Lab 9 \_ NguyenMinhCong\_22IT039

import numpy as np

import scipy

from sklearn.cluster import Kmeans

def init\_centers(X, k):

return X[np.random.choice(X.shape[0], k, replace=False)]

def assign\_labels(X, centers):

Distances = scipy.spatial.distance.cdist(X, centers)

labels = np.argmin(Distances, axis=1)

labels.reshape(labels.shape[0], 1)

return labels

def update\_centers(X, labels, K):

centers = np.zeros((K, X.shape[1]))

for k in range(K):

Xk = X[labels == k, :]

centers[k, :] = np.mean(Xk, axis=0)

return centers

def has\_converged(centers, new\_centers):

return set([tuple(a) for a in centers]) == set([tuple(a) for a in new\_c

enters])

def kmeans(X, K):

centers = [init\_centers(X, K)]

labels = []

n\_iter = 0 # Số vòng lặp

while True:

labels.append(assign\_labels(X, centers[-1]))

new\_center = update\_centers(X, labels[-1], K)

if has\_converged(centers[-1], new\_center):

break

centers.append(new\_center)

n\_iter += 1

return centers[-1], labels[-1], n\_iter

means = [[2, 2], [8, 3], [3, 6]]

cov = [[1, 0], [0, 1]]

N = 500

X0 = np.random.multivariate\_normal(means[0], cov, N)

X1 = np.random.multivariate\_normal(means[1], cov, N)

X2 = np.random.multivariate\_normal(means[2], cov, N)

X = np.concatenate((X0, X1, X2), axis=0)

K = 3

original\_label = np.asarray([0] \* N + [1] \* N + [2] \* N).T

center, label, n\_iter = kmeans(X, K)

kmean = KMeans(n\_clusters=3, random\_state=0)

kmean.fit(X)

pred\_label = kmean.predict(X)

print("Center found by our algorithm: ")

print(center)

print()

print("Center found by sklearn: ")

print(kmean.cluster\_centers\_)