

Name : Uddip Yalamanchili

USN : 1PE16CS170

PROGRAM :

```

from sklearn import preprocessing
from sklearn import datasets
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split #Use this for Sk Learn 0.20 version
#from sklearn.cross_validation import train_test_split # Used for older versions of sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn import metrics

#Input Data-->read the dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
dataset = pd.read_csv("Data_8_9.csv", names = names)
#print(dataset.head())

#seperate X and Y values
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:,4].values

...

iris = datasets.load_iris()

x = (iris.data)
x.columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']
x_norm = preprocessing.normalize(x)

y = (iris.target)
y.columns = ['Targets']

...

#split the dataset ----->try various ratios here
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)

#scale the training data and test data
scaler = StandardScaler()
scaler.fit(x_train)

x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)

#change the neighbors and see the effect
#Model Creation
classifier = KNeighborsClassifier(n_neighbors = 3 )
classifier.fit(x_train, y_train)

#Prediction
y_pred = classifier.predict(x_test)

for i in range(len(y_pred)):
    print ("Training Example : ",x_test[i])
    #print(x_test[i])
    print ("Actual Label : ",y_test[i])
    #print(y_test[i])
    print ("Predicted Label : ",y_pred[i])
    #print (y_pred[i])
    print ("-----")

print ("Confusion Matrix : ")
print(confusion_matrix(y_test, y_pred))
print ("")
print(metrics.accuracy_score(y_test, y_pred)*100)
print ("Classification Report : ")
print(classification_report(y_test, y_pred))

```

output :

```
uddipyalamanchili@pes:~/programs/knn$ python3 knn.py
Training Example : [0.54797785 0.54791629 0.30431143 0.30363935]
Actual Label : versicolor
Predicted Label : versicolor
-----
Training Example : [0.43542784 0.43529448 0.43734375 0.43689866]
Actual Label : versicolor
Predicted Label : versicolor
-----
Training Example : [-1.42164729 -1.42296537 -1.2920764 -1.29547244]
Actual Label : setosa
Predicted Label : setosa
-----
Training Example : [-0.01477219 -0.01519275 -0.22781785 -0.22939792]
Actual Label : versicolor
Predicted Label : versicolor
-----
Training Example : [0.71680286 0.71684901 0.43734375 0.43689866]
Actual Label : virginica
Predicted Label : versicolor
-----
Training Example : [1.05445288 1.05471443 1.36856998 1.36971387]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [1.11072789 1.11102534 1.5016023 1.50297319]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [1.67347793 1.67413439 1.10250534 1.10319524]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [1.3358279 1.33626896 0.96947302 0.96993592]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [-1.25282227 -1.25403265 -1.02601176 -1.02895381]
Actual Label : setosa
Predicted Label : setosa
-----
Training Example : [1.67347793 1.67413439 1.36856998 1.36971387]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [0.15405282 0.15373996 0.17127911 0.17038003]
Actual Label : versicolor
Predicted Label : versicolor
-----
Training Example : [-0.1273222 -0.12781456 -0.22781785 -0.22939792]
Actual Label : versicolor
Predicted Label : versicolor
-----
Training Example : [-1.19654727 -1.19772175 -1.2920764 -1.29547244]
Actual Label : setosa
Predicted Label : setosa
-----
Training Example : [-1.30909728 -1.31034356 -1.2920764 -1.29547244]
Actual Label : setosa
Predicted Label : setosa
-----
Training Example : [0.88562787 0.88578172 0.96947302 0.96993592]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [0.99817788 0.99840353 1.23553766 1.23645455]
Actual Label : virginica
Predicted Label : virginica
-----
Training Example : [0.66052786 0.6605381 0.43734375 0.43689866]
```

Actual Label : versicolor
Predicted Label : versicolor

Training Example : [0.32287783 0.32267268 0.17127911 0.17038003]

Actual Label : versicolor
Predicted Label : versicolor

Training Example : [0.77307787 0.77315991 0.43734375 0.43689866]

Actual Label : virginica
Predicted Label : versicolor

Confusion Matrix :

```
[[4 0 0]
 [0 7 0]
 [0 2 7]]
```

90.0

Classification Report :

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	4
versicolor	0.78	1.00	0.88	7
virginica	1.00	0.78	0.88	9
avg / total	0.92	0.90	0.90	20

uddipyalamanchili@pes:~/programs/knn\$