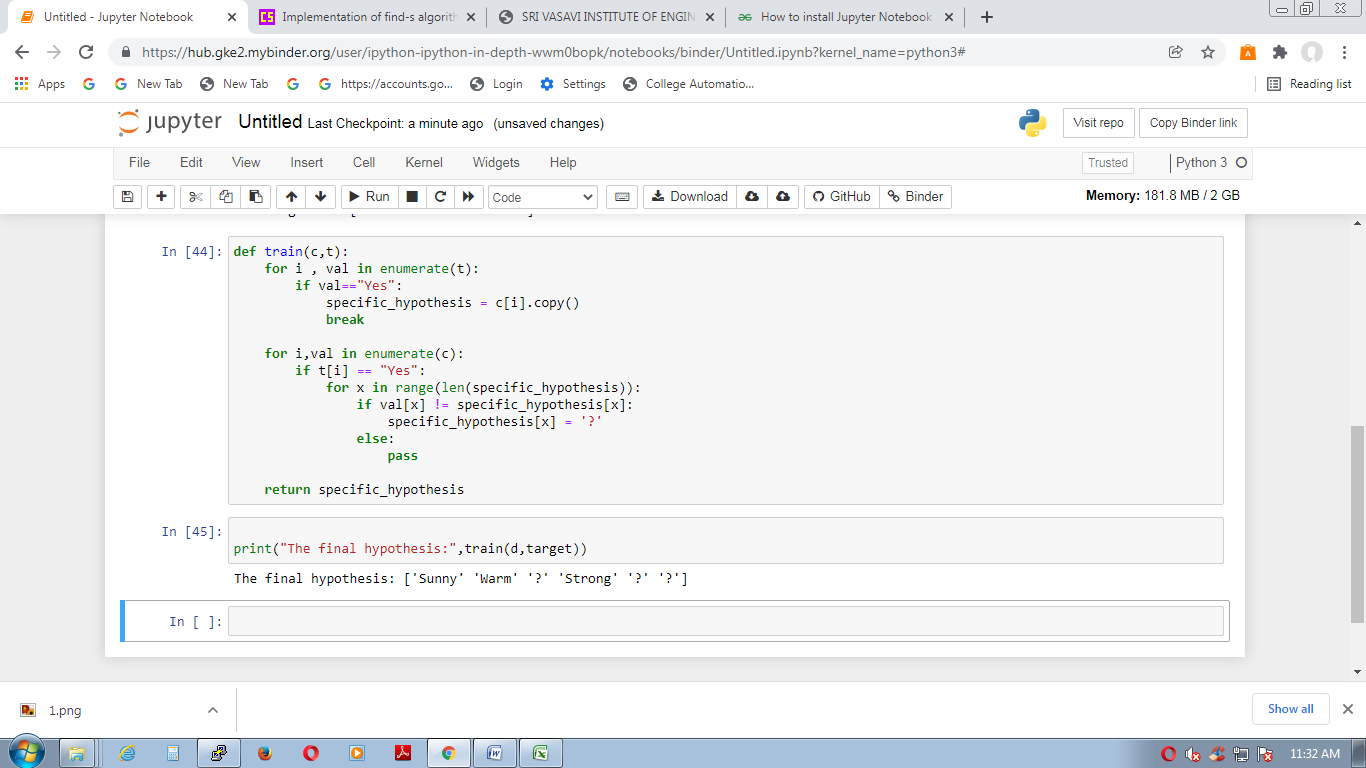
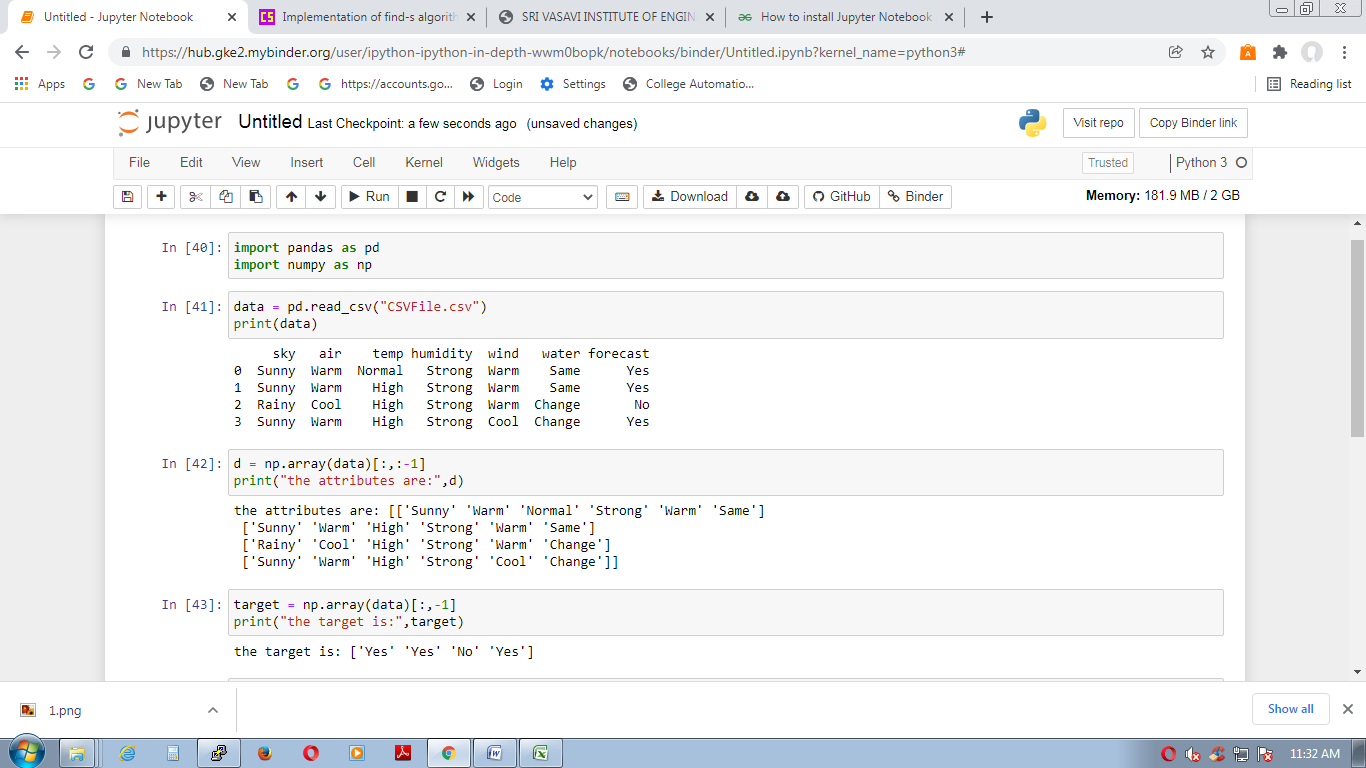
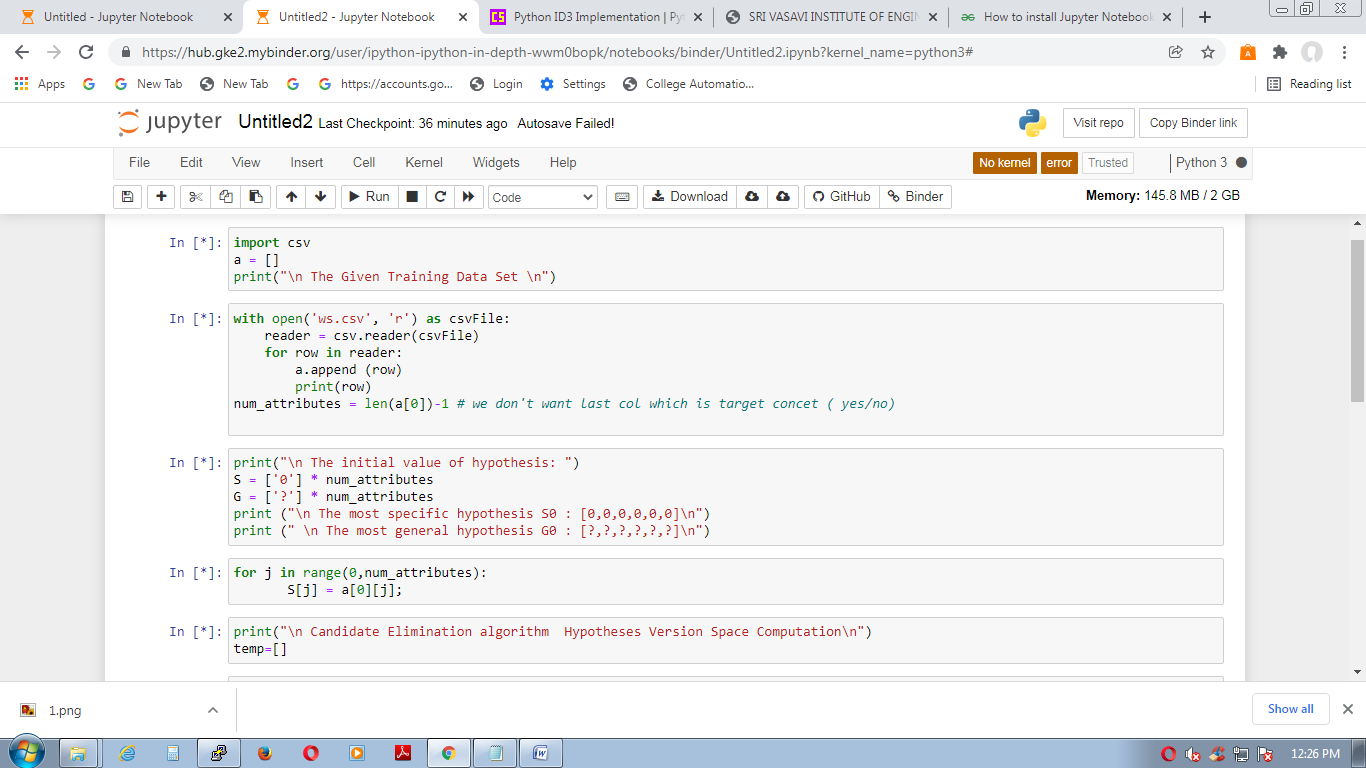
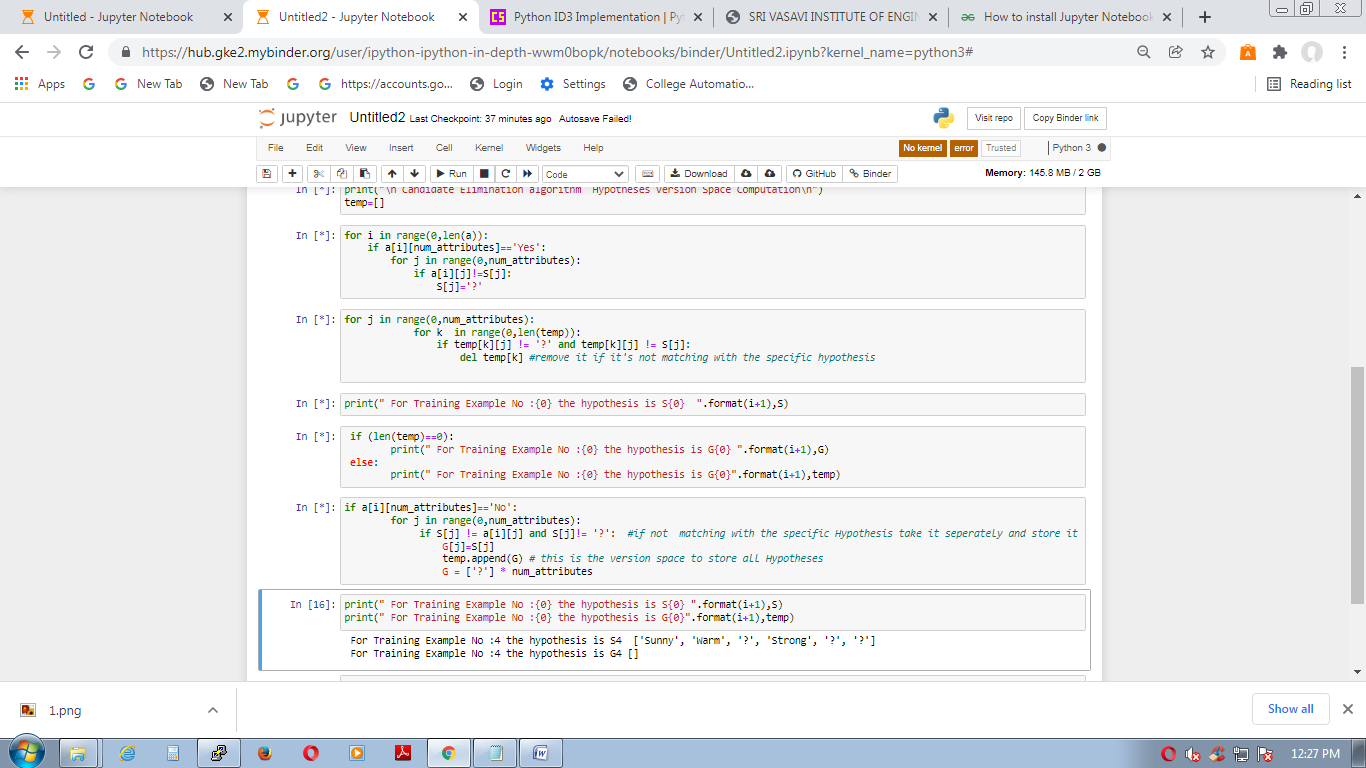
1.implement and demonstrate the **FIND-S** algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.



