Lab1 – Unsupervised machine learning

ให้ทำบน jupyter lab และทำรายงานส่งเป็น pdf

1. นำ Iris data set มาทำการเปรียบขนาดความคล้ายครึง โดยใช้ Hierarchy clustering ของ
   * 1. ความยาวและความกว้างของกลีบเลี้ยง
     2. ความยาวและความกว้างของกลีบดอก

และสรุปผลลัพธ์ที่ได้ออกมา เป็นการจำแนกได้ว่าความใกล้เคียงของดอกอยู่ประมาณไหน ให้ใช้ Rand Index, Entropy(Purity) ในการประเมิณ (2%)

import pandas as pd

Iris = pd.read\_csv('data/Iris.csv')

Table

Description automatically generatedIris

sepal = Iris[["SepalLengthCm","SepalWidthCm"]]

Table

Description automatically generatedsepal

from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

import matplotlib.pyplot as plt

linkage\_sepal = linkage(sepal, method='single', metric='euclidean')

dendrogram(linkage\_sepal, labels=Iris['Species'].to\_list())

plt.title('Single dendrogram sepal')

Chart, histogram

Description automatically generatedplt.show()

sepal\_cls = fcluster(linkage\_sepal, t=0, criterion='distance')

plt.scatter(sepal["SepalLengthCm"],sepal["SepalWidthCm"],c=sepal\_cls)

Chart, scatter chart

Description automatically generatedplt.show()

Chart, scatter chart

Description automatically generatedChart, histogram

Description automatically generatedของ petal ทำคล้ายๆ กัน

การประเมิณผล cluster ของ model

Rand index

from sklearn.metrics import cluster

sepal\_rand\_ind = cluster.rand\_score(Iris['Species'],sepal\_cls)

sepal\_rand\_ind

ผลลัพธ์

Rand sepal = 0.6722147651006711

Rand petal = 0.6798210290827741

Entropy

contingency\_matrix = cluster.contingency\_matrix(Iris['Species'],sepal\_cls)

Background pattern

Description automatically generatedcontingency\_matrix

import numpy as np

mij = contingency\_matrix

mi = contingency\_matrix.sum(axis=0)

pij = mij/mi

log2pij = np.log2(pij,out=np.zeros\_like(pij), where=(pij!=0))

print(pij.round(2))

print(log2pij.round(2))

ei = pij\*log2pij

ei = -1\*ei.sum(axis=0)

print('e\_i \n', ei.round(2))

m = contingency\_matrix.sum()

entropy = ((mi/m)\*ei).sum()

print('entropy =', entropy )

Entropy sepal = 0.1567318333621796

Entropy petal = 0.01836591668108979

Purity

purity\_sepal = np.sum(np.amax(contingency\_matrix, axis=0)) / np.sum(contingency\_matrix)

purity\_sepal

Purity sepal = 0.9266666666666666

Purity sepal = 0.9933333333333333

1. จงใช้ SOM ในการหา BMU ของความใกล้เคียงกันของข้อมูลในไฟล์ Healthcare-dataset-stroke-data.csv ระหว่าง Age กับ Average of glucose level และ Residence\_type เป็นตัวจำแนก และใช้ K-mean ในการจำแนกกราฟโดยใช้ neuron ใน SOM ดูความคล้ายคลึง ใช้ Silhouette Coefficient ในการประเมิณ (2%)

import pandas as pd

df = pd.read\_csv('data/Healthcare-dataset-stroke-data.csv')

Table

Description automatically generateddf

from sklearn.preprocessing import StandardScaler

X = df[["age","avg\_glucose\_level"]]

Y = df[["Residence\_type"]]

scaler = StandardScaler()

scaler= scaler.fit(X.values)

X\_scale = scaler.transform(X.values)

print(X\_scale.shape)

print(X.shape)

Text

Description automatically generated with medium confidence

import simpsom as sps

# train

net = sps.SOMNet(10, 10, X\_scale, PBC=True)

net.train()

Text

Description automatically generated

# model

cls = net.cluster(X\_scale, clus\_type='KMeans') #'MeanShift') #

# plot graph

net.nodes\_graph(colnum=1)

Background pattern

Description automatically generatednet.diff\_graph()

A picture containing text, outdoor object, honeycomb

Description automatically generated

# predict

import numpy as np

cls\_id = np.zeros([len(X\_scale)])

Background pattern

Description automatically generatedcls\_id

import matplotlib.pyplot as plt

for i in range(len(cls)):

cls\_id[cls[i]] = i

plt.scatter(X.values[cls[i], 0],

X.values[cls[i], 1], label='cluster='+str(i), alpha=.7)

plt.title("Clusters of Health care")

plt.xlabel("age")

plt.ylabel("glucose")

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5));

Chart, scatter chart

Description automatically generatedplt.show()

Text

Description automatically generated

Evaluation

Silhouette Coefficient

from sklearn.metrics import silhouette\_score

sil\_coeff = silhouette\_score(X, cls\_id, metric='euclidean')

sil\_coeff = 0.18583855137939662

1. จงใช้ K-Mean จำแนกกลุ่ม 3 กลุ่ม ของ Salary\_Data.csv และใช้ Sum square error ในการประเมิณ (2%)

import pandas as pd

df = pd.read\_csv('data/Salary\_Data.csv')

df

Table

Description automatically generated

from sklearn.cluster import KMeans

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

model.fit(df)

cls\_id = model.labels\_

cls\_id



centroid = model.cluster\_centers\_

Text

Description automatically generated with medium confidencecentroid

import matplotlib.pyplot as plt

plt.scatter(df["YearsExperience"],df["Salary"], c=cls\_id)

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

plt.title("K-mean of Year experience and salary")

plt.xlabel("Year experience")

plt.ylabel("Salary")

Chart, scatter chart

Description automatically generatedplt.show()

Evaluation

Sum square error

sse = model.inertia\_

sse = 2056842710.938375