Practical-1

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM:- Write a python program to handle the “Numpy” and “Matplotlib”

library.REQUIRED SOFTWARE :- Google Colab , Pycharm

THEORY :-

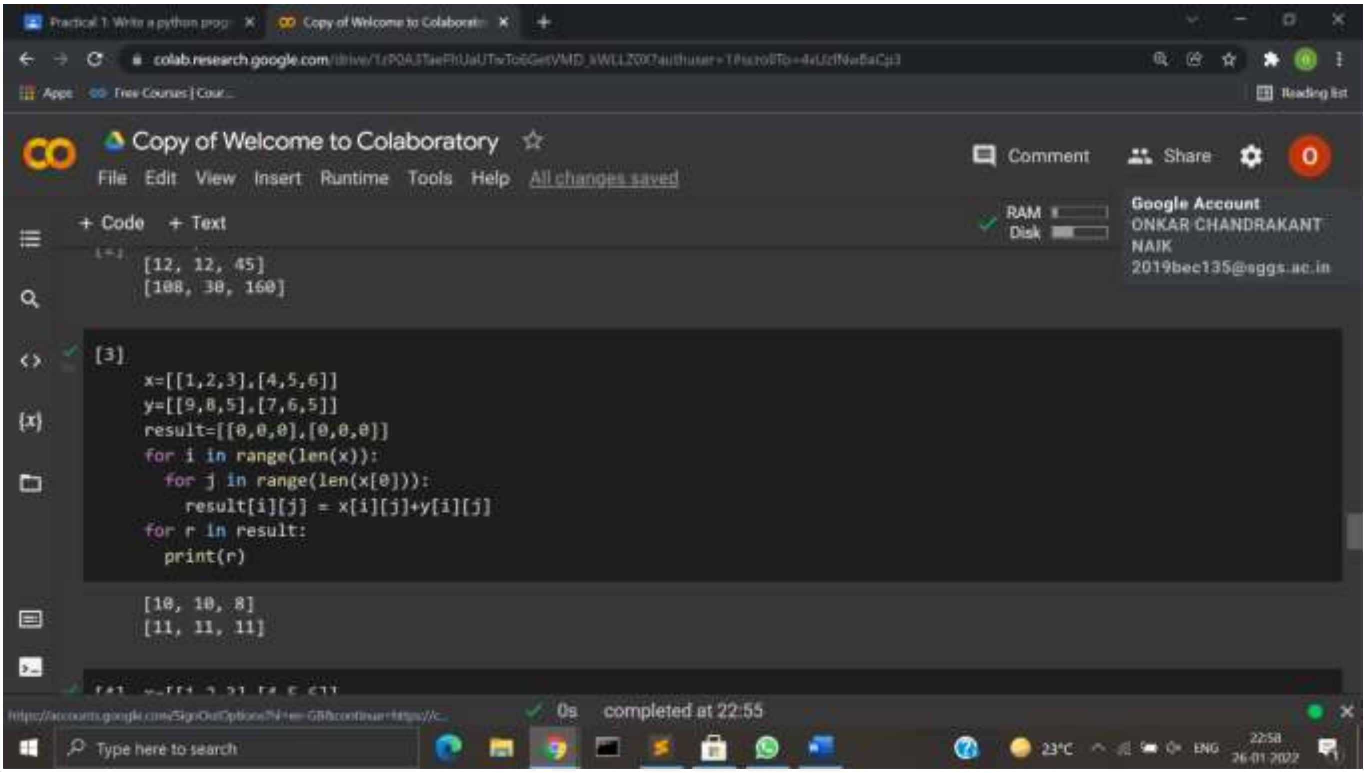
NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

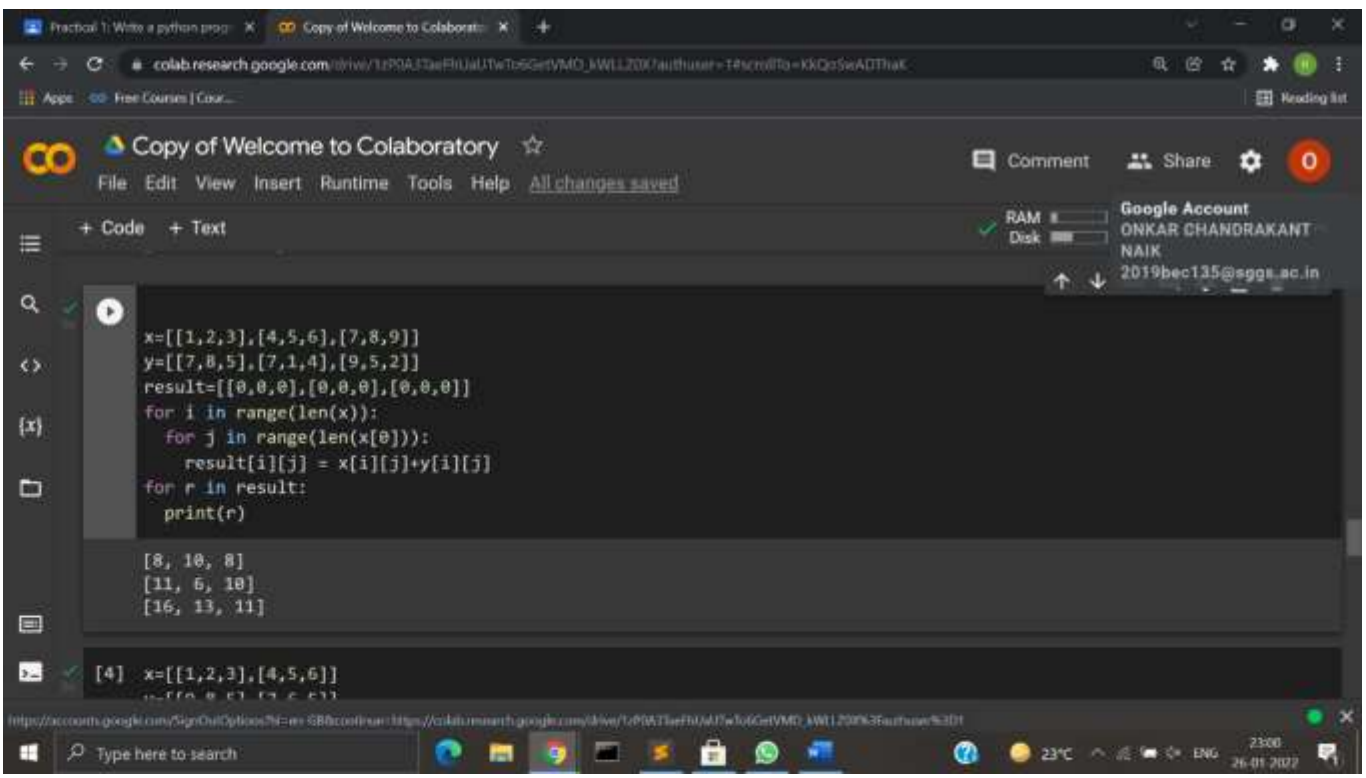
NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

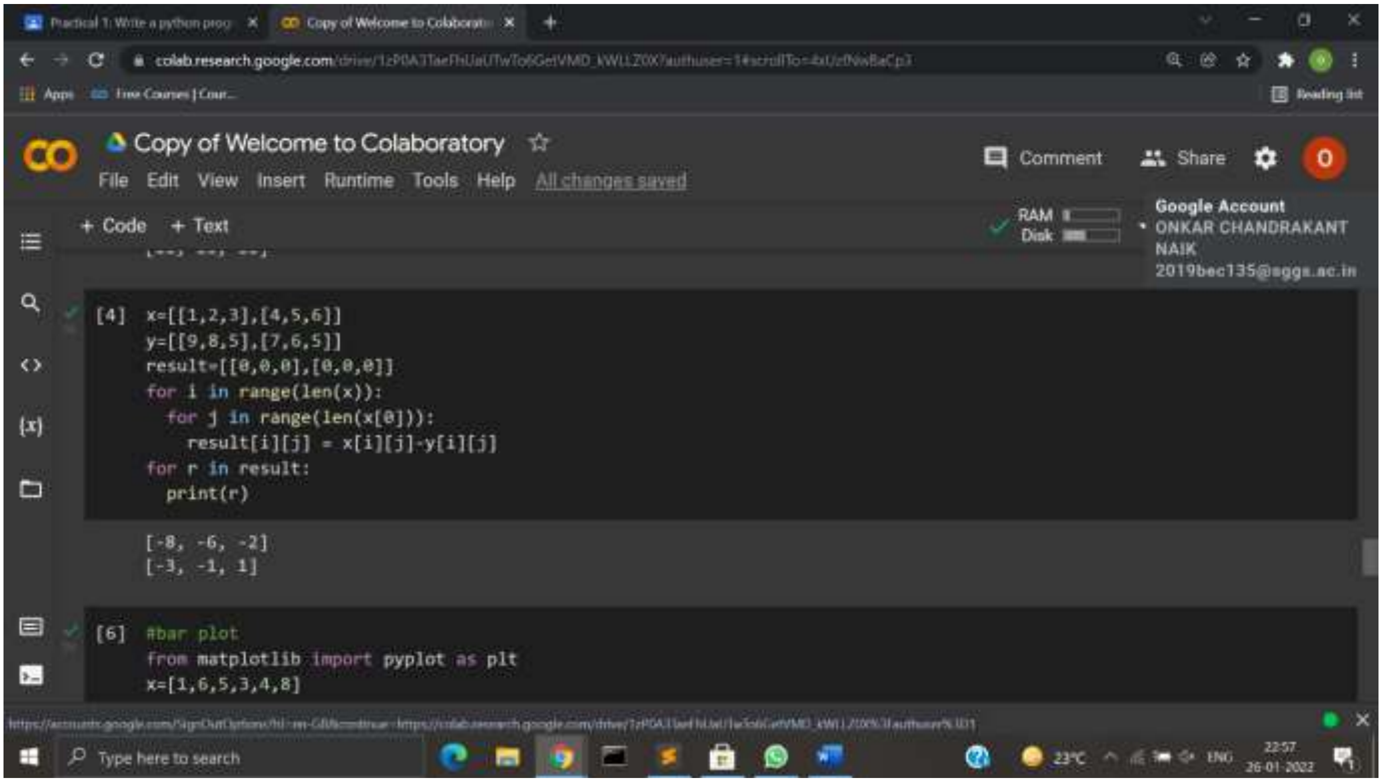
1.Define the multi-dimensional arrays (2x2 and 3x3) using Numpy library and perform various operations on it (addition, subtraction, element-wise multiplication, matrix multiplication) without inbuilt function. Print the respective outputs.

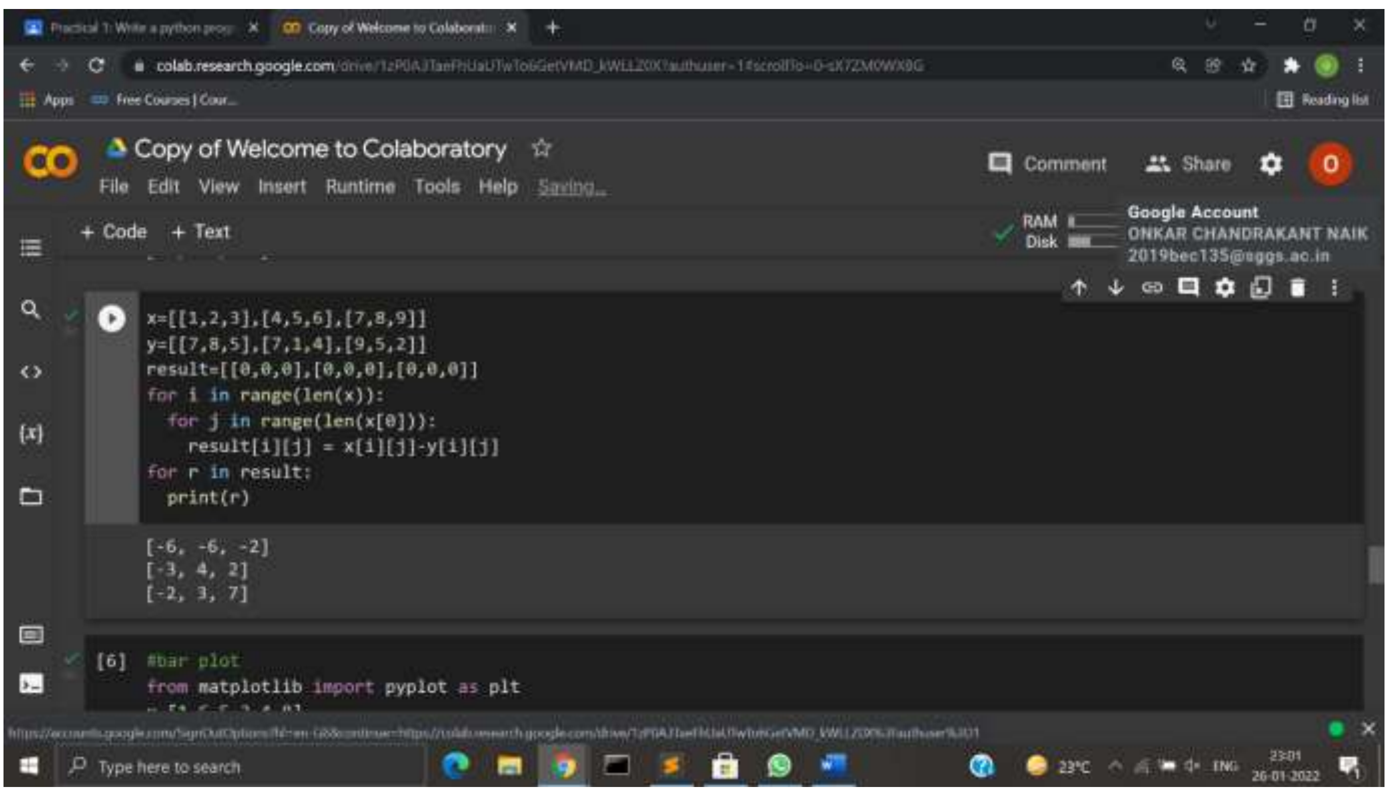
Addition:


Copy of Welcome to Colaboratory
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Google Account
ONKAR CHANDRAKANT NAIK
2019bec135@sggs.ac.in
[12, 12, 45]
[108, 30, 160]
[3]
x=[[1,2,3],[4,5,6]]
y=[[9,8,5],[7,6,5]]
result=[[0,0,0],[0,0,0]]
for i in range(len(x)):
for j in range(len(x[0])):
result[i][j] = x[i][j]+y[i][j]
for r in result:
print(r)
[10, 10, 8]
[11, 11, 11]
0s completed at 22:55
Type here to search
23°C
ENG
22:58
26-01-2022

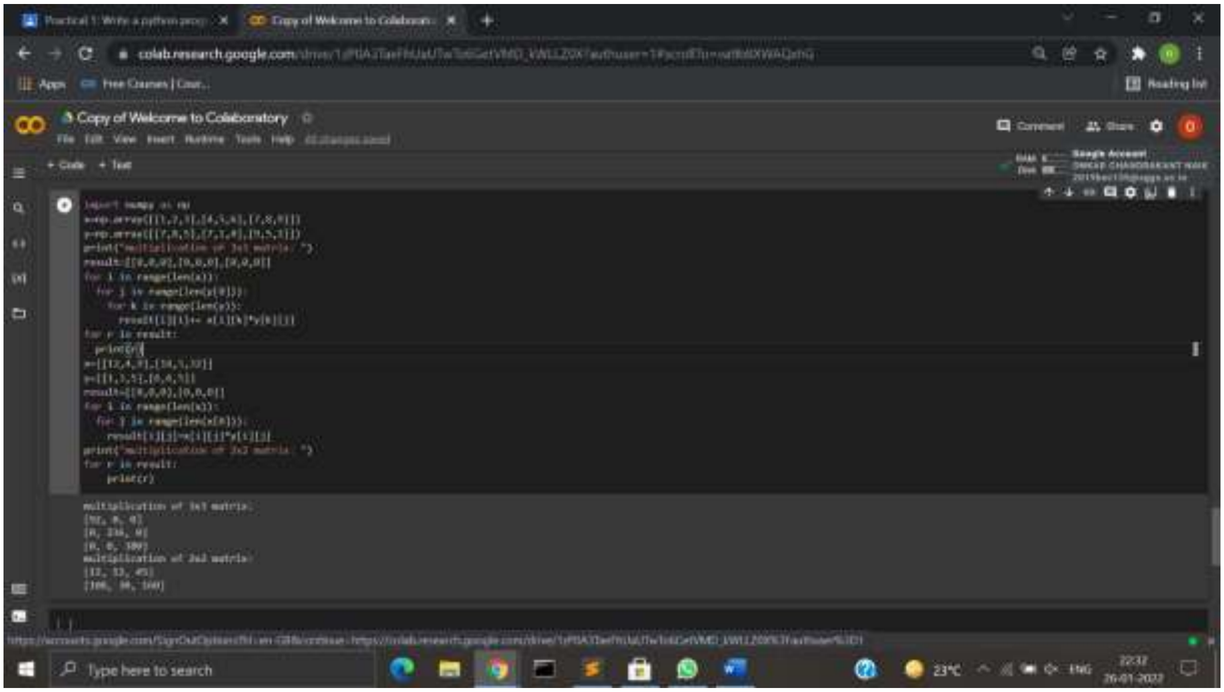

Copy of Welcome to Colaboratory
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Google Account
ONKAR CHANDRAKANT NAIK
2019bec135@sggs.ac.in
x=[[1,2,3],[4,5,6],[7,8,9]]
y=[[7,8,5],[7,1,4],[9,5,2]]
result=[[0,0,0],[0,0,0],[0,0,0]]
for i in range(len(x)):
for j in range(len(x[0])):
result[i][j] = x[i][j]+y[i][j]
for r in result:
print(r)
[8, 10, 8]
[11, 6, 10]
[16, 13, 11]
[4] x=[[1,2,3],[4,5,6]]
Type here to search
23°C
ENG
23:00
26-01-2022

Subtraction:

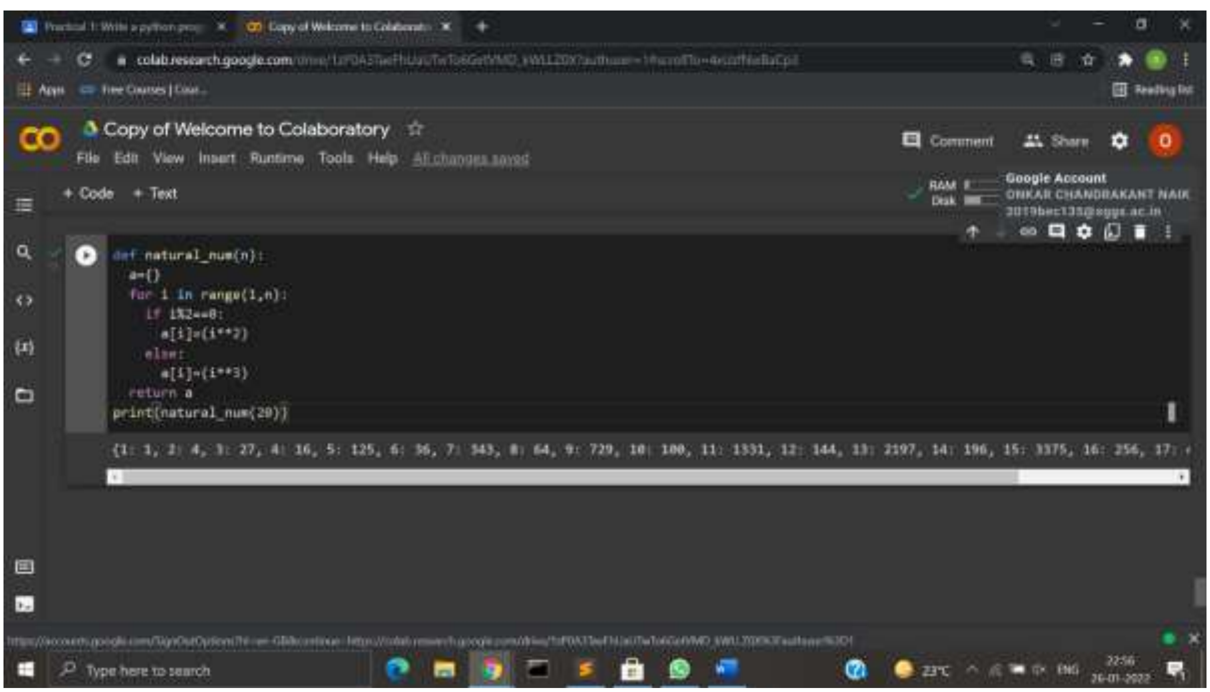




Multiplication:

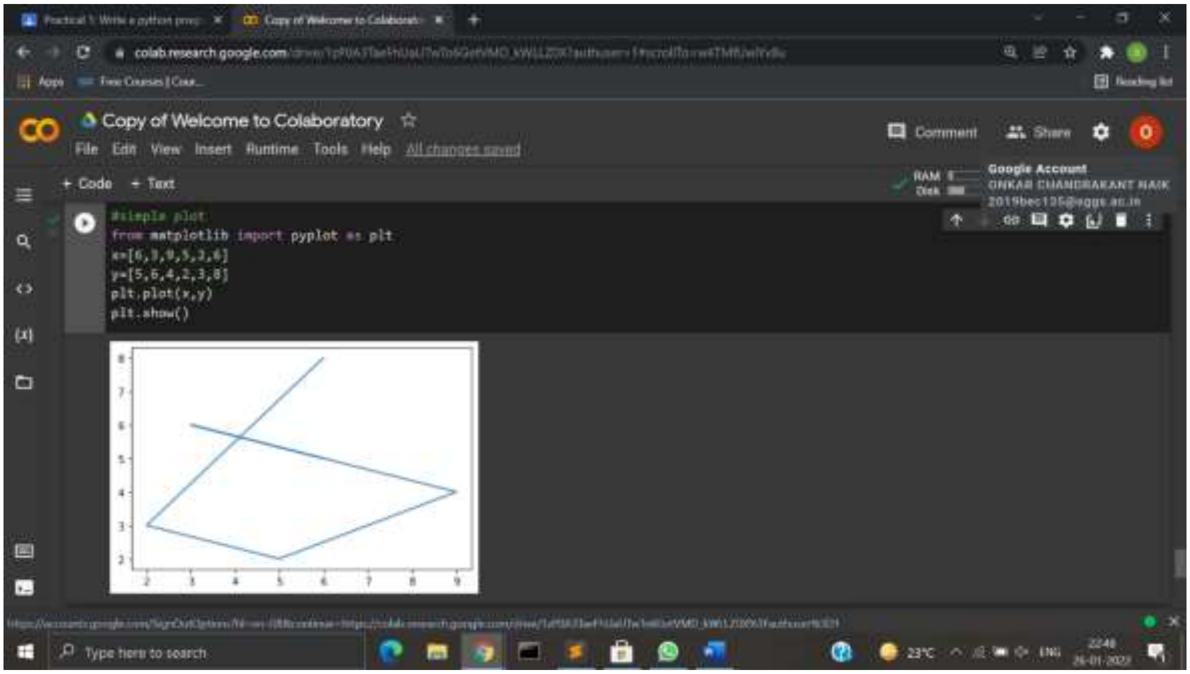


2.Write a python code to generate dictionary which consist of first 20 natural numbers as a keys. The even number keys must have square of number as value while odd number keys must have cube of number as value. Print the dictionary.

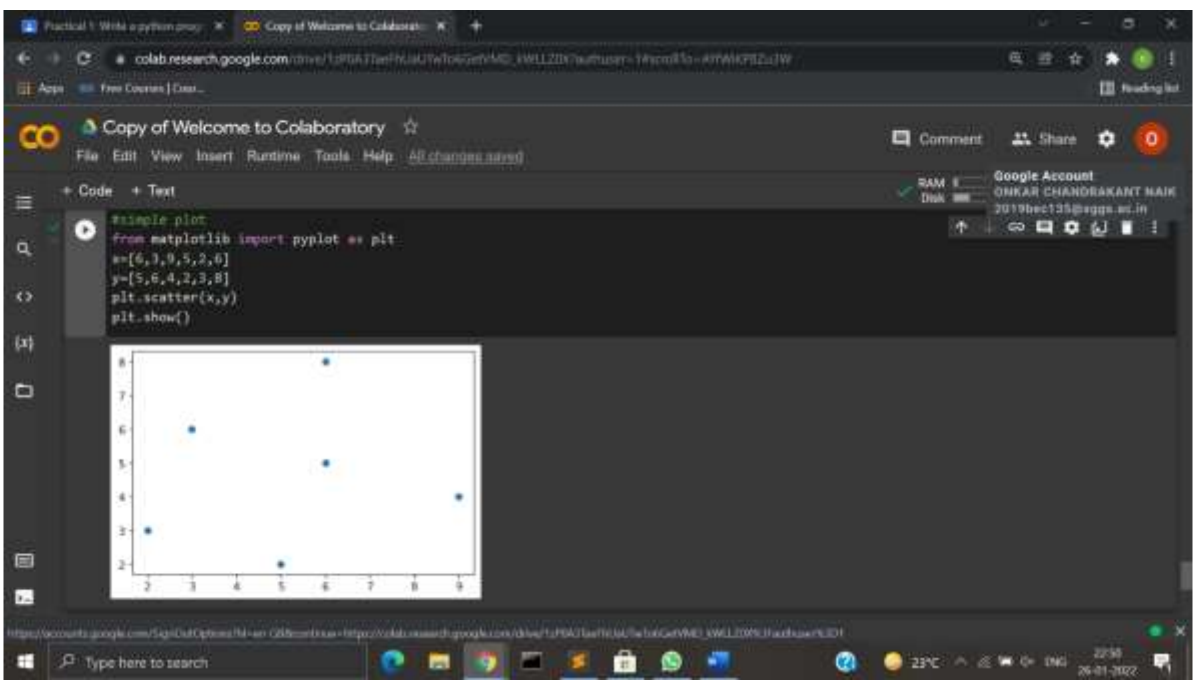


3.Define two variable having functional dependency. Plot the three graphs (simple plot, scatter plot and bar graph) between two variables using Matplotlib library.

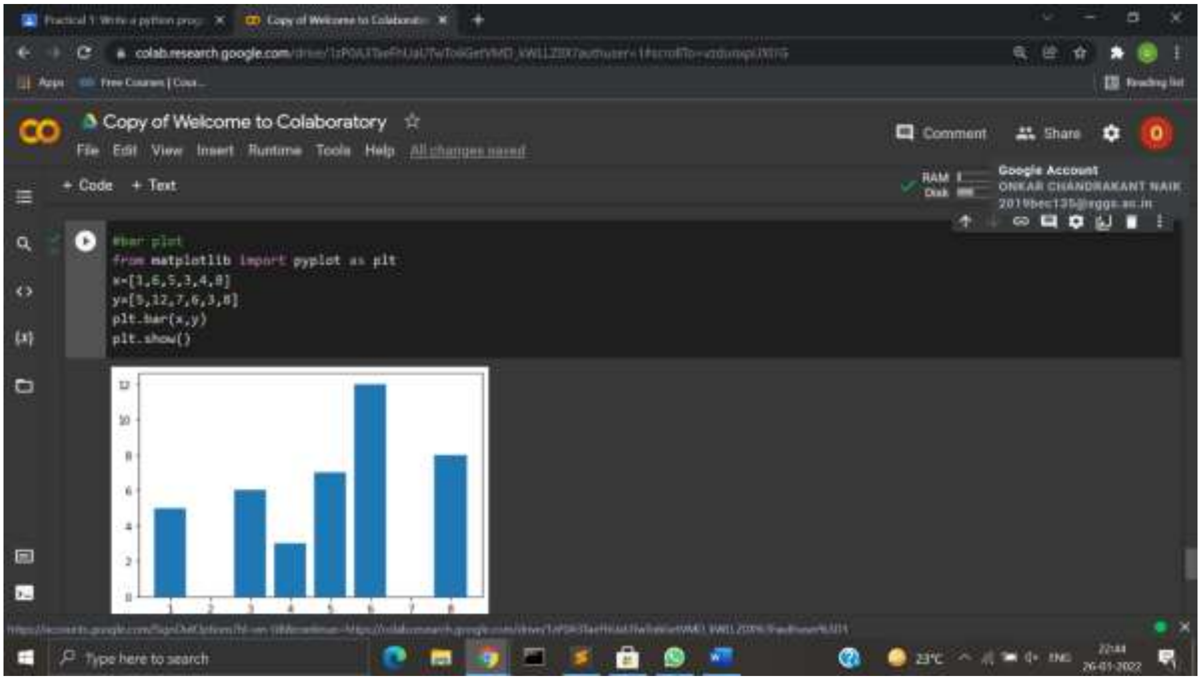
1)Simple Plot:



2)Scatter Plot:



3)Bar plot:



CONCLUSION : - Numpy is a basic Python package that provides an alternative to a regular Python list, a Numpy n-dimensional homogeneous array. A list is a very useful tool offered by Python, as it lets you store values of different types at once, and perform numerous operationson it while Matplotlib is the most powerful visualization library. To use it, you need to import itssub package pyplot.

Practical-2

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM:- Write a python program to plot 2D & 3D Gaussian distribution curve usingprobability of likelihood formula.

REQUIRED SOFTWARE :- Google Colab ,

PycharmTHEORY :-

The Gaussian distribution is the healthy-studied probability distribution. It is for nonstop-valued random variables. It is as well stated as the normal distribution. Its position makes from the factthat it has many computationally suitable properties.

There are several parts of machine learning that takes advantage from using a

Gaussiandistribution. Those areas are including;

- Gaussian processes
- Variational inference
- Reinforcement learning

It is similarly broadly used in other application areas such as

- Signal processing such as Kalman filter
- Control, for example linear quadratic regulator
- Statistics, as hypothesis testing

Importance of Gaussian Distribution

It is ever-present as a dataset with finite variance turns into Gaussian as long asdataset with free feature-probabilities is permitted to raise in size.

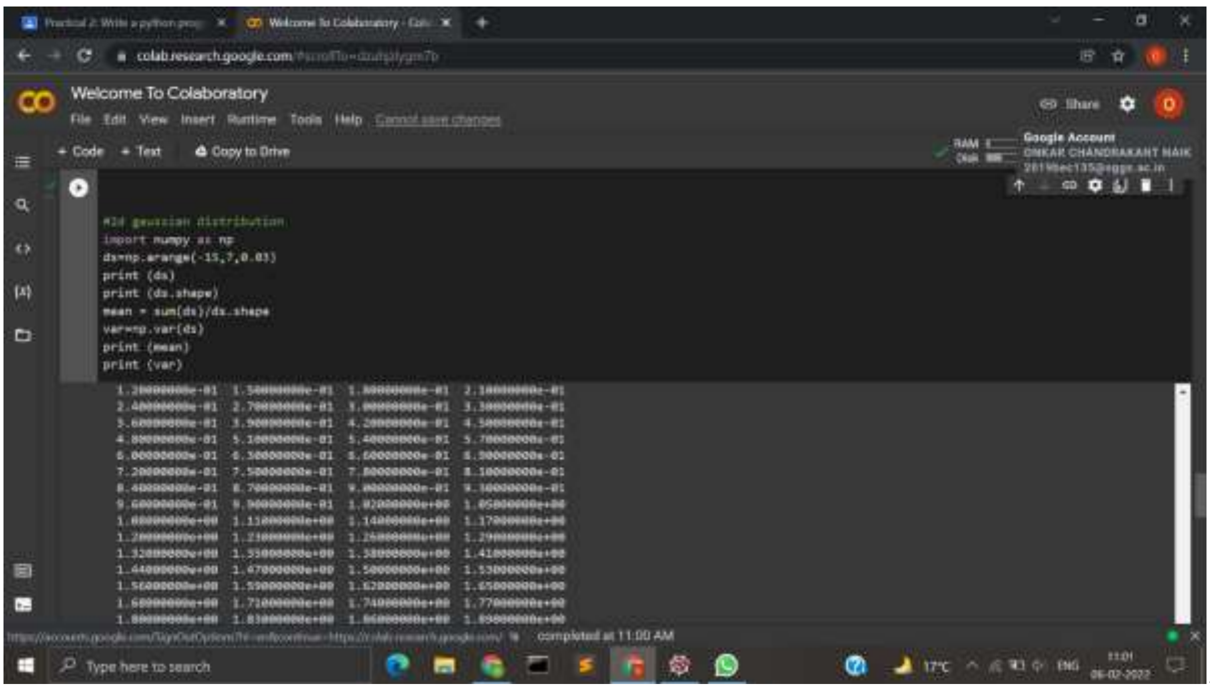
It is the most significant probability distribution in statistics as it turns many natural phenomena such as age, height, test-scores, IQ scores, and sum of the rolls of twocubes and so on.

Datasets with Gaussian distributions creates valid to a diversity of methods that decrease under parametric statistics.

