

20MCA241 DATA SCIENCE LAB

Lab Report Submitted By

ALEENA JOSEPH

Reg. No.: AJC20MCA-2008

In Partial fulfillment for the Award of the Degree Of

**MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Lab report, “**20MCA241 DATA SCIENCE LAB**” is the bonafide work of **ALEENA JOSEPH (Reg.No:AJC20MCA-2008)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Ms. Meera Rose Mathew

Lab In-Charge

CONTENT

S.No	Content	Date	Page No
1	Perform all matrix operation using python	24/11/2021	1
2	Program to perform SVD using python	01/12/2021	3
3	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	01/12/2021	4
4	Program to implement k-NN Classification using any random dataset without using in-build functions	01/12/2021	6
5	Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm	08/12/2021	8
6	Program to implement linear and multiple regression techniques using any standard dataset available in the public domain	08/01/2022	10
7	Program to implement Linear and Multiple regression techniques using any standard dataset available in public domain and evaluate its performance	15/01/2022	12
8	Program to implement Linear and Multiple regression techniques using cars dataset available in public domain and evaluate its performance	15/01/2022	14
9	Program to implement multiple linear regression techniques using Boston dataset available in the public domain and evaluate its performance and plotting graph.	15/01/2022	16
10	Program to implement decision tree using any standard dataset available in the public domain and find the accuracy of the algorithm	22/12/2021	18

11	Program to implement K-Means clustering technique using any standard dataset available in the public domain	05/01/2022	22
12	Program to implement K-Means clustering technique using any standard dataset available in the public domain	05/01/2022	25
13	Programs on convolutional neural network to classify images from any standard dataset in the public domain	02/02/2022	27
14	Program to implement a simple web crawler using python	16/02/2022	31
15	Program to implement a simple web crawler using python	16/02/2022	34
16	Program to implement scrap of any website	16/02/2022	36
17	Program for Natural Language Processing which performs n-grams	16/02/2022	38
18	Program for Natural Language Processing which performs n-grams (Using in built functions)	16/02/2022	39
19	Program for Natural Language Processing which performs speech tagging	16/02/2022	40
20	Write python program for natural language processing which perform chunking	23/02/2022	41
21	Write python program for natural language processing which perform chunking	23/02/2022	42

PROGRAM NO: 01**Date: 24/11/2021****AIM: Perform all matrix operation using python****Program Code:**

```
import numpy

x = numpy.array([[1, 2], [3, 4]])
y = numpy.array([[5, 6], [7, 8]])

print ("The matrices are: ")
print ("First matrix:")
print (x)
print ("Second matrix: ")
print(y)

#Addition--- add()
print ("matrix addition:")
print (numpy.add(x,y))

#Subtraction----- subtract()
print ("matrix Subtraction:")
print (numpy.subtract(x,y))

#Division----- divide()
print ("matrix Division")
print (numpy.divide(x,y))

#Multiplication----- multiply
print ("matrix Multiplication")
print (numpy.multiply(x,y))

#Product of matrix---- dot()
print ("Product of 2 matrix: ")
print (numpy.dot(x,y))

#Square root---- sqrt()
print ("Square root of matrix X: ")
print (numpy.sqrt(x))

#Summation ---- sum()
print ("Summation of matrix X: ")
print (numpy.sum(x,axis=0))
print ("Summation of matrix Y: ")
print (numpy.sum(y,axis=1))
```

```
#Transposition ---- T
print ("Transposition of matrix X: ")
print(x.T)
print ("Transposition of matrix Y: ")
print(y.T)
```

Output:

```
The matrices are:
First matrix:
[[1 2]
 [3 4]]
Second matrix:
[[5 6]
 [7 8]]
matrix addition:
[[ 6  8]
 [10 12]]
matrix Subtraction:
[[-4 -4]
 [-4 -4]]
matrix Division
[[0.2      0.33333333]
 [0.42857143 0.5      ]]
matrix Multiplication
[[ 5 12]
 [21 32]]
Product of 2 matrix:
[[19 22]
 [43 50]]
```

```
[40 50]]
Square root of matrix X:
[[1.      1.41421356]
 [1.73205081 2.      ]]
Square root of matrix X:
[[1.      1.41421356]
 [1.73205081 2.      ]]
Summation of matrix X:
[4 6]
Summation of matrix Y:
[11 15]
Transposition of matrix X:
[[1 3]
 [2 4]]
Transposition of matrix Y:
[[5 7]
 [6 8]]

Process finished with exit code 0
```

PROGRAM NO: 02**Date:** 01/12/2021**AIM:** Program to perform SVD using python**Program Code:**

```

from numpy import array
from scipy.linalg import svd          #a function in SCIPY
A = ([[8,4,5,7], [4,1,6,9], [6,1,0,9]]);

print("ACTUAL MATRIX IS: ")
print(A);
U, s, VT = svd(A)                    # U, s, VT are just 3 variables
                                     # U= decomposed s=inverse #VT=transpos

print("DECOMPOSED MATRIX: ")
print(U);

print("INVERSE MATRIX: ")
print(s);

print("TRANSPOS MATRIX: ")
print(VT);

```

OUTPUT:

```

C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users
ACTUAL MATRIX IS:
[[8, 4, 5, 7], [4, 1, 6, 9], [6, 1, 0, 9]]
DECOMPOSED MATRIX:
[[-0.62412923  0.35812463 -0.69441303]
 [-0.57698231  0.38802636  0.7186981 ]
 [-0.52683405 -0.84922452  0.03554702]]
INVERSE MATRIX:
[19.24390834  4.59023347  3.82122341]
TRANSPOS MATRIX:
[[-0.54365086 -0.18708951 -0.34205838 -0.74326128]
 [-0.14775821  0.21160151  0.89729234 -0.35813234]
 [-0.64566487 -0.52951811  0.21985719  0.50437116]
 [-0.51548824  0.79989555 -0.17182941  0.25478155]]

Process finished with exit code 0

```

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 KNeighborsClassifier
from sklearn.model_selection import train_test_split

#Split arrays or matrices into random train and test subsets

from sklearn.datasets import load_iris

#Load and return the iris dataset (classification).

from sklearn.metrics import accuracy_score

# to load dataset values
dataset = load_iris()

# features & target
d = dataset.data          # feature
t = dataset.target        # target

d_train, d_test, t_train, t_test = train_test_split(d, t, test_size=0.2, random_state=40)

knn = KNeighborsClassifier(n_neighbors=10)

knn.fit(d_train, t_train)

#Fit the k-nearest neighbors classifier from the training dataset.

print(knn.predict(d_test))

a = knn.predict(d_test)
ac = accuracy_score(t_test, a)  #store accuracy value
print("Accuracy value is : ")

print(ac)

```


Output:

```
C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/KNN.py  
[0 1 1 0 0 2 0 1 1 1 1 2 0 2 0 0 0 0 2 2 0 2 0 1 2 0 1 2 2 2]  
Accuracy value is :  
0.9  
  
Process finished with exit code 0
```

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 euclidian_distance(row1, row2):

    distance = 0.0

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

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

    return sqrt(distance)

# locat the most similar neighbor

def get_neighbors(train, test_row, num_neighbors):

    distances = list()

    for train_row in train:

        dist = euclidian_distance(test_row, train_row)

        distances.append((train_row, dist))

    distances.sort(key=lambda tup: tup[1])

    neighbors = list()

    for i in range(num_neighbors):

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

    return neighbors

#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]          #store the data of neighbors

    prediction = max(set(output_values), key=output_values.count)

    return prediction

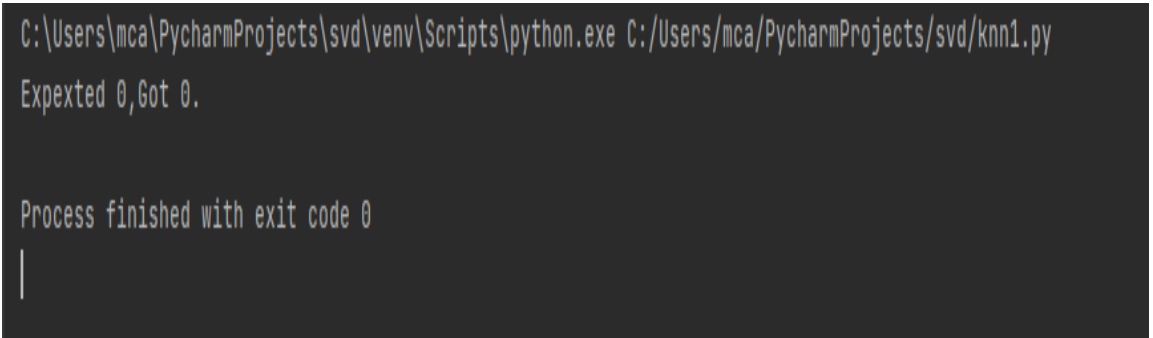
```

```
# test distance function

dataset = [[2.5477838, 2.753590, 0],
[1.45778788,2.7767373, 0],
[3.678838, 4.6788288, 0],
[1.436773, 1.53773, 0],
[3.76888389, 3.6748, 0],
[7.7848848, 2.759256, 1],
[5.782356, 2.246378, 1],
[6.777878, 1.49078, 1],
[8.677728889, -0.7588392, 1],
[7.675637, 3.59340, 1]]

prediction = predict_classification(dataset, dataset[0], 3)

print("Expexted %d,Got %d."%(dataset[0][-1],prediction))
```

Output:

```
C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/knn1.py
Expexted 0,Got 0.

Process finished with exit code 0
|
```

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:

```
# Random Forest Classification

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Fitting Random Forest Classification to the Training set
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

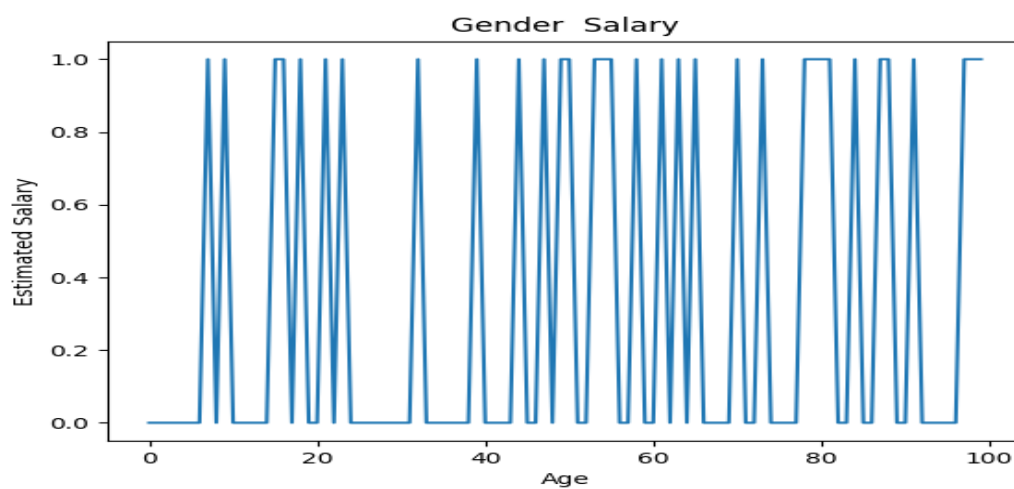
# Predicting the Test set results
y_pred = classifier.predict(X_test)
print(y_pred)
plt.plot(y_pred)
plt.title('Gender Salary')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
#plt.legend()
plt.show()
```

Output:

```

C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/nb.py
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 1
 0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1]

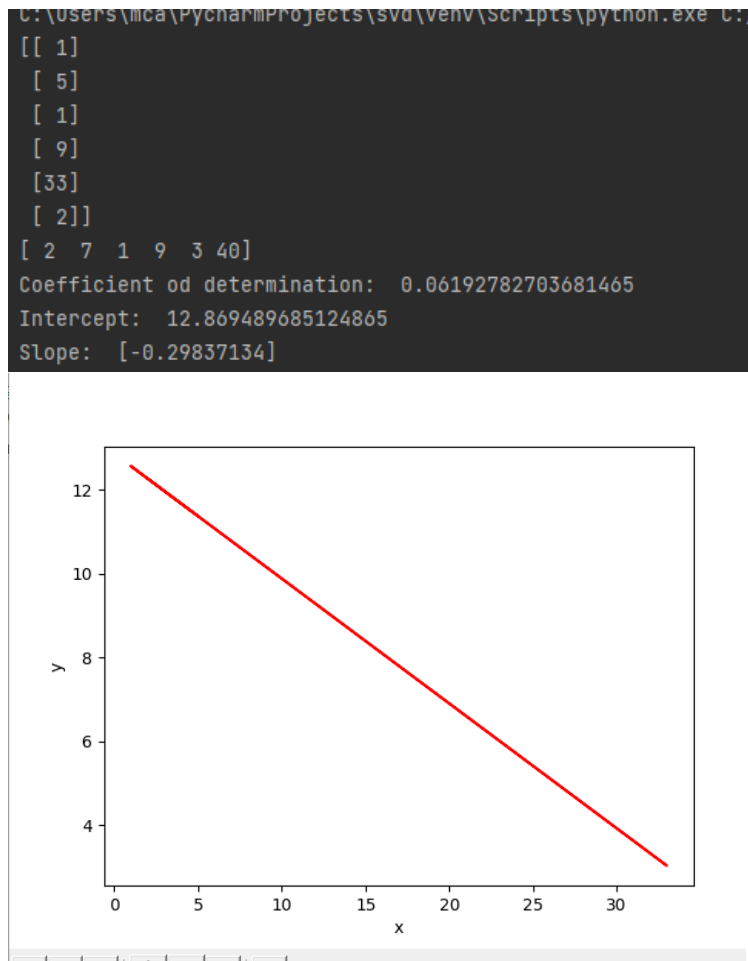
```



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 matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
x = np.array([1,5,1,9,33,2]).reshape((-1, 1))
y = np.array([2,7,1,9,3,40])
print(x)
print(y)
model=LinearRegression()
model.fit(x, y)
r_sq=model.score(x,y)
print('Coefficient of determination: ', r_sq)
print('Intercept: ', model.intercept_)
print('Slope: ', model.coef_)
y_pred=model.predict(x)

plt.plot(x, y_pred, color="r")
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Output:

PROGRAM NO: 07**Date: 15/12/2021****AIM: Program to implement linear and multiple regression techniques using any standard dataset available in the public domain****Program Code:**

```
import numpy as np
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="g",
               marker="o", s=30)

    # predicted response vector
    y_pred = b[0] + b[1] * x

    # plotting the regression line
    plt.plot(x, y_pred, color="r")

    # putting labels
    plt.xlabel('x')
    plt.ylabel('y')

    # function to show plot
    plt.show()
```



```
def main():
# observations / data
x = np.array([0, 2, 2, 3, 5, 5, 6, 8, 9, 10])
y = np.array([1, 3, 4, 6, 8, 10, 12, 13, 14, 16])

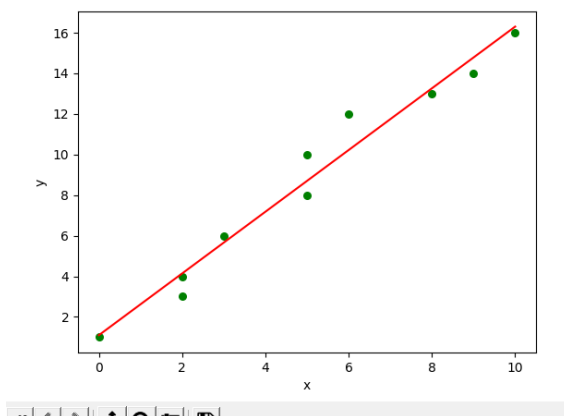
# estimating coefficients
b = estimate_coef(x, y)
print("Estimated coefficients:\nb_0 = { } \
\nb_1 = { }".format(b[0], b[1]))

# plotting regression line
plot_regression_line(x, y, b)

if __name__ == "__main__":
main()
```

Output:

```
C:\Users\mca\PycharmProjects\svd\venv\Scripts\pyt
Estimated coefficients:
b_0 = 1.0979591836734688
b_1 = 1.5204081632653061
```



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 as pd

import numpy as np

import matplotlib.pyplot as plt

dataset = pd.read_csv("cars.csv")

dataset.head()

dataset.describe()

X = dataset[['Weight', 'Volume']]

y = dataset['CO2']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X_train, y_train)

r2_score = regressor.score(X_test, y_test)

print("Accuracy: ")

print(r2_score*100,'%')


coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])

coeff_df

print("co-efficient of correlation: ")

print(regressor.coef_)
```

Output:

```
C:\Users\ajcemca\PycharmProjects\aleena\venv\Scripts\  
Accuracy:  
40.61589718966062 %  
co-efficient of correlation:  
[0.00728963 0.0076251 ]  
|  
Process finished with exit code 0
```

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

from sklearn import datasets, linear_model, metrics

from sklearn.metrics import mean_squared_error, r2_score

boston = datasets.load_boston(return_X_y=False)

X = boston.data

y = boston.target

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

reg = linear_model.LinearRegression()

reg.fit(X_train, y_train)

predicted = reg.predict(X_test)

# Regression coefficient

print('Coefficients are:\n', reg.coef_)

# Intercept

print("\nIntercept : ', reg.intercept_)

# variance score: 1 means perfect prediction

print('Variance score: ', reg.score(X_test, y_test))


# Mean Squared Error

print("Mean squared error: %.2f" % mean_squared_error(y_test, predicted))

# Original data of X_test

expected = y_test
```

```
# Plot a graph for expected and predicted values

plt.title('BOSTON Dataset')

plt.scatter(expected, predicted, c='b', marker='.', s=36)

plt.plot([0, 50], [0, 50], '--r')

plt.xlabel('Actual Price')

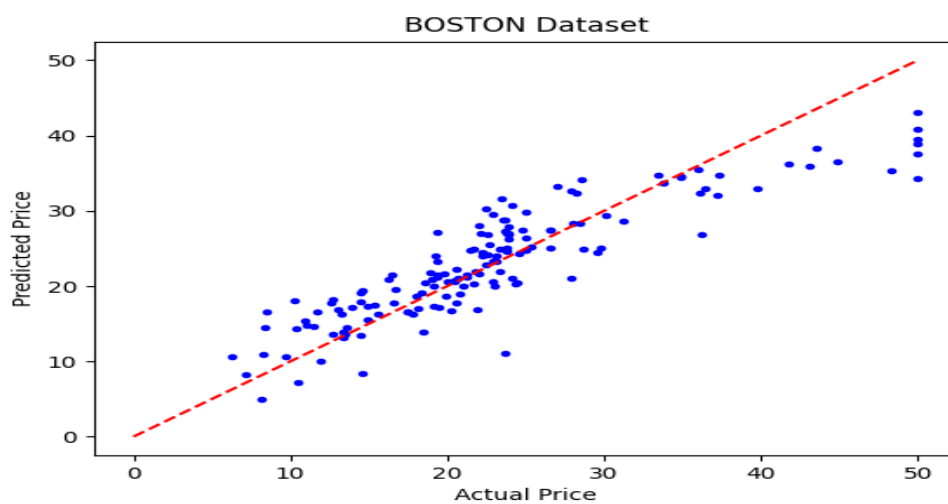
plt.ylabel('Predicted Price')

plt.show()
```

