20MCA241– DATA SCIENCE LAB

*Lab Report Submitted By*

**SREELAKSHMI MADHUSOODHANAN**

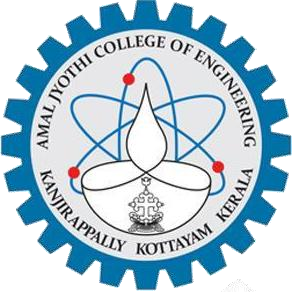
**Reg. No.: AJC21MCA-2099**

*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**

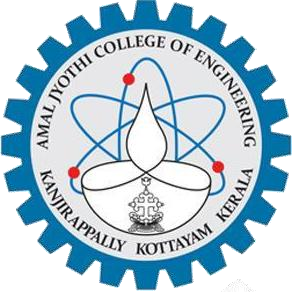
## [Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2022-2023**

**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 **SREELAKSHMI MADHUSOODHANAN (AJC21MCA-2099)** 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 2022-23.

**Ms.Sruthimol Kurian Rev.Fr.Dr.Rubin Thottupurathu Jose**

**Lab In-Charge Head of the Department**

**Internal Examiner External Exam**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
| CO2 | Use different packages and frameworks to implement regression and classification algorithms. | 60 |
| CO3 | Use different packages and frameworks to implement text classification using SVM and clustering using k-means | 60 |
| CO4 | Implement convolutional neural network algorithm using Keras framework. | 60 |
| CO5 | Implement programs for web data mining and natural language processing using NLTK | 60 |

**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/Needs improvement |
| CO2 | To what extend you are able to use different packages and frameworks to implement regression and classification algorithms? | Excellent/Very Good/Good Satisfactory/Needs improvement |
| 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/Needs improvement |
| CO4 | To what extend you are able to implement convolutional neural network algorithm using Keras framework? | Excellent/Very Good/Good Satisfactory/Needs improvement |
| 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/Needs improvement |

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No** | **Content** | **CO** | **Date** | **Page No.** |
| **1** | Data Handling with Pandas | **CO1** | **01-09-2022**  **12-09-2022** | 6 |
| **2** | Basic operation with Numpy | **CO1** | **22-08-2022** | 7 |
| **3** | Data visualization | **CO1** | **25-08-2022** | 8 |
| **4** | Matrix operation using Numpy | **CO1** | **11-08-2022** | 10 |
| **5** | Program to perform SVD | **CO1** | **11-08-2022** | 11 |
| **6** | Implementation of KNN- Classification | **CO2** | **19-09-2022**  **22-09-2022** | 12 |
| **7** | Implementation of Naive-Bayes Classification | **CO2** | **26-09-2022** | 13 |
| **8** | Program to handle Multiple Linear Regression | **CO2** | **10-10-2022** | 14 |
| **9** | Implementation of Decision Tree Classification | **CO3** | **17-10-2022** | 15 |
| **10** | Implementation of k- means clustering | **CO3** | **20-10-2022** | 16 |
| **11** | Implementation of CNN using Keras Network | **CO4** | **27-10-2022** | 17,18,19 |
| **12** | Scraping of any website | **CO5** | **31-10-202** | 20,21 |
| **13** | Performs n- grams using NLP | **CO5** | **03-11-2022** | 22 |
| **14** | Perform parts of speech tagging using NLP | **CO5** | **07-11-2022** | 23 |
| **15** | Data pre-processing using NLTK | **CO5** | **14-11-2022** | 24 |

**Experiment No. : 1**

**Aim**

Create a student table with columns Roll.no, Name, age, marks using pandas and do the following

a. select the top 2 rows

b. filter data based on some condition with mark&gt;80

c. filter in names first name start with &#39;N&#39; then remaining.