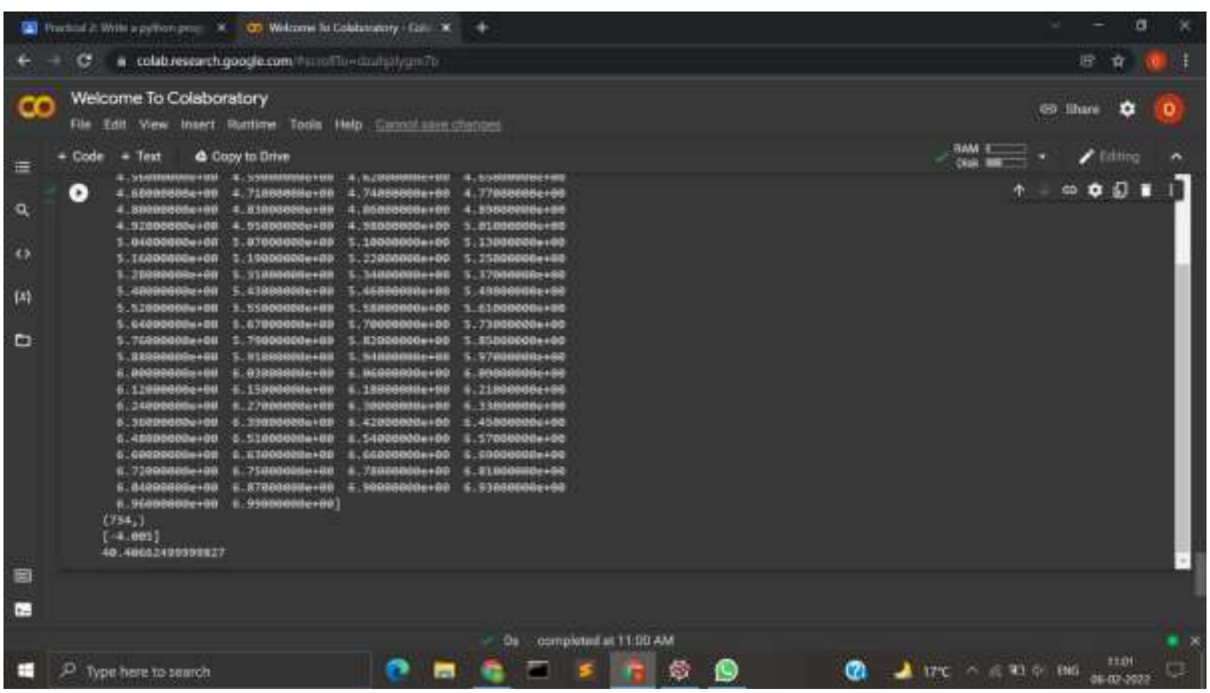
The approaches for example propagation of uncertainty and least squares parameterright are related only to datasets with normal or normal-like distributions.

Reviews and conclusions resulting from such analysis are intuitive. That also easy toexplain to audiences with basic knowledge of statistics.

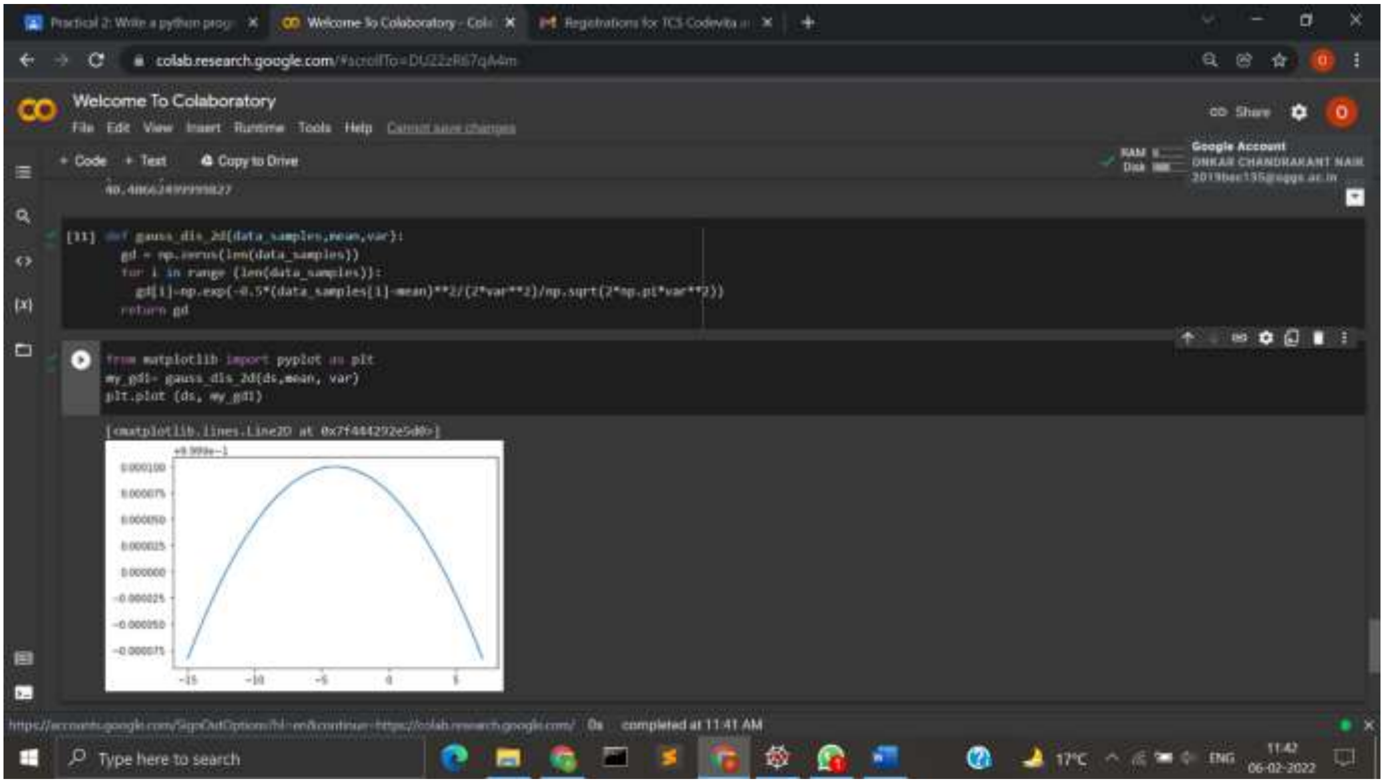
1.Define the data sample x in range (any) for 1D data. Decide mean & variance and print it



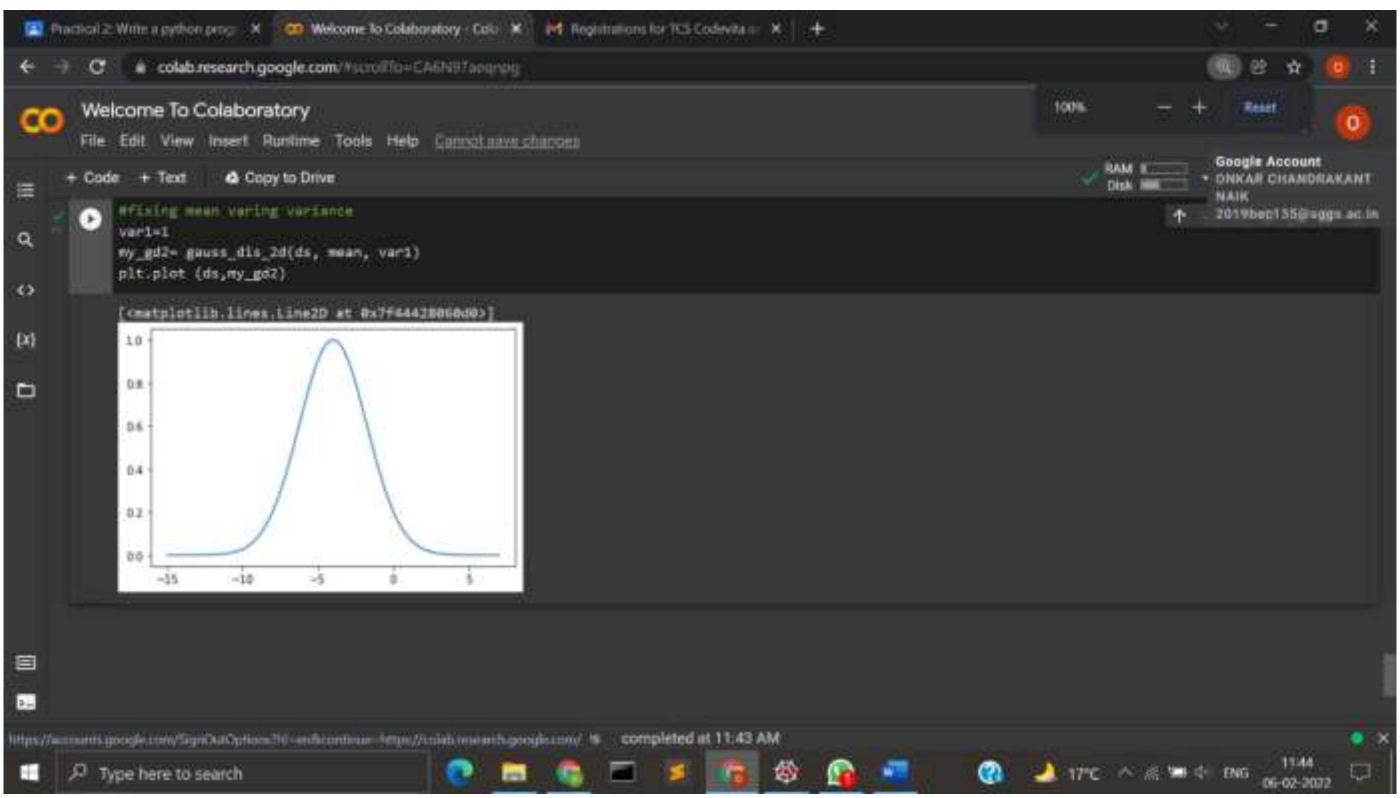
.



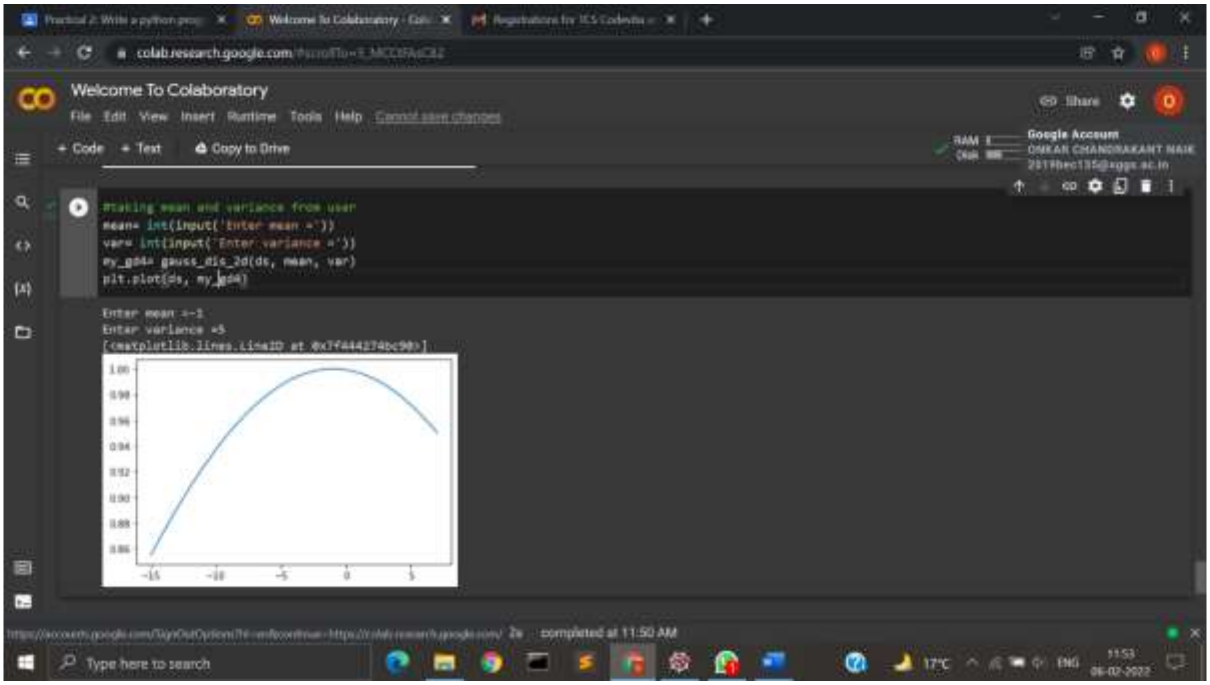
2. Use formula, and plot the 1D Gaussian curve.



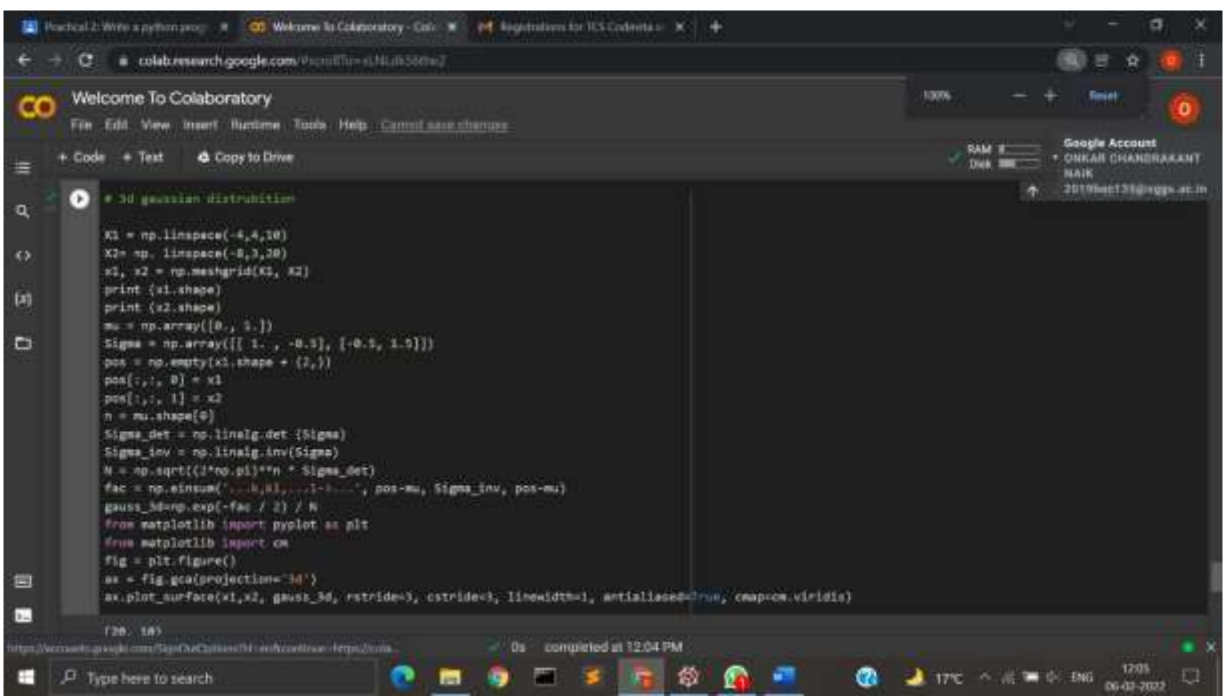
3. Fix the mean and vary the variance and plot the curve. Fix the variance and vary the mean and plot the curve.



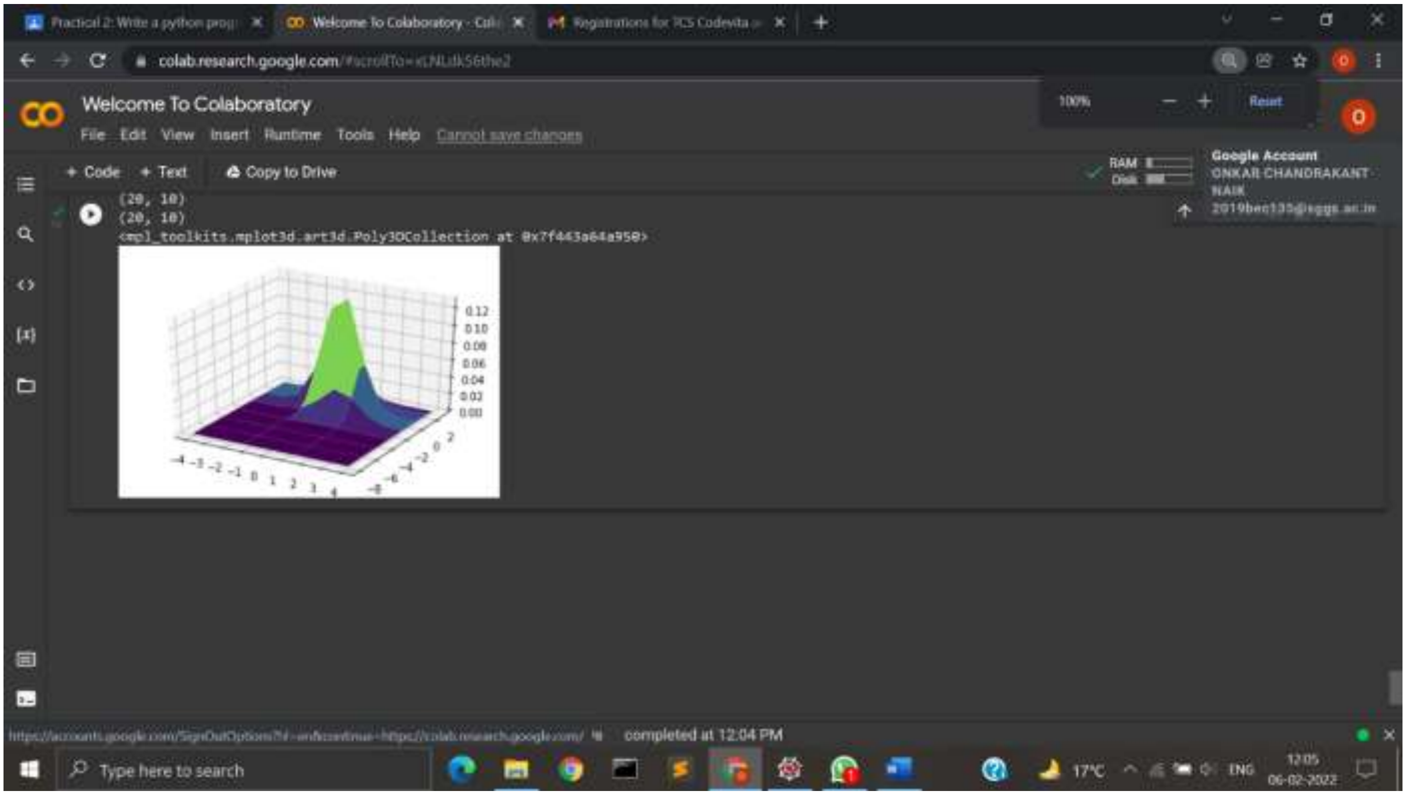
4. Ask the user for mean and variance and plot the 2D Gaussian distribution for it using function definition.



5. Define the data samples x1 and x2 for 2D data. Define X1,X2=meshgrid[x1:x2] and print its shape.



6. According to formula, compute pdf and plot 3D graph of Gaussian curve. Give proper labels.



Conclusion: In the given practical we plot 2d and 3d Gaussian distribution curve using probability of likelihood formula. From the graphs we can conclude that the peak of curve depends on mean value and the width of the curve depend on variance.

Practical-3

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM : Write a python program for linear regression modelling.

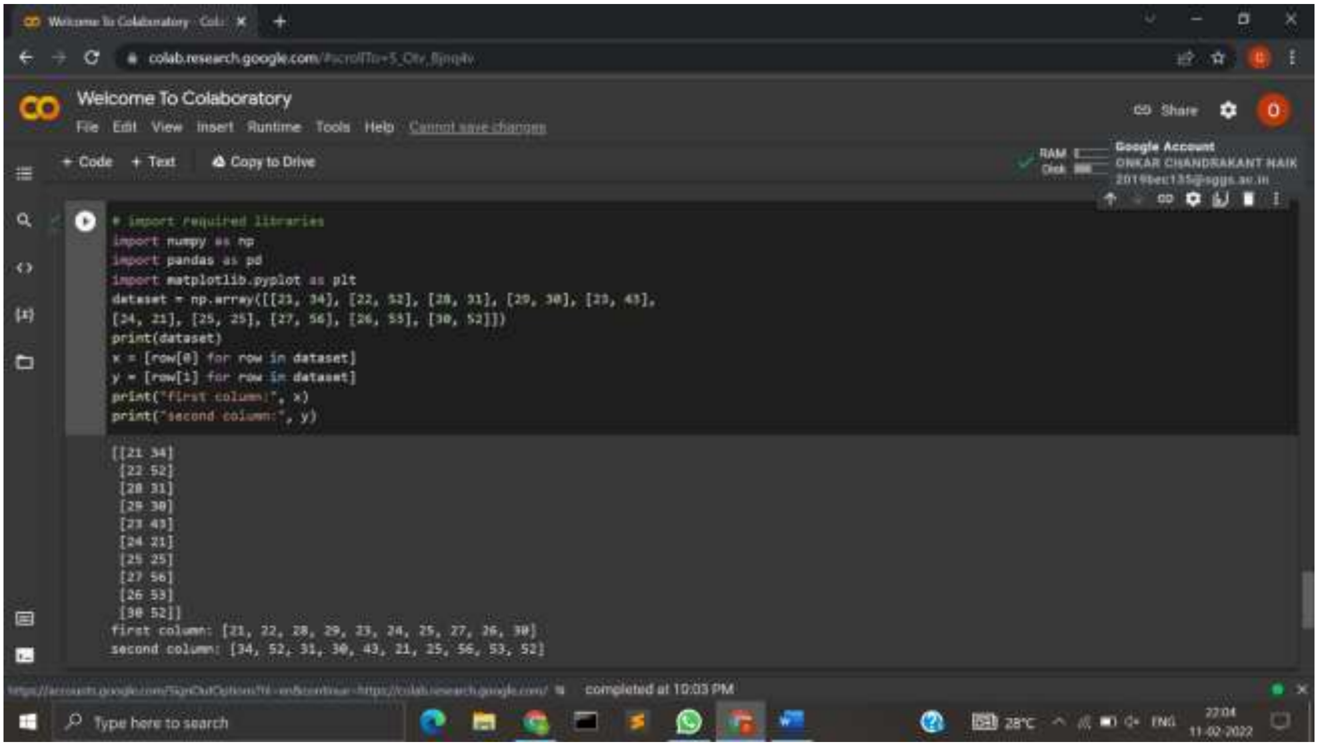
SOFTWARE :- Google Colab , Pycharm

THEORY :-

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

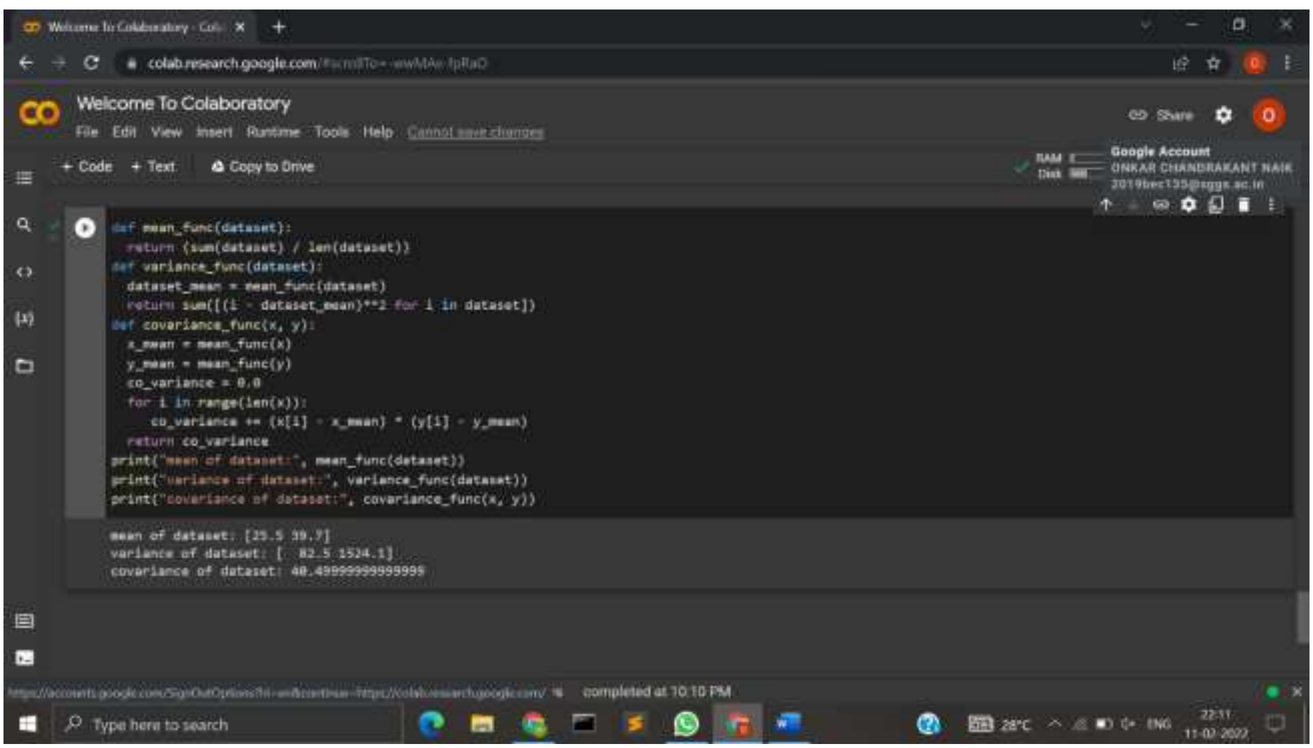
1. Create own dataset containing (10 rows and 2 column). Print the dataset.



```
# import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = np.array([[21, 34], [22, 52], [28, 31], [29, 30], [23, 43],
[24, 21], [25, 25], [27, 56], [26, 53], [30, 52]])
print(dataset)
x = [row[0] for row in dataset]
y = [row[1] for row in dataset]
print("first column:", x)
print("second column:", y)
```

```
[[21 34]
 [22 52]
 [28 31]
 [29 30]
 [23 43]
 [24 21]
 [25 25]
 [27 56]
 [26 53]
 [30 52]]
first column: [21, 22, 28, 29, 23, 24, 25, 27, 26, 30]
second column: [34, 52, 31, 30, 43, 21, 25, 56, 53, 52]
```

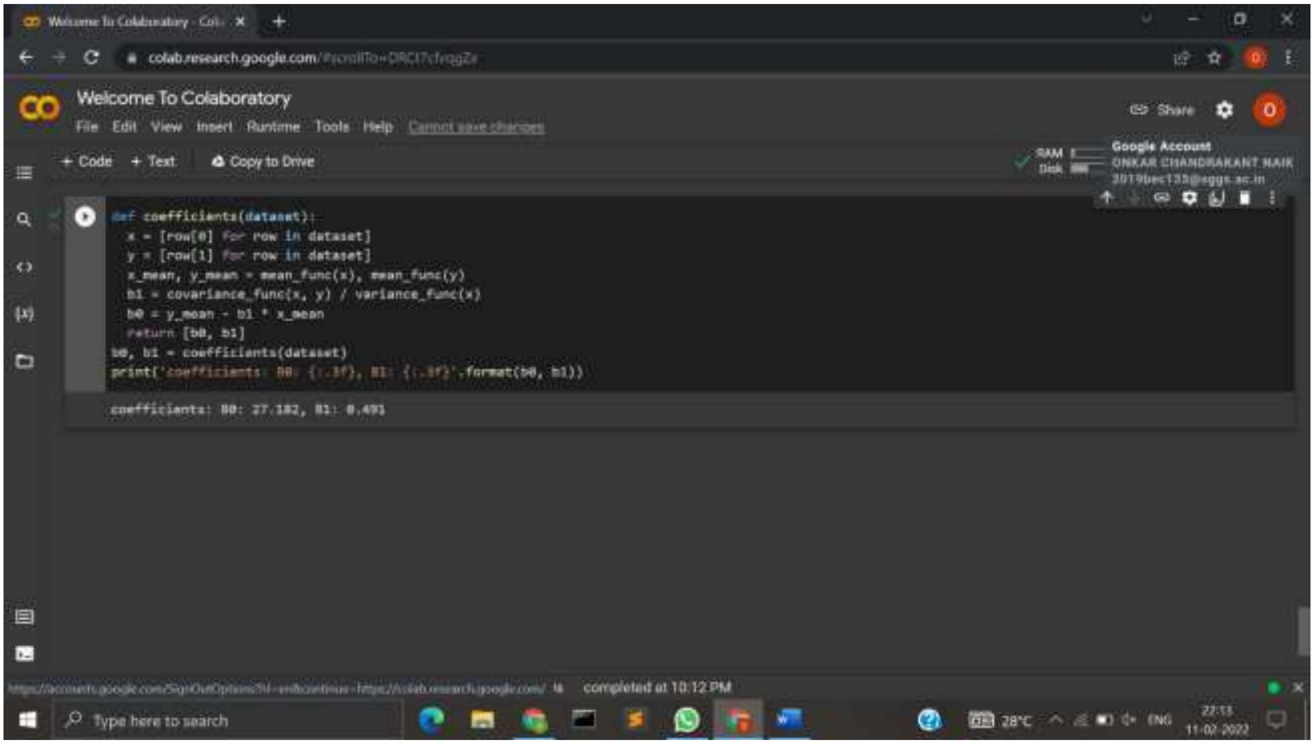
2. Define the functions for computing mean, variance and co-variance and also print the respective values for given dataset.



