20MCA241 DATA SCI ENCE LAB

Lab Report SubmittedBy

ALEX T HOMA S

Reg. No.:AJ C20MCA-2009

In Partial fulfillment for the Award of the Degree Of

MA ST ER OF COMPUTER APPLICATIONS (2 Y ear)

(MCA )

APJ A BDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL J YOTHI COLLEGE OF ENGINEERING

KANJ IRAPPALLY

[Affiliated to APJ A bdul K alam Technological University, Kerala. Approved by AICTE,

Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020- 2022



DEPA RT MENT OF COMPUTER APPLICATIONS

AMAL J YOTHI COLLEGE OF ENGINEERING

KANJ IRAPPALLY

CERTIFICATE

This is to certify that the Lab report,“20MCA241 DATA SCIENCE LAB**”**

is the bonafide work of A LEX THOMA S (Reg.No:A J C20MCA - 2009) in

partial fulfillment of the requirements for the award of the Degree of Master

of Computer A ppl i cati ons under APJ A bdul K alam T echnol ogi cal

University during the year 2021-22.

Ms. Meera Rose Mathew

Lab In-Charge



CONT ENT

Sl .No

Content

Date

Page

No

1

1

2

Perform all matrix operation using python

Program to perform SVD using python

24/11/2021

01/12/2021

4

5

3

Program to i mpl ement k-NN Classification

using any standard dataset available in the

public domain and find the accuracy of the

algorithm using in build function

01/12/2021

4

5

Program to i mpl ement k-NN Classification

using any random dataset without using in-

build functions

6

8

01/12/2021

08/12/2021

Program to implement Naïve Bayes Algorithm

using any standard dataset available in the

public domain and find the accuracy of the

algorithm

6

7

Program to i mpl ement linear and multiple

r egr essi on techniques using any standard

dataset available in the public domain

Program to i mpl ement L inear and Multiple

r egr essi on techniques using any standard

dataset available in public domain and evaluate

its performance

11

12

08/01/2022

08/01/2022

8

9

Program to i mpl ement L inear and Multiple

r egr essi on techniques using cars dataset

available in public domain and evaluate its

performance

14

15

17

15/01/2022

15/01/2022

22/12/2021

Program to i mpl ement multiple linear

r egr essi on techniques using Boston dataset

available in the public domain and evaluate its

performance and plotting graph

10

Program to implement decision tree using any

standard dataset available in the public domain

and find the accuracy of the algorithm

11

12

Program to i mpl ement K - Means clustering

technique using any standard dataset available

in the public domain

Program to i mpl ement K - Means clustering

technique using any standard dataset available

in the public domain

20

23

05/01/2022

05/01/2022

13

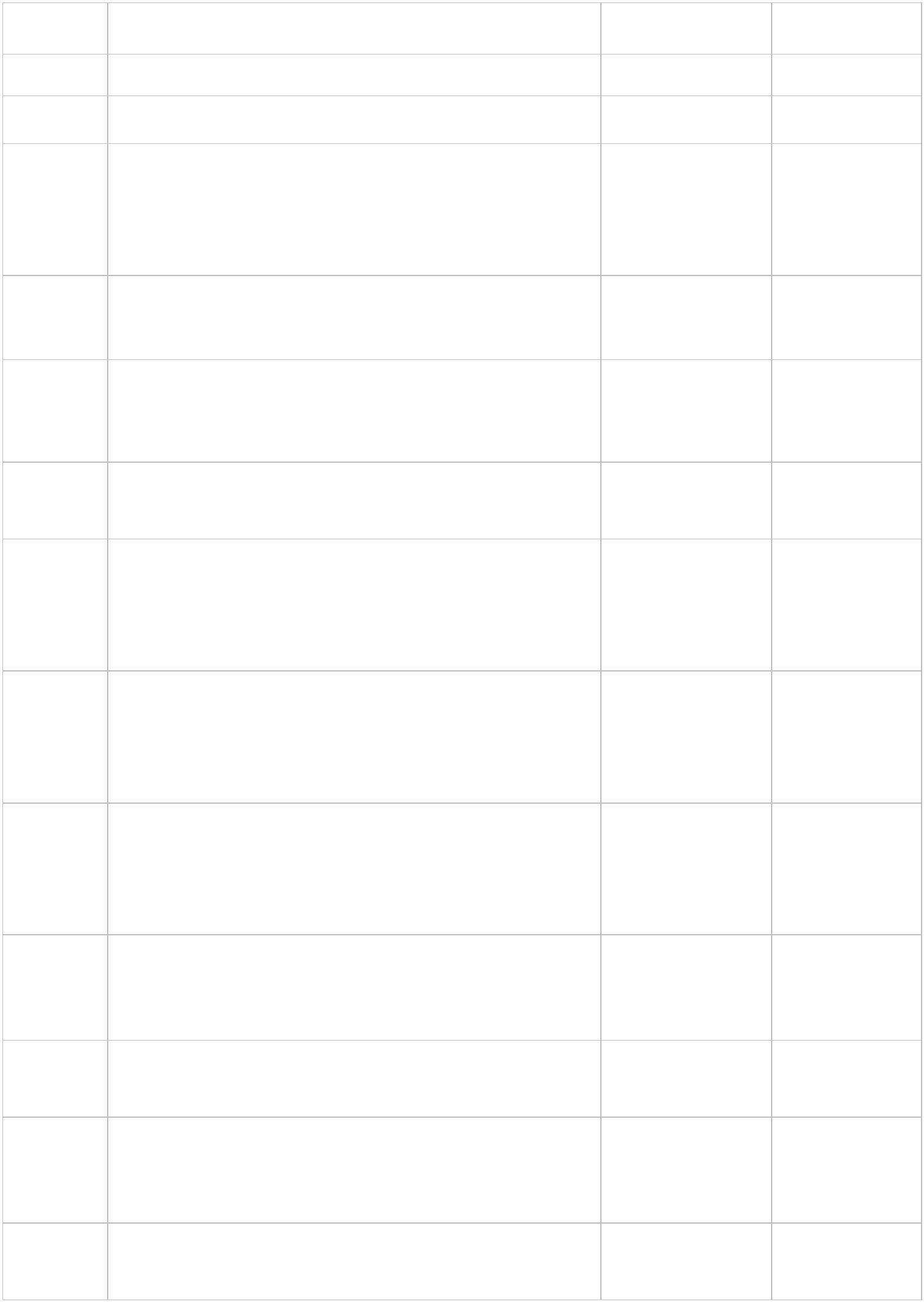
Programs on convolutional neural network to

classify images from any standard dataset in

the public domain

25

02/02/2022



14

15

Program to implement a simple web crawler

using python

30

32

16/02/2022

Program to i mpl ement a simple web cr awl er 16/02/2022

using python

16

17

Program to implement scrap of any website

16/02/2022

34

36

Program for Natural L anguage Processing 16/02/2022

which performs n-grams

18

19

Program for Natural L anguage Processing 16/02/2022

which perf orms n- gr ams (Using in built

functions)

37

38

Program for Natural L anguage Processing 16/02/2022

which performs speech tagging

Program for Natural L anguage Processing

which performs Chunking

20

23/02/2022

23/02/2022

40

Program for Natural L anguage Processing

which performs Chunking

42

21



1

PROGRAM NO : 01

Date: 24/ 11/ 2021

AIM : Perform all matrix operation using python.

PROGRAM CODE

import numpy as np

import random

def Pri ntMatri x(matri x\_ i n):

for x in range(0, matrix\_ in.shape[0]):

for y in range(0, matrix\_in.shape[1]):

print("%d \t" % (matrix\_in[x][y]), end='')

if (y % 3>1):

pri nt(" \n")

def Fil lMatrix(matrix\_ in):

for x in range(0, matrix\_ in.shape[0]):

for y in range(0, matrix\_in.shape[1]):

matrix\_ in[x][y] = random.randrange(2, 10) +2

matrix1 = np.ndarray((3,3))

matrix2 = np.ndarray((3,3))

Fi l l Matri x(matri x1)

Fi l l Matri x(matri x2)

add\_ resul ts = np.add(matri x1,matrix2)

sub\_ resul ts=np. subtract( matri x 1, matri x 2)

mul t\_ resul ts=np.mul ti pl y( matri x 1,matri x 2)

di v\_ resul ts=np.di vi de(matri x 1,matri x 2)

dot\_ resul ts=np.dot( matri x 1,matri x 2)

sqrt1\_ resul ts=np. sqrt( matri x 1)

sqrt2\_ resul ts=np. sqrt( matri x 2)

trans\_ resul ts=add\_ resul ts.T

pri nt( "Matri x 1:" )

Pri ntMatri x ( matri x 1)

pri nt( "Matri x 2:" )

Pri ntMatri x ( matri x 2)

pri nt("A ddi ng")

20MCA241 Data Science Lab

Dept. of Computer Applications



2

Pri ntMatri x ( add\_ resul ts)

print("Subtraction")

Pri ntMatri x ( sub\_ resul ts)

pri nt("Mul ti pl i cati on")

Pri ntMatri x ( mul t\_ resul ts)

print("Dot Operation")

Pri ntMatri x ( dot\_ resul ts)

pri nt("squareroot Operati on")

pri nt( " matri x 1")

Pri ntMatri x ( sqrt1\_ resul ts)

pri nt( " matri x 2")

Pri ntMatri x ( sqrt2\_ resul ts)

print("Transpose")

Pri ntMatri x ( trans\_ resul ts)

OUT PUT

Matri x 1:

4

6

9

4

11

6

4

11

5

Matri x 2:

8

10

10

8

11

8

9

11

10

A dding

12

17

17

14

21

14

15

13

22

Subtraction

-4

-5

1

-6

-5

0

1

-2

-5

20MCA241 Data Science Lab

Dept. of Computer Applications



3

Mul ti pl i cati on

32

66

72

40

36

110

48

121 50

Dot Operation

164 197 182

140 162 152

233 244 228

Squareroot Operati on

matri x 1

2

2

3

2

2

3

3

2

2

matri x 2

2

3

2

3

3

3

3

2

3

Transpose

12

14

21

17

13

14

17

22

15

Process finished with exit code

20MCA241 Data Science Lab

Dept. of Computer Applications



4

PROGRAM NO : 02

Date : 01/12/2021

AIM: Program to perform SVD (Singular value Decomposition) using Python.

PROGRAM CODE

from scipy. linalg import svd

from numpy import array

A=([[1,2,5], [2,0,1], [1,4,4]])

print(A )

X, B, T=svd(A)

print("decomposition")

print(X )

pri nt( " i nverse" )

print(B)

print("transpose")

pri nt( T)

OUT PUT

[[1, 2, 5], [2, 0, 1], [1, 4, 4]]

decomposition

[[- 0.68168247 - 0.26872313 - 0.68051223]

[- 0.15885378 - 0.85356116 0.49618427]

[ - 0.71419499 0.44634205 0.53916999]]

i nv erse

[7.87492 2.01650097 1.38540929]

transpose

[[- 0.21760031 - 0.53589686 - 0.81576017]

[ - 0.75849376 0.61885512 - 0.20421939]

[ 0.61427789 0.5743108 - 0. 54113749] ]

Process finished with exit code

20MCA241 Data Science Lab

Dept. of Computer Applications



5

PROGRAM NO : 03

Date : 01/12/2021

AIM :Program to implement k-NN Classification using any standard dataset available in the

public domain and find the accuracy of the algorithm using in build function.

