**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

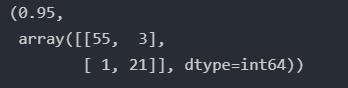
**CODE**

1. KNN classifier

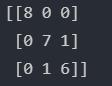
|  |
| --- |
| **from sklearn.datasets import load\_iris**  **from sklearn.model\_selection import train\_test\_split**  **from sklearn.neighbors import KNeighborsClassifier**  **from sklearn.metrics import confusion\_matrix, accuracy\_score**  **from sklearn.preprocessing import StandardScaler, RobustScaler**  **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.