20MCA241 – Data Science Lab

Lab Report Submitted By

Anina Elizebeth

AJC22MCA-2017

In Partial Fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS
(MCA TWO YEAR)
[Accredited by NBA]

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. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2024

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 **ANINA ELIZEBETH (AJC22MCA-2017)** 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 **2023-24.**

Mr. Ajith G.S
Lab In-Charge

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

Internal Examiner

External Examiner

Course Code	Course Name	Syllabus Year	L-T-P-C
20MCA241	Data Science Lab	2020	0-1-3-2

VISION

To promote an academic and research environment conducive for innovation centric technical education.

MISSION

MS1 - Provide foundations and advanced technical education in both theoretical and applied Computer Applications in-line with Industry demands.
MS2 - Create highly skilled computer professionals capable of designing and innovating real life solutions.
MS3 - Sustain an academic environment conducive to research and teaching focused to generate up-skilled professionals with ethical values.
MS4 - Promote entrepreneurial initiatives and innovations capable of bridging and contributing with sustainable, socially relevant technology solutions.

COURSE OUTCOME

CO	Outcome	Target
CO1	Use different python packages to perform numerical calculations, statistical computations and data visualization.	60.2
CO2	Use different packages and frameworks to implement regression and classification algorithms.	60.2
CO3	Use different packages and frameworks to implement text classification using SVM and clustering using K-means.	60.2
CO4	Implement convolutional neural network algorithm using Keras framework.	60.2
CO5	Implement programs for web data mining and natural language processing using NLTK.	60.2

COURSE END SURVEY

CO	Survey Question	Answer Format
CO1	To what extend you are able to use different python packages to perform numerical calculations, statistical computations and data visualization?	Excellent/Very Good/Good/Satisfactory/Poor
CO2	To what extend you are able to use different packages and frameworks to implement regression and classification algorithms?	Excellent/Very Good/Good/Satisfactory/Poor
CO3	To what extend you are able to use different packages and frameworks to implement text classification using SVM and clustering using K-means?	Excellent/Very Good/Good/Satisfactory/Poor
CO4	To what extend you are able to implement convolutional neural network algorithm using Keras framework?	Excellent/Very Good/Good/Satisfactory/Poor
CO5	To what extend you are able to implement programs for web data mining and natural language processing using NLTK?	Excellent/Very Good/Good/Satisfactory/Poor

CONTENT

17	Program to implement a simple web crawler using requests library	06-12-23	CO5	24
18	Program to implement a simple web crawler and parse the content using BeautifulSoup.	06-12-23	CO5	25
19	Implement problems on natural language processing - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK	07-12-23	CO5	26

Experiment No.: 1

Aim

Program to perform matrix operations. Use numpy as the python library and perform the operation using built in functions.

CO1

Use different python packages to perform numerical calculations, statistical computations and data visualization

Procedure

```
import numpy as np
def input_matrix(matrix_name, rows, cols):
matrix = []
print(f"Enter the elements for {matrix_name}:")
for i in range(rows):
row = []
for j in range(cols):
element = int(input(f"Enter the element in row {i + 1}, column {j + 1}: "))
row.append(element)
matrix.append(row)
return np.array(matrix)

rows = int(input(f"Enter the number of rows: "))
cols = int(input(f"Enter the number of columns: "))
matrix1 = input_matrix("matrix1", rows, cols)
matrix2 = input_matrix("matrix2", rows, cols)

print(matrix1)
print(matrix2)

sum_result = np.add(matrix1, matrix2)
print("Sum of matrices:")
print(sum_result)

sub = np.subtract(matrix1, matrix2)
print("Subtract of matrices:")
print(sub)

mul = np.multiply(matrix1, matrix2)
print("multiply of matrices:")
```

```
print(mul)

div = np.divide(matrix1, matrix2)
print("divide of matrices:")
print(div)

trans = np.transpose(matrix1)
print("transpose of matrix 1:")
print(trans)

trans = np.transpose(matrix2)
print("transpose of matrix 2:")
print(trans)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\experiment1.py
Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements for matrix1:
Enter the element in row 1, column 1: 1
Enter the element in row 1, column 2: 4
Enter the element in row 2, column 1: 7
Enter the element in row 2, column 2: 5
Enter the elements for matrix2:
Enter the element in row 1, column 1: 3
Enter the element in row 1, column 2: 4
Enter the element in row 2, column 1: 7
Enter the element in row 2, column 2: 9
[[1 4]
 [7 5]]
[[3 4]
 [7 9]]
Sum of matrices:
[[ 4  8]
 [14 14]]
Subtract of matrices:
[[-2  0]
 [ 0 -4]]
multiply of matrices:
[[ 3 16]
```

Result

The program was executed successfully and the output was obtained. Thus, CO1 has been attained.

Experiment No.: 2

Aim

Program to perform single value decomposition using numpy.

CO1

Use different python packages to perform numerical calculations, statistical computations and data visualization.