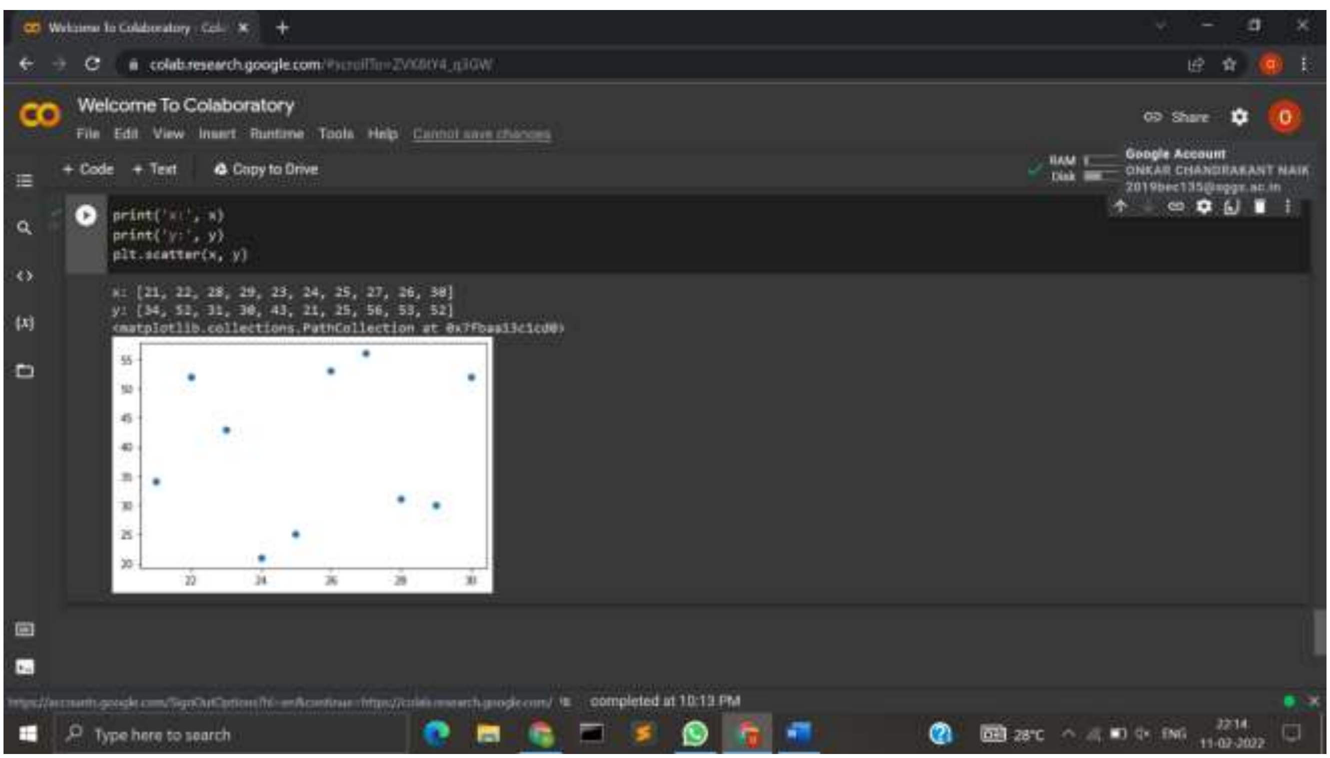
```
def mean_func(dataset):
  return (sum(dataset) / len(dataset))
def variance_func(dataset):
  dataset_mean = mean_func(dataset)
  return sum([(i - dataset_mean)**2 for i in dataset])
def covariance_func(x, y):
  x_mean = mean_func(x)
  y_mean = mean_func(y)
  co_variance = 0.0
  for i in range(len(x)):
     co_variance += (x[i] - x_mean) * (y[i] - y_mean)
  return co_variance
print("mean of dataset:", mean_func(dataset))
print("variance of dataset:", variance_func(dataset))
print("covariance of dataset:", covariance_func(x, y))
```

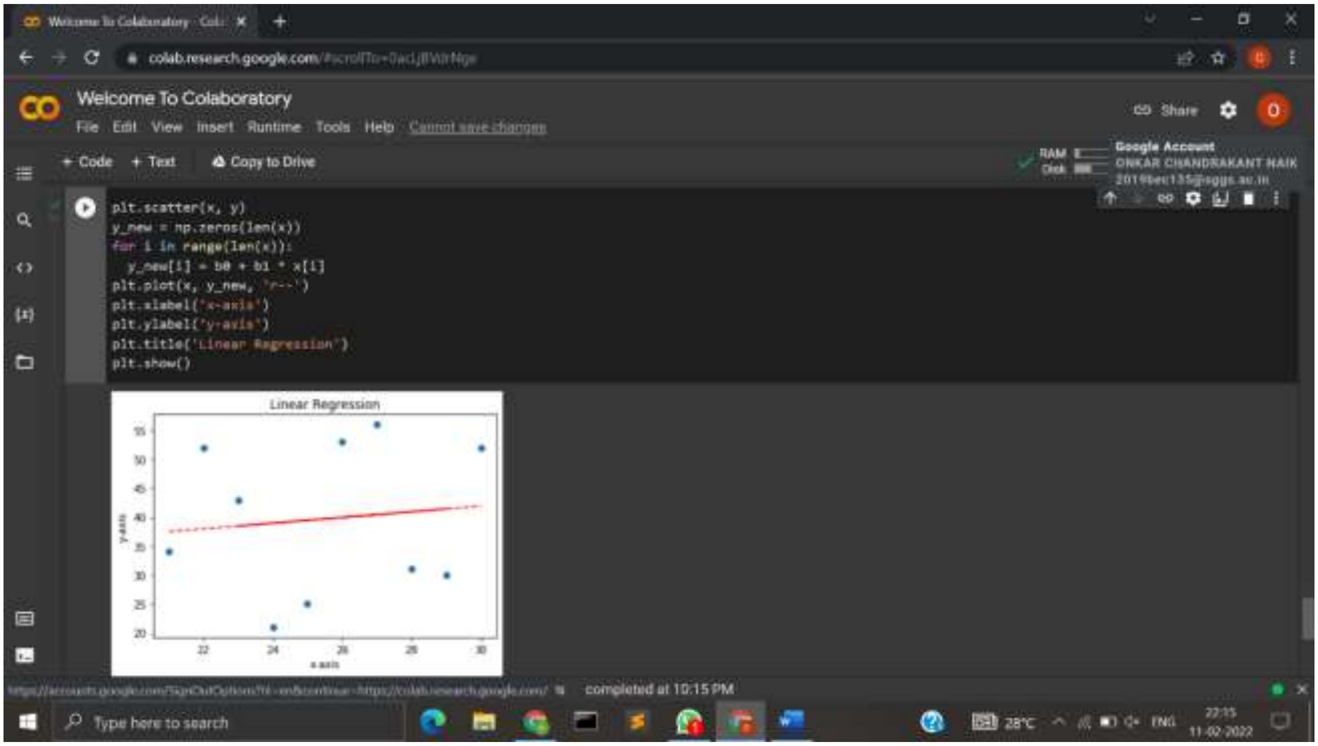
```
mean of dataset: [25.5 39.7]
variance of dataset: [  82.5 1524.1]
covariance of dataset: 40.49999999999995
```

3. Compute the linear regression coefficients using function and print it.

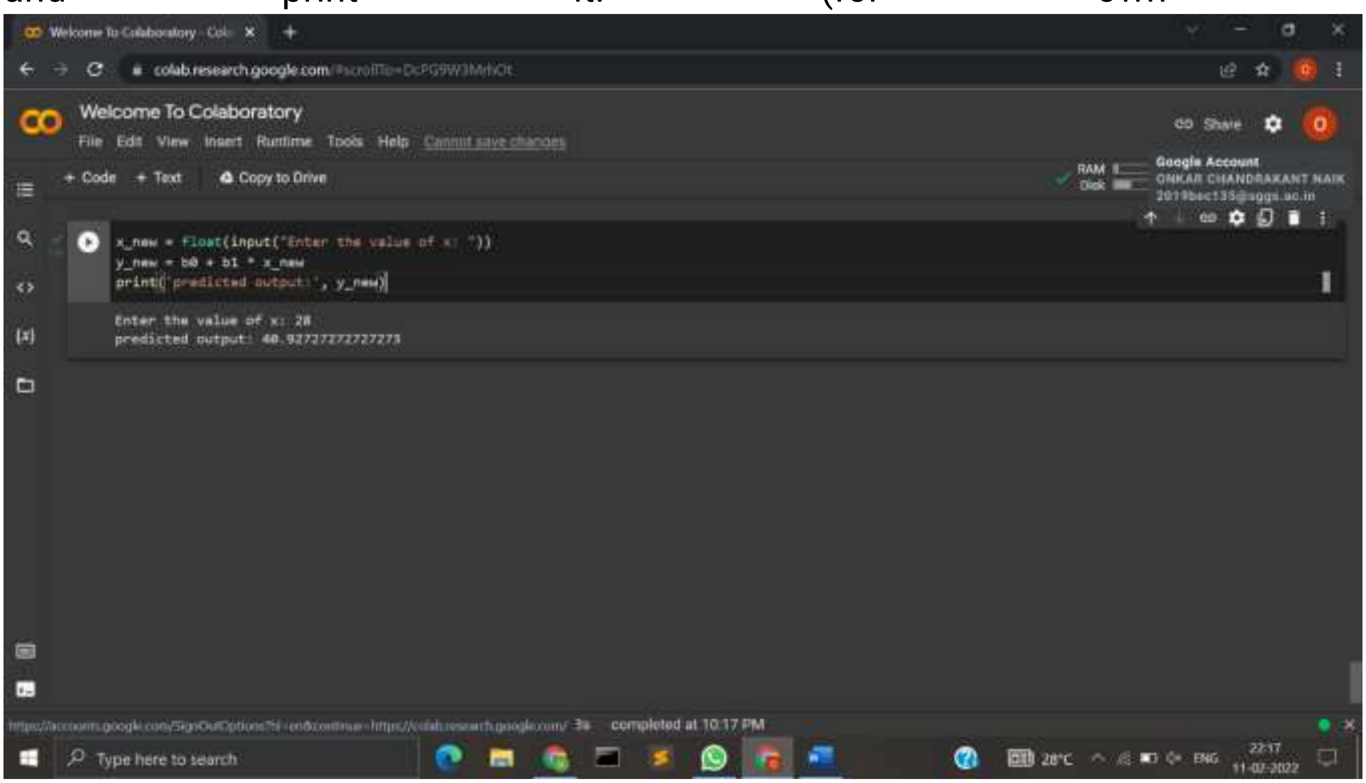


4. Plot the variables containing the values from dataset. Also plot the linear regression line using coefficients.

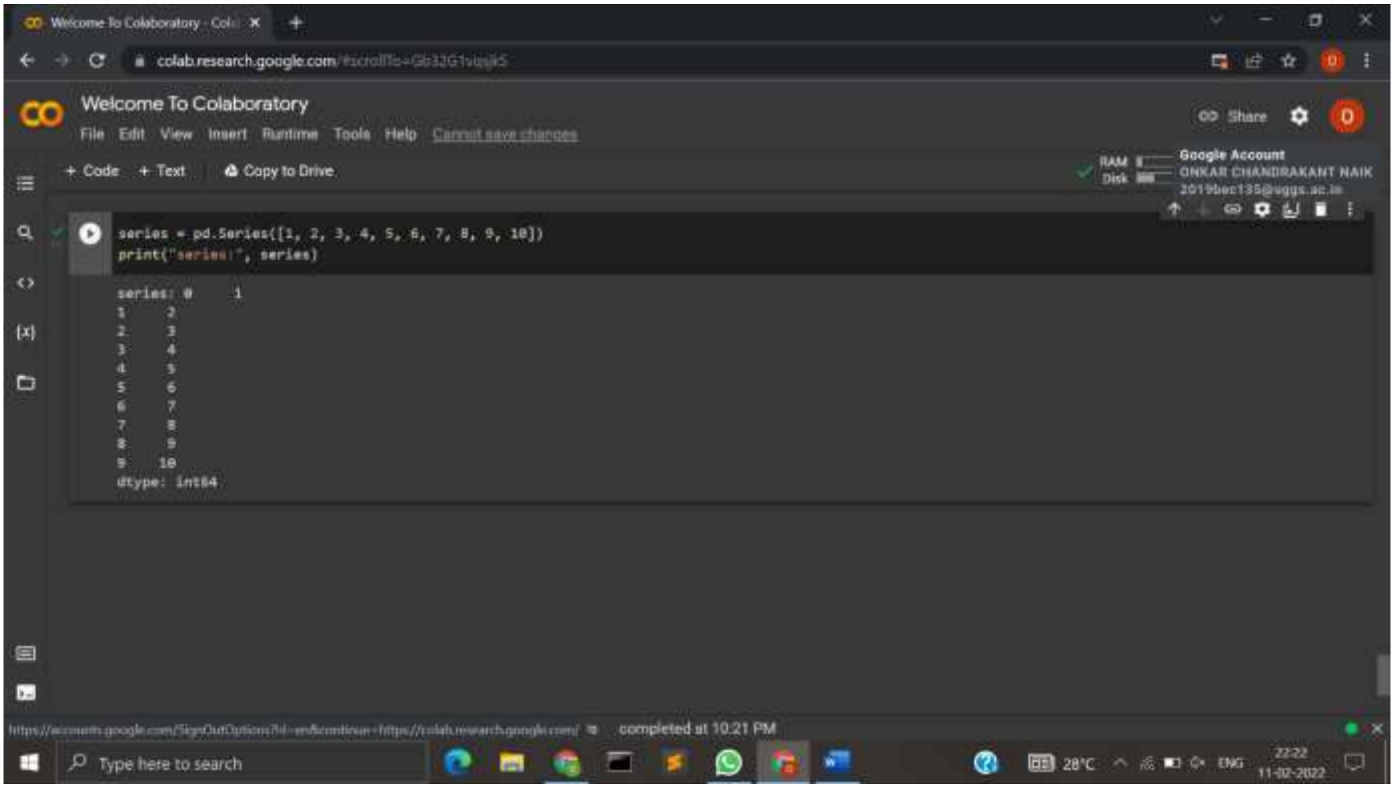




5. Find the predicted output for unknown input value using linear regression coefficients and print it. (for own dataset)



6. Create your own pandas Series and Da taFrame and print it.



CONCLUSION :- It is algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression showsthe linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Practical-4

Onkar Chandrakant Naik

2019BEC135

A65

AIM: Write a Python program for the implementation of the Bayes Theorem.REQUIRED OUTPUT :- Google Colab , Pycharm

THEORY :-

Bayes theorem is one of the most popular machine learning concepts that helps to calculate theprobability of occurring one event with uncertain knowledge while other one has already occurred.

Bayes' theorem can be derived using product rule and conditional probability of event X withknown event Y:

- According to the product rule we can express as the probability of event X with knownevent Y as follows;

1. P(X ? Y)= P(X|Y) P(Y)

- Further, the probability of event Y with known event X:

1. P(X ? Y)= P(Y|X) P(X)

Mathematically, Bayes theorem can be expressed by combining both equations on right handside. We will get:

Here, both events X and Y are independent events which means probability of outcome of bothevents does not depends one another.

The above equation is called as Bayes Rule or Bayes Theorem.

- P(X|Y) is called as **posterior**, which we need to calculate. It is defined as updated probability after considering the evidence.
- P(Y|X) is called the likelihood. It is the probability of evidence when hypothesis is true.

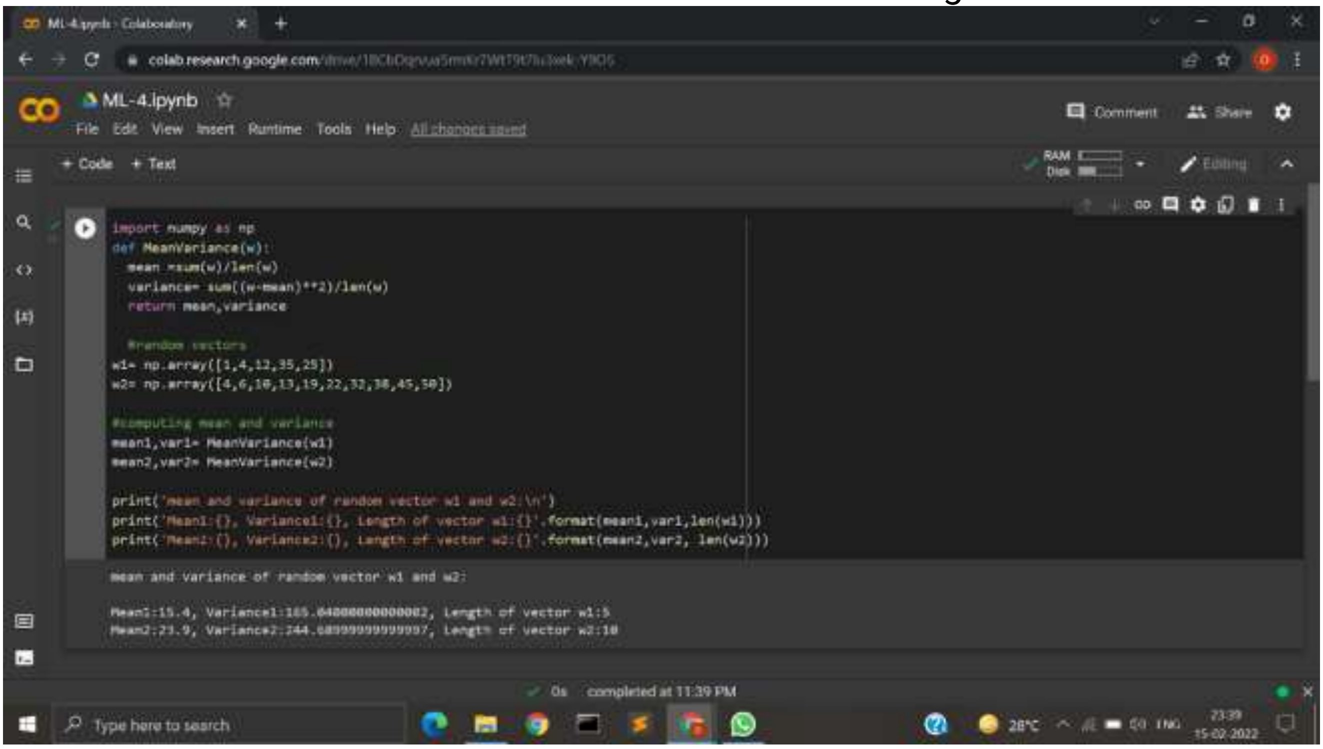
- P(X) is called the **prior probability**, probability of hypothesis before considering theevidence

  P(Y) is called marginal probability. It is defined as the probability of evidence under anyconsideration.

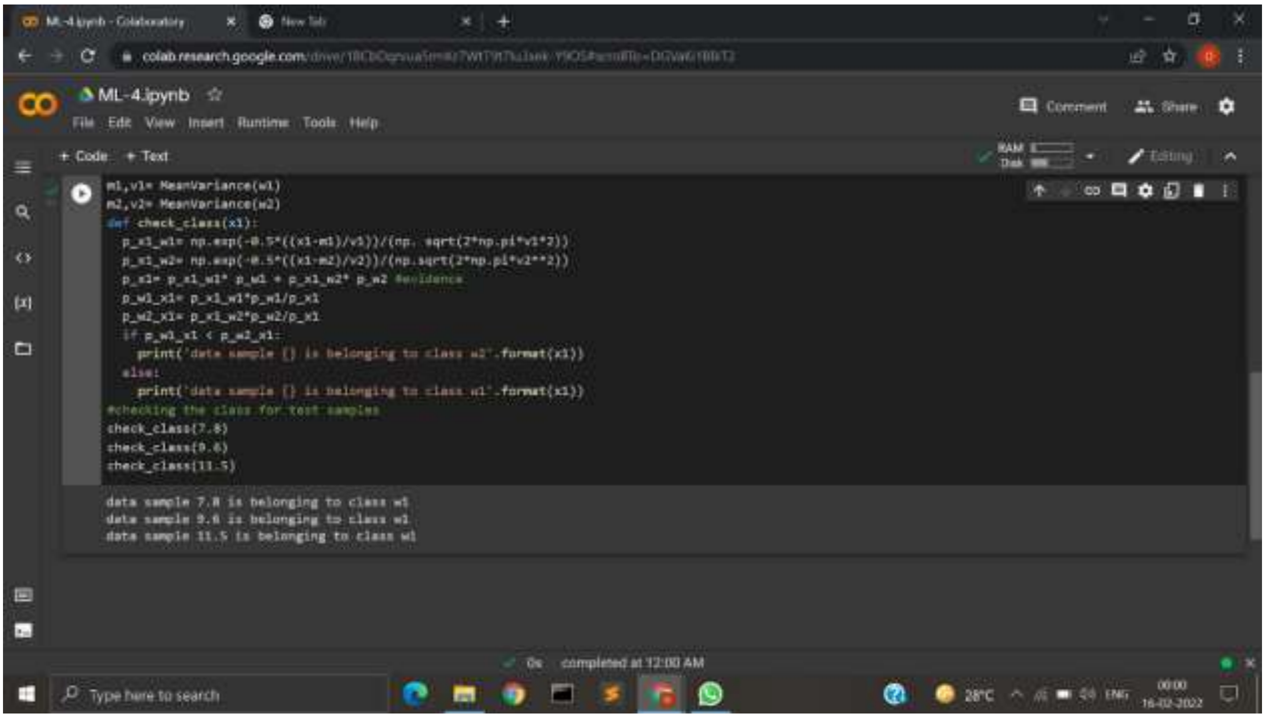
Hence, Bayes Theorem can be written as:

posterior = likelihood * prior / evidence

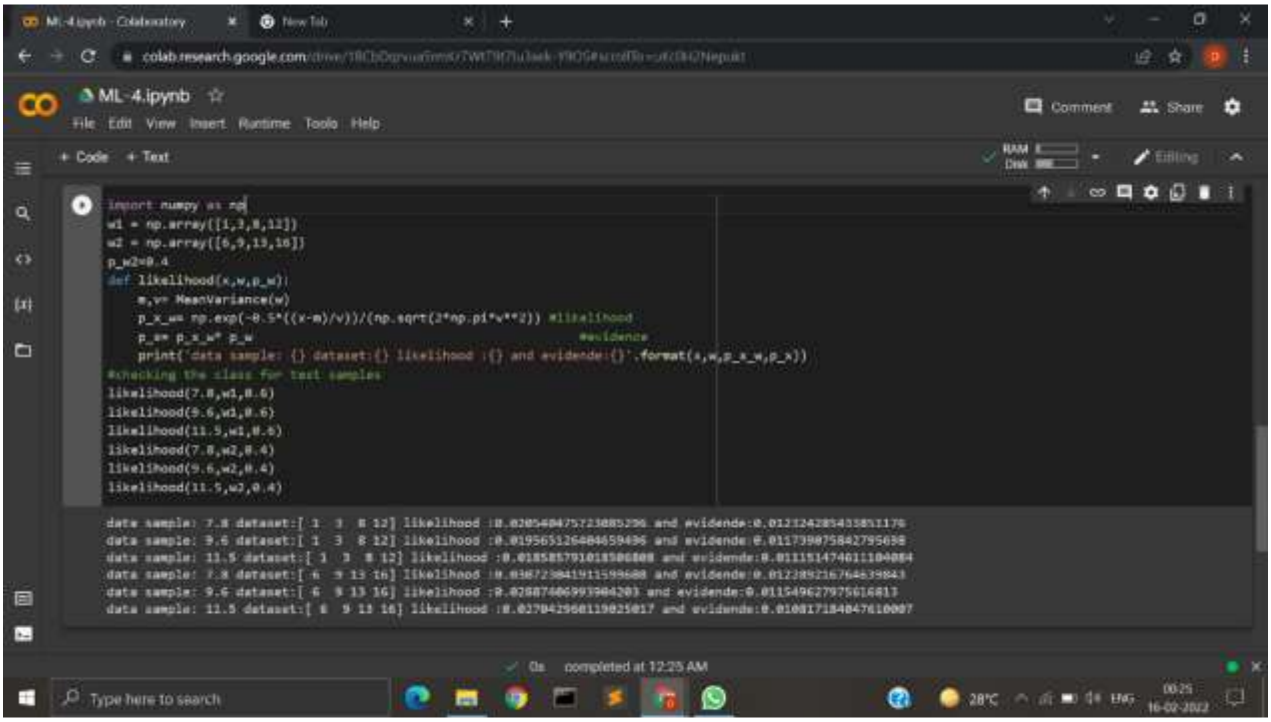
1. Define a function for computing the mean and variance of 1D data. Print mean and variance for two random vectors of length 10 and 5.



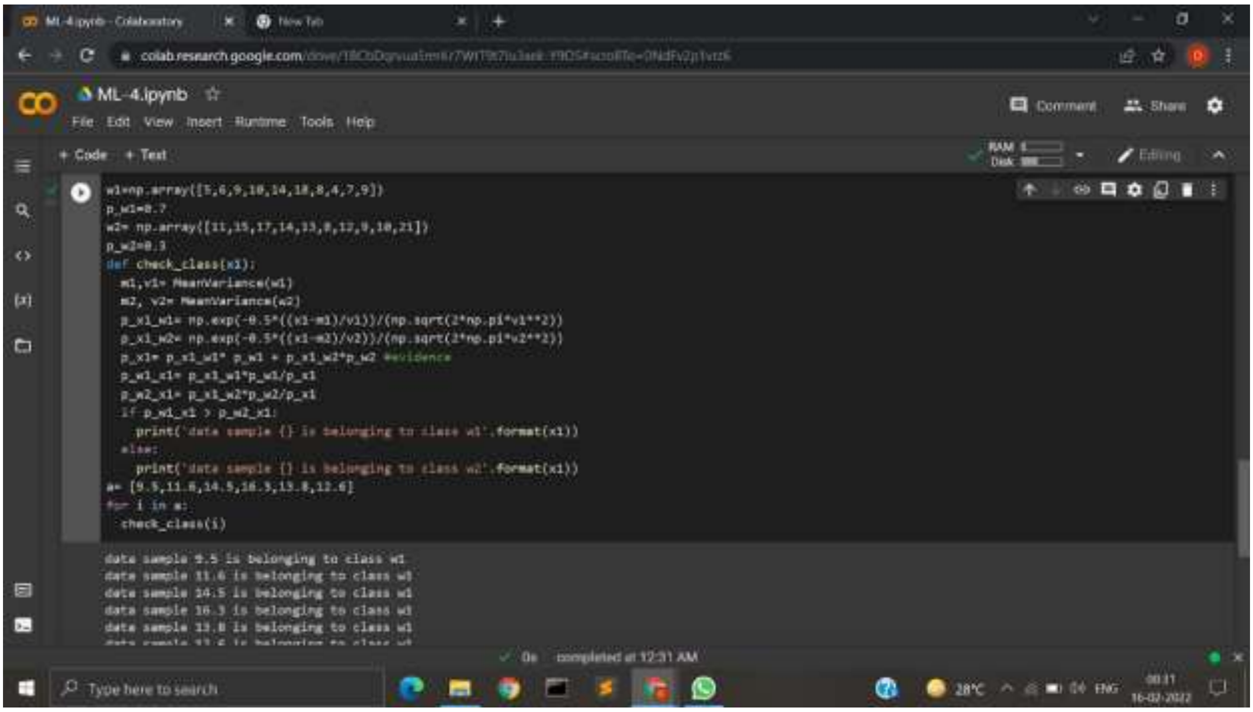
2. If w1 = [1,3,8,12] with p(w1) =0.6 and w2=[6,9,13,16] with p(w2)=0.4, find the class for test samples 7.8, 9.6 and 11.5 using baye's rule.



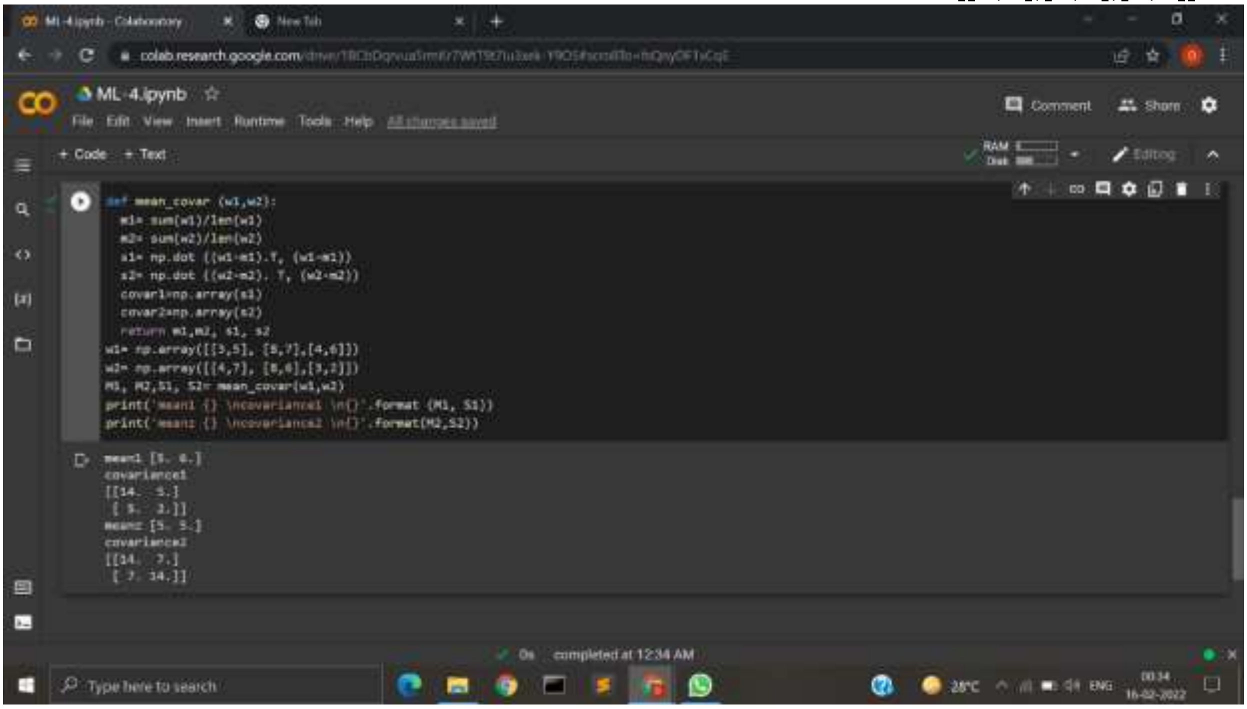
3. Define the function to compute the probability of likelihood and evidence for given 1D data and 1D test sample.



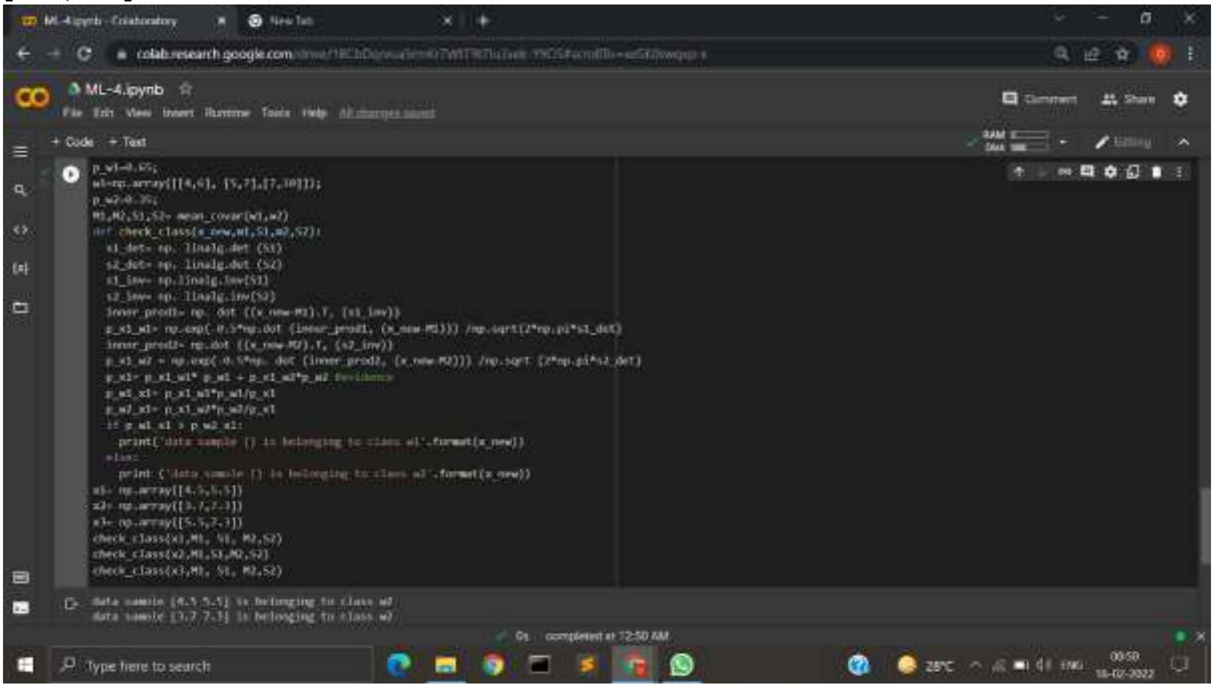
4. If w1= [5,6,9,10,14,18,8,4,7,9] with p(w1)=0.7 and w2 =[11,15,17,14,13,8,12,9,10,21] with p(w2)=0.3, find the class for test samples 9.5,11.6,14.5,16.3,13.8 and 12.6 using baye's rule by defining function.



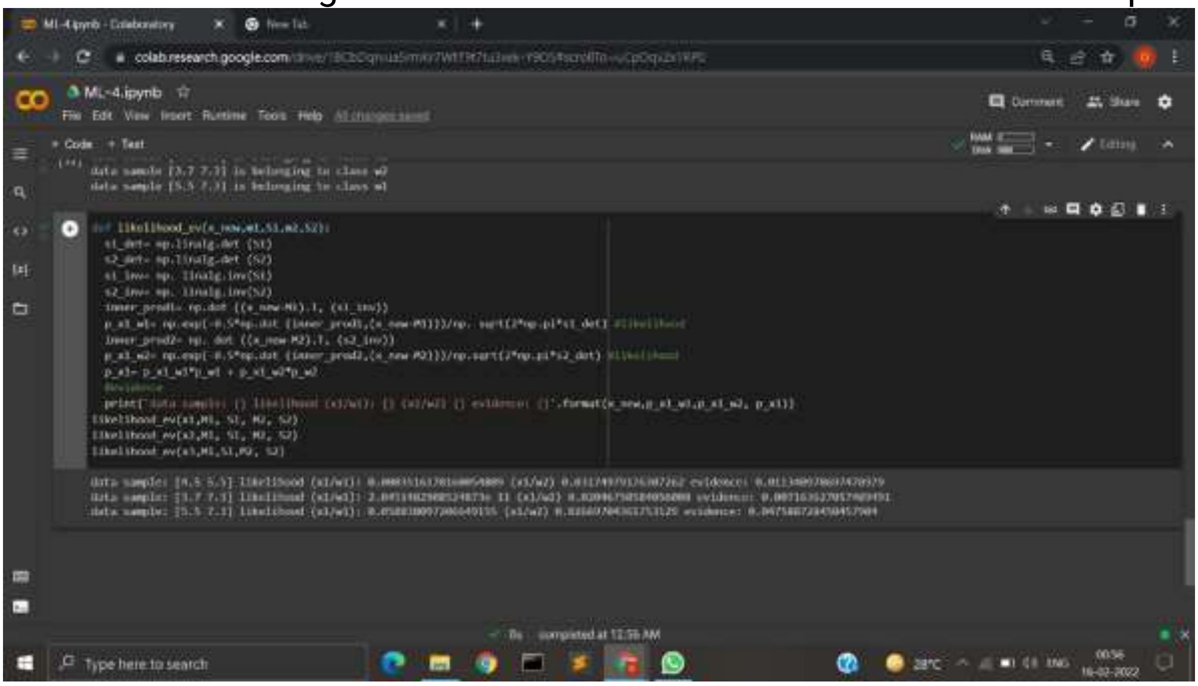
5. Define the function for computing the mean and covariance matrix for 2D data. Print the individual mean & co-variance matrix for [[3,5],[8,7],[4,6]] and [[4,7],[8,6],[3,2]].



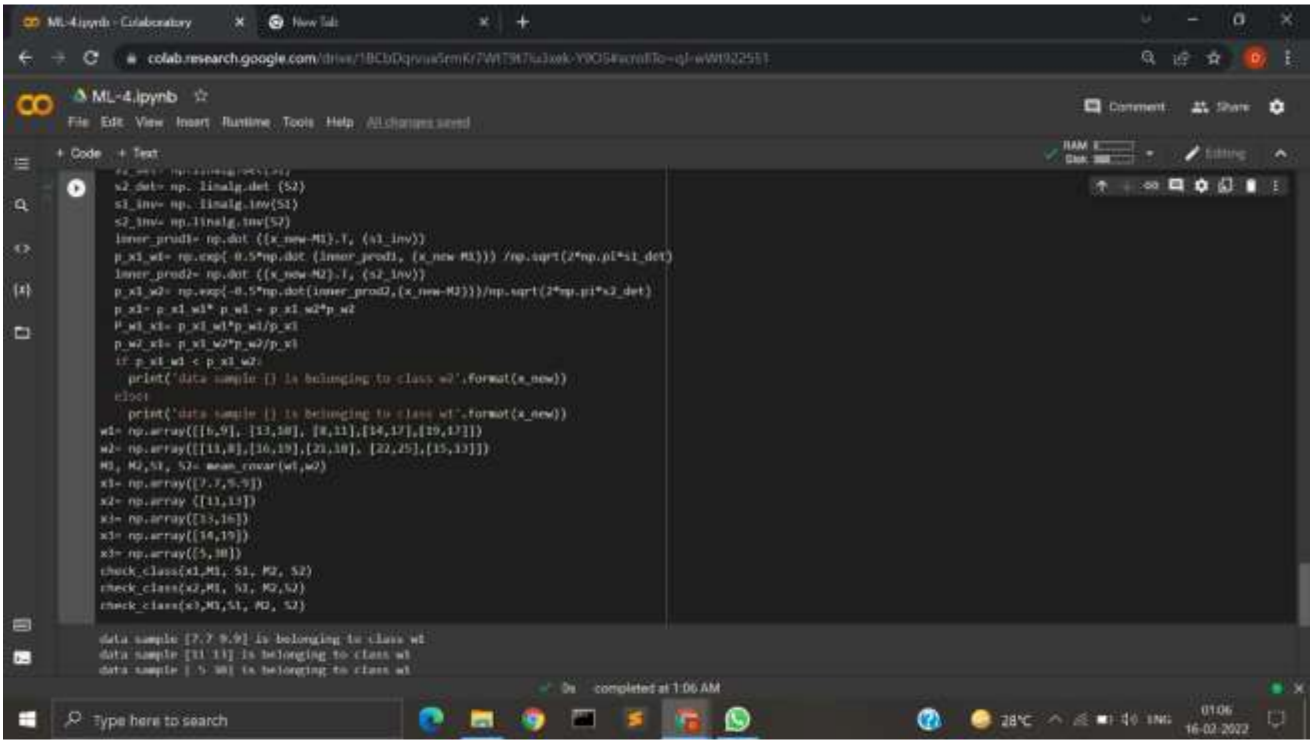
6. If w1=[[2,4],[4,3],[5,8]] with p(w1)=0.65 and w2=[[4,6],[5,7],[7,10]] with p(w2)=0.35, find the class for test samples [4.5, 5.5], [3.7, 5.8] and [5.5,7.3].



