ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

AI23 (23TNT1), FIT@HCMUS-VNUHCM

Báo cáo Lab 2

Đề tài: PCA và bài toán phân cụm

Môn học: Phương pháp toán cho Trí tuệ nhân tạo

Sinh viên thực hiện:
Nguyễn Đình Hà Dương (23122002)
Nguyễn Lê Hoàng Trung (23122004)
Đinh Đức Tài (23122013)
Hoàng Minh Trung (23122014)

Giáo viên hướng dẫn:
TS. Cấn Trần Thành Trung
ThS. Nguyễn Ngọc Toàn

Ngày 17 tháng 5 năm 2025

Mục lục

1 Giới thiệu

Đây là bài báo cáo cho **Lab 2 - PCA và bài toán phân cụm**, môn Phương pháp toán cho Trí tuệ nhân tạo, lớp Trí tuệ nhân tạo Khóa 2023 (23TNT1), Khoa Công nghệ thông tin, Trường Đại học Khoa học tự nhiên - Đại học Quốc gia TP.HCM. Trong bài báo cáo này, chúng tôi sẽ trình bày phương pháp **phân cụm dữ liệu** bằng kĩ thuật giảm chiều dữ liệu **PCA** cùng hai thuật toán phân cụm **K-Means** và **GMM** trên bộ dữ liệu IRIS và ABIDE II.

Báo cáo được thực hiện bởi nhóm các thành viên:

- Nguyễn Đình Hà Dương (23122002)
- Nguyễn Lê Hoàng Trung (23122004)
- Đinh Đức Tài (23122013)
- Hoàng Minh Trung (23122014)

Đường dẫn repository Github của báo cáo: https://github.com/ductai05/Math-For-AI [1]

Bảng phân công nhiệm vụ cho từng thành viên:

Bảng 1

Họ và tên	MSSV	Nhiệm vụ
Nguyễn Đình Hà Dương	23122002	- Code & báo cáo K-Means, GMM. - Báo cáo evaluation metrics.
Nguyễn Lê Hoàng Trung	23122004	- Code & báo cáo K-Means, GMM. - Code & báo cáo so sánh kết quả phân cụm.
Đinh Đức Tài	23122013	- Code & báo cáo Iris dataset. Code evaluation metrics. - Review code & báo cáo. Kết luận.
Hoàng Minh Trung	23122014	- Code Class MyPCA & Z-score & Hungary algorithm - Báo cáo MyPCA, EVR, CEVR

Các thư viện và công nghệ sử dụng:

- Numpy, Pandas: thư viện Python để xử lý số học, thao tác và xử lý dữ liệu.
- scikit-learn: thư viện học máy, dùng để tải dữ liệu IRIS
- Matplotlib: thư viện Python để trực quan hóa dữ liệu.
- Jupyter Notebook (thông qua jupyter, ipykernel): Môi trường làm việc tương tác cho phép kết hợp mã thực thi, văn bản mô tả (Markdown), công thức toán học và trực quan hóa trong cùng một tài liệu.
- Visual Studio Code: Trình soạn thảo mã nguồn (IDE).
- Git, Github: Quản lý dự án, lưu và chia sẻ source code.

2 Nền tảng toán học

Các kiến thức trong phần này được trích dẫn từ sách Giáo trình bài tập Xác suất thống kê [2], trường Đại học Khoa học tự nhiên, ĐHQG-HCM và một số trang thông tin khác.

2.1 Chuẩn hóa Z-score

Chuẩn hóa Z-score là phương pháp biến đổi dữ liệu có phân phối chuẩn bất kì về phân phối chuẩn hóa. Tức là, nếu u là giá trị chuẩn hóa của dữ liệu ban đầu thì $u \sim N(0, 1)$.

Dữ liệu sau quá trình chuẩn hóa thường được gọi là dữ liệu chuẩn hóa hoặc **điểm Z (Z-scores)**. Giá trị của Z-score thường nằm trong khoảng $[-3, 3]$.

Công thức chuẩn hóa đối với tổng thể: Nếu một biến ngẫu nhiên X tuân theo phân phối chuẩn (tức là $X \sim N(\mu, \sigma^2)$), thì điểm Z được tính bằng công thức:

$$Z = \frac{X - \mu}{\sigma}$$

Trong đó:

- Z: Điểm Z chuẩn hóa.
- X: Giá trị của biến ngẫu nhiên hoặc một giá trị cụ thể từ tổng thể.
- μ: Trung bình của tổng thể.
- σ: Độ lệch chuẩn của tổng thể.

