

## Program 8

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

### Program

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

iris_dataset = datasets.load_iris()
print("Iris dataset is loaded successfully")
print("*****")

x_train, x_test, y_train, y_test = train_test_split(iris_dataset.data, iris_dataset.target,
test_size=0.1)

for i in range(len(iris_dataset.target_names)):
    print("Label", i, "->", str(iris_dataset.target_names[i]))

print("*****")

KNNClassifier = KNeighborsClassifier(n_neighbors=2)
KNNClassifier.fit(x_train, y_train)
y_pred = KNNClassifier.predict(x_test)

print("Result of the Classification using KNN with K=1")
print("-----")
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:", KNNClassifier.score(x_test, y_test))
    print("-----")
```

## Result

Iris dataset is loaded successfully

\*\*\*\*\*

Label 0 -> setosa

Label 1 -> versicolor

Label 2 -> virginica

\*\*\*\*\*

Result of the Classification using KNN with K=1

-----

Sample: [5.1 3.8 1.5 0.3] Actual Label: 0 Predicted Label: 0

Classification Accuracy: 1.0

-----

Sample: [6.4 3.2 5.3 2.3] Actual Label: 2 Predicted Label: 2

Classification Accuracy: 1.0

-----

Sample: [5.7 4.4 1.5 0.4] Actual Label: 0 Predicted Label: 0

Classification Accuracy: 1.0

-----

Sample: [5.6 3. 4.1 1.3] Actual Label: 1 Predicted Label: 1

Classification Accuracy: 1.0

-----