7. Define the function to compute the probability of likelihood and evidence for given 2D data and 2D test samples.



8. If w1=[[6,9],[13,10],[8,11],[14,17],[19,17]] with p(w1)=0.63 and w2=[[11,8],[16,19],[21,18],[22,25],[15,13]] with p(w2)=0.37, find the class for test samples [7.7,9.9],[11,13],[13,16],[14,19],[5,30] using baye’s rule by defining function.



CONCLUSION :- Bayes' Theorem calculates the conditional probability of an event, based on the values of specific related known probabilities. At its simplest, Bayes' Theorem takes a test result and relates it to the conditional probability of that test result given other related events.

Practical-5

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

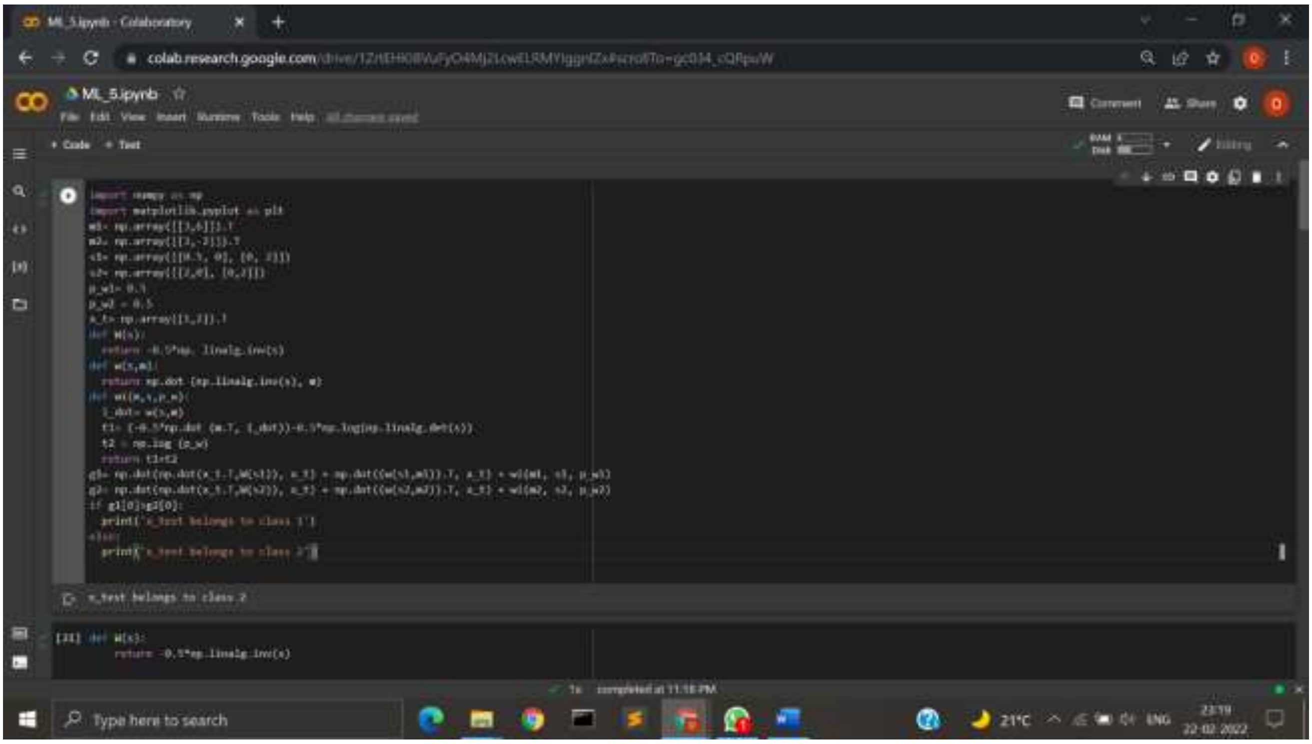
Aim: Write a python program to implement discriminant function and decision boundary.

REQUIRED SOFTWARE :- Google

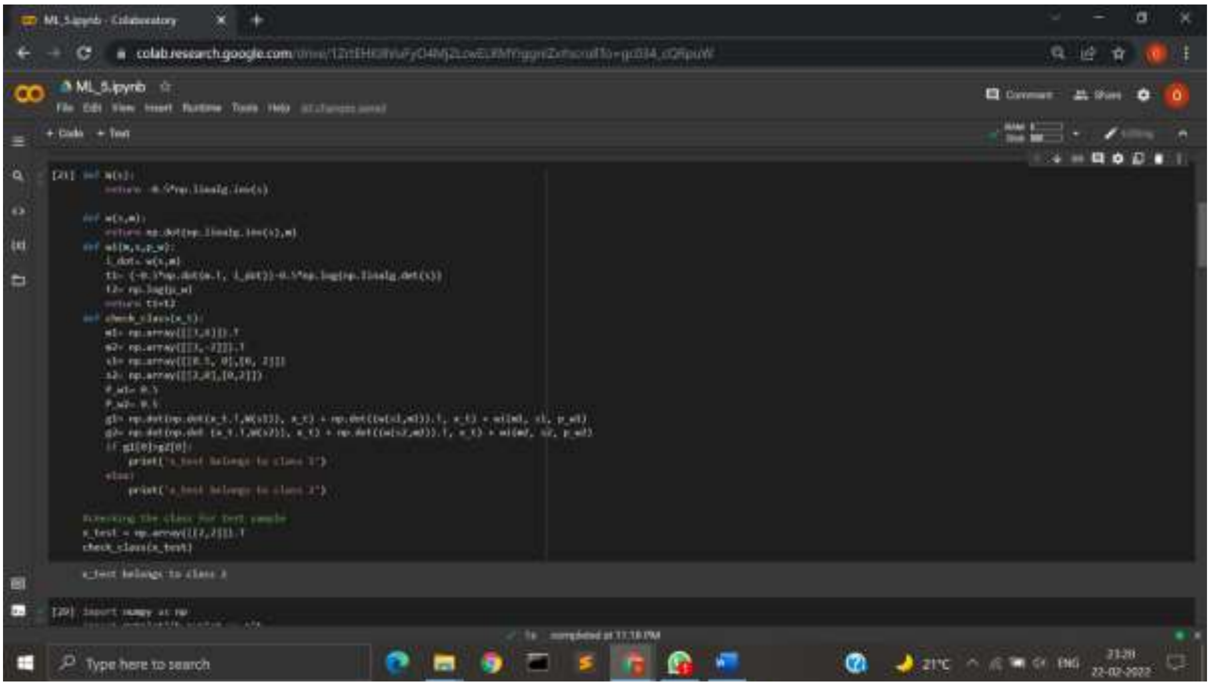
colab , PycharmTHEORY :-

Aim: Write a python program to implement discriminant function and decision boundary. A decision boundary is a line (in the case of of one class are on one side of that line, and all samples of the other class are on the opposite side of the line. The line *separates* one class from the other. If you have more than two features, the decision boundary is not a line, but a (hyper)-plane in the dimension of your feature space.

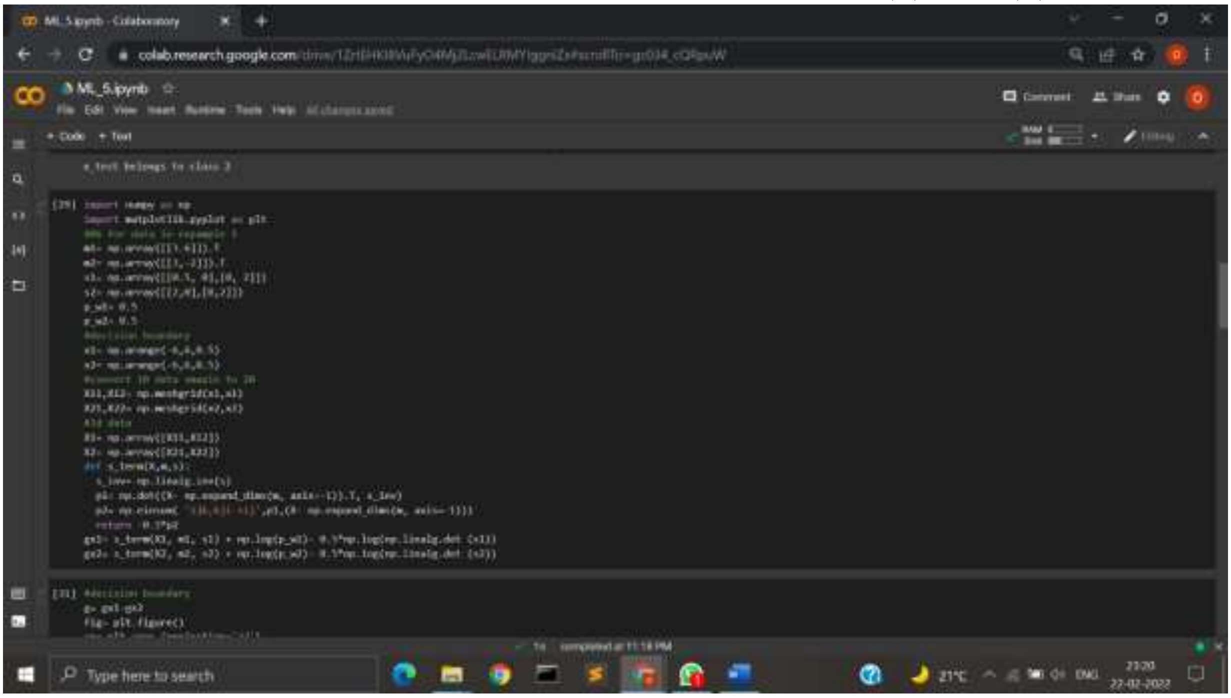
1. Classify test sample x_test = [1, 2]T using discriminant function for given data: m1 = [3, 6]T, m2= [3, -2]T, S1 = [[0.5, 0], [0, 2]], S2 = [[2, 0],[0, 2]], p(w1)=0.5, p(w2)=0.5.

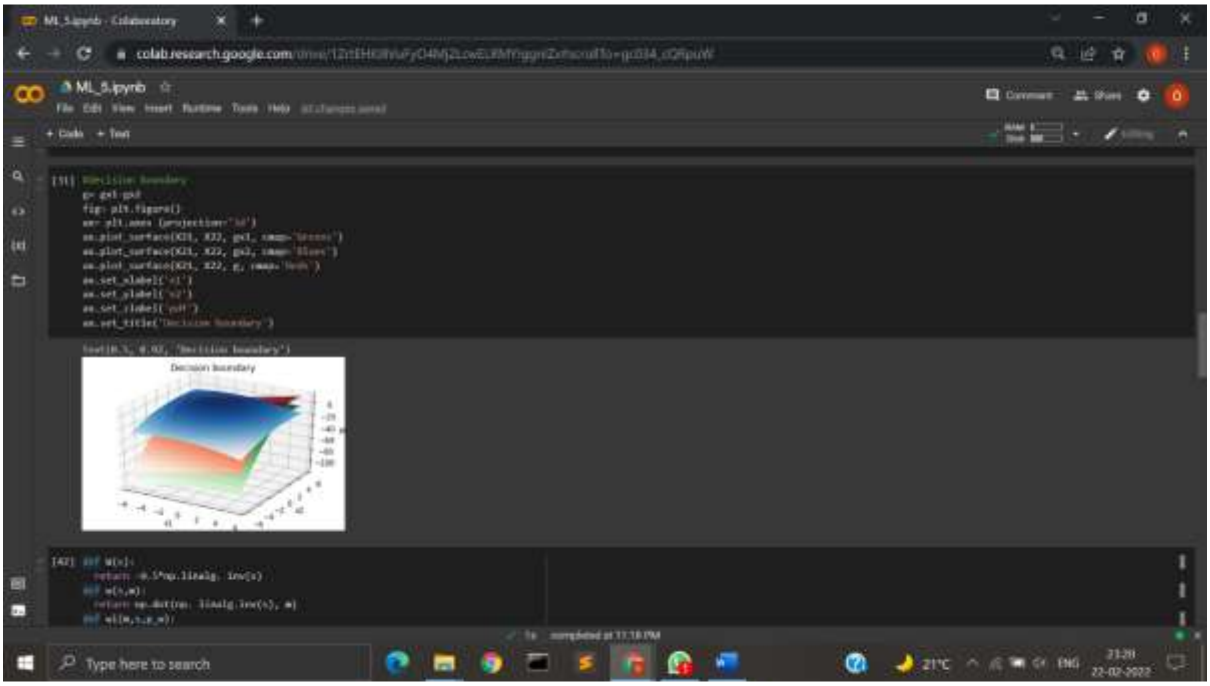


2. Classify test sample x_test = [2, 2]T for above data.

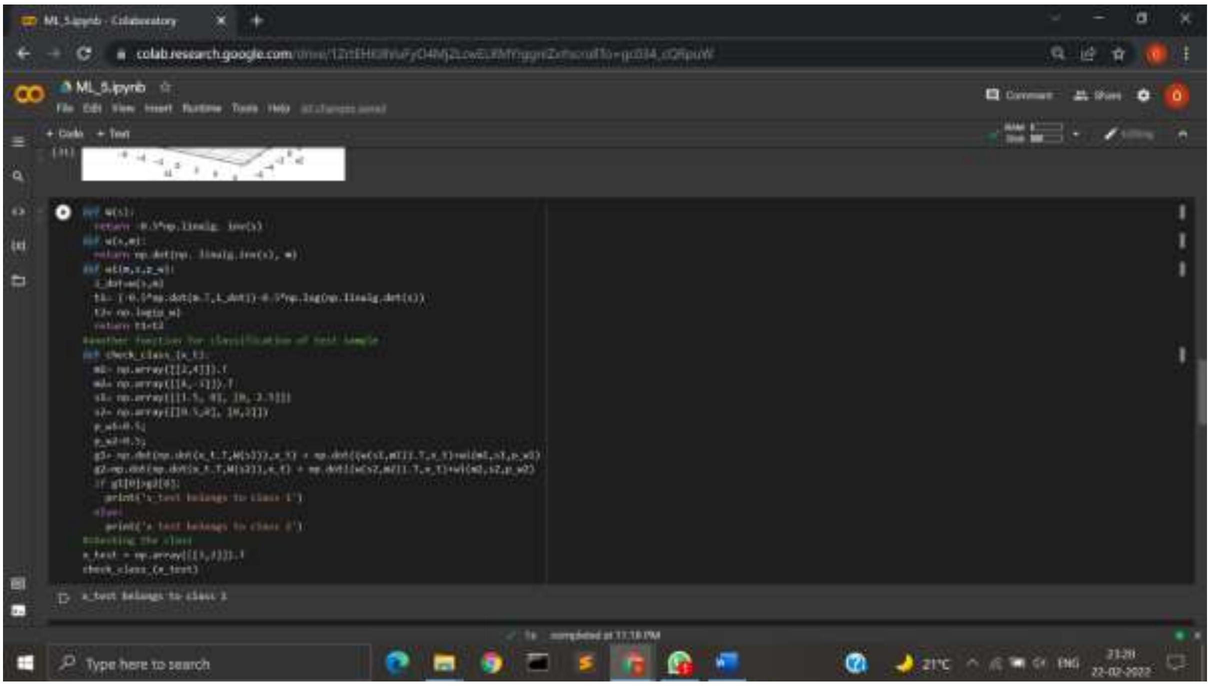


3. Find the Decision boundary for subtask 1. Take x1=np.arange(-6,6,0.5) & x2=np.arange(-6,6,0.5) and use np.meshgrid for converting 1D data to 2D data. Plot the 3D surface of g1(x), g2(x) and g.

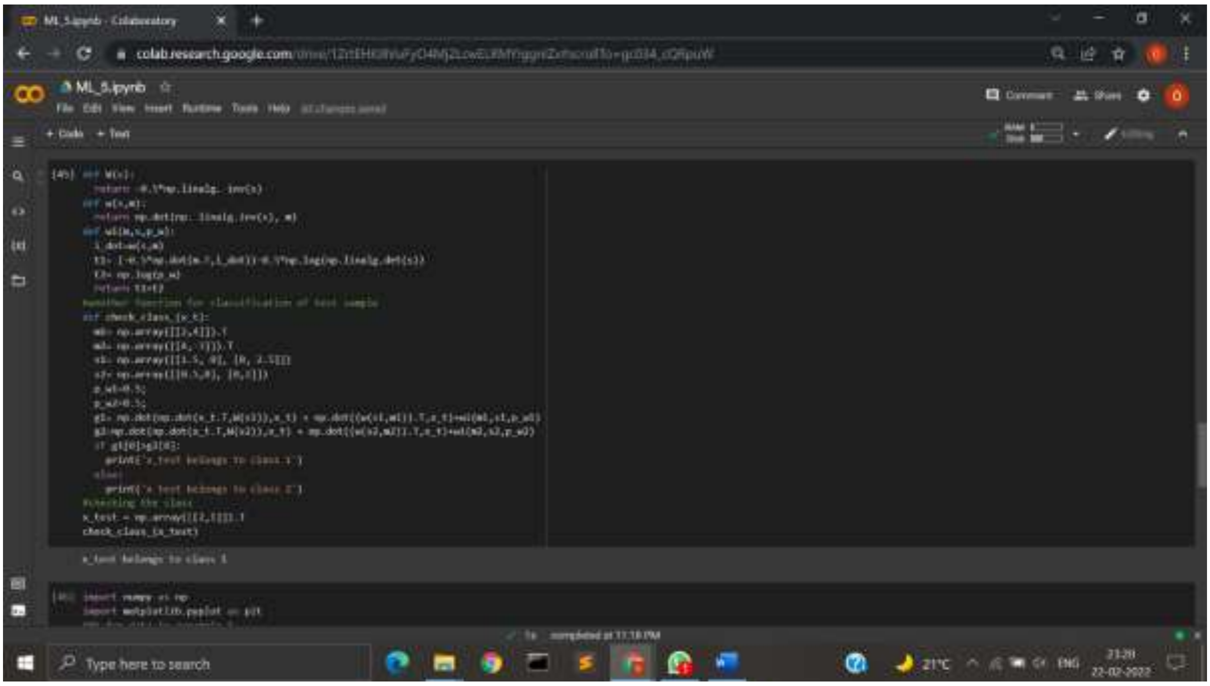




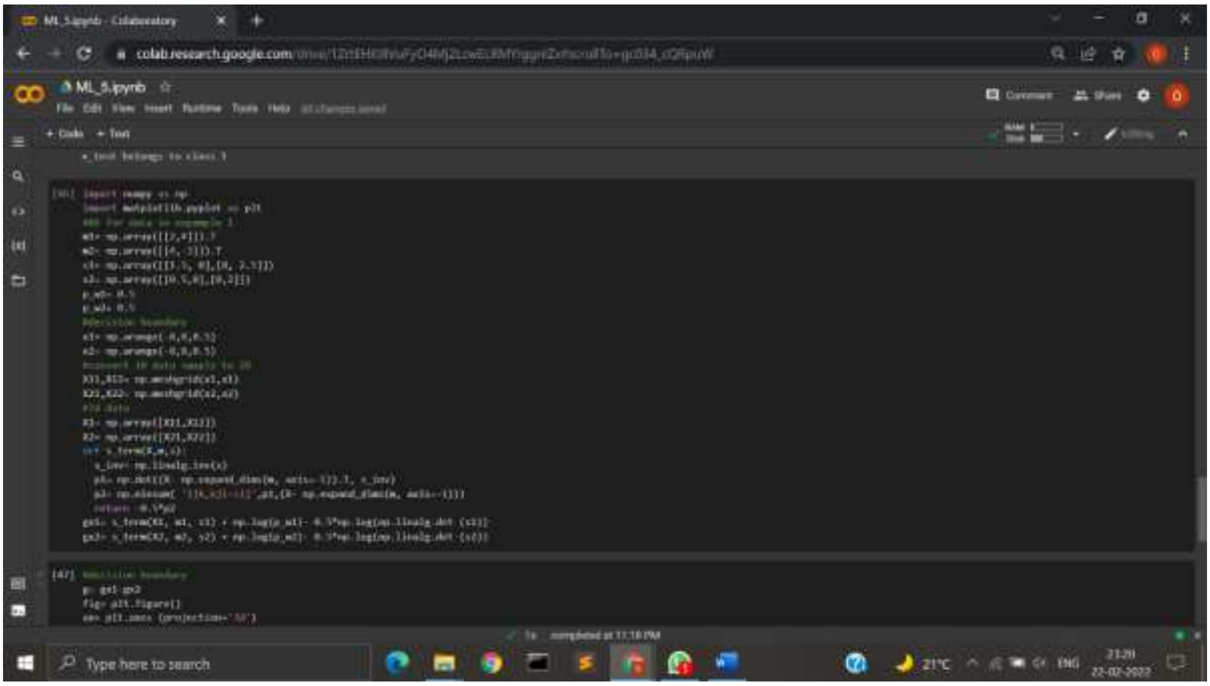
4. Classify test sample x_test = [3, 2]T using discriminant function for given data: m1 = [2, 4]T, m2= [4, -3]T, S1 = [[1.5, 0], [0, 2.5]], S2 = [[0.5, 0],[0, 2]], p(w1)=0.5, p(w2)=0.5

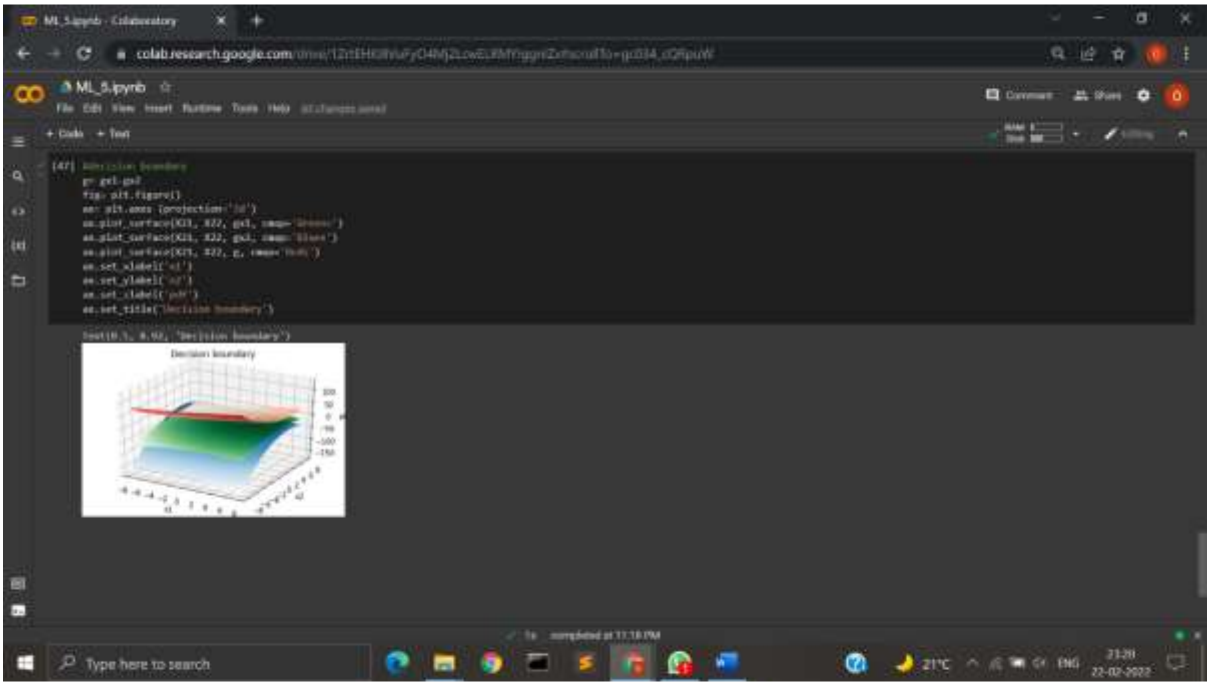


5. Classify test sample x_test = [2, 1]T for above data.



6. Find the Decision boundary for subtask 4. Take x1=np.arange(-8,8,0.5) & x2=np.arange(-8,8,0.5) and use np.meshgrid for converting 1D data to 2D data. Plot the 3D surface of g1(x), g2(x) and g.





CONCLUSION:- A decision boundary or decision surface is a hyper surface that partitions the underlying vector space into two sets, one for each class. The classifier will classify all the points on one side of the decision boundary as belonging to one class and all those on the otherside as belonging to the other class.

Practical-6

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM: - Write a python program to implement

Naïve Bayes Classifier.

SOFTWARE :- Google colab , Pycharm

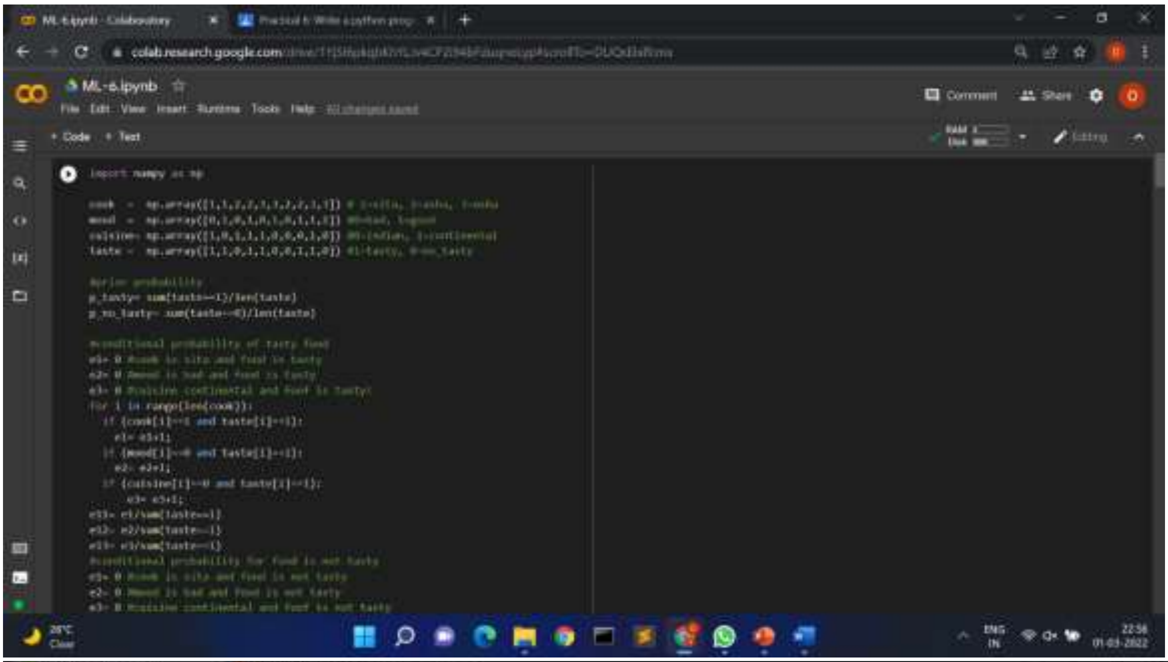
THEORY:-

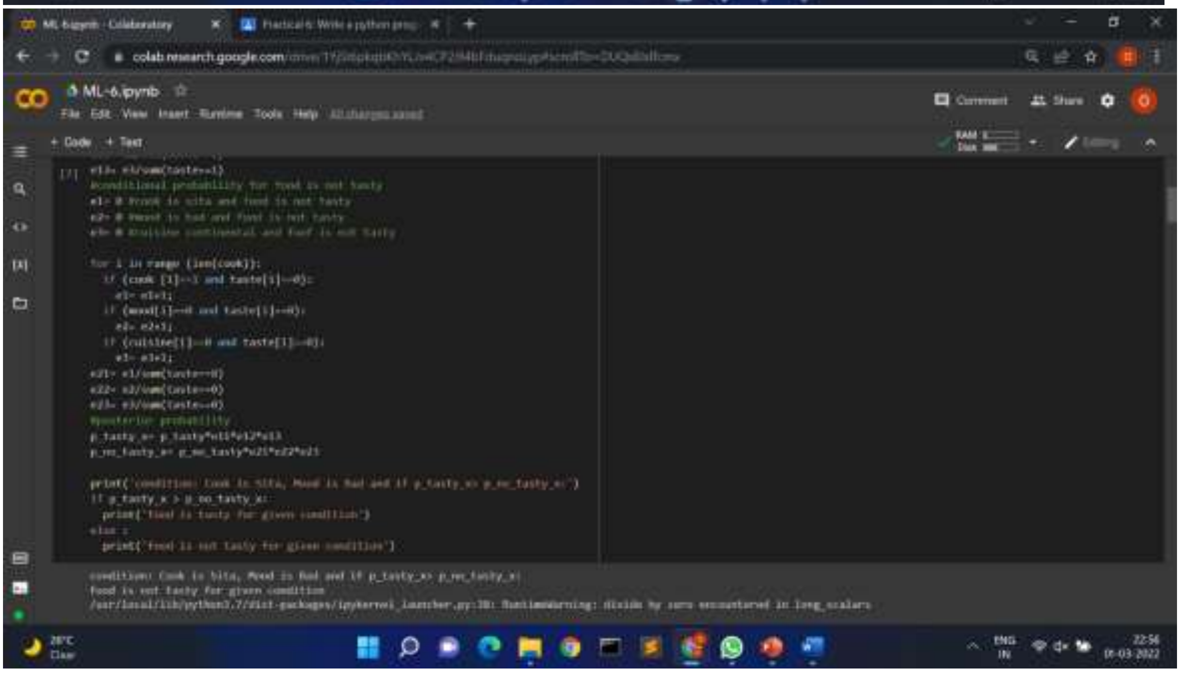
Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.
Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.[6] Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.[7]

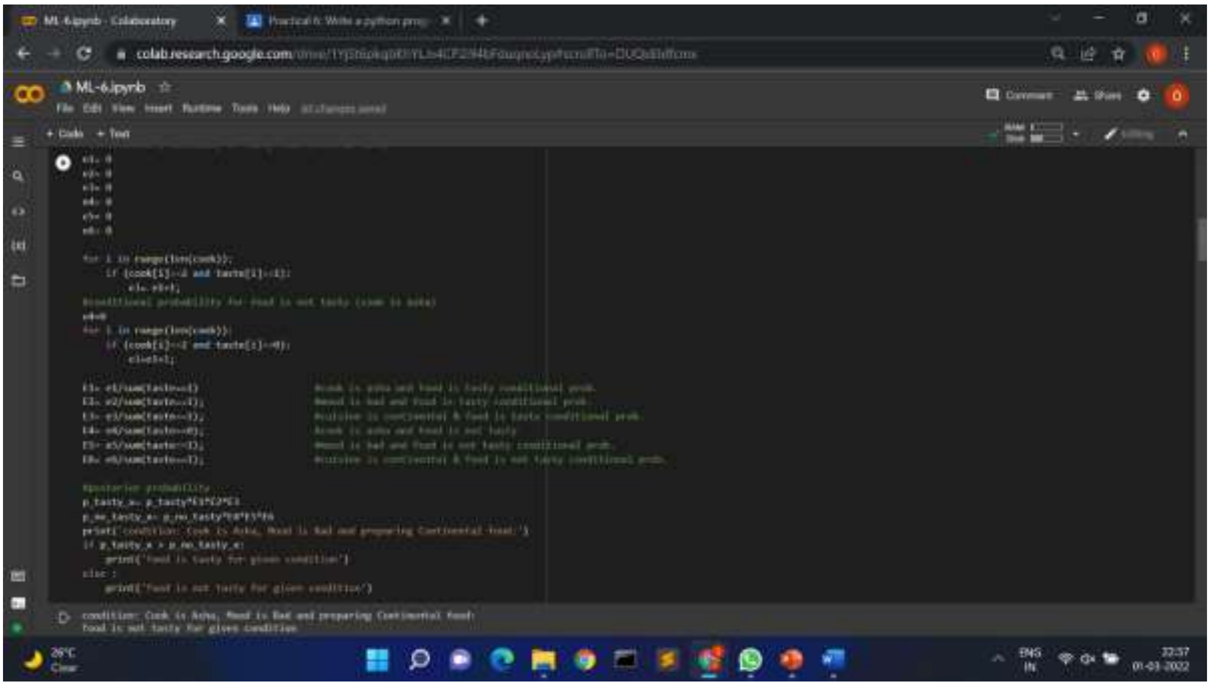
An advantage of naive Bayes is that it only requires a small number of training data to estimatethe parameters necessary for classification.

