BÀI 8: SỰ PHÂN TÍCH NHÓM (TT)

I. Mục tiêu:

Sau khi thực hành xong, sinh viên nắm được:

- Gom cụm dựa vào Mô hình hỗn hợp Gaussian.

- Gom cụm dựa vào thuật toán Mahalanobis k-means.

II. Tóm tắt lý thuyết:

1. Gom cụm dựa vào Mô hình hỗn hợp Gaussian

Mô hình hỗn hợp Gaussian (Gaussian Mixture Model hay GMM) là một mô hình xác suất giả sử rằng tất cả các điểm dữ liệu được tạo ra từ hỗn hợp của một số hữu hạn các phân phối Gaussian với các tham số chưa biết. Người ta có thể coi các mô hình hỗn hợp này là tổng quát hóa phân cụm k-means để kết hợp thông tin về cấu trúc hiệp phương sai của dữ liệu cũng như các tâm của Gaussian tiềm ẩn.
Thuật toán GMM là một phép gán (assigment) mềm dựa vào các xác suất trước đó.
Thuật toán này dựa vào khoảng cách Mahalanobis.
Phân phối GM có thể được viết như một siêu vị trí tuyến tính của K Gaussian.

$$p(x) = \sum_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)p(k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

với π_k (hay $p(k)$) là các hệ số mixing được biết như là xác suất của lớp k, $\sum_{k=1}^{K} \pi_k = 1$ và

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_k|}} exp\Big(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\Big).$$

Mục tiêu là để cực đại hóa log-likelihood của GMM

$$\begin{aligned}\log \prod_{i=1}^{N} p(x_i) &= \log \prod_{i=1}^{N} \Big\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\Big\} \\ &= \sum_{i=1}^{N} \log \Big\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\Big\}\end{aligned}$$

Thuật toán Expectation-Maximization (EM)

Khởi tạo các trung bình μ_k, ma trận hiệp phương sai Σ_k và các hệ số mixing $p(k)$.

a. Bước E: Với mỗi mẫu i, ước lượng cho mỗi lớp k sử dụng các giá trị tham số hiện tại

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_k \mathcal{N}(x_i|\mu_j, \Sigma_j)}$$

b. Bước M: Gán các điểm dữ liệu tới các cụm. Với mỗi lớp, ước lượng lại các tham số μ_k, Σ_k, π_k.

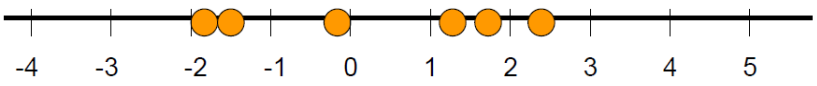
$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N} \text{ với } N_k = \sum_{i=1}^{N} \gamma_{ik}.$$

Lặp lại cho tới khi hội tụ.

Ví dụ: Cho $x = (x_1, x_2, \ldots, x_N)$ như hình



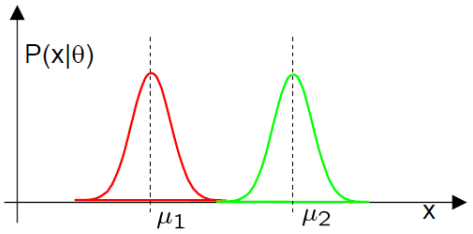
Mục tiêu là chúng ta sẽ làm phù hợp với mô hình Gaussian hỗn hợp với $K = 2$ thành phần.