PROGRAM CODE

from sklearn.neighbors import K NeighborsClassifier

from sklearn.model\_ selection import train\_ test\_ split

from sklearn.datasets import load\_ iris

from sklearn.metrics import accuracy\_ score

iris =load\_iris()

x=i ri s.data

y=i ri s.target

x\_ trai n,x\_ test,y\_ trai n,y\_ test=trai n\_ test\_ spl i t(x,y,test\_ si ze=0.2,random\_ state=42)

knn=K Nei ghborsCl assi f i er(n\_ nei ghbors=7)

knn.f i t(x\_ trai n,y\_ trai n)

pri nt( knn.predi ct( x \_ test) )

V =knn.predi ct(x\_ test)

result=accuracy\_ score (y\_ test, V )

print ("accuracy:", result)

OUT PUT

[102110122120000121120202222200]

accuracy: 0.9666666666666667

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



6

PROGRAM NO : 04

Date : 01/12/2021

AIM : Program to implement k-NN Classification using any random dataset without using in-

build functions.

PROGRAM CODE

from math import sqrt

def euclidean\_ distance(row1, row2):

distance =0.0

for i in range(len(row1) - 1):

distance += (row1[i] - row2[ i ] ) \*\* 2

return sqrt(di stance)

#Locate the most similar neighbors

def get\_ neighbors(train, test\_ row, num\_ neighbors):

distances =list()

for train\_row in train:

dist =euclidean\_ distance(test\_ row, train\_ row)

distances.append((train\_ row, dist))

di stances.sort(key=l ambda tup: tup[ 1] )

neighbors =list()

for i i n range(num\_ nei ghbors):

neighbors.append(distances[i][0])

return nei ghbors

#Make a classification prediction with neighbors

def predict\_ classification(train, test\_ row, num\_ neighbors):

neighbors = get\_ neighbors(train, test\_ row, num\_ neighbors)

output\_ values =[row[- 1] for row in neighbors]

prediction = max(set(output\_ values), key=output\_ values.count)

return predi cti on

#Test distance function

dataset =[[2.781, 2.550,0],

[1.465, 2.326,3],

[3.398, 4.429,5],

20MCA241 Data Science Lab

Dept. of Computer Applications



7

[1.388, 1.857,11],

[3.064, 3.393,3],

[7.624, 2.235,4],

[5.338, 2.775,8]]

prediction =predict\_ classification(dataset, dataset[0], 3)

print('Expected %d, Got %d.' % (dataset[0][-1], prediction))

OUT PUT

Expected 2, Got 3.

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



8

PROGRAM NO : 05

Date : 08/12/2021

AIM : Program to implement Naïve Bayes Algorithm using any standard dataset available in

the public domain and find the accuracy of the algorithm.

PROGRAM CODE

import pandas as pd

dataset = pd.read\_ csv('Social\_ Network\_ A ds.csv')

x =dataset.iloc[:, [2,3]].values

y =dataset.iloc[:,- 1].values

from sklearn.model\_ selection import train\_ test\_ split

x\_train, x\_test, y\_ trai n, y\_test =train\_test\_split(x, y, test\_ size=0.2, random\_ state=10)

f rom sklearn.preprocessi ng import StandardScaler

sc = StandardScaler()

x\_ train = sc.fit\_ transform(x\_ train)

x\_ test = sc.transform(x\_ test)

from sklearn.naive\_ bayes import GaussianNB

gnb = GaussianNB()

gnb.f i t(x\_ trai n, y\_ trai n)

y\_ pred = gnb.predict(x\_ test)

pri nt( y\_ pred)

from sklearn import metrics

print("A ccuracy", metrics.accuracy\_ score(y\_ test, y\_ pred) \* 100)

import numpy as nm

import matplotlib.pyplot as mtp

from matplotlib.colors import ListedColormap

x\_ set, y\_set =x\_train, y\_ train

X 1, X 2 =nm.meshgrid(nm.arange(start =x\_set[:, 0].min() - 1, stop =x\_set[:, 0].max() +1, step =

0.01),

nm.arange(start =x\_ set[:, 1].min() - 1, stop =x\_set[:, 1].max() +1, step =0.01))

mtp.contourf(X 1, X 2, gnb.predict(nm.array([X 1.ravel(), X 2.ravel()]).T).reshape(X 1.shape),

alpha =0.75, cmap =ListedColormap(('purple', 'green')))

mtp.xlim(X 1.min(), X 1.max())

mtp.ylim(X 2.min(), X 2.max())

20MCA241 Data Science Lab

Dept. of Computer Applications



9

fori, j i n enumerate(nm.uni que(y\_ set)):

mtp.scatter(x\_set[y\_set ==j, 0], x\_set[y\_set ==j, 1],

c = ListedColormap(('purple', 'green'))(i), label =j)

mtp.title('Naive Bayes (Training set)')

mtp.xl abel ('A ge')

mtp.yl abel ('Esti mated Sal ary')

mtp. l egend( )

mtp.show()

x\_ set, y\_set =x\_test, y\_ test

X 1, X 2 =nm.meshgrid(nm.arange(start =x\_set[:, 0].min() - 1, stop =x\_set[:, 0].max() +1, step =

0.01),

nm.arange(start =x\_ set[:, 1].min() - 1, stop =x\_set[:, 1].max() +1, step =0.01))

mtp.contourf(X 1, X 2, gnb.predict(nm.array([X 1.ravel(), X 2.ravel()]).T).reshape(X 1.shape),

alpha =0.75, cmap =ListedColormap(('purple', 'green')))

mtp.xlim(X 1.min(), X 1.max())

mtp.ylim(X 2.min(), X 2.max())

fori, j i n enumerate(nm.uni que(y\_ set)):

mtp.scatter(x\_set[y\_set ==j, 0], x\_set[y\_set ==j, 1],

c = ListedColormap(('purple', 'green'))(i), label =j)

mtp.title('Naive Bayes (test set)')

mtp.xl abel ('A ge')

mtp.yl abel ('Esti mated Sal ary')

mtp. l egend( )

mtp.show()

20MCA241 Data Science Lab

Dept. of Computer Applications



10

OUT PUT

C: / Users/ aj cemca/ Py charmProj ects/ A rav i nd/ nai v e.py

[0011010100001110000100011001100001101

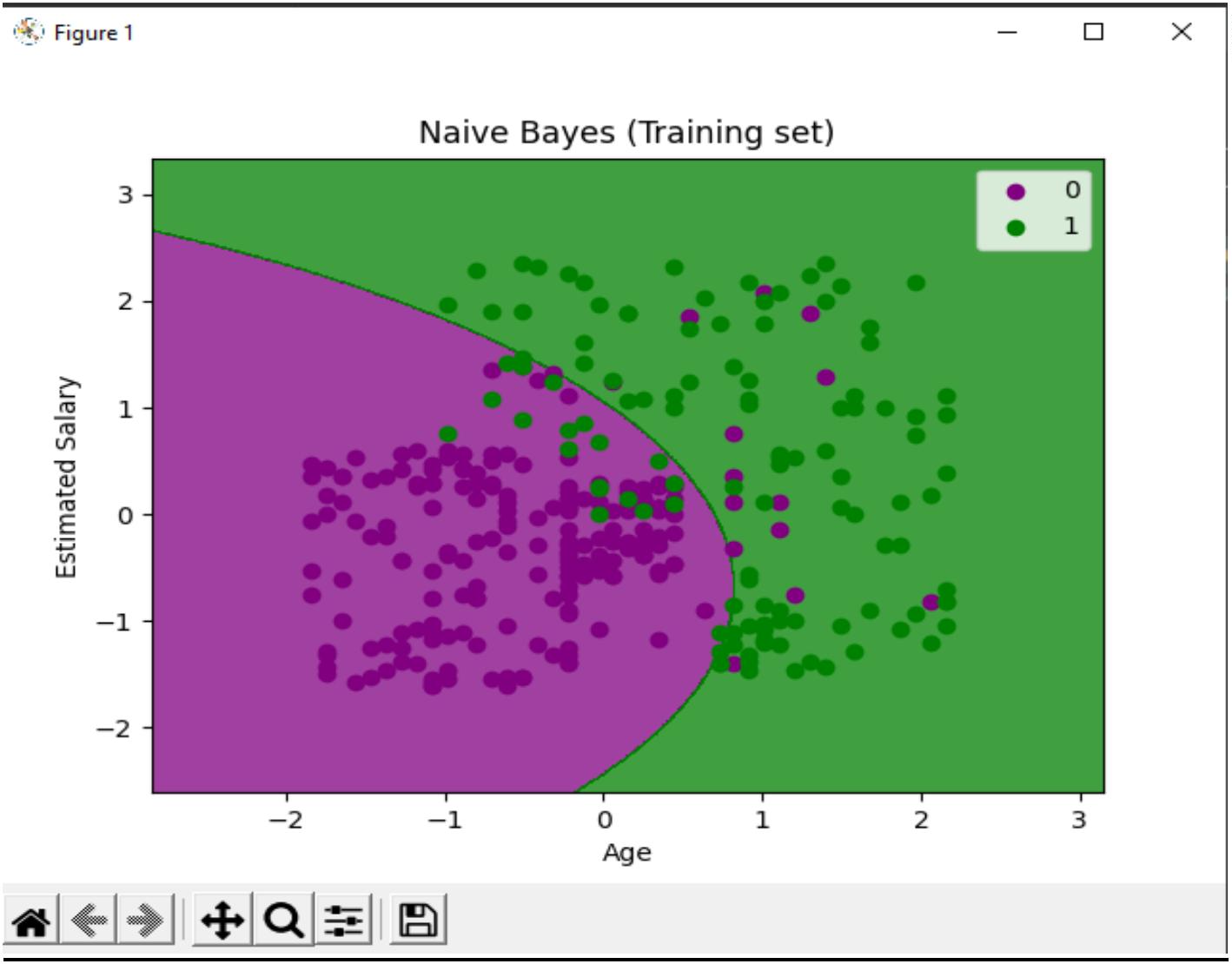
1000000001000011000101101011101000000

000011]

A ccuracy 91.25

20MCA241 Data Science Lab

Dept. of Computer Applications



11

PROGRAM NO : 06

Date : 08/12/2021

AIM : Program to implement linear and multiple regression techniques using any standard

dataset available in the public domain.

PROGRAM CODE

import numpy as np

f rom sklearn.linear\_ model i mport LinearRegressi on

x = np.array( [ 2,6,7,8] ) .reshape( ( - 1,1) )

y =np.array([16,7,8,9])

model = Li nearRegressi on( )

model .f i t(x,y)

r\_ sq =model.score(x,y)

print("Score: ",r\_ sq)

print("Intercept: ",model . i ntercept\_ )

print("Slope: ",model .coef \_ )

y\_ pred = model.predict(x)

print("Y - prediction : ",y\_ pred)

OUT PUT

C: / Users/ aj cemca/ Py charmProj ects/ A rav i nd/ l i near\_ regressi on.py

Score: 0.7556626506024098

Intercept: 17.759036144578314

Slope: [ - 1. 34939759]

Y -prediction : [15.06024096 9.6626506 8.31325301 6.96385542]

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



12

PROGRAM NO : 07

Date : 08/12/2021

AIM : Program to implement Linear and Multiple regression techniques using any standard

dataset available in public domain and evaluate its performance.