1.If Cook is Sita, Mood is Bad and preparing Continental food, is food tasty?

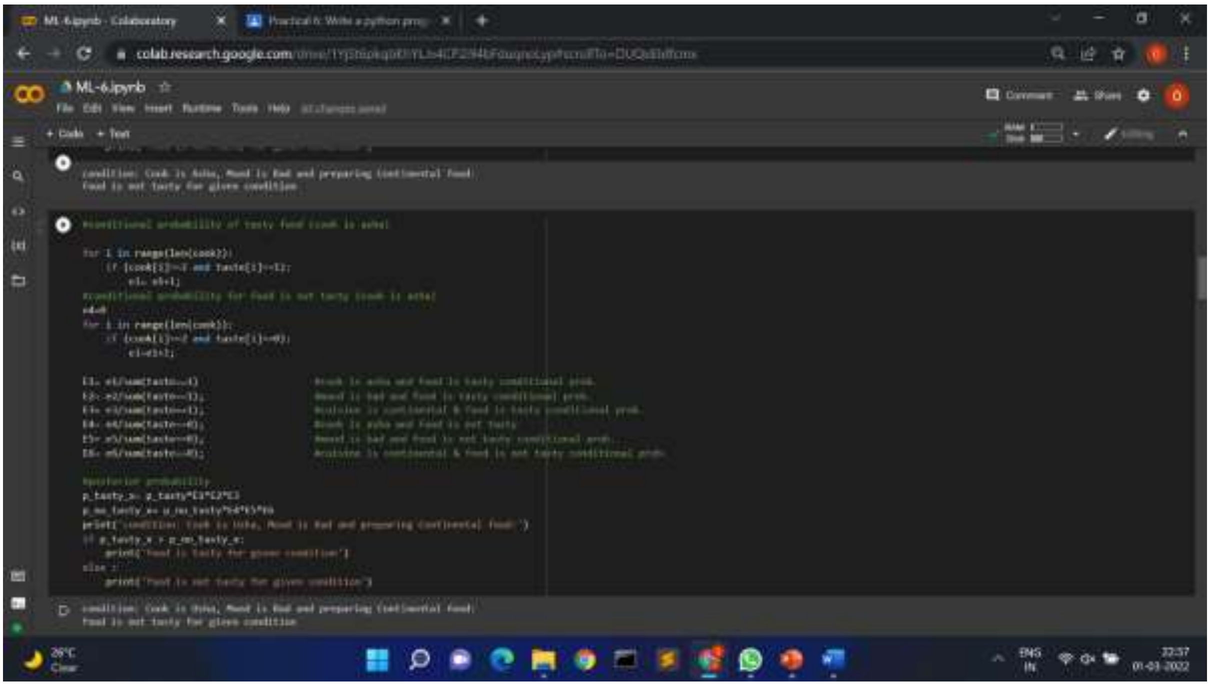




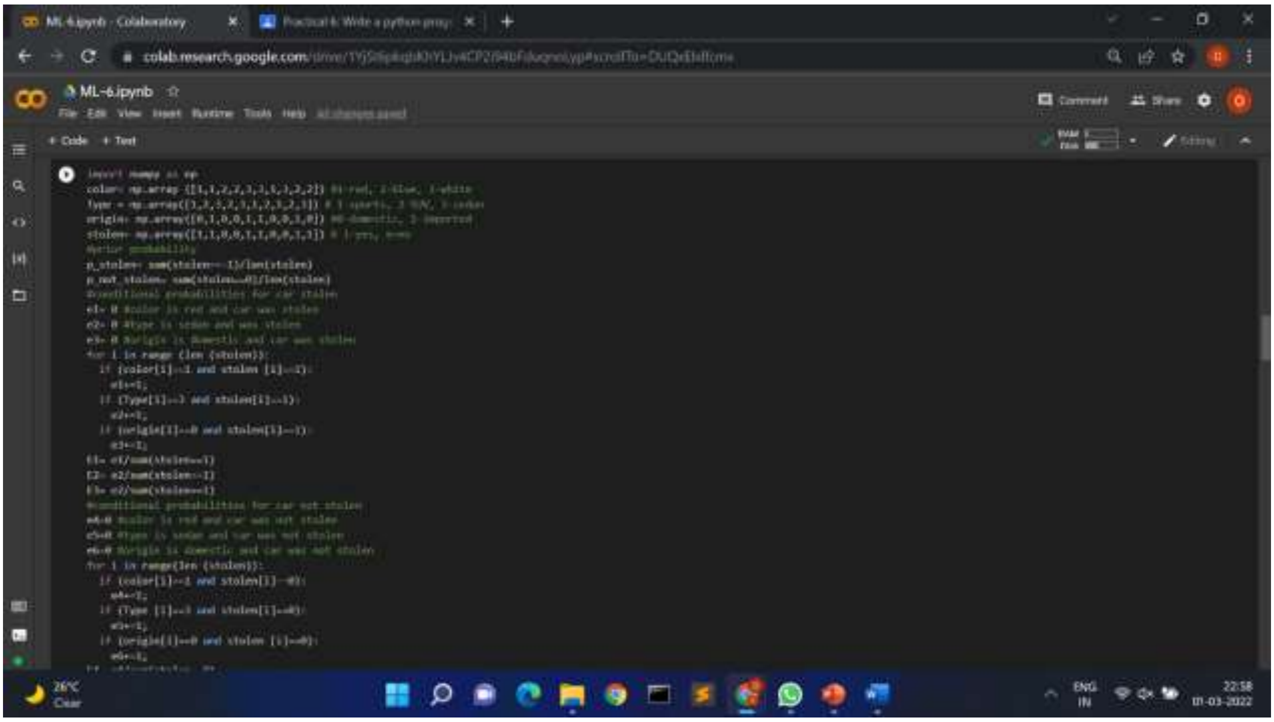
2.If Cook is Asha for above case, is food tasty?

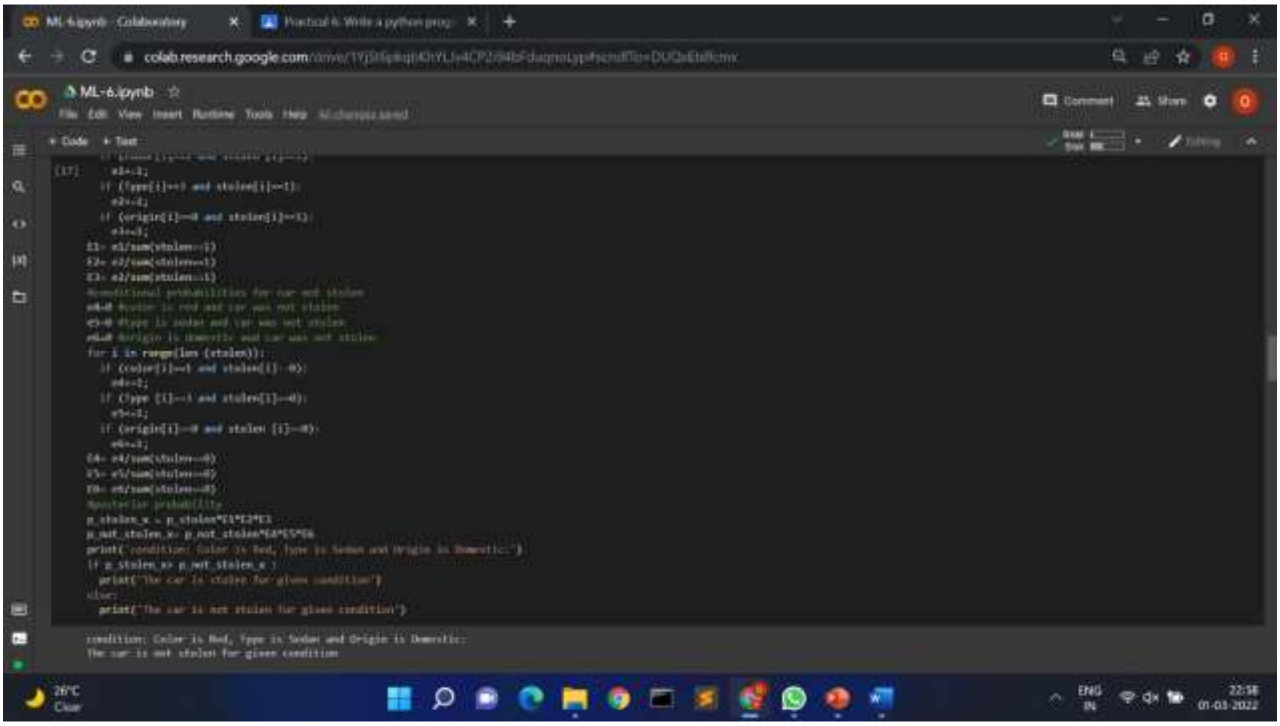


3.If Cook is Usha, Mood is Bad and preparing Continental food, is food tasty?

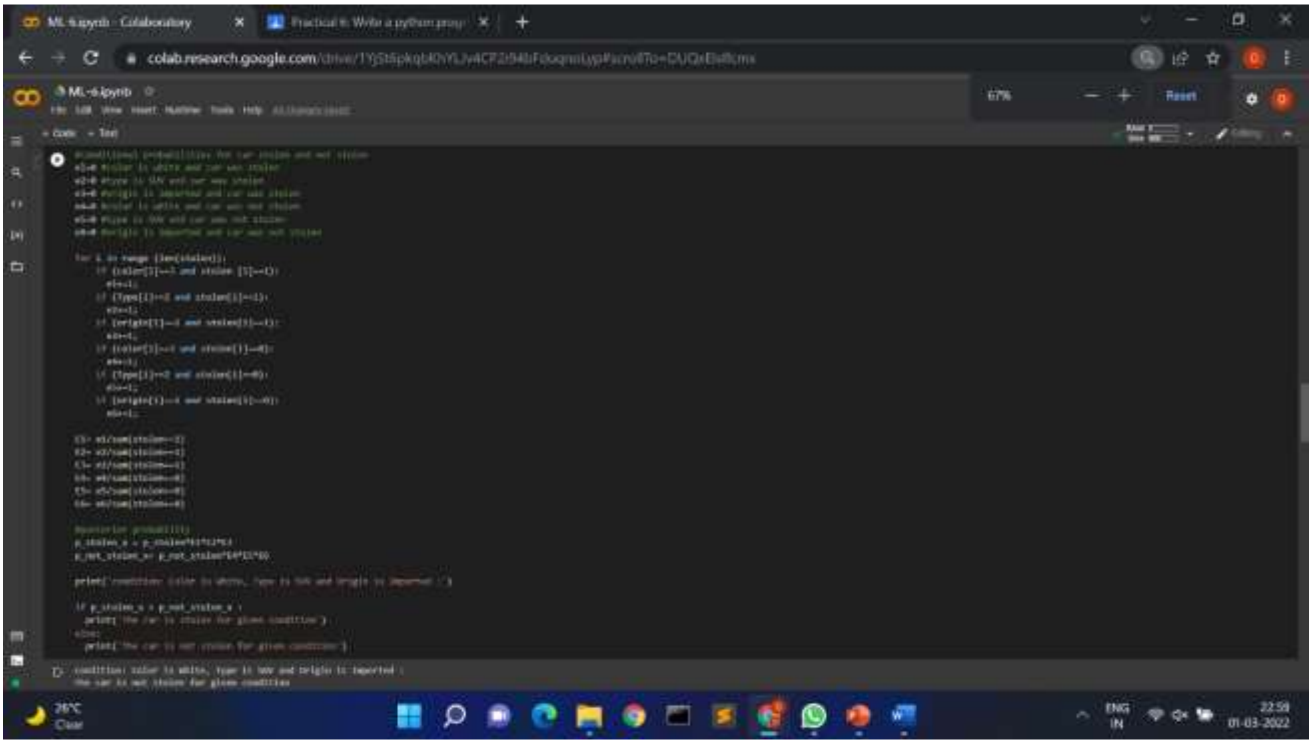


4.If car Colour is Red, Type is Sedan and Origin is Domestic, was car stolen?

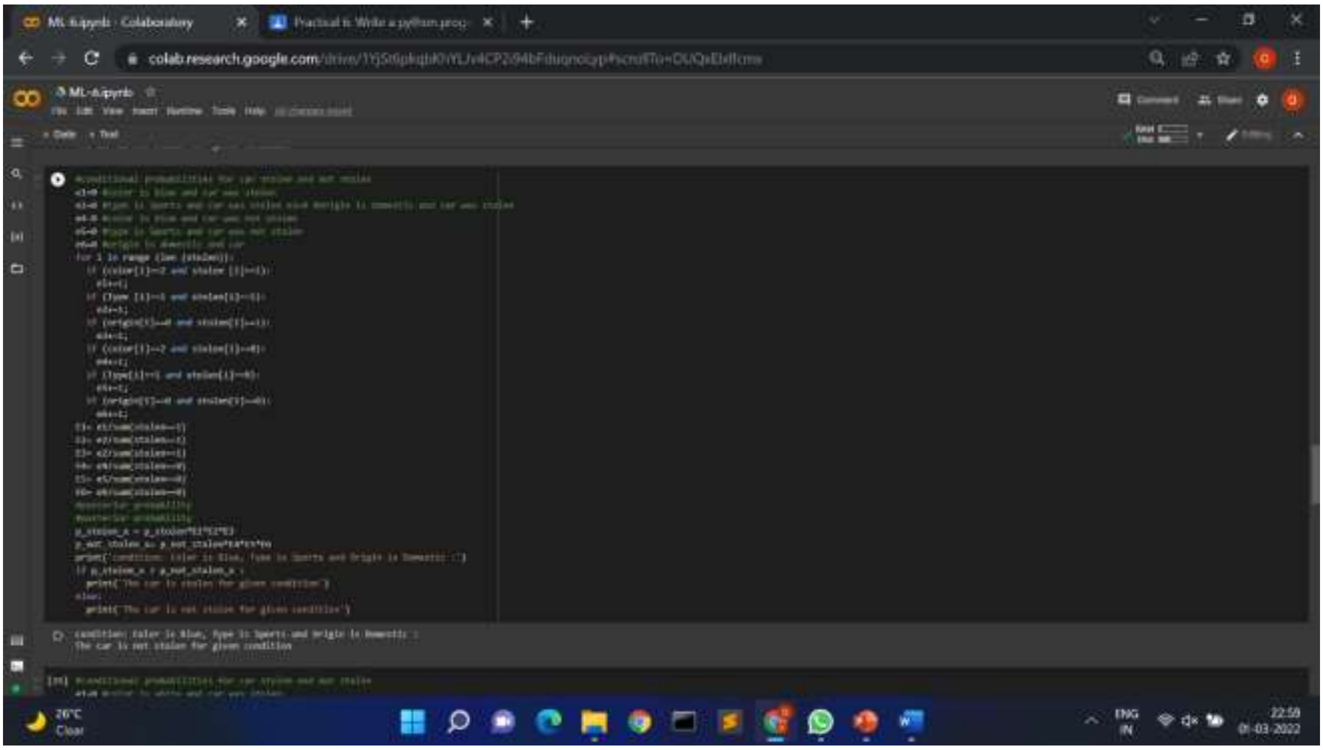




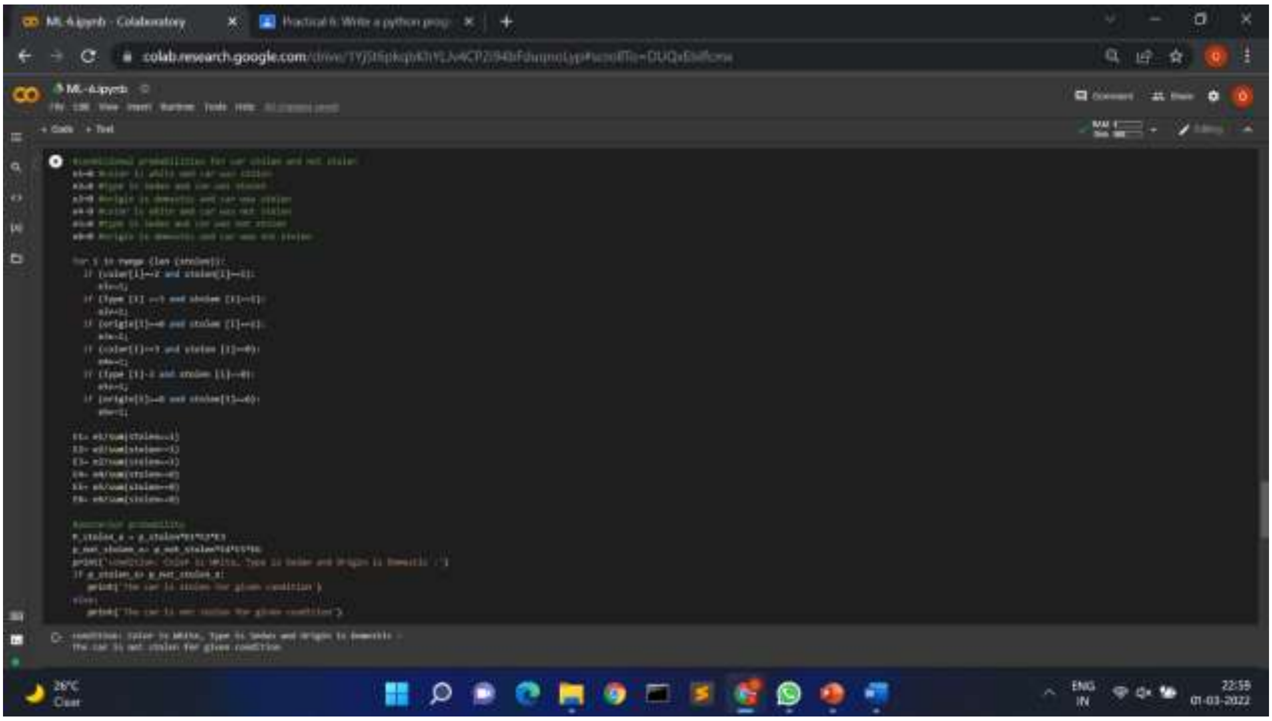
5.If car Colour is White, Type is SUV and Origin is Imported, was car stolen?



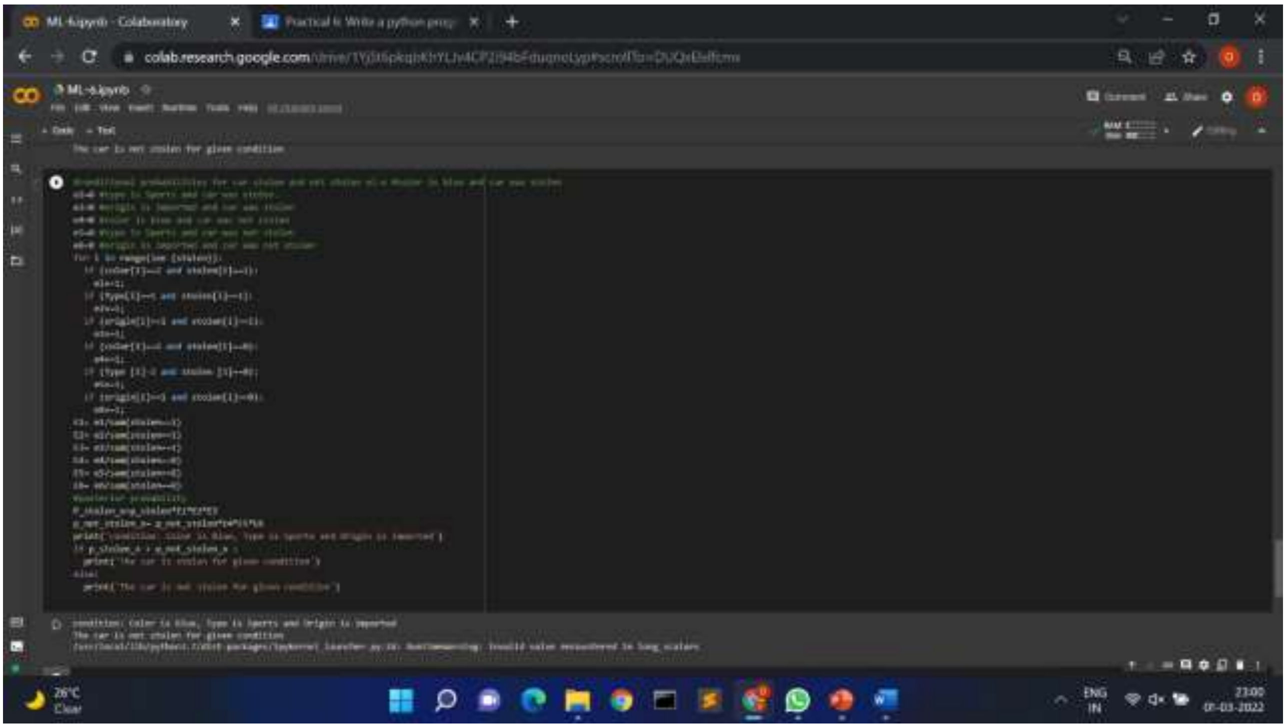
6.If car Colour is Blue, Type is Sports and Origin is Domestic, was car stolen?



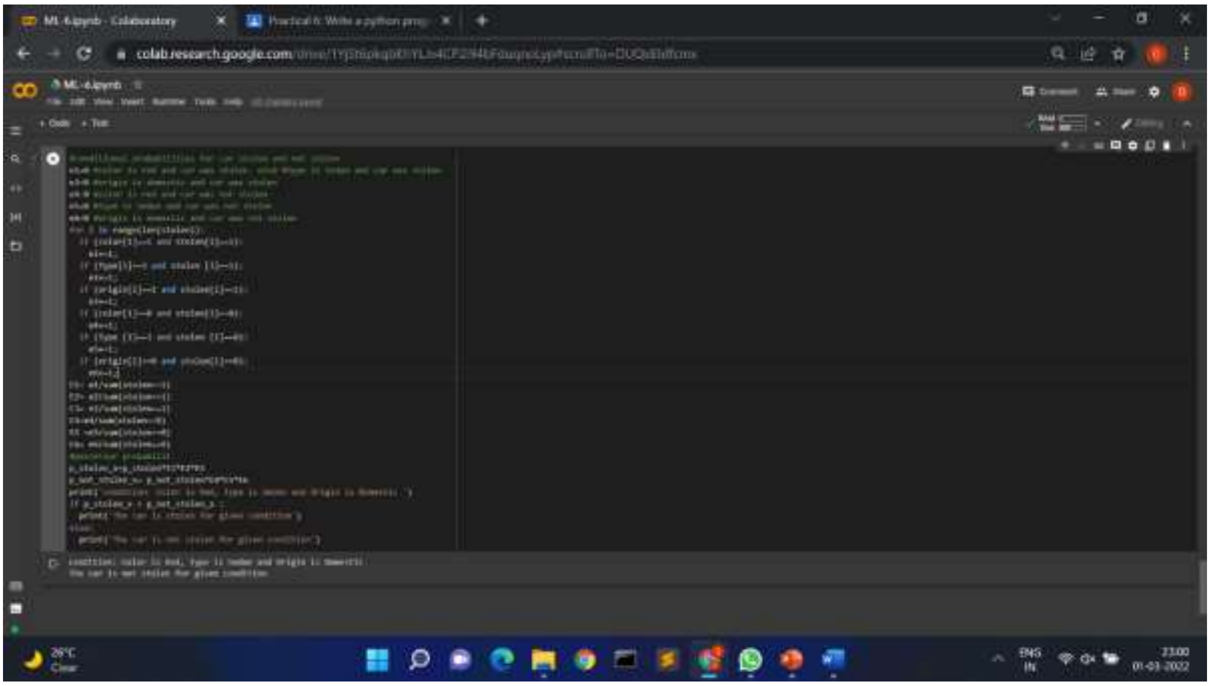
7.If car Colour is White, Type is Sedan and Origin is Domestic, was car stolen?



8.If car Colour is Bleu, Type is Sports and Origin is Imported, was car stolen?



9.If car Colour is Red, Type is Sedan and Origin is Domestic, was car stolen?



CONCLUSION: -

Hence, Naïve Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem to predict the outcome. It is a family of algorithms where all of them share a common principle. Naïve Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems, etc. They are fast and easy to implement

Practical-7

Name : Naik Onkar Chandrakant

Reg. No. : 2019BEC135

Roll No. : A65

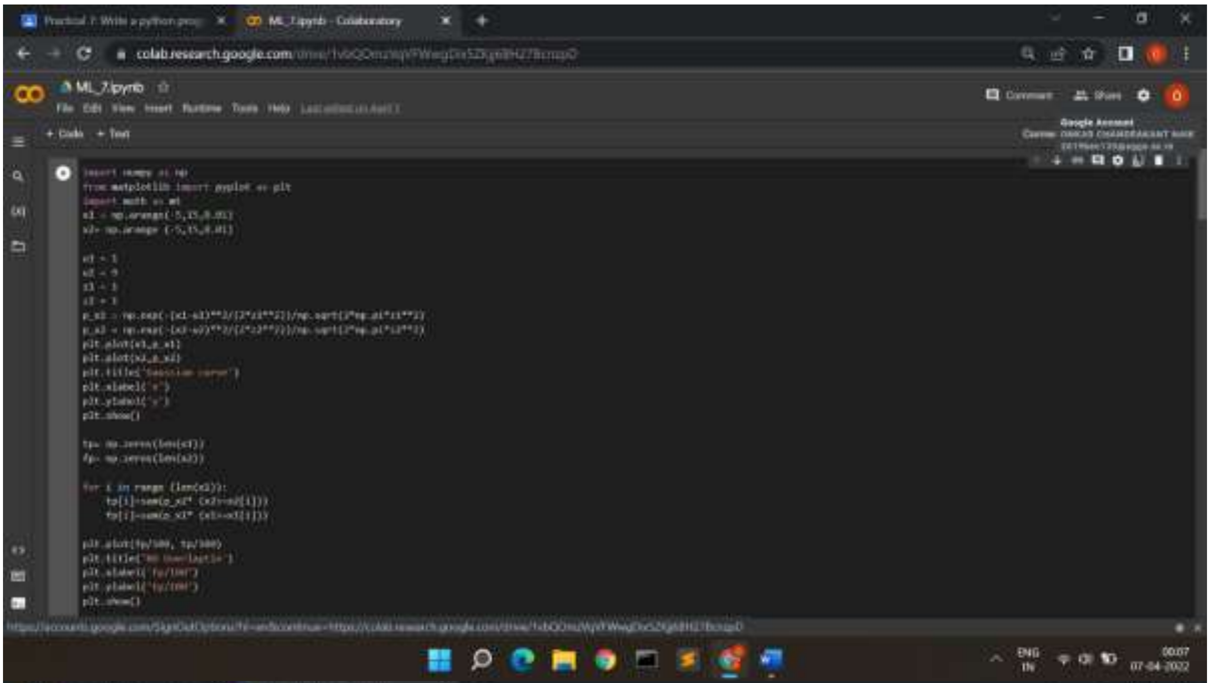
AIM : - Write a python program to plot Receiver Operating Characteristics (ROC) curves.REQUIRED SOFTWARE :- Google colab , Pycharm

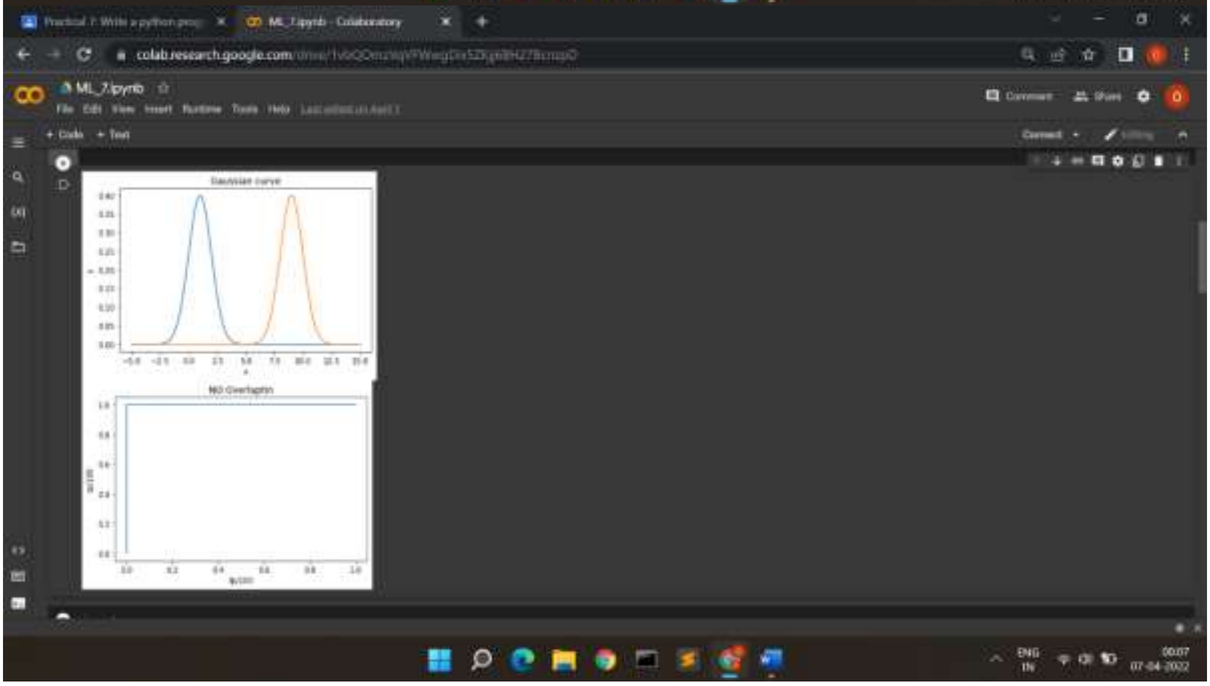
THEORY :-

ROC curve, also known as Receiver Operating Characteristics Curve, is a metric used to measure the performance of a classifier model. The ROC curve depicts the rate of true positives with respect to the rate of false positives, therefore highlighting the sensitivity of the classifier model. The ROC is also known as a relative operating characteristic curve, as it is acomparison of two operating characteristics, the True Positive Rate and the False Positive Rate, as the criterion changes. An ideal classifier will have a ROC where the graph would hit a true positive rate of 100% with zero false positives. We generally measure how many correct positive classifications are being gained with an increment in the rate of false positives.

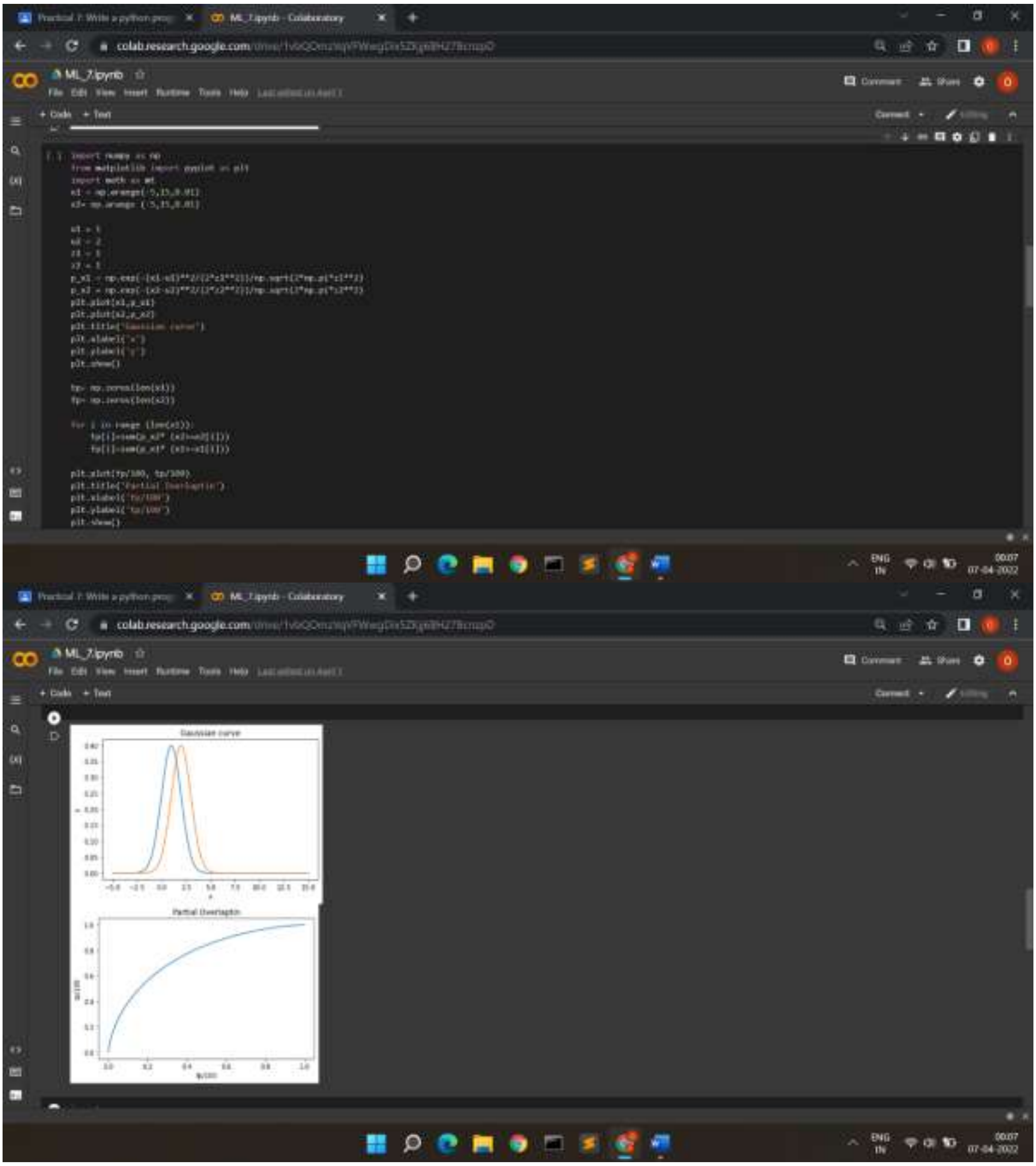
ROC curve can be used to select a threshold for a classifier, which maximizes the true positives and in turn minimizes the false positives. ROC Curves help determine the exact trade-off between the true positive rate and false-positive rate for a model using different measures of probability thresholds. ROC curves are more appropriate to be used when the observations present are balanced between each class. This method was first used in signal detection but is now also being used in many other areas such as medicine, radiology, natural hazards other than machine learning. A discrete classifier returns only the predicted class and gives a single point on the ROC space. But for probabilistic classifiers, which give aprobability or score that reflects the degree to which an instance belongs to one class rather than another, we can create a curve by changing the threshold for the score.