Mô hình:

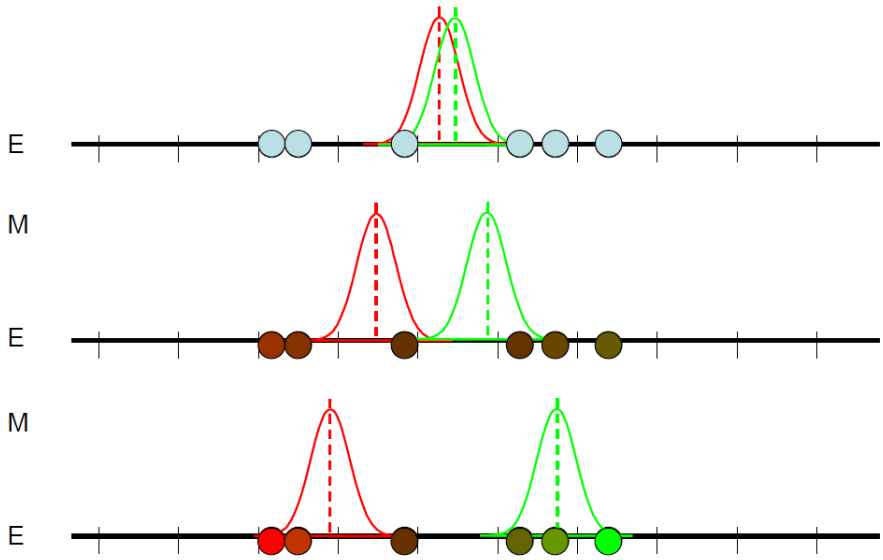
$$p(x_i|\theta) = \sum_{k} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

với $\sum_{k=1}^{K} \pi_k = 1$.

Các tham số $\theta = \{\pi, \mu, \sigma\}$. Ta cố định π, σ và chỉ ước lượng μ.



Các bước thực hiện thuật toán EM được minh họa như hình sau:



2. Gom cụm dựa vào thuật toán Mahalanobis k-means

Xét tập dữ liệu với k cụm. Giả sử cụm thứ r trong không gian d chiều có vector trung bình $\overline{\mu_r}$ có d chiều tương ứng và ma trận hiệp phương sai Σ_r có $d \times d$ chiều. Phần tử thứ (i, j)của ma trận này là hiệp phương sai giữa các chiều i và j trong cụm này. Khi đó, khoảng cách Mahalanobis $Maha(\overline{X}, \overline{\mu_r}, \Sigma_r)$ giữa điểm dữ liệu $\overline{X}$ và tâm cụm $\overline{\mu_r}$ được xác định như sau:

$$Maha(\overline{X}, \overline{\mu_r}, \Sigma_r) = \sqrt{(\overline{X} - \overline{\mu_r})\Sigma_r^{-1}(\overline{X} - \overline{\mu_r})^T}$$

Khoảng cách này được phát biểu như tỷ số ngoại lệ. Các giá trị lớn hơn của tỷ số ngoại lệ cho biết một khuynh hướng ngoại lệ lớn hơn. Sau khi tỷ số ngoại lệ đã được xác định, phân tích giá trị cực trị có thể được dùng để chuyển các tỷ số thành các nhãn nhị phân. Thuật toán được tóm tắt thông qua các bước sau:

1. Pick K points at random from the data set to be the initial clusters.

2. Calculate the Euclidean distance from each point in the data set to each cluster center.

3. Form the initial clusters by assigning each point to the cluster center whose distance is the least of the k distances.

4. Next, calculate the Mahalanobis distance from each cluster center to each of the N data points and assign each point to the nearest cluster center.

5. Recalculate $\widehat{\boldsymbol{\mu}}_k$ and $\widehat{\boldsymbol{\Sigma}}_k$ for $k = 1, \ldots, K$ and repeat step 5 until the clusters do not change.

III. Nội dung thực hành:

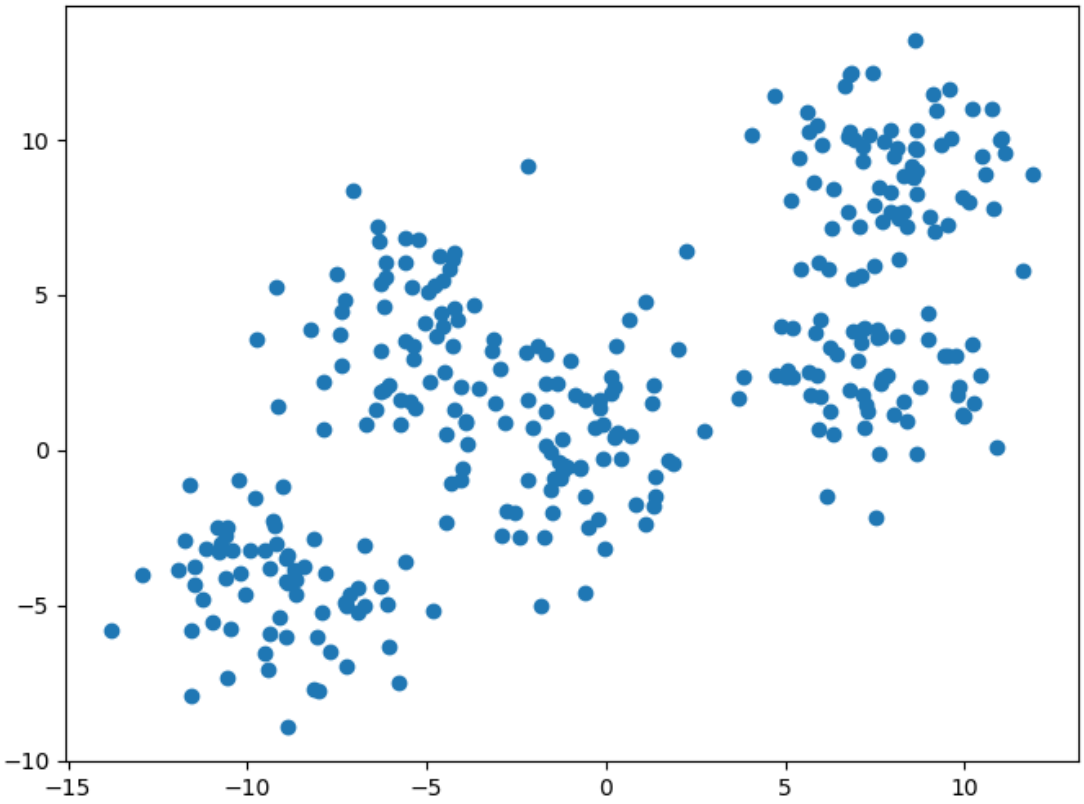
1. Gom cụm dựa vào Mô hình hỗn hợp Gaussian:

- Chuẩn bị dữ liệu

- Xác định GMM và điều chỉnh nó trên dữ liệu x. Chia dữ liệu thành 5 cụm, ta thu được các tâm của cụm như sau

```
from sklearn.mixture import GaussianMixture
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
from numpy import random
from pandas import DataFrame
from sklearn.cluster import KMeans

random.seed(234)
x, _ = make_blobs(n_samples=330, centers=5, cluster_std=1.84)
plt.figure(figsize=(8, 6))
plt.scatter(x[:,0], x[:,1])
plt.show()
```