PROGRAM CODE

import numpy as np

import matplotlib.pyplot as plt

x = np.array( [ 2,6,7,8] )

y =np.array([16,7,8,9])

n =np.size(x)

n\_x = np.mean( x )

n\_ y =np.mean(y)

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

SS\_ \_ xx = np.sum(x\* x)- n\* n\_x\*n\_x

b\_1 =SS\_\_xy/SS\_\_xx

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

y\_pred =b\_1 \* x +b\_0

pri nt( y\_ pred)

plt.scatter(x, y, color='red')

plt.plot(x, y\_ pred, col or='green')

plt.xlabel('X ')

pl t.yl abel ( 'y')

plt.show()

20MCA241 Data Science Lab

Dept. of Computer Applications



13

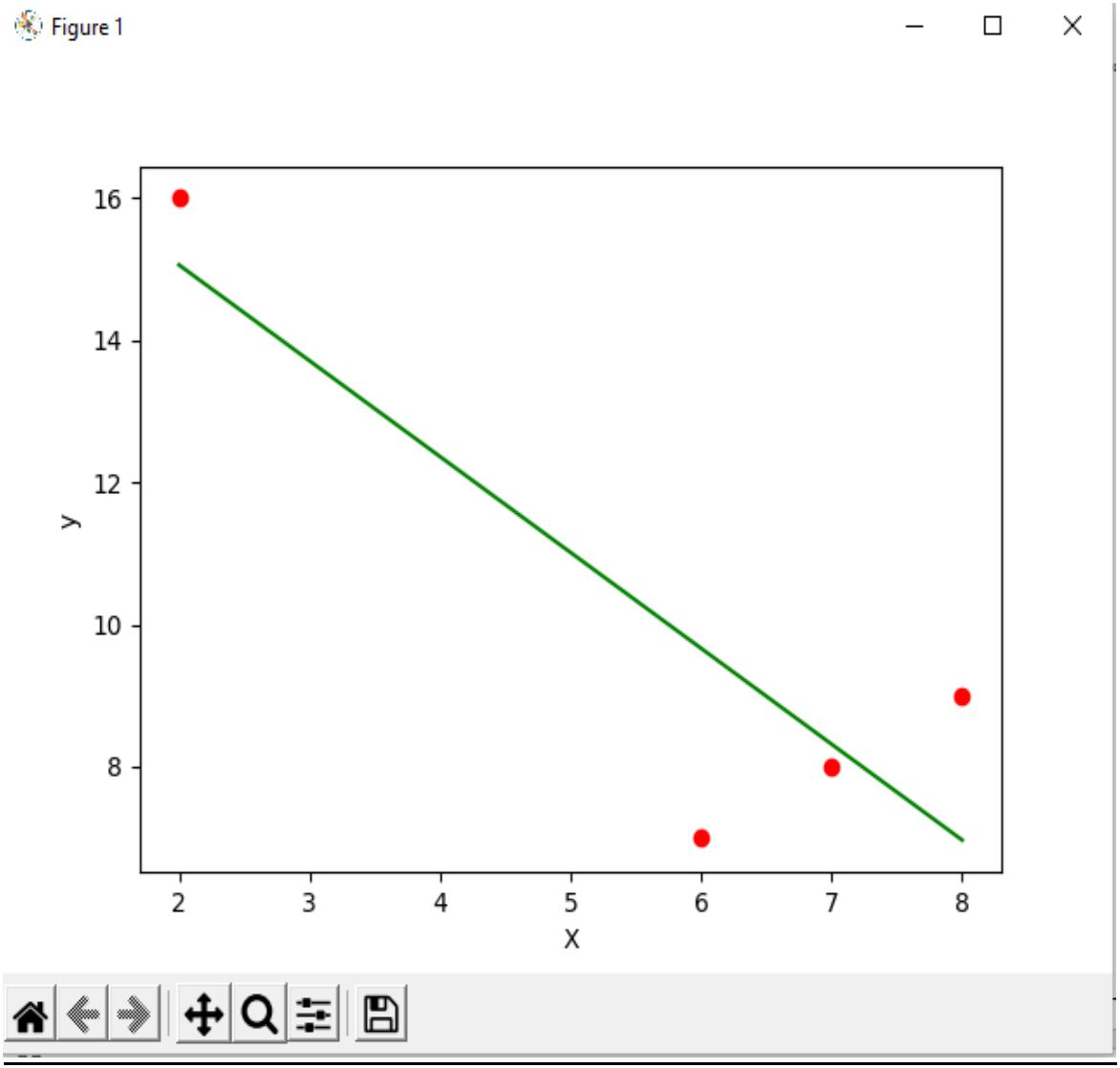
OUT PUT

C: / Users/ aj cemca/ Py charmProj ects/ A rav i nd/ l i near2.py

[15.06024096 9.6626506 8.31325301 6.96385542]

20MCA241 Data Science Lab

Dept. of Computer Applications



14

PROGRAM NO : 08

Date : 15/12/2021

AIM : Program to implement Linear and Multiple regression techniques using cars dataset

available in public domain and evaluate its performance

PROGRAM CODE

import pandas

from sklearn import linear\_ model

df = pandas.read\_ csv( " cars.csv" )

X = df [['Wei ght', 'V ol ume']]

y =df['CO2']

regr = l i near\_ model .Li nearRegressi on()

regr.fit(X , y)

#predict the CO2

predictedCO2 = regr.predict([[2300, 1300]])

print(predictedCO2)

OUT PUT

C: / Users/ aj cemca/ Py charmProj ects/ A rav i nd/ ML R.py

[107.2087328]

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



15

PROGRAM NO : 09

Date : 15/12/2021

AIM : Program to implement multiple linear regression techniques using Boston dataset

available in the public domain and evaluate its performance and plotting graph.

PROGRAM CODE

import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear\_model, metrics

from sklearn.metrics import r2\_ score

boston = datasets.l oad\_ boston(return\_ X \_ y=Fal se)

X =boston.data

y =boston.target

from sklearn.model\_ selection import train\_ test\_ split

X\_train, X\_test, y\_ trai n, y\_ test =train\_ test\_ split(X , y, test\_ size=0.4,random\_ state=1)

reg = l i near\_ model .Li nearRegressi on()

reg.fit(X \_ train, y\_ trai n)

V =reg.predi ct(X \_ test)

result=r2\_ score(y\_ test, V )

print("accuracy :", result)

pri nt( 'Coef f i ci ents: ', reg.coef\_)

pri nt('V ari ance score:{}'.f ormat(reg.score(X \_ test, y\_ test)))

20MCA241 Data Science Lab

Dept. of Computer Applications



16

OUT PUT

accuracy : 0.7209056672661767

Coef f i ci ents: [ - 8.95714048e- 02 6.73132853e-02 5.04649248e-02 2. 18579583e+00

- 1. 72053975e+01 3. 63606995e+00 2.05579939e- 03 - 1.36602886e+00

2.89576718e- 01 - 1.22700072e- 02 - 8.34881849e- 01 9.40360790e-03

- 5. 04008320e- 01]

V ari ance score: 0.7209056672661767

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



17

PROGRAM NO : 10

Date : 22/12/2021

AIM : Program to implement decision tree using any standard dataset available in the public

domain and find the accuracy of the algorithm

PROGRAM CODE

Import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

f rom skl earn.preprocessi ng i mport Label Encoder

from sklearn.model\_ selection import train\_ test\_ split

f rom sklearn.tree import DecisionTreeClassifier

f rom sklearn.metrics import classification\_ report, confusion\_ matrix

f rom skl earn.tree i mport pl ot\_ tree

df =sns.l oad\_ dataset( 'i ri s')

pri nt( df .head( ) )

pri nt(df .i nf o())

df .i snul l ().any()

print(df.shape)

sns.pairplot(data=df, hue ='species')

pl t.savef i g(" pne.png" )

sns.heatmap(df.corr())

pl t.savef i g("nex t.png" )

target =df['speci es']

df1 =df.copy()

df1 = df1.drop('speci es', axi s=1)

print(df1.shape)

pri nt( df 1.head( ) )

x=df1

print(target)

le = LabelEncoder()

target = le.f it\_ transf orm(target)

print(target)

y= target

20MCA241 Data Science Lab

Dept. of Computer Applications



18

x\_train, x\_test, y\_ trai n, y\_test =train\_test\_split(x, y, test\_ size=0.2, random\_ state= 42)

print("training split input" , x\_train.shape)

print("test split input",x\_ test.shape)

dtree=Deci si onT reeCl assi f i er()

dtree.f i t(x\_ trai n, y\_ trai n)

print("decision tree classifer created")

y\_ pred = dtree.predict(x\_ test)

pri nt("cl assi f i cati on report- \n",cl assi f i cati on\_ report(y\_ test,y\_ pred))

cm = confusion\_ matrix(y\_ test,y\_ pred)

pl t.f i gure(f i gsi ze=(5,5))

sns. heatmap( data=cm, l i newi dths=. 5, annot=T rue, square=T rue, cmap='B l ues')

pl t.yl abel ('A ctual l abel ')

plt.xlabel('predicted label ')

all\_ sample\_ title = 'A ccuracy Score:{0}'.format(dtree.score(x\_ test,y\_ test))

plt.ti tl e(all \_ sampl e\_ ti tl e,size=12)

plt.savefig("two.png")

pl t.f i gure(f i gsi ze=(20,20))

dec\_ tree=pl ot\_ tree( deci si on\_ tree=dtree,f eature\_ names=df 1.col umns,cl ass\_ names=[ " setosa" , " verci c

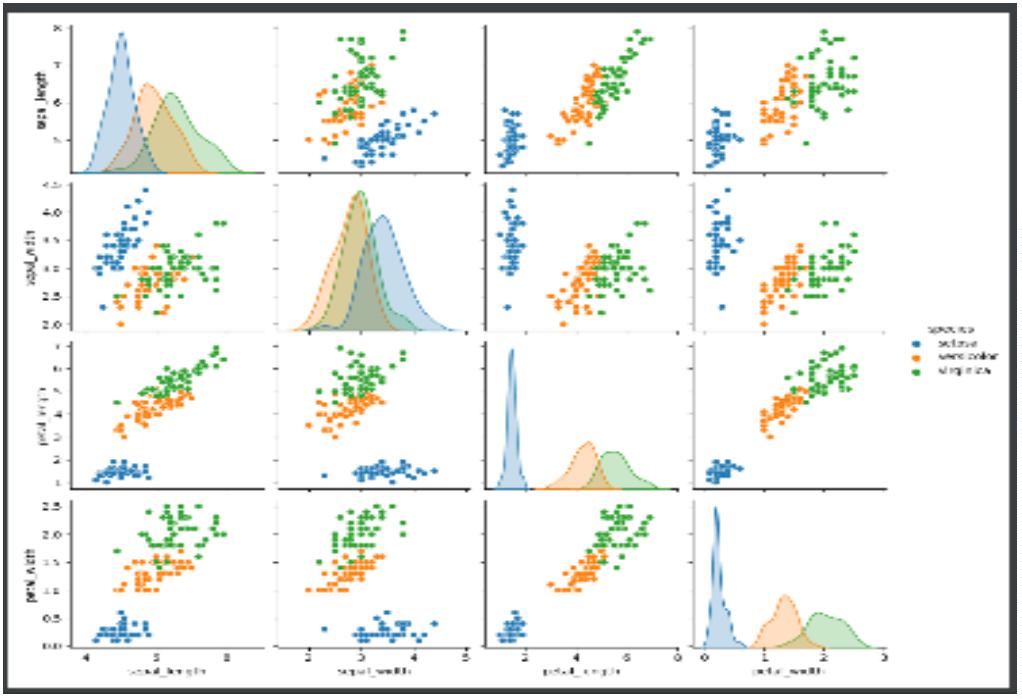
ol or","vergi ni ca"],f i l l ed=True ,preci si on=4,rounded=True)

pl t.savef i g( " three.png" )