Procedure

```
import numpy as np
matrix = np.array([[1,2,3],
[4,5,6],
[7,5,6],
])
U, S, VT = np.linalg.svd(matrix)
print("U matrix:")
print(U)
print("s value:")
print(np.diag(S))
print("VT matrix")
print(VT)
recon=np.dot(U,np.dot(np.diag(S),VT))
print(recon)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\experiment2.py
U matrix:
[[-0.25149997 -0.57729037 -0.77684206]
 [-0.62108755 -0.51931716  0.58699228]
 [-0.74229241  0.63011547 -0.22793961]]
s value:
[[14.01164552  0.          0.        ]
 [ 0.          2.14097673  0.        ]
 [ 0.          0.          0.30001393]]
VT matrix
[[-0.56609318 -0.52241542 -0.63766812]
 [ 0.82030284 -0.28052112 -0.49840862]
 [-0.08149697  0.80522669 -0.58733995]]
[[1. 2. 3.]
 [4. 5. 6.]
 [7. 5. 6.]]

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO1 has been attained.

Experiment No.: 3

Aim

Program to perform data visualization using python library matplotlib.

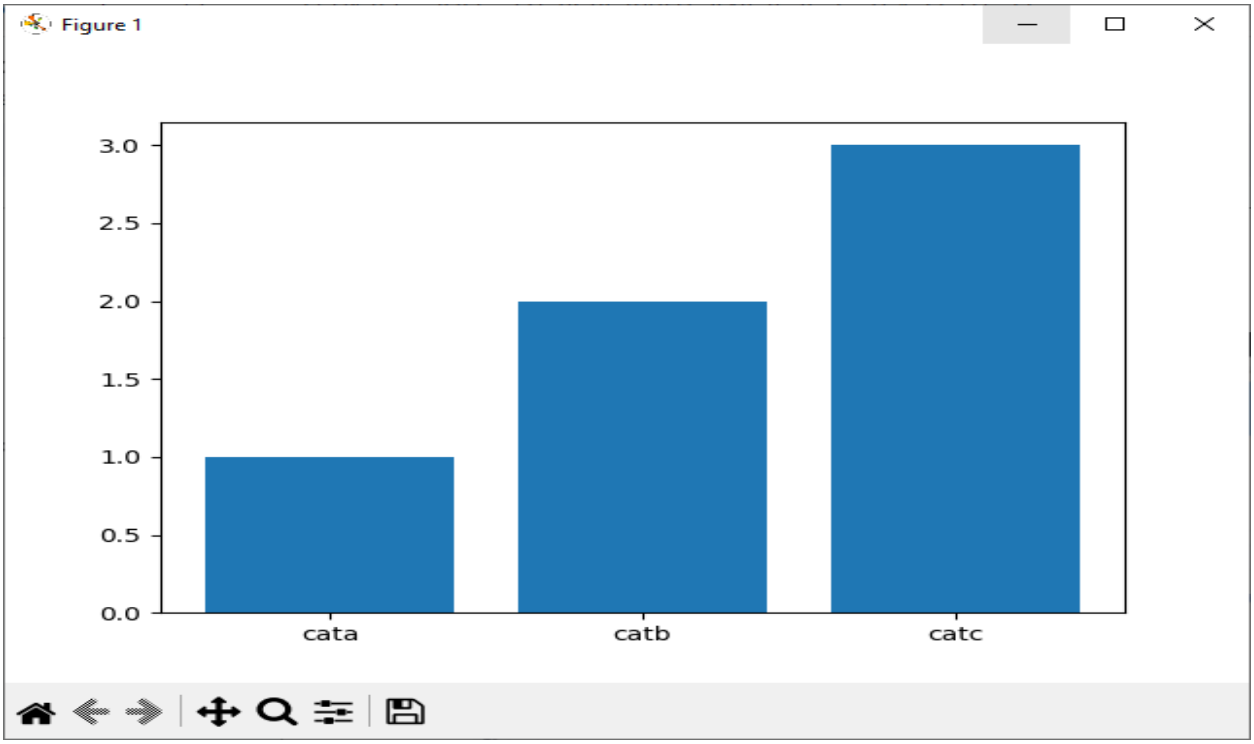
CO1

Use different python packages to perform numerical calculations, statistical computations and data visualization.

Procedure

```
import matplotlib.pyplot as plt

category= ['cata','catb','catc']

value=[1,2,3]

plt.bar(category,value)

