**U19EC046 | ML | LAB 4**

**AIM**

Write a program to implement the K nearest neighbour classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

**ALGORITHM**

1. Import necessary libraries
2. Read the dataset using pandas
3. Split the dataset into features and results
4. Split the features and results into training and testing dataset using scikit learn

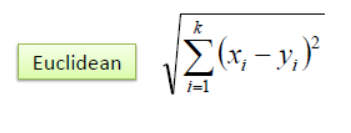
test\_train\_split

1. Apply pre-processing if needed, use scikit-learn preprocessing class
2. Create an instance of KNN classifier, tune properly taking k the numer of neighbors
3. Train the model using fit method
4. Predict the result using predict method on model
5. Find the accuracy and confusion matrix using scikit-learn matrices

**THEORY:**

K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbours are classified.

In the classification setting, the K-nearest neighbor algorithm essentially boils down to forming a majority vote between the K most similar instances to a given “unseen” observation. Similarity is defined according to a distance metric between two data points. A popular one is the Euclidean distance method



**Pros of KNN**

1. Simple to implement
2. Flexible to feature/distance choices
3. Naturally handles multi-class cases
4. Can do well in practice with enough representative data

**Cons of KNN**

1. Need to determine the value of parameter K (number of nearest neighbours)
2. Computation cost is quite high because we need to compute the distance of each query instance to all training samples.
3. Storage of data
4. Must know we have a meaningful distance function.

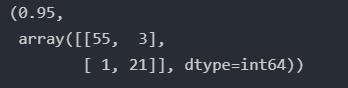
**CODE**

1. KNN classifier

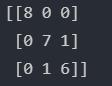
|  |
| --- |
| **from sklearn.datasets import load\_iris**  **from sklearn.model\_selection import train\_test\_split**  **from sklearn.neighbors import KNeighborsClassifier**  **from sklearn.metrics import confusion\_matrix, accuracy\_score**  **from sklearn.preprocessing import StandardScaler, RobustScaler**  **import numpy as np**  **import pandas as pd**  **df = pd.read\_csv('Social\_Network\_Ads.csv')**  **X = df.iloc[:,:-1]**  **Y = df.iloc[:,-1]**  **x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=0)**  **sc = StandardScaler()**  **x\_train = sc.fit\_transform(x\_train)**  **x\_test = sc.transform(x\_test)**  **model = KNeighborsClassifier(n\_neighbors=3, metric='euclidean')**  **model.fit(x\_train, y\_train)**  **model.score(x\_test, y\_test), confusion\_matrix(y\_pred=model.predict(x\_test), y\_true=y\_test)** |

**OUTPUT**

1. KNN classifier
2. Social networking Ads



1. Iris Dataset





**CONCLUSION**

In this practical we have studied and implements KNN classifier using scikit-learn library.