OUT PUT

20MCA241 Data Science Lab

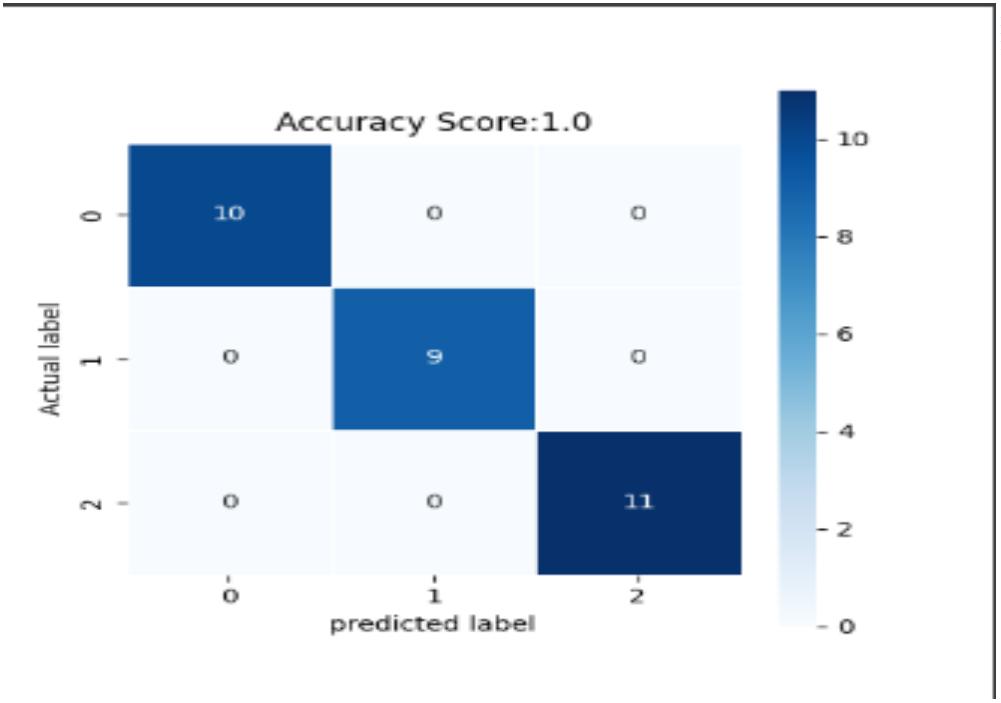
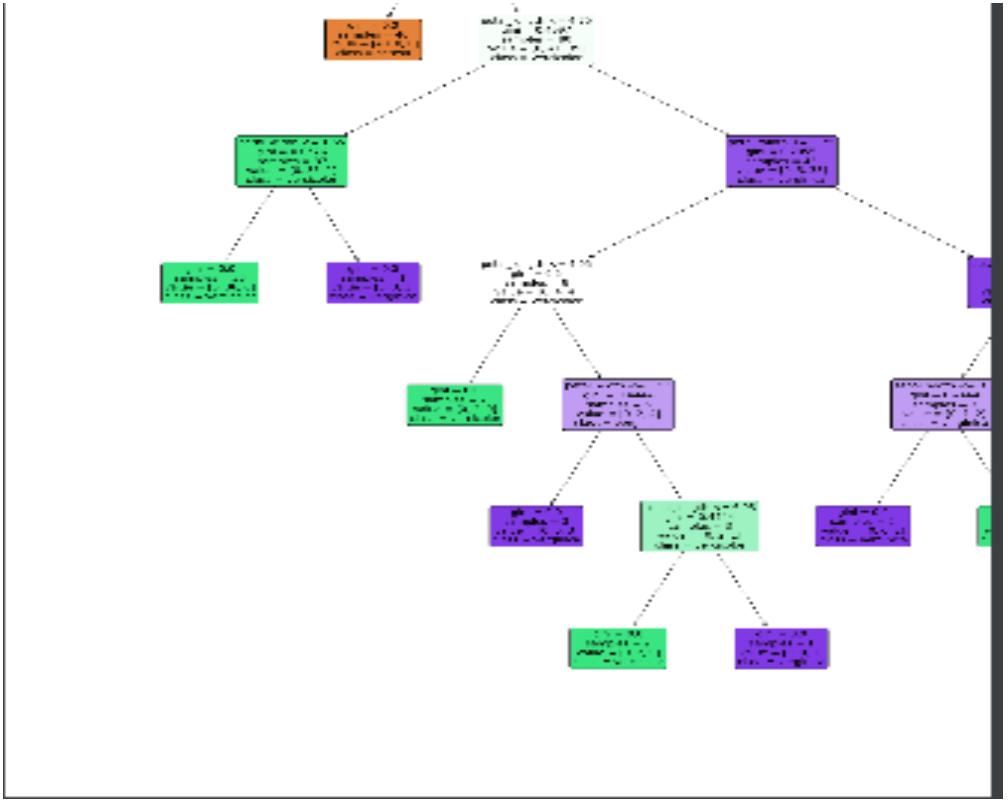
Dept. of Computer Applications



19

20MCA241 Data Science Lab

Dept. of Computer Applications



20

PROGRAM NO : 11

Date : 05/01/2022

AIM : Program to implement K -Means clustering technique using any standard dataset

available in the public domain

PROGRAM CODE

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

dataset=pd.read\_ csv( 'Mal l \_ Customers. csv ')

x=dataset.i l oc[:,[3,4]].val ues

print(x)

from sklearn.cluster import KMeans

wcss\_ list=[]

for i in range(1,11):

kmeans=K Means( n\_ cl usters=i ,i ni t='k- means++',random\_ state=42)

kmeans.f i t(x)

wcss\_ l i st.append( kmeans.i nerti a\_ )

mtp.plot(range(1,11), wcss\_ list)

mtp.title('The Elbow Method Graph')

mtp.xl abel('Number of clusters(k)')

mtp.yl abel ('wcss\_ l i st')

mtp.show()

20MCA241 Data Science Lab

Dept. of Computer Applications



21

kmeans=K Means( n\_ cl usters=5, i ni t='k- means++', random\_ state=42)

y\_ predi ct=kmeans.f i t\_ predi ct(x)

pri nt( 'predi ct=', y\_ predi ct)

mtp.scatter(x[y\_ predi ct==0,0],x[y\_ predi ct==0,1],s=100,c='bl ue',l abel ='Cl uster 1')

mtp.scatter(x[y\_ predi ct==1,0],x[y\_ predi ct==1,1],s=100,c='red',l abel ='Cl uster 2')

mtp.scatter(x[y\_ predi ct==2,0],x[y\_ predi ct==2,1],s=100,c='green',l abel ='Cl uster 3')

mtp.scatter(x[y\_ predi ct==3,0],x[y\_ predi ct==3,1],s=100,c='yel l ow',l abel ='Cl uster 4')

mtp.scatter( x[ y\_ predi ct==4,0] ,x[ y\_ predi ct==4,1] ,s=100,c='magenta',l abel ='Cl uster 5')

mtp.scatter( kmeans.cl uster\_ centers\_ [ : ,0] ,kmeans.cl uster\_ centers\_ [ : ,1] ,s=300,c='bl ack')

mtp.title('Clusters of Customer')

mtp.xl abel ( 'A nnual Income(k$)')

mtp.ylabel('Spending Score (1- 100)')

mtp. l egend( ) ;

mtp.show()

OUT PUT

C:/Users/ajcemca/PycharmProjects/Aravind/kmeans.py

[[ 15 39]….

[137 18]

[137 83]]

predict=[2323232323232323232323232323232323232

3232320230000000000000000000000000000

0000000000000000000000000000000000000

20MCA241 Data Science Lab

Dept. of Computer Applications



22

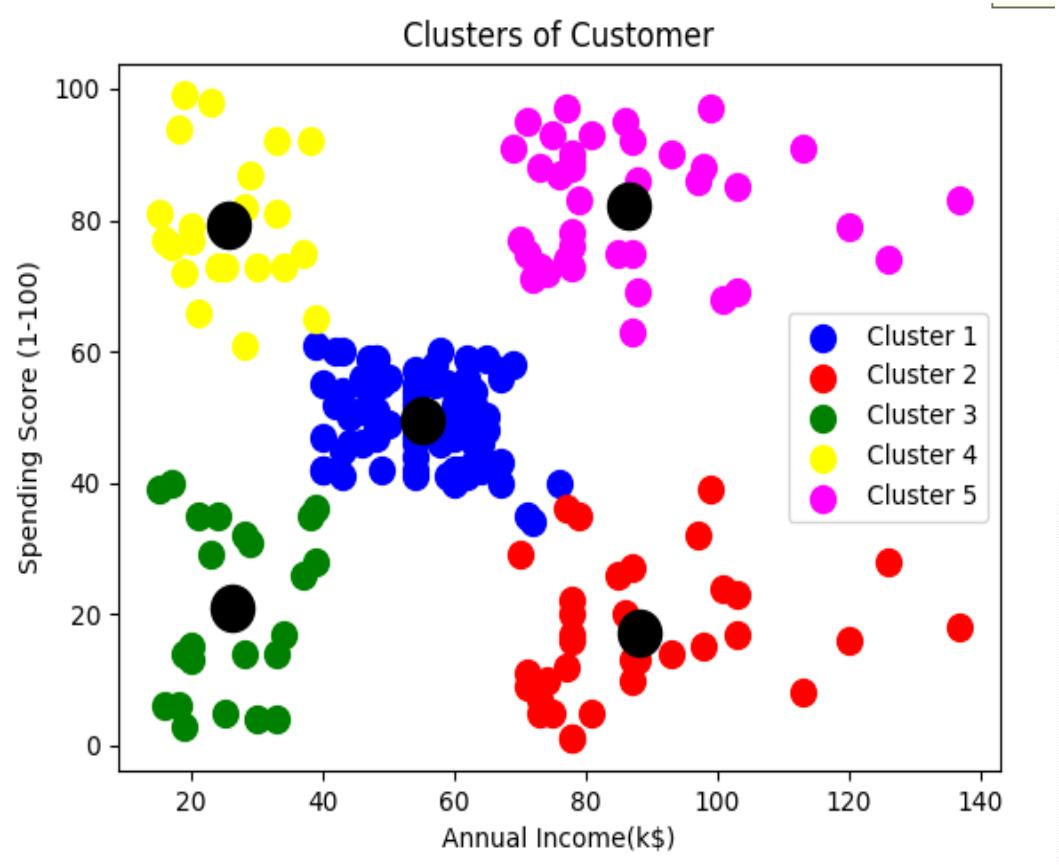
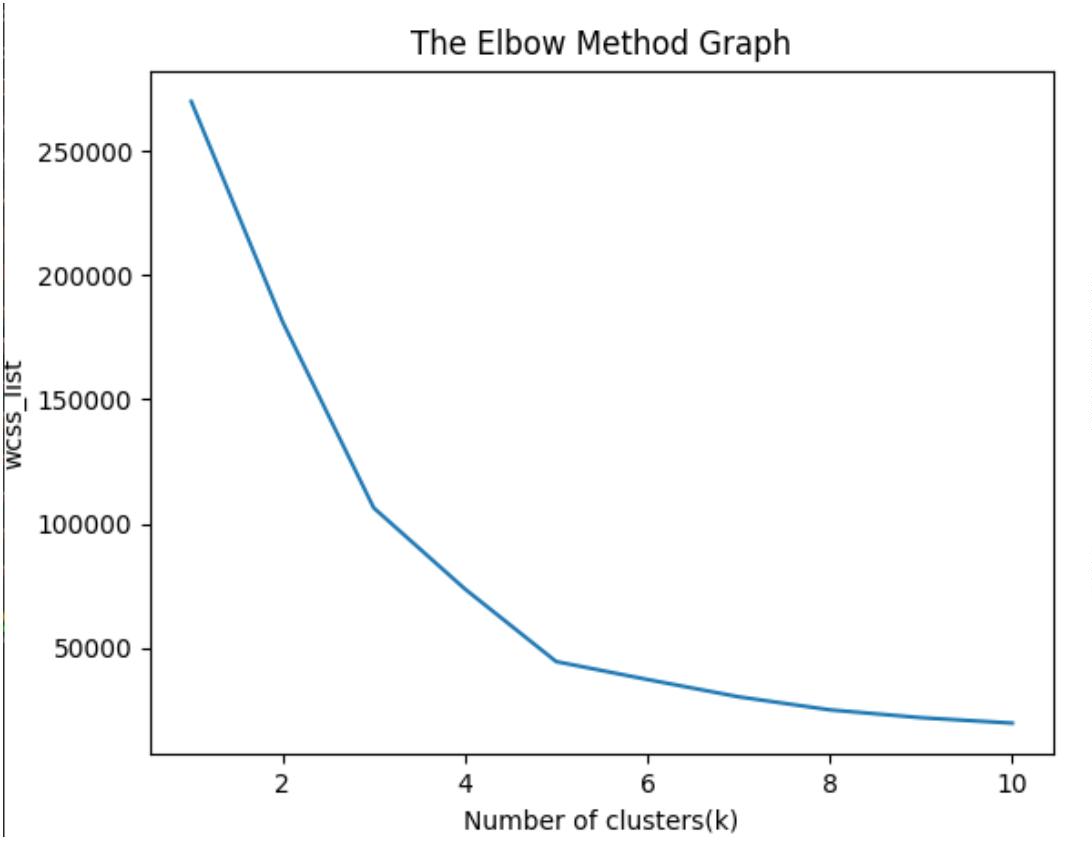
0000000000004140414140414141414041414