plt.show()
```

Output



Result

The program was executed successfully and the output was obtained. Thus, CO1 has been attained.

Experiment No.: 4

Aim

Program to implement KNN classification using any standard dataset available in the public domain and find the accuracy of algorithm (Iris Dataset).

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
iris= load_iris()
x=iris.data #feature
y=iris.target #targetVariable
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)#trainingFeature
,traingTarget
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
print(knn.predict(x_test))
V=knn.predict(x_test)
result=accuracy_score(y_test, V)
print("accuracy:" ,result)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\venv\exp5.py
[1 0 2 1 1 0 1 2 2 1 2 0 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]
accuracy: 0.9666666666666667

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 5

Aim

Program to implement KNN classification using any standard dataset available in the public domain and find the accuracy of algorithm (Load Digits).

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_digits
from sklearn.metrics import accuracy_score
digit= load_digits()
x=digit.data #feature
y=digit.target #targetVariable
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=62)#trainingFeature
,traingTarget
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
print(knn.predict(x_test))
V=knn.predict(x_test)
result=accuracy_score(y_test, V)
print("accuracy:" ,result)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\exp6.py
[7 7 9 5 4 3 9 0 6 5 5 6 3 9 7 4 1 5 5 3 3 4 4 5 2 3 8 2 6 4 5 3 2 5 5 0 7
 2 9 2 2 4 0 2 8 3 2 8 9 2 7 0 6 0 8 6 8 3 9 9 3 5 2 3 4 6 0 6 1 0 4 7 7 4
 4 8 1 9 9 5 8 7 1 7 8 1 4 5 7 4 4 7 2 2 1 8 3 9 8 3 8 4 3 7 8 0 0 4 8 8 0
 9 5 4 2 8 5 5 6 2 9 7 9 3 5 2 1 5 8 9 9 9 9 9 5 8 0 2 3 1 8 1 0 4 3 4 9 6
 1 7 7 1 3 8 9 8 5 5 9 6 4 7 0 0 4 7 4 6 4 9 4 1 1 0 8 5 2 3 2 3 1 8 1 5 8
 8 3 6 5 0 5 7 3 6 1 1 4 4 0 2 6 3 7 7 7 6 6 1 6 3 8 8 8 6 5 1 3 8 4 0 8 1
 2 3 9 1 1 3 5 6 5 4 5 5 7 6 0 6 8 8 1 4 5 6 1 6 7 2 7 6 9 8 6 9 2 8 4 8 2
 8 7 6 4 6 4 6 1 2 0 9 3 5 0 0 6 6 4 8 4 5 0 7 4 9 9 0 7 6 6 3 7 2 3 9 9 8
 7 5 8 3 2 4 1 2 4 1 6 2 0 2 6 7 9 6 6 1 7 9 1 1 0 1 0 7 9 2 5 8 0 1 9 9 0
 9 7 7 3 2 1 0 7 3 6 2 4 8 5 3 8 3 9 0 2 1 7 2 6 4 4 9]
accuracy: 0.9805555555555555

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 6

Aim

Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of algorithm (Iris Dataset)

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
iris= load_iris()
x=iris.data #feature
y=iris.target #targetVariable
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)#trainingFeature,traingTarget
clf=GaussianNB()
clf.fit(x_train,y_train)
print(clf.predict(x_test))
V=clf.predict(x_test)
result=accuracy_score(y_test, V)
print("accuracy:" ,result)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\experi6.py
[1 0 2 1 1 0 1 2 1 1 2 0 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]
accuracy: 1.0

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 7

Aim

Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of algorithm (Breast Cancer Dataset)

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import accuracy_score,classification_report
data=load_breast_cancer()
x=data.data
y=data.target #targetVariable
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)#trainingFeature,traingTarget
clf=GaussianNB()
clf.fit(x_train,y_train)
print(clf.predict(x_test))
V=clf.predict(x_test)
result=accuracy_score(y_test, V)
print("accuracy:" ,result)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\eg.py
[1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0
 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0
 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0
 1 1 0]
accuracy: 0.9736842105263158
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 8

Aim

Give one dimensional dataset represented with numpy array. Write a program to calculate slope and intercept

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
import numpy as np
from sklearn.linear_model import LinearRegression

y= np.array([55, 60, 65, 70, 80]).reshape(-1, 1)
x = np.array([52, 54, 56, 58, 62])

model = LinearRegression()
model.fit(y,x)
slope = model.coef_[0]
intercept = model.intercept_

print(f"Slope: {slope}")
print(f"Intercept: {intercept}")
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\ajcemca\PycharmProjects\pythonProject\exp 14.py"
Slope: 0.3999999999999998
Intercept: 30.00000000000001

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 9

Aim

Program to implement simple linear regression using any standard dataset available in the public domain and find r2 score.

CO2

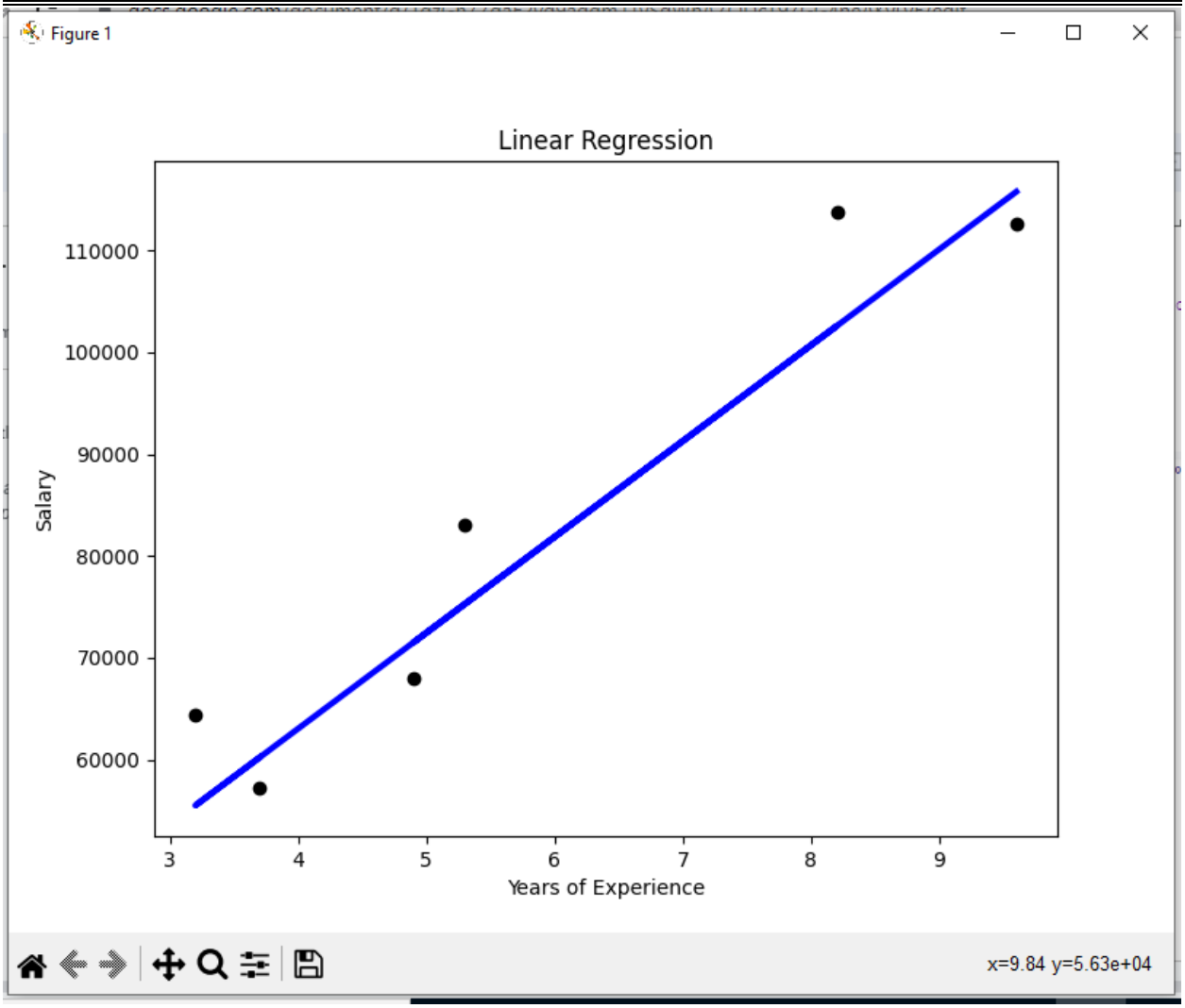
Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

data = pd.read_csv('Salary_Data.csv')
x = data['YearsExperience'].values.reshape(-1, 1)
y = data['Salary'].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

clf = LinearRegression()
clf.fit(x_train, y_train)
predictions = clf.predict(x_test)
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
print("R-squared:", r2)
plt.figure(figsize=(8, 6))
plt.scatter(x_test, y_test, color='black')
plt.plot(x_test, clf.predict(x_test), color='blue', linewidth=3)
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Linear Regression')
plt.show()
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\salary.py
R-squared: 0.9024461774180497
```



Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 10

Aim

Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance

CO2

Use different packages and frameworks to implement regression and classification algorithms.

Procedure

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.datasets import fetch_california_housing
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
data = fetch_california_housing()
x_train, x_test, y_train, y_test = train_test_split(data.data, data.target, test_size=0.2, random_state=42)
l = LinearRegression()
l.fit(x_train, y_train)
predictions = l.predict(x_test)
mse = mean_squared_error(y_test, predictions)
result_df = pd.DataFrame({'Actual': y_test, 'Predicted': predictions})
print(result_df)
print("\nMean Squared Error:", mse)
print('Coefficients:', l.coef_)
print('Intercept:', l.intercept_)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\exp15.py
       Actual  Predicted
0     0.47700   0.719123
1     0.45800   1.764017
2     5.00001   2.709659
3     2.18600   2.838926
4     2.78000   2.604657
...       ...        ...
4123  2.63300   1.991746
4124  2.66800   2.249839
4125  5.00001   4.468770
4126  0.72300   1.187511
4127  1.51500   2.009403

[4128 rows x 2 columns]

Mean Squared Error: 0.5558915986952425
Coefficients: [ 4.48674910e-01  9.72425752e-03 -1.23323343e-01  7.83144907e-01
 -2.02962058e-06 -3.52631849e-03 -4.19792487e-01 -4.33708065e-01]
Intercept: -37.02327770606397

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO2 has been attained.

Experiment No.: 11

Aim

Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm (Iris Dataset)

CO3

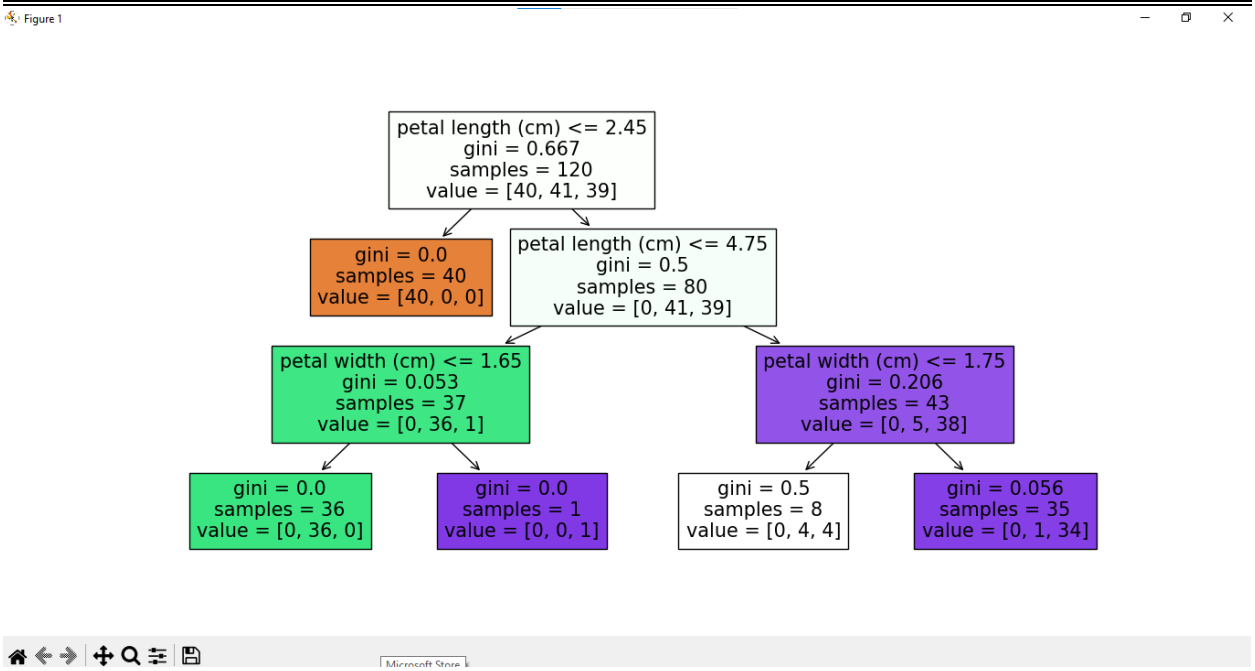
Use different packages and frameworks to implement text classification using SVM and clustering using k-means

