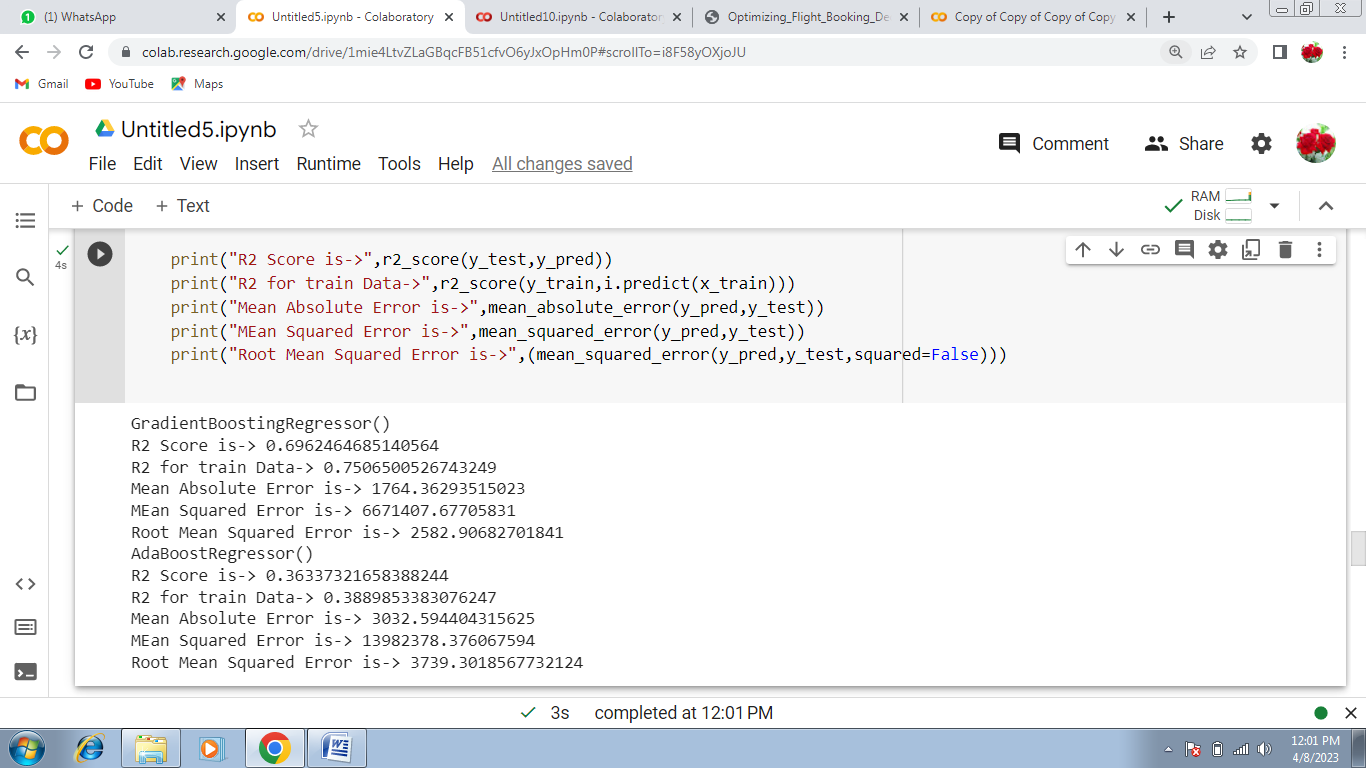


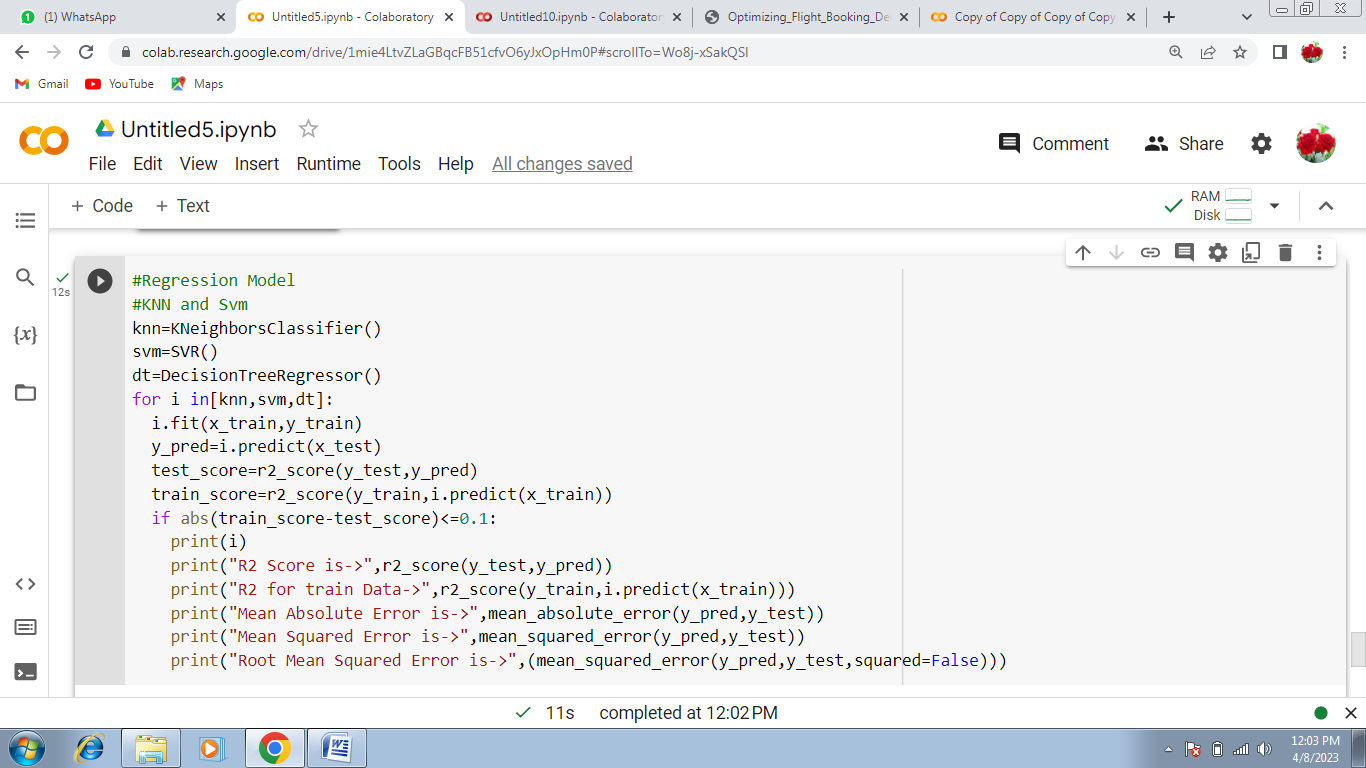


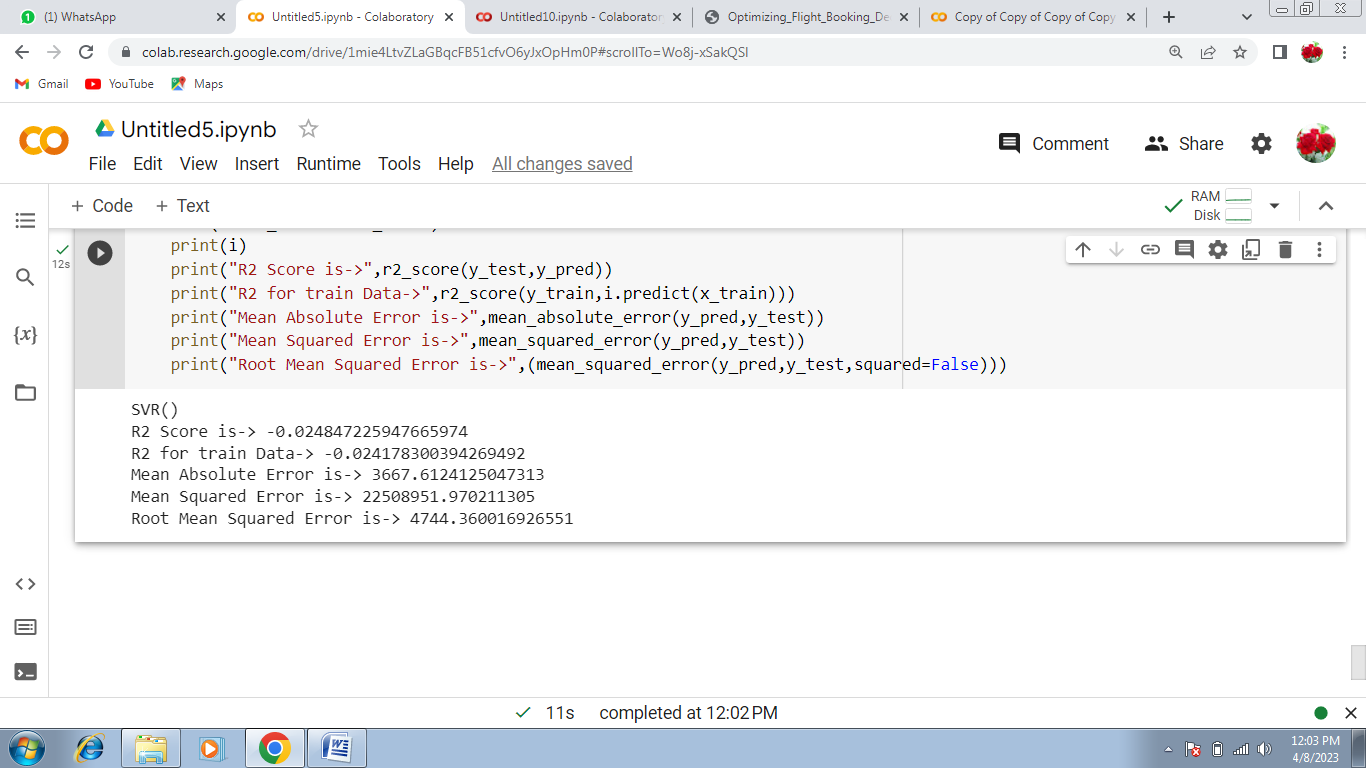


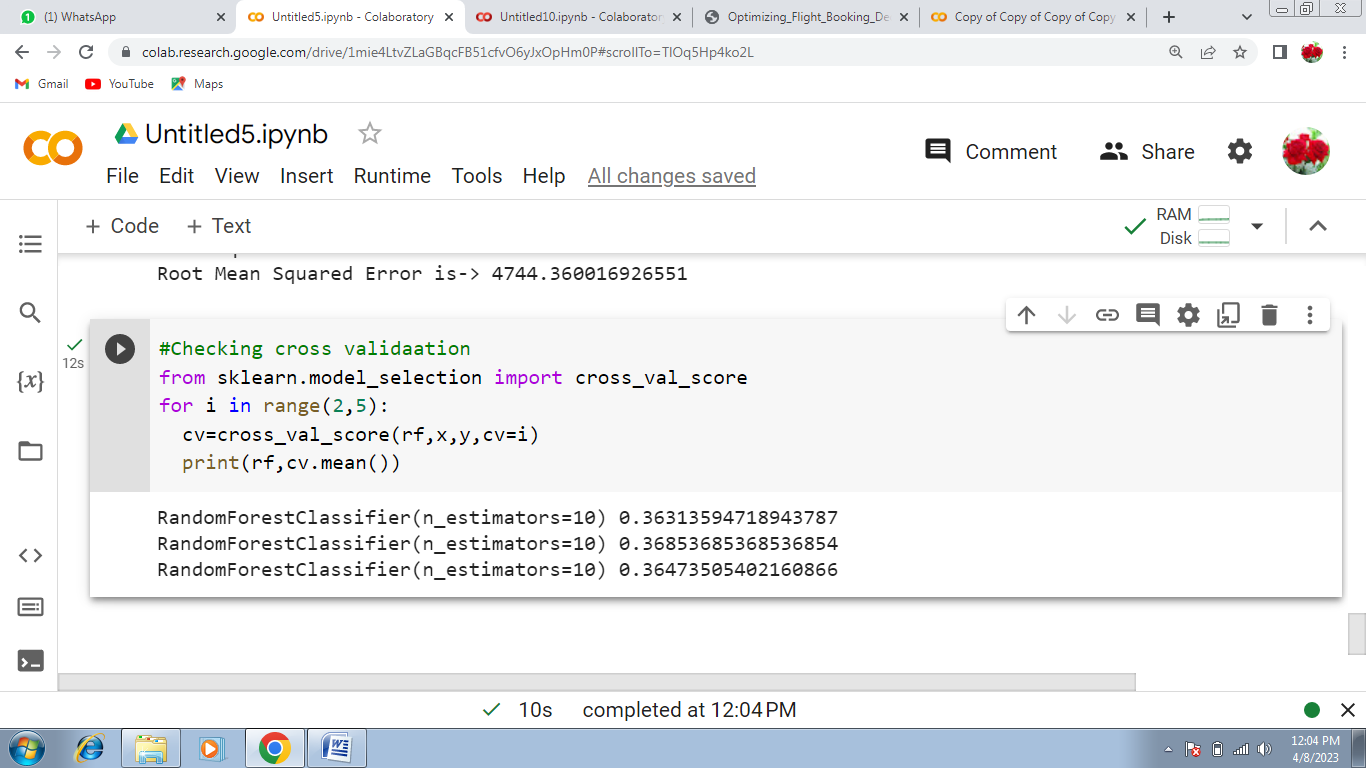


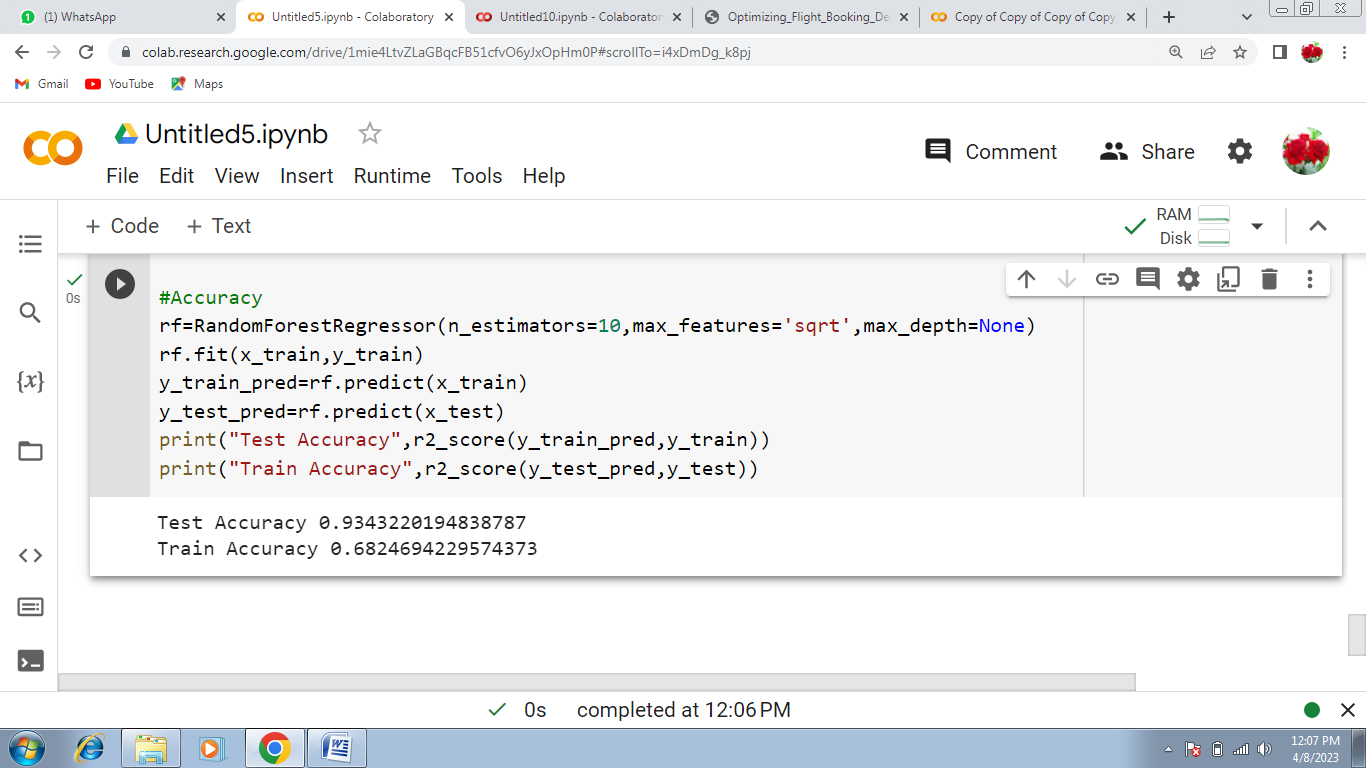