1414141414141414141414141414141414141

414141414141414]

20MCA241 Data Science Lab

Dept. of Computer Applications



23

PROGRAM NO : 12

Date : 05/01/2022

AIM : Program to implement K -Means clustering technique using any standard dataset

available in the public domain.

PROGRAM CODE

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

dataset=pd.read\_ csv( 'worl d\_ country\_ and\_ usa\_ states\_ l ati tude\_ and\_ l ongi tude\_ val ues.csv')

x=dataset.i l oc[:,[1,2]].val ues

print(x)

from sklearn.cluster import KMeans

wcss\_ list=[]

for i in range(1,11):

kmeans=K Means( n\_ cl usters=i ,i ni t='k- means++',random\_ state=42)

kmeans.f i t(x)

wcss\_ l i st.append( kmeans.i nerti a\_ )

mtp.plot(range(1,11), wcss\_ list)

mtp.title('The Elbow Method Graph')

mtp.xl abel('Number of clusters(k)')

mtp.yl abel ('wcss\_ l i st')

mtp.show()

kmeans=K Means( n\_ cl usters=3, i ni t='k- means++', random\_ state=42)

y\_ predi ct=kmeans.f i t\_ predi ct(x)

pri nt( 'predi ct=', y\_ predi ct)

20MCA241 Data Science Lab

Dept. of Computer Applications



24

mtp.scatter(x[y\_ predi ct==0,0],x[y\_ predi ct==0,1] ,s=100,c='bl ue',l abel ='Cl uster 1')

mtp.scatter(x[y\_ predi ct==1,0],x[y\_ predi ct==1,1],s=100,c='red',l abel ='Cl uster 2')

mtp.scatter(x[y\_ predi ct==2,0],x[y\_ predi ct==2,1],s=100,c='green',l abel ='Cl uster 3')

mtp.scatter( kmeans.cl uster\_ centers\_ [ : ,0] ,kmeans.cl uster\_ centers\_ [ : ,1] ,s=300,c='bl ack')

mtp.title('Clusters of world Country')

mtp.xlabel('latitude')

mtp.ylabel('longitude')

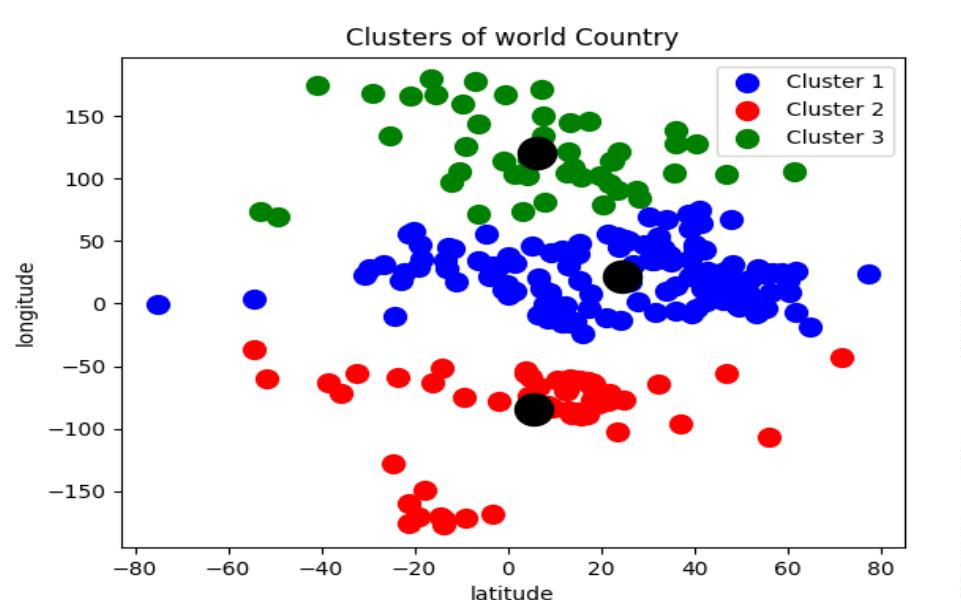
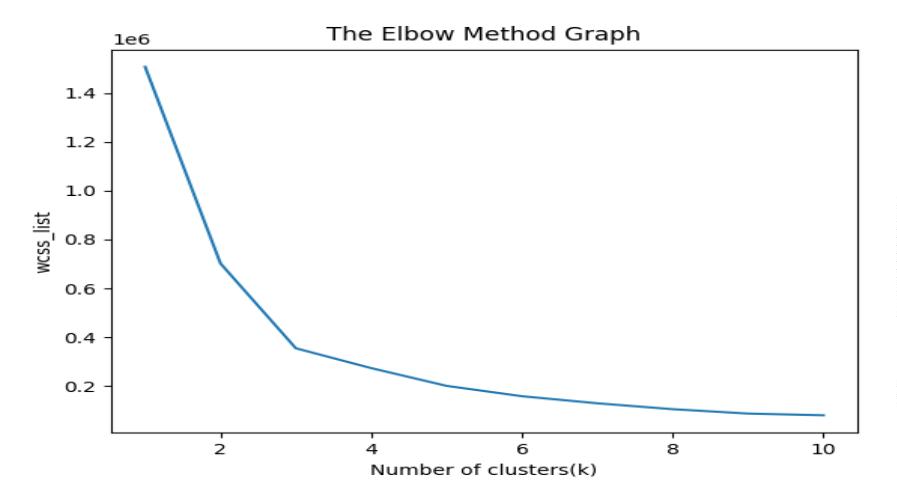
mtp. l egend( ) ;

mtp.show()

OUT PUT

20MCA241 Data Science Lab

Dept. of Computer Applications



25

PROGRAM NO : 13

Date : 02/02/2022

AIM : Programs on convolutional neural network to classify images from any standard

dataset in the public domain.

PROGRAM CODE

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow import keras

np.random.seed(42)

#tf .set.random. seed(42)

f ashi on\_ mni st = keras.datasets.f ashi on\_ mni st

(X \_ train, y\_train), (X\_test, y\_ test) = f ashi on\_ mni st.l oad\_ data()

print(X \_ train.shape, X \_ test.shape)

X\_train =X\_train / 255.0

X\_test =X\_test / 255.0

plt.imshow(X \_ train[1], cmap='binary')

plt.show()

np.uni que( y\_ test)

class\_names =['T-Shirt/Top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker',

'8ag', 'Ankle Boot']

n\_rows =5

n\_cols =10

plt.figure(figsize=(n\_ cols \* 1.4, n\_rows \* 1.6) )

20MCA241 Data Science Lab

Dept. of Computer Applications



26

for row in range(n\_rows):

for col in range(n\_ cols):

i ndex =n\_cols \* row +col

plt.subplot(n\_rows, n\_cols, index +1)

plt.imshow(X \_ train[index], cmap='binary', interpolation='nearest')

pl t.axi s('of f ')

pl t.ti tl e(cl ass\_ names[ y\_ trai n[ i ndex]])

plt.show()

model \_ CNN = keras.model s.Sequenti al ()

model\_ CNN.add(keras.layers.Conv2D(f ilters=32, kernel\_ size=7, padding='same',

activation='relu', input\_ shape=[28, 28, 1]))

model \_ CNN.add( keras.l ayers.Max Pool i ng2D( pool \_ si ze=2) )

model\_ CNN.add(keras.layers.Conv2D(f ilters=64, kernel\_ size=3, padding='same',

acti vati on='rel u') )

model \_ CNN.add( keras.l ayers.Max Pool i ng2D( pool \_ si ze=2) )

model\_ CNN.add(keras.layers.Conv2D(f ilters=32, kernel\_ size=3, padding='same',

acti vati on='rel u') )

model \_ CNN.add( keras.l ayers.Max Pool i ng2D( pool \_ si ze=2) )

model \_ CNN.summary( )

model \_ CNN. add( keras. l ayers. Fl atten( ) )

model \_ CNN.add(keras.l ayers.Dense(uni ts=128, acti vati on='rel u'))

model \_ CNN.add(keras.l ayers.Dense(uni ts=64, acti vati on='rel u'))

model \_ CNN.add(keras.l ayers.Dense(uni ts=10, acti vati on='sof tmax'))

model \_ CNN.summary( )

20MCA241 Data Science Lab

Dept. of Computer Applications



27

model \_ CNN.compi l e(l oss='sparse\_ categori cal \_ crossentropy', opti mi zer='adam',

metri cs=[ 'accuracy'] )

X \_train =X \_train[..., np.newaxis]

X\_test =X \_test[..., np.newaxis]

history\_ CNN = model\_ CNN.fit(X \_ train, y\_ train, epochs=2, validation\_ split=0.1)

pd.DataFrame( hi story\_ CNN.hi story) . pl ot( )

pl t.gri d( T rue)

pl t.xl abel ( 'epochs')

pl t. yl abel ( 'l oss/ accuracy')

plt.title('Training and validation plot')

plt.show()

test\_ loss, test\_ accuracy = model\_ CNN.evaluate(X \_ test, y\_ test)

print(' Test Loss :{}, Test Accuracy : {}'.format(test\_loss, test\_accuracy))