**CO1**

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

data visualization

**Program and Output**

import pandas as pd

s1 = pd.DataFrame({ 'RollNo': ['S1', 'S2', 'S3', 'S4', 'S5'],

         'name': ['Nirmal Fenton', 'Ryder Storey', 'Bryce Jensen', 'Nil Bernal', 'Kwame Morin'],

 'age':[23,56,12,13,14], 'marks': [20, 210, 190, 222, 30]})

print(s1.head(2))

**Output**

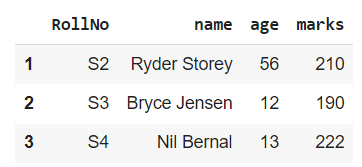
RollNo name age marks

0 S1 Nirmal Fenton 23 20

1 S2 Ryder Storey 56 210

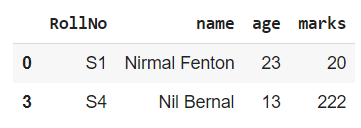
s1[s1['marks']>80]

**Output**



s1[s1['name'].str.startswith('N')]

**Output**



**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No. : 2**

**Aim**

Numpy array creation and basic operations, Initialization, array indexing.

**CO1**

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

**Program and Output**

import pandas as pd

import numpy as np

print(pd.Series(np.array([1,2,3,4,5,6,7]), index=['a','b','c','d','e','f','g']))

**Output**

a 1

b 2

c 3

d 4

e 5

f 6

g 7

dtype: int64

print(pd.Series(np.array([1,2,3,4,5,6,7]), index=['a','b','c','d','e','f','g'])\*2)

**Output**

a 2

b 4

c 6

d 8

e 10

f 12

g 14

dtype: int64

print(pd.Series(np.array([1,2,3,4,5,6,7]), index=['a','b','c','d','e','f','g'])\*\*2)

**Output**

a 1

b 4

c 9

d 16

e 25

f 36

g 49

dtype: int64

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No. : 3**

**Aim**

Plot a graph by matplotlib library

**CO1**

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

**Program and Output**

import matplotlib.pyplot as plt

# initializing the data

x = [10, 20, 30, 40]

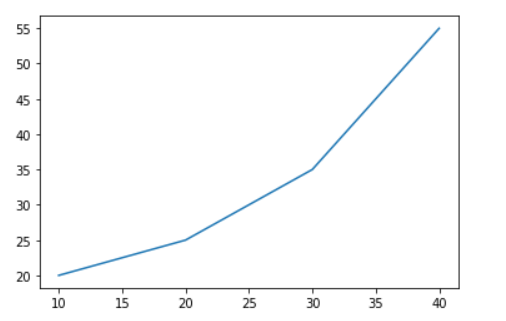
y = [20, 25, 35, 55]

# plotting the data

plt.plot(x, y)

plt.show()

**Output**



import matplotlib.pyplot as plt

x1 = [1, 3, 4, 5, 6, 7, 9]

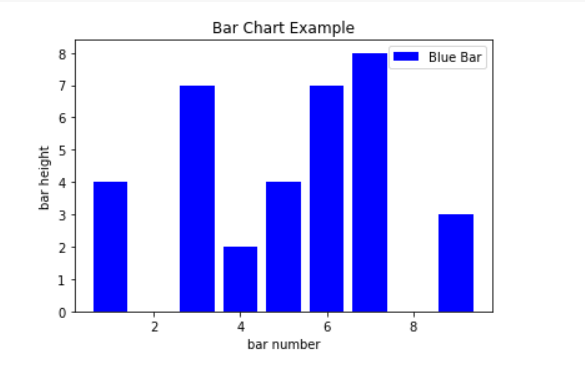
y1 = [4, 7, 2, 4, 7, 8, 3]

plt.bar(x1, y1, label="Blue Bar", color='b')

plt.plot()

plt.show()

**Output**



import matplotlib.pyplot as plt

x2 = [1, 2, 3, 4, 5]

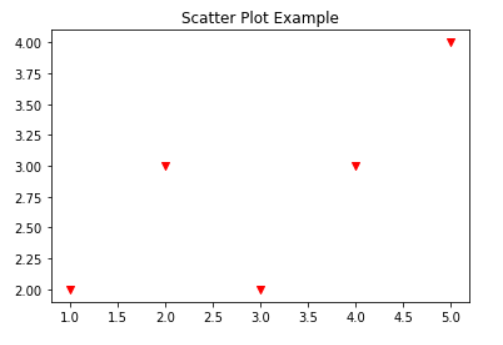
y2 = [2, 3, 2, 3, 4]

plt.scatter(x2, y2, marker='v', color='r')

plt.title('Scatter Plot Example')

plt.show()