Output:

```
Coefficients are:
[-9.85424717e-02  6.07841138e-02  5.91715401e-02  2.43955988e+00
 -2.14699650e+01  2.79581385e+00  3.57459778e-03 -1.51627218e+00
  3.07541745e-01 -1.12800166e-02 -1.00546640e+00  6.45018446e-03
 -5.68834539e-01]
```

```
Intercept : 46.39649387182355
Variance score: 0.7836295385076291
Mean squared error: 19.83
```



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

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import plot_tree
df = sns.load_dataset('iris')
print(df.head())
print(df.info())
df.isnull().any()
print(df.shape)

sns.pairplot(data=df, hue = 'species')
plt.savefig("pne.png")

#correlation matrix
sns.heatmap(df.corr())
plt.savefig("one.png")
target = df['species']
df1 = df.copy()
df1 = df1.drop('species', axis=1)
print(df1.shape)
print(df1.head())

#defining attributes
x=df1
print(target)

#label encoding
le = LabelEncoder()
target = le.fit_transform(target)    #learn scaling parameters(species)
print(target)
y=target
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

print("Training split input: ", x_train.shape)
print("Testing split input: ", x_test.shape)
```

```

#defining the decision tree algorithm
dtree = DecisionTreeClassifier()
dtree.fit(x_train, y_train)
print('Decision tree classifier created')

#predicting the value of test data
y_pred = dtree.predict(x_test)
print("Classification report: \n", classification_report(y_test,y_pred))
cm = confusion_matrix(y_test,y_pred)
plt.figure(figsize=(5,5))

sns.heatmap(data=cm,linewidths=.5,annot=True,square=True,cmap='Blues')

plt.ylabel('Actual label')
plt.xlabel('Predicted label')

all_sample_title = 'Accuracy score: {0}'.format(dtree.score(x_test, y_test))
plt.title(all_sample_title, size=15)
plt.savefig("two.png")

plt.figure(figsize=(20,20))
dec_tree = plot_tree(decision_tree=dtree,feature_names=df1.columns,class_names=["setosa",
"vercolor", "virginica"], filled=True,precision=4,rounded=True)
plt.savefig("three.png")

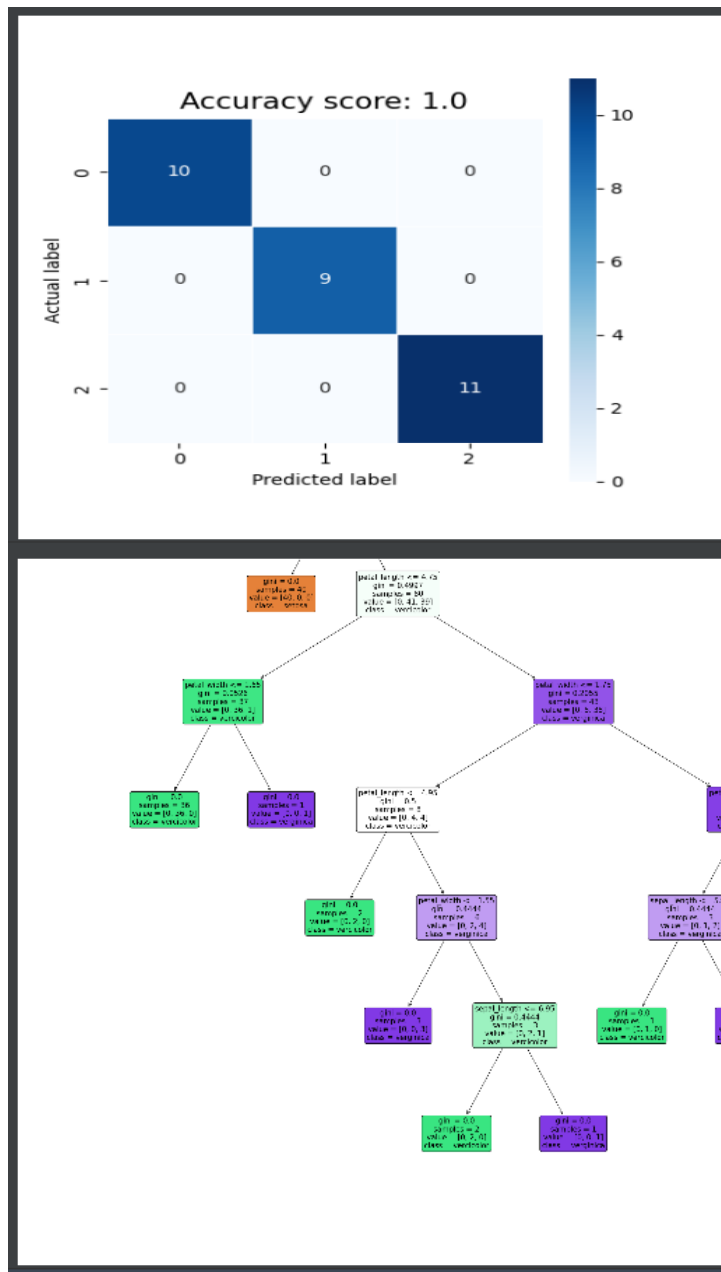
```

Output:

```

   sepal_length  sepal_width  petal_length  petal_width species
0         5.1         3.5         1.4         0.2  setosa
1         4.9         3.0         1.4         0.2  setosa
2         4.7         3.2         1.3         0.2  setosa
3         4.6         3.1         1.5         0.2  setosa
4         5.0         3.6         1.4         0.2  setosa
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   sepal_length    150 non-null    float64
 1   sepal_width     150 non-null    float64
 2   petal_length    150 non-null    float64
 3   petal_width     150 non-null    float64
 4   species         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 5.3+ KB
None
(150, 5)
(150, 4)
   sepal_length  sepal_width  petal_length  petal_width
0         5.1         3.5         1.4         0.2
1         4.9         3.0         1.4         0.2
2         4.7         3.2         1.3         0.2
3         4.6         3.1         1.5         0.2
4         5.0         3.6         1.4         0.2
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
...
145  virginica
146  virginica
147  virginica
148  virginica
149  virginica

```

PROGRAM NO: 11**Date: 05/01/2021**

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('Mall_Customers.csv')

x=dataset.iloc[:,[3,4]].values

print(x)

from sklearn.cluster import KMeans

wcss_list=[]

for i in range(1,11):

kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)

kmeans.fit(x)

wcss_list.append(kmeans.inertia_)

mtp.plot(range(1,11),wcss_list)

mtp.title('The Elbow Method Graph')

mtp.xlabel('Number of clusters(k)')

mtp.ylabel('wcss_list')

mtp.show()

kmeans=KMeans(n_clusters=5,init='k-means++',random_state=42)

y_predict=kmeans.fit_predict(x)

print(y_predict)
```

```

mtp.scatter(x[y_predict ==0,0],x[y_predict ==0,1],s=100,c='blue',label='cluster 1')
mtp.scatter(x[y_predict ==1,0],x[y_predict ==1,1],s=100,c='green',label='cluster 2')
mtp.scatter(x[y_predict ==2,0],x[y_predict ==2,1],s=100,c='red',label='cluster 3')
mtp.scatter(x[y_predict ==3,0],x[y_predict ==3,1],s=100,c='cyan',label='cluster 4')
mtp.scatter(x[y_predict ==4,0],x[y_predict ==4,1],s=100,c='magenta',label='cluster 5')

mtp.scatter(kmeans.cluster_centers_[0,0],kmeans.cluster_centers_[0,1],s=300,c='black',label='cluster')

mtp.title('Clusters of customers')
mtp.xlabel('Annual Income (K$)')
mtp.ylabel('Spending Score(1-100)')
mtp.legend()
mtp.show()

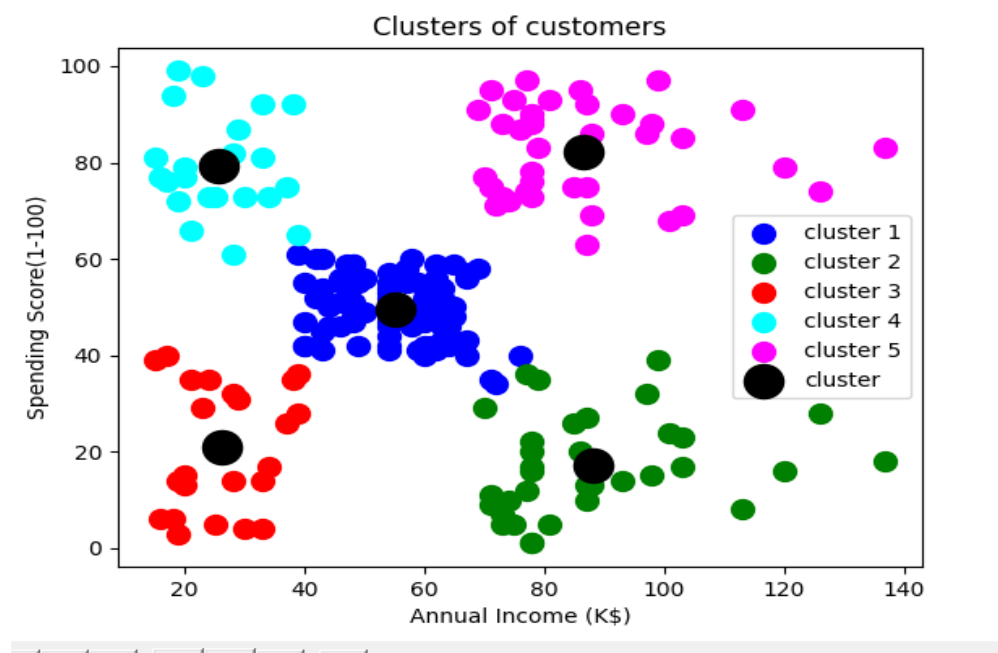
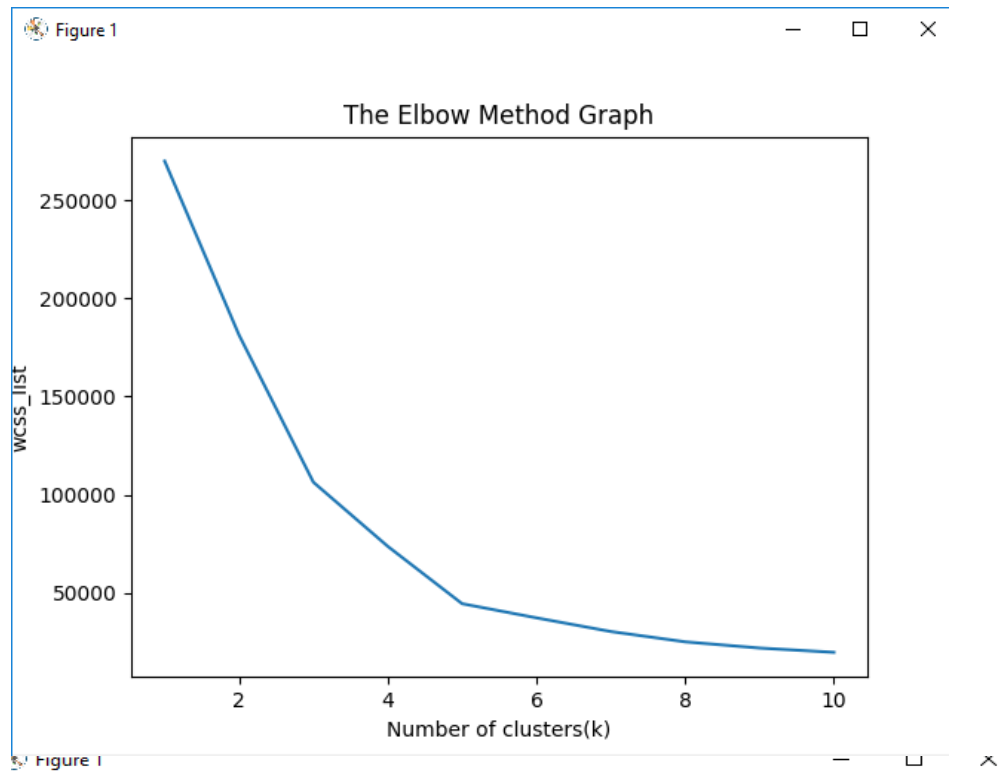
```