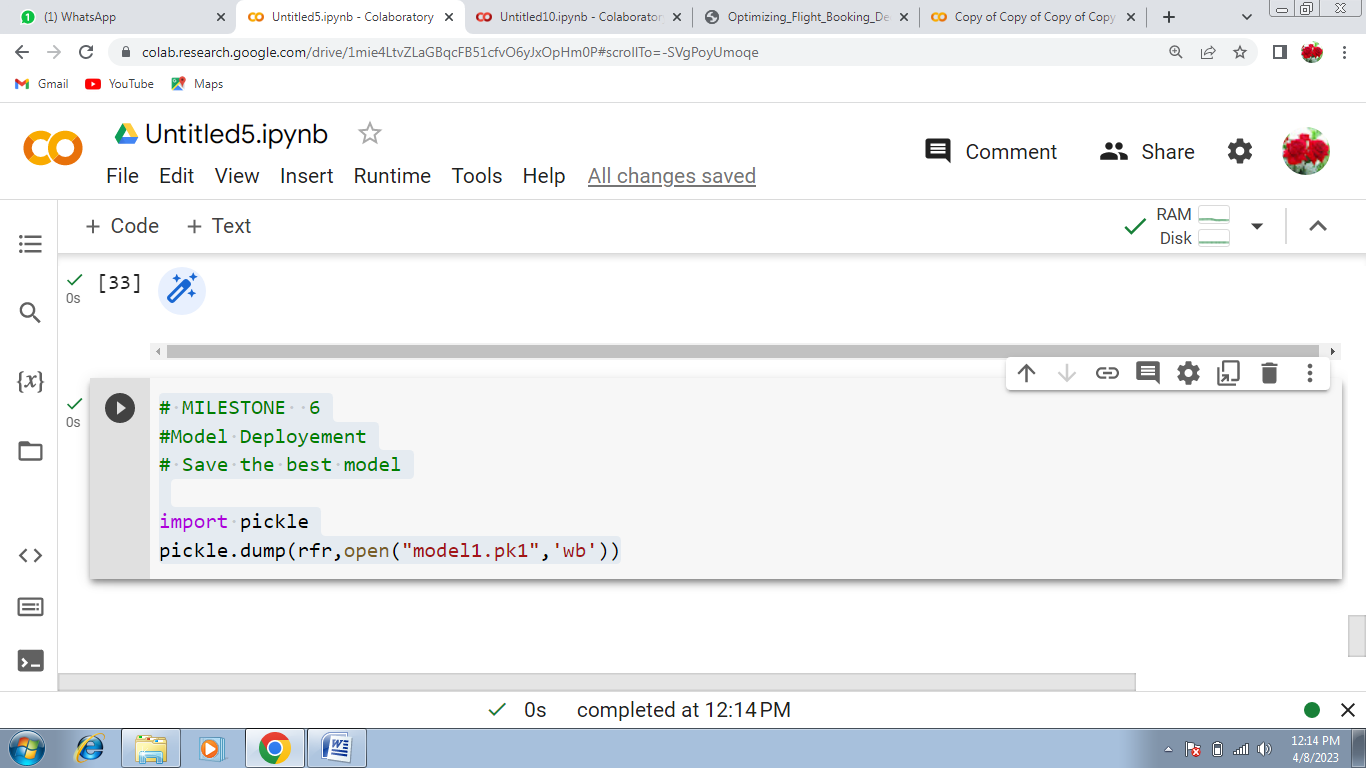






import pickle

pickle.dump(rfr,open("model1.pk1",'wb'))



**#importing libraries**

import matplotlib.pyplot as plt

import numpy as nm

import pandas as pd

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

#from sklearn.preprocessing import labelEncoder

from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor

from sklearn.tree import DecisionTreeRegressor

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVR

from sklearn.metrics import r2\_score,mean\_absolute\_error,mean\_squared\_error

from scipy import stats

import warnings

import pickle

warnings.filterwarnings('ignore')

**#Read the Dataset**

data=pd.read\_csv("/content/drive/MyDrive/Data\_train.csv")

data.head()

for i in data:

print(i,data[i].unique())

**#Checking value in Destination**

data['Destination'].value\_counts()

data.info()