Procedure

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
data = load_iris()
x = data.data
y = data.target # Target Variable
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
clf = DecisionTreeClassifier(max_depth=3)
clf.fit(x_train, y_train)
plt.figure(figsize=(15, 10))
plot_tree(clf, filled=True, feature_names=data.feature_names)
plt.show()
print(clf.predict(x_test))
V = clf.predict(x_test)
result = accuracy_score(y_test, V)
print("Accuracy:", result)
report = classification_report(y_test, V, target_names=data.target_names)
print("\nClassification Report:\n", report)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\expr9.py
[1 0 2 1 1 0 1 2 1 1 2 0 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]
accuracy: 1.0

 classification              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30


Process finished with exit code 0
```



Result

The program was executed successfully and the output was obtained. Thus, CO3 has been attained.

Experiment No.: 12

Aim

Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm (Breast Cancer Dataset)

CO3

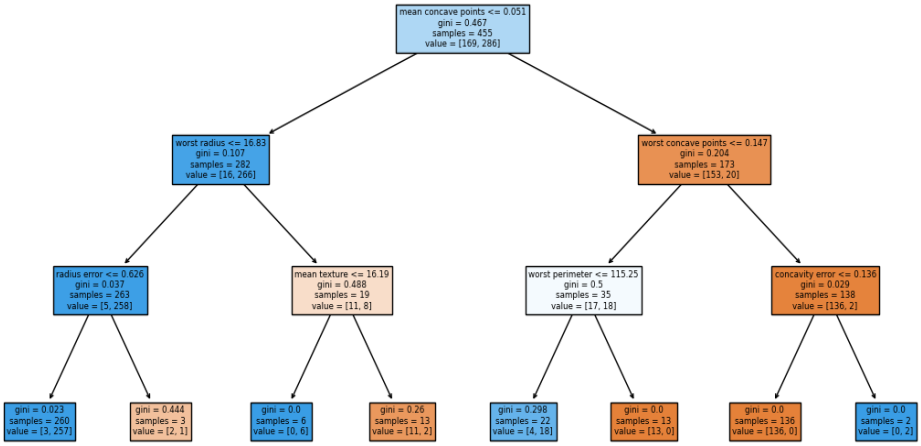
Use different packages and frameworks to implement text classification using SVM and clustering using k-means

Procedure

```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import accuracy_score, classification_report
data = load_breast_cancer()
x = data.data
y = data.target # Target Variable
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
clf = DecisionTreeClassifier(max_depth=3)
clf.fit(x_train, y_train)
plt.figure(figsize=(15, 10))
plot_tree(clf, filled=True, feature_names=data.feature_names)
plt.show()
print(clf.predict(x_test))
V = clf.predict(x_test)
result = accuracy_score(y_test, V)
print("Accuracy:", result)
report = classification_report(y_test, V, target_names=data.target_names)
print("\nClassification Report:\n", report)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\exp9eg.py
[1 0 0 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0
 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 1 0
 1 1 1 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0
 1 1 0]
accuracy: 0.9385964912280702

 classification              precision    recall  f1-score   support

   malignant       0.93      0.91      0.92        43
      benign       0.94      0.96      0.95        71

    accuracy                           0.94       114
   macro avg       0.94      0.93      0.93       114
weighted avg       0.94      0.94      0.94       114
```



Result

The program was executed successfully and the output was obtained. Thus, CO3 has been attained.

Experiment No.: 13

Aim

Program to implement k-means clustering technique using any standard dataset available in the public domain (Iris Dataset).

CO3

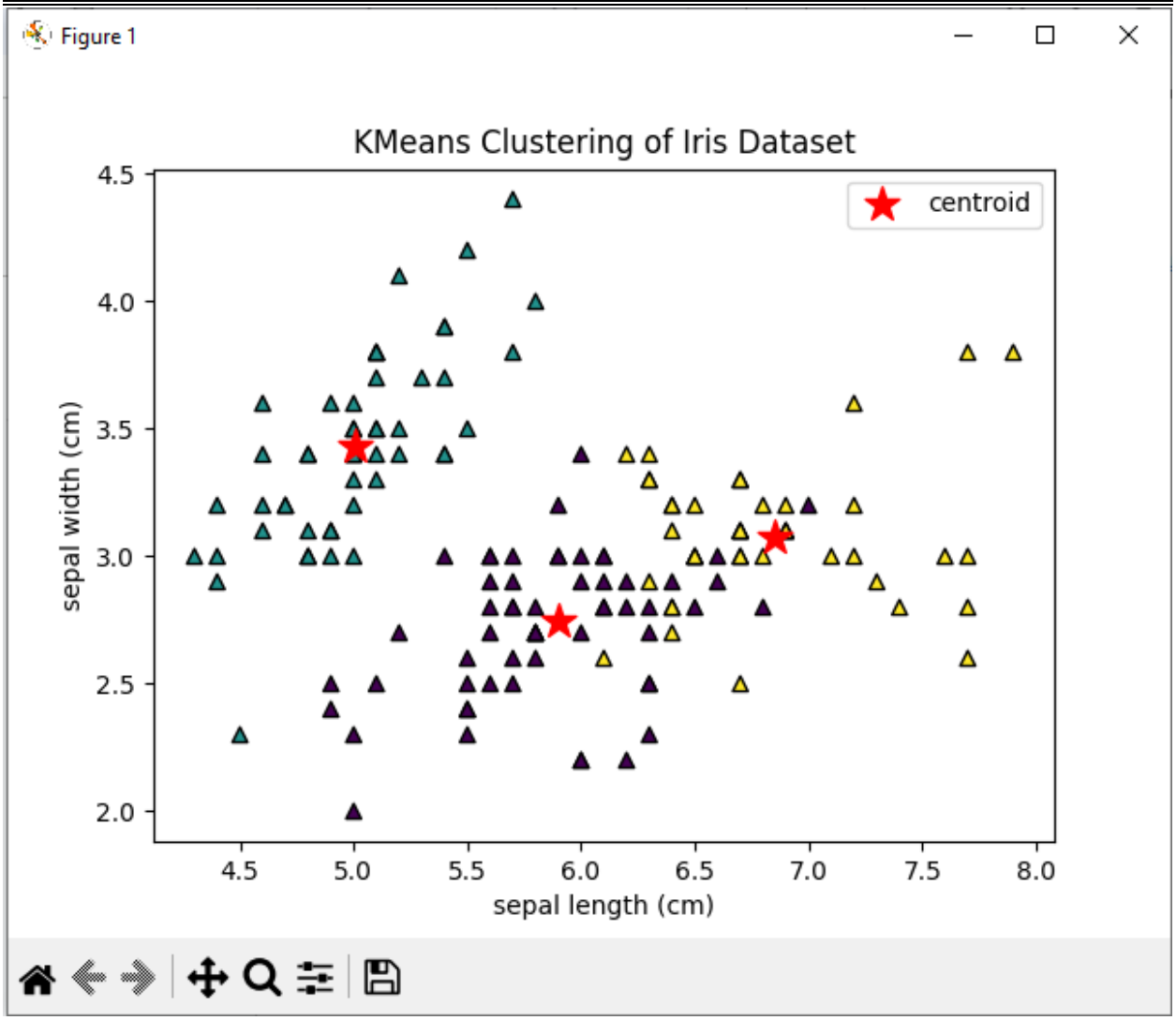
Use different packages and frameworks to implement text classification using SVM and clustering using k-means.

Procedure

