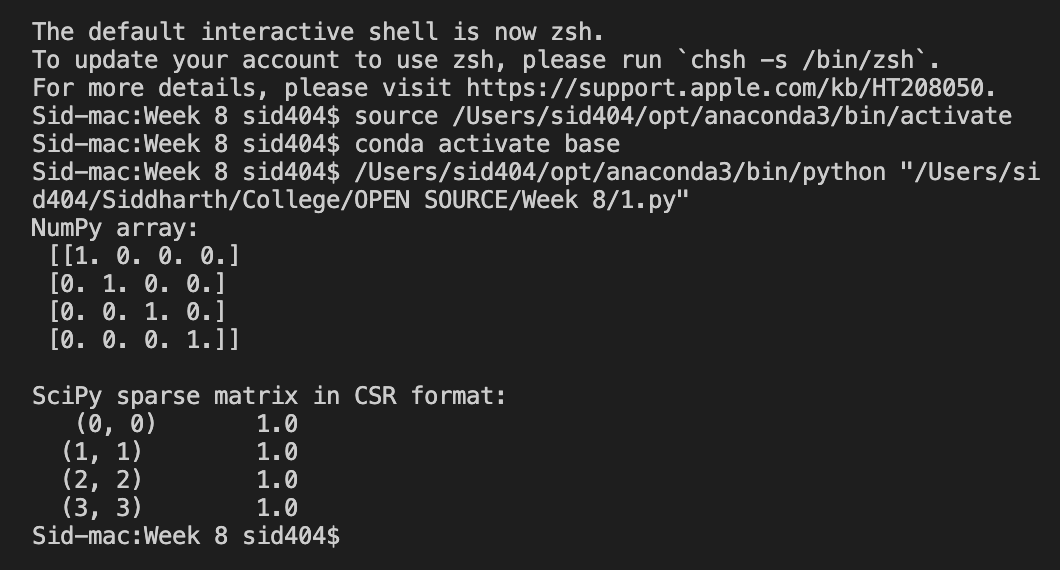
**Assignment – 8**

Name: Siddharth Goyal

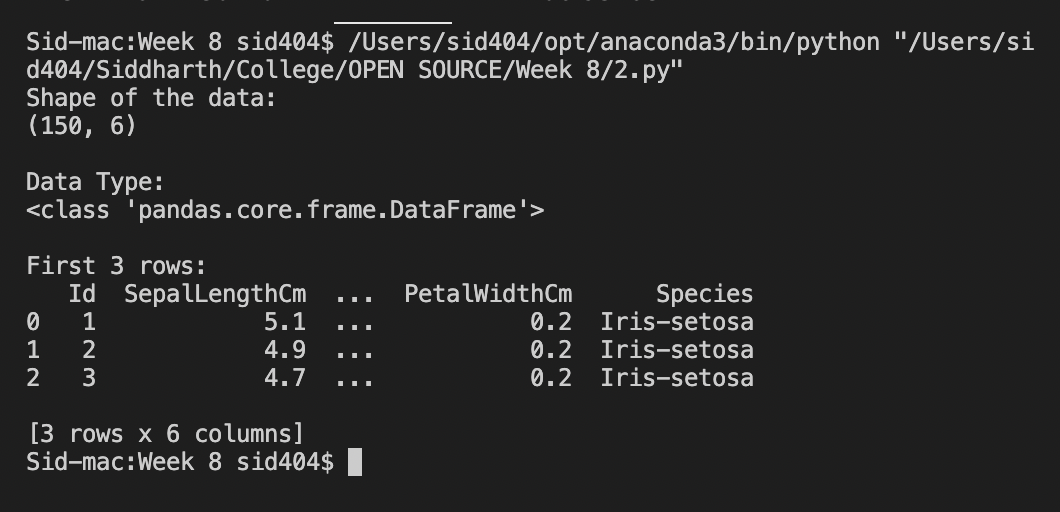
Enroll: 9918103232

Batch: F8



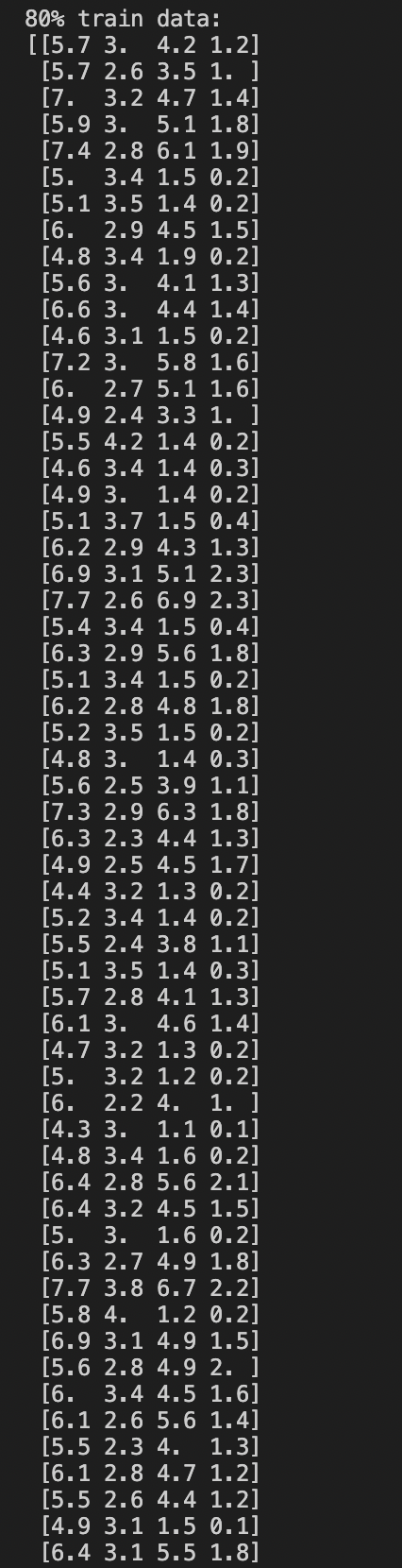
|  |
| --- |
| import numpy as np  from scipy import sparse  eye = np.eye(4)  print("NumPy array:\n", eye)  sparse\_matrix = sparse.csr\_matrix(eye)  print("\nSciPy sparse CSR matrix:\n", sparse\_matrix) |

2)



|  |
| --- |
| import pandas as pd  data = pd.read\_csv("iris.csv")  print("Shape of the data:")  print(data.shape)  print("\nData Type:")  print(type(data))  print("\nFirst 3 rows:")  print(data.head(3)) |

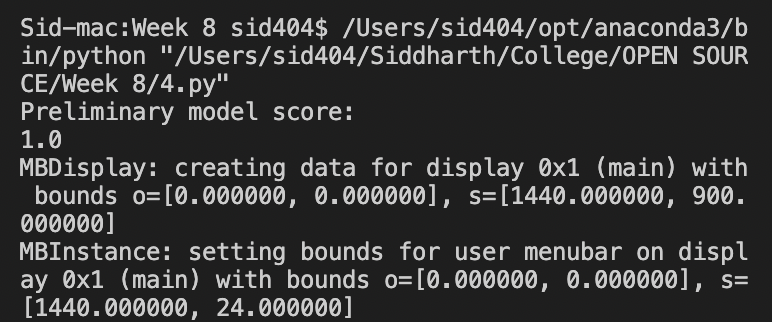
3)

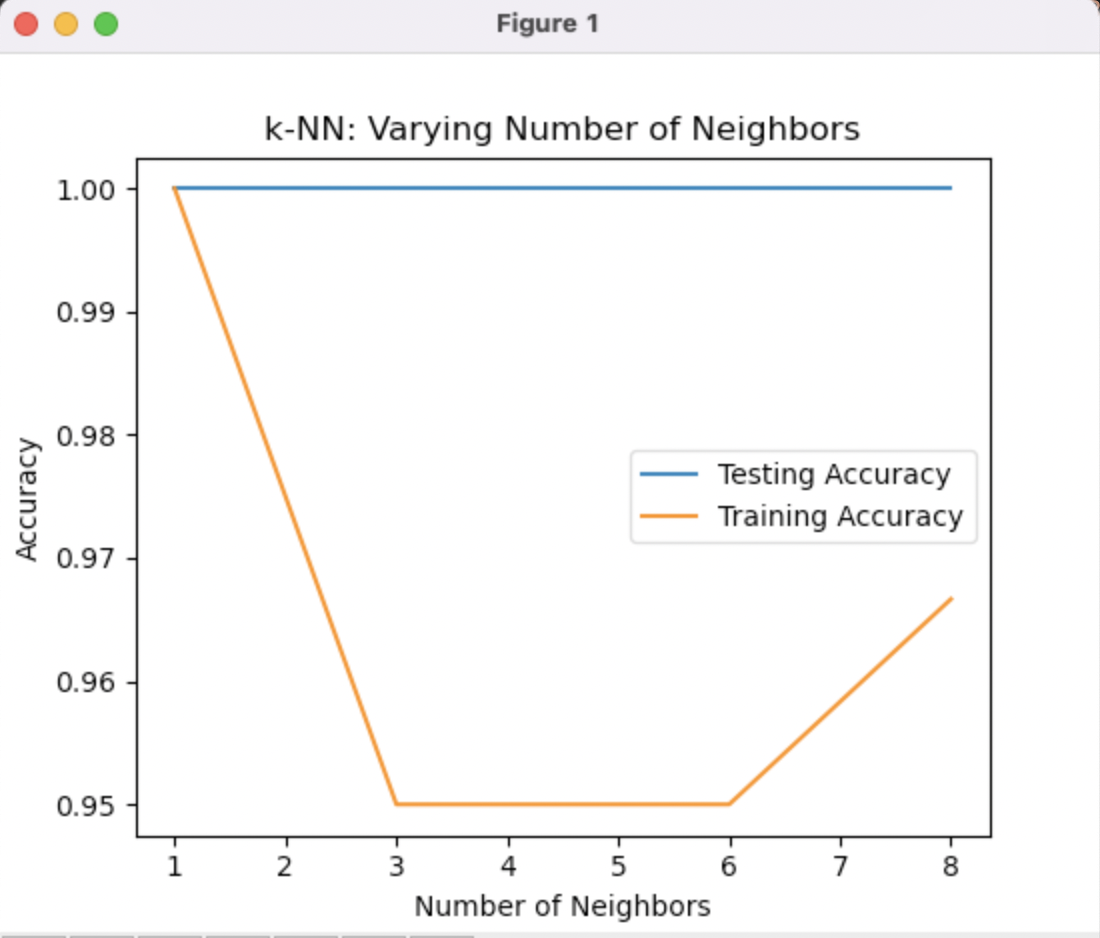




|  |
| --- |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  iris = pd.read\_csv("iris.csv")  #Drop id column  iris = iris.drop(columns='(150, 4, setosa, versicolor, virginica)',axis=1)  X = iris.iloc[:, :-1].values  y = iris.iloc[:, 4].values  #Split arrays or matrices into random train and test subsets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)  print("\n70% train data:")  print(X\_train)  print(y\_train)  print("\n30% test data:")  print(X\_test)  print(y\_test) |