1)Plot the “No Overlap” condition of two Gaussian distributions. Also plot the ROC curve for same. Give proper labeling to plot.

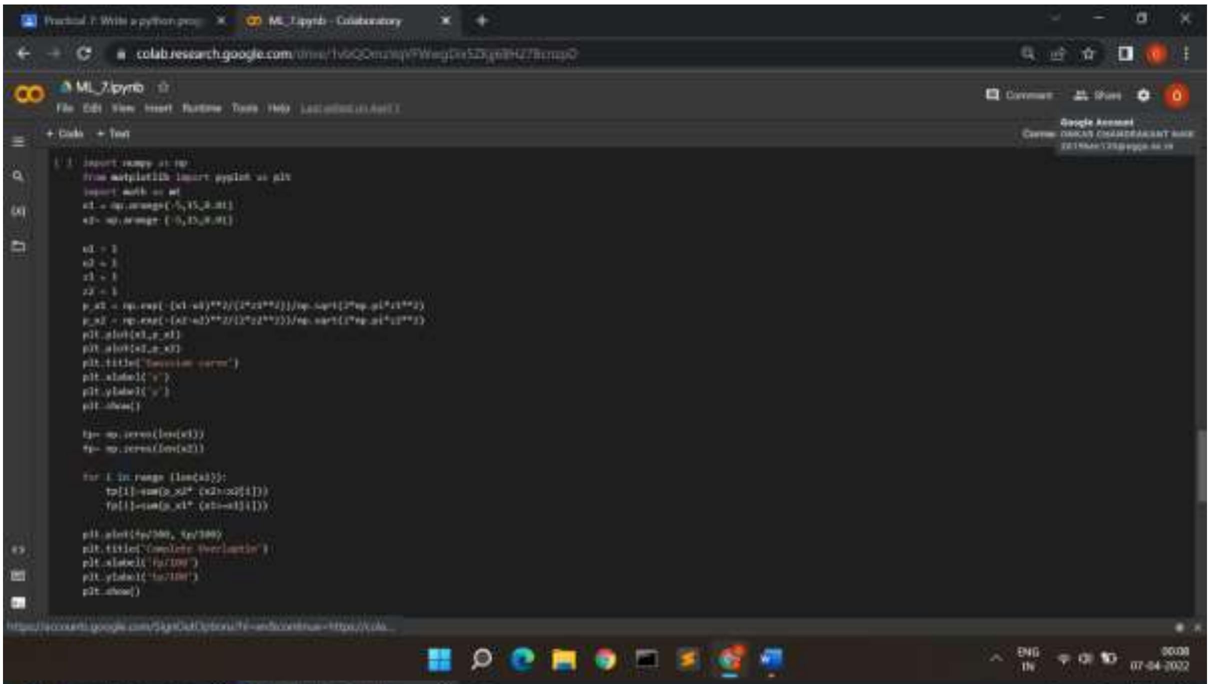


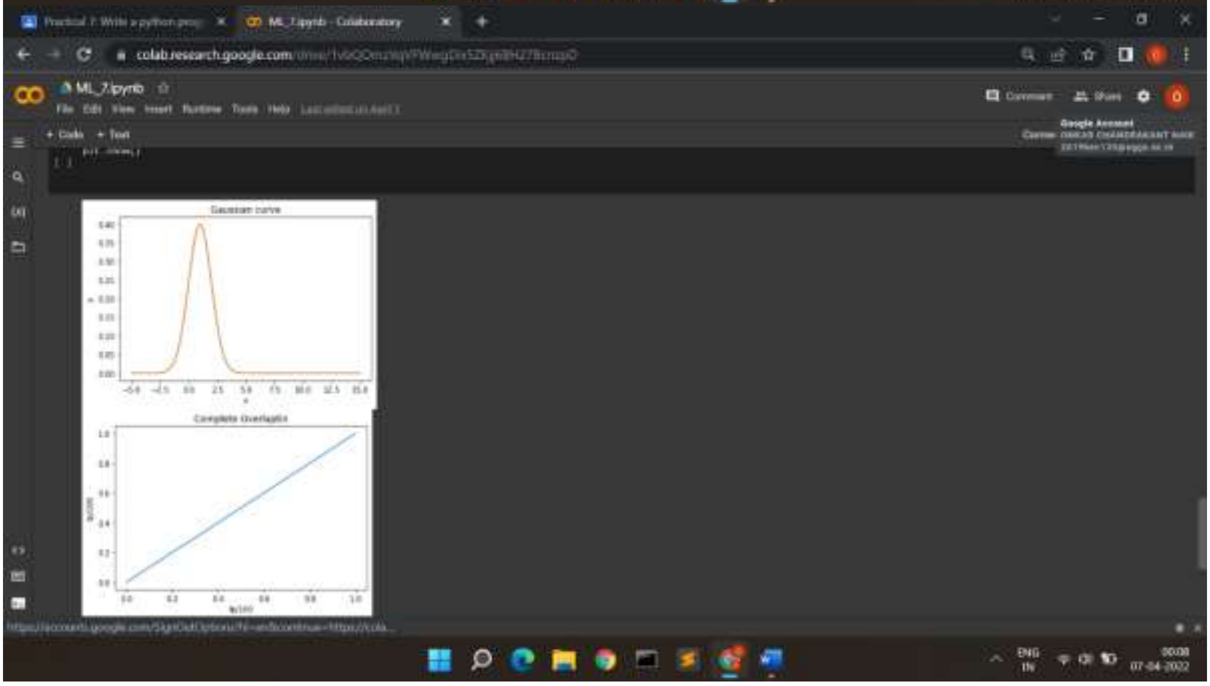


2)Plot the “Partial Overlap” condition of two Gaussian distributions. Also plot the ROC curve for same. Give proper labeling to plot.



3)Plot the “Complete Overlap” condition of two Gaussian distributions. Also plot the ROC curve for same. Give proper labeling to plot.





CONCLUSION :- An ROC curve **(**receiver operating characteristic curve**)** is a graph showing the performance of a classification model at all classification thresholds. An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives

Practical-8

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM :- Write a python program for Multivariate Regression Modeling

SOFTWARE :- Google colab , Pycharm

THEORY :-

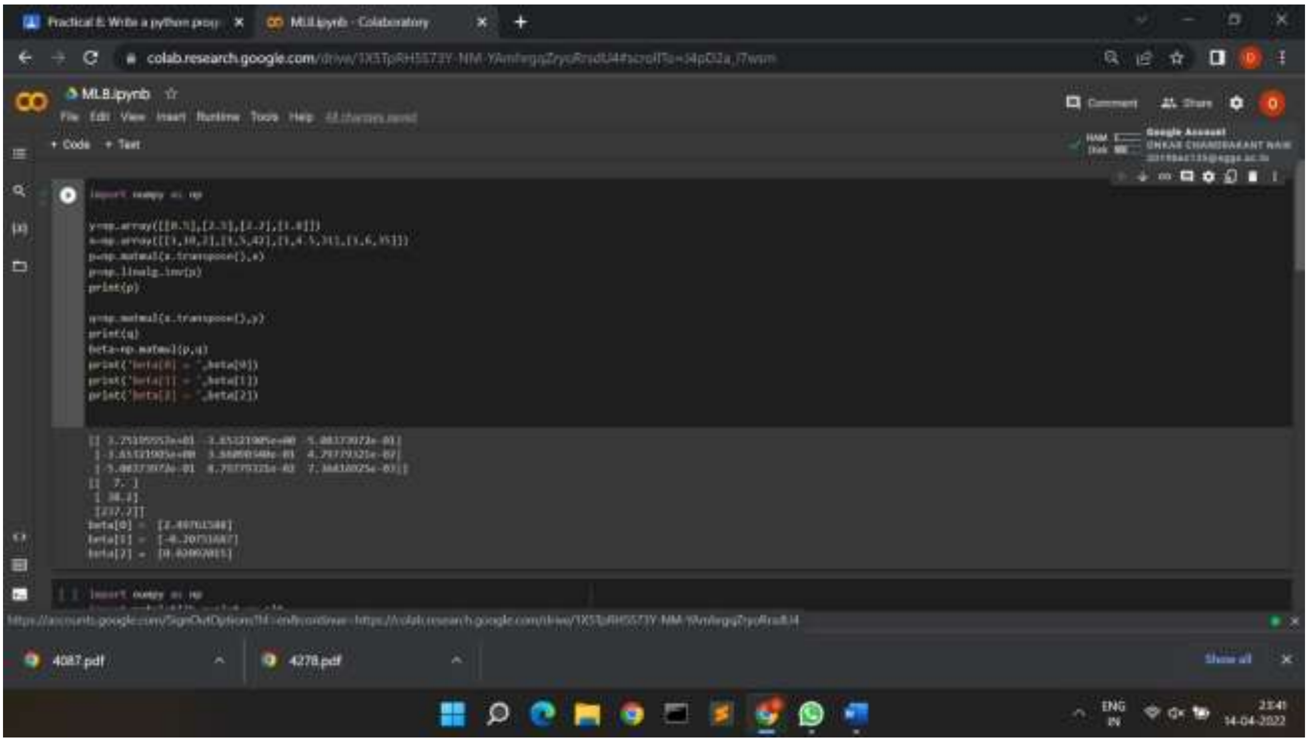
Multivariate regression is a technique used to measure the degree to which the various independent variable and various dependent variables are linearly related to each other.The relation is said to be linear due to the correlation between the variables. Once the multivariate regression is applied to the dataset, this method is then used to predict the behaviour of the response variable based on its corresponding predictor variables.

Multivariate regression is commonly used as a supervised algorithm in machine learning, amodel to predict the behaviour of dependent variables and multiple independent variables.

Characteristics of multivariate regression

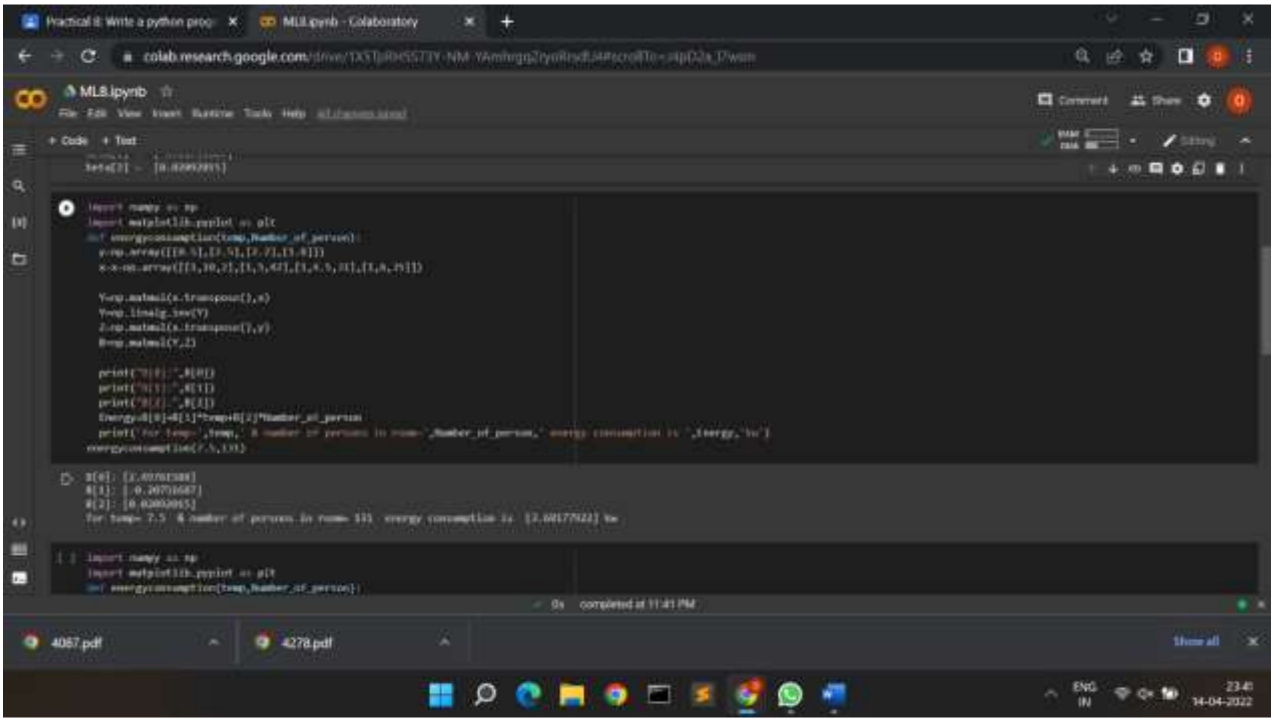
- Multivariate regression allows one to have a different view of the relationshipbetween various variables from all the possible angles.
- It helps you to predict the behaviour of the response variables depending onhow the predictor variables move.
- Multivariate regression can be applied to various machine learning fields,economic, science and medical research studies.

1.Consider the attached dataset (Mult_reg.png) of power consumption containing (4 rows and 3 column). Define the function for computing the regression coefficients. Print out the coefficients.

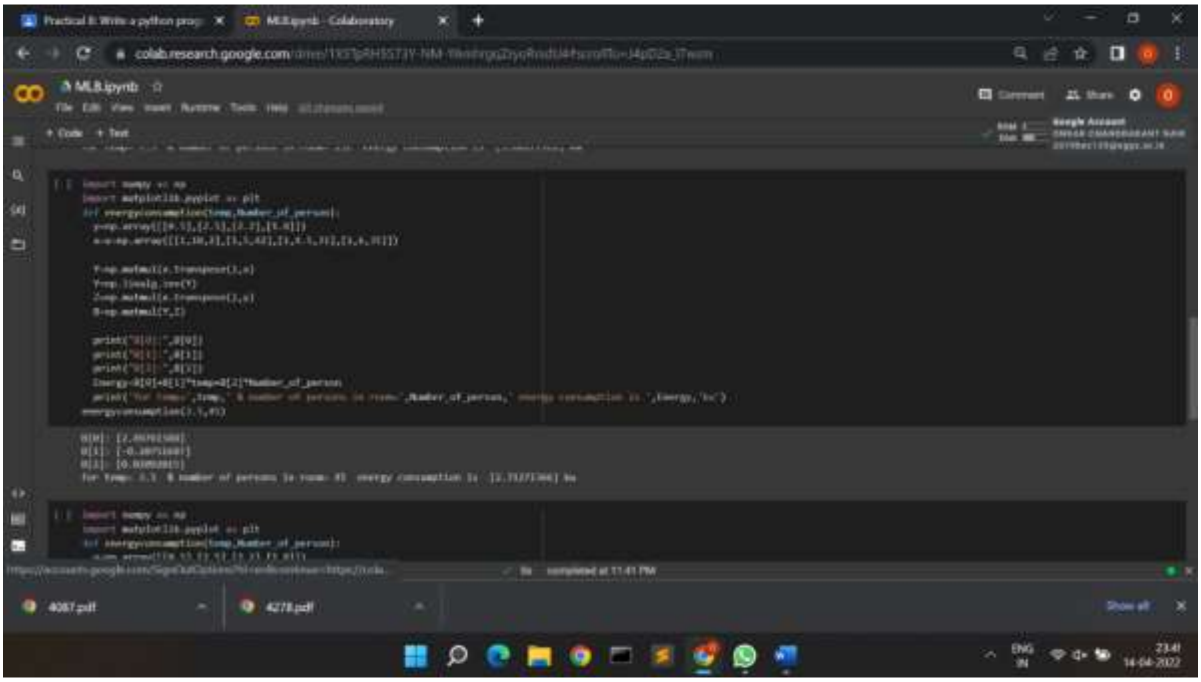


2.Define the function for computing energy consumption for given regression coefficients as inputs as follows. Print the output describing energy required (KW) for given temp and no. of persons.

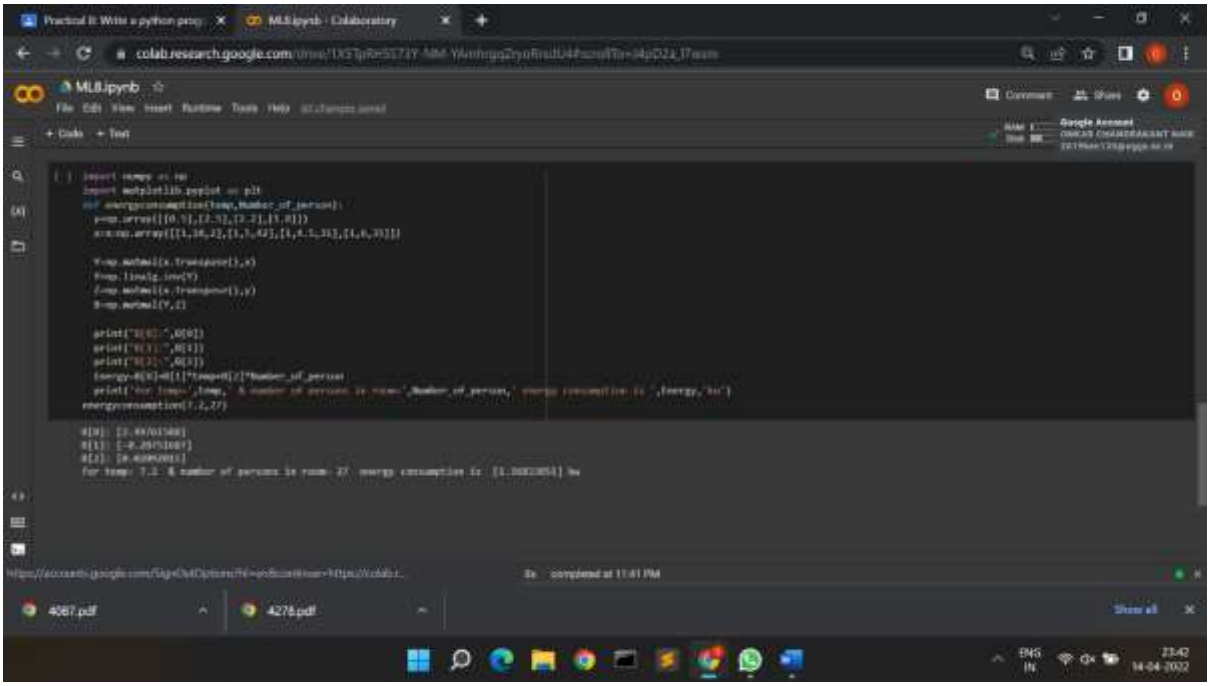
I. temp = 7.5 and No. of persons in room = 131



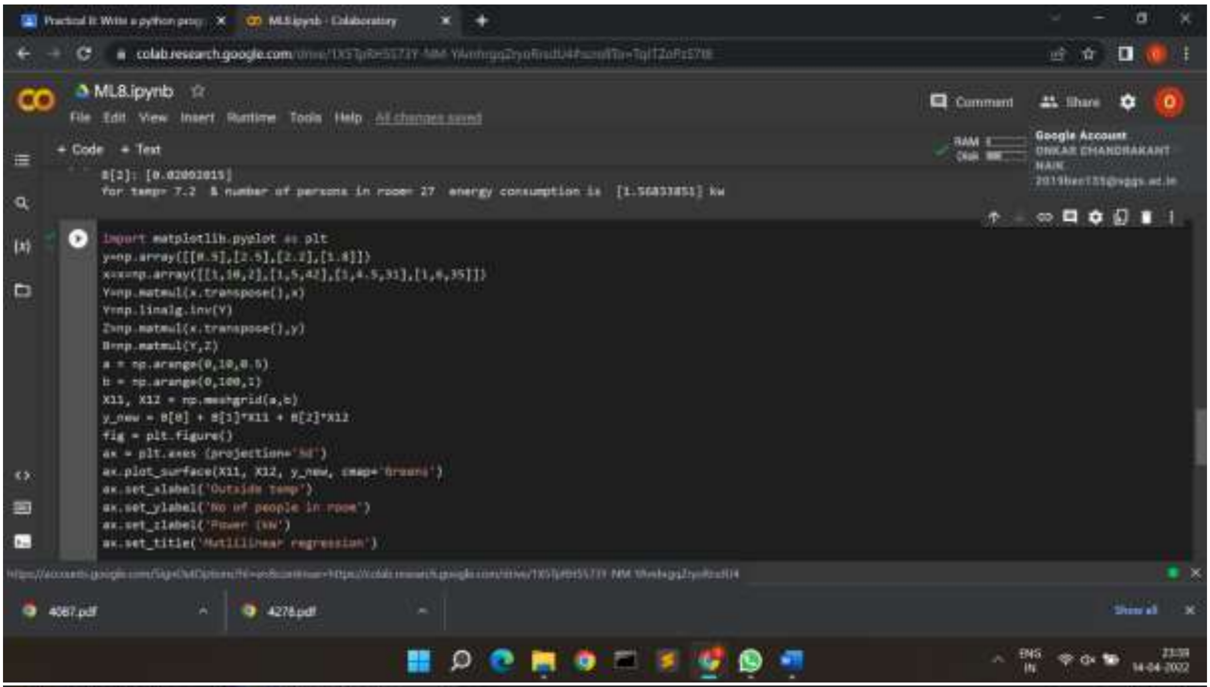
II. temp = 3.5 and No. of persons in room = 45

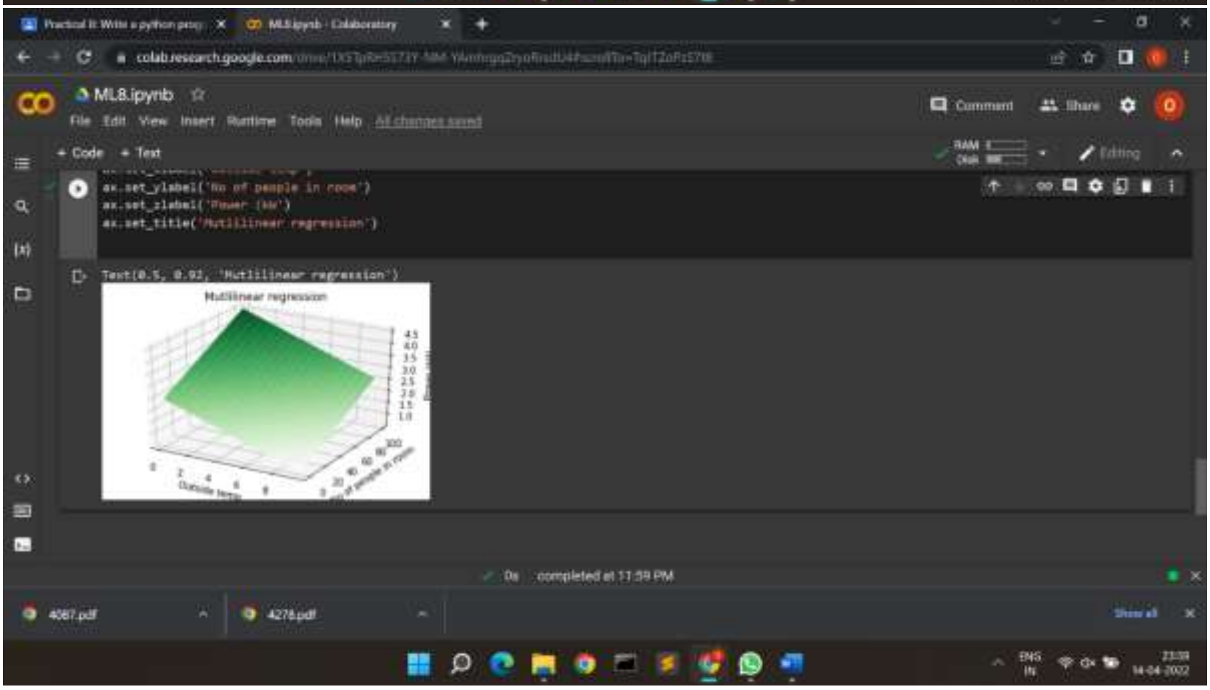


III. temp = 7.2 and No. of persons in room = 27



3.Plot the 3D graph for power consumption data using multivariate regression coefficients. (fit the hyperplane)





CONCLUSION :- Multivariate Regression is a supervised machine learning algorithm involving multiple data variables for analysis. Multivariate regression is an extension of multiple regression with one dependent variable and multiple independent variables. Based on the number of independent variables,we try to predict the output.

Practical-9

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM :- Write a python program for Linear Regression using Gradient Descentalgorithm.

REQUIRED SOFTWARE :- Google

Colab , PychamTHEORY:-

Gradient descent is a name for a generic class of computer algorithms which

minimize a function. These algorithms achieve this end by starting with initial parameter values and iteratively moving towards a set of parameter values that minimize some cost function or metric–that's the descent part. The movement toward best-fit is achieved by taking the derivative of the variable or variables involved, towards the direction with the lowest (calculus-defined) gradient–that's thegradient part.

Gradient descent is an important concept in computer science, and an illustrative example of why CS has kind of overtaken statistics in importance when it comes tomachine learning: it's a general-purpose tool that can be used to "brute force" an optimal solution in a wide range of scenarios, which doesn't have the elegance, closed-form solution, and unfortunate sheer mathematical inpalatability of a statistical solution.

Ordinary linear regression is a good and simple way of demonstrating how gradient descent works. We start with some error function. We could use any metric we want,but in OLS the obvious one is the residual sum of squares.

Given a sequence of points, yiyi, and a sequence of points predicted by ourmodel, y^iy^i, RSS is:

error(m,b)=∑i=1n(yi−yi^)

Our objective is to minimize this value. Inserting our linear regression model in forthe y^iy^i predictions, and assuming (for the sake of simplicty) that we're doing regression on only one variable, we get:

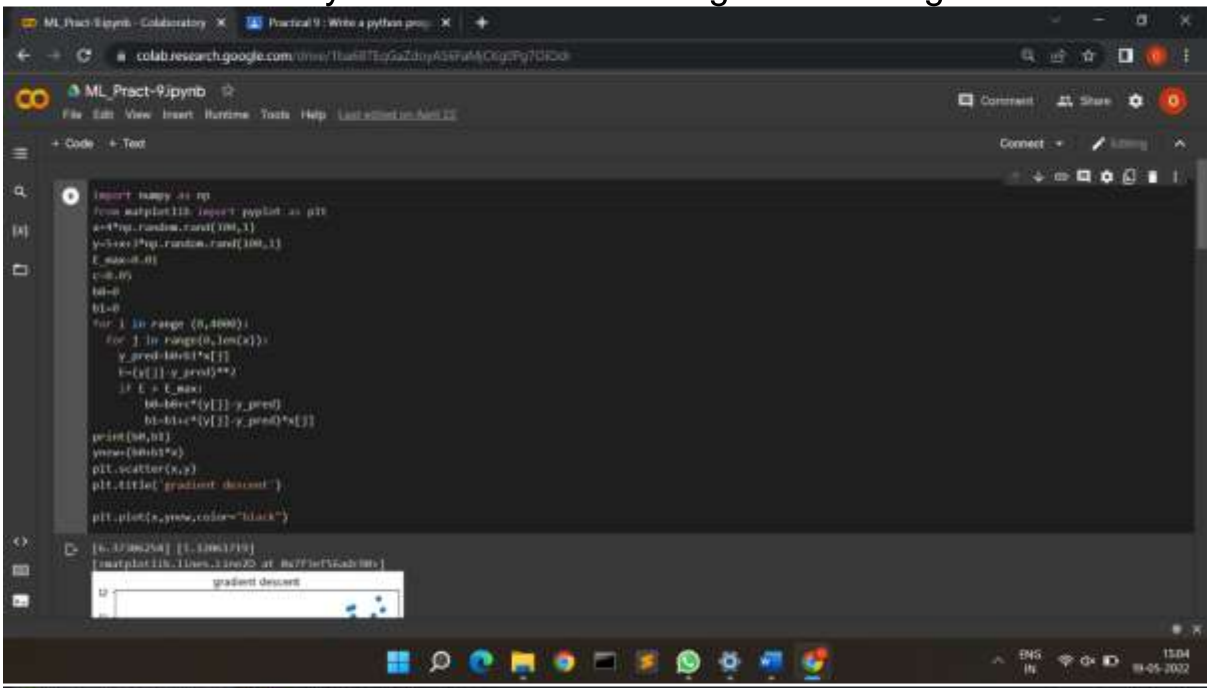
RSS=∑i=1n(yi−(mxi+b))2RSS=∑i=1n(yi−

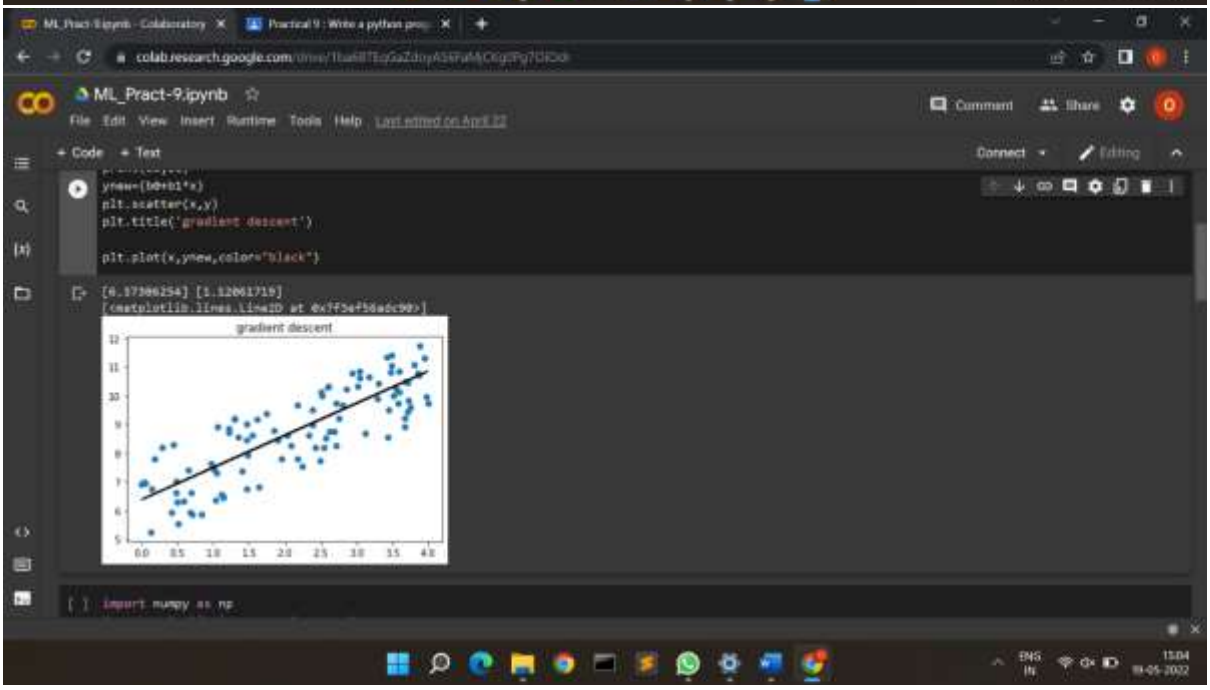
(mxi+b))2Where bb is the intercept and mm is the slope
of the line of best fit.

