**51.How to implement KNN algorithm in python?**

**Objective:**

* To implement KNN algorithm using python.

**Process:**

* Import library.
* Read inbuild data set.
* Declare X and Y variable.
* Split the data into train and test.
* Fit the X and Y in to the model.
* Make predictions.
* Calculate confusion matrix and classification report.

**Input:**

* Iris data set.

**Output:**

* Confusion matrix
* Classification report

**Source code:**

#import library functions

import pandas as pd

#took inbuild iris data set

url = ("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data")

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

dataset = pd.read\_csv(url, names=names)

print(dataset.head())

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 4].values

#print if you wnat to know X and y values

#print(X)

#print(y)

#split the data set

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

#split the sample data in to train and test data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

#print(X\_train)

#print(y\_train)

#scaling the input train and test data

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(X\_train)

#scaling the data

X\_train = scaler.transform(X\_train)

X\_test = scaler.transform(X\_test)

#print(X\_train)

#print(y\_train)

#print(X\_train.shape)

#import KNN algorithm from sklearn

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=3)

result=classifier.fit(X\_train, y\_train)

print(result)

#make prediction

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

#print the confusion matrix and classification report

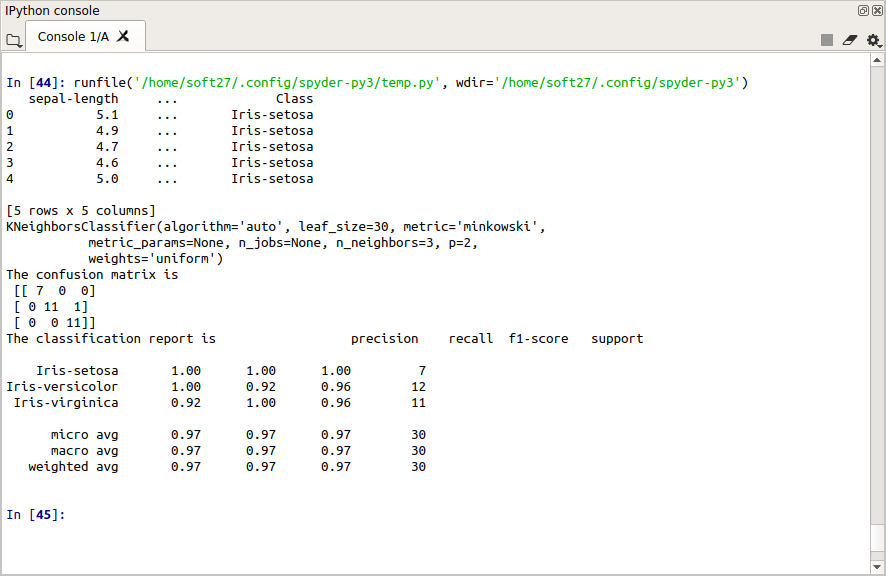
from sklearn.metrics import classification\_report, confusion\_matrix

print("The confusion matrix is\n",confusion\_matrix(y\_test, y\_pred))

print("The classification

report is",classification\_report(y\_test, y\_pred))

**Screen shot:**

****