```
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
import matplotlib.pyplot as py

iris=load_iris()
x=iris.data
y=iris.target

km=KMeans(n_clusters=3,random_state=42)
km.fit(x)

cluster=km.labels_
print(cluster)
centroid=km.cluster_centers_
print(centroid)

py.scatter(x[:,0],x[:,1],c=cluster,cmap='viridis',marker='^',edgecolors='black')
py.scatter(centroid[:,0],centroid[:,1],marker="*",s=200,c='red',label='centroid')
py.xlabel(iris.feature_names[0])
py.ylabel(iris.feature_names[1])
py.title('KMeans Clustering of Iris Dataset')
py.legend()
py.show()
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The d
  super()._check_params_vs_input(X, default_n_init=10)
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
[[5.9016129  2.7483871  4.39354839 1.43387097]
 [5.006      3.428      1.462      0.246     ]
 [6.85       3.07368421 5.74210526 2.07105263]]
```



Result

The program was executed successfully and the output was obtained. Thus, CO3 has been attained.

Experiment No.: 14

Aim

Program to implement k-means clustering technique using any standard dataset available in the public domain (Breast Cancer Dataset)

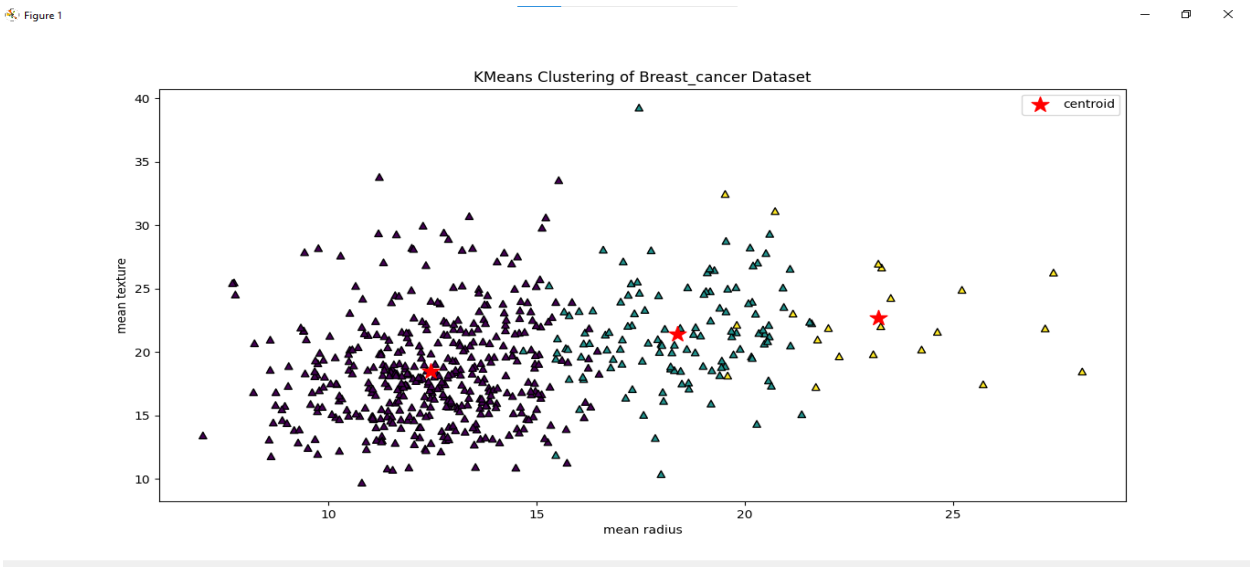
CO3

Use different packages and frameworks to implement text classification using SVM and clustering using k-means.

Procedure

```
from sklearn.datasets import load_breast_cancer
from sklearn.cluster import KMeans
import matplotlib.pyplot as py

iris=load_breast_cancer()
x=iris.data
y=iris.target
km=KMeans(n_clusters=3,random_state=42)
km.fit(x)
cluster=km.labels_
print(cluster)
centroid=km.cluster_centers_
print(centroid)

py.scatter(x[:,0],x[:,1],c=cluster,cmap='viridis',marker='^',edgecolors='black')
py.scatter(centroid[:,0],centroid[:,1],marker="*",s=200,c='red',label='centroid')
py.xlabel(iris.feature_names[0])
py.ylabel(iris.feature_names[1])
py.title('KMeans Clustering of Breast_cancer Dataset')
py.legend()
py.show()
```

Output