**Output**



**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No. : 4**

**Aim**

Perform all matrix operation using python (using numpy)

**CO1**

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

**Program and Output**

import numpy as np

a = np.array([1, 2, 3])   # Create a rank 1 array

print("type: " ,type(a))            # Prints "<class 'numpy.ndarray'>"

print("shape: " ,a.shape)            # Prints "(3,)"

print(a[0], a[1], a[2])   # Prints "1 2 3"

a[0] = 5                  # Change an element of the array

print(a)                  # Prints "[5, 2, 3]"

b = np.array([[1,2,3],[4,5,6]])    # Create a rank 2 array

print("\n shape of b:",b.shape)                     # Prints "(2, 3)"

print(b[0, 0], b[0, 1], b[1, 0])   # Prints "1 2 4"

a = np.zeros((3,3))   # Create an array of all zeros

print("All zeros matrix:\n  " ,a)              # Prints "[[ 0.  0.]

b = np.ones((1,2))    # Create an array of all ones

print("\nAll ones matrix:\n  " ,b)              # Prints "[[ 1.  1.]]"

d = np.eye(2)         # Create a 2x2 identity matrix

print("\n identity matrix: \n",d)              # Prints "[[ 1.  0.]

e = np.random.random((2,2))  # Create an array filled with random values

print("\n random matrix: \n",e)

**Output**

type: <class 'numpy.ndarray'>

1 2 3

[5 2 3]

shape of b: (2, 3)

1 2 4

All zeros matrix:

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

All ones matrix:

[[1. 1.]]

identity matrix:

[[1. 0.]

[0. 1.]]

random matrix:

[[0.50738093 0.49587583]

[0.85821263 0.69582347]]

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No. : 5**

**Aim**

Program to Perform SVD (Singular Value Decomposition) in Python

**CO1**

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

**Program and Output**

from numpy import array

from scipy.linalg import svd

# define a matrix

A = array([[1, 2], [3, 4], [5, 6]])

print("A: \n", A)

# SVD

U, s, VT = svd(A)

print("\nU: \n", U)

print("\ns: \n", s)

print("\nV^T: \n", VT)

**Output**

A:

[[1 2]

[3 4]

[5 6]]

U:

[[-0.2298477 0.88346102 0.40824829]

[-0.52474482 0.24078249 -0.81649658]

[-0.81964194 -0.40189603 0.40824829]]

s:

[9.52551809 0.51430058]

V^T:

[[-0.61962948 -0.78489445]

[-0.78489445 0.61962948]]

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No. : 6**

**Aim**

Program to implement k-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm.

**CO2**

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

**Program and Output**

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import pandas as pd

from sklearn.datasets import load\_iris

data = load\_iris()

df = pd.DataFrame(data.data, columns=data.feature\_names)

df['target'] = data.target

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(df[data.feature\_names], df['target'], random\_state=42,test\_size=0.1)

clf = KNeighborsClassifier(n\_neighbors = 5)

clf.fit(X\_train, Y\_train)

y\_pred=clf.predict(X\_test)

# comparing actual response values (y\_test) with predicted response values (y\_pred)

from sklearn import metrics

print("KNN model accuracy(in %):", metrics.accuracy\_score(Y\_test, y\_pred)\*100)

**Output**

KNN model accuracy(in %): 100.0

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No. : 7**

**Aim**

Program to implement Naive Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm

**CO2**

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

**Program and Output**

from sklearn.datasets import load\_iris

iris = load\_iris()

X = iris.data

y = iris.target

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.naive\_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(X\_train, y\_train)

y\_pred = gnb.predict(X\_test)

from sklearn import metrics

print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy\_score(y\_test, y\_pred)\*100)