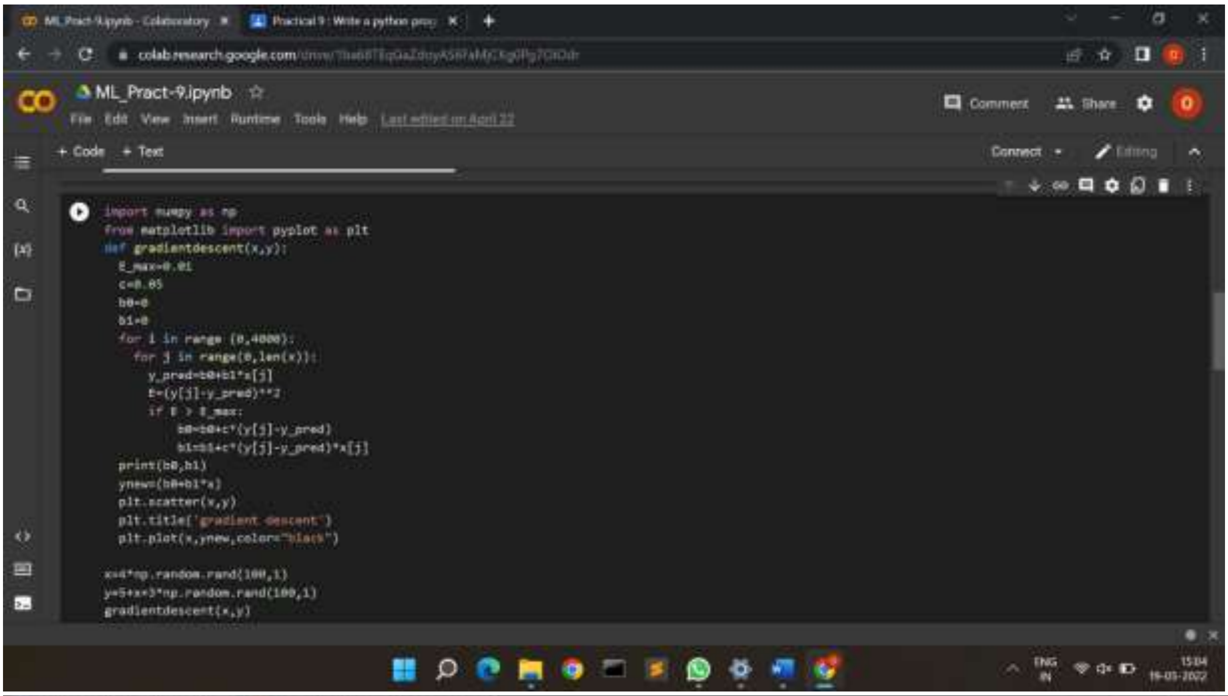
Now we need to take the gradient. Since this is an equation of two variables
(bb and mm) the gradient will consist of two partial derivatives. Hence the gradient is:

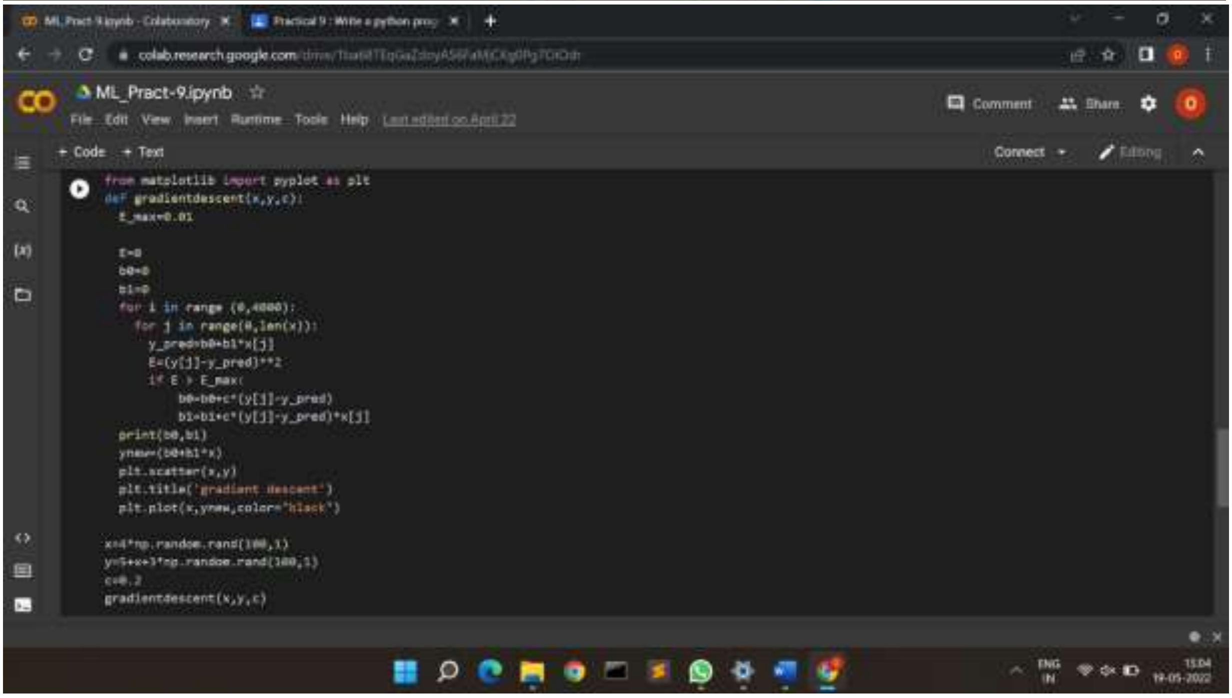
⟨∂∂b(RSS),∂∂m(RSS)⟩=⟨−2∑i=1n(yi−(mxi+b)),−2∑i=1nxi(yi−(mxi+
b))⟩⟨∂∂b(RSS),∂∂m(
RSS)⟩=⟨−2∑i=1n(yi−(mxi+b)),−2∑i=1nxi(yi−(mxi+b)
)⟩

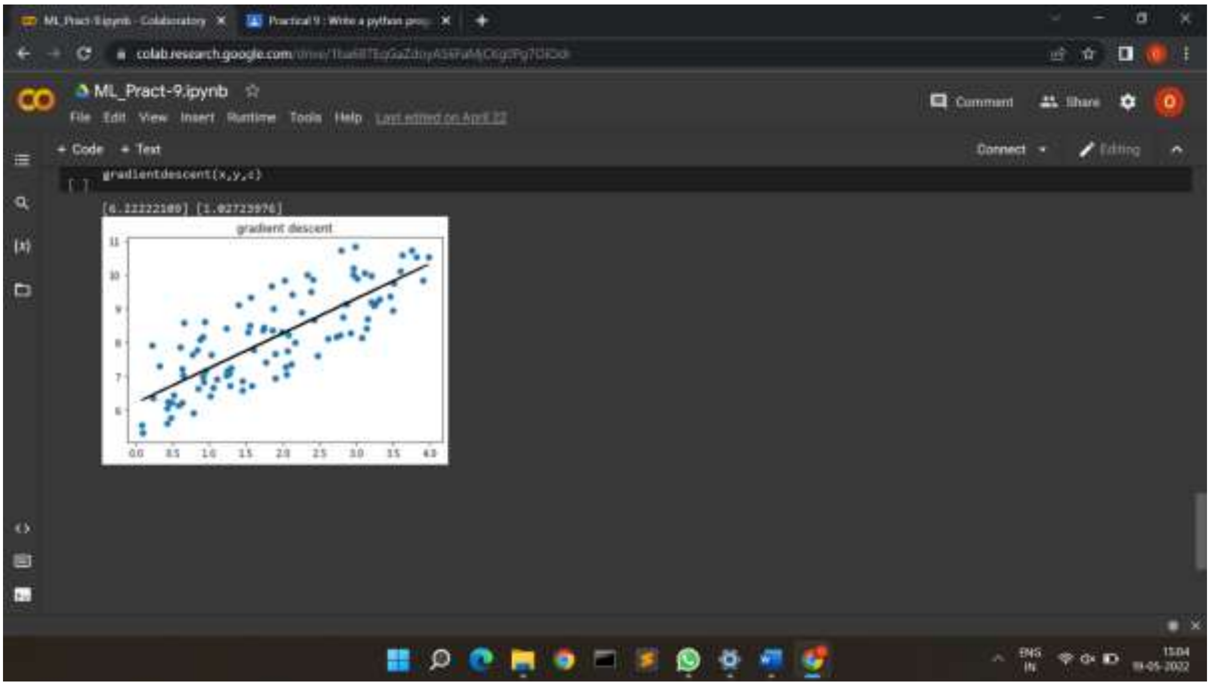
To solve, take a step in the negative gradient direction every iteration. Eventually wewill have something that converges.






ML_Pract-9.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
Editing
Comment Share
import numpy as np
from matplotlib import pyplot as plt
def gradientdescent(x,y):
  E_max=0.01
  c=0.05
  b0=0
  b1=0
  for i in range (0,4000):
    for j in range(0,len(x)):
      y_pred=b0+b1*x[j]
      E=(y[j]-y_pred)**2
      if E > E_max:
        b0=b0+c*(y[j]-y_pred)
        b1=b1+c*(y[j]-y_pred)*x[j]
  print(b0,b1)
  ynew=(b0+b1*x)
  plt.scatter(x,y)
  plt.title('gradient descent')
  plt.plot(x,ynew,color="black")
x=4*np.random.rand(100,1)
y=5+x+3*np.random.rand(100,1)
gradientdescent(x,y)


ML_Pract-9.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
Editing
Comment Share
from matplotlib import pyplot as plt
def gradientdescent(x,y,c):
  E_max=0.01
  E=0
  b0=0
  b1=0
  for i in range (0,4000):
    for j in range(0,len(x)):
      y_pred=b0+b1*x[j]
      E=(y[j]-y_pred)**2
      if E > E_max:
        b0=b0+c*(y[j]-y_pred)
        b1=b1+c*(y[j]-y_pred)*x[j]
  print(b0,b1)
  ynew=(b0+b1*x)
  plt.scatter(x,y)
  plt.title('gradient descent')
  plt.plot(x,ynew,color="black")
x=4*np.random.rand(100,1)
y=5+x+3*np.random.rand(100,1)
c=0.2
gradientdescent(x,y,c)



CONCLUSION :- The goal of the gradient descent algorithm is to minimize the givenfunction (say cost function). To achieve this goal, it performs two steps iteratively is to Compute the gradient (slope), the first order derivative of the function at that pointand to Make a step (move) in the direction opposite to the gradient, opposite direction of slope increase from the current point by alpha times the gradient at that point

Practical-10

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

Roll No : A65

AIM : - Write a python program to implement Parzen Window Density estimation for any randomly generated signal.

REQUIRED SOFTWARE:- Google

colab ,

HEORY :-

Parzen windows classification is a technique for nonparametric density estimation, which can also be used for classification. Using a given kernel function, the technique approximates a given training set distribution via a linear combination ofkernels centered on the observed points. In this work, we separately approximate densities for each of the two classes, and we assign a test point to the class with maximal posterior probability.

The resulting algorithm is extremely simple and closely related to support vectormachines. The decision function is

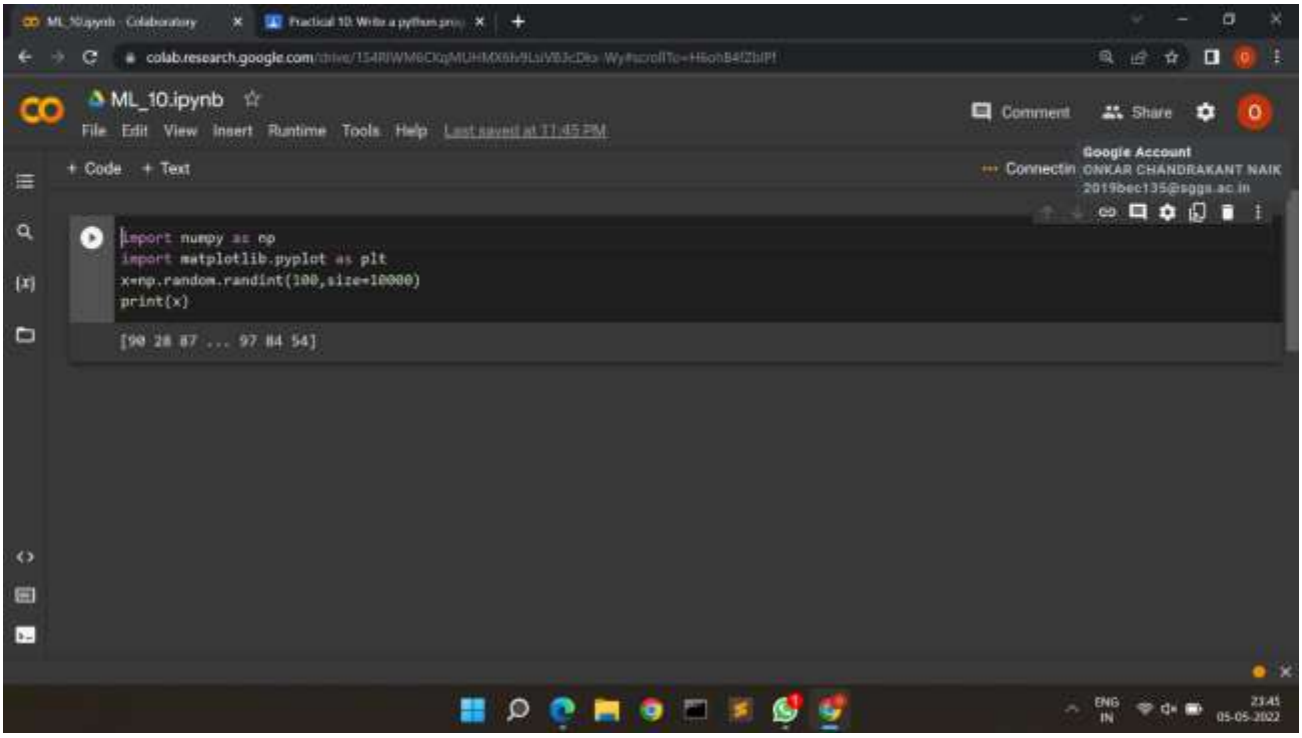
$$f(\mathbf{X}) = sign(\sum y_i K(\mathbf{X}_i, \mathbf{X})),$$

where the kernel function *K* is the radial basis function of Equation 2, without normalization applied to the inputs. As for the radial basis SVM, a constant is addedto the kernel function whenever the two inputs are identical (Equation 3).

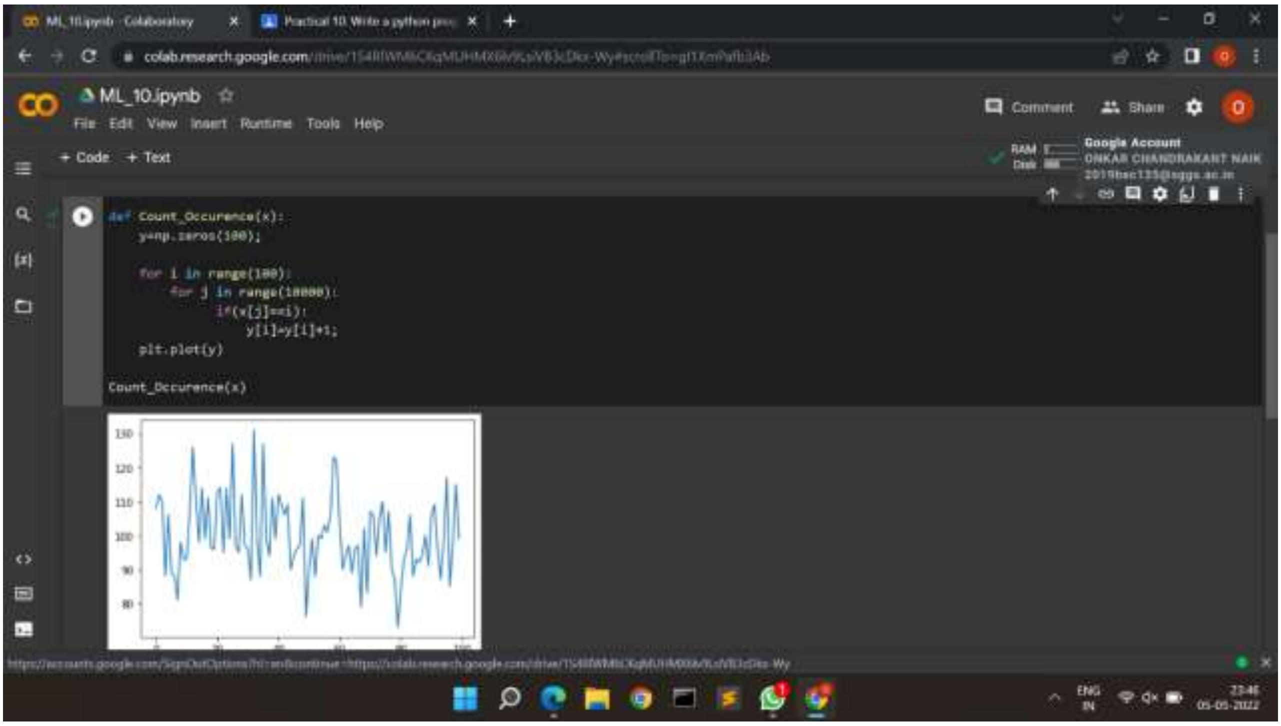
The Parzen windows classification algorithm does not require any training phase; however, the lack of sparseness makes the test phase quite slow. Furthermore, although asymptotical convergence guarantees on the perfomance of Parzen windows classifiers exist [Duda and Hart, 1973], no such guarantees exist for finitesample sizes.

Parzen windows can be regarded as a generalization of *k*-nearest neighbor techniques. Rather than choosing the *k* nearest neighbors of a test point and labelling the test point with the weighted majority of its neighbors' votes, one can consider all points in the voting scheme and assign their weight by means of the kernel function. With Gaussian kernels, the weight decreases exponentially with thesquare of the distance, so far away points are practically irrelevant. The width σ of the Guassian determines the relative weighting of near and far points. Tuning this parameter controls the predictive power of the system. We have empirically optimized the value of σ.

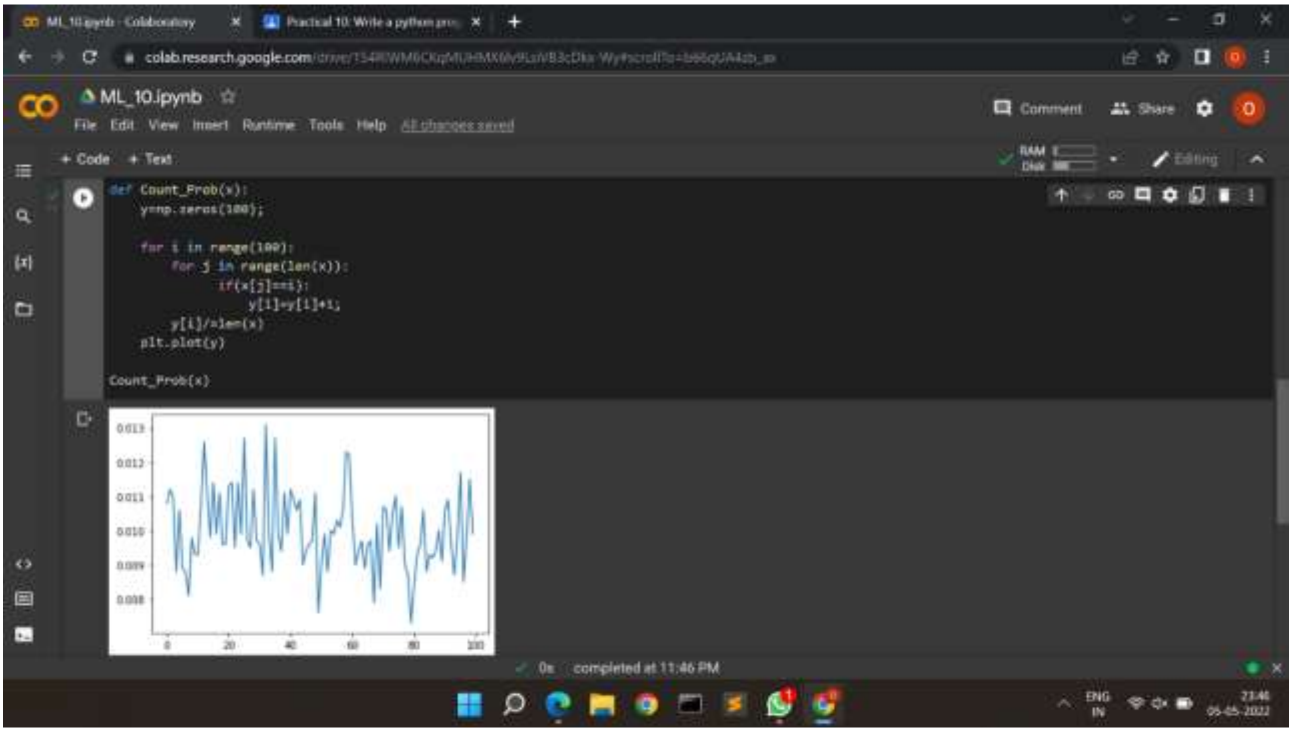
1.Generate a random signal of size 10,000 in the range 0 to 99.



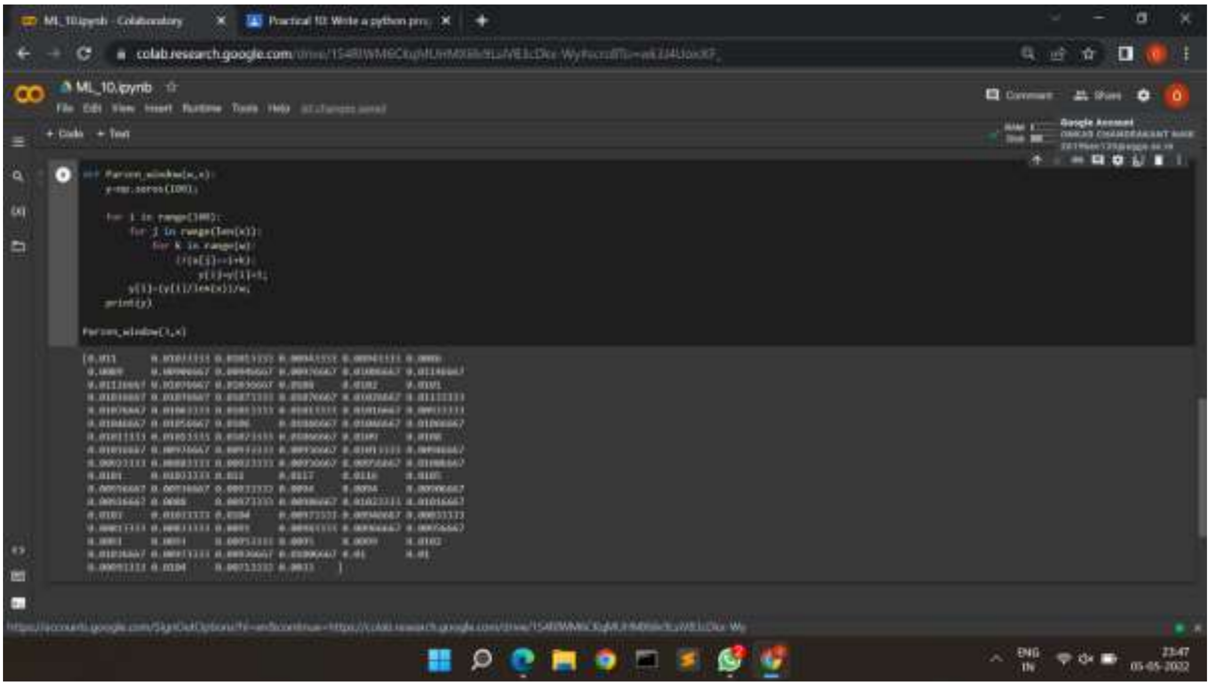
2.Count the occurrence of each sample from signal. Plot the occurrence of each sample.



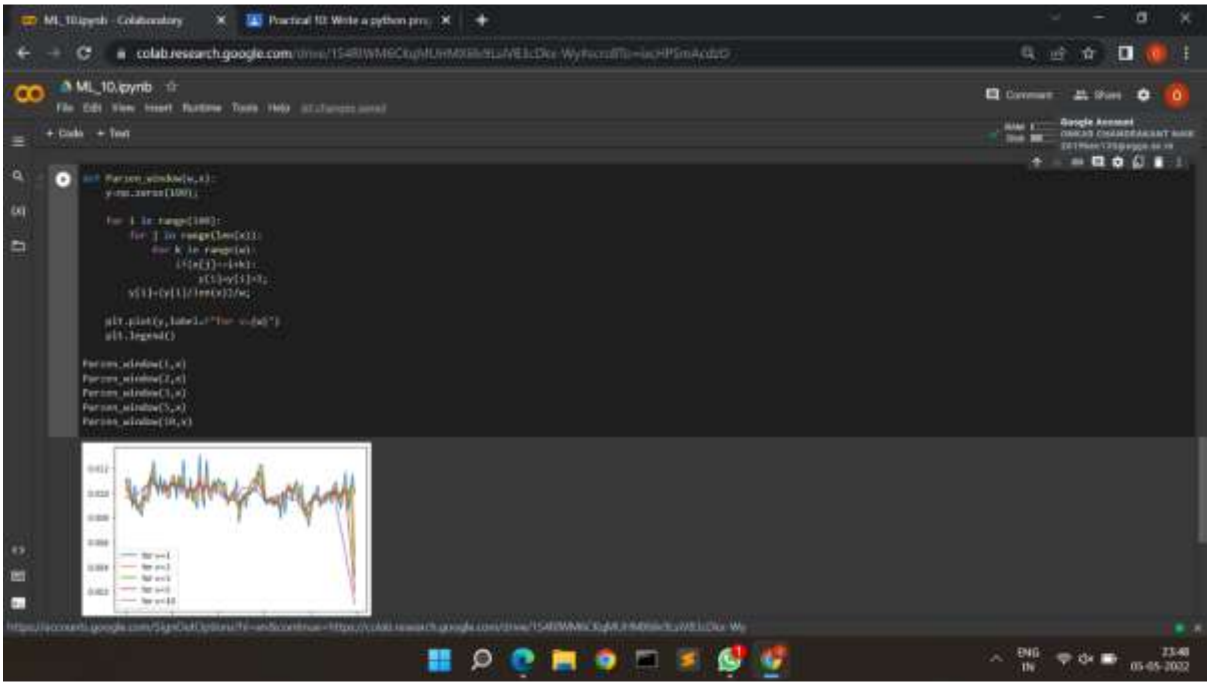
3.Plot the probability of each sample.



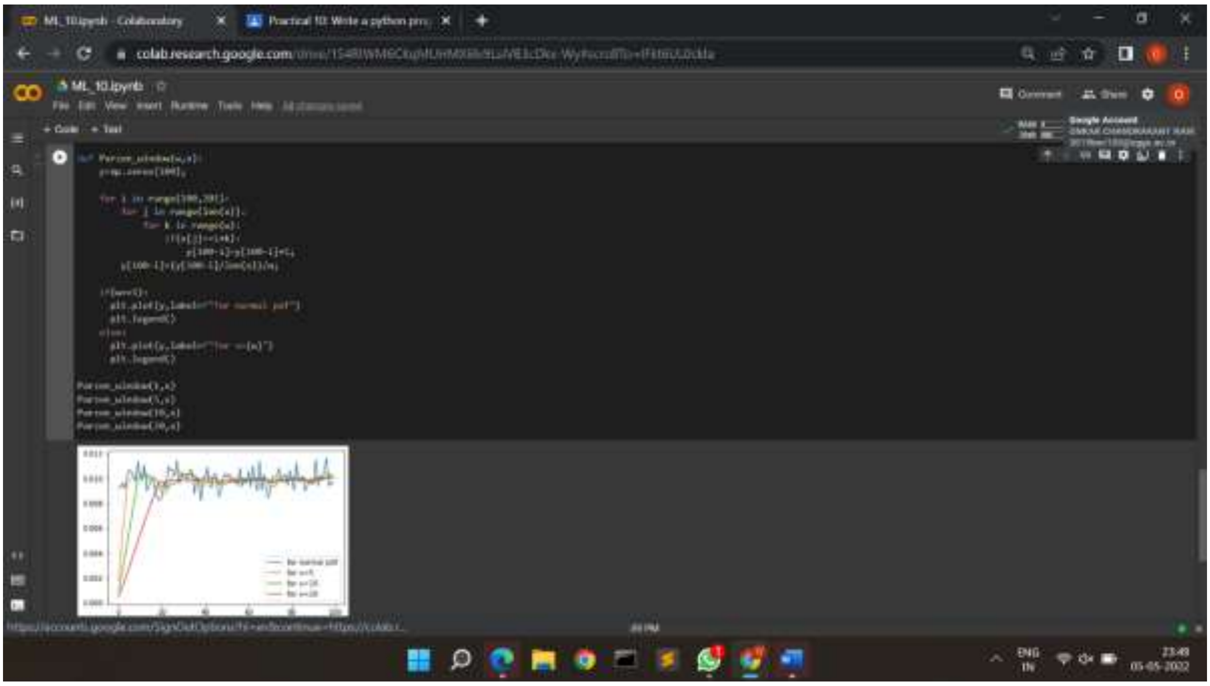
4.Define the function to compute the Parzen window density using formula. (pdf = k/n/V , where k-> no. of samples fall within window, n-> total samples, V->width of window)



5.Plot the normal pdf of signal and Parzen window density function of signal for various window size (V=1,2,3,5,10).



6.Generate a random data of size 20,000 in the range 100 to 200. Plot the normal pdf of data and Parzen window density function of data for various window size (V=1,5,10,20).



CONCLUSION :- It is used to derive a density function f(X) is used to implement a Bayes Classifier. When we have a new sample feature f(x) and when there is a need to compute the value of the class conditional densities f(x) is used f(X) takes sample input data value and returns the density estimate of the given data sample.

Practical-11

Name: Naik Onkar Chandrakant

Reg No : 2019BEC135

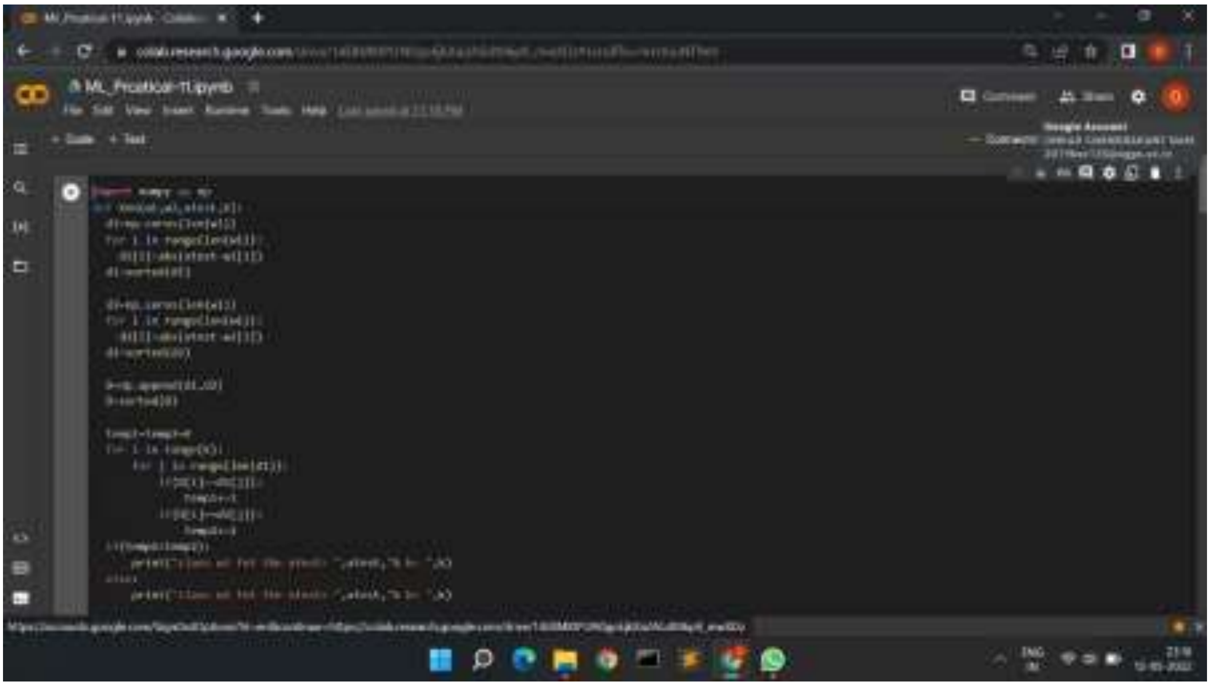
Roll No : A65

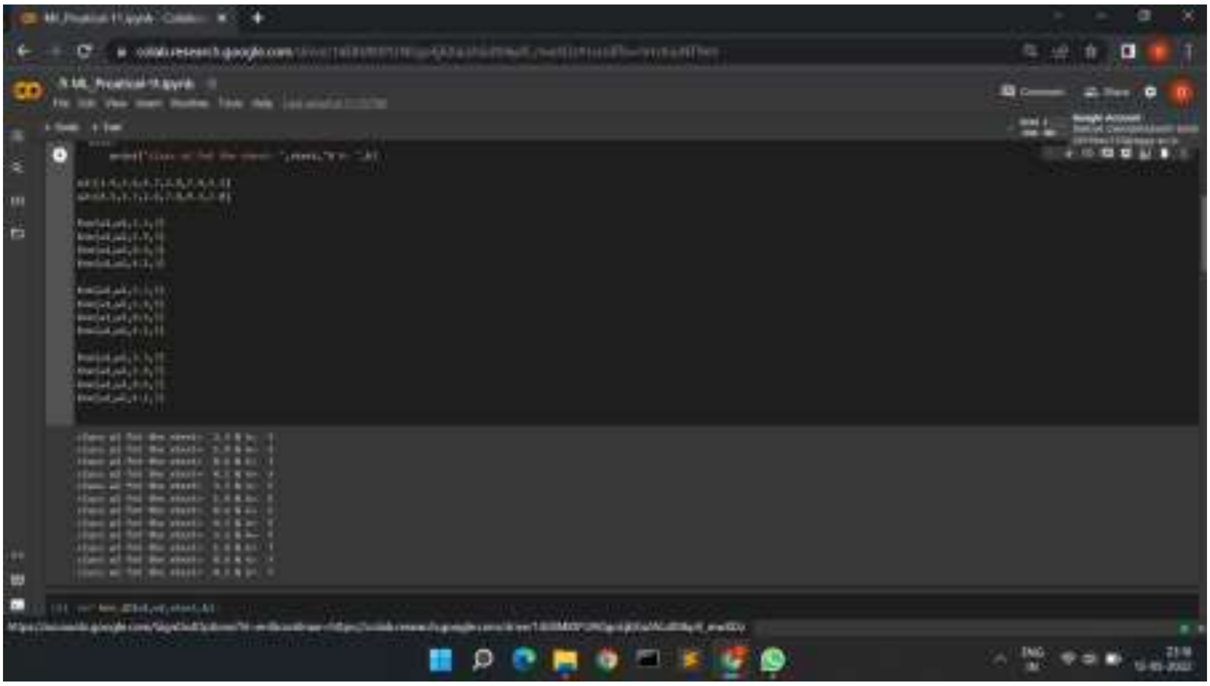
AIM : - Write a python program for implementation of K-nearest neighboralgorithm.

