**“Experiment – 6”**

Name: **SUMIT KUMAR** UID: **20BCS8226**

Branch: **BE-CSE** Section/Group: **MM 808 A**

Semester: **5th** Date of Submission: **23/10/22**

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')

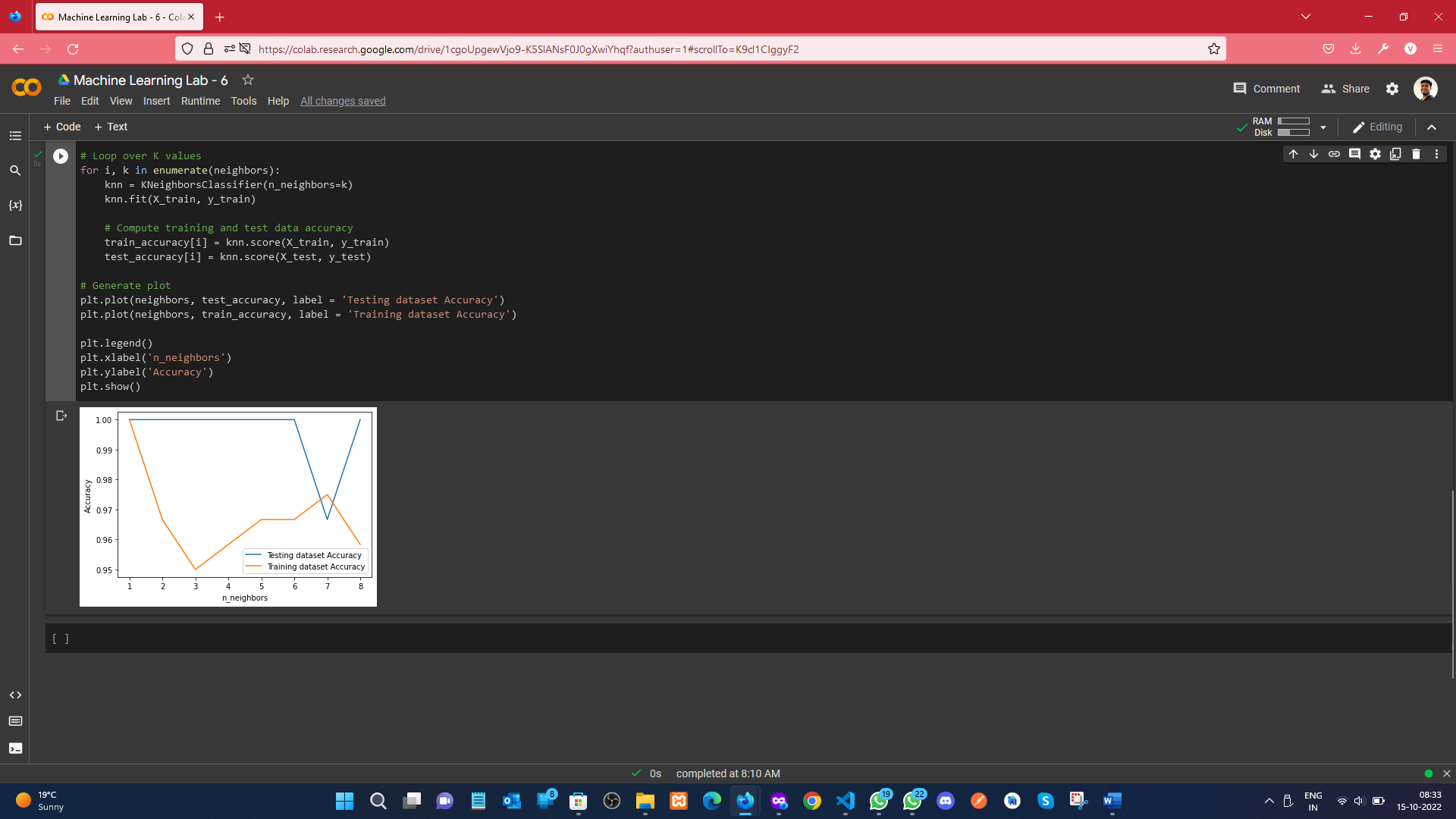
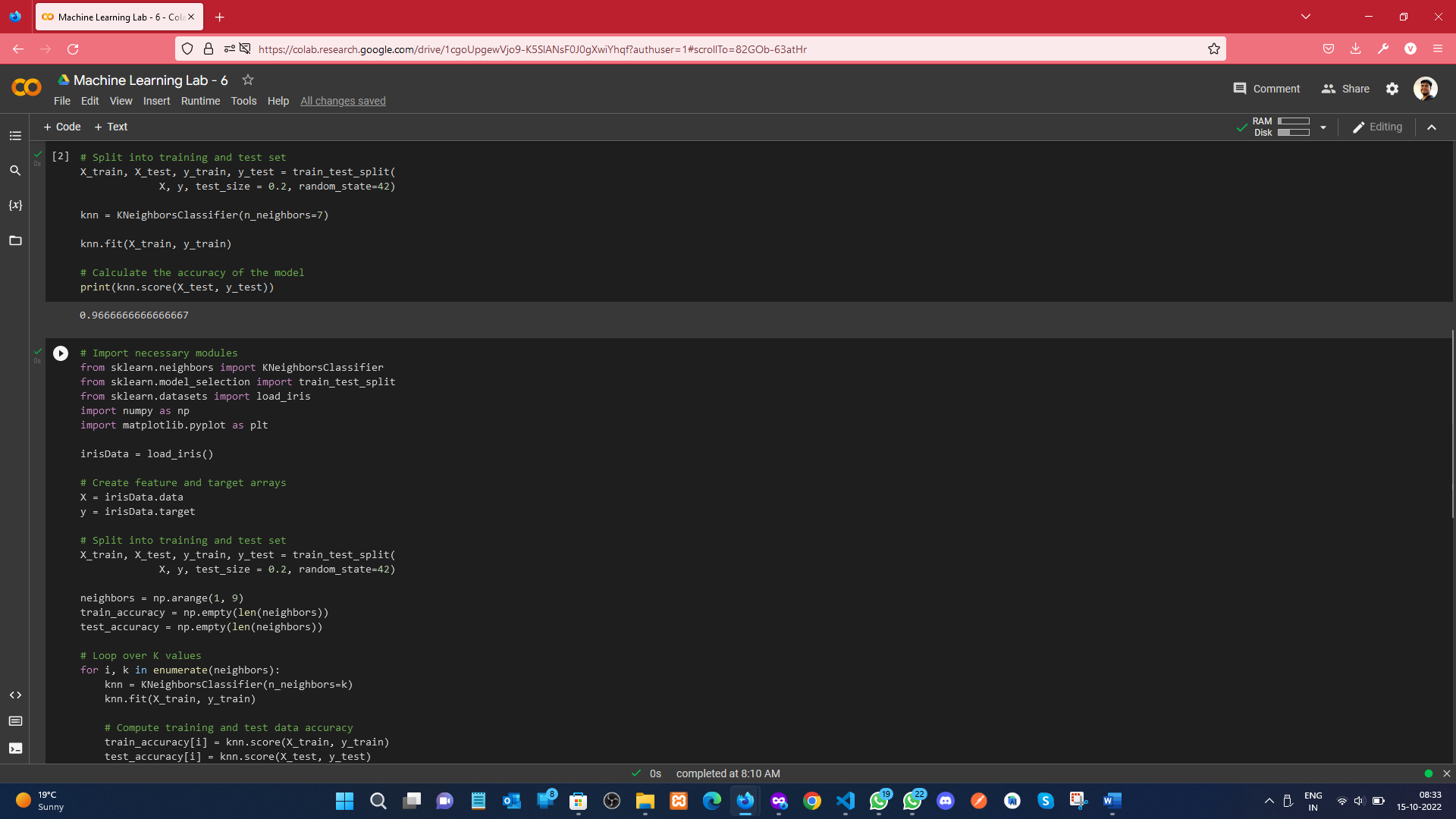
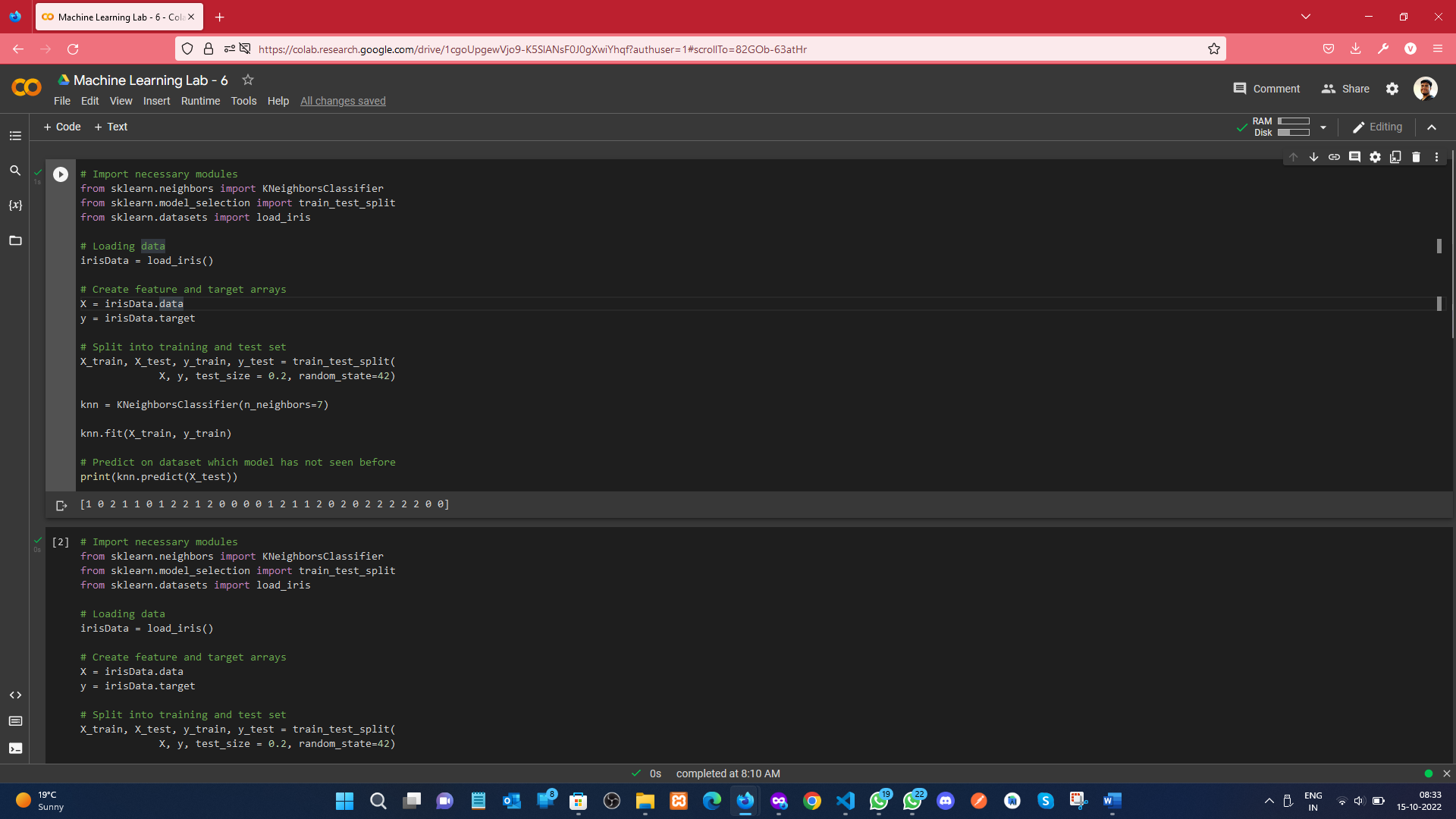
plt.legend()

plt.xlabel('n\_neighbors')

plt.ylabel('Accuracy')

plt.show()

**4. Result/Output/Writing Summary:**



**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. |  |  |  |
|  |  |  |  |