



COURSE LABORATORY MANUAL

1. EXPERIMENT NO: 9

2. TITLE: **K-NEAREST NEIGHBOUR**

3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

4. AIM:

- Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

5. THEORY:

- K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data.

• Algorithm

Input: Let m be the number of training data samples. Let p be an unknown point.

Method:

1. Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
2. for i=0 to m
 Calculate Euclidean distance d(arr[i], p).
3. Make set S of K smallest distances obtained. Each of these distances correspond to an already classified data point.
4. Return the majority label among S.

6. PROCEDURE / PROGRAMME :

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import datasets
```

```
iris=datasets.load_iris()
print("Iris Data set loaded...")
```

```
x_train, x_test, y_train, y_test = train_test_split(iris.data,iris.target,test_size=0.1)
#random_state=0
```

```
print("Dataset is split into training and testing samples...")
print("Size of training data and its label",x_train.shape,y_train.shape)
print("Size of training data and its label",x_test.shape, y_test.shape)
for i in range(len(iris.target_names)):
    print("Label", i , "-",str(iris.target_names[i]))
```

```
classifier = KNeighborsClassifier(n_neighbors=1)
```

```
classifier.fit(x_train, y_train)
```



COURSE LABORATORY MANUAL

```
y_pred=classifier.predict(x_test)

print("Results of Classification using K-nn with K=1 ")
for r in range(0,len(x_test)):
    print(" Sample:", str(x_test[r]), " Actual-label:", str(y_test[r]),
          " Predicted-label:", str(y_pred[r]))

print("Classification Accuracy :", classifier.score(x_test,y_test));
```

7. RESULTS & CONCLUSIONS:

Result-1

Iris Data set loaded...
Dataset is split into training and testing samples...
Size of training data and its label (135, 4) (135,)
Size of training data and its label (15, 4) (15,)
Label 0 - setosa
Label 1 - versicolor
Label 2 - virginica
Results of Classification using K-nn with K=1
Sample: [4.4 3. 1.3 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 2.5 3. 1.1] Actual-label: 1 Predicted-label: 1
Sample: [6.1 2.8 4. 1.3] Actual-label: 1 Predicted-label: 1
Sample: [6. 2.7 5.1 1.6] Actual-label: 1 Predicted-label: 2
Sample: [6.7 2.5 5.8 1.8] Actual-label: 2 Predicted-label: 2
Sample: [5.1 3.8 1.5 0.3] Actual-label: 0 Predicted-label: 0
Sample: [6.7 3.1 4.4 1.4] Actual-label: 1 Predicted-label: 1
Sample: [4.8 3.4 1.6 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 3.5 1.4 0.3] Actual-label: 0 Predicted-label: 0
Sample: [5.4 3.7 1.5 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.7 2.8 4.1 1.3] Actual-label: 1 Predicted-label: 1
Sample: [4.5 2.3 1.3 0.3] Actual-label: 0 Predicted-label: 0
Sample: [4.4 2.9 1.4 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 3.5 1.4 0.2] Actual-label: 0 Predicted-label: 0
Sample: [6.2 3.4 5.4 2.3] Actual-label: 2 Predicted-label: 2
Classification Accuracy : 0.93

Result-2

Iris Data set loaded...
Dataset is split into training and testing samples...
Size of training data and its label (135, 4) (135,)
Size of training data and its label (15, 4) (15,)
Label 0 - setosa
Label 1 - versicolor
Label 2 - virginica
Results of Classification using K-nn with K=1
Sample: [6.5 3. 5.5 1.8] Actual-label: 2 Predicted-label: 2
Sample: [5.7 2.8 4.1 1.3] Actual-label: 1 Predicted-label: 1
Sample: [6.6 3. 4.4 1.4] Actual-label: 1 Predicted-label: 1



COURSE LABORATORY MANUAL

Sample: [6.9 3.1 5.1 2.3] Actual-label: 2 Predicted-label: 2
Sample: [5.1 3.8 1.9 0.4] Actual-label: 0 Predicted-label: 0
Sample: [7.2 3.2 6. 1.8] Actual-label: 2 Predicted-label: 2
Sample: [5.5 2.6 4.4 1.2] Actual-label: 1 Predicted-label: 1
Sample: [6. 2.9 4.5 1.5] Actual-label: 1 Predicted-label: 1
Sample: [5.1 3.7 1.5 0.4] Actual-label: 0 Predicted-label: 0
Sample: [5.2 3.4 1.4 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5. 3.5 1.6 0.6] Actual-label: 0 Predicted-label: 0
Sample: [4.9 3.1 1.5 0.1] Actual-label: 0 Predicted-label: 0
Sample: [5. 3. 1.6 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.7 3. 4.2 1.2] Actual-label: 1 Predicted-label: 1
Sample: [5.8 2.7 5.1 1.9] Actual-label: 2 Predicted-label: 2
Classification Accuracy : 1.0

8. LEARNING OUTCOMES :

- The student will be able to implement k-Nearest Neighbour algorithm to classify the iris data set and Print both correct and wrong predictions.

9. APPLICATION AREAS:

- Recommender systems
- Classification problems

10. REMARKS: