**EXPERIMENT 7**

Q1. LOGISTIC REGRESSION IMPLEMENTATION ON IRIS DATASET

ANS.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

a = pd.read\_csv("/content/Iris (1).csv")

print(a)

feature\_columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm','PetalWidthCm']

X = a[feature\_columns].values

Y = a['Species'].values

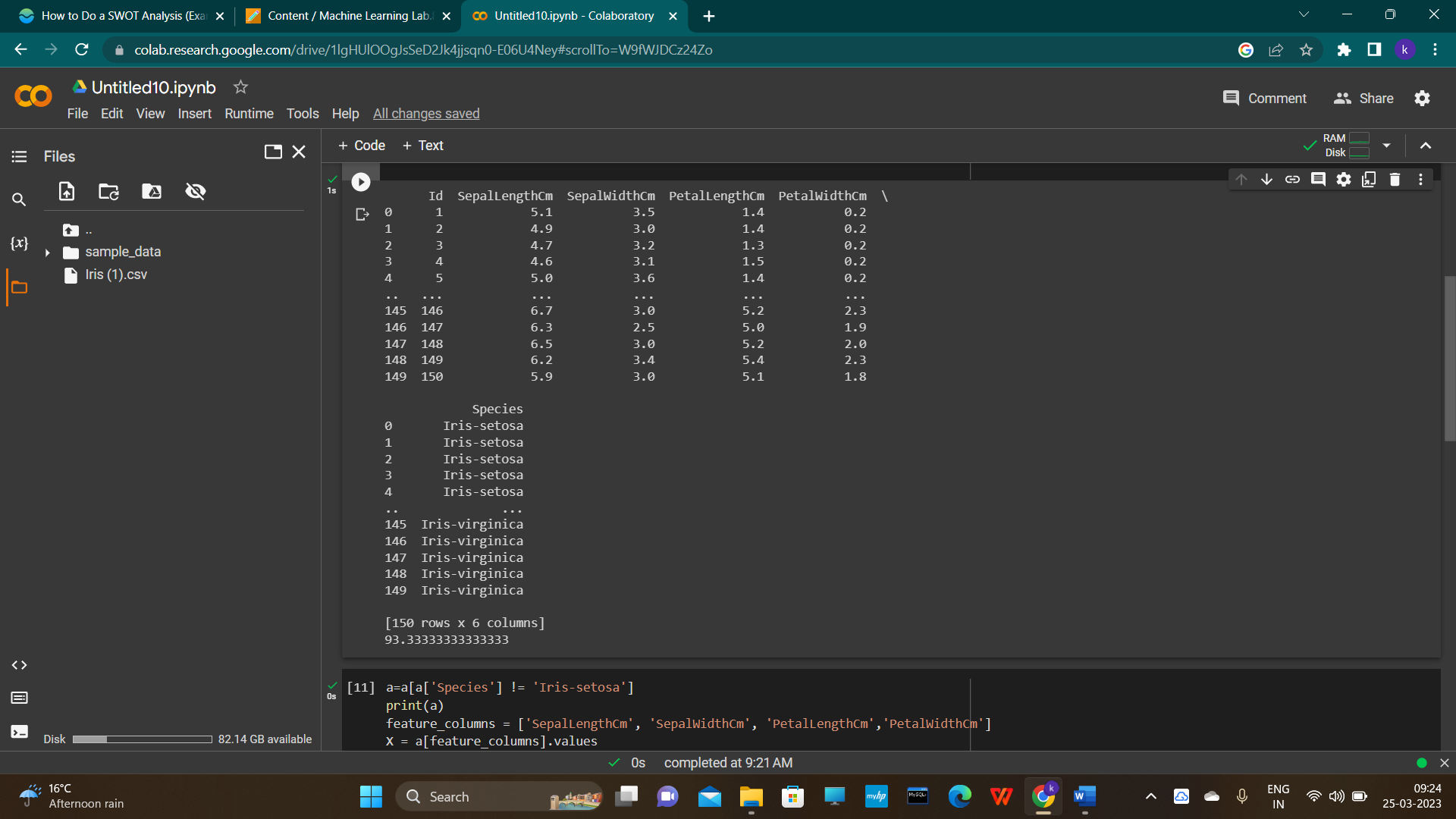
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3)

lr = LogisticRegression(C=100.0, multi\_class='ovr')

lr.fit(X\_train, Y\_train)

Y\_predict = lr.predict(X\_test)

print(metrics.accuracy\_score(Y\_test, Y\_predict)\*100)



AFTER REMOVING ONE CLASS ( SETOSA ) IN CONTINUATION WITH THE UPPER CODE -

a=a[a['Species'] != 'Iris-setosa']

print(a)

feature\_columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm','PetalWidthCm']

X = a[feature\_columns].values

Y = a['Species'].values

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3)

lr = LogisticRegression(C=100.0, multi\_class='ovr')

lr.fit(X\_train, Y\_train)

Y\_predict = lr.predict(X\_test)

print(metrics.accuracy\_score(Y\_test, Y\_predict)\*100)

