Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

Aim: To obtain the classification of various classes using k-Nearest Neighbor approach

IDE: Google Colab

Theory:

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

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

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Methodology:

1. Load the basic libraries and packages
2. Load the dataset
3. Analyse the dataset
4. Pre-process the data

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

5. Visualize the Data
6. Separate the feature and prediction value columns
7. Select the number K of the neighbors
8. Calculate the Euclidean distance of K number of neighbors
9. Take the K nearest neighbors as per the calculated Euclidean distance.
10. Among these k neighbors, count the number of the data points in each category.
11. Assign the new data points to that category for which the number of the neighbor is maximum.
12. Our model is ready..

Pre Lab Exercise:

a. What is K in KNN-approach?

b. Which are the types of distance metrices?

c. What can be the criteria of selection of the value of K?

d. What are the advantages of KNN algorithm?

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

Program (Code/Results):

1. Importing Libraries and packages

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

2. Loading and describing the dataset

```
[ ] names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
    df = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data",names=names)
    df
```

	sepal-length	sepal-width	petal-length	petal-width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

3. Analyzing the data

```
[ ] df.shape
(150, 5)

[ ] df.describe()
```

	sepal-length	sepal-width	petal-length	petal-width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

4. Processing the class labels:

```
trainingClass = df.values[:,-1]
trainingClass
```

```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
```

5. Encoding each unique class label as a numeric value, preparing the labels for the k-NN algorithm.

Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Artificial Intelligence (01CT0616)	Aim: To obtain the classification of various classes using k-Nearest Neighbor approach	
Experiment No: 03	Date:	Enrollment No: 92200133003

```
unique_list = list(set(trainingClass))
unique_list
```

```
['Iris-versicolor', 'Iris-virginica', 'Iris-setosa']
```

```
for i in range(len(trainingClass)):
    for j  in range(len(unique_list)):
        if trainingClass[i] == unique_list[j]:
            trainingClass[i] = j
trainingClass
```

```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 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, 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, 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], dtype=object)
```

6. Calculating Euclidian Distance

```
import math
def euclidean_distance(rowsi,rowsj):
  distance = 0.0
  for i in range(len(rowsi)):
    distance += (rowsi[i] - rowsj[i])**2
  return (math.sqrt(distance))
```

7. Training and Testing of the data

```
training = df.values[:,0:4]
```

```
testing = df.values[149,0:4]
```

Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Artificial Intelligence (01CT0616)	Aim: To obtain the classification of various classes using k-Nearest Neighbor approach	
Experiment No: 03	Date:	Enrollment No: 92200133003

8. Calculating the distance and sorting it

```
[ ]  distance = []
     for i in range(len(training)):
       distance.append([euclidean_distance(training[i],testing),trainingClass[i]])

[ ]  distance.sort()
```

```
distance

[[0.0, 1],
 [0.282842712474618, 1],
 [0.31622776601683766, 1],
 [0.33166247903553997, 1],
 [0.33166247903553997, 1],
 [0.3605551275463989, 0],
 [0.37416573867739383, 0],
 [0.4582575694955842, 1],
 [0.4690415759823428, 1],
 [0.5385164807134497, 1],
 [0.5385164807134502, 1],
 [0.5830951894845301, 1],
 [0.6082762530298215, 0],
 [0.6244997998398398, 1],
 [0.6403124237432847, 1],
 [0.6403124237432849, 1],
 [0.6480740698407857, 1],
 [0.6480740698407862, 1],
 [0.6557438524301997, 1],
 [0.6633249580710797, 1],
 [0.6708203932499363, 0],
 [0.6708203932499368, 0],
 [0.6855654600401041, 0],
 [0.7211102550927978, 1],
 [0.734846922834953, 0],
 [0.7348469228349535, 0],
 [0.7549834435270746, 0],
 [0.7615773105863908, 1],
```

```
 [4.106093033529562, 2],
 [4.110960958218893, 2],
 [4.113392760240626, 2],
 [4.1400483088968905, 2],
 [4.141255848169731, 2],
 [4.141255848169731, 2],
 [4.149698784249286, 2],
 [4.153311931459037, 2],
 [4.172529209005013, 2],
 [4.173727350941841, 2],
 [4.178516483155236, 2],
 [4.189272013130682, 2],
 [4.2130748865881795, 2],
 [4.217819341792628, 2],
 [4.217819341792628, 2],
 [4.224926034855522, 2],
 [4.240283009422838, 2],
 [4.298837052040936, 2],
 [4.302324952859791, 2],
 [4.315089802078283, 2],
 [4.33358973600409, 2],
 [4.374928570845472, 2],
 [4.387482193696061, 2],
 [4.3920382511995495, 2],
 [4.628174586162454, 2],
 [4.631414470763764, 2]]
```

9. Giving k a value and implementing the voting mechanism and getting the maximum from it

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

```
k=5
voting = [0 for i in range(len(unique_list))]
for i in range(1,k+1):
  voting[distance[i][1]] = voting[distance[i][1]] + 1
```

```
[ ] voting
```

```
[1, 4, 0]
```

```
[ ] unique_list[voting.index(max(voting))]
```

```
'Iris-virginica'
```

Observation and Result Analysis:

a. Nature of the dataset

b. During Training Process

c. After the training Process

d. Observation over the Learning Curve

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To obtain the classification of various classes using k-Nearest Neighbor approach**	
Experiment No: 03	**Date:**	**Enrollment No: 92200133003**

Post Lab Exercise:

a. Is KNN useful in regression-based problems?

b. Why KNN is a non-parametric algorithm?

c. What are the assumptions of KNN approach?

d. How can the KNN approach make the prediction of the unseen dataset?

e. Why is KNN called a Lazy Learner?

f. Why should we not use KNN for large dataset?