```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\lib\site-packages\sklearn\cluster\_kmeans.py
  super()._check_params_vs_input(X, default_n_init=10)
[1 1 1 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1 2 0 0 0 0 2 1 1 0 1 1 1 1 0 1 1 1 1 0
 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
 0 1 0 1 1 0 0 0 2 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0
 0 0 0 0 0 0 0 1 1 0 1 2 0 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 0 0 0 1 2 0 2 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 2 1 1 0 0
 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 2 0 0 0 0 1 0 0 1 0 2 1 0 0 0 0 1 2 0 0
 0 1 0 0 0 0 0 0 1 0 0 1 0 0 2 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0 1 0 1
 0 1 1 1 0 1 2 0 0 0 0 0 0 2 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0
 0 0 1 0 1 0 2 0 0 0 1 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 2 2
 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0
 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 1 1
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 0]
[[1.24468918e+01 1.85046588e+01 8.03803294e+01 4.86458118e+02
  9.48157176e-02 9.04881882e-02 6.08800016e-02 3.25494682e-02
  1.77951765e-01 6.35771765e-02 3.00681647e-01 1.21837294e+00
  2.12940400e+00 2.32080188e+01 7.17541647e-03 2.33490235e-02
  2.84143873e-02 1.05452329e-02 2.07137600e-02 3.75171835e-03
  1.38918094e+01 2.45948235e+01 9.09125412e+01 6.04658353e+02
  1.29845529e-01 2.21074000e-01 2.14822228e-01 8.97035082e-02
  2.82468471e-01 8.32831059e-02]
 [1.83820325e+01 2.14148780e+01 1.21238537e+02 1.05796098e+03
  1.00221870e-01 1.40414797e-01 1.58604959e-01 9.06387805e-02
  1.91033333e-01 6.06883740e-02 6.40926016e-01 1.20443577e+00
  4.50100813e+00 7.53708943e+01 6.57197561e-03 3.09567967e-02
  4.08994309e-02 1.53647805e-02 2.00486992e-02 3.93508943e-03
  2.22162602e+01 2.86411382e+01 1.47833333e+02 1.52278862e+03
  1.39408780e-01 3.45692358e-01 4.26761789e-01 1.81023984e-01
  3.15549593e-01 8.64585366e-02]
 [2.32147619e+01 2.27285714e+01 1.55066667e+02 1.70276190e+03
  1.05001429e-01 1.73405714e-01 2.44971429e-01 1.35852381e-01
  1.88309524e-01 5.93747619e-02 1.13901429e+00 1.25883333e+00
  8.19842857e+00 1.81798571e+02 7.06723810e-03 3.64780952e-02
  4.95609524e-02 1.62100000e-02 1.99633333e-02 3.84780952e-03
  2.95500000e+01 3.02228571e+01 2.00490476e+02 2.70328571e+03
  1.42195238e-01 3.90485714e-01 5.27814286e-01 2.29571429e-01
  2.94823810e-01 8.26404762e-02]]
```

Result

The program was executed successfully and the output was obtained. Thus, CO3 has been attained.

Experiment No.: 15

Aim

Program to implement text classification using support vector machine.

CO3

Use different packages and frameworks to implement text classification using SVM and clustering using k-means

Procedure

```
from sklearn.datasets import fetch_20newsgroups
from sklearn.metrics import classification_report, accuracy_score
from sklearn.svm import SVC
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split

cat = ["alt.atheism", "soc.religion.christian", "comp.graphics", "sci.med"]
twenty_train = fetch_20newsgroups(subset="train", categories=cat, shuffle=True,
random_state=42)

vector = TfidfVectorizer()
X = vector.fit_transform(twenty_train.data)
y = twenty_train.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train, y_train)
predictions = svm_classifier.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
class_report = classification_report(y_test, predictions,
target_names=twenty_train.target_names)

print("Accuracy:", accuracy)
print("Classification Report:")
print(class_report)

new_data = [
  "I have a question about computer graphics",
  "This is a medical-related report",
]
```

```
X_new_tfidf = vector.transform(new_data)

new_predictions = svm_classifier.predict(X_new_tfidf)

for i, text in enumerate(new_data):
  print(f"Text: {text}")
  print(f"Predicted category: {twenty_train.target_names[new_predictions[i]]}")
  print("\n++++++++++++++++++\n")
```

Output

```
Classification Report:
                        precision    recall  f1-score   support

           alt.atheism       0.96      0.95      0.96        86
         comp.graphics       0.91      1.00      0.96       107
               sci.med       0.98      0.95      0.97       132
soc.religion.christian       0.98      0.94      0.96       127

              accuracy                           0.96       452
             macro avg       0.96      0.96      0.96       452
          weighted avg       0.96      0.96      0.96       452

Text: I have a question about computer graphics
Predicted category: comp.graphics

++++++++++++++++++

Text: This is a medical-related report
Predicted category: sci.med

++++++++++++++++++


Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO3 has been attained.

Experiment No.: 16

Aim

Program on artificial neural network to classify images from any standard dataset in the public domain using Keras framework.

CO4

Implement convolutional neural network algorithm using Keras framework.