2. For a given set of training data examples stored in a .csv file, implement and demonstrate the

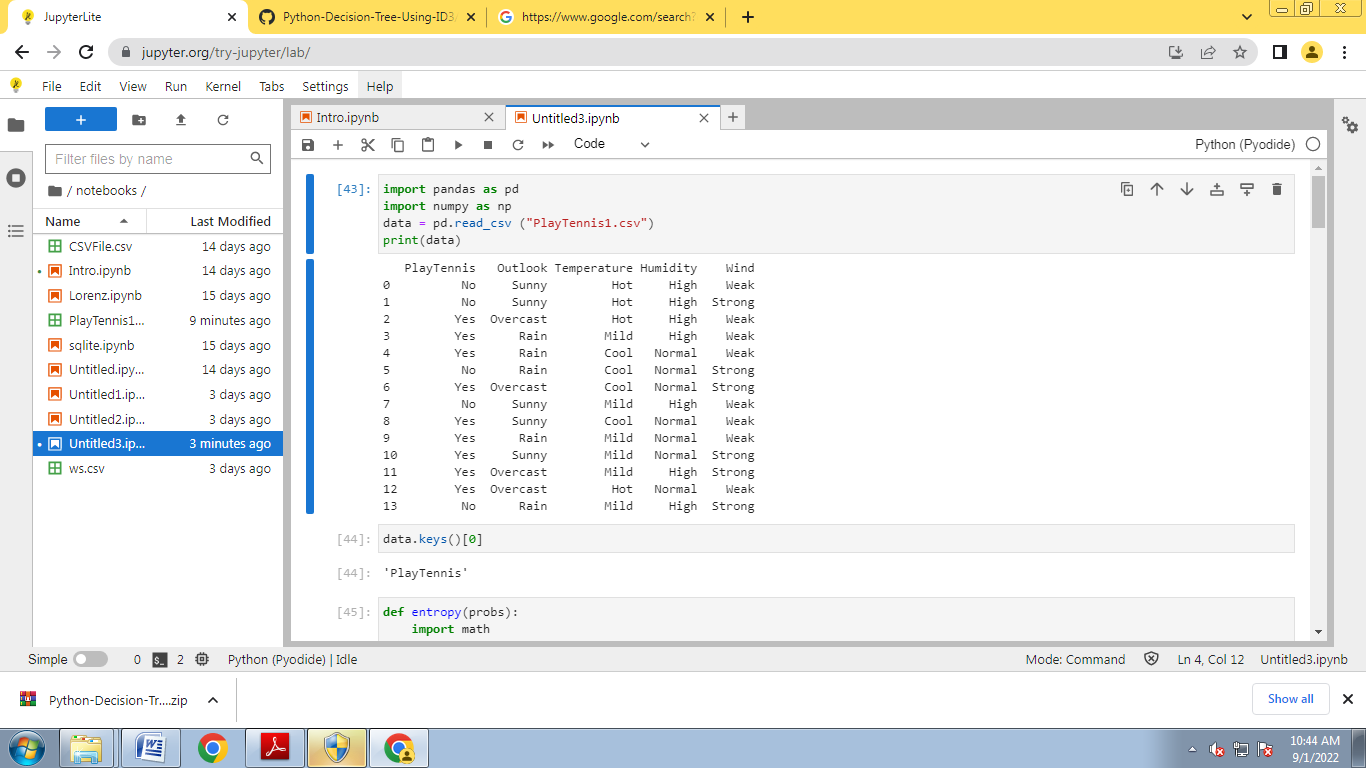
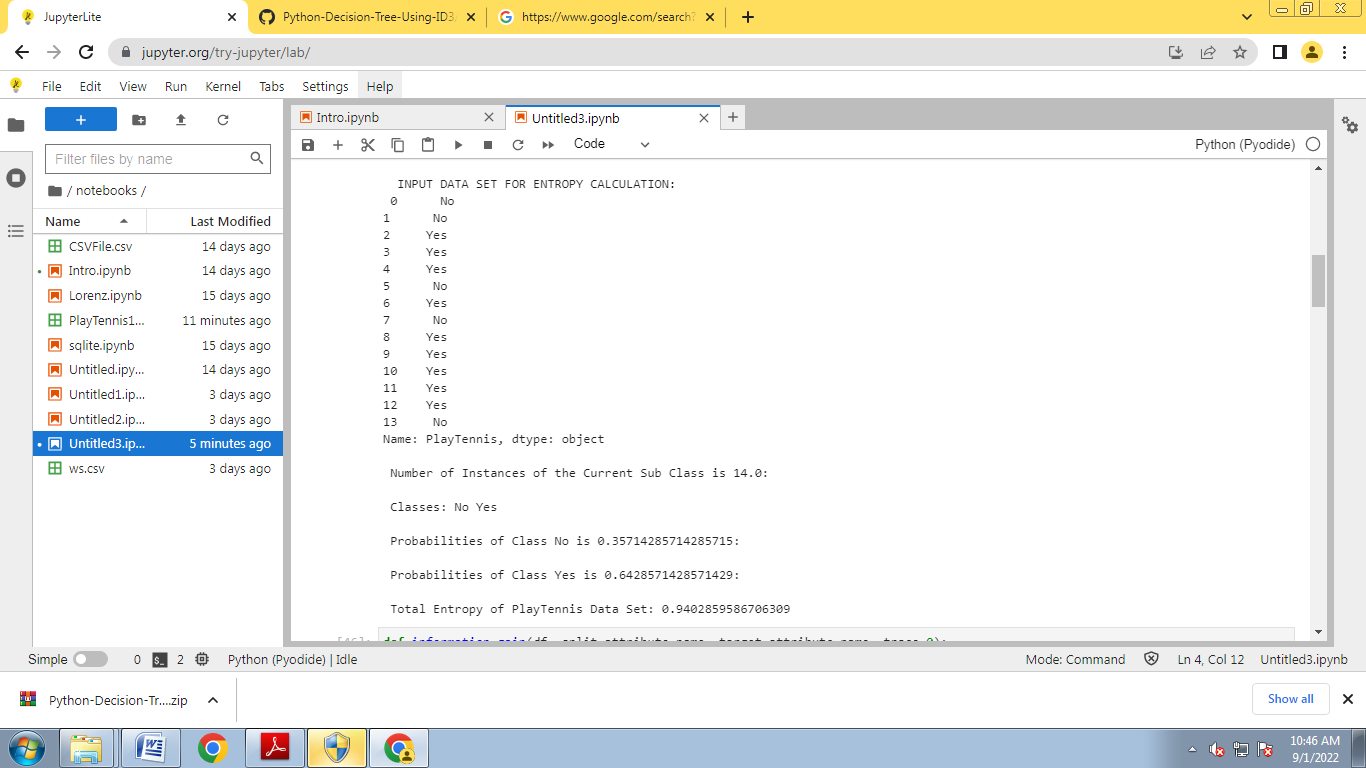
candidate elimination algorithm to output a description of the set of all hypotheses consistent with the training examples

