Sri Indu Institute of Engineering and Technology (Autonomous)

(Formerly RVR Institute of Engineering & Technology)

An Autonomous Institution Under UGC

ESTD : 2007

NAAC Accredited. Recognized Under 2(f) of UGC Act 1956

EAMCET CODE: INDI Approved by AICTE, New Delhi, & Affiliated to JNTUH, Hyderabad. JNTUH CODE: X3

Laboratory Manual

Of

Machine Learning

DEPARTMENTOFCOMPUTERSCIENCEAND ENGINEERING

Sri Indu Institute of Engineering and Technology

Sheriguda (v),Ibrahimpatnam (M),R.R.Dist-501510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE VISION

To become a premier institute of academic excellence by providing the world class education that transforms individuals into high intellectuals, by evolving them as empathetic and responsible citizens through continuous improvement.

INSTITUTEMISSION

IM1: To offer outcome-based education and enhancement of technical and practical skills.

IM2: To continuous assess of teaching-learning process through institute-industry collaboration.

IM3: To be a centre of excellence for innovative and emerging fields in technology development with state-of-art facilities to faculty and students fraternity.

IM4: To create an enterprising environment to ensure culture, ethics and social responsibility among the stake holders

SRIINDU INSTITUTEOFENGINEERING ANDTECHNOLOGY

(AnAutonomousInstitution underUGC)

Accredited by NAAC with A+ Grade, Recognized under 2(f) of UGC Act 1956(ApprovedbyAICTE,New DelhiandAffiliatedtoJNTUH,Hyderabad)

Khalsa Ibrahimpatnam, Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist., Telangana – 501 510Website: https://siiet.ac.in/

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGDEPARTMENTVISION

To become a prominent knowledge hub for learners, strive for educational excellence with innovative and industrial techniques so as to meet the global needs.

DEPARTMENTMISSION

DM1: To provide ambience that enhances innovations, problem solving skills, leadership qualities, decision making, team-spirit and ethical responsibilities.

DM2: To impart quality education with professional and personal ethics, so as to meet the challenging technological needs of the industry and society.

DM3: To provide academic infrastructure and develop linkage with the world class organizations to strengthen industry-academic relationships for learners.

DM4: To provide and strengthen new concepts of research in the thrust area of Computer Science and Engineering to reach the needs of Government and Society.

PROGRAMEDUCATIONAL OBJECTIVES

Program Educational objectives are to Promote:

PEO1: To develop trained graduates with strong academic and technical skills of modern computer science and engineering.

PEO2: To promote trained graduates with leadership qualities and the ability to solve real time problems using current techniques and tools in inter disciplinary environment.

PEO3: To motivate the graduates towards lifelong learning through continuing education and professional development.

PROGRAMSPECIFICOUTCOMES

PSO1:Professional Skills: To implement computer programsofvaryingcomplexityintheareasrelatedtoWebDesign,CloudComputing,NetworkSecurityandArtificialIntelligence.

PSO 2: Problem-Solving Skills: To develop quality products using open ended programming environment.

PROGRAMOUTCOMES

PO1: ENGINEERINGKNOWLEDGE: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: PROBLEMANALYSIS: Identify, formulate, research literature, and analyze complex engineering problem sreaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: DESIGN/DEVELOPMENT OF SOLUTIONS: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: CONDUCTINVESTIGATIONS OF COMPLEX PROBLEMS: Use research-based
Knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

PO5:.MODERN TOOL USAGE: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: THE ENGINEER AND SOCIETY: Apply reasoning informed by the contextual knowledge o assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: ENVIRONMENT AND SUSTAINABILITY: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: ETHICS: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: INDIVIDUAL AND TEAM WORK: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: COMMUNICATION: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

PO11: PROJECT MANAGEMENT AND FINANCE: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12: LIFE-LONG LEARNING: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MACHINE LEARNING LAB (Course Code: CS15543)

B.Tech. III Year I Sem.	**L**	**P**	**T**	**C**
	0	**0**	**3**	**1.5**

LIST OF EXPERIMENTS

<table>
<tr><th>S. No.</th><th>Name of the Experiment</th></tr>
<tr><td>1.</td><td>The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is the probability that a student is absent given that today is Friday? Apply Bayer’s rule in python to get the result. (Ans: 15%)</td></tr>
<tr><td>2.</td><td>Extract the data from database using python</td></tr>
<tr><td>3.</td><td>Implement k-nearest neighbors classification using python</td></tr>
<tr><td>4.</td><td>Given the following data, which specify classifications for nine Combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)
<table>
<tr><th>VAR1</th><th>VAR2</th><th>CLASS</th></tr>
<tr><td>1.713</td><td>1.586</td><td>0</td></tr>
<tr><td>0.180</td><td>1.786</td><td>1</td></tr>
<tr><td>0.353</td><td>1.240</td><td>1</td></tr>
<tr><td>0.940</td><td>1.566</td><td>0</td></tr>
<tr><td>1.486</td><td>0.759</td><td>1</td></tr>
<tr><td>1.266</td><td>1.106</td><td>0</td></tr>
<tr><td>1.540</td><td>0.419</td><td>1</td></tr>
<tr><td>0.459</td><td>1.799</td><td>1</td></tr>
<tr><td>0.773</td><td>0.186</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>5.</td><td>The following training examples map descriptions of individualsonto high, medium and low credit-worthiness.</td></tr>
</table>

	medium skiing design single twenties no ->high Risk high golf trading married forties yes ->low Risk Low speedway transport married thirties yes ->med Risk medium football banking single thirties yes ->low Risk high flying media married fifties yes ->high Risk Low football security single twenties no ->med Risk medium golf media single thirties yes ->med Risk medium golf transport married forties yes ->low Risk high skiing banking single thirties yes ->high Risk Low golf unemployed married forties yes ->high-risk Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `med Risk' in the dataset?
6.	Implement linear regression using python.
7.	Implement Naïve Bayes theorem to classify the English text
8.	Implement an algorithm to demonstrate the significance of genetic algorithm

INTRODUCTION TO MACHINE LEARNING LAB

Machine Learning is used anywhere from automating mundane tasks to offering intelligent insights, industries in every sector try to benefit from it. You may already be using a device that utilizes it. For example, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home. But there are much more examples of ML in use.

- **Prediction:** Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
- **Image recognition**: Machine learning can be used for face detection in an image as well. There is aseparate category for each person in a database of several people.
- **Speech Recognition:** It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialing, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
- **Medical diagnoses:** ML is trained to recognize cancerous tissues.
- **Financial industry: T**rading companies use ML in fraud investigations and credit checks.

Types of Machine Learning?

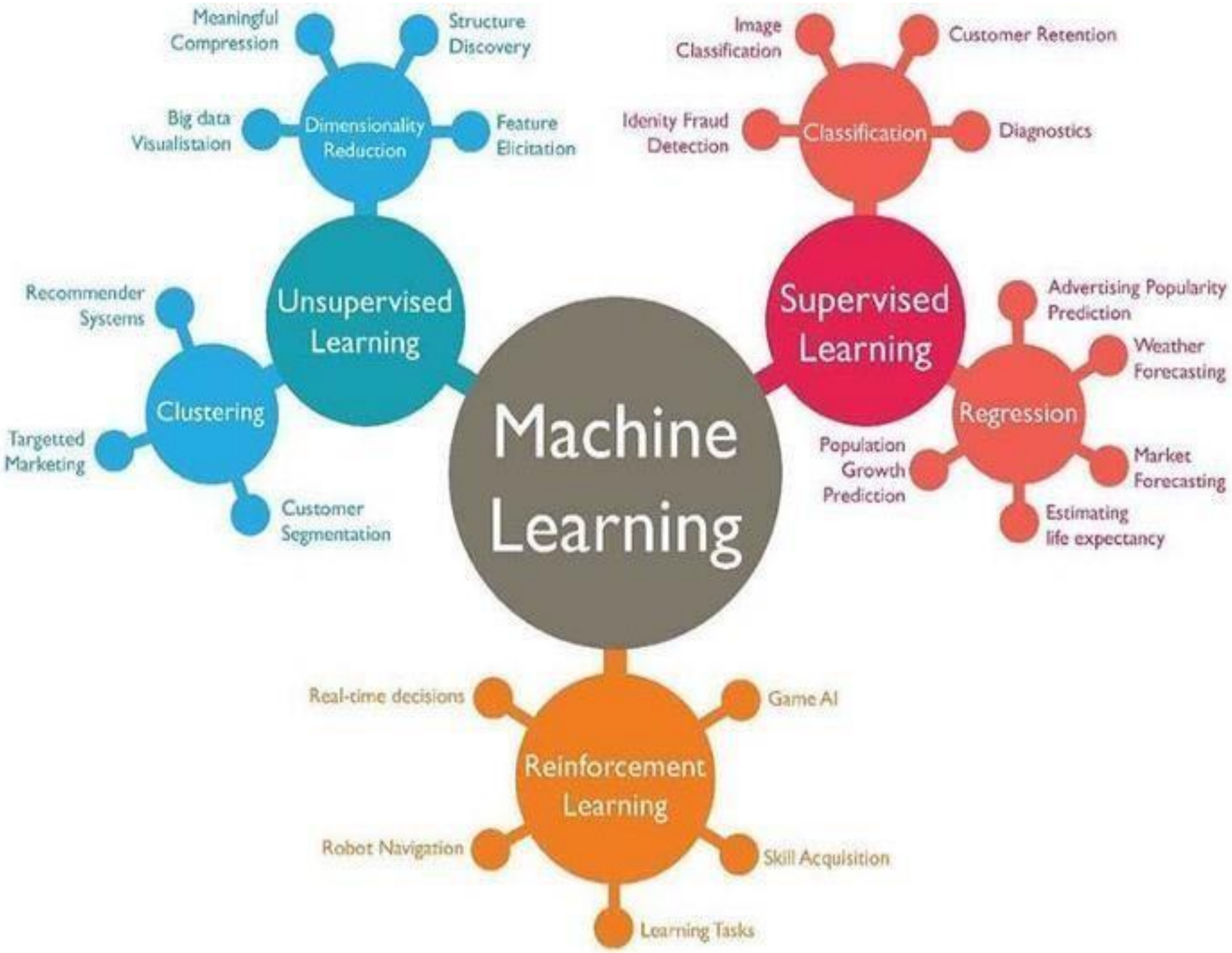
Machine learning can be classified into 3 types of algorithms