Output:

```

C:\Users\ajcemca\PycharmProje
[[ 15  39]
 [ 15  81]
 [ 16   6]
 [ 16  77]
 [ 17  40]
 [ 17  76]
 [ 18   6]
 [ 18  94]
 [ 19   3]
 [ 19  72]
 [ 19  14]
 [ 19  99]
 [ 20  15]
 [ 20  77]
 [ 20  13]

```



PROGRAM NO: 12**Date:** 05/01/2021**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('world_country_and_usa_states_latitude_and_longitude_values.csv')
x=dataset.iloc[:,[1,2]].values
print(x)

from sklearn.cluster import KMeans
wcss_list=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
    mtp.plot(range(1,11),wcss_list)
    mtp.title('The Elbow Method Graph')
    mtp.xlabel('Number of clusters(k)')
    mtp.ylabel('wcss_list')

mtp.show()

kmeans=KMeans(n_clusters=3,init='k-means++',random_state=42)
y_predict=kmeans.fit_predict(x)
print(y_predict)

mtp.scatter(x[y_predict ==0,0],x[y_predict ==0,1],s=100,c='blue',label='cluster 1')
mtp.scatter(x[y_predict ==1,0],x[y_predict ==1,1],s=100,c='green',label='cluster 2')
mtp.scatter(x[y_predict ==2,0],x[y_predict ==2,1],s=100,c='red',label='cluster 3')
mtp.scatter(kmeans.cluster_centers_[0,0],kmeans.cluster_centers_[0,1],s=300,c='black',label='cluster')
mtp.title('Clusters of customers')
mtp.xlabel('Annual Income (K$)')
mtp.ylabel('Spending Score(1-100)')
mtp.legend()
mtp.show()

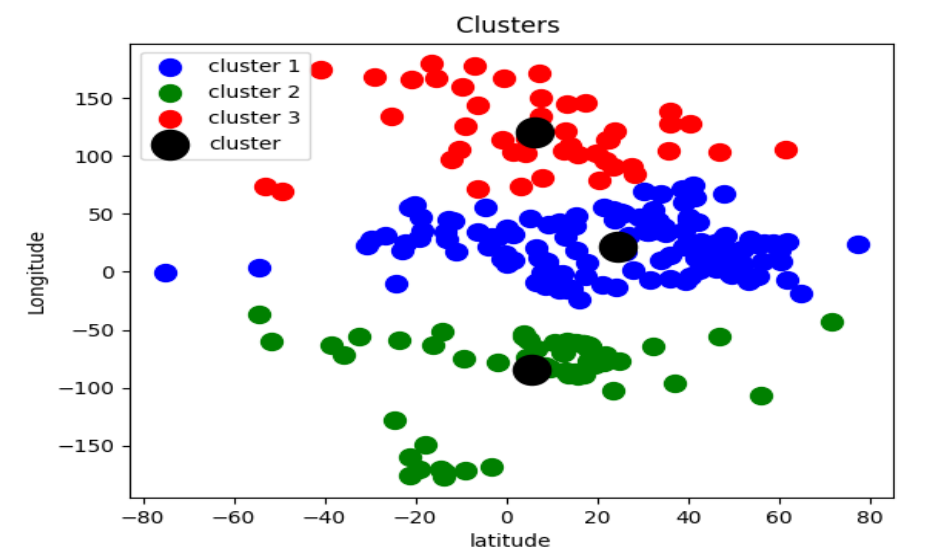
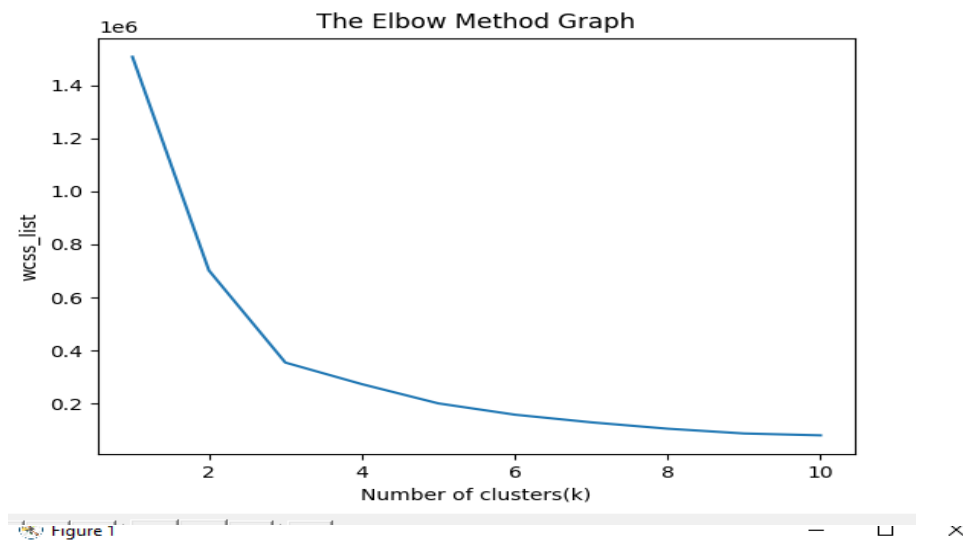
```

Output:

```

C:\Users\ajcemca\PycharmProjects\Rmca_DLMLLab_28.
[[ 4.25462450e+01  1.60155400e+00]
 [ 2.34240760e+01  5.38478180e+01]
 [ 3.39391100e+01  6.77099530e+01]
 [ 1.70608160e+01 -6.17964280e+01]
 [ 1.82205540e+01 -6.30686150e+01]
 [ 4.11533320e+01  2.01683310e+01]
 [ 4.00690990e+01  4.50381890e+01]
 [ 1.22260790e+01 -6.90600870e+01]
 [-1.12026920e+01  1.78738870e+01]
 [-7.52509730e+01 -7.13890000e-02]
 [-3.84160970e+01 -6.36166720e+01]
 [-1.42709720e+01 -1.70132217e+02]
 [ 4.75162310e+01  1.45500720e+01]
 [-2.52743980e+01  1.33775136e+02]
 [ 1.25211100e+01 -6.99683380e+01]
 [ 4.01431050e+01  4.75769270e+01]
 [ 4.39158860e+01  1.76790760e+01]
 [ 1.31938870e+01 -5.95431980e+01]
 [ 2.36849940e+01  9.03563310e+01]

```



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)

fashion_mnist=keras.datasets.fashion_mnist

(x_train,y_train),(x_test,y_test)=fashion_mnist.load_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.unique(y_test)

class_names=['T-shirt/Top','Trouser','Pullover','Dress','Coat','Sandal','Shirt','Sneaker','Bag','Ankle
Boot']

n_rows=5

n_cols=10

plt.figure(figsize=(n_cols * 1.4,n_rows * 1.6))

for row in range(n_rows):

for col in range(n_cols):

index=n_cols * row +col
```

```

plt.subplot(n_rows,n_cols,index+1)

plt.imshow(x_train[index],cmap='binary',interpolation='nearest')

plt.axis('off')

plt.title(class_names[y_train[index]])

plt.show()

model_CNN=keras.models.Sequential()

model_CNN.add(keras.layers.Conv2D(filters=32,kernel_size=7,padding='same',activation='relu',i
nput_shape=[28,28,1]))

model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))

model_CNN.add(keras.layers.Conv2D(filters=64,kernel_size=3,padding='same',activation='relu'))

model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))

model_CNN.add(keras.layers.Conv2D(filters=32,kernel_size=3,padding='same',activation='relu'))

model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))

model_CNN.summary()

model_CNN.add(keras.layers.Flatten())

model_CNN.add(keras.layers.Dense(units=128,activation='relu'))

model_CNN.add(keras.layers.Dense(units=64,activation='relu'))

model_CNN.add(keras.layers.Dense(units=10,activation='softmax'))

model_CNN.summary()

model_CNN.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['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(history_CNN.history).plot()

plt.grid(True)

plt.xlabel('epochs')