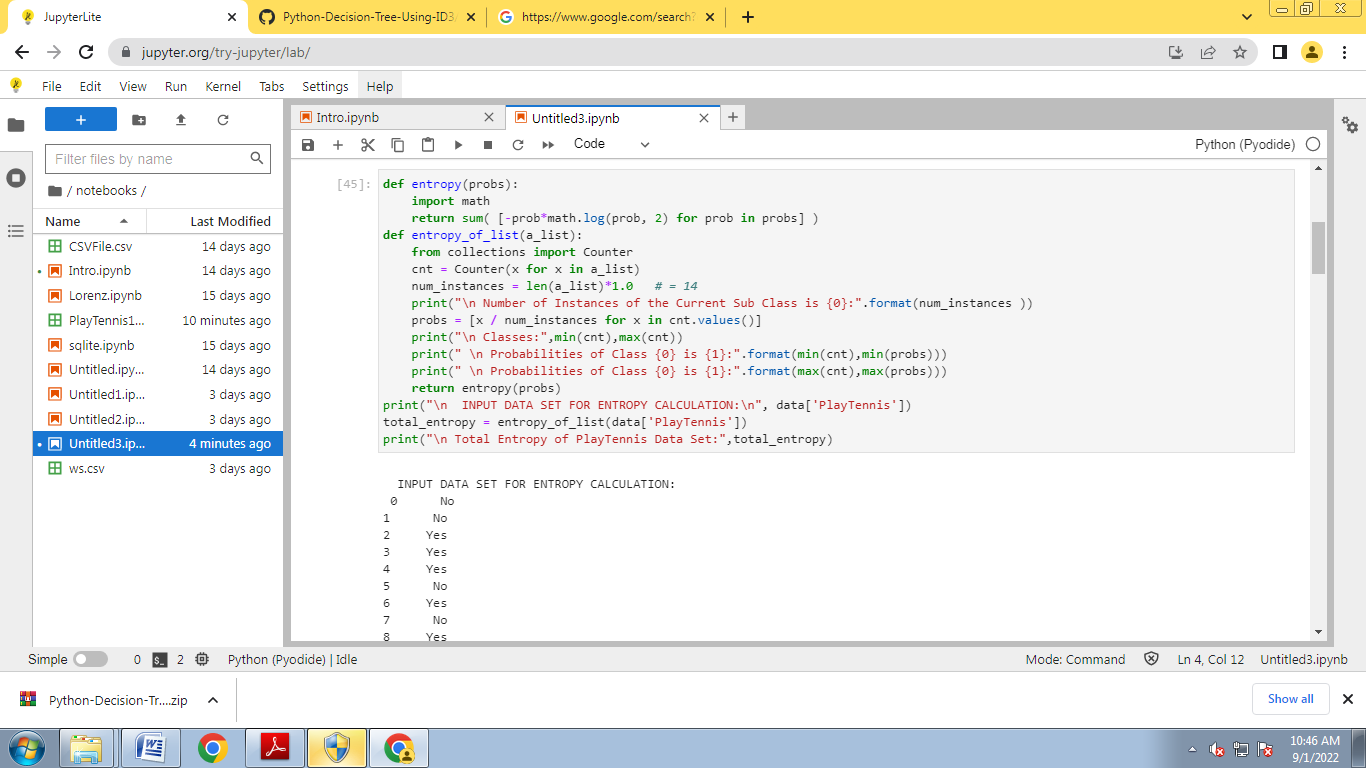


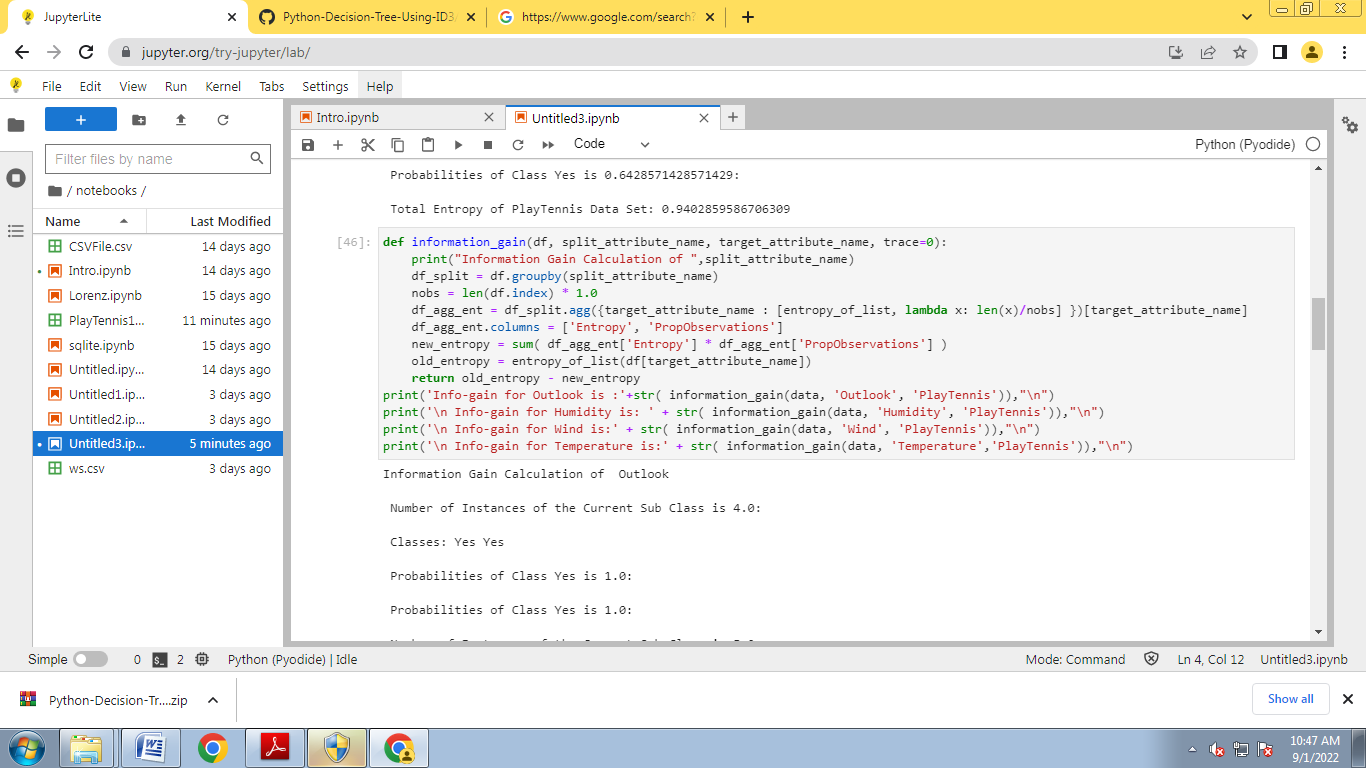
h

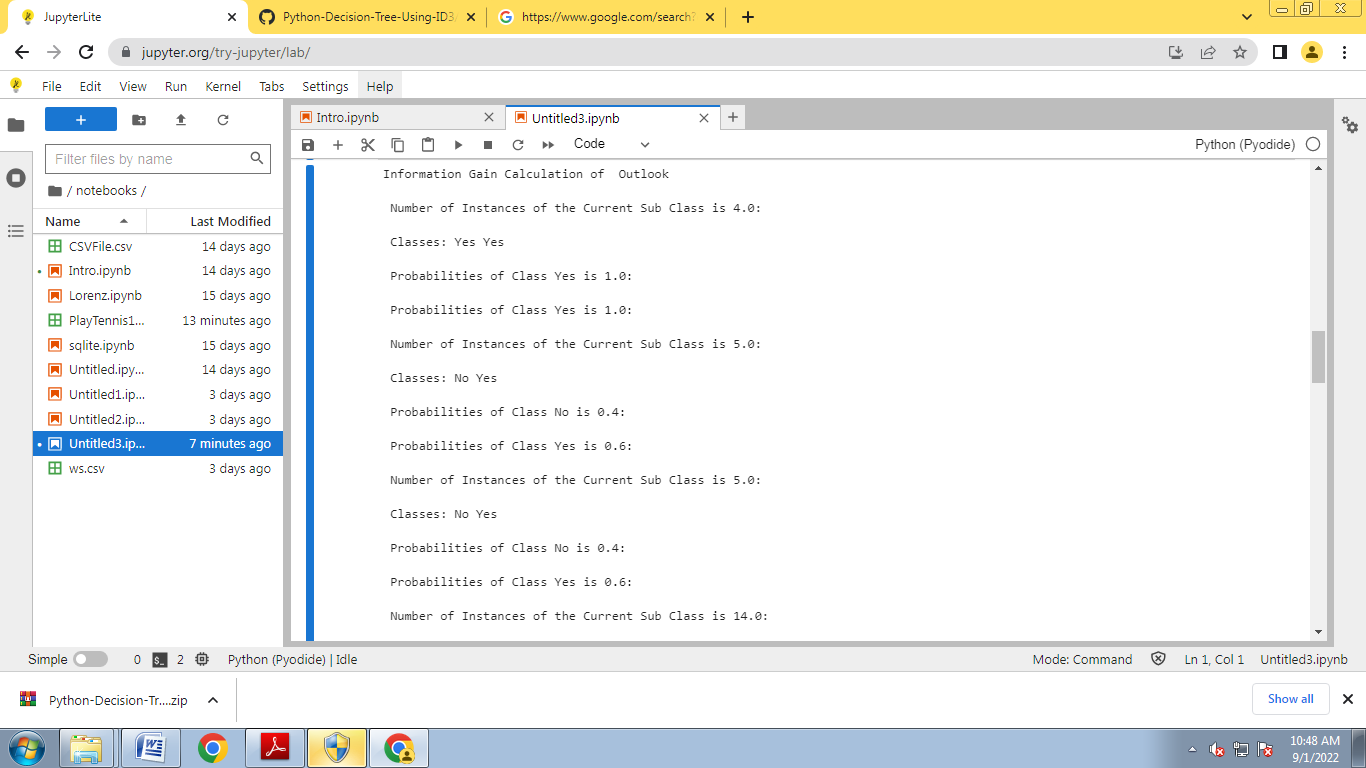
**EXPERIMENT-3:**

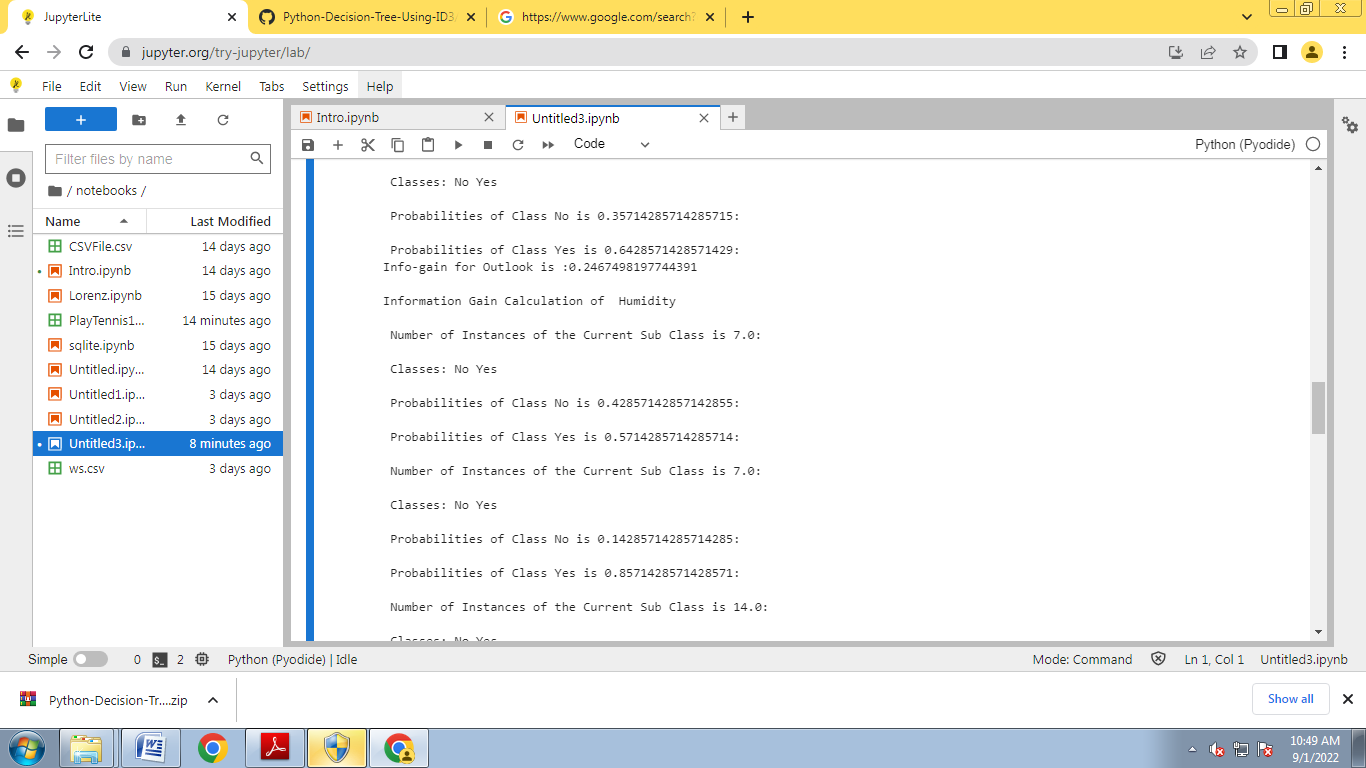
Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample

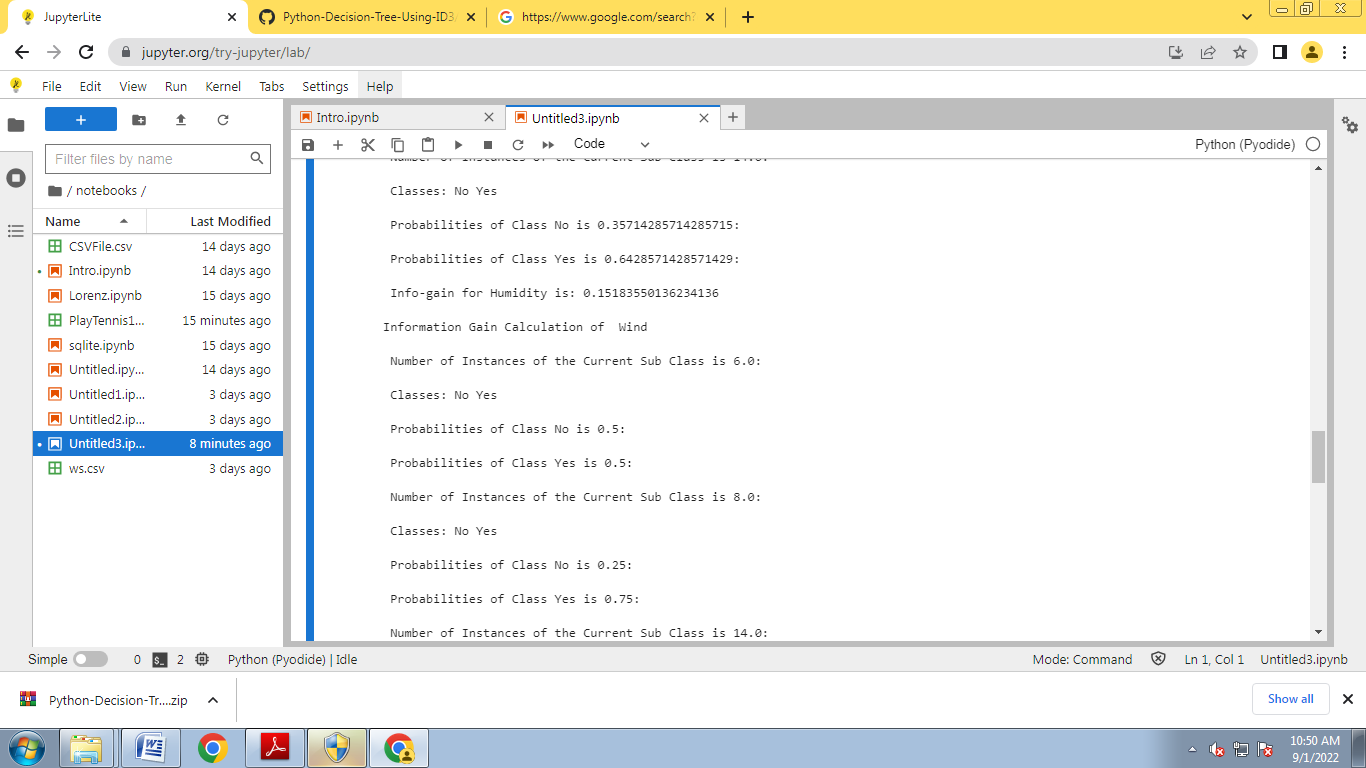


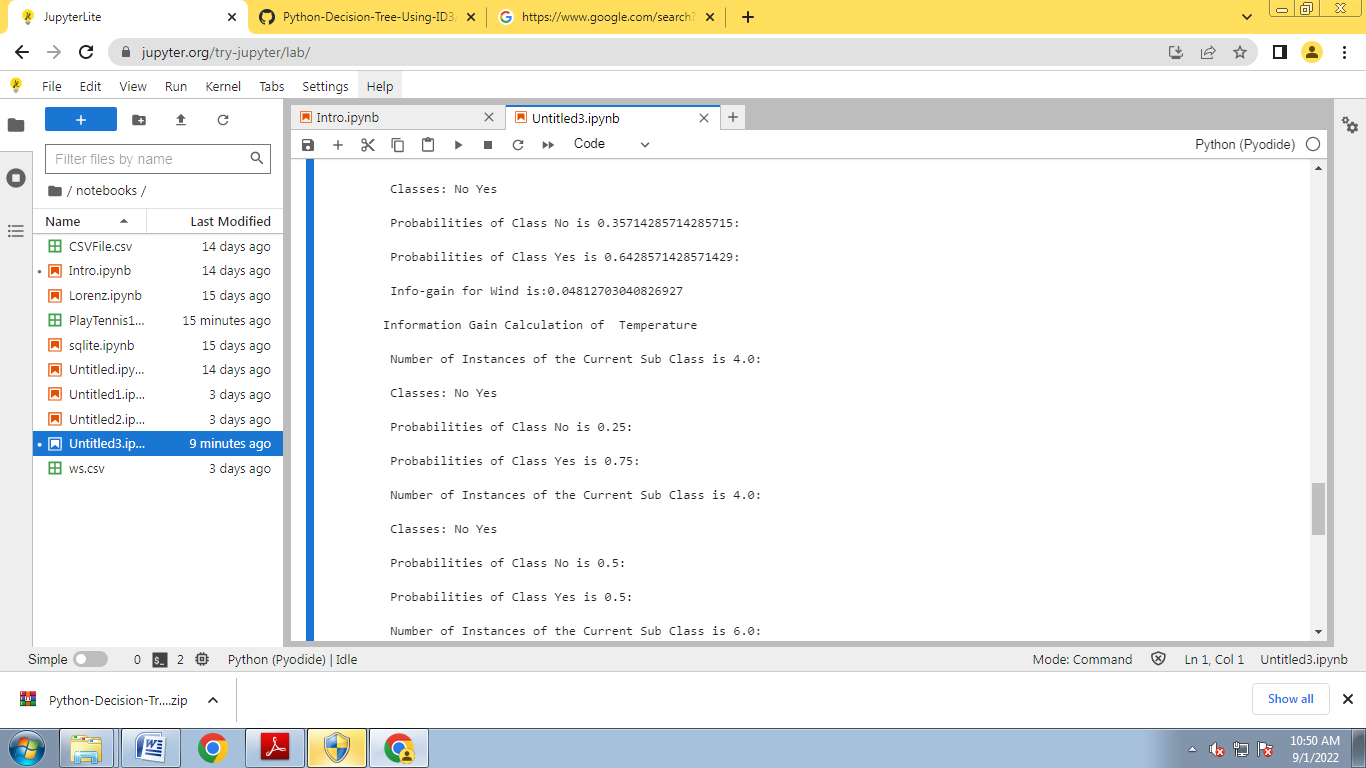


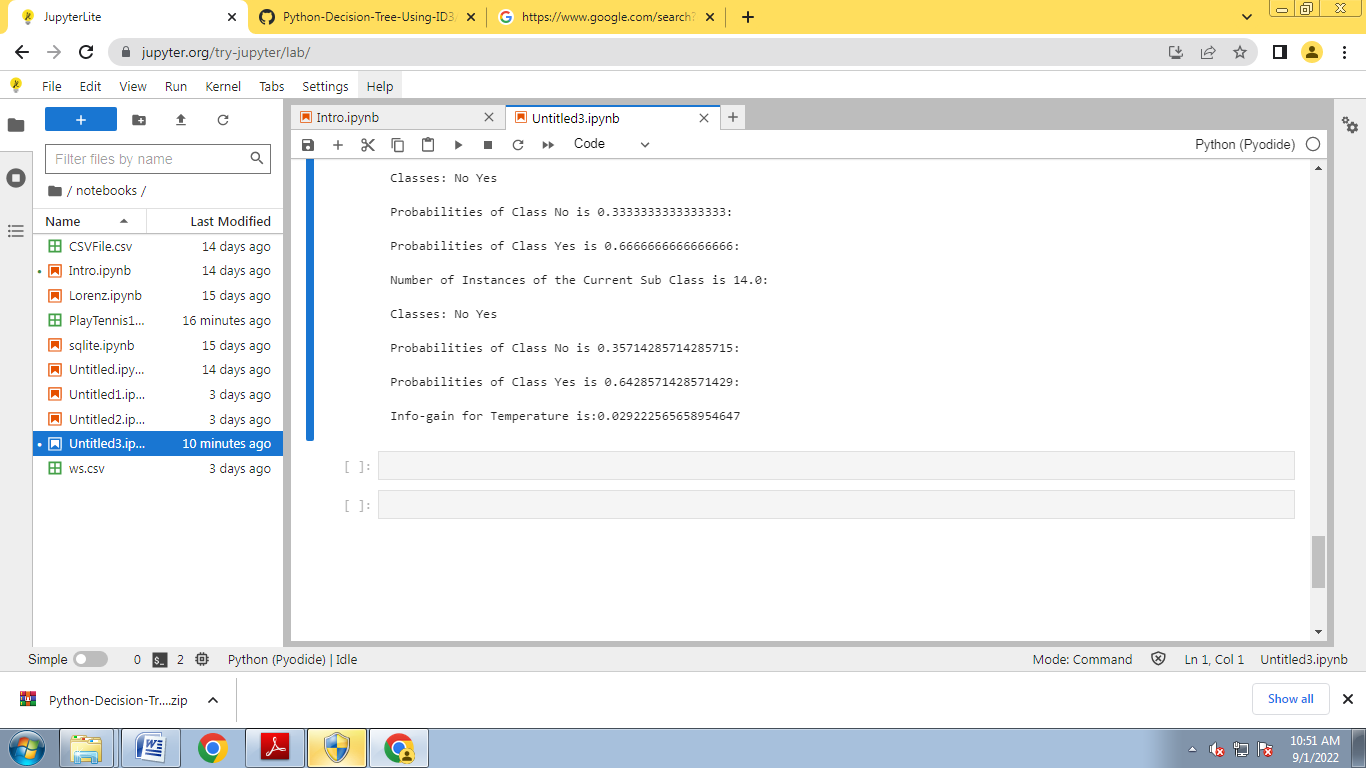












**Experiment-4:**

Exercises to solve the real-world problems using the following machine learning methods:

1. Linear Regression b) Logistic Regression c) Binary Classifier

import numpy as np

import matplotlib.pyplot as plt

def estimate\_coef(x, y):

# number of observations/points

n = np.size(x)

# mean of x and y vector

m\_x = np.mean(x)

m\_y = np.mean(y)