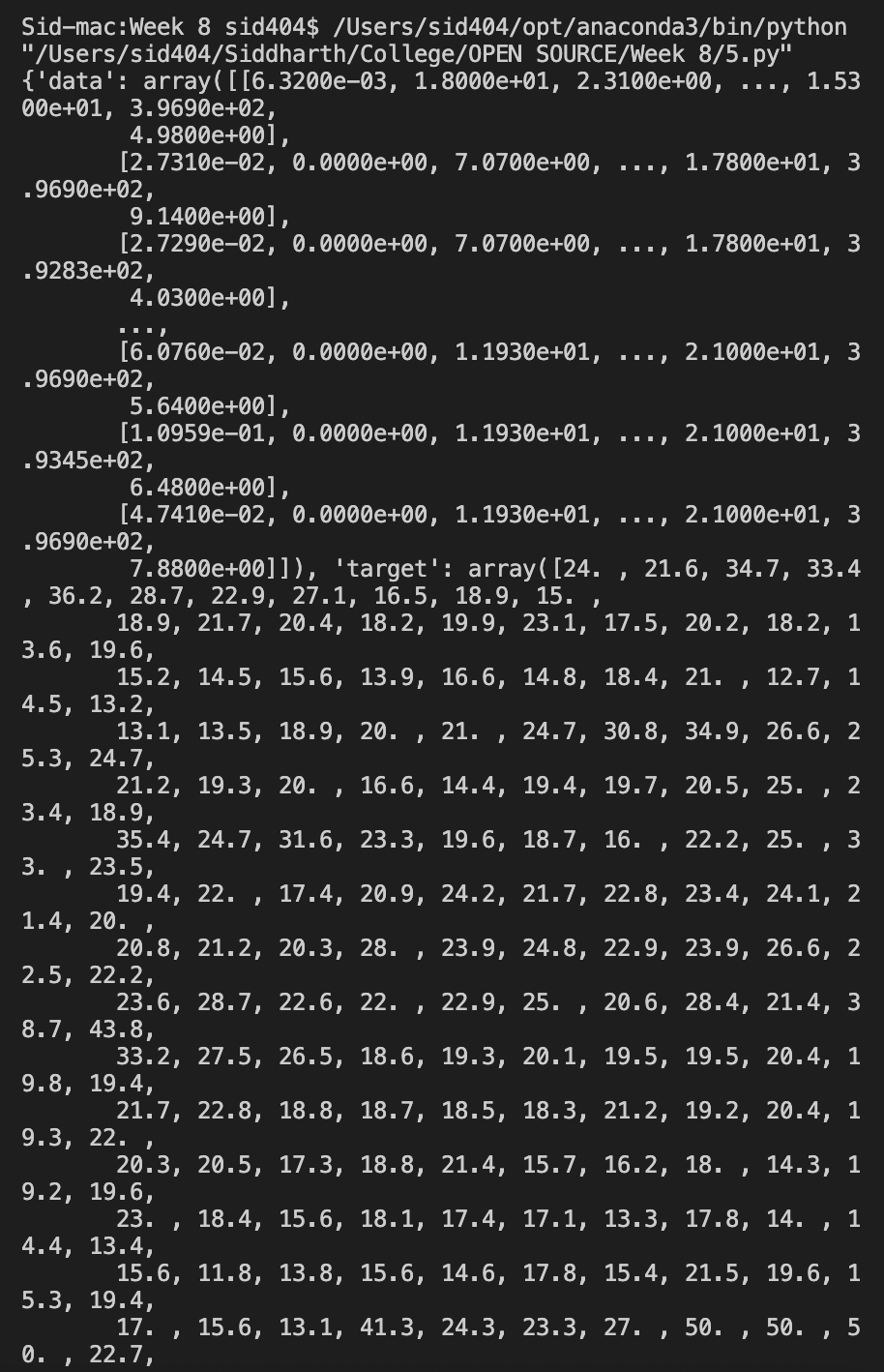
4)