**Output**

Gaussian Naive Bayes model accuracy(in %): 100.0

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No. : 8**

**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.

**Program and Output**

import numpy as np

from sklearn.linear\_model import LinearRegression

x = [[0, 1], [5, 1], [15, 2], [25, 5], [35, 11], [45, 15], [55, 34], [60, 35]]

y = [4, 5, 20, 14, 32, 22, 38, 43]

x, y = np.array(x), np.array(y)

model = LinearRegression().fit(x, y)

r\_sq = model.score(x, y)

print(f"coefficient of determination: {r\_sq}")

print(f"intercept: {model.intercept\_}")

print(f"coefficients: {model.coef\_}")

y\_pred = model.predict(x)

print(f"predicted response:\n{y\_pred}")

**Output**

coefficient of determination: 0.8615939258756775

intercept: 5.52257927519819

coefficients: [0.44706965 0.25502548]

predicted response:

[ 5.77760476 8.012953 12.73867497 17.9744479 23.97529728 29.4660957

38.78227633 41.27265006]

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No. : 9**

**Aim**

Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.

**CO3**

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

**Program and Output**

import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import pandas as pd

from sklearn.datasets import load\_iris

data = load\_iris()

df = pd.DataFrame(data.data, columns=data.feature\_names)

df['target'] = data.target

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(df[data.feature\_names], df['target'], random\_state=42,test\_size=0.1)

clf = DecisionTreeClassifier()

clf.fit(X\_train, Y\_train)

y\_pred=clf.predict(X\_test)

from sklearn import metrics

print("Decision tree model accuracy(in %):", metrics.accuracy\_score(Y\_test, y\_pred)\*100)

**Output**

Decision tree model accuracy(in %): 100.0

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No. : 10**

**Aim**

Program to implement k- means clustering technique using any standard dataset available in the public domain

**CO3**

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

**Program and Output**

from sklearn.cluster import KMeans

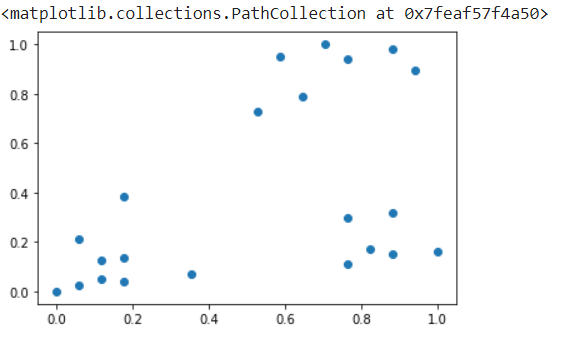
import pandas as pd

from matplotlib import pyplot as plt

df = pd.read\_csv("income.csv")

plt.scatter(df.Age,df['Income($)'])

**Output**



km = KMeans(n\_clusters=3)

y\_predicted = km.fit\_predict(df[['Age','Income($)']]) df['cluster']=y\_predicted

df1 = df[df.cluster==0] df2 = df[df.cluster==1] df3 = df[df.cluster==2]

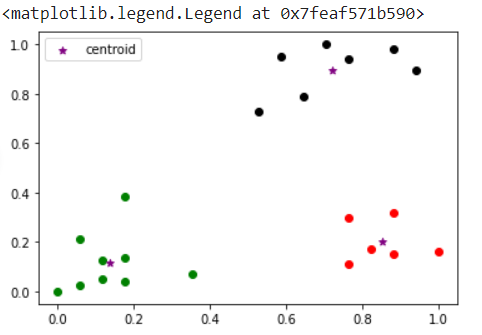
plt.scatter(df1.Age,df1['Income($)'],color='green') plt.scatter(df2.Age,df2['Income($)'],color='red')

plt.scatter(df3.Age,df3['Income($)'],color='black')

plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],color='purple',marker='\*',label='centroid')

plt.legend()

**Output**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No. : 11**

**Aim**

Implementation of CNN using keras network

**CO4**

Implement convolutional neural network algorithm using Keras framework.

**Program and Output**

import tensorflow as tf

from tensorflow.keras import datasets, layers, models

import matplotlib.pyplot as plt

(train\_images, train\_labels), (test\_images, test\_labels) = datasets.cifar10.load\_data()

train\_images, test\_images = train\_images / 255.0, test\_images / 255.0

class\_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',

               'dog', 'frog', 'horse', 'ship', 'truck']

plt.figure(figsize=(10,10))

for i in range(25):

    plt.subplot(5,5,i+1)

    plt.xticks([])

    plt.yticks([])

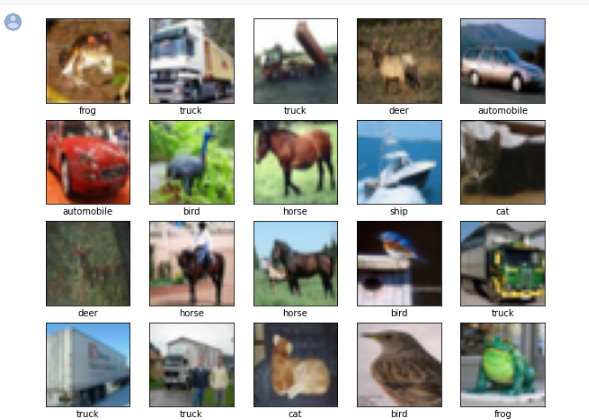
    plt.grid(False)

    plt.imshow(train\_images[i])

    plt.xlabel(class\_names[train\_labels[i][0]])

plt.show()

**Output**



model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)))

model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.summary()

model.add(layers.Flatten())

model.add(layers.Dense(64, activation='relu'))

model.add(layers.Dense(10))

model.summary()

**Output**

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 30, 30, 32) 896

max\_pooling2d (MaxPooling2D (None, 15, 15, 32) 0

)

conv2d\_1 (Conv2D) (None, 13, 13, 64) 18496

max\_pooling2d\_1 (MaxPooling (None, 6, 6, 64) 0

2D)

conv2d\_2 (Conv2D) (None, 4, 4, 64) 36928

flatten (Flatten) (None, 1024) 0

dense (Dense) (None, 64) 65600

dense\_1 (Dense) (None, 10) 650

=================================================================

Total params: 122,570

Trainable params: 122,570

Non-trainable params: 0

model.compile(optimizer='adam',

              loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

              metrics=['accuracy'])

history = model.fit(train\_images, train\_labels, epochs=5,

                    validation\_data=(test\_images, test\_labels))

**Output**

Epoch 1/5

1563/1563 [==============================] - 16s 5ms/step - loss: 1.5253 - accuracy: 0.4442 - val\_loss: 1.2627 - val\_accuracy: 0.5531

Epoch 2/5

1563/1563 [==============================] - 8s 5ms/step - loss: 1.1625 - accuracy: 0.5867 - val\_loss: 1.1056 - val\_accuracy: 0.6121

Epoch 3/5

1563/1563 [==============================] - 8s 5ms/step - loss: 1.0065 - accuracy: 0.6467 - val\_loss: 0.9735 - val\_accuracy: 0.6567

Epoch 4/5

1563/1563 [==============================] - 7s 5ms/step - loss: 0.9101 - accuracy: 0.6816 - val\_loss: 0.9356 - val\_accuracy: 0.6720

Epoch 5/5

1563/1563 [==============================] - 7s 5ms/step - loss: 0.8382 - accuracy: 0.7062 - val\_loss: 0.9111 - val\_accuracy: 0.6862

test\_loss, test\_acc = model.evaluate(test\_images,  test\_labels, verbose=2)

print(test\_acc)

Output

0.6862000226974487

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No. : 12**

**Aim**

Program to implement scrap of any website

**CO5**

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

**Program and Output**

import requests

from bs4 import BeautifulSoup

URL = "http://www.ajce.in"

r = requests.get(URL)

soup = BeautifulSoup(r.content, 'html5lib')

print(soup.prettify())