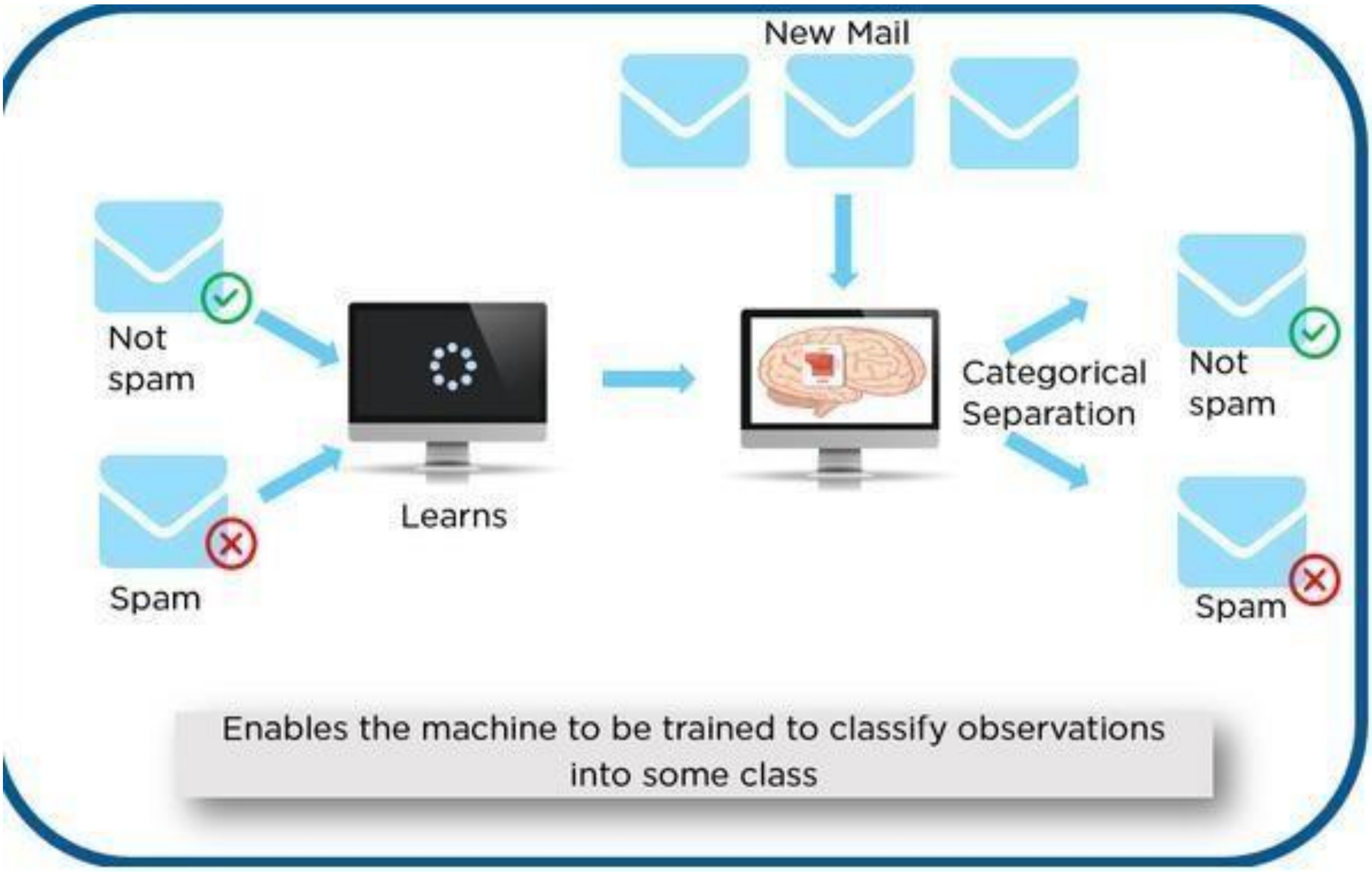
1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Overview of Supervised Learning Algorithm

In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label.

The goal is to approximate the mapping function so well that when you have new input data (x) that youcan predict the output variables (Y) for that data.




New Mail
Not spam
Spam
Learns
Categorical Separation
Not spam
Spam
Enables the machine to be trained to classify observations into some class

As shown in the above example, we have initially taken some data and marked them as ‘Spam’ or ‘Not Spam’. This labeled data is used by the training supervised model, this data is used to train the model.

Once it is trained we can test our model by testing it with some test new mails and checking of the model is able to predict the right output.

Types of Supervised learning

- **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

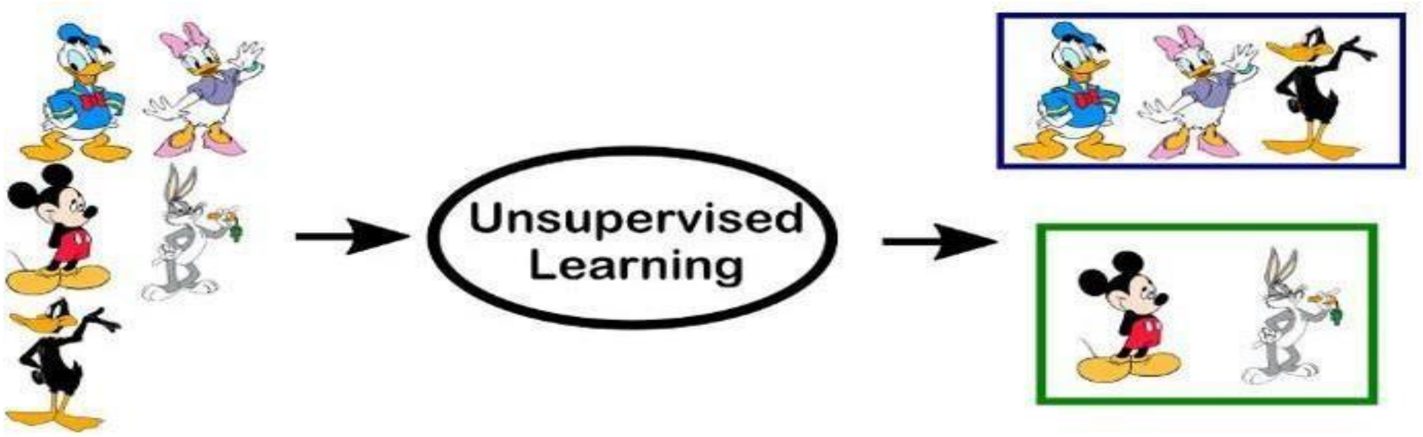
Overview of Unsupervised Learning Algorithm

In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

Types of Unsupervised learning:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

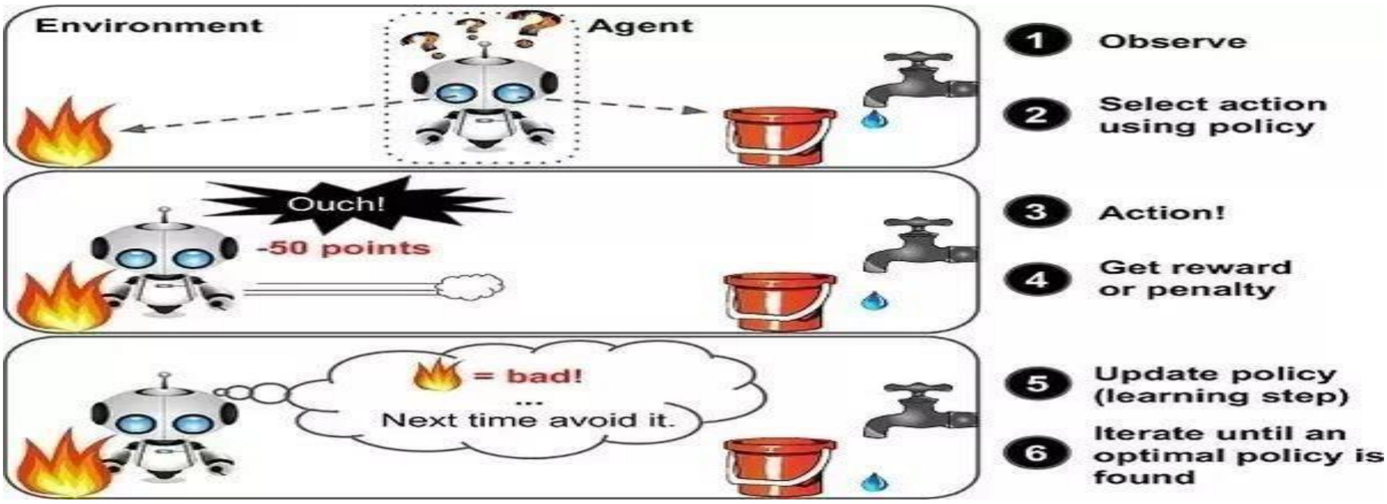
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.



Example of Unsupervised Learning

Overview of Reinforcement Learning

A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty. It is a type of dynamic programming that trains algorithms using a system of reward and punishment.


Environment
Agent
Ouch!
-50 points
= bad!
...
Next time avoid it.
1 Observe
2 Select action using policy
3 Action!
4 Get reward or penalty
5 Update policy (learning step)
6 Iterate until an optimal policy is found

in the above example, we can see that the agent is given 2 options i.e. a path with water or a path with fire. A reinforcement algorithm works on reward a system i.e. if the agent uses the fire path then the rewards are subtracted and agent tries to learn that it should avoid the fire path. If it had chosen the water path or the safe path then some points would have been added to the reward points, the agent then would try to learn what path is safe and what path isn’t.

It is basically leveraging the rewards obtained; the agent improves its environment knowledge to select the next action.