REQUIRED SOFTWARE:- Google colab , PycharmTHEORY :-

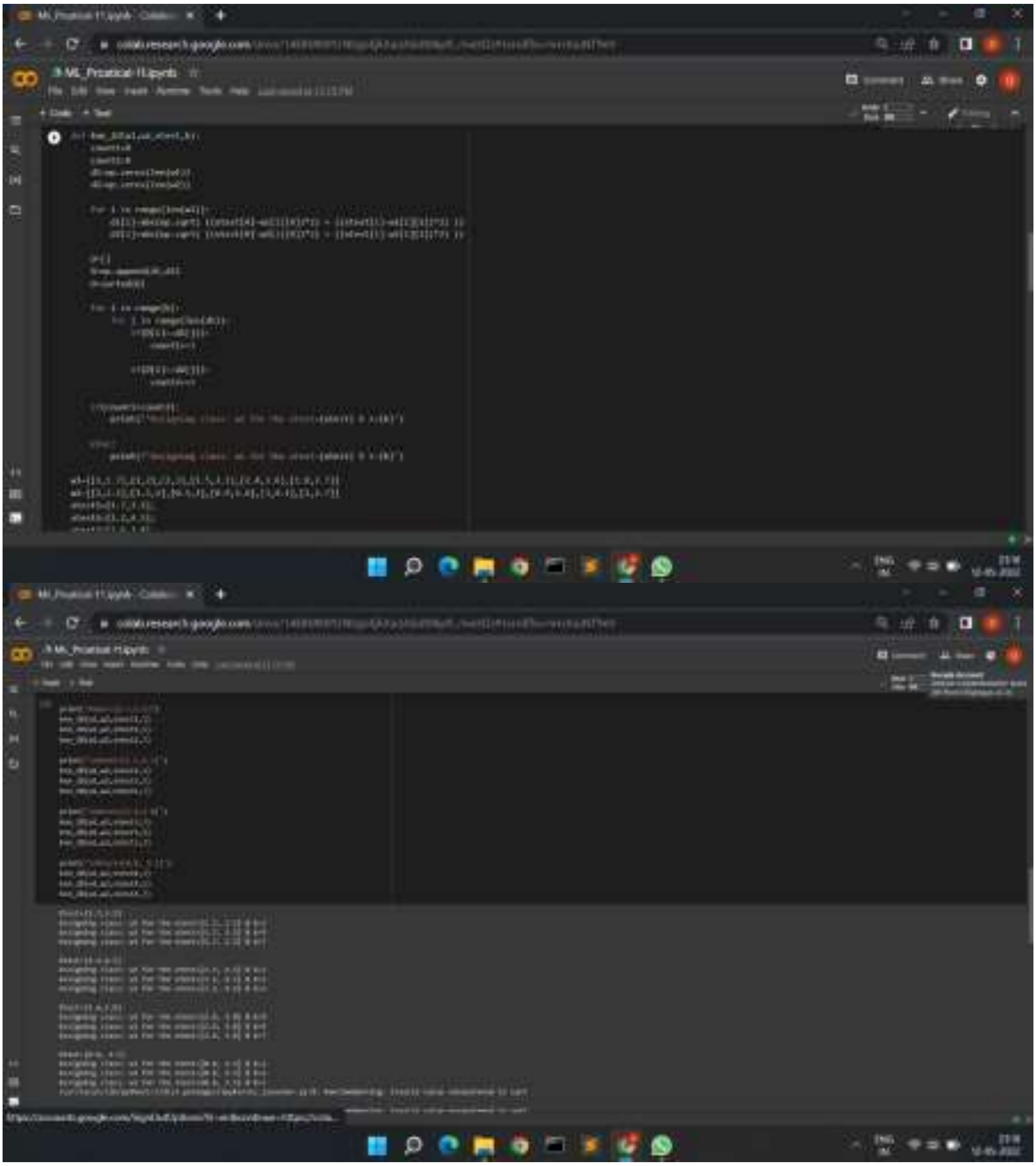
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms basedon Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

1. Classify the test sample for given 1D data using knn with k values as 3, 5 and 7. w1= [1.4,3.6,4.7,2.8,7.4,5.3,6.7,3.2,4.8], w2=[4.5,3.7,2.4,7.8,9.3,7.0,6.4,8.3,5.1], x_test= 3.3, 1.9, 8.6 & 4.1

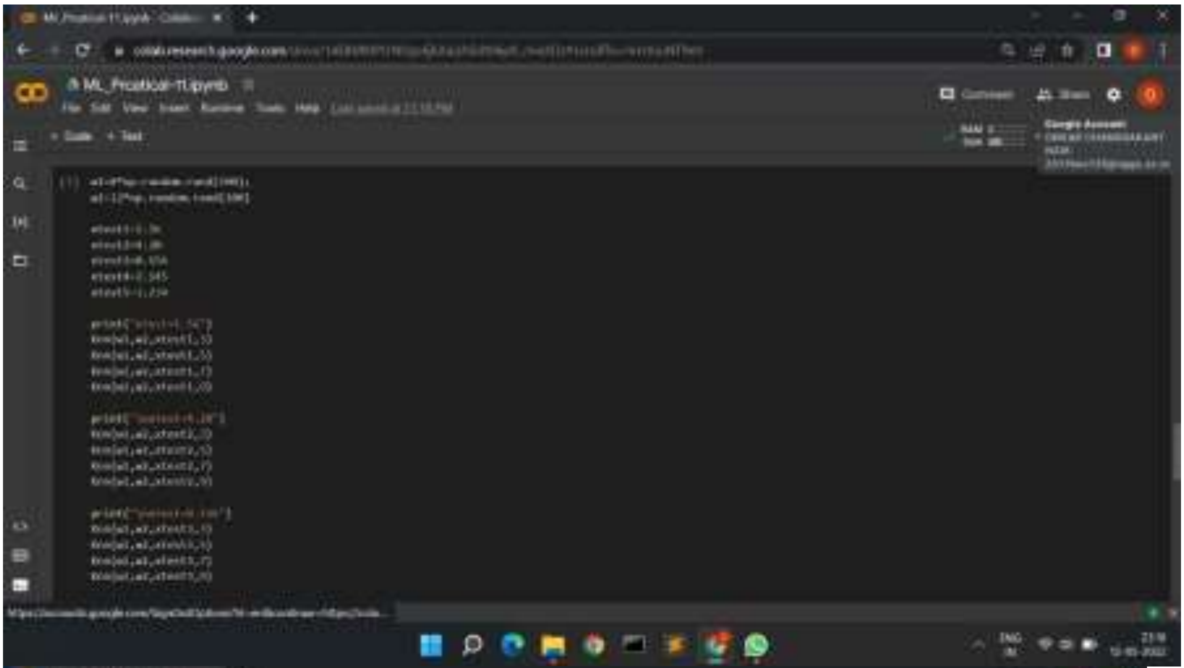


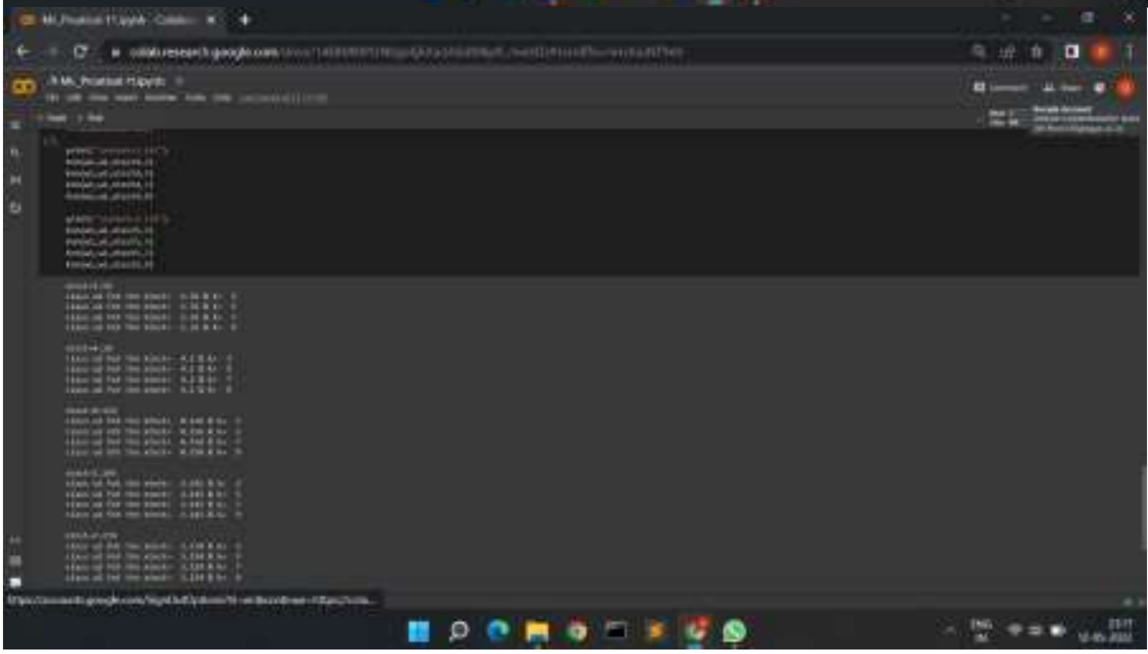


2. Classify the test sample for given 2D data using knn with k values as 3, 5 and 7.w1={[1,1.7],[1,2],[2,3],[1.5,3.1],[2.4,3.6],[1.8,2.7]}, w2={[1,1.5],[1.5,2],[0.5,1],[0.9,1.6],[3,4.1],[2,2.7]}, x_test= [1.7,3.2],[1.2,4.5],[2.6,3.8] & [0.8, 2.5]



3.Generate the random 1D data consisting of 100 samples in any desired range for two class. Vary the ‘k’ value [3,5,7,9] and find the class for 5 random test sample.





CONCLUSION :- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to theavailable categories. K-NN algorithm stores all the available data and classifies a newdata point based on the similarity. This means when new data appears then it can beeasily classified into a well suite category by using K- NN algorithm.

Practical-12

Naik Onkar Chandrakant

A65

2019BEC135

AIM :- Write a python program to partition the data into two groups using K-means clustering algorithm.

REQUIRED SOFTWARE :- Google colab ,

PycharmTHEORY :-

K-Means Clustering is an unsupervised learning algorithm that is used to solve theclustering problems in machine learning or data science.

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters,and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. Themain aim of this algorithm is to minimize the sum of distances between the data pointand their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-numberof clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

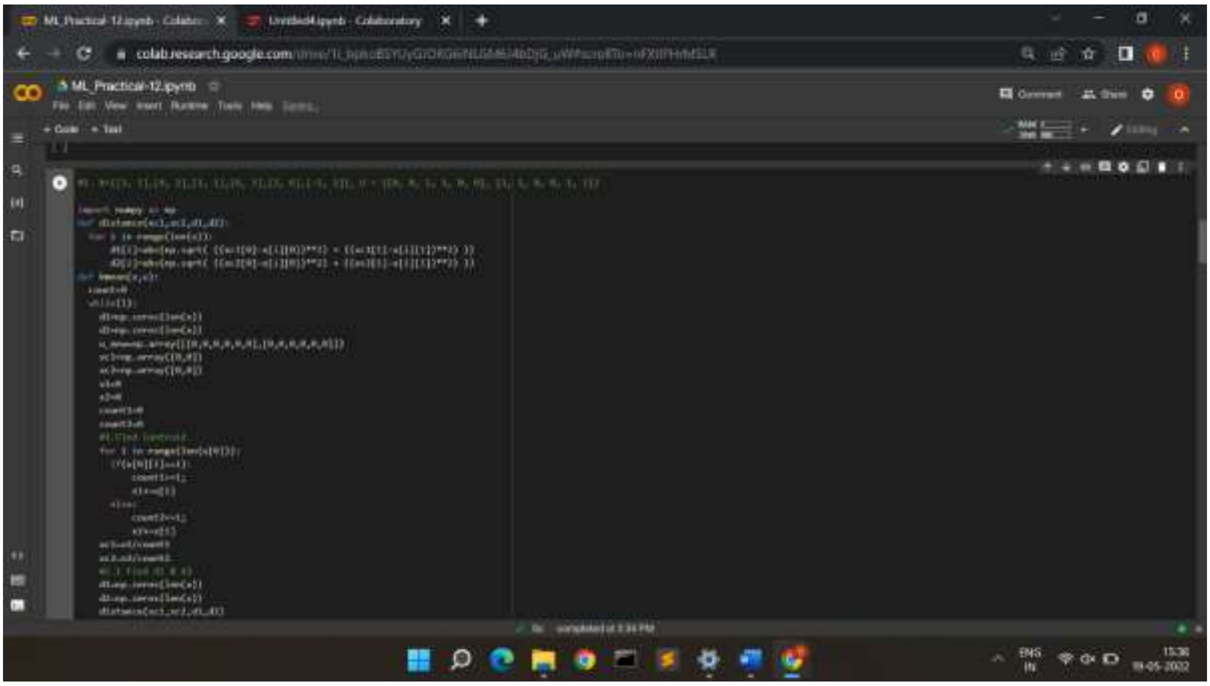
The k-means clustering algorithm mainly performs two tasks:

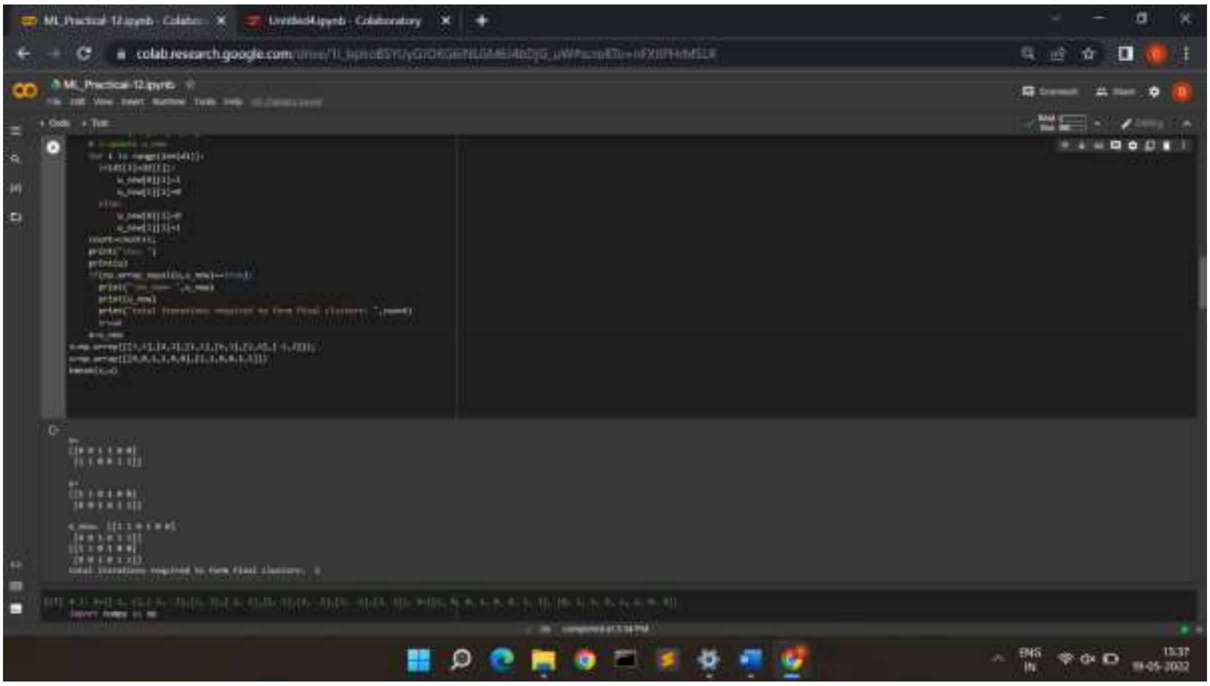
- Determines the best value for K center points or centroids by an iterativeprocess.

- Assigns each data point to its closest k-center. Those data points which arenear to the particular k-center, create a cluster.

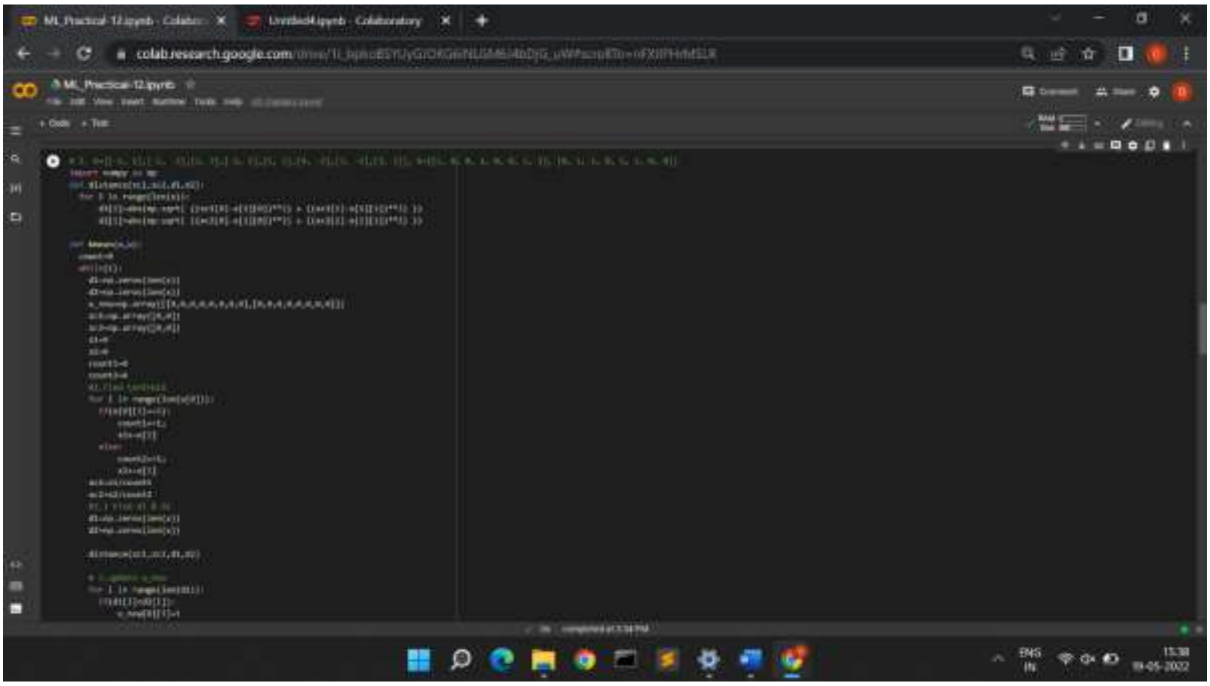
Hence each cluster has datapoints with some commonalities, and it is away from otherclusters.

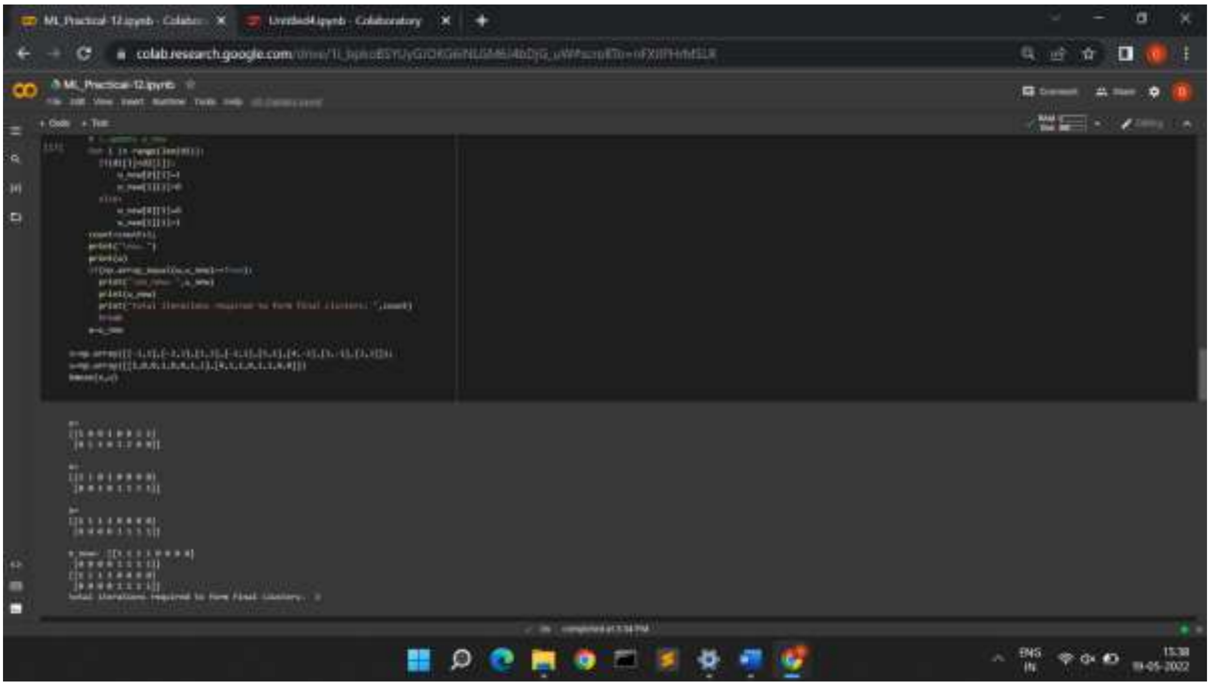
1.X={[3, 1],[4, 2],[1, 1],[6, 3],[2, 6],[-1, 2]}, U = {[0, 0, 1, 1, 0, 0], [1, 1, 0, 0, 1, 1]}



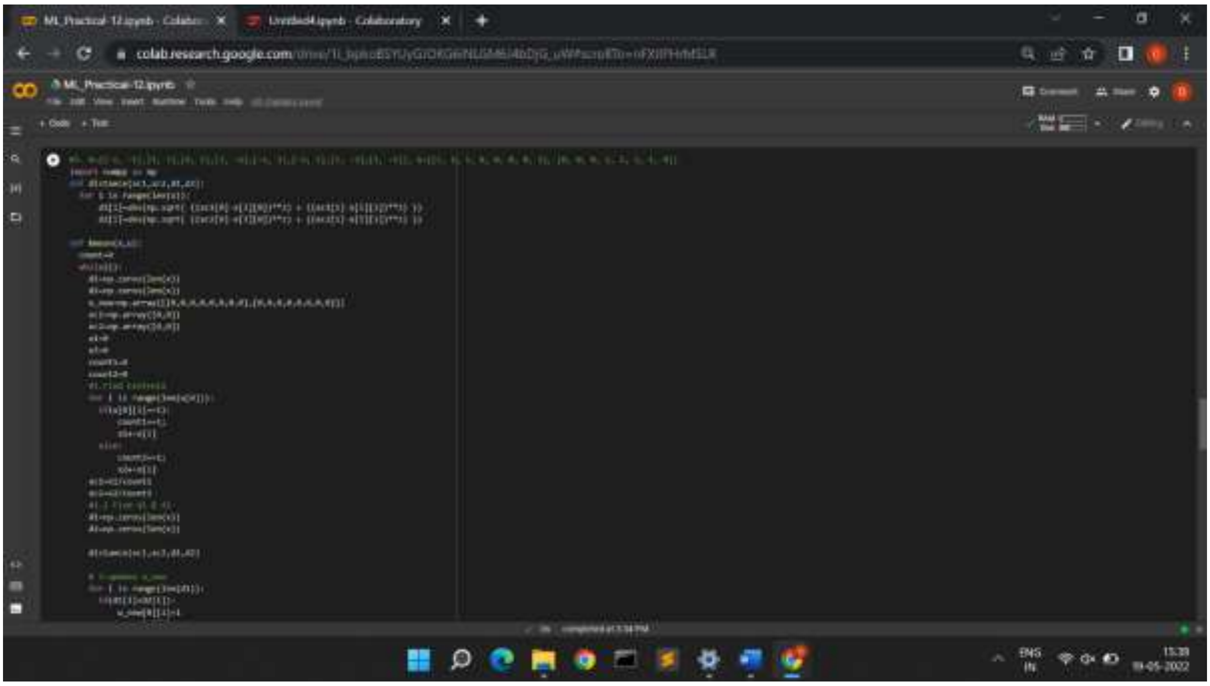


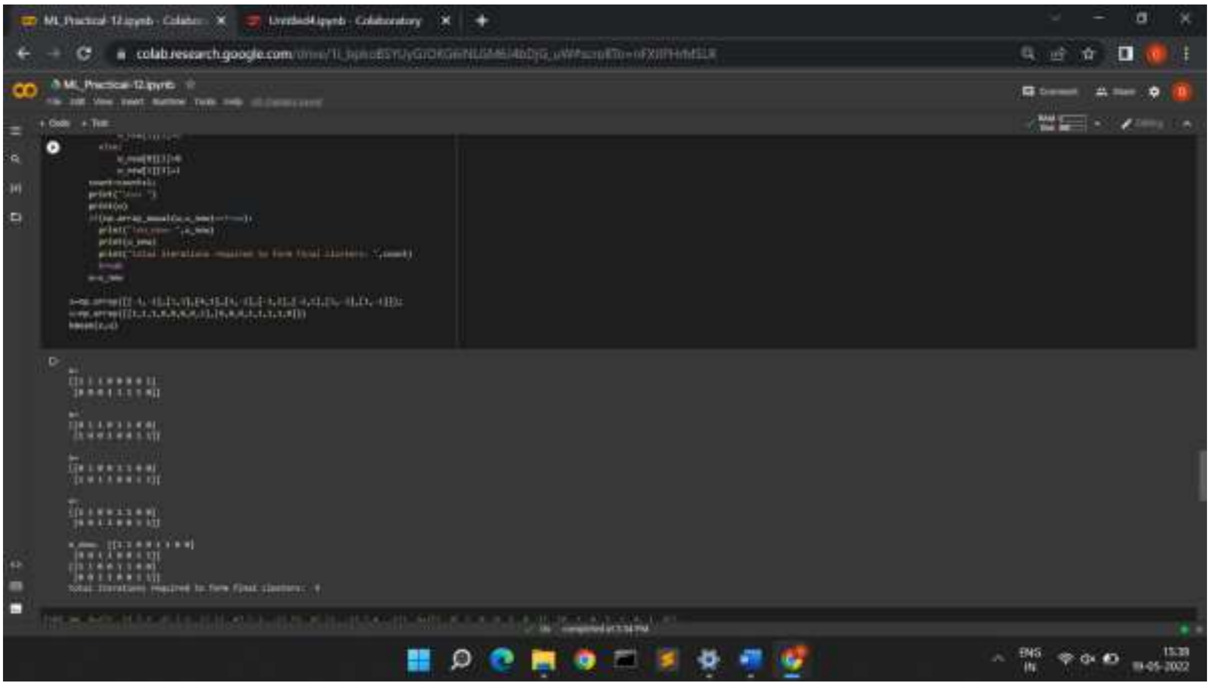
2.X={[-1, 1],[-2, -3],[1, 3],[-2, 1],[5, 1],[4, -2],[3, -1],[3, 3]}, U={[1, 0, 0, 1, 0, 0, 1, 1], [0, 1, 1, 0, 1, 1, 0, 0]}



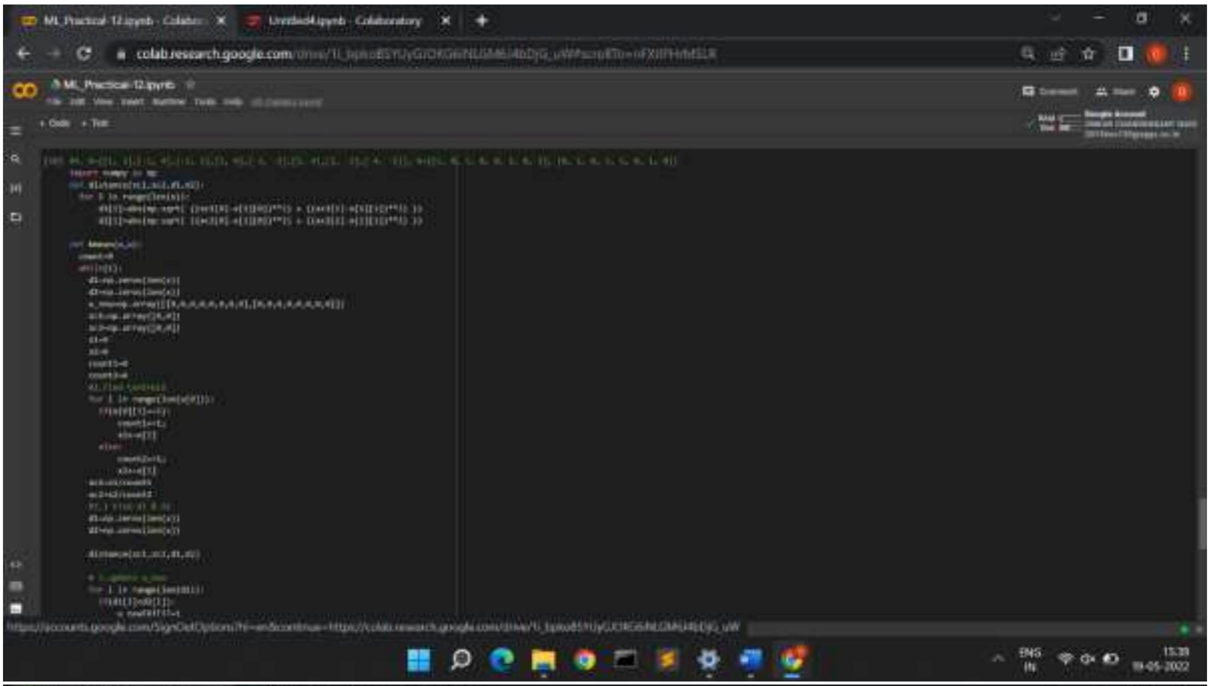


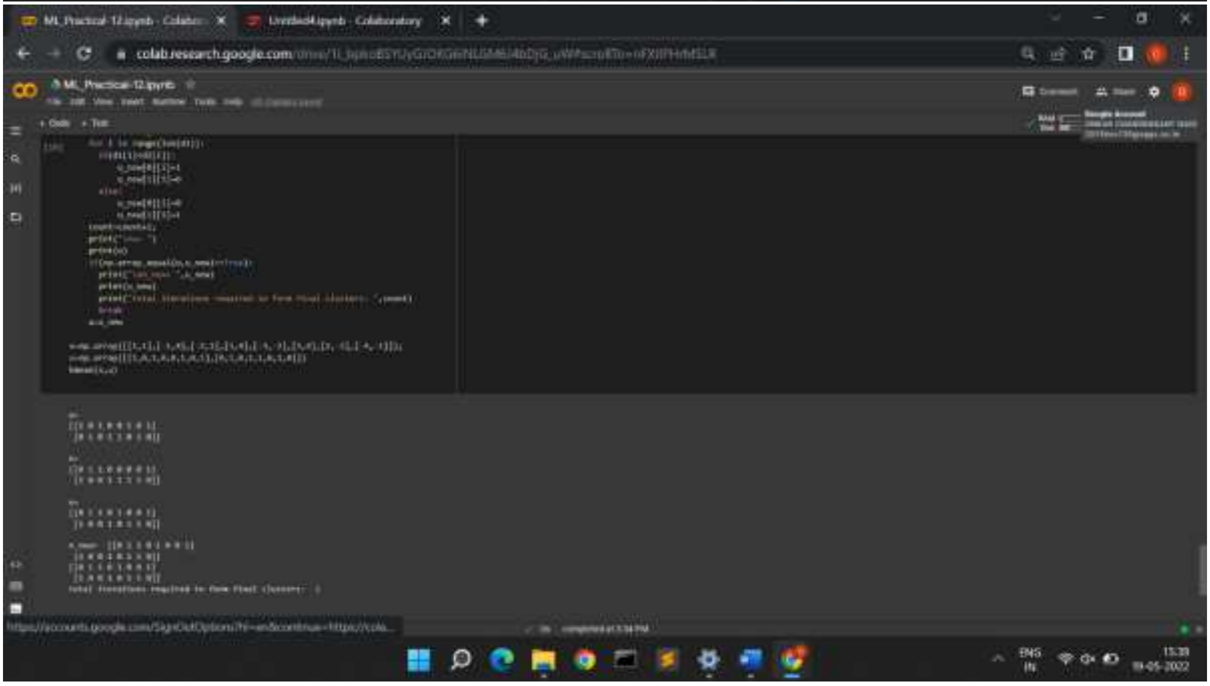
3.X={[-1, -1],[1, 3],[4, 1],[3, -2],[-1, 2],[-2, 1],[5, -3],[1, -1]}, U={[1, 1, 1, 0, 0, 0, 0, 1], [0, 0, 0, 1, 1, 1, 1, 0]}





4. X={[1, 1],[-1, 4],[-2, 1],[3, 4],[-3, -2],[5, 4],[2, -2],[-4, -1]}, U={[1, 0, 1, 0, 0, 1, 0, 1], [0, 1, 0, 1, 1, 0, 1, 0]}





CONCLUSION :- It allows us to cluster the data into different groups and a convenientway to discover the categories of groups in the unlabelled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.