**Output**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8"/>

<title>

Amal Jyothi College of Engineering | B Tech honours, B Tech honours degree in ktu, FIRST ENGINEERING COLLEGE in Kerala to secure NAAC A grade. Engineering Admissions Kerala, KTU, Kerala Engineering Admissions, admissions in engineering, APJ Abdul Kalam Technological University, dual degree mca kerala, integrated MCA kerala, Kerala Technological University, Fiber optics training in kerala, Fiber optics training in kottayam, research promoting institution,institution for innovation,technolgy business incubator,IELTS training,GATE coaching,in-house internship,placement training,clean campus,beautiful campus, institution well connected by road,catholic institution, ANFOT, Fiber Training,best infrastructure engineering college kerala, MCA Colleges in Kerala, MCA in Engineering College Kerala, MCA LE College Kerala,Best MCA Course in Kerala, MCA Kerala, KTU MCA, Best College in KTU, Best College under KTU,Best MCA College under KTU,Best MCA College in KTU, highest intake engineering college kerala, top self financing engineering college in kerala, engineering admission, best engineering college kerala, nri girls hostel, top engineering colleges kerala, top 10 engineering colleges kerala, top 10 engineering colleges india, metallurgy, chemical engineering, civil admission kerala, mechanical admission kerala, computer science admission kerala, automobile admission kerala, eee, ece admissions, MCA 2 year, dual degree mca, integrated MCA, MCA best College, best engineering college, best college hostels, best food, top college in kerala, kerala top engineering college, amal jyothi, amal jyothi college of engineering, amal jyothi engineering college, amaljyothi, [www.amaljyothi.com](http://www.amaljyothi.com/), amal jyothi college of engineering kanjirapally, jyothi engineering college, amaljyothi college of engineering, ajce, jyothi college of engineering, jyothi college, B Tech in &amp; Construction Management, M Tech in Machine Design, M Tech in Power Electronics &amp; Power Systems, M Tech in Nano Technology, nanotechnology, nano science &amp; technology kerala,

</title>

<meta content="width=device-width, initial-scale=1" name="viewport"/>

<script type="text/javascript">

<!--

if (screen.width <= 699) {

document.location = "[https://m.ajce.in](https://m.ajce.in/)";

}

</script>

<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->

<link href="assets/css/main.css" rel="stylesheet"/>

<!--Bootstrap Stylesheet [ REQUIRED ]-->

<link href="css/bootstrap.css" rel="stylesheet"/>

<!--Nifty Stylesheet [ REQUIRED ]-->

<link href="css/nifty.css" rel="stylesheet"/>

<!--Animate.css [ OPTIONAL ]-->

<link href="css/animate.min.css" rel="stylesheet"/>

<link href="ajce.ico" rel="icon" type="image/ico"/>

<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->

<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->

<link href="../ajce.ico" rel="icon" type="image/ico"/>

<style>

.alert-title a{

border-bottom:0px;

}

</style>

</head>

<!--TIPS-->

<!--You may remove all ID or Class names which contain "demo-", they are only used for demonstration. -->

<body>

<script>

setTimeout(function(){

window.location.href = '<https://ajce.in/home/index.html>';

}, 10000);

</script>

<div class="effect aside-float aside-bright mainnav-lg" id="container">

</div>

<div id="wrapper">

<div id="bg">

</div>

<div id="overlay">

</div>

<div id="main">

<!-- Header -->

<header id="header">

<img alt="" height="100" src="300x300png.png" style="vertical-align:middle" width="100"/>

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No. : 13**

**Aim**

Program for Natural Language Processing which performs n-grams(Using inbuilt functions)

**CO5**

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

**Program and Output**

import nltk

from nltk.util import ngrams

text = "this is a very good book to study";

Ngrams = ngrams(sequence=nltk.wordpunct\_tokenize(text), n=3)

for grams in Ngrams:

 print(grams)

**Output**

('this', 'is', 'a')

('is', 'a', 'very')

('a', 'very', 'good')

('very', 'good', 'book')

('good', 'book', 'to')

('book', 'to', 'study')

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No. : 14**

**Aim**

Program for Natural Language Processing which perform parts of speech tagging.