```



```

plt.ylabel('loss/accuracy')

plt.title("Training and validation plot")

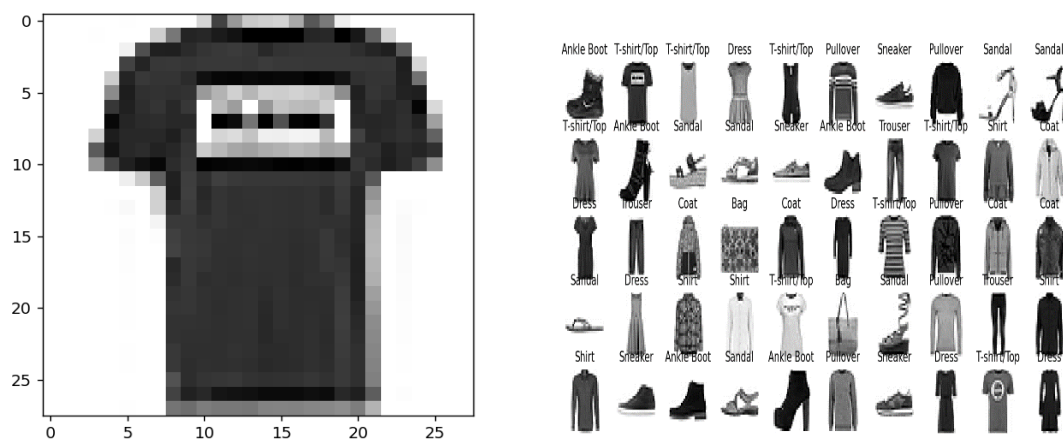
plt.show()

test_loss,test_accuracy=model_CNN.evaluate(x_test,y_test)

print("Test Loss:{ }','Test Accuracy:{ }'.format(test_loss,test_accuracy))

```

Output:



```

(60000, 28, 28) (10000, 28, 28)
2022-02-02 12:03:16.761271: W tensorflow/stream_executor/platform/default/dso_loader.cc:43: Warning:
2022-02-02 12:03:16.763256: W tensorflow/core/common_runtime/gpu/gpu_device.cc:171: Cannot
Skipping registering GPU devices...
2022-02-02 12:03:16.773939: I tensorflow/core/platform/cpu_feature_guard.cc:151: This
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)              (None, 28, 28, 32)       1600
max_pooling2d (MaxPooling2D) (None, 14, 14, 32)       0
conv2d_1 (Conv2D)            (None, 14, 14, 64)       18496
max_pooling2d_1 (MaxPooling2D) (None, 7, 7, 64)         0
conv2d_2 (Conv2D)            (None, 7, 7, 32)         18464
max_pooling2d_2 (MaxPooling2D) (None, 3, 3, 32)         0
-----
Total params: 38,560
Trainable params: 38,560
Non-trainable params: 0

```

```

Trainable params: 38,560
Non-trainable params: 0
-----
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)             (None, 28, 28, 32)       1600
max_pooling2d (MaxPooling2D) (None, 14, 14, 32)       0
conv2d_1 (Conv2D)           (None, 14, 14, 64)       18496
max_pooling2d_1 (MaxPooling2D) (None, 7, 7, 64)       0
conv2d_2 (Conv2D)           (None, 7, 7, 32)         18464
max_pooling2d_2 (MaxPooling2D) (None, 3, 3, 32)         0
flatten (Flatten)           (None, 288)              0
dense (Dense)               (None, 128)              36992
dense_1 (Dense)             (None, 64)               8256
dense_2 (Dense)             (None, 10)               650

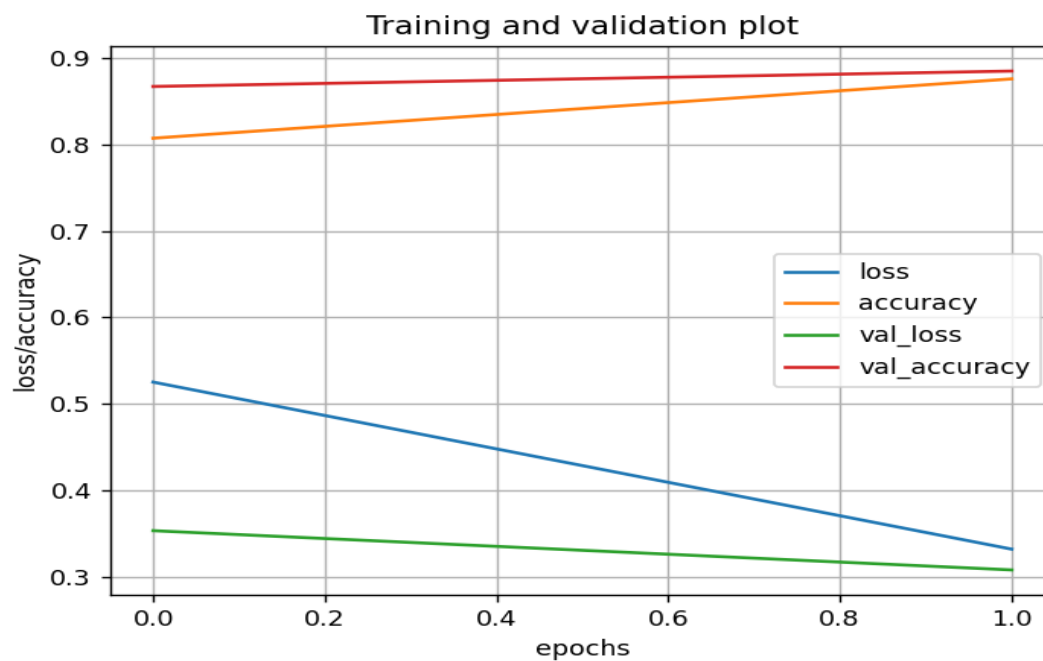
```

```

=====
Total params: 84,458
Trainable params: 84,458
Non-trainable params: 0
-----
Epoch 1/2
1688/1688 [=====] - 74s 43ms/step - loss: 0.5097 - accuracy: 0.8133 - val_loss: 0.3481 - val_accuracy: 0.8688
Epoch 2/2
1688/1688 [=====] - 73s 43ms/step - loss: 0.3272 - accuracy: 0.8795 - val_loss: 0.3289 - val_accuracy: 0.8763
313/313 [=====] - 4s 13ms/step - loss: 0.3441 - accuracy: 0.8721
Test Loss :0.34412604570388794, Test Accuracy : 0.8720999956130981

Process finished with exit code 0

```



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
url = "https://rottentomatoes.com/top/bestofrt/"
header = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/63.0.3239.132 Safari/537.36 QIHU 360SE'
}
f = requests.get(url,headers=header)
movies_lst = []
soup = BeautifulSoup(f.content, 'lxml')
movies = soup.find('table',{
    'class': 'table'
}).find_all('a')
print(movies)
num = 0
for anchor in movies:
    urls = 'https://rottentomatoes.com' + anchor['href']
    movies_lst.append(urls)
    print(movies_lst)
    num += 1
    movie_url = urls
    movie_f = requests.get(movie_url, headers=header)
    movie_soup = BeautifulSoup(movie_f.content, 'lxml')
    movie_content = movie_soup.find('div', {
        'class': 'movie_synopsis clamp clamp-6 js-clamp'
    })
    print(num, urls, '\n', 'Movies: ' + anchor.string.strip())
    print('Movies info: ' + movie_content.string.strip())

```

Output:

```

C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/Lab/webscrap.py
[<a class="unstyled articleLink" href="/m/it_happened_one_night">
    It Happened One Night (1934)</a>, <a class="unstyled articleLink" href="/m/citizen_kane">
    Citizen Kane (1941)</a>, <a class="unstyled articleLink" href="/m/the_wizard_of_oz_1939">
    The Wizard of Oz (1939)</a>, <a class="unstyled articleLink" href="/m/modern_times">
    Modern Times (1936)</a>, <a class="unstyled articleLink" href="/m/black_panther_2018">
    Black Panther (2018)</a>, <a class="unstyled articleLink" href="/m/parasite_2019">
    Parasite (Gisaengchung) (2019)</a>, <a class="unstyled articleLink" href="/m/avengers_endgame">
    Avengers: Endgame (2019)</a>, <a class="unstyled articleLink" href="/m/1003707-casablanca">
    Casablanca (1942)</a>, <a class="unstyled articleLink" href="/m/knives_out">
    Knives Out (2019)</a>, <a class="unstyled articleLink" href="/m/us_2019">
    Us (2019)</a>, <a class="unstyled articleLink" href="/m/toy_story_4">
    Toy Story 4 (2019)</a>, <a class="unstyled articleLink" href="/m/lady_bird">
    Lady Bird (2017)</a>, <a class="unstyled articleLink" href="/m/mission_impossible_fallout">
    Mission: Impossible - Fallout (2018)</a>, <a class="unstyled articleLink" href="/m/blackkklansman">
    BlackKkKlansman (2018)</a>, <a class="unstyled articleLink" href="/m/get_out">
    Get Out (2017)</a>, <a class="unstyled articleLink" href="/m/the_irishman">
    The Irishman (2019)</a>, <a class="unstyled articleLink" href="/m/godfather">
    The Godfather (1972)</a>, <a class="unstyled articleLink" href="/m/mad_max_fury_road">

```

```

Shadow of a Doubt (1943)</a>, <a class="unstyled articleLink" href="/m/call_me_by_your_name">
Call Me by Your Name (2018)</a>, <a class="unstyled articleLink" href="/m/psycho">
Psycho (1960)</a>, <a class="unstyled articleLink" href="/m/1917_2019">
1917 (2020)</a>, <a class="unstyled articleLink" href="/m/la_confidential">
L.A. Confidential (1997)</a>, <a class="unstyled articleLink" href="/m/the_florida_project">
The Florida Project (2017)</a>, <a class="unstyled articleLink" href="/m/war_for_the_planet_of_the_apes">
War for the Planet of the Apes (2017)</a>, <a class="unstyled articleLink" href="/m/paddington_2">
Paddington 2 (2018)</a>, <a class="unstyled articleLink" href="/m/beatles_a_hard_days_night">
A Hard Day's Night (1964)</a>, <a class="unstyled articleLink" href="/m/widows_2018">
Widows (2018)</a>, <a class="unstyled articleLink" href="/m/never_rarely_sometimes_always">
Never Rarely Sometimes Always (2020)</a>, <a class="unstyled articleLink" href="/m/baby_driver">
Baby Driver (2017)</a>, <a class="unstyled articleLink" href="/m/spider_man_homecoming">
Spider-Man: Homecoming (2017)</a>, <a class="unstyled articleLink" href="/m/godfather_part_ii">
The Godfather, Part II (1974)</a>, <a class="unstyled articleLink" href="/m/the_battle_of_algiens">
The Battle of Algiers (La Battaglia di Algeri) (1967)</a>]
The Battle of Algiers (La Battaglia di Algeri) (1967)</a>]
['https://rottentomatoes.com/m/it_happened_one_night']
1 https://rottentomatoes.com/m/it_happened_one_night
Movies: It Happened One Night (1934)
Movies info: In Frank Capra's acclaimed romantic comedy, spoiled heiress Ellie Andrews (Claudette Colbert) impetuously marries the scheming King Westley, T
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane']
2 https://rottentomatoes.com/m/citizen_kane
Movies: Citizen Kane (1941)
Movies info: When a reporter is assigned to decipher newspaper magnate Charles Foster Kane's (Orson Welles) dying words, his investigation gradually reveal
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_wizard_of_oz_1939']
3 https://rottentomatoes.com/m/the_wizard_of_oz_1939
Movies: The Wizard of Oz (1939)
Movies info: When a tornado rips through Kansas, Dorothy (Judy Garland) and her dog, Toto, are whisked away in their house to the magical land of Oz. They
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_wizard_of_oz_1939', '
4 https://rottentomatoes.com/m/modern_times
Movies: Modern Times (1936)
Movies info: This comedic masterpiece finds the iconic Little Tramp (Charlie Chaplin) employed at a state-of-the-art factory where the inescapable machiner
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_wizard_of_oz_1939', '
5 https://rottentomatoes.com/m/black_panther_2018
Movies: Black Panther (2018)
Movies info: After the death of his father, T'Challa returns home to the African nation of Wakanda to take his rightful place as king. When a powerful ene
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_wizard_of_oz_1939', '
6 https://rottentomatoes.com/m/parasite_2019
Movies: Parasite (Gisaengchung) (2019)
Movies info: Greed and class discrimination threaten the newly formed symbiotic relationship between the wealthy Park family and the destitute Kim clan.
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_wizard_of_oz_1939', '

Movies info: Now that Chris and his girlfriend, Rose, have reached the meet-the-parents milestone of datin
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
16 https://rottentomatoes.com/m/the_irishman
Movies: The Irishman (2019)
Movies info: In the 1950s, truck driver Frank Sheeran gets involved with Russell Bufalino and his Pennsylv
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
17 https://rottentomatoes.com/m/godfather
Movies: The Godfather (1972)
Movies info: Widely regarded as one of the greatest films of all time, this mob drama, based on Mario Puzo
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
18 https://rottentomatoes.com/m/mad_max_fury_road
Movies: Mad Max: Fury Road (2015)
Movies info: Years after the collapse of civilization, the tyrannical Immortan Joe enslaves apocalypse sur
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
19 https://rottentomatoes.com/m/spider_man_into_the_spider Verse
Movies: Spider-Man: Into the Spider-Verse (2018)
Movies info: Bitten by a radioactive spider in the subway, Brooklyn teenager Miles Morales suddenly develo
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
20 https://rottentomatoes.com/m/moonlight_2016
Movies: Moonlight (2016)
Movies info: A look at three defining chapters in the life of Chiron, a young black man growing up in Miam
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https
21 https://rottentomatoes.com/m/sunset_boulevard
Movies: Sunset Boulevard (1950)
Movies info: An aging silent film queen refuses to accept that her stardom has ended. She hires a young sc
['https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https

```

```
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/call_me_by_your_name' ]
87 https://rottentomatoes.com/m/call_me_by_your_name
Movies: Call Me by Your Name (2018)
Movies info: It's the summer of 1983, and precocious 17-year-old Elio Perlman is spending the days with his father's friend, a charming and handsome boy who happens to be gay.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/psycho' ]
88 https://rottentomatoes.com/m/psycho
Movies: Psycho (1960)
Movies info: Phoenix secretary Marion Crane (Janet Leigh), on the lam after stealing $40,000 from her employer, checks into a remote, isolated hotel where the menacing new resident, a disturbed man, has a special way of making his enemies disappear.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/1917' ]
89 https://rottentomatoes.com/m/1917
Movies: 1917 (2020)
Movies info: During World War I, two British soldiers -- Lance Cpl. Schofield and Lance Cpl. Blake -- are assigned to deliver a message to the front lines.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/la_confidential' ]
90 https://rottentomatoes.com/m/la_confidential
Movies: L.A. Confidential (1997)
Movies info: Three policemen, each with his own motives and obsessions, tackle the corruption surrounding the murder of a movie starlet in 1950s Los Angeles.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_florida_project' ]
91 https://rottentomatoes.com/m/the_florida_project
Movies: The Florida Project (2017)
Movies info: Set in the shadow of the most magical place on Earth, 6-year-old Moonee and her two best friends spend their summer at a budget motel on the edge of Disney World.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/war_for_the_planet_of_the_apes' ]
92 https://rottentomatoes.com/m/war_for_the_planet_of_the_apes
Movies: War for the Planet of the Apes (2017)
Movies info: Caesar (Andy Serkis) and his apes are forced into a deadly conflict with an army of humans determined to destroy the apes once and for all.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/paddington_2' ]
93 https://rottentomatoes.com/m/paddington_2
Movies: Widows (2018)
Movies info: A police shootout leaves four thieves dead during an explosive armed robbery attempt in Chicago.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/never_rarely_sometimes_always' ]
96 https://rottentomatoes.com/m/never_rarely_sometimes_always
Movies: Never Rarely Sometimes Always (2020)
Movies info: Faced with an unintended pregnancy and a lack of local support, Autumn and her cousin, Skylar, travel to a distant state to get an abortion.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/baby_driver' ]
97 https://rottentomatoes.com/m/baby_driver
Movies: Baby Driver (2017)
Movies info: Talented getaway driver Baby (Ansel Elgort) relies on the beat of his personal soundtrack to be the best at his job.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/spider_man_homecoming' ]
98 https://rottentomatoes.com/m/spider_man_homecoming
Movies: Spider-Man: Homecoming (2017)
Movies info: Thrilled by his experience with the Avengers, young Peter Parker returns home to live with his aunt and uncle.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/godfather_part_ii' ]
99 https://rottentomatoes.com/m/godfather_part_ii
Movies: The Godfather, Part II (1974)
Movies info: The compelling sequel to "The Godfather," contrasting the life of Corleone father and son.
[ 'https://rottentomatoes.com/m/it_happened_one_night', 'https://rottentomatoes.com/m/citizen_kane', 'https://rottentomatoes.com/m/the_battle_of_algiers' ]
100 https://rottentomatoes.com/m/the_battle_of_algiers
Movies: The Battle of Algiers (La Battaglia di Algeri) (1967)
Movies info: Paratrooper commander Colonel Mathieu (Jean Martin), a former French Resistance fighter during World War II, is sent to 1950s Algiers to lead the French army in the fight against the Algerian National Liberation Front.