# calculating cross-deviation and deviation about x

SS\_xy = np.sum(y\*x) - n\*m\_y\*m\_x

SS\_xx = np.sum(x\*x) - n\*m\_x\*m\_x

# calculating regression coefficients

b\_1 = SS\_xy / SS\_xx

b\_0 = m\_y - b\_1\*m\_x

return (b\_0, b\_1)

def plot\_regression\_line(x, y, b):

# plotting the actual points as scatter plot

plt.scatter(x, y, color = "m",marker = "o", s = 30)

# predicted response vector

y\_pred = b[0] + b[1]\*x

# plotting the regression line

plt.plot(x, y\_pred, color = "g")

# putting labels

plt.xlabel('x')

plt.ylabel('y')

# function to show plot

plt.show()

def main():

# observations / data

x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])

# estimating coefficients

b = estimate\_coef(x, y)

print("Estimated coefficients:\nb\_0 = {} \

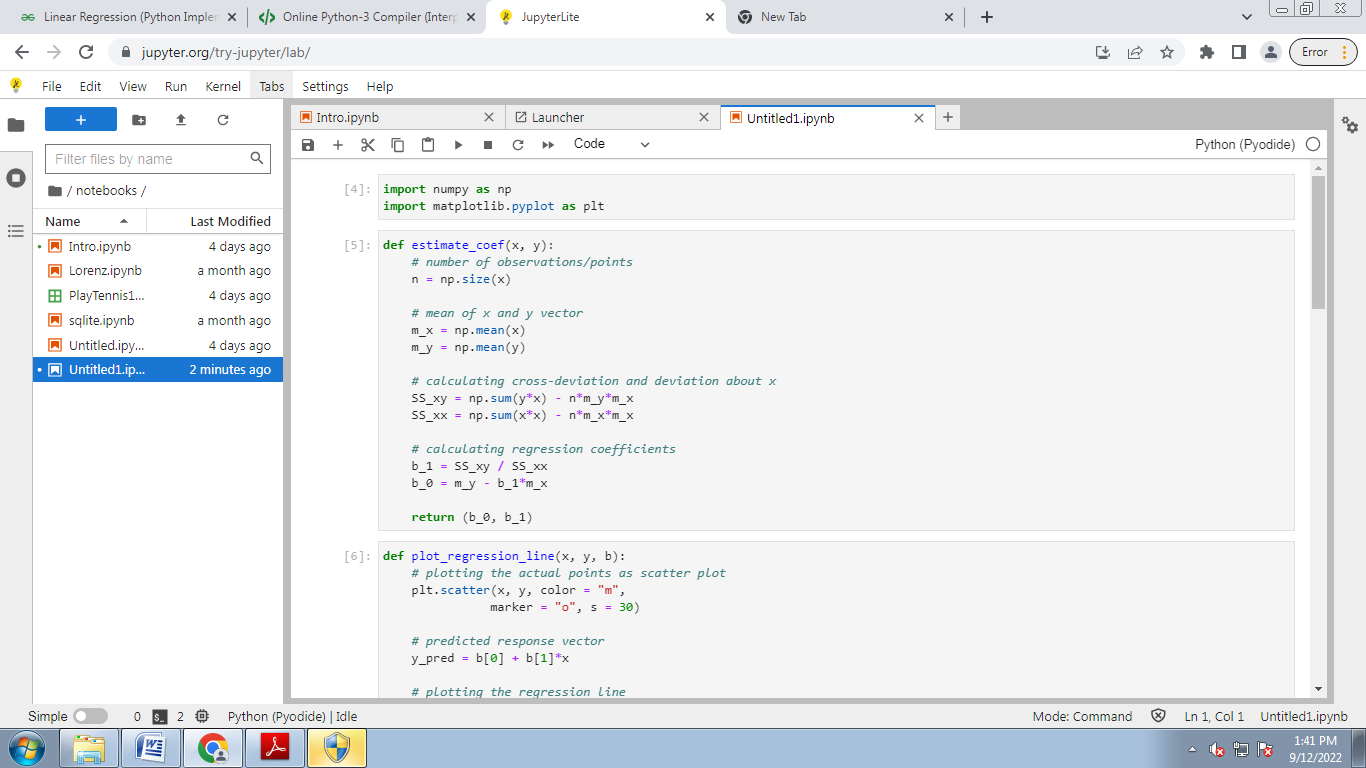
\nb\_1 = {}".format(b[0], b[1]))

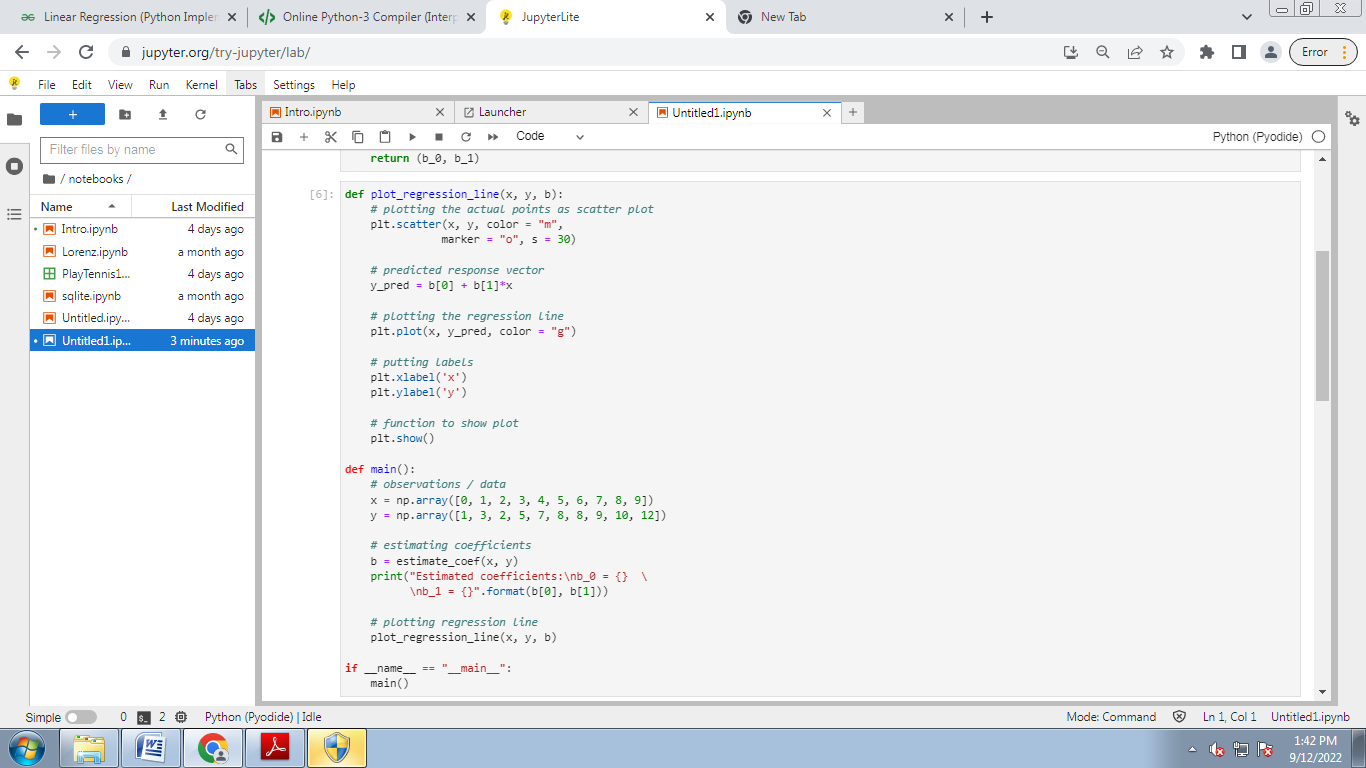
# plotting regression line

plot\_regression\_line(x, y, b)

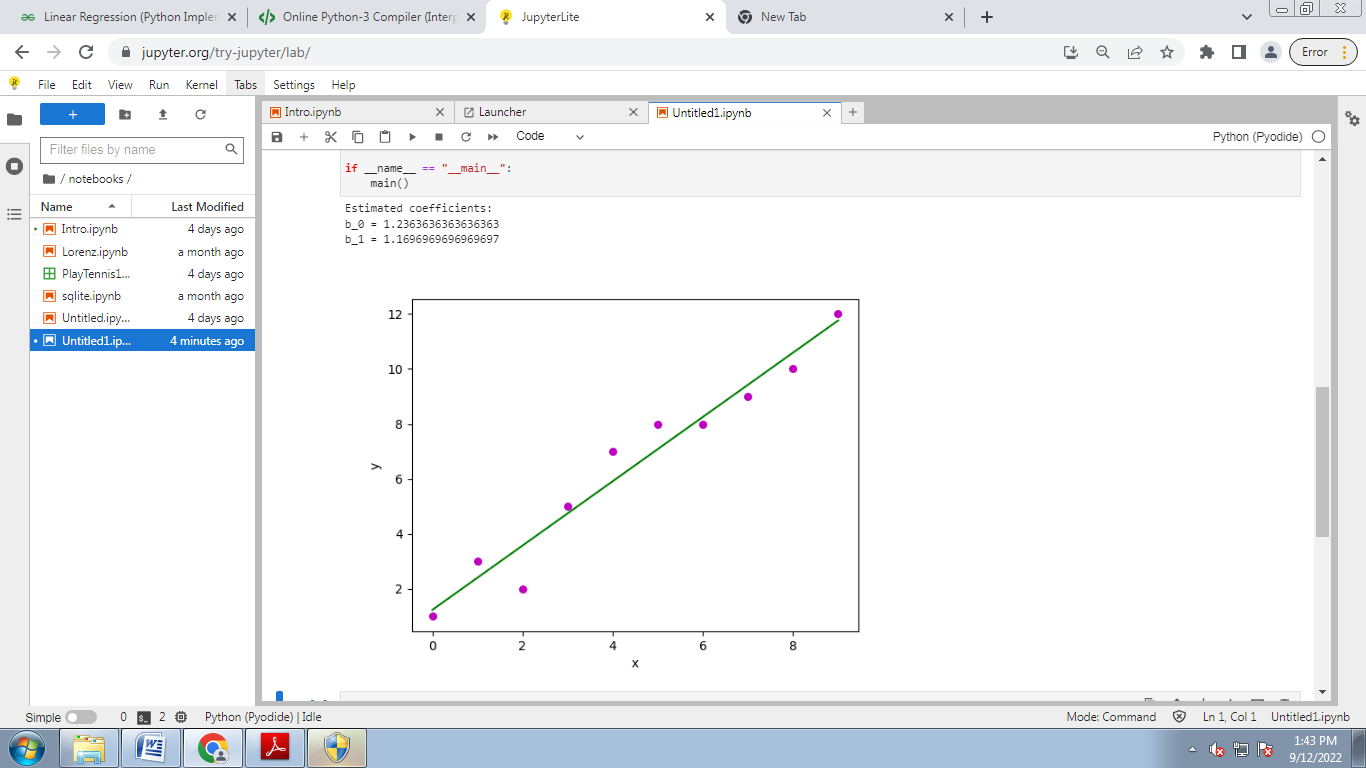
if \_\_name\_\_ == "\_\_main\_\_":

main()