Procedure

```
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.utils import to_categorical
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train / 255.0
x_test = x_test / 255.0
x_train = x_train.reshape(-1, 28 * 28)
x_test = x_test.reshape(-1, 28 * 28)
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
model = Sequential([
  Dense(128, activation='relu', input_shape=(28 * 28,)),
  Dense(64, activation='relu'),
  Dense(10, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5, batch_size=32, validation_data=(x_test, y_test))
loss, accuracy = model.evaluate(x_test, y_test)
print(f'Test Accuracy: {accuracy}')
```

Output

```
Epoch 1/5
WARNING:tensorflow:From C:\Users\ajcemca\PycharmProjects\pythonProject\venv\lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.Ra

WARNING:tensorflow:From C:\Users\ajcemca\PycharmProjects\pythonProject\venv\lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.

1875/1875 [==============================] - 5s 2ms/step - loss: 0.2376 - accuracy: 0.9310 - val_loss: 0.1260 - val_accuracy: 0.9606
Epoch 2/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0999 - accuracy: 0.9690 - val_loss: 0.1004 - val_accuracy: 0.9692
Epoch 3/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0718 - accuracy: 0.9785 - val_loss: 0.0901 - val_accuracy: 0.9731
Epoch 4/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0546 - accuracy: 0.9828 - val_loss: 0.0803 - val_accuracy: 0.9759
Epoch 5/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0436 - accuracy: 0.9859 - val_loss: 0.0842 - val_accuracy: 0.9746
313/313 [==============================] - 0s 952us/step - loss: 0.0842 - accuracy: 0.9746
Test Accuracy: 0.9746000170707703

Process finished with exit code 0
```

Result

The program was executed successfully and the output was obtained. Thus, CO4 has been attained.

Experiment No.: 17

Aim

Program to implement a simple web crawler using requests library.

CO4

Implement convolutional neural network algorithm using Keras framework.

Procedure

```
import requests

def simple_scraper(url):
    response = requests.get(url)
    if response.status_code == 200:
        print("Content:")
        print(response.text)
    else:
        print("Failed to fetch the page. Status code:", response.status_code)

url_to_scrape = "https://ajce.in"
simple_scraper(url_to_scrape)
```

Output

```
C:\Users\91989\PycharmProjects\sem3\venv\Scripts\python.exe C:\Users\91989\PycharmProjects\sem3\EXP19.py
Content:
<!DOCTYPE html>
<html lang="en">


<head><meta charset="windows-1252">

<title>Amal Jyothi College of Engineering (Autonomous)</title>
<meta name="viewport" content="width=device-width, initial-scale=1" />
                <script type="text/javascript">
                        <!--
                        if (screen.width <= 699) {
                        document.location = "/m/index.html";
                        }

                        </script>
        <!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->
        <link rel="stylesheet" href="assets/css/main.css" />

    <!--Bootstrap Stylesheet [ REQUIRED ]-->
    <link href="css/bootstrap.css" rel="stylesheet">
```

Result

The program was executed successfully and the output was obtained. Thus, CO4 has been attained.

Experiment No.: 18

Aim

Program to implement a simple web crawler and parse the content using BeautifulSoup.

CO5

Implement programs for web data mining and natural language processing using NLTK.

Procedure

```
import requests
from bs4 import BeautifulSoup
def simple_scraper(url):
  response = requests.get(url)
  if response.status_code == 200:
     soup=BeautifulSoup(response.content,'html.parser')
     print("Title:",soup.title.string)
     print("Content:")
     print(soup.get_text())
  else:
     print("Failed to fetch the page. Status code:", response.status_code)
url_to_scrape = "https://ajce.in"
simple_scraper(url_to_scrape)
```

Output

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\ajcemca\PycharmProjects\pythonProject\we.py
Title: Amal Jyothi College of Engineering (Autonomous)
Content:


Amal Jyothi College of Engineering (Autonomous)

  Amal Jyothi College of Engineering



  KERALA'S LARGEST INFRASTRUCTURE FOR ENGINEERING EDUCATION WITH 7 NBA ACCREDITED PROGRAMS


  HOME
  B TECH
  M TECH
  M C A
  IQAC

  VIDEO

  360°
  FACULTY
  HOSTELS
```

Result

The program was executed successfully and the output was obtained. Thus, CO5 has been attained.

Experiment No.: 19

Aim

Implement problems on natural language processing - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK

CO5

Implement programs for web data mining and natural language processing using NLTK

Procedure

```
import nltk
nltk.download('brown')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
from nltk.tokenize import word_tokenize
from nltk.util import ngrams
from nltk.corpus import brown
from nltk.chunk import RegexpParser
sentence = "The quick brown fox jumps over the lazy dog"
tokens = word_tokenize(sentence)
print(tokens)
pos_tags = nltk.pos_tag(tokens)
print("Part-of-Speech Tagging: ")
print(pos_tags)
text = brown.words(categories='news')[:1000]
bigrams = list(ngrams(text, 2))
freq_dist = nltk.FreqDist(bigrams)
print("\n N-gram Analysis (Bigrams with Smoothing): ")
for bigram in bigrams:
    print(f"{bigram}: {freq_dist[bigram]}")
tagged_sentence = nltk.pos_tag(word_tokenize("The quick brown fox jumps over the lazy dog"))
grammar = r"NP: {<DT>?<JJ>*<NN>}"
cp = RegexpParser(grammar)
result = cp.parse(tagged_sentence)
print("\n Chunking with Regular Expressions and POS tags: ")
print(result)
```

Output

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
Part-of-Speech Tagging:
[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'), ('lazy', 'JJ'), ('dog', 'NN')]

 N-gram Analysis (Bigrams with Smoothing):
('The', 'Fulton'): 1
('Fulton', 'County'): 6
('County', 'Grand'): 1
('Grand', 'Jury'): 1
('Jury', 'said'): 1
('said', 'Friday'): 1
('Friday', 'an'): 1
('an', 'investigation'): 1
('investigation', 'of'): 1
('of', "Atlanta's"): 1
("Atlanta's", 'recent'): 1
('recent', 'primary'): 1
('primary', 'election'): 1
('election', 'produced'): 1
('produced', '``'): 1
('``', 'no'): 1
('no', 'evidence'): 1
('evidence', "''"): 1
("''", 'that'): 1
```

Result

The program was executed successfully and the output was obtained. Thus, CO5 has been attained.