63130500157 รุ่งภพ จรัสพัฒน์

เริ่มต้นด้วยการ import

import numpy as np

import matplotlib.pyplot as plt

from sklearn import cluster

import pandas as pd

from scipy.cluster.hierarchy import dendrogram, linkage

from scipy.spatial import distance\_matrix

data = pd.read\_csv("data/data2Dset1.csv",header=None)

data.columns = ["A","B"]

data.head()

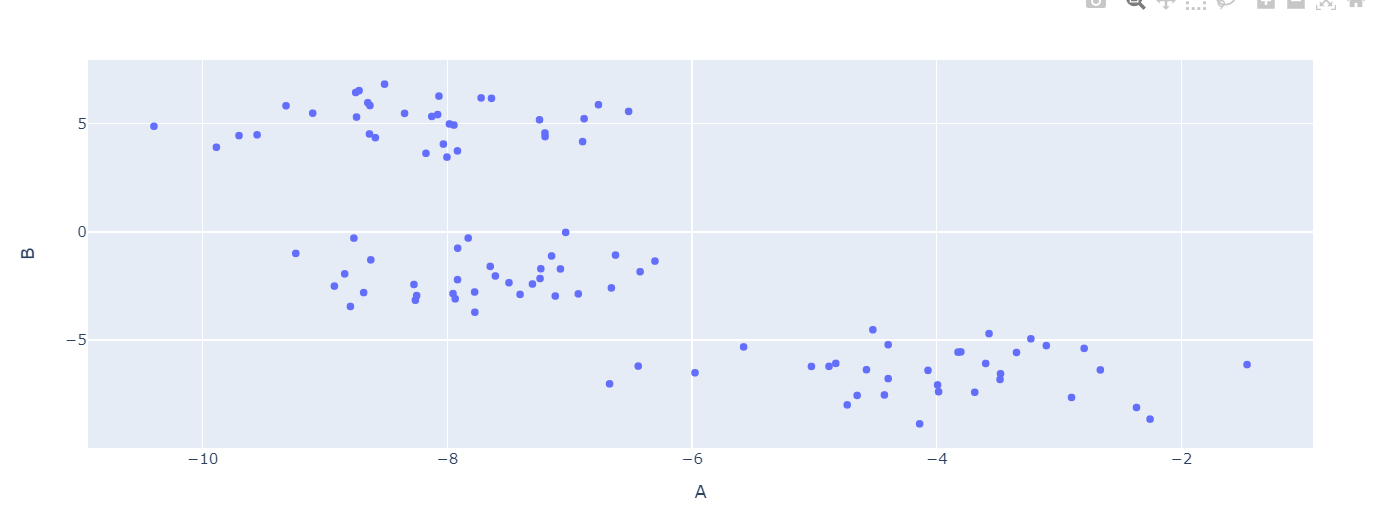


1. Plot จุดข้อมูล data1

import plotly.express as px

fig = px.scatter(data1, x="A", y="B")

fig.show()



1. Data2Dset1 เริ่มด้วยการเขียนโปรแกรม plot จุดข้อมูลโดยใช้วิธี kmeans

K=1

model\_kmeans = cluster.KMeans(n\_clusters=1, max\_iter=50, random\_state=1)

model\_kmeans.fit(data1)

data1['cluster\_id'] = model\_kmeans.labels\_

centroids = model\_kmeans.cluster\_centers\_

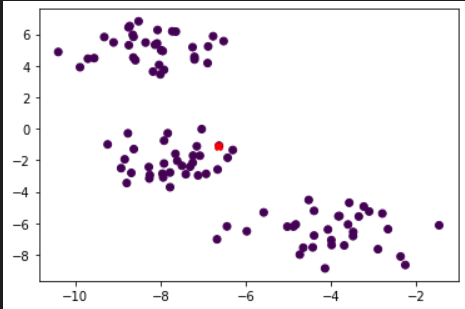
print(centroids)



plt.scatter(data1['A'],data1['B'], c=data1['cluster\_id'])

plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')

plt.show()



ผลลัพธ์ K=1

K=2

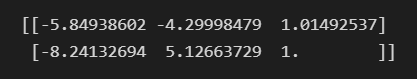
model\_kmeans = cluster.KMeans(n\_clusters=2, max\_iter=50, random\_state=1)

model\_kmeans.fit(data1)

data1['cluster\_id'] = model\_kmeans.labels\_

centroids = model\_kmeans.cluster\_centers

print(centroids)