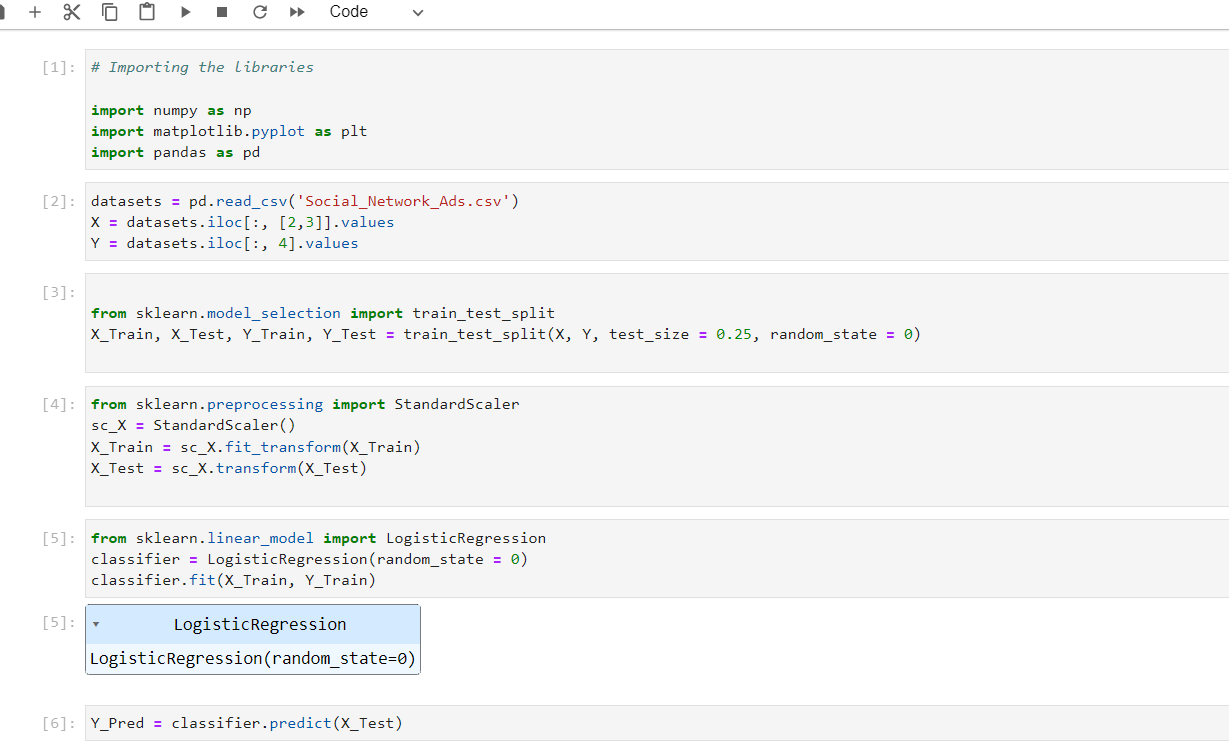




**OUTPUT:**



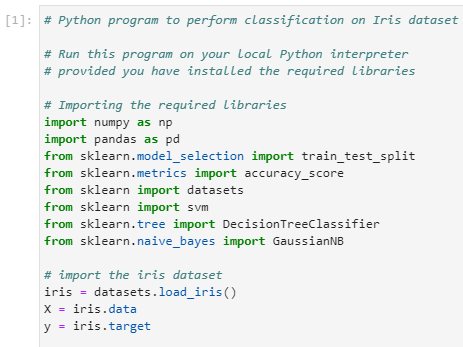
B.LOGISTIC REGRESSION

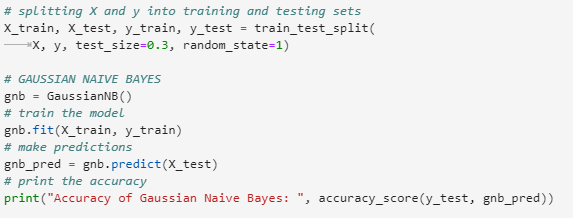


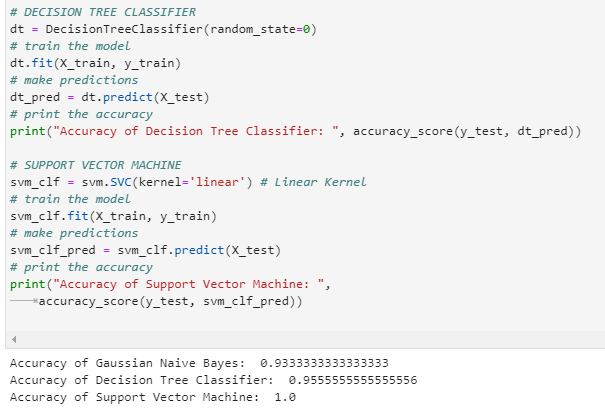




4C.BINARY CLASSIFIER







Experiment-5: Develop a program for Bias, Variance, Remove duplicates , Cross Validation

##### CROSS VALIDATION CODE##############



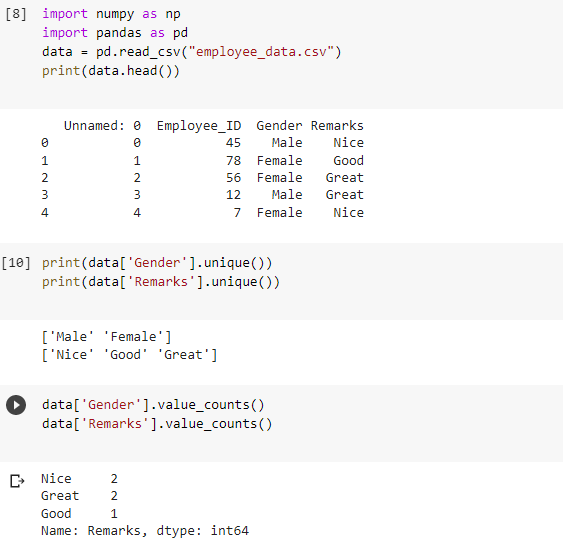
##########Remove Duplicates################

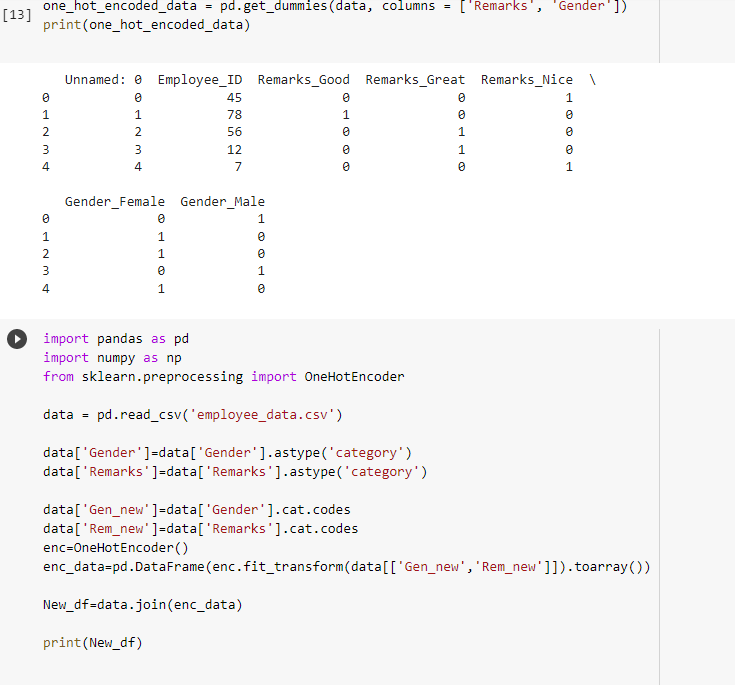


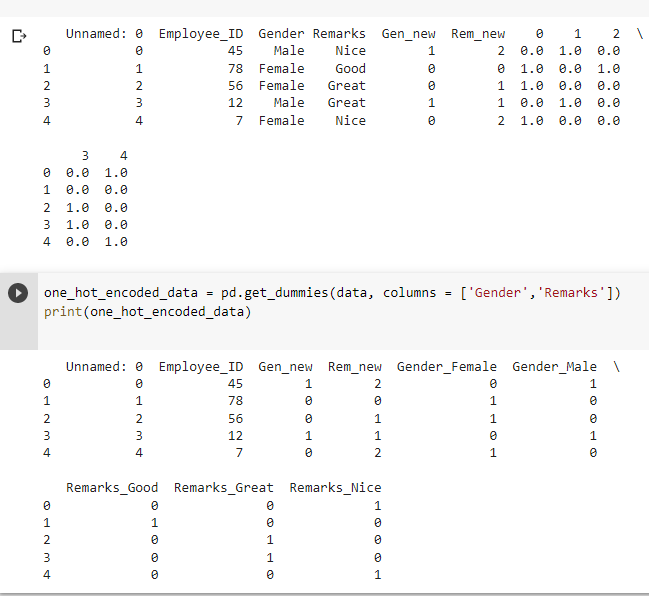
Experiment-6: Write a program to implement Categorical Encoding, One-hot Encoding

NOTE:Requires employee dataset

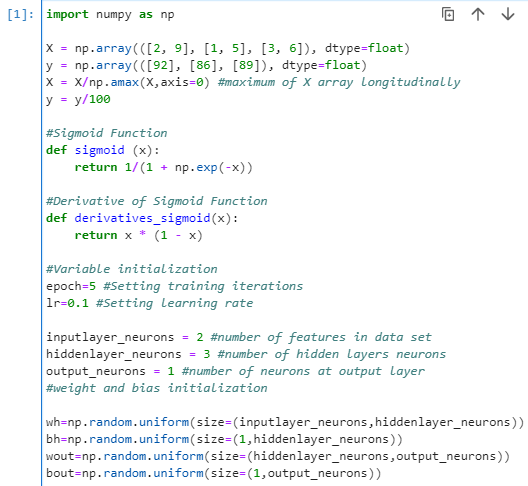
Program:







Experiment-7: Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.





