Experiment – 6

Student Name: Vivek Kumar **UID: 21BCS8129**
Branch: BE-CSE(LEET) **Section/Group: WM-20BCS-616/A**
Semester: 5th **Date of Performance: 12/10/2022**
Subject Name: Machine Learning Lab **Subject Code: 20CSP-317**

1. Aim/Overview of the practical:

Implement K-Nearest Neighbor on any data set

2. Task to be done/ Which logistics used:

Implement K-Nearest Neighbor on any data set using sklearn.

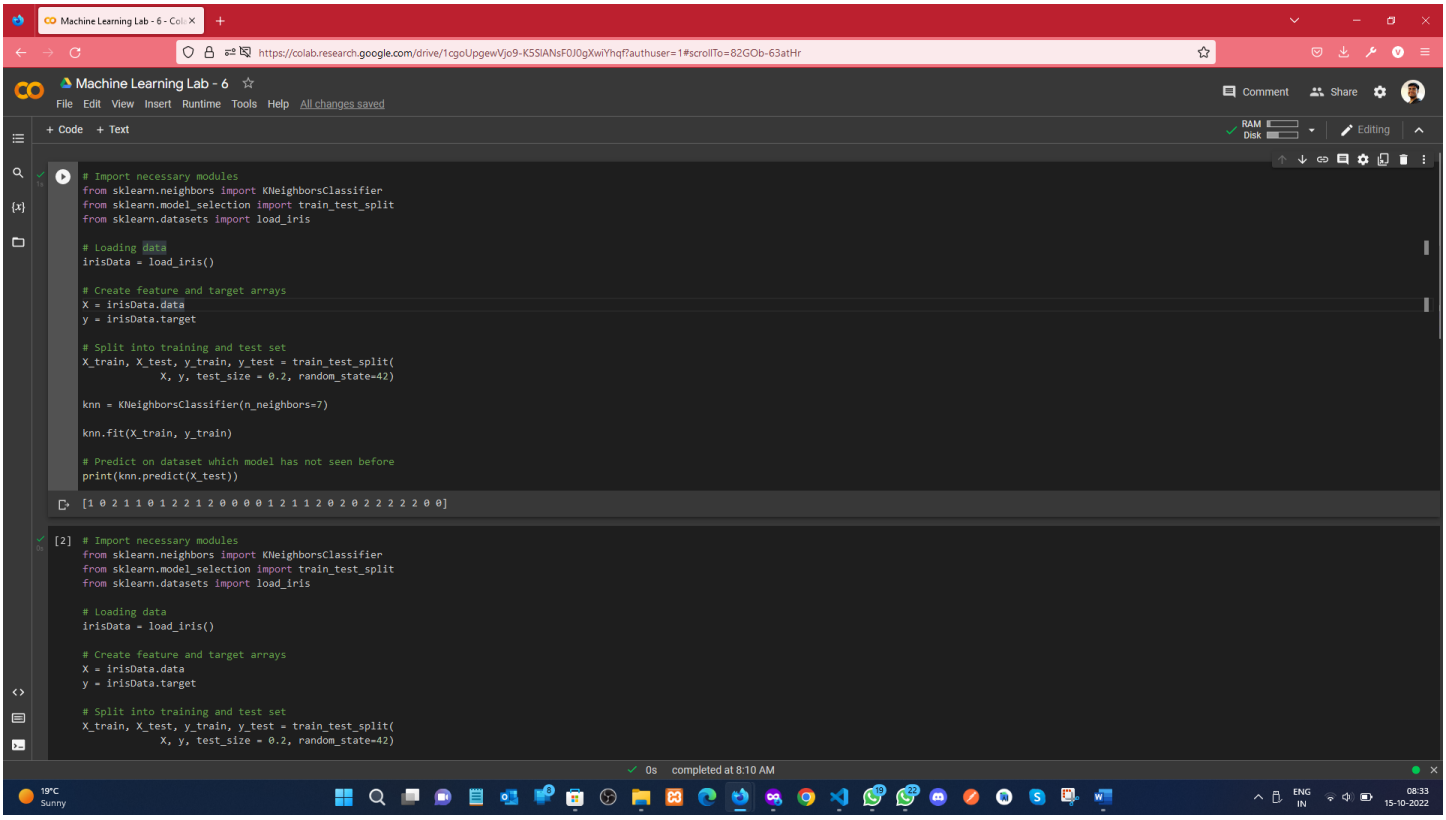
3. Steps for experiment/practical/Code:

```
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Loading data
irisData = load_iris()
# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size = 0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)
# Predict on dataset which model has not seen before
print(knn.predict(X_test))

# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
```

```
# Loading data
irisData = load_iris()

# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size = 0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Calculate the accuracy of the model
print(knn.score(X_test, y_test))

# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
import numpy as np
import matplotlib.pyplot as plt

irisData = load_iris()
# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size = 0.2, random_state=42)

neighbors = np.arange(1, 9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))
```

```
# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    # Compute training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)

# Generate plot
plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')
plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

plt.legend()
plt.xlabel('n_neighbors')
plt.ylabel('Accuracy')
plt.show()
```

4. Result/Output/Writing Summary:



Machine Learning Lab - 6 - Colab

https://colab.research.google.com/drive/1cgoUpgewVjo9-K5SlANsF0J0gXwiYhqf?authuser=1#scrollTo=82GOb-63atHr

Machine Learning Lab - 6

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Editing

+ Code + Text

```
[2] # Split into training and test set
    X_train, X_test, y_train, y_test = train_test_split(
                X, y, test_size = 0.2, random_state=42)

    knn = KNeighborsClassifier(n_neighbors=7)

    knn.fit(X_train, y_train)

    # Calculate the accuracy of the model
    print(knn.score(X_test, y_test))
```

```
0.9666666666666667
```

```
# Import necessary modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
import numpy as np
import matplotlib.pyplot as plt

irisData = load_iris()

# Create feature and target arrays
X = irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size = 0.2, random_state=42)

neighbors = np.arange(1, 9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)

    # Compute training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)
```

0s completed at 8:10 AM

19°C Sunny ENG IN 08:33 15-10-2022

Machine Learning Lab - 6 - Colab

https://colab.research.google.com/drive/1cgoUpgewVjo9-K5SlANsF0J0gXwiYhqf?authuser=1#scrollTo=K9d1CIggyF2

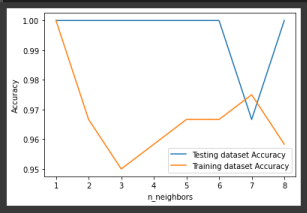
Machine Learning Lab - 6

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Editing

+ Code + Text

```
# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)

    # Compute training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)

# Generate plot
plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')
plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

plt.legend()
plt.xlabel('n_neighbors')
plt.ylabel('Accuracy')
plt.show()
```




```
[ ]
```

0s completed at 8:10 AM

19°C Sunny ENG IN 08:33 15-10-2022

Learning outcomes (What I have learnt):

1. Understood the concept of KNeighborsClassifier
2. Learnt how to load the iris dataset, and splitting the dataset.
3. Predicting the test data on KNN.
4. Plot the legend graph of Accuracy of the Dataset.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			