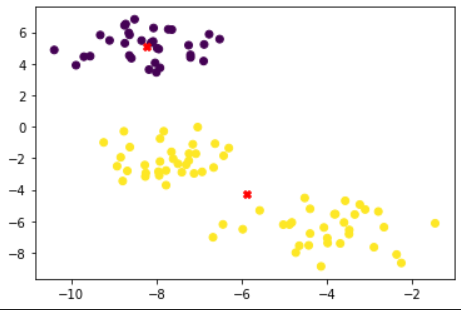
\_

plt.scatter(data1['A'],data1['B'], c=data1['cluster\_id'])

plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')

plt.show()

ผลลัพธ์ K=2



K=3

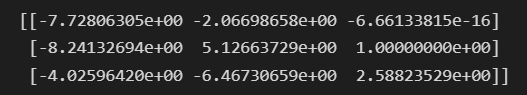
model\_kmeans = cluster.KMeans(n\_clusters=3, max\_iter=50, random\_state=1)

model\_kmeans.fit(data1)

data1['cluster\_id'] = model\_kmeans.labels\_

centroids = model\_kmeans.cluster\_centers\_

print(centroids)

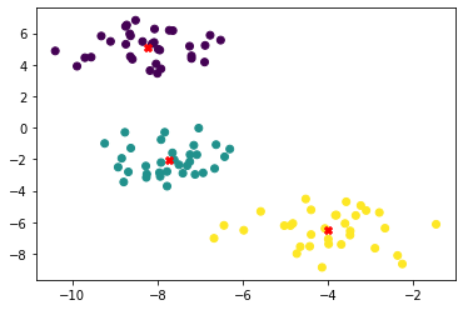


plt.scatter(data1['A'],data1['B'], c=data1['cluster\_id'])

plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')

plt.show()

ผลลัพธ์ K=3



K=4

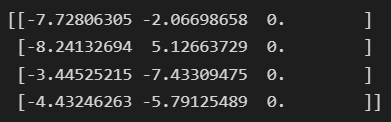
model\_kmeans = cluster.KMeans(n\_clusters=4, max\_iter=50, random\_state=1)

model\_kmeans.fit(data1)

data1['cluster\_id'] = model\_kmeans.labels\_

centroids = model\_kmeans.cluster\_centers\_

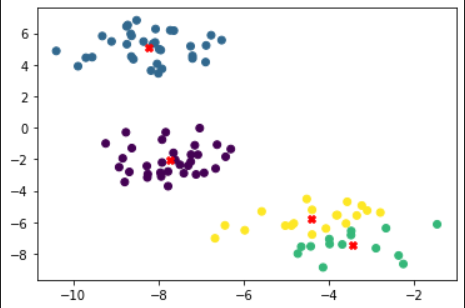
print(centroids)



plt.scatter(data1['A'],data1['B'], c=data1['cluster\_id'])

plt.scatter(centroids[:,0],centroids[:,1],marker='X',c='r')

plt.show()

ผลลัพธ์K=4

จากผลลัพธ์สรุปได้ว่า K=3 ดีที่สุดสำหรับ Data2Dset1 เพราะจะข้อมูลจะเกาะกลุ่มกันเป็นกลุ่มใหญ่ๆอย่างเห็นได้ชัดสามกลุ่ม ถ้าเป็นK4 จะมีตัวข้อมูลที่เกาะกลุ่มกันมากและถ้าเป็นK1,K2 จะเกิดข้อมูลที่ไม่แน่ชัด

1. เขียนโปแกรมจัดกลุ่มชุดข้อมูลที่อ่านเข้ามา โดยใช้วิธี Hierarchical Clustering

4.1) ให้เลือกใช้method ที่ต่างกัน 3 แบบ แสดง dendrogram ที่ได้แต่ละแบบ

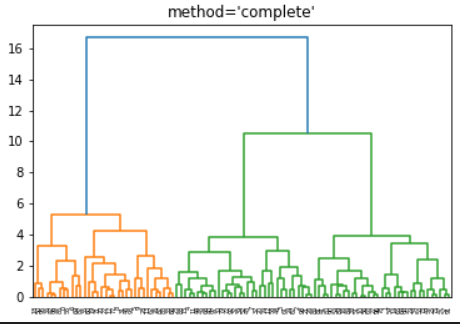
Complete:

linkage\_data = linkage(data1, method='complete' , metric='euclidean')

dendrogram(linkage\_data)

plt.title("method='complete'")

plt.show()



