Experiment: 2.3

Student Name: Nishant Kumar Mehta	**UID:** 21BCS-*3402*
Branch: CSE	**Section/Group:** IOT-602/B
Semester: 5th	**Date:** 5/10/2023
Subject Name: AIML Lab	**Subject Code:** 21CSH-316

AIM: Implement K-Nearest Neighbor on any data set

1. Tools/Resource Used:

Goggle CoLab (Online Compiler)

2. Algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

3. Program Code:

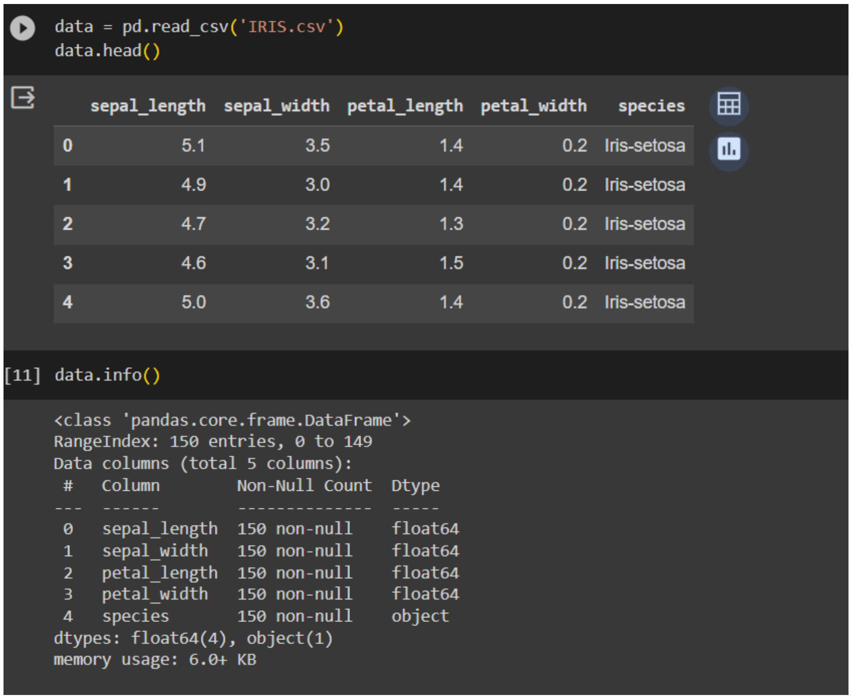
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import precision_recall_curve, roc_auc_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

```
data = pd.read_csv('IRIS.csv')
data.head()
data.info()

tmp = data.drop('species', axis=1)
tmp.head()

X = data.drop(['species', 'species'], axis=1)
y = data['species']

print(X.shape)
print(y.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=5)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

k_range = list(range(1,26))
scores=[]
for k in k_range:
   knn = KNeighborsClassifier(n_neighbors=k)
   knn.fit(X, y)
   y_pred = knn.predict(X)
   scores.append(metrics.accuracy_score(y,y_pred))

plt.plot(k_range,scores)
plt.xlabel('Value of k for KNN')
plt.ylabel('Accuracy Score')
plt.title('Accuracy Scores for Values of k of KNN')
plt.show()

k_range = list(range(1,26))
scores=[]
for k in k_range:
  knn = KNeighborsClassifier(n_neighbors=k)
  knn.fit(X_train, y_train)
  y_pred = knn.predict(X_test)
  scores.append(metrics.accuracy_score(y_test, y_pred))

plt.plot(k_range,scores)
plt.xlabel('Value of k for KNN')
plt.ylabel('Accuracy Score')
plt.title('Accuracy Scores for Values of k of KNN')
plt.show()

knn = KNeighborsClassifier(n_neighbors=12)
```

knn.fit(X, y)

#make a prediction for an example of an output sample observation
prediction = knn.predict([[6,3,4,2]])
print(prediction)