20MCA241 Data Science Lab

Dept. of Computer Applications

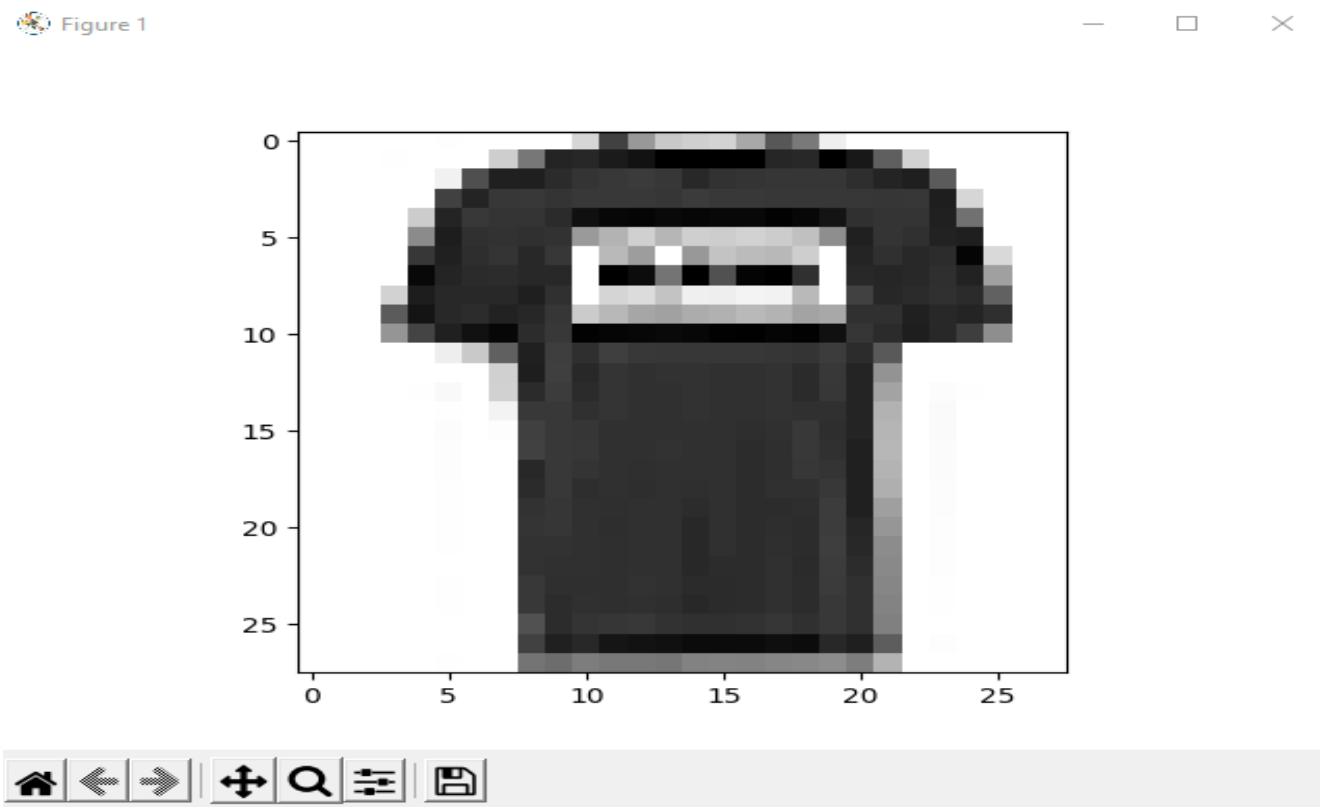


28

OUT PUT

20MCA241 Data Science Lab

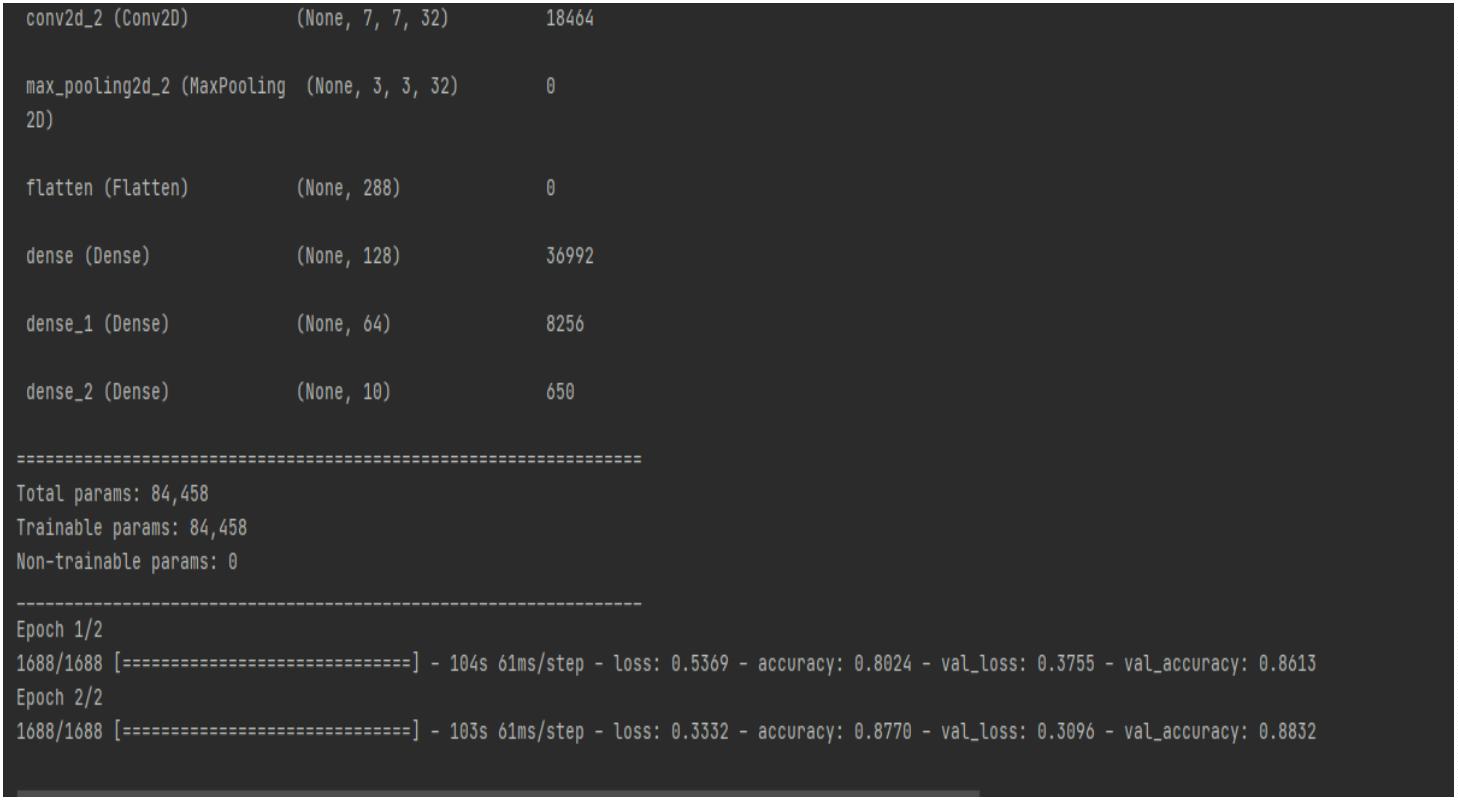
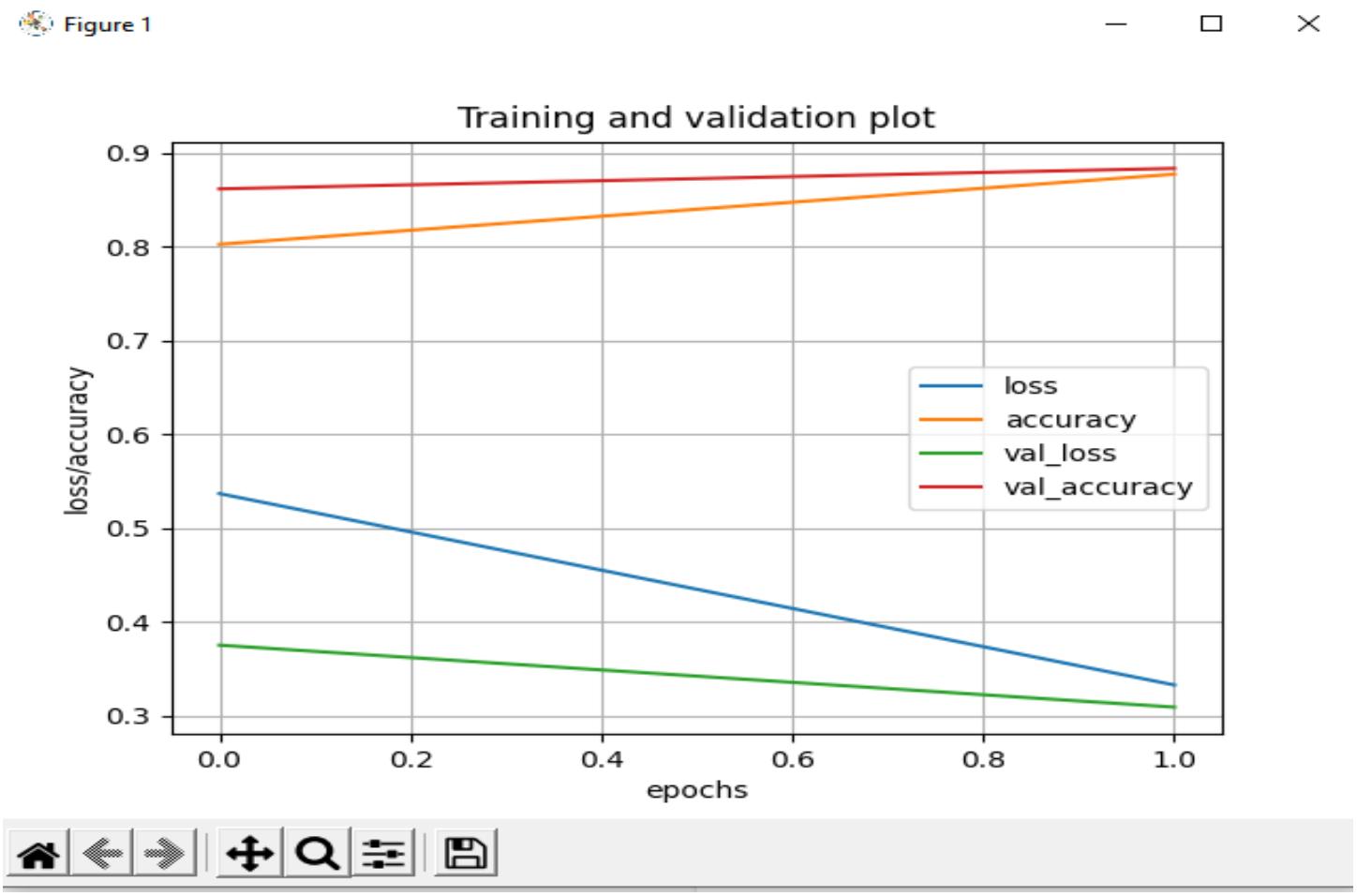
Dept. of Computer Applications



29

20MCA241 Data Science Lab

Dept. of Computer Applications



30

PROGRAM NO : 14

Date : 16/02/2022

AIM : Program to implement a simple web crawler using python.