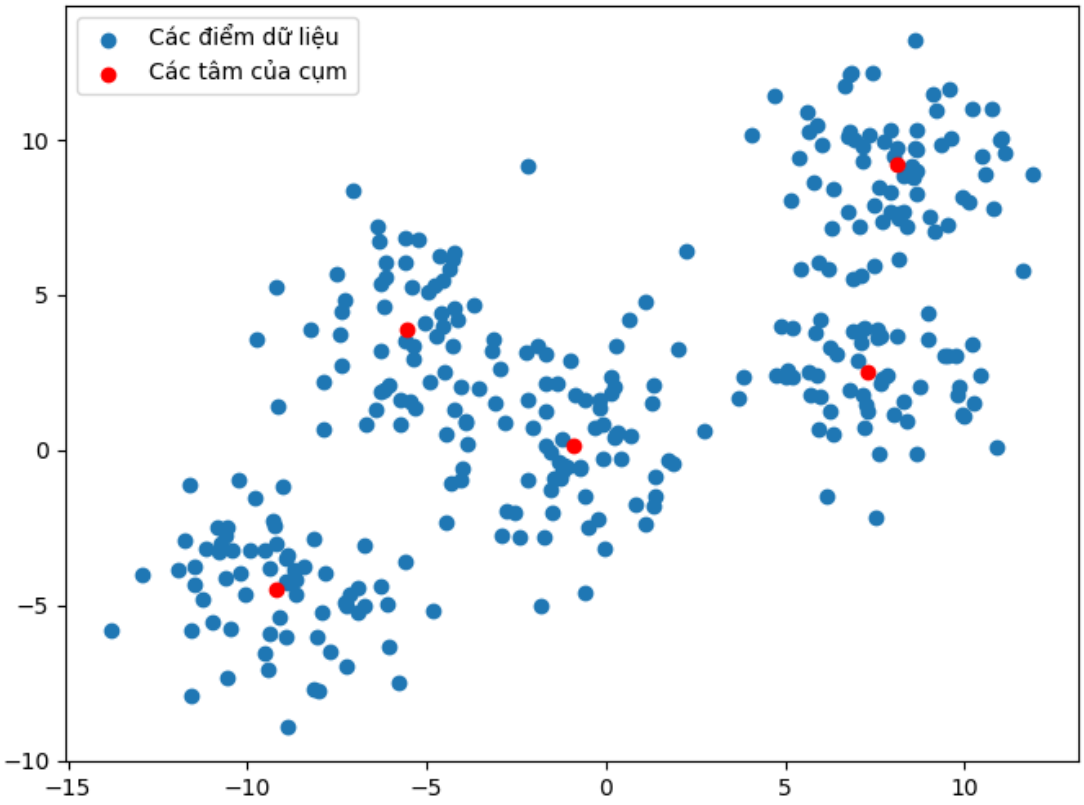



```
gm = GaussianMixture(n_components=5).fit(x)
centers = gm.means_
print(centers)
```

```
[[-5.55839384  3.87288254]
 [ 7.28633633  2.54355009]
 [-9.1828898  -4.4785241 ]
 [ 8.10880025  9.23182354]
 [-0.90236602  0.16316915]]
```

- Vẽ các tâm này và các điểm dữ liệu

```
plt.figure(figsize=(8, 6))
plt.scatter(x[:,0], x[:,1], label="Các điểm dữ liệu")
plt.scatter(centers[:,0], centers[:,1],c='r', label="Các tâm của cụm")
plt.legend()
plt.show()
```



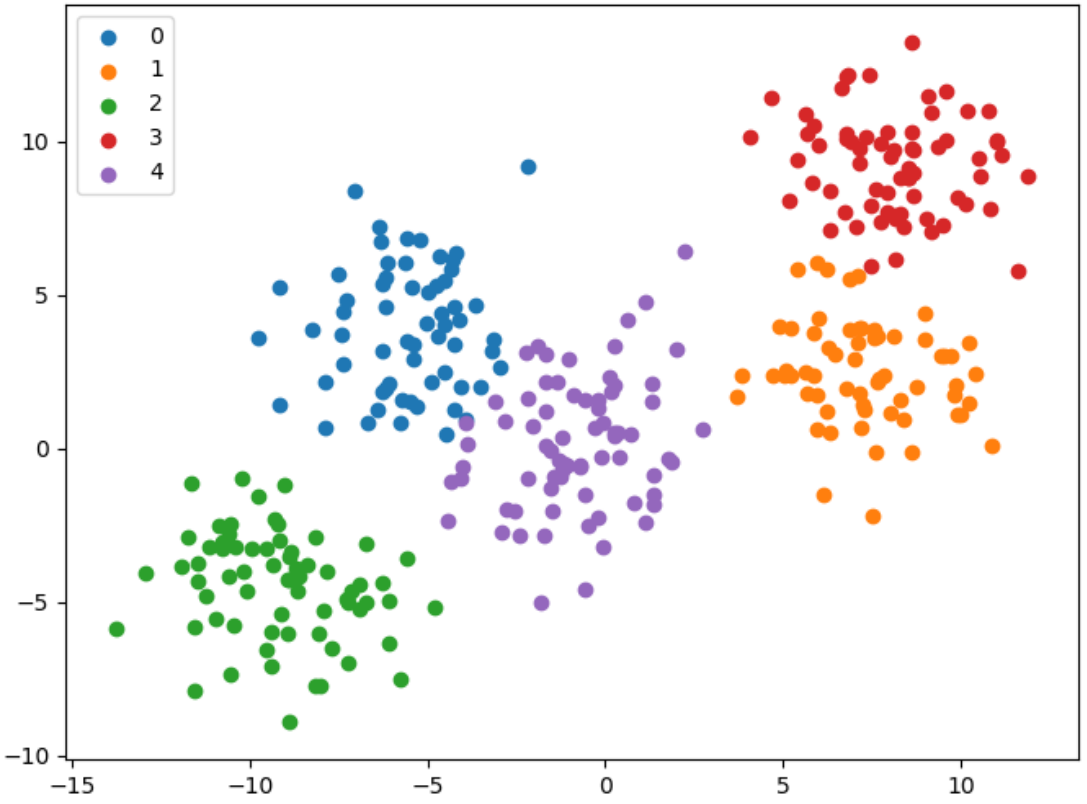
- Dự đoán dữ liệu x với mô hình được huấn luyện, gom nhóm các phần tử và trực quan hóa các cụm trong biểu đồ .

```
pred = gm.predict(x)

df = DataFrame({'x':x[:,0], 'y':x[:,1], 'label':pred})
groups = df.groupby('label')

ig, ax = plt.subplots(figsize=(8, 6))
for name, group in groups:
    ax.scatter(group.x, group.y, label=name)

ax.legend()
plt.show()
```



