Date .....

Expt. No. ....6.... Page No....32....

write a program to implement k-Nearest Neighbour algorithm to classify the iris dataset. Print both correct & 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)
print("Dataset is split into training and testing...")
print("Size of training data and its label", x_train.
                                shape, y_train.shape)
print("Size of testing data and its label", x_test.
                                shape, y_test.shape)
for i in range(len(iris.target_names)):
    print("Label", i, "-", str(iris.target_names[i]))
classifier = KNeighborsClassifier(n_neighbors = 1)
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));
```

Date .....

Expt. No. .....6..... Page No. ....34.....

```
from sklearn.metrics import classification_report,
                            confusion_matrix
from sklearn.metrics import accuracy_score
print('Confusion matrix')
print(confusion_matrix(y_test, y_pred))
print('accuracy matrix')
print(classification_report(y_test, y_pred))
print("correct prediction", accuracy_score(y_test,
                                            y_pred))
print("wrong prediction", 1-accuracy_score(y_test,
                                            y_pred))
```

Output:

```
Iris Data set loaded
Dataset is split into training and testing...
size of training data and its label (135, 4) (135,)
size of testing data and its label (15, 4) (15,)
Label 0 - setosa
Label 1 - versicolor
Label 2 - virginica
Result of classification using k-nn with k = 1
Sample : [6.4 3.1 5.5 1.8] Actual Label : 2 Predicted label: 2
Sample : [6.5 3.  5.8 2.2] Actual Label : 2 Predicted label: 2
Sample : [4.9 3.1 1.5 0.1] Actual Label : 0 Predicted label: 0
Sample : [5.6 3.  4.5 1.5] Actual Label : 1 Predicted label: 1
Sample : [4.6 3.2 1.4 0.2] Actual Label : 0 Predicted label: 0
Sample : [5.5 2.4 3.7 1. ] Actual Label : 1 Predicted label: 1
Sample : [6.4 2.9 4.3 1.3] Actual Label : 1 Predicted label: 1
Sample : [6.  3.  4.8 1.8] Actual Label : 2 Predicted label: 2
Sample : [6.  2.2 4.  1. ] Actual Label : 1 Predicted label: 1
Sample : [5.1 3.5 1.4 0.2] Actual Label : 0 Predicted label: 0
Sample : [5.5 3.5 1.3 0.2] Actual Label : 0 Predicted label: 0
Sample : [5.  2.  3.5 1. ] Actual Label : 1 Predicted label: 1
Sample : [5.  3.5 1.6 0.6] Actual Label : 0 Predicted label: 0
Sample : [5.  3.4 1.6 0.4] Actual Label : 0 Predicted label: 0
Sample : [5.8 2.7 5.1 1.9] Actual Label : 2 Predicted label: 2
```

Classification Accuracy : 1.0

confusion matrix

[[6 0 0]
 [0 5 0]
 [0 0 4]]

accuracy matrics

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6
1	1.00	1.00	1.00	5
2	1.00	1.00	1.00	4
accuracy			1.00	15
macro avg	1.00	1.00	1.00	15
weighted avg	1.00	1.00	1.00	15

correct prediction 1.0

wrong prediction 0.0