OUTPUT:

-----------Epoch- 1 Starts----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

[[0.85338953]

[0.83991098]

[0.85027323]]

-----------Epoch- 1 Ends----------

-----------Epoch- 2 Starts----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

[[0.85380192]

[0.84030846]

[0.85068423]]

-----------Epoch- 2 Ends----------

-----------Epoch- 3 Starts----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

[[0.85420851]

[0.84070049]

[0.85108948]]

-----------Epoch- 3 Ends----------

-----------Epoch- 4 Starts----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

[[0.85460942]

[0.84108715]

[0.8514891 ]]

-----------Epoch- 4 Ends----------

-----------Epoch- 5 Starts----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

[[0.85500476]

[0.84146857]

[0.8518832 ]]

-----------Epoch- 5 Ends----------

Input:

[[0.66666667 1. ]

[0.33333333 0.55555556]

[1. 0.66666667]]

Actual Output:

[[0.92]

[0.86]

[0.89]]

Predicted Output:

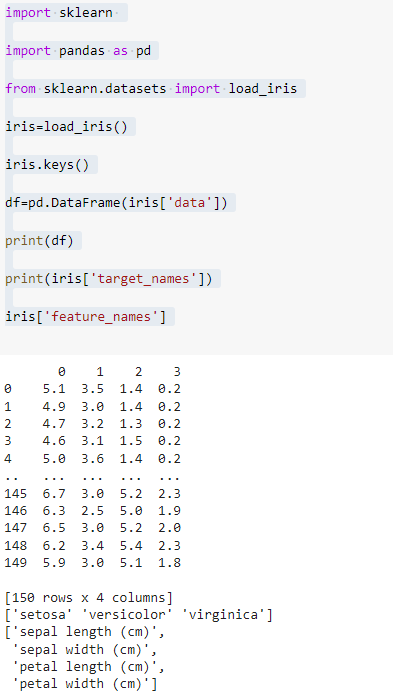
[[0.85500476]

[0.84146857]

[0.8518832 ]]

Experiment-8: Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.#####iris dataset#####

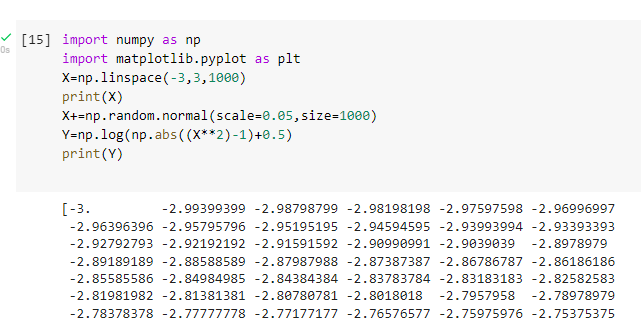
PROGRAM:

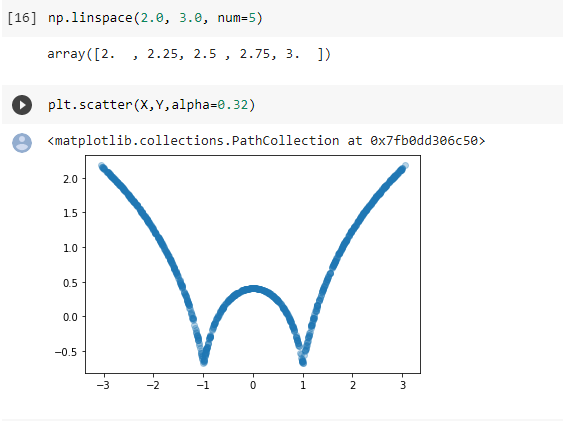


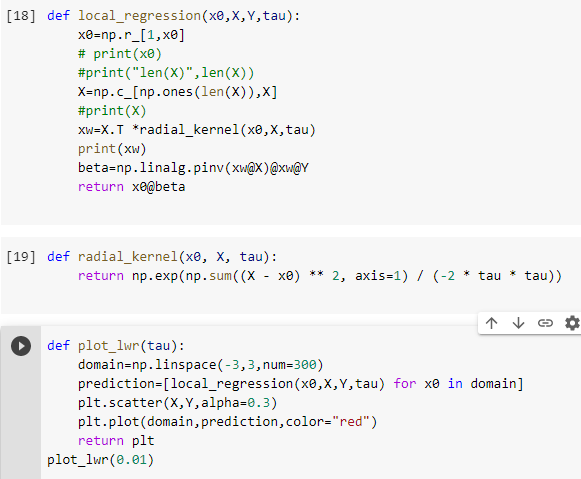


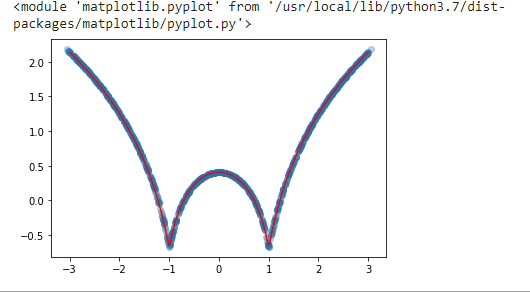
Experiment-9: Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

########google colab is required#################





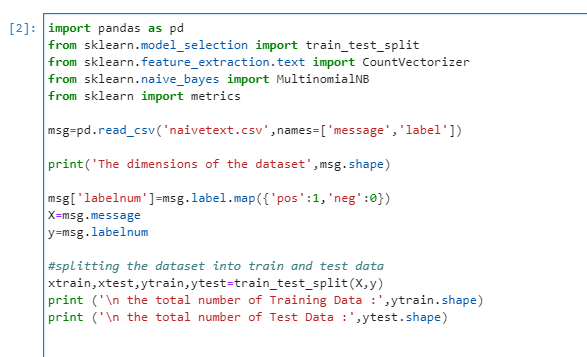


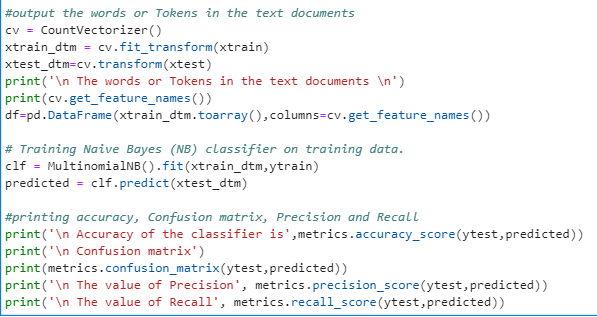


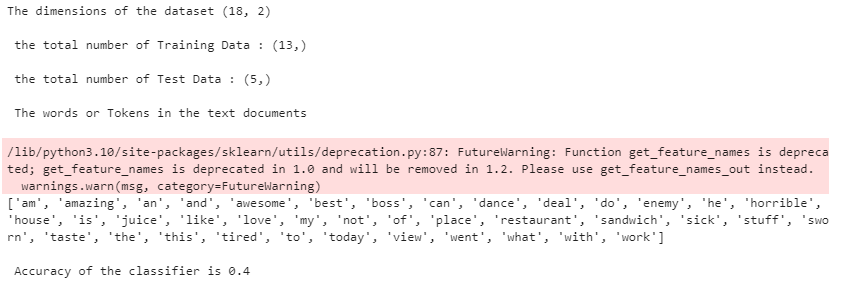
**Experiment-10:**

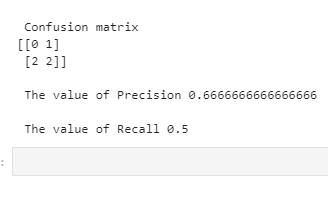
Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

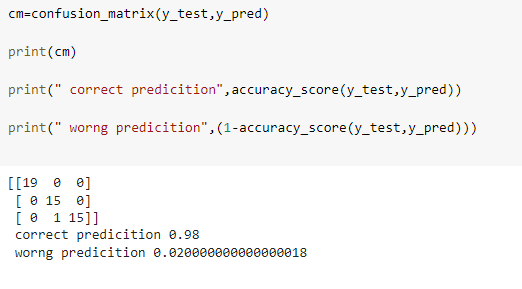
NOTE:Naivetest data set is required





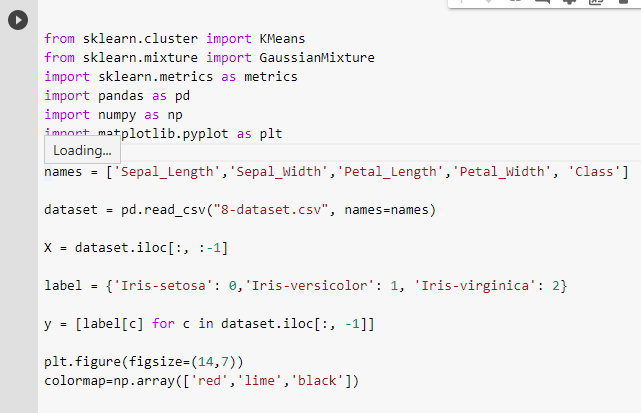


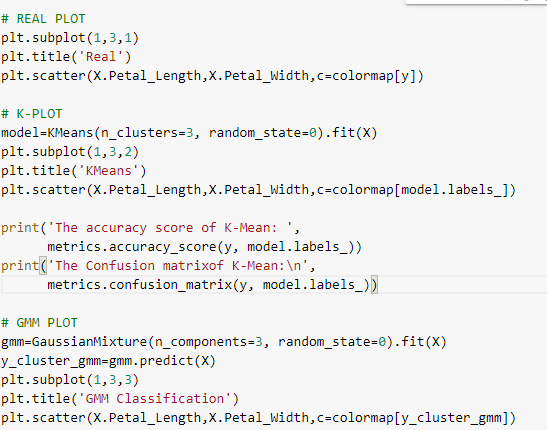


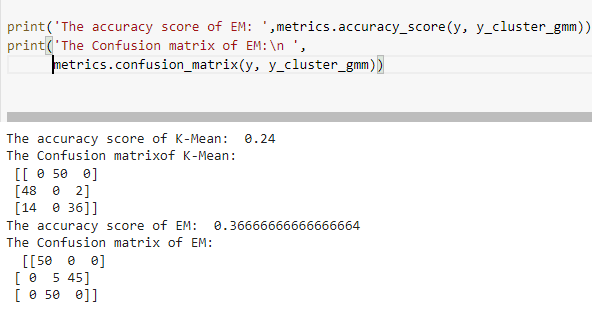


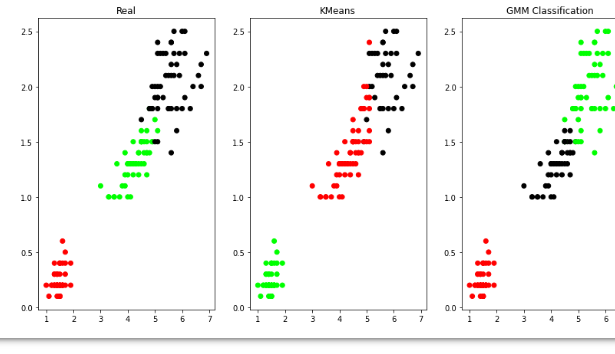
Experiment-11: Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