PROGRAM 1

The probability that it is Friday and that a student is absent is 3%. Since thereare5school days in a week, the probability that it is Friday is20%. What is the probability that a student is absent given that today is Friday? Apply Bayes' rule in python to get the result. (Ans: 15%)

AIM: To find the probability that a student is absent given that today is Friday.

DESCRIPTION:

Machine learning is a method of data analysis that automates analytical model building of data set. Using the implemented algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look. Naive bayes algorithm is one of the most popular machines learning technique. In this article we will look how to implement Naive Baye's algorithm using python.

Before someone can understand Bayes' theorem, they need to know a couple of related concepts first, namely, the idea of Conditional Probability, and Bayes' Rule.

Conditional Probability is just what is the probability that something will happen, given that something else has already happened.

Let say we have a collection of people. Some of them are singers. They are either male or female. If we select a random sample, what is the probability that this person is a male? what is the probability that this person is a male and singer? Conditional Probability is the best option here. We can calculate probability like,

P(Singer & Male) = P(Male) x P(Singer / Male)

What is Bayes rule?

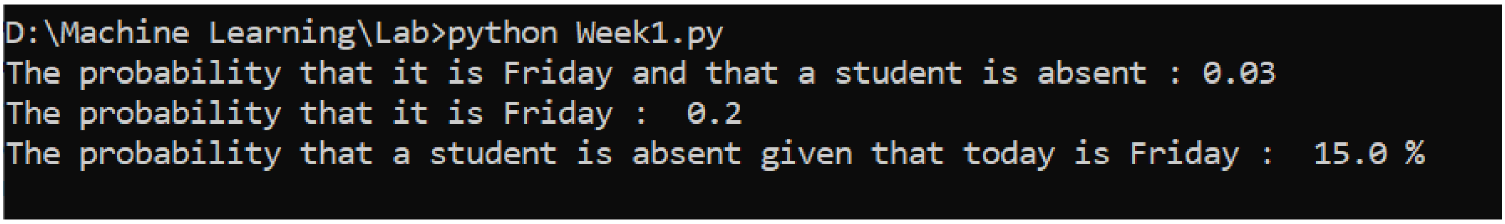
We can simply define Bayes rule like this. Let A1, A2, …, An be a set of mutually exclusive events that together form the sample space S. Let B be any event from the same sample space, such that P(B) > 0. Then, P (Ak | B) = P(Ak ∩ B) / P(A1 ∩ B) + P(A2 ∩ B) + . . . + P(An ∩ B)

What is Bayes classifier?

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features in machine learning. Basically we can use above theories and equations for classification problem.

```
=================================
Explanation:
=================================
       F: Friday
       A: Absent

       Based on the given problem statement,

       The probability that it is Friday and that a student is absent is 3%
       i.e.
       P(A ∩ F) = 3% = 3 / 100 = 0.03

       and

       The probability that it is Friday is 20%
       i.e.

       P(F)=20% = 20/100 = 0.2

       Then,

       The probability that a student is absent given that today is Friday
       P(A | F)

       By the definition of Baye’s rule (conditional probability), we have

       P(A | F) = P(A ∩ F) / P(F)


===============================
Source Code :
===============================

'''

# The probability that it is Friday and that a student is absent is 3%
pAF=0.03
print ("The probability that it is Friday and that a student is absent:", pAF)
# The probability that it is Friday is 20%
pF=0.2
print ("The probability that it is Friday: “, pF)
# The probability that a student is absent given that today is Friday
result=(pAF/pF)
# Display the Result
print ("The probability that a student is absent given that today is Friday: ", resul
t * 100,"%")
```

Output:



```
D:\Machine Learning\Lab>python Week1.py
The probability that it is Friday and that a student is absent : 0.03
The probability that it is Friday :  0.2
The probability that a student is absent given that today is Friday :  15.0 %
```

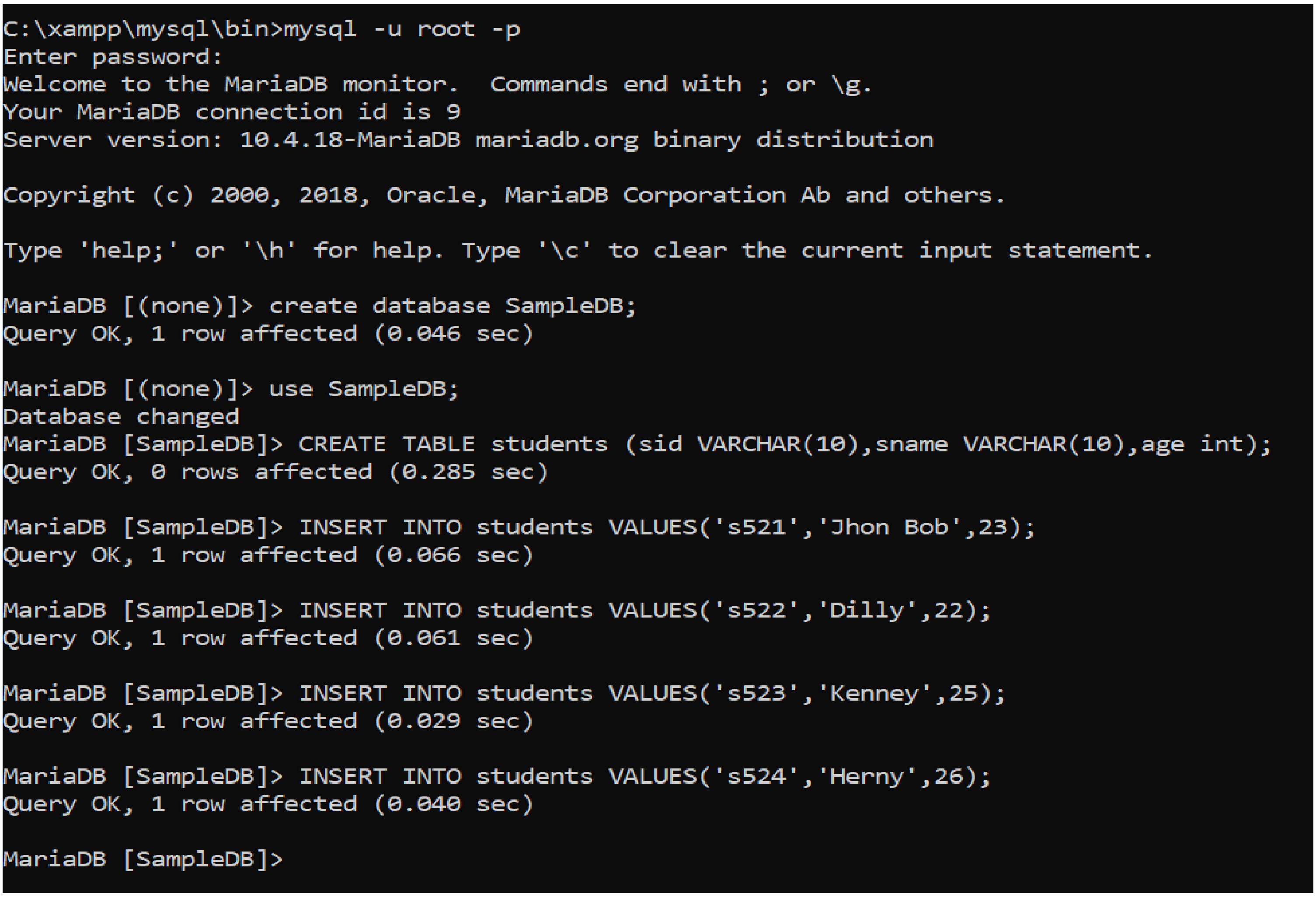
PROGRAM 2:

EXTRACT THE DATA FROM DATABASE USING PYTHON

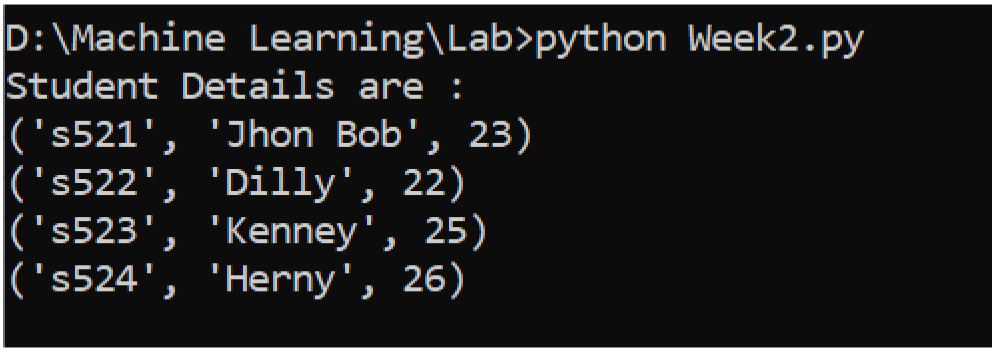
```
=================================
Explanation:
=================================

===> First You need to Create a Table (students) in Mysql Database (SampleDB)

---> Open Command prompt and then execute the following command to enter into MySQL prompt.

--> mysql -u root -p


And then, you need to execute the following commands at MySQL prompt to create table in the database.

--> create database SampleDB;

--> use SampleDB;

--> CREATE TABLE students (sid VARCHAR(10),sname VARCHAR(10),age int);

--> INSERT INTO students VALUES('s521','Jhon Bob',23);
--> INSERT INTO students VALUES('s522','Dilly',22);
--> INSERT INTO students VALUES('s523','Kenney',25);
--> INSERT INTO students VALUES('s524','Herny',26);


===> NeXT, Open Command prompt and then execute the following command to install mysql.connector package to connect with mysql database through python.

--> pip install mysql.connector (Windows)
--> sudo apt-get install mysql.connector (Linux)

===============================
Source Code:
===============================. '''

import mysql.connector
# Create the connection object
myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "",database="SampleDB")
# Creating the cursor object
cur = myconn.cursor()
# Executing the query
cur.execute("select * from students")
# Fetching the rows from the cursor object
result = cur.fetchall()
print("Student Details are:")
```

```
# Printing the result
for x in result:
       print(x);
# Commit the transaction
myconn.commit()
# Close the connection
myconn.close()
```

Output:



```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.18-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database SampleDB;
Query OK, 1 row affected (0.046 sec)

MariaDB [(none)]> use SampleDB;
Database changed
MariaDB [SampleDB]> CREATE TABLE students (sid VARCHAR(10),sname VARCHAR(10),age int);
Query OK, 0 rows affected (0.285 sec)

MariaDB [SampleDB]> INSERT INTO students VALUES('s521','Jhon Bob',23);
Query OK, 1 row affected (0.066 sec)

MariaDB [SampleDB]> INSERT INTO students VALUES('s522','Dilly',22);
Query OK, 1 row affected (0.061 sec)

MariaDB [SampleDB]> INSERT INTO students VALUES('s523','Kenney',25);
Query OK, 1 row affected (0.029 sec)

MariaDB [SampleDB]> INSERT INTO students VALUES('s524','Herny',26);
Query OK, 1 row affected (0.040 sec)

MariaDB [SampleDB]>
```



```
D:\Machine Learning\Lab>python Week2.py
Student Details are :
('s521', 'Jhon Bob', 23)
('s522', 'Dilly', 22)
('s523', 'Kenney', 25)
('s524', 'Herny', 26)
```

PROGRAM 3

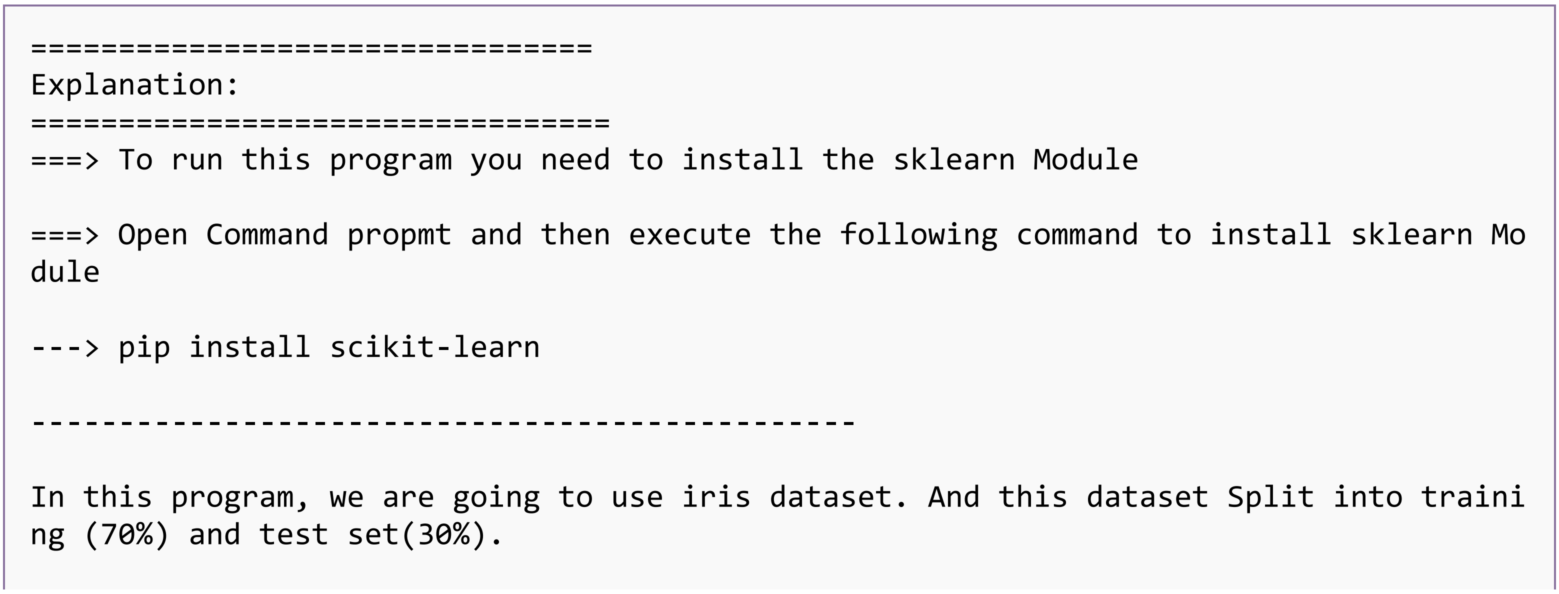
AIM: To Implement k-neighbor’s classification using python

DESCRIPTION:

This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line.
Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to over fit the data and resulting in complex models.
It’s very important to have the right k-value when analyzing the dataset to avoid over fitting and under fitting of the dataset.

KNN MODEL REPRESENTATION:

The model representation for KNN is the entire training dataset.
It is as simple as that. KNN has no model other than storing the entire dataset, so there is no learning required.
Efficient implementations can store the data using complex data structures like k-d trees to make look-up and matching of new patterns during prediction efficient.
Because the entire training dataset is stored, you may want to think carefully about the consistency of your training data. It might be a good idea to curate it, update it often as new data becomes available and remove erroneous and outlier data.

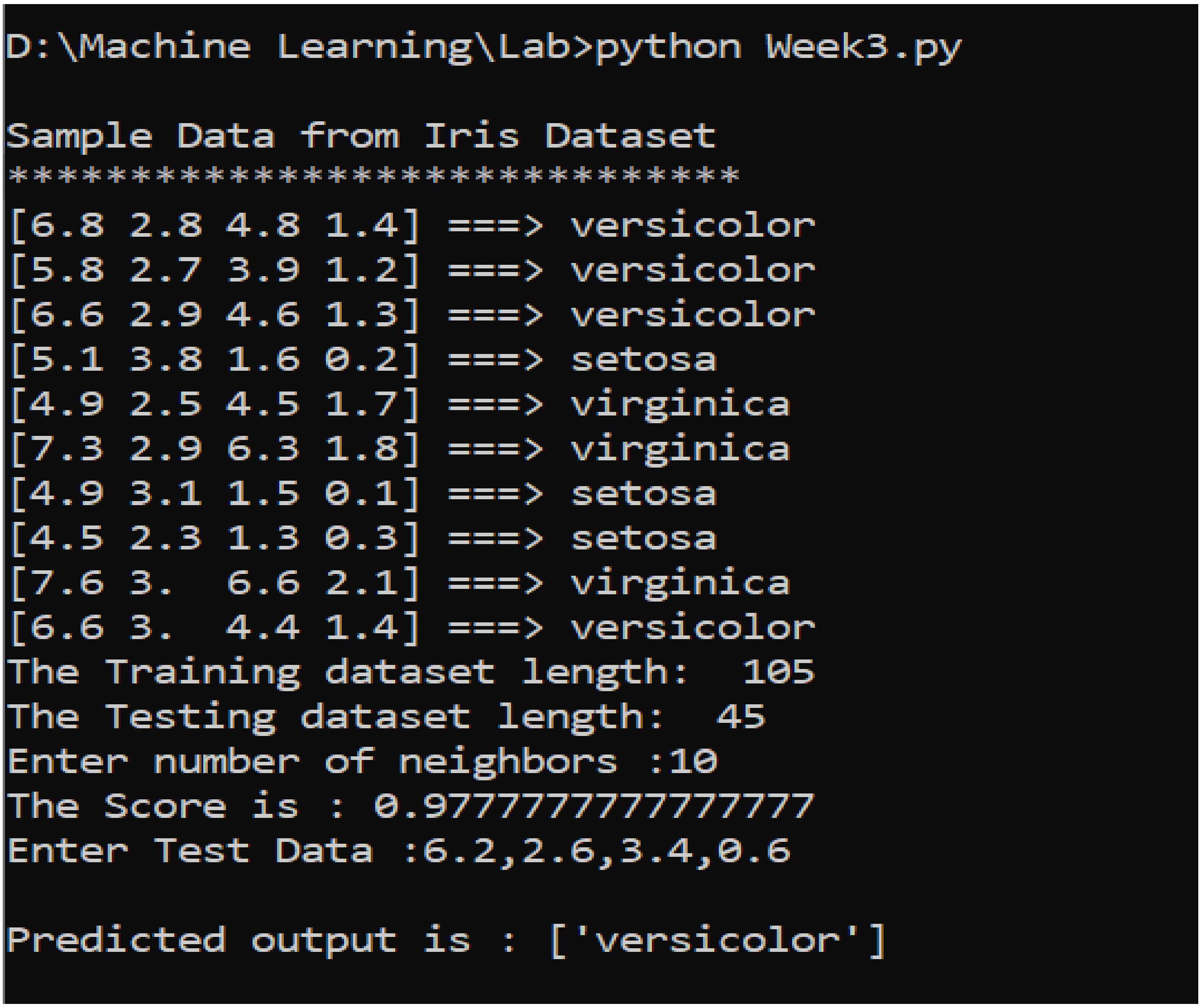


```
================================
Explanation:
=================================
===> To run this program you need to install the sklearn Module

===> Open Command propmt and then execute the following command to install sklearn Mo
dule

---> pip install scikit-learn

-----------------------------------------------

In this program, we are going to use iris dataset. And this dataset Split into traini
ng (70%) and test set(30%).
```

```
The iris dataset conatins the following features

---> sepal length (cm)
---> sepal width (cm)
---> petal length (cm)
---> petal width (cm)

The Sample data in iris dataset format is  [5.4 3.4 1.7 0.2]

Where  5.4 ---> sepal length (cm)
              3.4 ---> sepal width (cm)
              1.7 ---> petal length (cm)
              0.2 ---> petal width (cm)


================================
Source Code:
================================

'''

# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
import random

# Loading data
data_iris = load_iris()
# To get list of target names
label_target = data_iris.target_names
print()
print("Sample Data from Iris Dataset")
print("*"*30)
# to display the sample data from the iris dataset
for i in range(10):
       rn = random.randint(0,120)
       print(data_iris.data[rn],"===>",label_target[data_iris.target[rn]])

# Create feature and target arrays
X = data_iris.data
y = data_iris.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
                     X, y, test_size = 0.3, random_state=1)

print("The Training dataset length: ",len(X_train))
print("The Testing dataset length: ",len(X_test))
try:
       nn = int(input("Enter number of neighbors :"))
       knn = KNeighborsClassifier(nn)

       knn.fit(X_train, y_train)
```

```
        # to display the score
        print("The Score is :",knn.score(X_test, y_test))
        # To get test data from the user
        test_data = input("Enter Test Data:").split(",")
        for i in range(len(test_data)):
                test_data[i] = float(test_data[i])
        print()

        v = knn.predict([test_data])
        print("Predicted output is:",label_target[v])
except:
        print("Please supply valid input......")
```

