```
# Python program to demonstrate
# KNN classification algorithm
# on IRIS dataset
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
from sklearn.model_selection import train_test_split
iris_dataset=load_iris()
print("\n IRIS FEATURES \ TARGET NAMES: \n ", iris_dataset.target_names)
for i in range(len(iris_dataset.target_names)):
    print("\n[{0}]:[{1}]".format(i,iris_dataset.target_names[i]))
```

```
 IRIS FEATURES \ TARGET NAMES:
  ['setosa' 'versicolor' 'virginica']

[0]:[setosa]

[1]:[versicolor]

[2]:[virginica]
```

```
print("\n IRIS DATA :\n",iris_dataset["data"])
X_train, X_test, y_train, y_test = train_test_split(iris_dataset["data"], iris_dataset["target"], random_state=0)
```

```
[ 6.   3.   4.8  1.8]
[ 6.9  3.1  5.4  2.1]
[ 6.7  3.1  5.6  2.4]
[ 6.9  3.1  5.1  2.3]
[ 5.8  2.7  5.1  1.9]
[ 6.8  3.2  5.9  2.3]
[ 6.7  3.3  5.7  2.5]
[ 6.7  3.   5.2  2.3]
[ 6.3  2.5  5.   1.9]
[ 6.5  3.   5.2  2. ]
[ 6.2  3.4  5.4  2.3]
[ 5.9  3.   5.1  1.8]]
```

```
print("\n Target :\n",iris_dataset["target"])
```

```
Target :
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2]
```

```
print("\n X TRAIN \n", X_train)
n_train=X_train.shape[0]
print(n_train)
```

```
[ 5.9  3.2  4.8  1.8]
 [ 6.3  2.3  4.4  1.3]
 [ 5.5  3.5  1.3  0.2]
 [ 5.1  3.7  1.5  0.4]
 [ 4.9  3.1  1.5  0.1]
 [ 6.3  2.9  5.6  1.8]
 [ 5.8  2.7  4.1  1. ]
 [ 7.7  3.8  6.7  2.2]
 [ 4.6  3.2  1.4  0.2]]
112
```

```
print("\n X TEST \n", X_test)
n_test=X_test.shape[0]
print(n_test)
```

```
 X TEST
 [[ 5.8  2.8  5.1  2.4]
 [ 6.   2.2  4.   1. ]
 [ 5.5  4.2  1.4  0.2]
 [ 7.3  2.9  6.3  1.8]
 [ 5.   3.4  1.5  0.2]
 [ 6.3  3.3  6.   2.5]
 [ 5.   3.5  1.3  0.3]
 [ 6.7  3.1  4.7  1.5]
 [ 6.8  2.8  4.8  1.4]
 [ 6.1  2.8  4.   1.3]
 [ 6.1  2.6  5.6  1.4]
 [ 6.4  3.2  4.5  1.5]
 [ 6.1  2.8  4.7  1.2]
 [ 6.5  2.8  4.6  1.5]
 [ 6.1  2.9  4.7  1.4]
 [ 4.9  3.1  1.5  0.1]
 [ 6.   2.9  4.5  1.5]
 [ 5.5  2.6  4.4  1.2]
 [ 4.8  3.   1.4  0.3]
 [ 5.4  3.9  1.3  0.4]
 [ 5.6  2.8  4.9  2. ]
 [ 5.6  3.   4.5  1.5]
 [ 4.8  3.4  1.9  0.2]
 [ 4.4  2.9  1.4  0.2]
 [ 6.2  2.8  4.8  1.8]
 [ 4.6  3.6  1.   0.2]
 [ 5.1  3.8  1.9  0.4]
 [ 6.2  2.9  4.3  1.3]
 [ 5.   2.3  3.3  1. ]
 [ 5.   3.4  1.6  0.4]
 [ 6.4  3.1  5.5  1.8]
 [ 5.4  3.   4.5  1.5]
 [ 5.2  3.5  1.5  0.2]
 [ 6.1  3.   4.9  1.8]
 [ 6.4  2.8  5.6  2.2]
 [ 5.2  2.7  3.9  1.4]
 [ 5.7  3.8  1.7  0.3]
 [ 6.   2.7  5.1  1.6]]
38
```

```
print("\n Y TRAIN \n", y_train)
```

```
 Y TRAIN
 [1 1 2 0 2 0 0 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0 1
 0 2 1 0 1 2 1 0 2 2 2 2 0 0 2 2 0 2 0 2 2 0 0 2 0 0 0 1 2 2 0 0 0 1 1 0 0
 1 0 2 1 2 1 0 2 0 2 0 0 2 0 2 1 1 1 2 2 1 1 0 1 2 2 0 1 1 1 1 0 0 0 2 1 2
 0]
```

```
print("\n Y TEST \n", y_test)
```

```
 Y TEST
 [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 1]
```

```
kn = KNeighborsClassifier(n_neighbors=3)
kn.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
```

```
x_new = np.array([[5, 2.9, 1, 0.2]])
print("\n XNEW \n",x_new)
```

```
 XNEW
 [[ 5.   2.9  1.   0.2]]
```

```
prediction = kn.predict(x_new)
```

```
print("\n Predicted target value: {}\n".format(prediction))
```

```
 Predicted target value: [0]
```

```
print("\n Predicted feature name: {}\n".format (iris_dataset["target_names"][prediction]))
```

```
 Predicted feature name: ['setosa']
```

```
i=1
x= X_test[i]
x_new = np.array([x])
print("\n XNEW \n",x_new)
for i in range(len(X_test)):
    x = X_test[i]
    x_new = np.array([x])
    prediction = kn.predict(x_new)
    print("\n Actual : {0} {1}, Predicted :{2}{3}".format(y_test[i],iris_dataset["target_names"][y_test[i]],prediction,
```

```
 Actual : 1 versicolor, Predicted :[1]['versicolor']

 Actual : 2 virginica, Predicted :[2]['virginica']

 Actual : 1 versicolor, Predicted :[1]['versicolor']

 Actual : 1 versicolor, Predicted :[1]['versicolor']

 Actual : 1 versicolor, Predicted :[1]['versicolor']
```

```
Actual : 2 virginica, Predicted :[2]['virginica']
Actual : 0 setosa, Predicted :[0]['setosa']
Actual : 0 setosa, Predicted :[0]['setosa']
Actual : 1 versicolor, Predicted :[1]['versicolor']
Actual : 1 versicolor, Predicted :[1]['versicolor']
Actual : 0 setosa, Predicted :[0]['setosa']
Actual : 2 virginica, Predicted :[2]['virginica']
Actual : 1 versicolor, Predicted :[1]['versicolor']
Actual : 0 setosa, Predicted :[0]['setosa']
Actual : 2 virginica, Predicted :[2]['virginica']
Actual : 2 virginica, Predicted :[2]['virginica']
Actual : 1 versicolor, Predicted :[1]['versicolor']
Actual : 0 setosa, Predicted :[0]['setosa']
Actual : 1 versicolor, Predicted :[2]['virginica']
```

```
print("\n TEST SCORE[ACCURACY]: {:.2f}\n".format(kn.score(X_test, y_test)))
```

```
TEST SCORE[ACCURACY]: 0.97
```