Process finished with exit code 0
```

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')
    links = soup.find_all('a')
    for link in links:
        if 'href' in link.attrs:
            if link['href'].startswith('/wiki') and ":" not in link['href']:
                if link['href'] not in pages_crawled:
                    new_link = fhttps://en.wikipedia.org{link\['href'\]}
                    pages_crawled.append(link['href'])
                    try:
                        with open('data.csv', 'a') as file:
                            file.write(f'{soup.title.text}; {soup.h1.text}; {link["href"]}\n')
                    except:
                        continue
                    crawler(new_link)

crawler("https://en.wikipedia.org")
```

Output:

```

1 Wikipedia, the free encyclopedia; Main Page; /wiki/Wikipedia
2 Wikipedia - Wikipedia; Wikipedia; /wiki/Main_Page
3 Wikipedia, the free encyclopedia; Main Page; /wiki/Free_content
4 Free content - Wikipedia; Free content; /wiki/Definition_of_Free_Cultural_Works
5 Definition of Free Cultural Works - Wikipedia; Definition of Free Cultural Works; /wiki/Free_c
6 Free-culture movement - Wikipedia; Free-culture movement; /wiki/Free_culture_(disambiguation)
7 Free Culture - Wikipedia; Free Culture; /wiki/Free_Culture_(book)
8 Free Culture (book) - Wikipedia; Free Culture (book); /wiki/Lawrence_Lessig
9 Lawrence Lessig - Wikipedia; Lawrence Lessig; /wiki/Lawrence_Lessig
0 Lawrence Lessing - Wikipedia; Lawrence Lessing; /wiki/Science_writer
1 Science journalism - Wikipedia; Science journalism; /wiki/Scientific_journalism
2 Scientific journalism - Wikipedia; Scientific journalism; /wiki/Science_journalism
3 Science journalism - Wikipedia; Science journalism; /wiki/Scientific_writing
4 Scientific writing - Wikipedia; Scientific writing; /wiki/Science_writing
5 Science journalism - Wikipedia; Science journalism; /wiki/Science_communication
6 Science communication - Wikipedia; Science communication; /wiki/Science_publishing
7 Scientific literature - Wikipedia; Scientific literature; /wiki/Medical_literature
8 Medical literature - Wikipedia; Medical literature; /wiki/Edwin_Smith_Papyrus
9 Medical literature - Wikipedia; Medical literature; /wiki/New_York_Academy_of_Medicine
0 New York Academy of Medicine - Wikipedia; New York Academy of Medicine; /wiki/Eclecticism_in_a
1 Eclecticism in architecture - Wikipedia; Eclecticism in architecture; /wiki/Basilica
2 Basilica - Wikipedia; Basilica; /wiki/Basilicas in the Catholic Church
3 Liberal Democrats - Wikipedia; Liberal Democrats; /wiki/Liberal_Democrats_(UK)
4 Liberal Democrats (UK) - Wikipedia; Liberal Democrats (UK); /wiki/Leader_of_the_Liberal_Democrats
5 Leader of the Liberal Democrats - Wikipedia; Leader of the Liberal Democrats; /wiki/Leader_of_t
6 Leader of the Liberal Party (UK) - Wikipedia; Leader of the Liberal Party (UK); /wiki/Liberal_P
7 Liberal Party (UK) - Wikipedia; Liberal Party (UK); /wiki/Liberal_Party_(UK,_1989)
8 Liberal Party (UK, 1989) - Wikipedia; Liberal Party (UK, 1989); /wiki/Party_leader
9 Party leader - Wikipedia; Party leader; /wiki/Political_party
0 Political party - Wikipedia; Political party; /wiki/Political_party_(disambiguation)
1 Political party (disambiguation) - Wikipedia; Political party (disambiguation); /wiki/Ideology
2 Ideology - Wikipedia; Ideology; /wiki/Belief
3 Belief - Wikipedia; Belief; /wiki/Belief_(disambiguation)
4 Belief (disambiguation) - Wikipedia; Belief (disambiguation); /wiki/Belief#Religion
5 Belief - Wikipedia; Belief; /wiki/Epistemology
6 Epistemology - Wikipedia; Epistemology; /wiki/Theory_of_knowledge_(disambiguation)
7 Theory of knowledge (disambiguation) - Wikipedia; Theory of knowledge (disambiguation); /wiki/T

```

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/inspirational-quotes"
r = requests.get(URL)
print(r.content)

soup = BeautifulSoup(r.content, 'xml')
print(soup.prettify())

quotes = []

table = soup.find('div', attrs={'id': 'all_quotes'})

for row in table.findAll('div',
                        attrs={'class': '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)

filename = 'insp_QT.csv'
with open(filename, 'w', newline='') as f:
    w = csv.DictWriter(f, ['theme', 'url', 'img', 'lines', 'author'])
    w.writeheader()
    for quote in quotes:
        w.writerow(quote)

```


Output:

```

1  theme,url,img,lines,author
2  LOVE,/inspirational-quotes/7444-where-there-is-love-there-is-life,https://assets.passiton.com/
3  LOVE,/inspirational-quotes/7439-at-the-touch-of-love-everyone-becomes-a-poet,https://assets.pa
4  FRIENDSHIP,/inspirational-quotes/8304-a-friend-may-be-waiting-behind-a-stranger-s-face,https:/
5  FRIENDSHIP,/inspirational-quotes/3331-whenever-we-are-it-is-our-friends-that-make,https://asse
6  FRIENDSHIP,/inspirational-quotes/8303-find-a-group-of-people-who-challenge-and,https://assets.
7  FRIENDSHIP,/inspirational-quotes/8302-there-s-not-a-word-yet-for-old-friends-who-ve,https://as
8  FRIENDSHIP,/inspirational-quotes/7435-there-are-good-ships-and-wood-ships-ships-that,https://a
9  PERSISTENCE,/inspirational-quotes/6377-at-211-degrees-water-is-hot-at-212-degrees,https://asse
10 PERSISTENCE,/inspirational-quotes/8301-the-key-of-persistence-opens-all-doors-closed,https://a
11 PERSISTENCE,/inspirational-quotes/7918-you-keep-putting-one-foot-in-front-of-the,https://asset
12 PERSISTENCE,/inspirational-quotes/7919-to-persist-with-a-goal-you-must-treasure-the,https://as
13 PERSISTENCE,/inspirational-quotes/8300-failure-cannot-cope-with-persistence,https://assets.pas
14 INSPIRATION,/inspirational-quotes/8298-though-no-one-can-go-back-and-make-a-brand-new,https://
15 INSPIRATION,/inspirational-quotes/8297-a-highly-developed-values-system-is-like-a,https://asse
16 INSPIRATION,/inspirational-quotes/7066-just-don-t-give-up-trying-what-you-really-want,https://
17 INSPIRATION,/inspirational-quotes/8296-when-we-strive-to-become-better-than-we-are,https://ass
18 INSPIRATION,/inspirational-quotes/8299-the-most-important-thing-is-to-try-and-inspire,https://
19 OVERCOMING,/inspirational-quotes/6828-bad-things-do-happen-how-i-respond-to-them,https://asset
20 OVERCOMING,/inspirational-quotes/8294-show-me-someone-who-has-done-something,https://assets.pa
21 OVERCOMING,/inspirational-quotes/6137-its-not-the-load-that-breaks-you-down-its-the,https://as
22 OVERCOMING,/insoirational-quotes/6805-getting-over-a-painful-experience-is-much-like,https://a

```

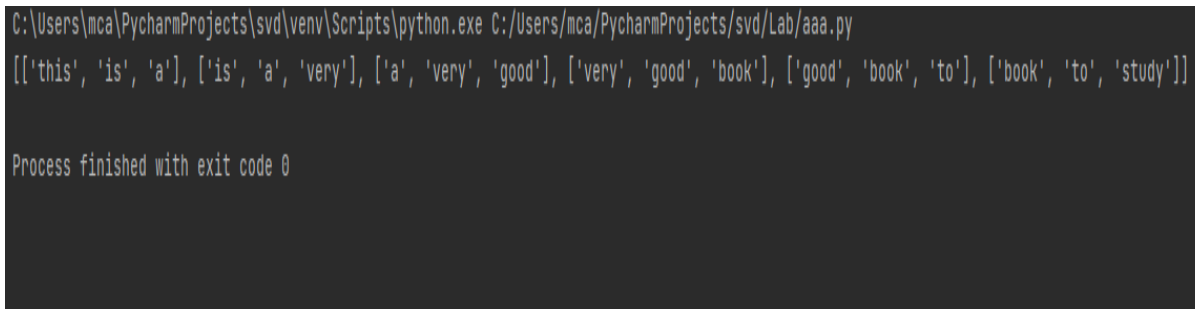
```

CREATIVITY,/inspirational-quotes/6577-the-creative-is-the-place-where-no-one-else-has,https://assets.passiton.com/quotes/quote_art
CREATIVITY,/inspirational-quotes/7345-creativity-is-allowing-yourself-to-make,https://assets.passiton.com/quotes/quote_artwork/734
CREATIVITY,/inspirational-quotes/7487-creativity-requires-the-courage-to-let-go-of,https://assets.passiton.com/quotes/quote_artwor
HUMILITY,/inspirational-quotes/8295-i-am-the-me-i-choose-to-be,https://assets.passiton.com/quotes/quote_artwork/8295/medium/202201
CREATIVITY,/inspirational-quotes/7809-creative-people-do-not-see-things-for-what-they,https://assets.passiton.com/quotes/quote_art
HOPE,/inspirational-quotes/8291-there-was-never-a-night-or-a-problem-that-could,https://assets.passiton.com/quotes/quote_artwork/8
HOPE,/inspirational-quotes/3560-hope-is-a-state-of-mind-not-of-the-world,https://assets.passiton.com/quotes/quote_artwork/3560/med
HOPE,/inspirational-quotes/6827-just-as-one-cannot-live-without-dreams-one,https://assets.passiton.com/quotes/quote_artwork/6827/m
HOPE,/inspirational-quotes/8290-we-have-always-held-to-the-hope-the-belief,https://assets.passiton.com/quotes/quote_artwork/8290/m
HOPE,/inspirational-quotes/7457-hope-smiles-from-the-threshold-of-the-year-to,https://assets.passiton.com/quotes/quote_artwork/745

```

PROGRAM NO: 17**Date: 16/02/2022****AIM: Program for Natural Language Processing which performs n-grams.****Program Code:**

