

Program-8: 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.

```
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
for i in range(len(iris.target_names)):
    print("Label", i , "-",str(iris.target_names[i]))
classifier = KNeighborsClassifier(n_neighbors=2)
classifier.fit(x_train, y_train)
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));
```

Output:

```
Iris Data set loaded...
Label 0 - setosa
Label 1 - versicolor
Label 2 - virginica
Results of Classification using K-nn with K=1
Sample: [5.  3.6 1.4 0.2] Actual-label: 0 Predicted-label: 0
Classification Accuracy : 0.9333333333333333
Sample: [4.5 2.3 1.3 0.3] Actual-label: 0 Predicted-label: 0
Classification Accuracy : 0.9333333333333333
Sample: [5.1 3.5 1.4 0.3] Actual-label: 0 Predicted-label: 0
Classification Accuracy : 0.9333333333333333
Sample: [6.1 2.6 5.6 1.4] Actual-label: 2 Predicted-label: 1
Classification Accuracy : 0.9333333333333333
Sample: [4.4 2.9 1.4 0.2] Actual-label: 0 Predicted-label: 0
Classification Accuracy : 0.9333333333333333
Sample: [5.2 3.5 1.5 0.2] Actual-label: 0 Predicted-label: 0
Classification Accuracy : 0.9333333333333333
Sample: [6.2 3.4 5.4 2.3] Actual-label: 2 Predicted-label: 2
```