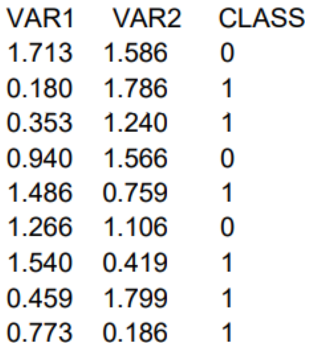
Output:



```
D:\Machine Learning\Lab>python Week3.py

Sample Data from Iris Dataset
******************************
[6.8 2.8 4.8 1.4] ===> versicolor
[5.8 2.7 3.9 1.2] ===> versicolor
[6.6 2.9 4.6 1.3] ===> versicolor
[5.1 3.8 1.6 0.2] ===> setosa
[4.9 2.5 4.5 1.7] ===> virginica
[7.3 2.9 6.3 1.8] ===> virginica
[4.9 3.1 1.5 0.1] ===> setosa
[4.5 2.3 1.3 0.3] ===> setosa
[7.6 3.  6.6 2.1] ===> virginica
[6.6 3.  4.4 1.4] ===> versicolor
The Training dataset length:  105
The Testing dataset length:  45
Enter number of neighbors :10
The Score is : 0.9777777777777777
Enter Test Data :6.2,2.6,3.4,0.6

Predicted output is : ['versicolor']
```

PROGRAM-4

4.Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3centroids)

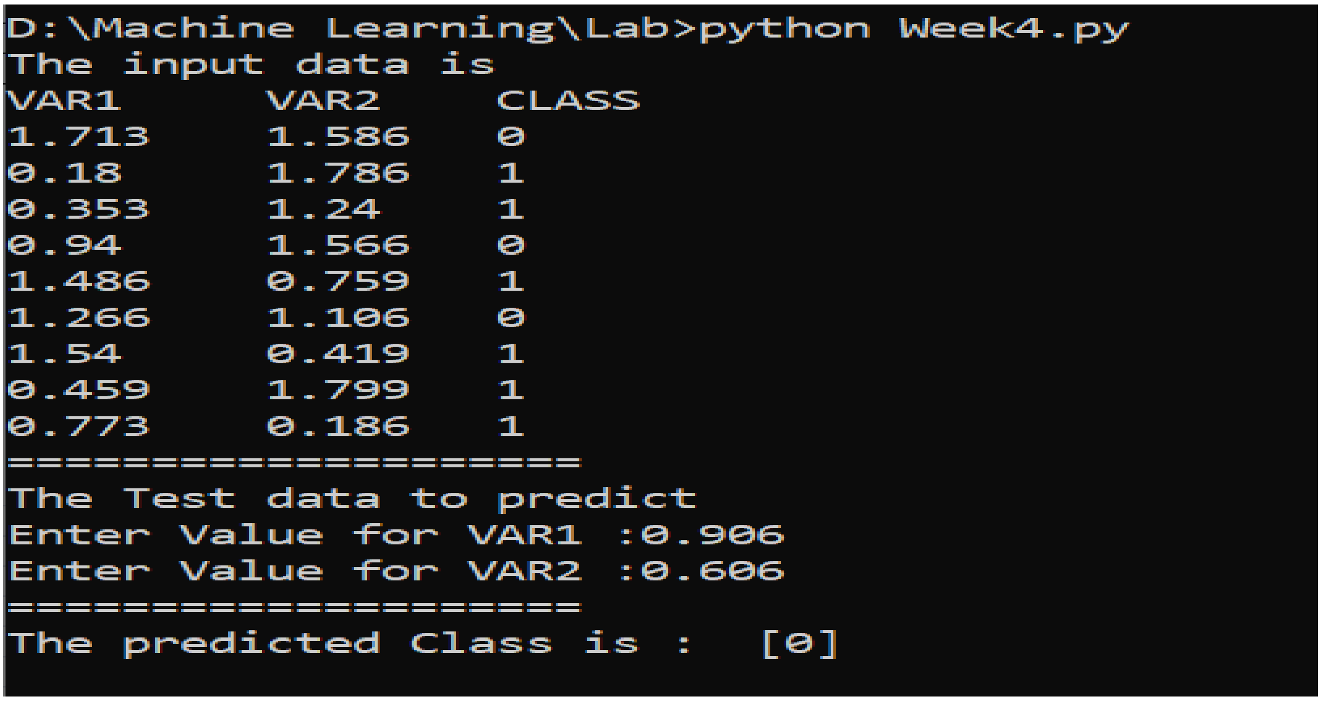


VAR1	VAR2	CLASS
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

```
=================================
Explanation:
=================================
===> To run this program you need to install the sklearn Module

===> Open Command propmt and then execute the following command to install sklearn Mo
dule

---> pip install scikit-learn

In this program, we are going to use the following data

VAR1 VAR2 CLASS
1.713 1.586 0
0.180 1.786 1
0.353 1.240 1
0.940 1.566 0
1.486 0.759 1
1.266 1.106 0
1.540 0.419 1
0.459 1.799 1
0.773 0.186 1

And, we need apply k-means clustering with 3 means (i.e., 3 centroids)


Finally, you need to predict the class for the VAR1=0.906 and VAR2=0.606

===============================
Source Code:
===============================
'''
from sklearn.cluster import KMeans
import numpy as np
```

```
X = np.array([[1.713,1.586], [0.180,1.786], [0.353,1.240],
[0.940,1.566], [1.486,0.759], [1.266,1.106],[1.540,0.419],[0.459,1.799],[0.773,0.186]
])
y=np.array([0,1,1,0,1,0,1,1,1])
kmeans = KMeans(n_clusters=3, random_state=0).fit(X,y)
print("The input data is ")
print("VAR1 \t VAR2 \t CLASS")
i=0
for val in X:
       print(val[0],"\t",val[1],"\t",y[i])
       i+=1
print("="*20)
# To get test data from the user
print("The Test data to predict ")
test_data = []
VAR1 = float(input("Enter Value for VAR1 :"))
VAR2 = float(input("Enter Value for VAR2 :"))
test_data.append(VAR1)
test_data.append(VAR2)
print("="*20)
print("The predicted Class is : ",kmeans.predict([test_data]))
```

Output: -



```
D:\Machine Learning\Lab>python Week4.py
The input data is
VAR1     VAR2     CLASS
1.713    1.586    0
0.18     1.786    1
0.353    1.24     1
0.94     1.566    0
1.486    0.759    1
1.266    1.106    0
1.54     0.419    1
0.459    1.799    1
0.773    0.186    1
====================
The Test data to predict
Enter Value for VAR1 :0.906
Enter Value for VAR2 :0.606
====================
The predicted Class is :   [0]
```

PROGRAM 5

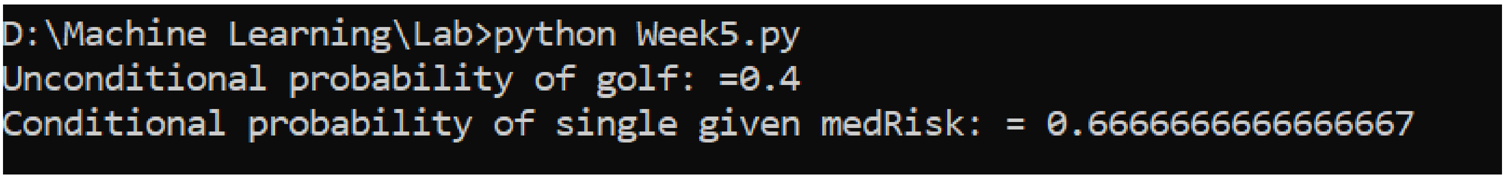
5.The Following Training Examples Map Descriptions of Individuals onto High, Medium and low Credit-Worthiness.

```
medium skiing design single twenties
no ->highRisk

high golf trading married forties yes
->lowRisk
low speedway transport married thirties
yes ->medRisk
medium football banking single thirties
yes ->lowRisk
high flying media married fifties yes -

>highRisk

low football security single twenties
no ->medRisk
medium golf media single thirties yes -
>medRisk
medium golf transport married forties
yes ->lowRisk
high skiing banking single thirties yes
->highRisk
low golf unemployed married forties yes
->highRisk
```

```
Input attributes are (from left to right) income, recreation, job, status, age-group,
home-owner. Find the unconditional probability of 'golf' and the conditional probabil
ity of 'single' given 'medRisk' in the dataset

=================================
Explanation:
=================================

In the given data set,

----> The total number of records are 10.

----> The number of records which contains 'golf' are 4.

----> Then, the Unconditional probability of golf:

                     = The number of records which contains 'golf' / total number of
records
                     = 4 / 10
                     = 0.4

******************************
```

```
To find the Conditional probability of single given medRisk,

---> S : single
---> MR : medRisk

---> By the definition of Baye's rule( conditional probability ), we have

       P(S | MR) = P(S ∩ MR) / P(MR)

       Based on the given problem statement,

    P(S ∩ MR) = The number of MedRisk with Single records / total number of Records
                       = 2 / 10 = 0.2
    and

    P(MR) = The number of records with MedRisk /total number of Records
                = 3 / 10 = 0.3

       Then, the Conditional probability of single given medRisk

       P(S | MR) = 0.2 / 0.3
                       = 0.66666

===============================
Source Code :
===============================
'''
total_Records=10
numGolfRecords=4
unConditionalprobGolf=numGolfRecords / total_Records
print("Unconditional probability of golf: ={}".format(unConditionalprobGolf))
#conditional probability of 'single' given 'medRisk'
numMedRiskSingle=2
numMedRisk=3
probMedRiskSingle=numMedRiskSingle/total_Records
probMedRisk=numMedRisk/total_Records
conditionalProb=(probMedRiskSingle/probMedRisk)
print("Conditional probability of single given medRisk: ={}".format(conditionalProb))
```