########## requires colab and data set 8 ###############



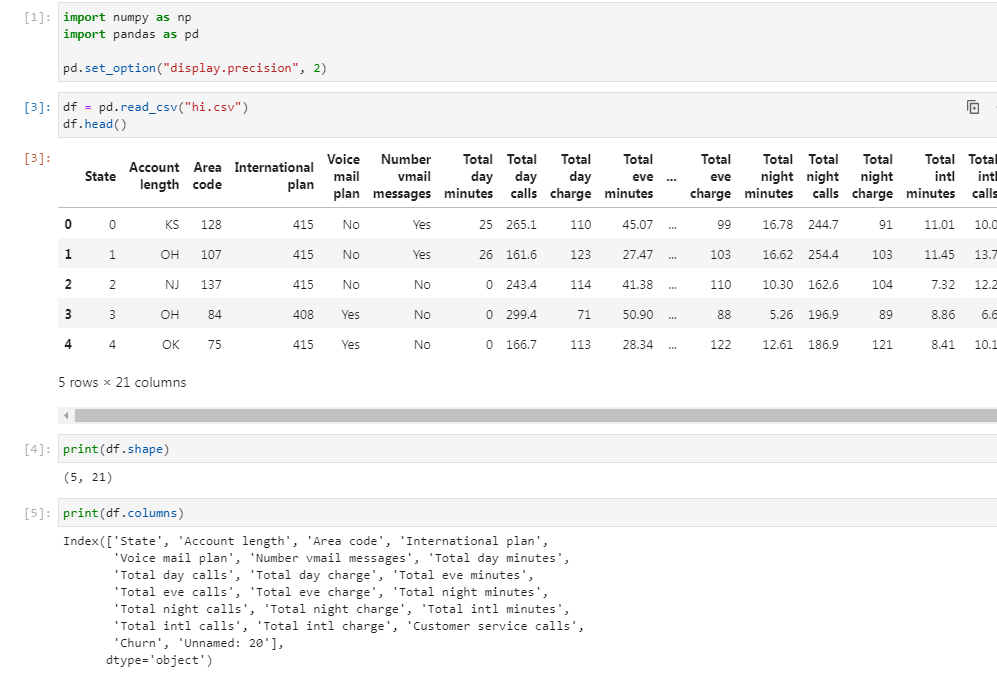




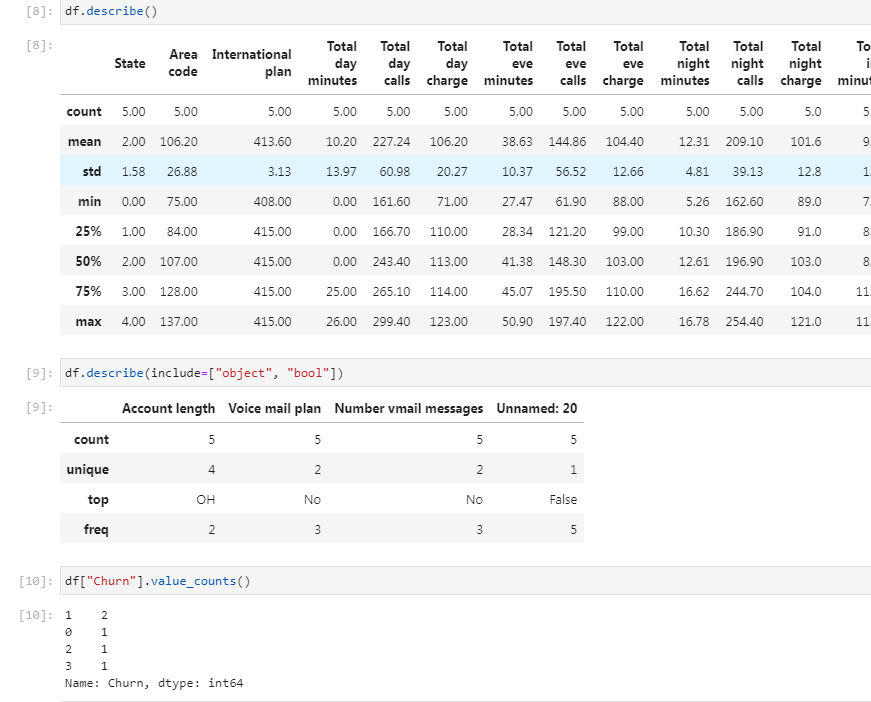


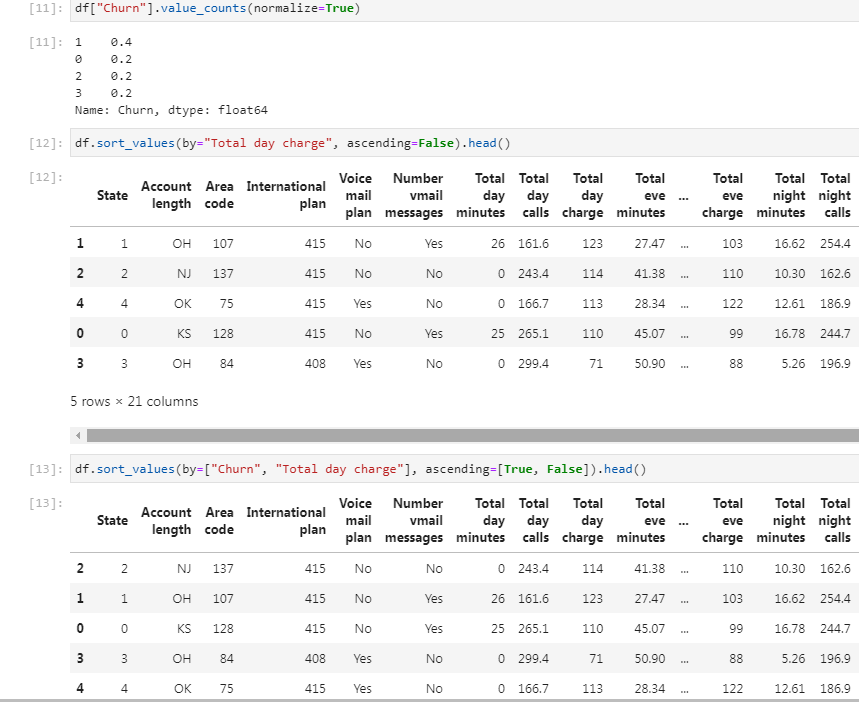
Experiment-12: Exploratory Data Analysis for Classification using Pandas or Matplotlib.

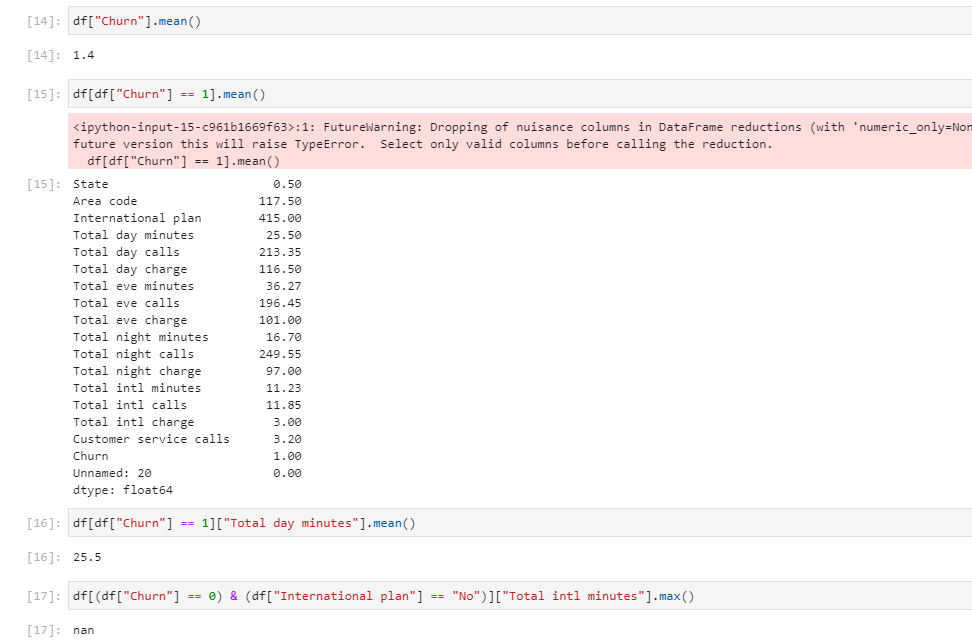
Note:Requires hi csv data

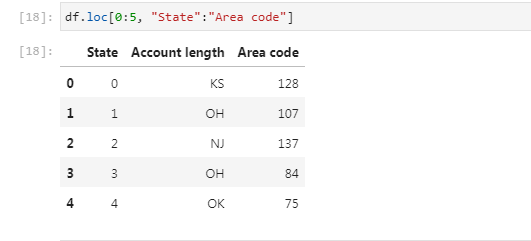








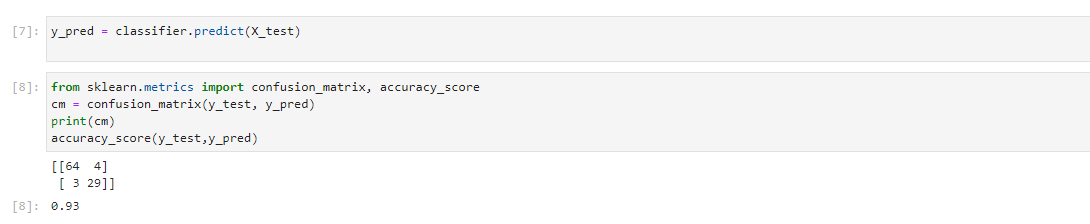


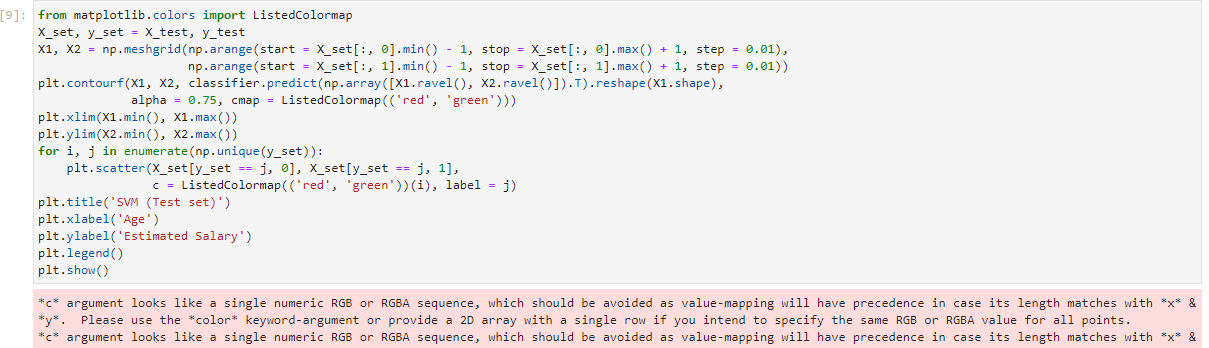


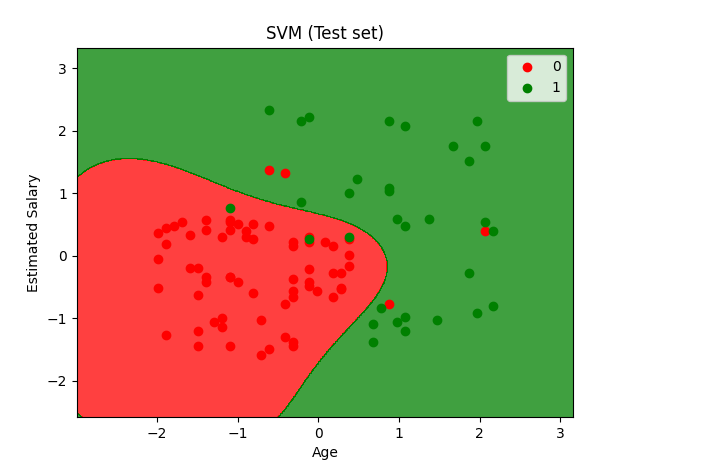
Experiment-14: Write a program to Implement Support Vector Machines and Principle Component Analysis

NOTE:Requires social networking adds data









Experiment-15: Write a program to Implement Principle Component Analysis

#############WINE csv file required#############