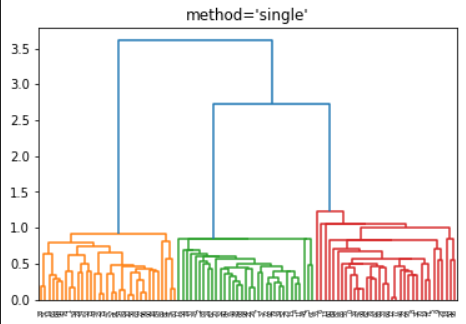
Single:

linkage\_data = linkage(data1, method='single' , metric='euclidean')

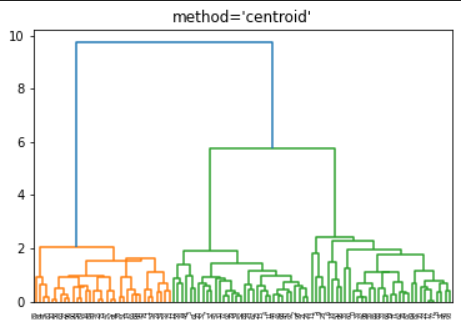
dendrogram(linkage\_data)

plt.title("method='single'")

plt.show()



Centroid:



4.2) เลือก cut-off โดยกำหนด criterion=’distance’ และให้นักศึกษาเลือกระบุค่า t ที่คิดว่า เหมาะสม สำหรับแต่ละ dendrogram ที่ได้ในข้อ 4.1)

Complete: ค่า t ที่เหมาะที่สุดคือ 6

cluster\_id = fcluster(linkage\_data,t=6,criterion='distance')

plt.scatter(data1["A"],data1["B"],c=cluster\_id)

plt.show()

Single: ค่า t ที่เหมาะที่สุดคือ 2

cluster\_id = fcluster(linkage\_data,t=2,criterion='distance')

plt.scatter(data1["A"],data1["B"],c=cluster\_id)

plt.show()

Centroid: ค่า t ที่เหมาะที่สุดคือ 3 ถึง 5

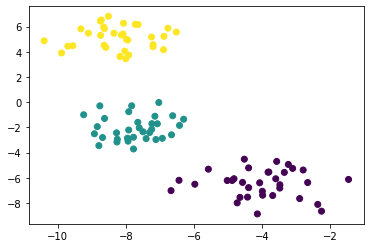
cluster\_id = fcluster(linkage\_data,t=5,criterion='distance')

plt.scatter(data1["A"],data1["B"],c=cluster\_id)

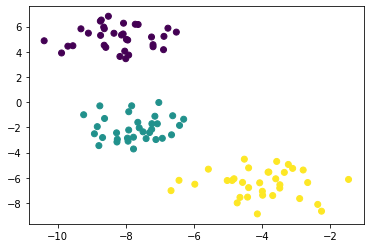
plt.show()

4.3) Plot ผลกการจัดกลุ่ม ที่ได้แต่ละแบบในข้อ 4.2)

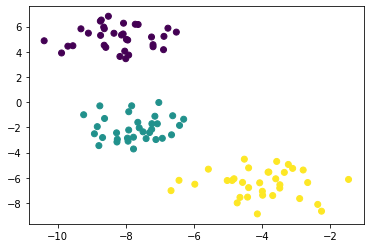
Complete:



Single:



Centroid:



6. เขียนบรรยายสรุปผลการทดลอง แสดงความคิดเห็น วิธีใด เหมาะกับ ชุดข้อมูลแบบไหน แต่ละวิธีมีข้อดี/ ข้อเสีย อย่างไร

* Data2DSet1 จากการทดลองพบว่า วิธี k-Means สามารถทำได้ง่ายกว่า สามารถแบ่งกลุ่มข้อมูลได้ง่ายกว่า และตัวข้อมูลมีดูมีความละเอียดแบ่งเป็นกลุ่มๆอย่างเห็นได้ชัด