```
def generate_ngrams(text, WordToCombine):
    Words = text.split()
    output = []
    for i in range(len(Words) - WordToCombine + 1):
        output.append(Words[i:i + WordToCombine])
    return output
x=generate_ngrams(text='this is a very good book to study',WordToCombine=3)
print(x)
```

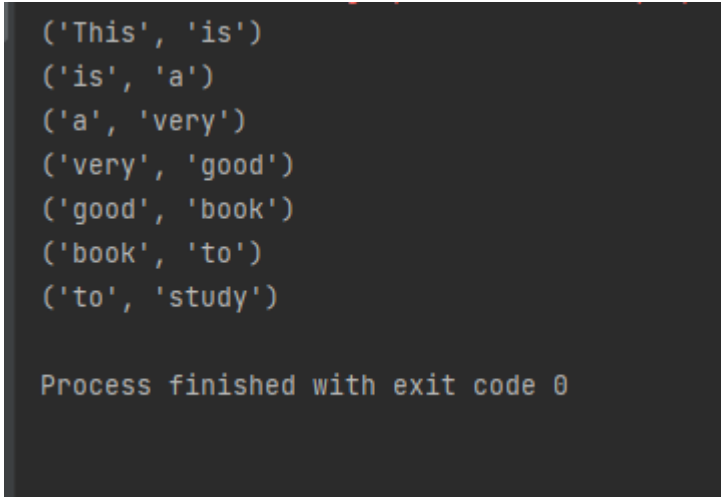
Output:

```
C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/Lab/aaa.py
[['this', 'is', 'a'], ['is', 'a', 'very'], ['a', 'very', 'good'], ['very', 'good', 'book'], ['good', 'book', 'to'], ['book', 'to', 'study']]

Process finished with exit code 0
```

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
from nltk.util import ngrams
nltk.download('punkt')
samplText = 'This is a very good book to study'
GRAMS = ngrams(sequence=nltk.word_tokenize(samplText), n=2)
for grams in GRAMS:
    print(grams)
```

Output:

```
('This', 'is')
('is', 'a')
('a', 'very')
('very', 'good')
('good', 'book')
('book', 'to')
('to', 'study')

Process finished with exit code 0
```

PROGRAM NO: 19**Date: 16/02/2022****AIM: Program for Natural Language Processing which performs speech tagging.****Program Code:**

```

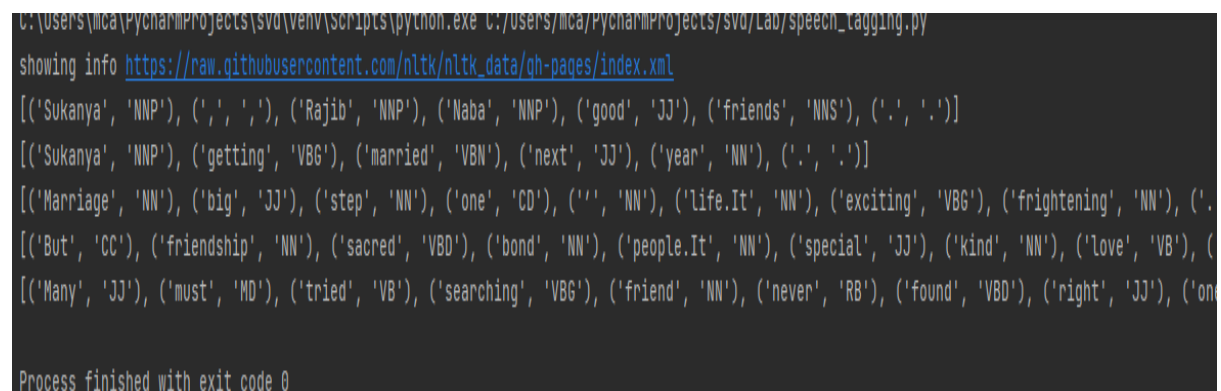
import nltk
nltk.download()
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

stop_words = set(stopwords.words('english'))

txt = "Sukanya, Rajib and Naba are my good friends. " \
      "Sukanya is getting married next year. " \
      "Marriage is a big step in one's life." \
      "It is both exciting and frightening. " \
      "But friendship is a sacred bond between people." \
      "It is a special kind of love between us. " \
      "Many of you must have tried searching for a friend " \
      "but never found the right one."
tokenized = sent_tokenize(txt)
for i in tokenized:
    wordsList = nltk.word_tokenize(i)

    wordsList = [w for w in wordsList if not w in stop_words]
    tagged = nltk.pos_tag(wordsList)
    print(tagged)

```

Output


```

C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/Lab/speech_tagging.py
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]
[('Sukanya', 'NNP'), ('getting', 'VBG'), ('married', 'VBN'), ('next', 'JJ'), ('year', 'NN'), ('.', '.')]
[('Marriage', 'NN'), ('big', 'JJ'), ('step', 'NN'), ('one', 'CD'), ('', 'NN'), ('life.It', 'NN'), ('exciting', 'VBG'), ('frightening', 'NN'), ('.')]
[('But', 'CC'), ('friendship', 'NN'), ('sacred', 'VBD'), ('bond', 'NN'), ('people.It', 'NN'), ('special', 'JJ'), ('kind', 'NN'), ('love', 'VB'), ('.')]
[('Many', 'JJ'), ('must', 'MD'), ('tried', 'VB'), ('searching', 'VBG'), ('friend', 'NN'), ('never', 'RB'), ('found', 'VBD'), ('right', 'JJ'), ('one', 'NN')]

Process finished with exit code 0

```

PROGRAM NO: 20**Date: 23/02/2022****AIM: Write python program for natural language processing which perform chunking.****Program Code:**

```

import nltk
new = "The big cat ate the little mouse who was after the fresh cheese"
new_tokens = nltk.word_tokenize(new)
print(new_tokens)

new_tag = nltk.pos_tag(new_tokens)
print(new_tag)

grammer = "NP: {<DT>?<JJ>*<NN>}"
chunkParser = nltk.RegexpParser(grammer)
chunked = chunkParser.parse(new_tag)
print(chunked)
chunked.draw()

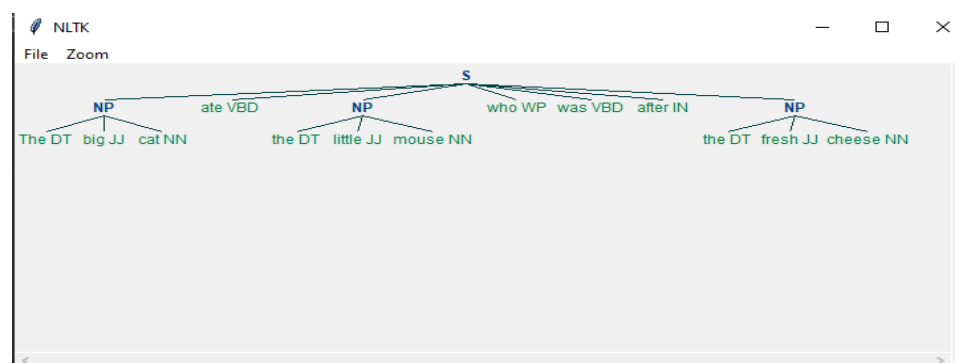
```

Output:

```

C:\Users\mca\PycharmProjects\svd\venv\Scripts\python.exe C:/Users/mca/PycharmProjects/svd/chunking.py
['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/JJ cat/NN)
  ate/VBD
  (NP the/DT little/JJ mouse/NN)
  who/WP
  was/VBD
  after/IN
  (NP the/DT fresh/JJ cheese/NN))

```



PROGRAM NO: 21**Date: 23/02/2022****AIM: Write python program for natural language processing which perform chunking.****Program Code:**

```
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.
"""

tokenized = nltk.sent_tokenize(sample_text)
for i in tokenized:
    words = nltk.word_tokenize(i)
    # print(words)
    tagged_words = nltk.pos_tag(words)
    # print(tagged_words)
    chunkGram = r"""VB: { }"""
    chunkParser = nltk.RegexpParser(chunkGram)
    chunked = chunkParser.parse(tagged_words)
    print(chunked)
    chunked.draw()
```

Output:

```

(S
  Rama/NNP
  killed/VBD
  Ravana/NNP
  to/TO
  save/VB
  Sita/NNP
  from/IN
  Lanka.The/NNP
  legend/NN
  of/IN
  the/DT
  Ramayan/NNP
  is/VBZ
  the/DT
  most/RBS
  popular/JJ
  Indian/JJ
  epic.A/NN
  lot/NN
  of/IN
  movies/NNS
  and/CC
  serials/NNS
  have/VBP
  already/RB
  been/VBN
  shot/VBN
  in/IN
  several/JJ
  languages/NNS
  here/RB
  in/IN
  India/NNP
  based/VBN
  on/IN
  the/DT
  Ramayana/NNP
  ./.)

```

Rama NNP killed VBD Ravana NNP to TO save VB Sita NNP from IN Lanka.The NNP legend NN of IN the DT Ramayan NNP is VBZ