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]

data.head()

data.Total\_Stops.unique()

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.head()

#data.dropna(inplace=True)

#data.isnull().sum()

#In the similar manner, we split the Dep\_time column, and create separate columns for departure hours and minutes

data['Dep\_hour'] = pd.to\_datetime(data['Dep\_Time']).dt.hour

data['Dep\_min'] = pd.to\_datetime(data['Dep\_Time']).dt.minute

data.drop('Dep\_Time',axis=1,inplace=True)

data['Arrival\_hour'] = pd.to\_datetime(data['Arrival\_Time']).dt.hour

data['Arrival\_min'] = pd.to\_datetime(data['Arrival\_Time']).dt.minute

data.drop('Arrival\_Time',axis=1,inplace=True)

data.head()

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

data['Travel\_Hours']=data.Duration.str[1]

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[0]

data['Travel\_Mins']=data.Travel\_Mins.str.split('m')

data['Travel\_Mins']=data.Travel\_Mins.str[1]

data.head()

data.Additional\_Info.unique()

data.isnull().sum()

categorical=['Airline','Source','Destination','Additional\_Info','City1']

numerical=['Total\_Stops','Date','Month','Year','Dep\_Time\_Hour','Dep\_Time\_Mins','Arrival\_Date','Arrivel\_Time\_Hour','Arrival\_Time\_Mins','Travel\_Hours','Travel\_Mins']

data.head()

**#Label Encoder**

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.Additional\_Info=le.fit(data.Additional\_Info)

data.head(10)

data=data[['Airline','Source','Destination','Date','Month','Year','Dep\_hour','Dep\_min','Arrival\_hour','Arrival\_min','Price']]

data.head()

**#Descriptive Stastical**

data.describe()

**#Visual Analysis**

c=1

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

categorical=['Airline','Source','Destination','Additional\_Info']

#for i in categorical:

# plt.subplot(6,3,c)

#sns.countplot(x=data[i])

#plt.xticks(rotation=90)

#plt.tight\_layout(pad=3.0)

#c=c+1

#plt.show()

#Displot function in Price column

plt.figure(figsize=(15,8))

sns.displot(data.Price)

**#Correlation using HeatMap**

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

**#detecting the outlies**

sns.boxplot(data['Price'])

y=data['Price']

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

**#scalling the data**

ss=StandardScaler()

x\_scaled=ss.fit\_transform(x)

x\_scaled=pd.DataFrame(x\_scaled,columns=x.columns)

x\_scaled.head()

**#Split Train and Test**

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y)

x\_train.head()

**#using ensembele techniques**

rf=RandomForestRegressor()

gb=GradientBoostingRegressor()

ad=AdaBoostRegressor()

for i in[rf,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\_squared\_error(y\_pred,y\_test,squared=False)))

**#KNN and Svm**

knn=KNeighborsClassifier()

svm=SVR()

dt=DecisionTreeRegressor()

for i in[knn,svm,dt]:

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)<=0.1:

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\_squared\_error(y\_pred,y\_test,squared=False)))

#from sklearn.model\_selection import cross\_val\_score

#for i in range(2,5):

# cv=cross\_val\_score(rf,x,y,cv=i)

#print(rf,cv.mean())

**#Accuracy**

rf=RandomForestRegressor(n\_estimators=10,max\_features='sqrt',max\_depth=None)

rf.fit(x\_train,y\_train)

y\_train\_pred=rf.predict(x\_train)

y\_test\_pred=rf.predict(x\_test)

print("Test Accuracy",r2\_score(y\_train\_pred,y\_train))

print("Train Accuracy",r2\_score(y\_test\_pred,y\_test))

knn=KNeighborsClassifier(n\_neighbors=2,algorithm='auto',metric\_params=None,n\_jobs=1)

knn.fit(x\_train,y\_train)

y\_train\_pred=rf.predict(x\_train)

y\_test\_pred=rf.predict(x\_test)

print("Test Accuracy",r2\_score(y\_train\_pred,y\_train))

print("Train Accuracy",r2\_score(y\_test\_pred,y\_test))

#Evaluating the perfoermance

#price\_list=pd.DataFrame({'Price':data})

#price\_list

**#Save the model**

pickle.dump(rf,open('model1.pkl','wb'))

data.head()