**CO5**

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

**Program and Output**

import nltk

from nltk.tag import DefaultTagger

exptagger = DefaultTagger('NN')

exptagger.tag\_sents([['Hi', ','], ['How', 'are', 'you', '?']])

**Output**

[[('Hi', 'NN'), (',', 'NN')], [('How', 'NN'), ('are', 'NN'), ('you', 'NN'), ('?', 'NN')]]

import nltk

from nltk.tag import untag

untag([('Tutorials', 'NN'), ('Point', 'NN')])

**Output**

['Tutorials', 'Point']

sentence = """At eight o'clock on Thursday morning

Arthur didn't feel very good."""

tokens = nltk.word\_tokenize(sentence)

tagged = nltk.pos\_tag(tokens)

print(tagged)

**Output**

['At', 'eight', "o'clock", 'on', 'Thursday', 'morning', 'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']

[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'NN'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), ('Arthur', 'NNP'), ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')]

text ="learn php from guru99 and make study easy".split()

print("After Split:",text)

tokens\_tag = nltk.pos\_tag(text)

print("After Token:",tokens\_tag)

**Output**

After Split: ['learn', 'php', 'from', 'guru99', 'and', 'make', 'study', 'easy']

After Token: [('learn', 'JJ'), ('php', 'NN'), ('from', 'IN'), ('guru99', 'NN'), ('and', 'CC'), ('make', 'VB'), ('study', 'NN'), ('easy', 'JJ')]

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No. : 15**

**Aim:**

Data preprocessing with NLTK

1. Counting Tags
2. Bigrams
3. Trigrams
4. Stop Words
5. Stemming

**CO5**

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

**Program and Output**

!pip install -q wordcloud

import wordcloud

import nltk

nltk.download('stopwords')

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

import pandas as pd

import unicodedata

import numpy as np

import string

1. from collections import Counter

import nltk

text = "Guru99 is one of the best sites to learn WEB, SAP, Ethical Hacking and much more online."

lower\_case = text.lower()

tokens = nltk.word\_tokenize(lower\_case)

tags = nltk.pos\_tag(tokens)

counts = Counter( tag for word,  tag in tags)

print(counts)

Output

Counter({'NN': 5, ',': 2, 'VBZ': 1, 'CD': 1, 'IN': 1, 'DT': 1, 'JJS': 1, 'NNS': 1, 'TO': 1, 'VB': 1, 'JJ': 1, 'CC': 1, 'RB': 1, 'JJR': 1, '.': 1})

1. import nltk

text = "Guru99 is a totally new kind of learning experience."

Tokens = nltk.word\_tokenize(text)

output = list(nltk.bigrams(Tokens))

print(output)

**Output**

[('Guru99', 'is', 'a'), ('is', 'a', 'totally'), ('a', 'totally', 'new'), ('totally', 'new', 'kind'), ('new', 'kind', 'of'), ('kind', 'of', 'learning'), ('of', 'learning', 'experience'), ('learning', 'experience', '.')]

1. import nltk

text = "Guru99 is a totally new kind of learning experience."

Tokens = nltk.word\_tokenize(text)

output = list(nltk.trigrams(Tokens)) print(output)

**Output**

[('Guru99', 'is', 'a'), ('is', 'a', 'totally'), ('a', 'totally', 'new'), ('totally', 'new', 'kind'), ('new', 'kind', 'of'), ('kind', 'of', 'learning'), ('of', 'learning', 'experience'), ('learning', 'experience', '.')]

1. from nltk.corpus import stopwords

print(stopwords.words('english'))

en\_stopwords = stopwords.words('english')

def remove\_stopwords(text):

    result = []

    for token in text:

        if token not in en\_stopwords:

            result.append(token)

    return result

text = "this is the only solution of that question".split()  remove\_stopwords(text)

**Output**

['solution', 'question']

1. from nltk.stem import PorterStemmer

from nltk.tokenize import word\_tokenize

ps = PorterStemmer()

sentence = "Programmers program with programming languages"

words = word\_tokenize(sentence)

for w in words:

    print(w, " : ", ps.stem(w))

**Output**

Programmers : programm

program : program

with : with

programming : program

languages : languag

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.