Output:



```
D:\Machine Learning\Lab>python Week5.py
Unconditional probability of golf: =0.4
Conditional probability of single given medRisk: = 0.6666666666666667
```

PROGRAM 6:

6. Implement linear regression using python

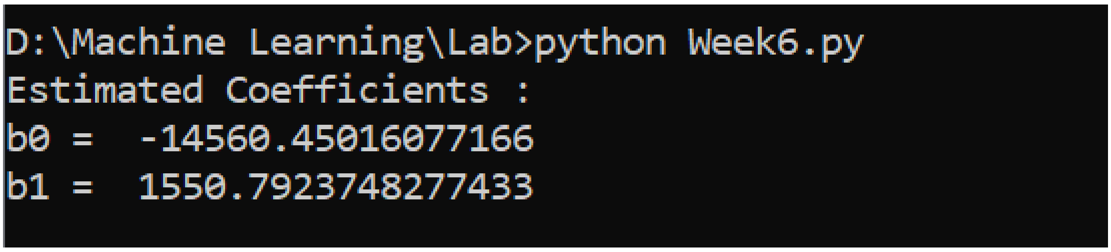
```
================================
Explanation:
=================================

===> To run this program you need to install the pandas Module

---> pandas Module is used to read csv files

===> To install, Open Command propmt and then execute the following command

---> pip install pandas


And, then you need to install the matplotlib Module

---> matplotlib Module is used to plot the graphs

===> To install, Open Command propmt and then execute the following command

---> pip install matplotlib

Finally, you need to create dataset called "Age_Income.csv" file.

===============================
Source Code:
===============================

'''
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# To read data from Age_Income.csv file
dataFrame = pd.read_csv('Age_Income.csv')
# To place data in to age and income vectors
age = dataFrame['Age']
income = dataFrame['Income']
# number of points
num = np.size(age)
# To find the mean of age and income vector
mean_age = np.mean(age)
mean_income = np.mean(income)

# calculating cross-deviation and deviation about age
CD_ageincome = np.sum(income*age) - num*mean_income*mean_age
CD_ageage = np.sum(age*age) - num*mean_age*mean_age

# calculating regression coefficients
b1 = CD_ageincome / CD_ageage
```

```
b0 = mean_income - b1*mean_age
# to display coefficients
print("Estimated Coefficients :")
print("b0 = ",b0,"\nb1 = ",b1)
# To plot the actual points as scatter plot
plt.scatter(age, income, color = "b",marker = "o")
# TO predict response vector
response_Vec = b0 + b1*age
# To plot the regression line
plt.plot(age, response_Vec, color = "r")
# Placing labels
plt.xlabel('Age')
plt.ylabel('Income')
# To display plot
plt.show(
```

Age_Income.csv(Data Set)

```
Age, Income
25,25000
23,22000
24,26000
28,29000
34,38600
32,36500
42,41000
55,81000
45,47500
```

Output:



```
D:\Machine Learning\Lab>python Week6.py
Estimated Coefficients :
b0 =  -14560.45016077166
b1 =  1550.7923748277433
```

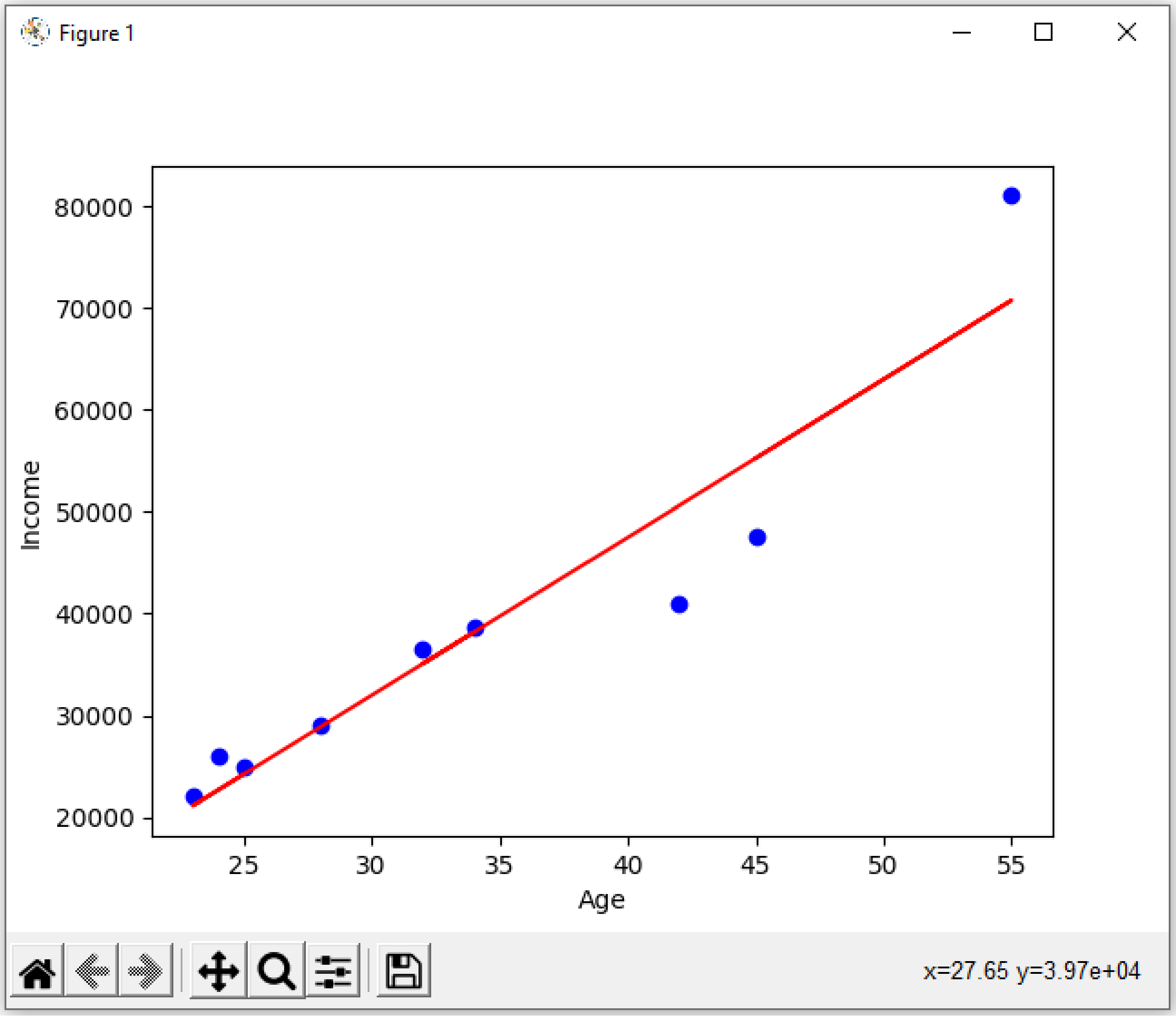


Figure 1
80000
70000
60000
50000
40000
30000
20000
Income
25
30
35
40
45
50
55
Age
x=27.65 y=3.97e+04

PROGRAM 7:

7.Implement linear regression using python

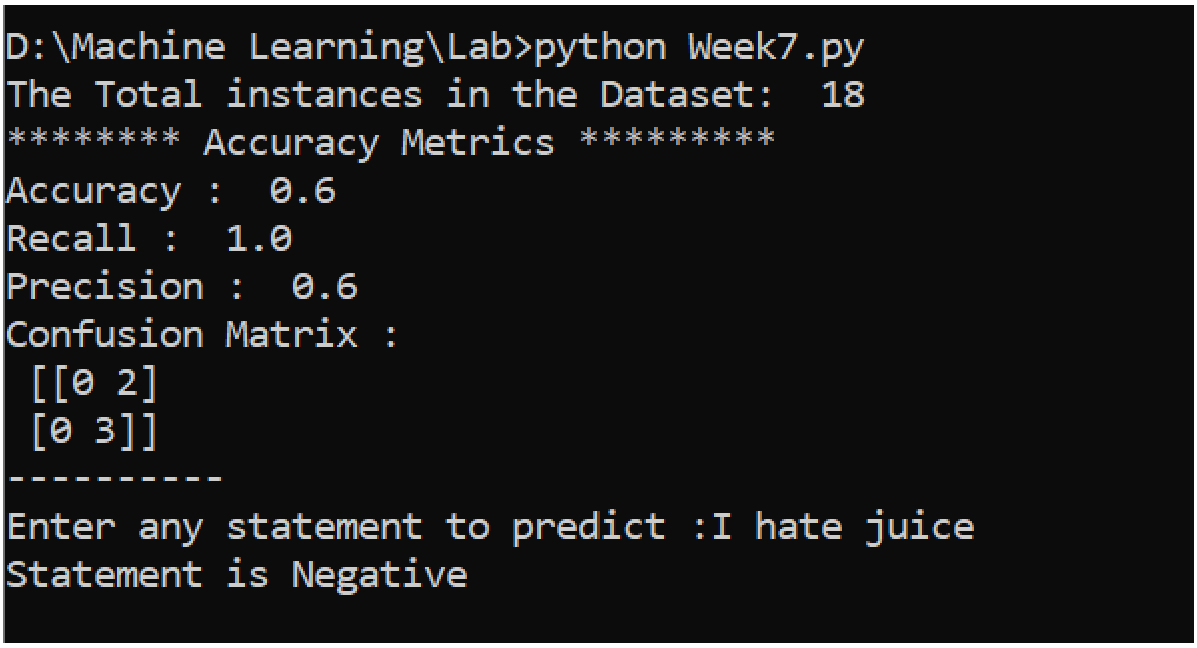
```
=================================
Explanation:
=================================

===> To run this program you need to install the pandas Module

---> pandas Module is used to read csv files

===> To install, Open Command propmt and then execute the following command

---> pip install pandas


And, then you need to install the sklearn Module

===> Open Command propmt and then execute the following command to install sklearn Mo
dule

---> pip install scikit-learn

Finally, you need to create dataset called "Statements_data.csv" file.

===============================
Source Code :
===============================

'''
import pandas as pd
from sklearn. model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall
_score
msglbl_data = pd.read_csv('Statements_data.csv', names=['Message', 'Label'])
print("The Total instances in the Dataset: ", msglbl_data.shape[0])
msglbl_data['labelnum'] = msglbl_data.Label.map({'pos': 1, 'neg': 0})
# place the data in X and Y Vectors
X = msglbl_data["Message"]
Y = msglbl_data.labelnum
# to split the data into train se and test set
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y)
count_vect = CountVectorizer()
Xtrain_dims = count_vect.fit_transform(Xtrain)
Xtest_dims = count_vect.transform(Xtest)
df = pd.DataFrame(Xtrain_dims.toarray(),columns=count_vect.get_feature_names_out())
```

```
clf = MultinomialNB()
# to fit the train data into model
clf.fit(Xtrain_dims, Ytrain)
# to predict the test data
prediction = clf.predict(Xtest_dims)
print('******** Accuracy Metrics *********')
print('Accuracy : ', accuracy_score(Ytest, prediction))
print('Recall : ', recall_score(Ytest, prediction))
print('Precision : ',precision_score(Ytest, prediction))
print('Confusion Matrix : \n', confusion_matrix(Ytest, prediction))
print(10*"-")
# to predict the input statement
test_stmt = [input("Enter any statement to predict :")]
test_dims = count_vect.transform(test_stmt)
pred = clf.predict(test_dims)
for stmt,lbl in zip(test_stmt,pred):
       if lbl == 1:
              print("Statement is Positive")
       else:
              print("Statement is Negative")
```