PROGRAM CODE

import requests

import lxml

from bs4 import BeautifulSoup

#import beautifulsoup4

url ="https://www.rottentomatoes.com/top/bestofrt/"

headers ={ 'User-Agents' : 'Mozilla/5.0 (Windows NT 6.1; WOW64) A ppl eWebK i t/537.36

(KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36 QIHU 36OSE'}

f = requests.get(url, headers = headers)

movies\_list =[]

soup = BeautifulSoup(f.content, 'html.parser')

movies =soup.find('table', {'class' : 'table'}) .find\_ all('a')

pri nt( movi es)

num =0

for anchor in movies:

url s = 'https://www.rottentomatoes.com' + anchor['href ']

movi es\_ l i st.append( url s)

pri nt(movi es\_ l i st)

num +=1

movi e\_ url=urls

#movie\_ url=movies\_ lst

mov i e\_ f =requests.get( mov i e\_ url ,headers=headers)

20MCA241 Data Science Lab

Dept. of Computer Applications



31

movi e\_ soup=Beauti f ul Soup(movi e\_ f .content,'l xml ')

movi e\_ content=movi e\_ soup.f i nd('di v',{

'class':'movie\_ synopsis clamp clamp- 6 js- clamp'

})

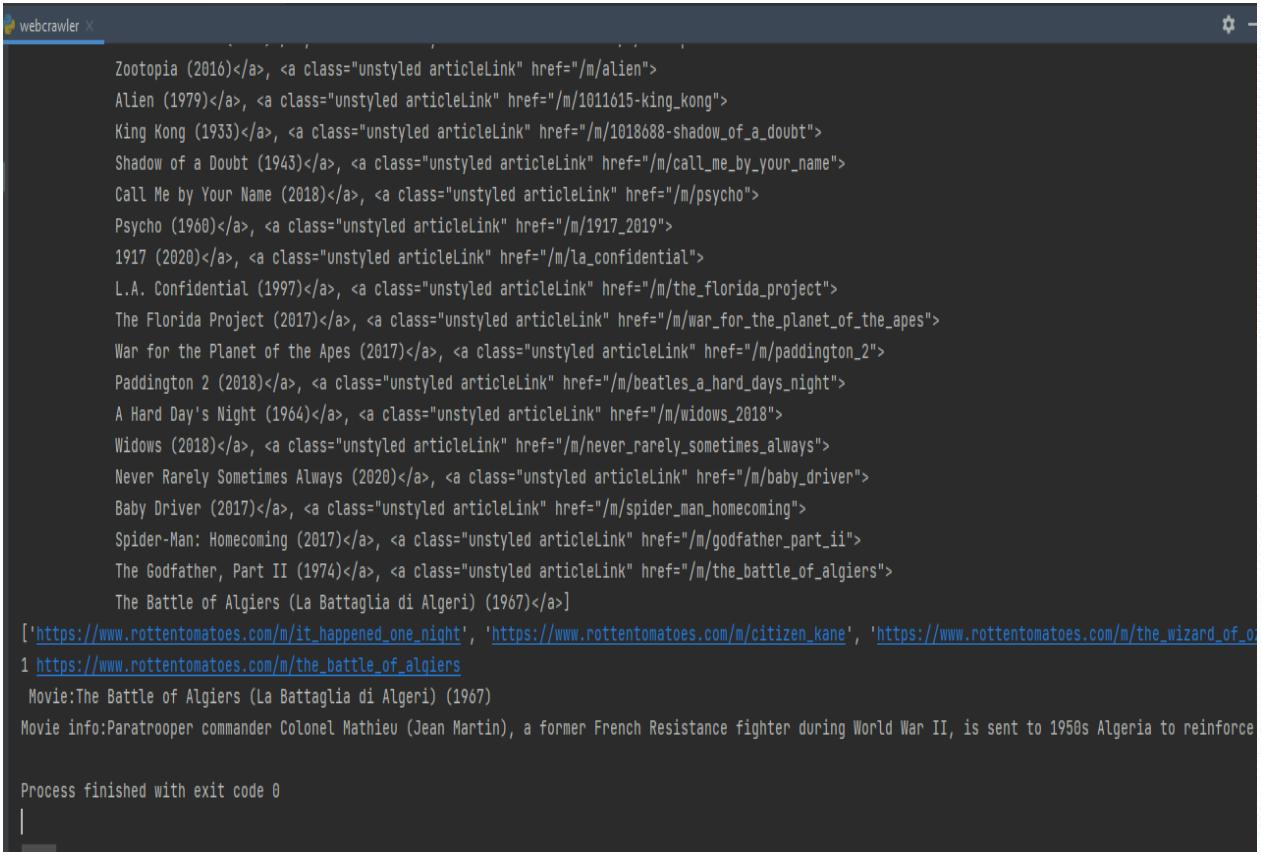
pri nt(num,url s,'\n','Movi e:' + anchor.stri ng.stri p())

print('Movie info:' +movie\_ content.string.strip())

OUT PUT

20MCA241 Data Science Lab

Dept. of Computer Applications



32

PROGRAM NO : 15

Date : 16/02/2022

AIM : Program to implement a simple web crawler using python.

PROGRAM CODE

from bs4 import BeautifulSoup

import requests

pages\_ crawled =[ ]

def crawler(url):

page =requests.get(url )

soup=BeautifulSoup(page.text,'html.parser')

l inks=soup.f i nd\_ al l('a')

for link in links:

if 'href' in link.attrs:

if link['href'].startswith('/wiki') and ':' not in link['href']:

if l i nk['href '] not in pages\_ crawled:

new\_ l i nk = f"https://en.wi ki pedi a.org{li nk['href ']}"

pages\_crawled.append(link['href'])

try:

wi th open('data.csv','a') as file:

f i l e.wri te(f '{soup.ti tl e.text}:{l i nk["href "]}\n')

crawl er( new\_ l i nk)

except:

continue

crawler(['https://en.wikipedia.org')](https://en.wikipedia.org/)

20MCA241 Data Science Lab

Dept. of Computer Applications

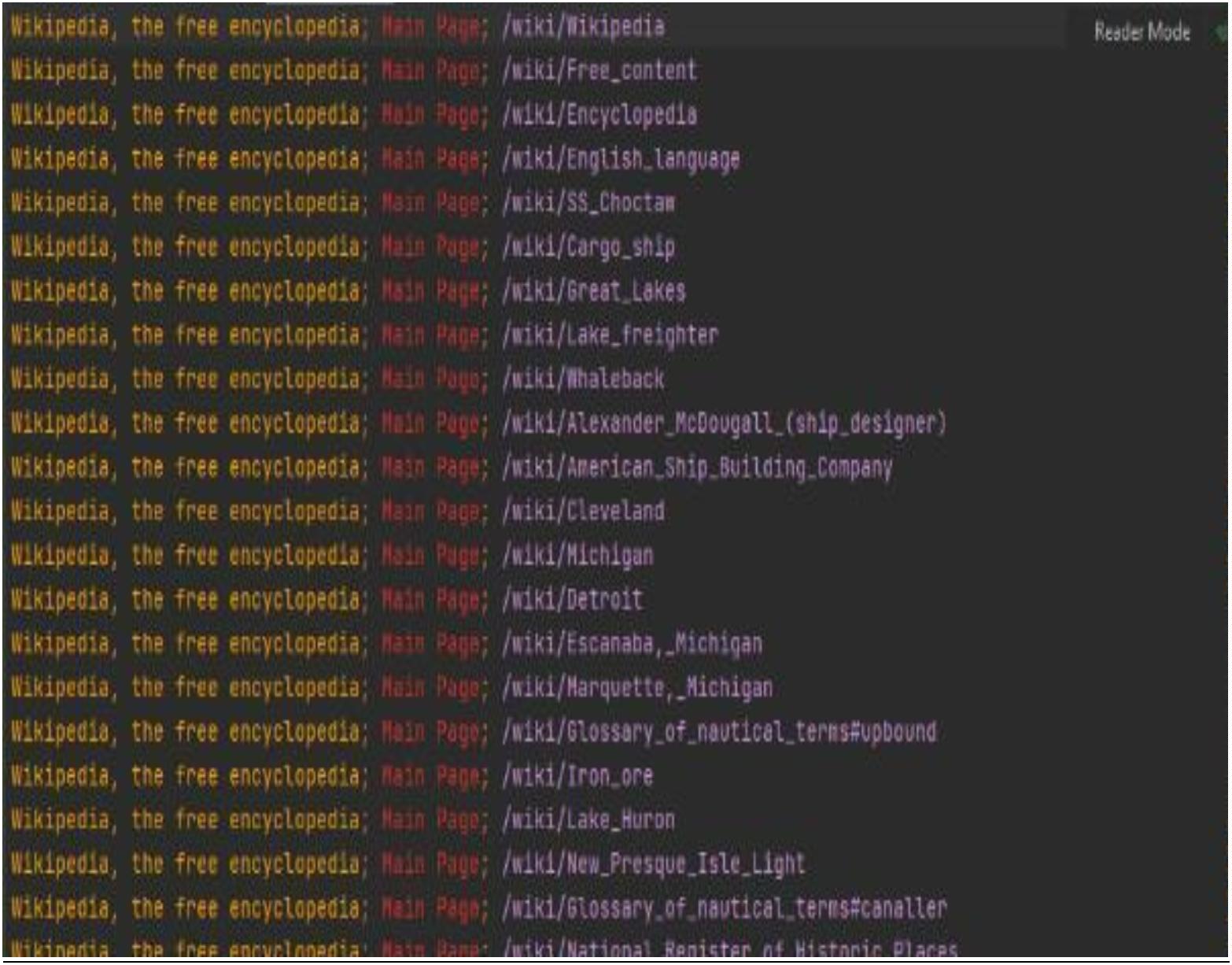


33

OUT PUT

20MCA241 Data Science Lab

Dept. of Computer Applications



34

PROGRAM NO : 16

Date : 16/02/2022

AIM : Program to implement scrap of any website.

PROGRAM CODE

import requests

from bs4 import BeautifulSoup

import csv

URL = "http://www.values.com/inspi rati onal - quotes"

r = requests.get(URL)

print(r.content)

soup =BeautifulSoup(r.content, 'lxml')

pri nt( soup.pretti f y( ) )

quotes =[]

table =soup.find('div', attrs={'id': 'al l \_ quotes'})

for row in table.findAll('div',

attrs={'cl ass': 'col- 6 col- lg- 3 text- center margin- 30px- bottom sm- margin- 30px-

top'}):

quote ={}

quote['theme'] =row.h5.text

quote['url'] = row.a[ 'href ']

quote['img'] =row.img['src']

quote['lines'] =row.img['alt'].split(" #")[0]

quote['author'] =row.img['alt'].split(" #")[1]

quotes.append(quote)

20MCA241 Data Science Lab

Dept. of Computer Applications



35

f ilename = 'inspirational \_ quotes.csv'

with open(filename, 'w', newl i ne='') as f:

w =csv.DictWriter(f, ['theme', 'url', 'i mg', 'lines', 'author'])

w.writeheader()

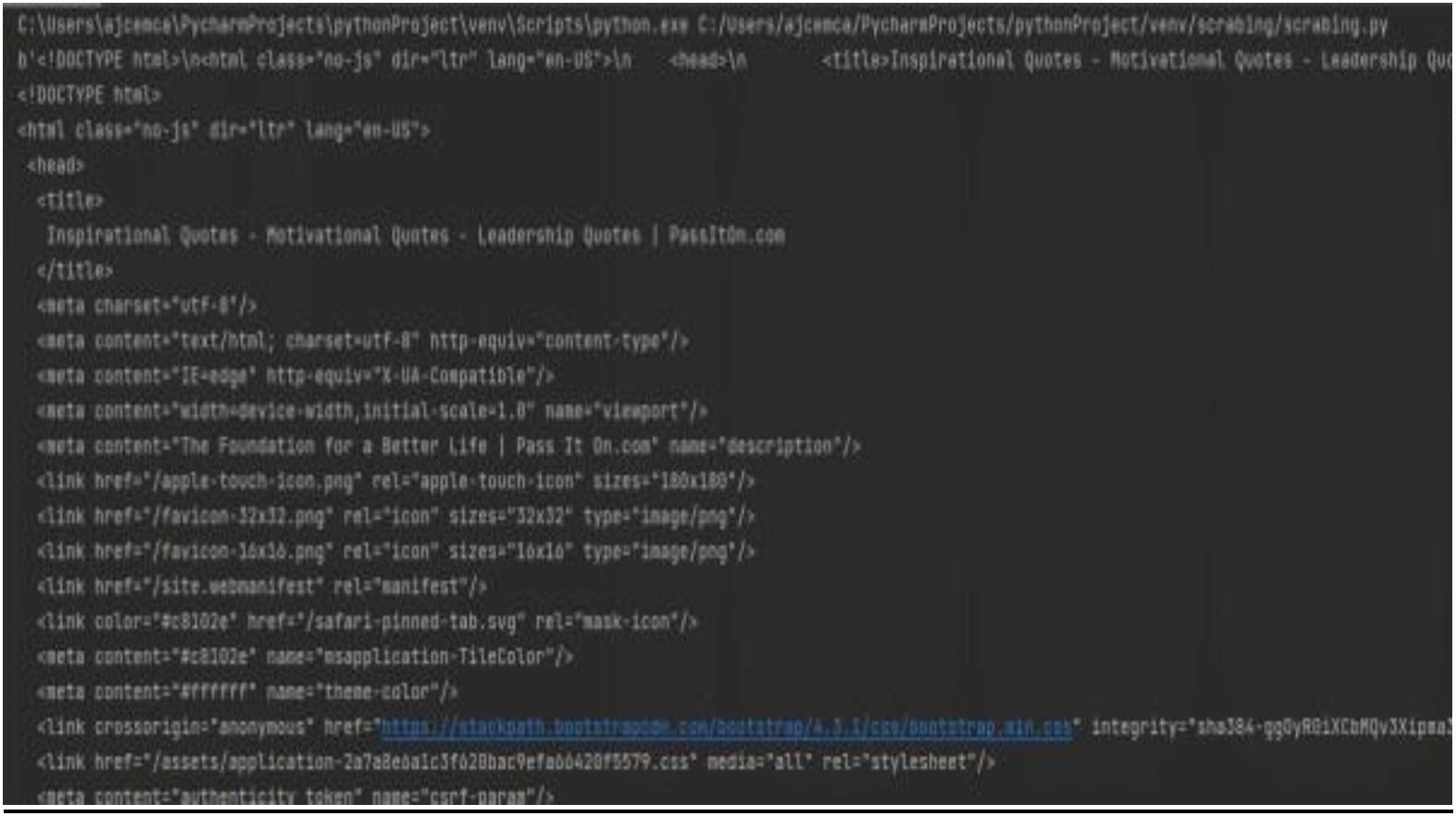
for quote in quotes:

w.writerow(quote)

OUT PUT

20MCA241 Data Science Lab

Dept. of Computer Applications



36

PROGRAM NO : 17

Date : 16/02/2022

AIM : Program for Natural Language Processing which performs n-grams.

PROGRAM CODE

def generate\_ ngrams( tex t, WordsT oCombi ne) :

words=text. spl i t( )

output=[ ]

for i in range(len(words) - WordsT oCombi ne+1) :

output.append(words[i:i +WordsT oCombi ne] )

return output

x=generate\_ ngrams(text='this is a very good book to study',WordsToCombine=3)

pri nt( x)

OUT PUT

[['this', 'is', 'a'], ['is', 'a', 'very'], ['a', 'very', 'good'], ['very', 'good', 'book'], ['good', 'book', 'to'],

['book', 'to', 'study']]

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



37

PROGRAM NO : 18

Date : 16/02/2022

AIM : Program for Natural Language Processing which performs n-grams (Using in built

functions).

PROGRAM CODE

import nltk

nltk.download('punkt')

from nltk.util import ngrams

sampleText='this is a very good book to study'

NGRA MS=ngrams( sequence=nl tk.word\_ tokeni ze( sampl eT ex t) ,n=2)

for grams in NGRAMS:

pri nt( grams)

OUT PUT

('this', 'is')

('is', 'a')

('a', 'very')

('very', 'good')

('good', 'book')

('book', 'to')

('to', 'study')

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



38

PROGRAM NO : 19

Date : 16/02/2022

AIM : Program for Natural Language Processing which performs speech tagging.

PROGRAM CODE

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_ tokenize,sent\_ tokenize

#nl tk. downl oad( 'stopwords')

#nl tk. downl oad( 'av eraged\_ perceptron\_ tagger')

stop\_ words=set( stopwords. words( 'engl i sh') )

txt="A mmu,How are you."\

"A rchana,i am fine.How are you"\

"Sukanya is getting married next year"\

"Marriage is a big step in ones life"\

"yes it is a big event"\

"okey bye Ammu."

tokeni zed=sent\_ tokeni ze( tx t)

for i in tokenized:

wordsLi st=nl tk.word\_ tokeni ze( i )

wordsList=[w for w in wordsList if not w in stop\_words]

tagged=nl tk. pos\_ tag( wordsLi st)

print(tagged)

20MCA241 Data Science Lab

Dept. of Computer Applications



39

OUT PUT

[('A mmu', 'NNP'), (',', ','), ('How', 'NNP'), ('you.A rchana', 'NNP'), (',', ','), ('fine.How', 'NN'),

('youSukanya', 'RB'), ('getting', 'V BG'), ('married', 'VBN'), ('next', 'J J '), ('yearMarriage', 'NN'), ('big',

'JJ'), ('step', 'NN'), ('ones', 'NNS'), ('lifeyes', 'RB'), ('big', 'J J '), ('eventokey', 'NN'), ('bye', 'NN'),

('A mmu', 'NNP'), ('.', '.')]

Process finished with exit code 0

20MCA241 Data Science Lab

Dept. of Computer Applications



40

PROGRAM NO : 20

Date : 23/02/2022

AIM : Program for Natural Language Processing which performs Chunking.

PROGRAM CODE

import nltk

new="The big cat ate the little mouse who was after the fresh cheese"

new\_ tokens=nl tk.word\_ tokeni ze( new)

pri nt( new\_ tokens)

new\_ tag=nl tk.pos\_ tag( new\_ tokens)

pri nt( new\_ tag)

grammer=r" NP: {<DT>?<J J >\* <NN>}"

chunkParser=nltk.RegexpParser(grammer)

chunked=chunkParser.parse(new\_tag)

print(chunked)

chunked.draw()

OUT PUT

['The', 'big', 'cat', 'ate', 'the', 'little', 'mouse', 'who', 'was', 'after', 'the', 'fresh', 'cheese']

[('The', 'DT'), ('big', 'JJ'), ('cat', 'NN'), ('ate', 'VBD'), ('the', 'DT'), ('little', 'JJ'), ('mouse', 'NN'), ('who',

'WP'), ('was', 'VBD'), ('after', 'IN'), ('the', 'DT'), ('fresh', 'JJ'), ('cheese', 'NN')]

(S

(NP The/DT big/J J cat/NN)

ate/V BD

(NP the/DT little/J J mouse/NN)

who/WP

20MCA241 Data Science Lab

Dept. of Computer Applications



41

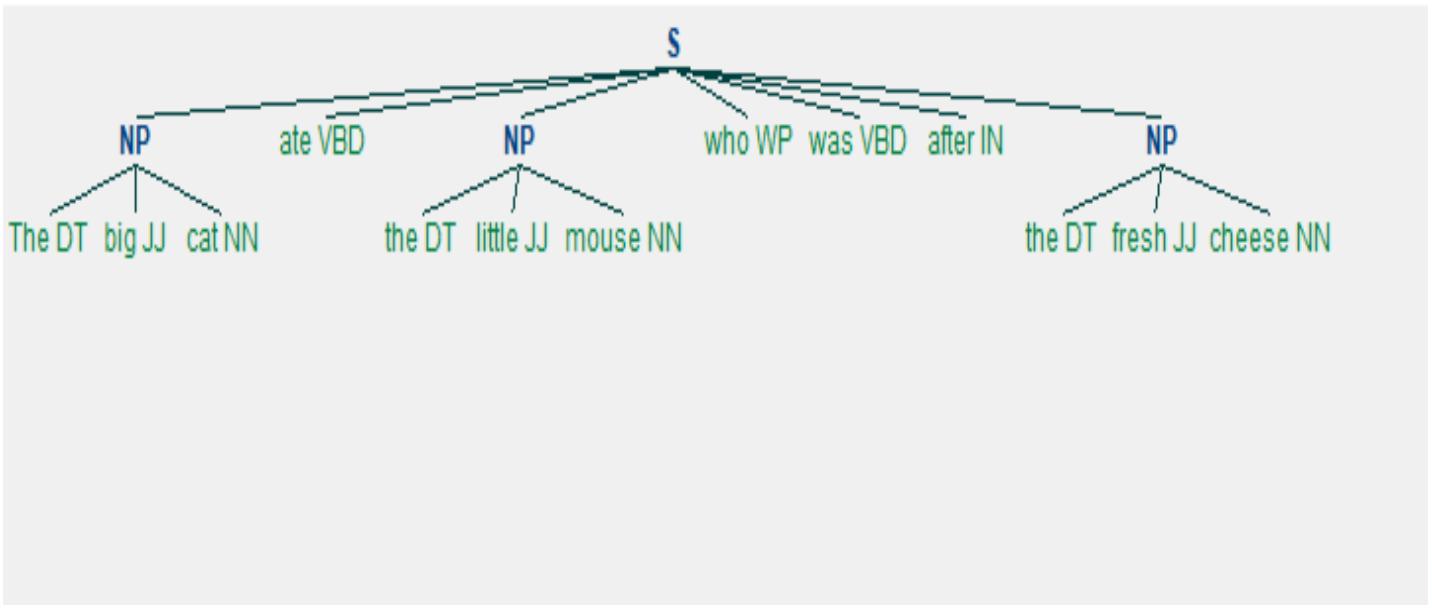
was/V BD

af ter/ IN

(NP the/DT f resh/J J cheese/NN))

20MCA241 Data Science Lab

Dept. of Computer Applications



42

PROGRAM NO : 21

Date: 23- 02- 2022

A im: Write a python program for natural program language processing with chunking.

Program:

import nltk

nltk.download('averaged\_perceptron\_tagger')

sample\_text ="""Rama killed Ravana to save sita from Lanka. The legend of the

Ramayan is the most popular Indian epic. A lot of movies and serials have already

been shot in several languages here in India based on the Ramayana. """

tokeni zed = nl tk.sent\_ tokeni ze(sampl e\_ text)

for i in tokenized:

words = nltk.word\_ tokenize(i)

tagged\_ words = nl tk.pos\_ tag(words)

chunkGram =r"""V B: {}"""

chunkParser = nl tk.Regex pParser( chunkGram)

chunked = chunkParser. parse( tagged\_ words)

print(chunked)

chunked.draw()

Output:

20MCA241 Data Science Lab

Dept. of Computer A pplications



43

20MCA241 Data Science Lab

Dept. of Computer A pplications