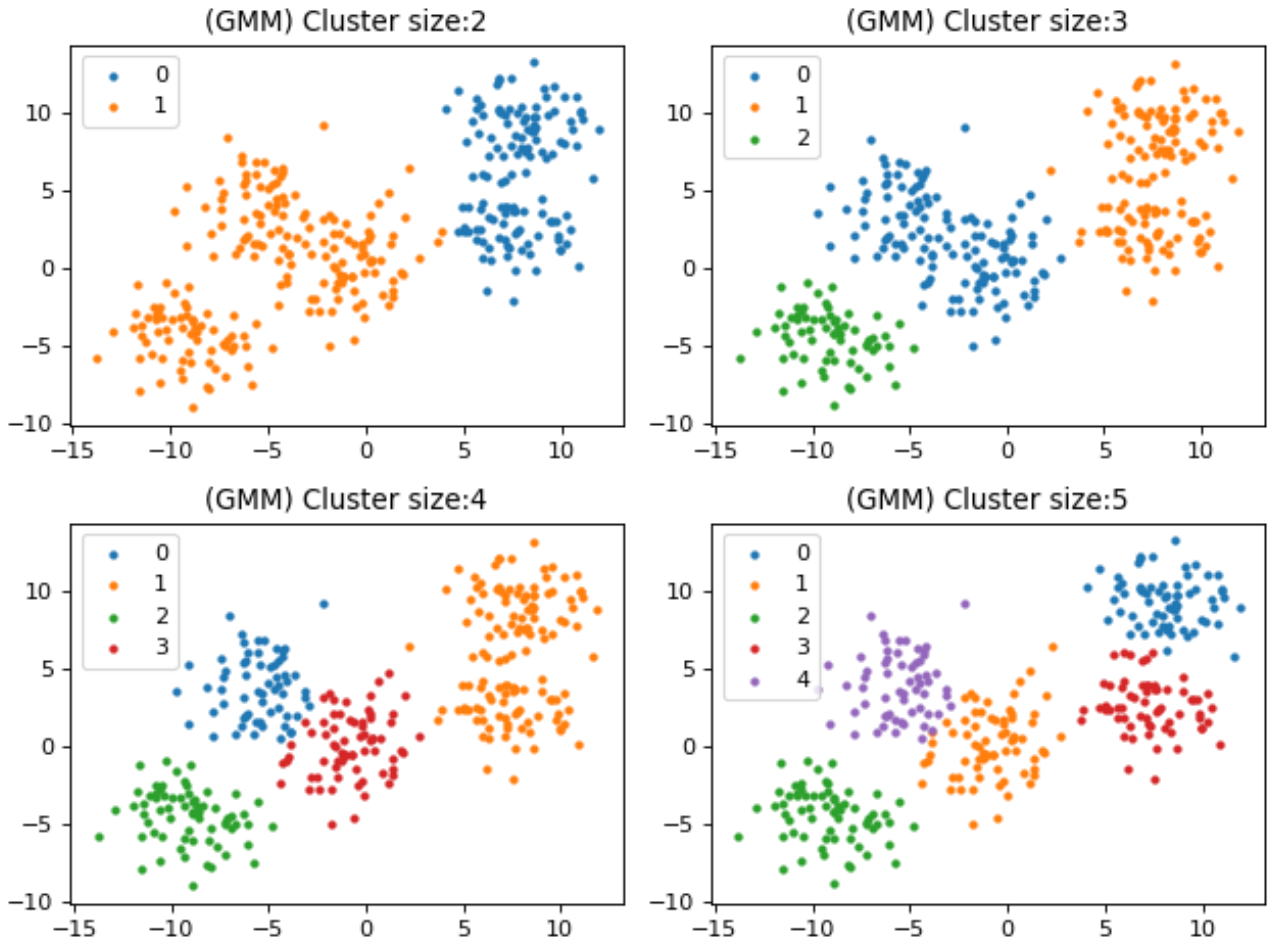
- Thay đổi số cụm và vẽ các cụm này trong cùng đồ thị

```
f = plt.figure(figsize=(8, 6), dpi=80)
f.add_subplot(2, 2, 1)

for i in range(2, 6):
 gm = GaussianMixture(n_components=i).fit(x)
 pred = gm.predict(x)
 df = DataFrame({'x':x[:,0], 'y':x[:,1], 'label':pred})
 groups = df.groupby('label')
 f.add_subplot(2, 2, i-1)
 for name, group in groups:
    plt.scatter(group.x, group.y, label=name, s=8)
    plt.title("(GMM) Cluster size:" + str(i))
    plt.legend()

plt.tight_layout()

plt.show()
```