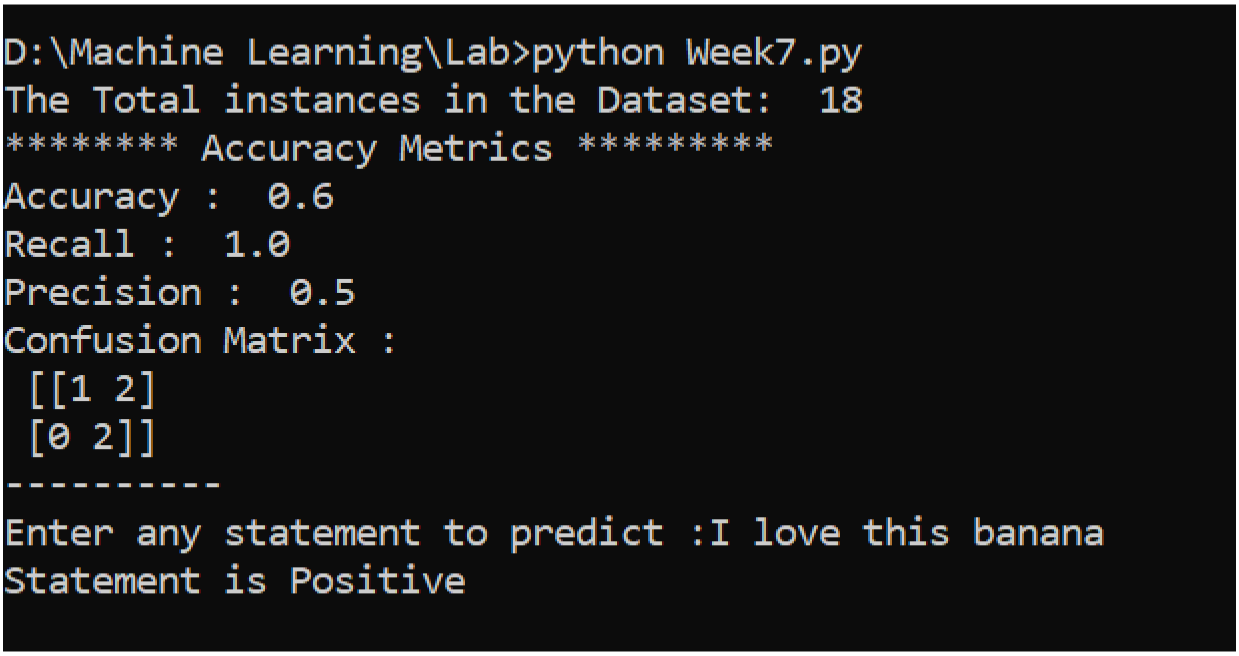
Statements_data.csv(Data Set)

```
This is very good place,pos
I like this biryani,pos
I feel very happy,pos
This is my best work,pos
I do not like this restaurant,neg
I am tired of this stuff,neg
I can't deal with this,neg
What an idea it is,pos
My place is horrible,neg
This is an awesome place,pos
I do not like the taste of this juice,neg
I love to sing,pos
I am sick and tired,neg
I love to dance,pos
What a great holiday,pos
That is a bad locality to stay,neg
We will have good fun tomorrow,pos
I hate this food,neg
```

Output:



```
D:\Machine Learning\Lab>python Week7.py
The Total instances in the Dataset:  18
******** Accuracy Metrics *********
Accuracy :  0.6
Recall :  1.0
Precision :  0.6
Confusion Matrix :
 [[0 2]
 [0 3]]
----------
Enter any statement to predict :I hate juice
Statement is Negative
```



```
D:\Machine Learning\Lab>python Week7.py
The Total instances in the Dataset:  18
******** Accuracy Metrics *********
Accuracy :  0.6
Recall :  1.0
Precision :  0.5
Confusion Matrix :
 [[1 2]
 [0 2]]
----------
Enter any statement to predict :I love this banana
Statement is Positive
```

PROGRAM 8

8.Implement an algorithm to demonstrate the significance of genetic algorithm

Source Code:

```
 import numpy
# Parameter initialization
genes = 2
chromosomes = 10
mattingPoolSize = 6
offspringSize = chromosomes - mattingPoolSize
lb = -5
ub = 5
populationSize = (chromosomes, genes)
generations = 3
#Population initialization
population = numpy.random.uniform(lb, ub, populationSize)
for generation in range(generations):
print(("Generation:", generation+1))
fitness = numpy.sum(population*population, axis=1)
print("\npopulation")
print(population)
print("\nfitness calcuation")
print(fitness)
# Following statement will create an empty two dimensional array to store
parents
parents = numpy.empty((mattingPoolSize, population.shape[1]))
# A loop to extract one parent in each iteration
for p in range(mattingPoolSize):
# Finding index of fittest chromosome in the population
fittestIndex = numpy.where(fitness == numpy.max(fitness))
# Extracting index of fittest chromosome
fittestIndex = fittestIndex[0][0]
# Copying fittest chromosome into parents array
parents[p, :] = population[fittestIndex, :]
# Changing fitness of fittest chromosome to avoid reselection of that
chromosome
fitness[fittestIndex] = -1
print("\nParents:")
print(parents)
```

```
# Following statement will create an empty two dimensional array to store
offspring
offspring = numpy.empty((offspringSize, population.shape[1]))
for k in range(offspringSize):
#Determining the crossover point
crossoverPoint = numpy.random.randint(0,genes)
# Index of the first parent.
parent1Index = k%parents.shape[0]
# Index of the second.
parent2Index = (k+1)%parents.shape[0]
# Extracting second half of the offspring
offspring[k, crossoverPoint:] = parents[parent2Index, crossoverPoint:]
print("\nOffspring after crossover:")
print(offspring)
# Implementation of random initialization mutation.
for index in range(offspring.shape[0]):
randomIndex = numpy.random.randint(1,genes)
randomValue = numpy.random.uniform(lb, ub, 1)
offspring [index, randomIndex] = offspring [index, randomIndex] +
randomValue
print("\n Offspring after Mutation")
print(offspring)
population[0:parents.shape[0], :] = parents
population[parents.shape[0]:, :] = offspring
print("\nNew Population for next generation:")
print(population)
fitness = numpy.sum(population*population, axis=1)
fittestIndex = numpy.where(fitness == numpy.max(fitness))
# Extracting index of fittest chromosome
fittestIndex = fittestIndex[0][0]
# Getting Best chromosome
fittestInd = population[fittestIndex, :]
bestFitness = fitness[fittestIndex]
print("\nBest Individual:")
print(fittestInd)
print("\nBest Individual's Fitness:")

print(bestFitness)
```