4. Output/Result:

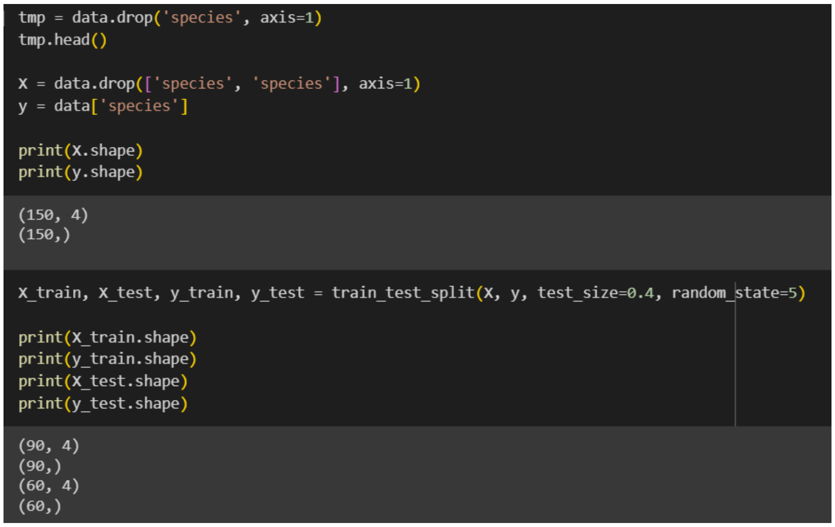


```
data = pd.read_csv('IRIS.csv')
data.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
[11] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

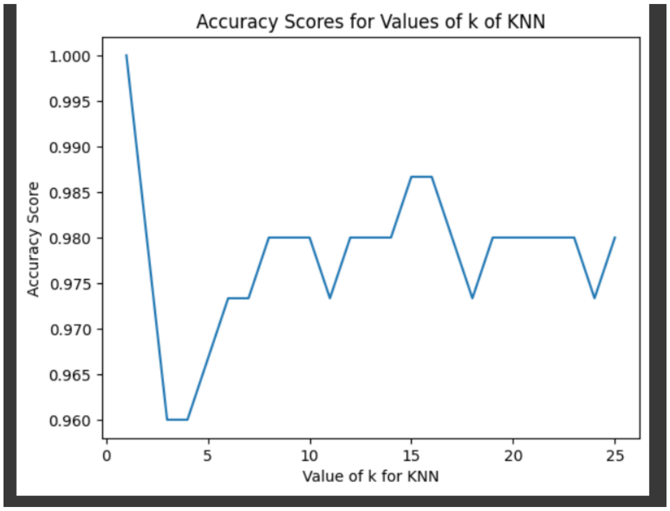


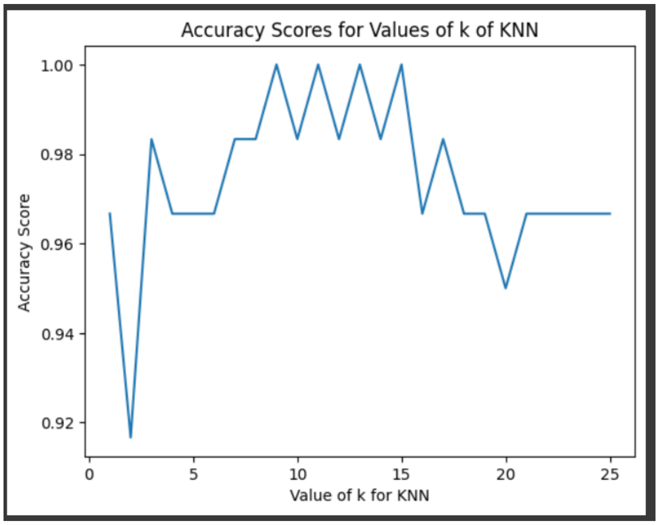
```
tmp = data.drop('species', axis=1)
tmp.head()

X = data.drop(['species', 'species'], axis=1)
y = data['species']

print(X.shape)
print(y.shape)
```

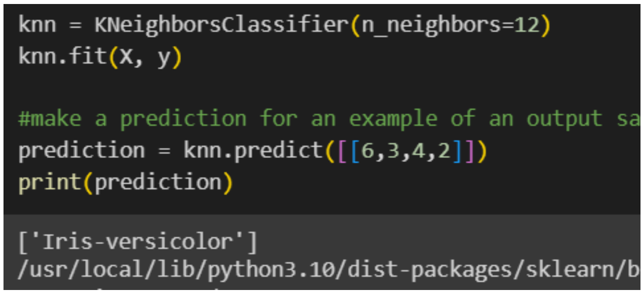
```
(150, 4)
(150,)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=5)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(90, 4)
(90,)
(60, 4)
(60,)
```







```
knn = KNeighborsClassifier(n_neighbors=12)
knn.fit(X, y)

#make a prediction for an example of an output sa
prediction = knn.predict([[6,3,4,2]])
print(prediction)
```

```
['Iris-versicolor']
/usr/local/lib/python3.10/dist-packages/sklearn/b
```

5. Learning Outcomes:

- To Learn about Meta-data and different clustering functions
- To learn About Different KNN Techniques
- To Learn about Cluster Model or algorithms