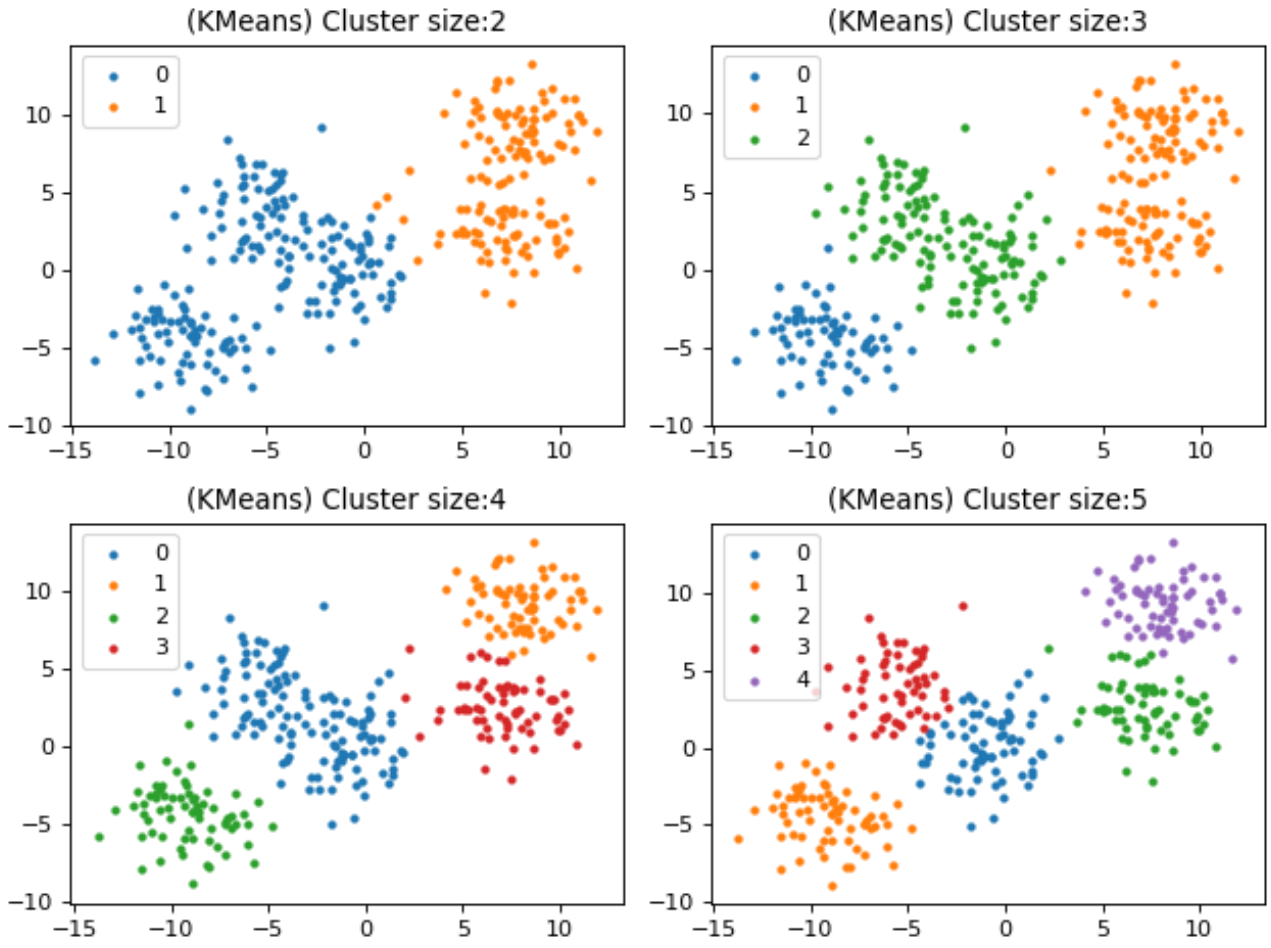


- Gom cụm sử dụng k-means

```
f = plt.figure(figsize=(8, 6), dpi=80)
f.add_subplot(2, 2, 1)

for i in range(2, 6):
 km = KMeans(n_clusters=i).fit(x)
 pred = km.predict(x)
 df = DataFrame({'x':x[:,0], 'y':x[:,1], 'label':pred})
 groups = df.groupby('label')
 f.add_subplot(2, 2, i-1)
 for name, group in groups:
    plt.scatter(group.x, group.y, label=name, s=8)
    plt.title("(KMeans) Cluster size:" + str(i))
    plt.legend()

plt.tight_layout()
plt.show()
```



2. Cài đặt thuật toán Mahalanobis k-means

- Cài đặt thuật toán.

- Trình bày tóm tắt phần code do em tự code trong file báo cáo. Em có nhận xét gì về GMM, thuật toán k-means và Mahalanobis k-means?