Output: -

```
('Generation:', 1)
population
[[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[ 0.19027998 -1.92076011]
```

```
[-4.51396933 -1.89463461]
[ 0.79755849 3.43265172]
[-1.54352966 3.94293134]
[ 2.63471426 -3.51067942]
[-1.02184282 -4.64715438]
[ 1.49179561 -2.11644882]
[ 1.46421881 4.06826713]]

fitness calcuation
[27.834257 26.27830909 3.72552586 23.96555941 12.41919734 17.92919137
19.26658925 22.64020659 6.70480973 18.69473415]

Parents:
[[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[-4.51396933 -1.89463461]
[-1.02184282 -4.64715438]
[ 2.63471426 -3.51067942]
[ 1.46421881 4.06826713]]

Offspring after crossover:
[[ 2.68516631 -4.36671398]
[ 2.68516631 -1.89463461]
[-1.02184282 -4.64715438]
[-1.02184282 -3.51067942]]

Offspring after Mutation
[[ 2.68516631 -4.32266198]
[ 2.68516631 2.13424051]
[-1.02184282 -7.79018038]
[-1.02184282 -5.94231314]]

New Population for next generation:
[[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[-4.51396933 -1.89463461]
[-1.02184282 -4.64715438]
[ 2.63471426 -3.51067942]
[ 1.46421881 4.06826713]
[ 2.68516631 -4.32266198]
[ 2.68516631 2.13424051]
[-1.02184282 -7.79018038]
[-1.02184282 -5.94231314]]

('Generation:', 2)

population
[[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[-4.51396933 -1.89463461]
```

```
[-1.02184282 -4.64715438]
[ 2.63471426 -3.51067942]
[ 1.46421881 4.06826713]
[ 2.68516631 -4.32266198]
[ 2.68516631 2.13424051]
[-1.02184282 -7.79018038]
[-1.02184282 -5.94231314]]

fitness calcuation
[27.834257 26.27830909 23.96555941 22.64020659 19.26658925 18.69473415
25.89552471 11.76510065 61.73107316 36.35524821]

Parents:
[[-1.02184282 -7.79018038]
[-1.02184282 -5.94231314]
[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[ 2.68516631 -4.32266198]
[-4.51396933 -1.89463461]]

Offspring after crossover:
[[-1.02184282 -5.94231314]
[-1.02184282 2.17457345]
[ 2.68516631 -4.36671398]
[ 2.68516631 -4.32266198]]

Offspring after Mutation
[[ -1.02184282 -10.47230381]
[ -1.02184282 -0.25526352]
[ 2.68516631 0.08928766]
[ 2.68516631 -4.96076826]]

New Population for next generation:
[[ -1.02184282 -7.79018038]
[ -1.02184282 -5.94231314]
[ 4.80681675 2.17457345]
[ 2.68516631 -4.36671398]
[ 2.68516631 -4.32266198]
[ -4.51396933 -1.89463461]
[ -1.02184282 -10.47230381]
[ -1.02184282 -0.25526352]
[ 2.68516631 0.08928766]
[ 2.68516631 -4.96076826]]
```

Program 9:

9.Implement the finite words classification system using Backpropagation algorithm

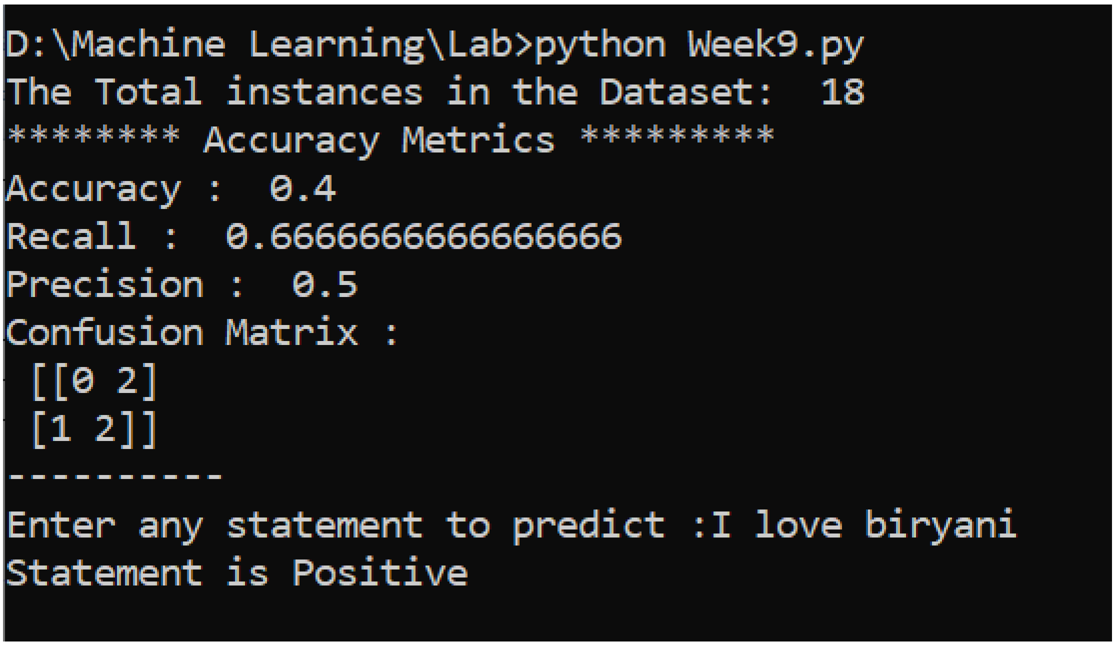
```
=================================
Explanation:
=================================

===> To run this program you need to install the pandas Module

---> pandas Module is used to read csv files

===> To install, Open Command propmt and then execute the following command

---> pip install pandas


And, then you need to install the sklearn Module

===> Open Command propmt and then execute the following command to install sklearn Module

---> pip install scikit-learn

===> Open Command propmt and then execute the following command to install sklearn-neural network Module

---> pip install scikit-neural network


Finally, you need to create dataset called "Statements_data.csv" file.

===============================
Source Code :
===============================

'''
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
msglbl_data = pd.read_csv('Statements_data.csv', names=['Message', 'Label'])
print("The Total instances in the Dataset: ", msglbl_data.shape[0])
msglbl_data['labelnum'] = msglbl_data.Label.map({'pos': 1, 'neg': 0})
# place the data in X and Y Vectors
X = msglbl_data["Message"]
Y = msglbl_data.labelnum
```

```
# to split the data into train se and test set
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y)

count_vect = CountVectorizer()
Xtrain_dims = count_vect.fit_transform(Xtrain)
Xtest_dims = count_vect.transform(Xtest)
df = pd.DataFrame(Xtrain_dims.toarray(),columns=count_vect.get_feature_names_out())
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), random_state=1)
# to fit the train data into model
clf.fit(Xtrain_dims, Ytrain)
# to predict the test data
prediction = clf.predict(Xtest_dims)
print('******** Accuracy Metrics *********')
print('Accuracy : ', accuracy_score(Ytest, prediction))
print('Recall : ', recall_score(Ytest, prediction))
print('Precision : ',precision_score(Ytest, prediction))
print('Confusion Matrix : \n', confusion_matrix(Ytest, prediction))
print(10*"-")
# to predict the input statement
test_stmt = [input("Enter any statement to predict :")]
test_dims = count_vect.transform(test_stmt)
pred = clf.predict(test_dims)
for stmt,lbl in zip(test_stmt,pred):
        if lbl == 1:
                print("Statement is Positive")
        else:
                print("Statement is Negative")
```

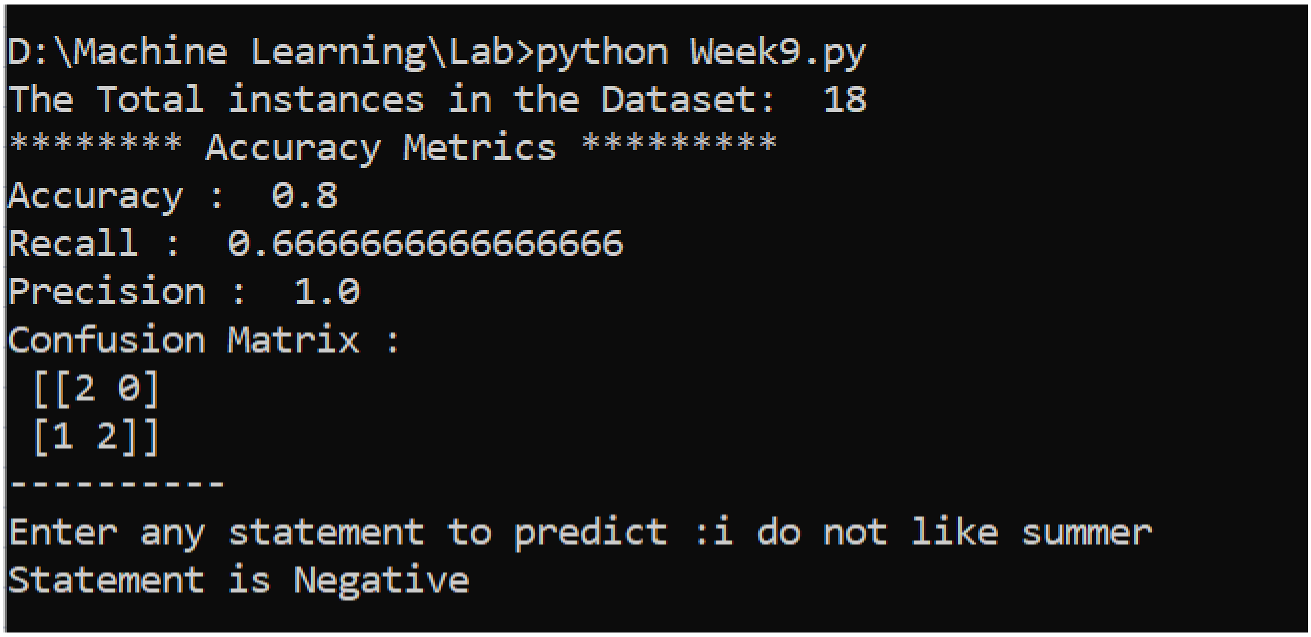
Statements_data.csv(Data Set)

```
This is very good place,pos
I like this biryani,pos
I feel very happy,pos
This is my best work,pos
I do not like this restaurant,neg
I am tired of this stuff,neg
I can't deal with this,neg
What an idea it is,pos
My place is horrible,neg
This is an awesome place,pos
I do not like the taste of this juice,neg
I love to sing,pos
I am sick and tired,neg
I love to dance,pos
What a great holiday,pos
That is a bad locality to stay,neg
We will have good fun tomorrow,pos
I hate this food,neg
```

Output:



```
D:\Machine Learning\Lab>python Week9.py
The Total instances in the Dataset:  18
******** Accuracy Metrics *********
Accuracy :  0.4
Recall :  0.6666666666666666
Precision :  0.5
Confusion Matrix :
 [[0 2]
 [1 2]]
----------
Enter any statement to predict :I love biryani
Statement is Positive
```



```
D:\Machine Learning\Lab>python Week9.py
The Total instances in the Dataset:  18
******** Accuracy Metrics *********
Accuracy :  0.8
Recall :  0.6666666666666666
Precision :  1.0
Confusion Matrix :
 [[2 0]
 [1 2]]
----------
Enter any statement to predict :i do not like summer
Statement is Negative
```