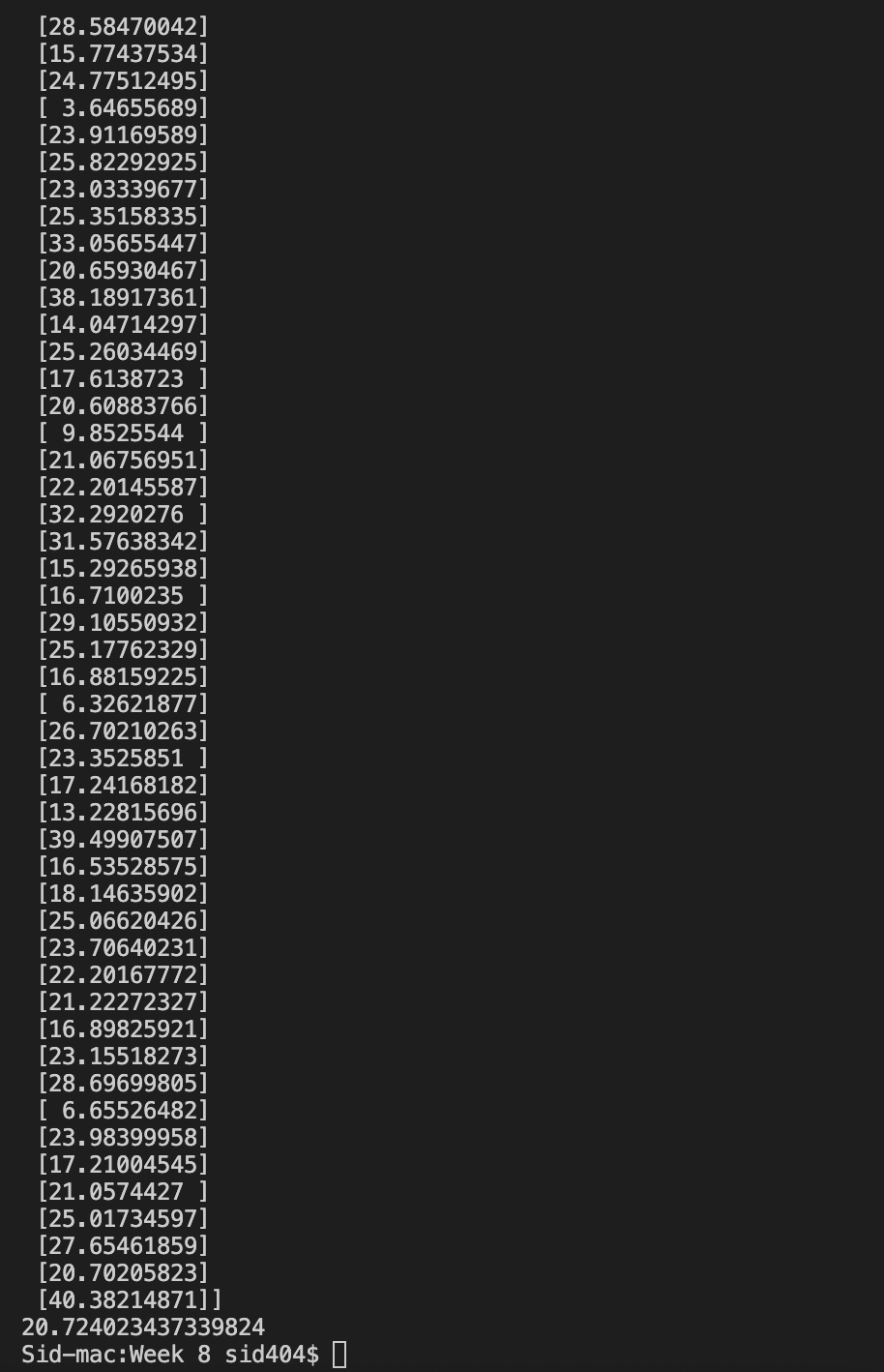




|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  import numpy as np  from sklearn.neighbors import KNeighborsClassifier  from sklearn.model\_selection import train\_test\_split  iris = pd.read\_csv("iris.csv")  iris = iris.drop('Id',axis=1)  X = iris.iloc[:, :-1].values  y = iris.iloc[:, 4].values  #Split arrays or matrices into train and test subsets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)  knn = KNeighborsClassifier(n\_neighbors=7)  knn = KNeighborsClassifier(n\_neighbors=6)  knn.fit(X\_train,y\_train)  print("Preliminary model score:")  print(knn.score(X\_test,y\_test))  no\_neighbors = np.arange(1, 9)  train\_accuracy = np.empty(len(no\_neighbors))  test\_accuracy = np.empty(len(no\_neighbors))  for i, k in enumerate(no\_neighbors):  knn = KNeighborsClassifier(n\_neighbors=k)  knn.fit(X\_train,y\_train)    train\_accuracy[i] = knn.score(X\_train, y\_train)  test\_accuracy[i] = knn.score(X\_test, y\_test)  plt.title('k-NN: Varying Number of Neighbors')  plt.plot(no\_neighbors, test\_accuracy, label = 'Testing Accuracy')  plt.plot(no\_neighbors, train\_accuracy, label = 'Training Accuracy')  plt.legend()  plt.xlabel('Number of Neighbors')  plt.ylabel('Accuracy')  plt.show() |

5)





|  |
| --- |
| import pandas as pd  import numpy as np  from sklearn import linear\_model  from sklearn.model\_selection import train\_test\_split  from sklearn.datasets import load\_boston  boston = load\_boston()  print(boston)  df\_x = pd.DataFrame(boston.data, columns = boston.feature\_names)  df\_y = pd.DataFrame(boston.target)  df\_x.describe()  reg = linear\_model.LinearRegression()  x\_train, x\_test, y\_train, y\_test = train\_test\_split(df\_x, df\_y, test\_size=0.33, random\_state=42)  reg.fit(x\_train, y\_train)  print(reg.coef\_)  y\_pred = reg.predict(x\_test)  print(y\_pred)  y\_pred[2]  y\_test[0]  from sklearn.metrics import mean\_squared\_error  print(mean\_squared\_error(y\_test, y\_pred)) |