Cse23254 lab 5

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score, calinski\_harabasz\_score, davies\_bouldin\_score

data = pd.read\_csv("features\_raw (1).csv")

X = data.iloc[:, :-1].values

y = data.iloc[:, -1].values

# A1

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X[:, [0]], y, test\_size=0.3)

reg1 = LinearRegression().fit(X\_train, y\_train)

y\_train\_pred1 = reg1.predict(X\_train)

y\_test\_pred1 = reg1.predict(X\_test)

# A2

mse\_train1 = mean\_squared\_error(y\_train, y\_train\_pred1)

rmse\_train1 = np.sqrt(mse\_train1)

mape\_train1 = np.mean(np.abs((y\_train - y\_train\_pred1) / y\_train)) \* 100

r2\_train1 = r2\_score(y\_train, y\_train\_pred1)

mse\_test1 = mean\_squared\_error(y\_test, y\_test\_pred1)

rmse\_test1 = np.sqrt(mse\_test1)

mape\_test1 = np.mean(np.abs((y\_test - y\_test\_pred1) / y\_test)) \* 100

r2\_test1 = r2\_score(y\_test, y\_test\_pred1)

# A3

X\_train2, X\_test2, y\_train2, y\_test2 = train\_test\_split(X, y, test\_size=0.3)

reg2 = LinearRegression().fit(X\_train2, y\_train2)

y\_train\_pred2 = reg2.predict(X\_train2)

y\_test\_pred2 = reg2.predict(X\_test2)

mse\_train2 = mean\_squared\_error(y\_train2, y\_train\_pred2)

rmse\_train2 = np.sqrt(mse\_train2)

mape\_train2 = np.mean(np.abs((y\_train2 - y\_train\_pred2) / y\_train2)) \* 100

r2\_train2 = r2\_score(y\_train2, y\_train\_pred2)

mse\_test2 = mean\_squared\_error(y\_test2, y\_test\_pred2)

rmse\_test2 = np.sqrt(mse\_test2)

mape\_test2 = np.mean(np.abs((y\_test2 - y\_test\_pred2) / y\_test2)) \* 100

r2\_test2 = r2\_score(y\_test2, y\_test\_pred2)

# A4

kmeans2 = KMeans(n\_clusters=2, random\_state=0, n\_init="auto").fit(X)

labels2 = kmeans2.labels\_

centers2 = kmeans2.cluster\_centers\_

# A5

sil\_score2 = silhouette\_score(X, labels2)

ch\_score2 = calinski\_harabasz\_score(X, labels2)

db\_score2 = davies\_bouldin\_score(X, labels2)

# A6

sil\_scores = []

ch\_scores = []

db\_scores = []

k\_range = range(2, 11)

for k in k\_range:

km = KMeans(n\_clusters=k, random\_state=0, n\_init="auto").fit(X)

sil\_scores.append(silhouette\_score(X, km.labels\_))

ch\_scores.append(calinski\_harabasz\_score(X, km.labels\_))

db\_scores.append(davies\_bouldin\_score(X, km.labels\_))

plt.plot(k\_range, sil\_scores, label="Silhouette")

plt.plot(k\_range, ch\_scores, label="CH Score")

plt.plot(k\_range, db\_scores, label="DB Index")

plt.legend()

plt.show()

# A7

distortions = []

for k in range(2, 20):

km = KMeans(n\_clusters=k, random\_state=0, n\_init="auto").fit(X)

distortions.append(km.inertia\_)

plt.plot(range(2, 20), distortions)

plt.show()