Công thức chuẩn hóa đối với dữ liệu mẫu:

Trong thực tế, chúng ta thường làm việc với dữ liệu mẫu và không biết μ và σ. Khi đó, chúng ta sẽ ước lượng chúng bằng trung bình mẫu ($\bar{x}$) và độ lệch chuẩn mẫu (s). Điểm Z cho một giá trị cụ thể x trong mẫu được tính bằng công thức:

$$z = \frac{x - \bar{x}}{s}$$

Trong đó:

- z: Giá trị chuẩn hóa (điểm Z) của x dựa trên mẫu.
- x: Giá trị dữ liệu gốc trong mẫu.
- $\bar{x}$: Trung bình mẫu của dữ liệu (sample mean).
- s: Độ lệch chuẩn mẫu của dữ liệu.

2.2 Hiệp phương sai (Covariance)

Hiệp phương sai (Covariance) [3] là thước đo mối liên hệ tuyến tính giữa hai biến ngẫu nhiên X và Y. Ký hiệu: $cov(X, Y)$. Hiệp phương sai giữa hai biến ngẫu nhiên X và Y còn được định nghĩa là kỳ vọng của tích giữa độ lệch của X và Y so với giá trị kỳ vọng của chúng.

Công thức tính hiệp phương sai:

$$cov(X, Y) = E[(X - E(X))(Y - E(Y))]$$

Công thức tính trên tổng thể:

$$cov(X, Y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_X)(y_i - \mu_Y)$$

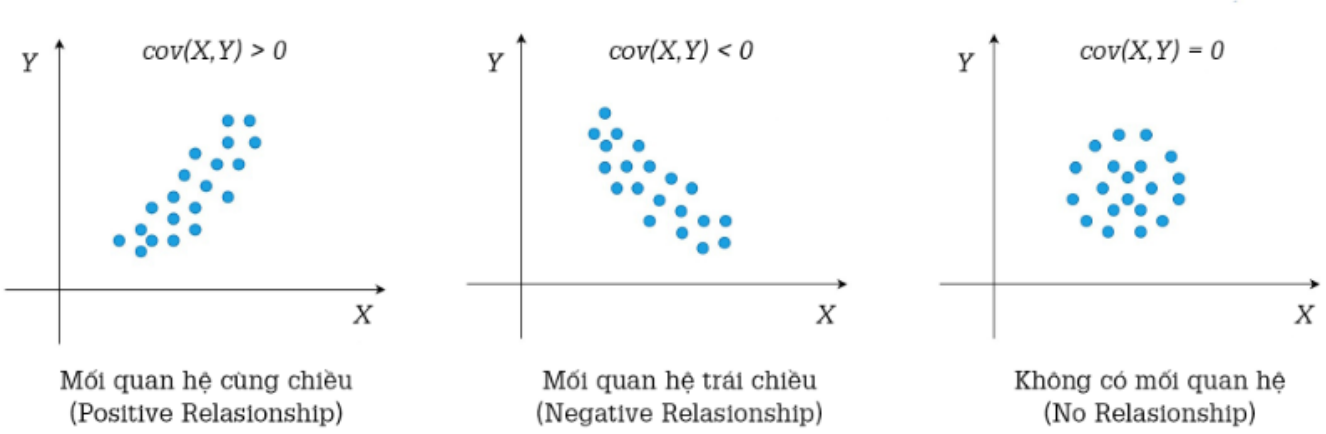
Công thức tính trên mẫu:

$$cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

Trong đó:

- x_i, y_i là giá trị của quan sát thứ i.
- μ_X, μ_Y là giá trị trung bình của tổng thể.
- $\bar{x}, \bar{y}$ là giá trị trung bình của mẫu.
- N là tổng số quan sát của tổng thể.
- n là tổng số quan sát của mẫu.

Trực quan bằng đồ thị:



Hình 1: Minh họa hiệp phương sai giữa hai biến ngẫu nhiên X và Y.

Đồ thị trên minh họa ba trường hợp có thể xảy ra khi tính hiệp phương sai:

- Khi $\text{cov}(X, Y) > 0$: Hai biến X và Y có quan hệ tuyến tính thuận, khi X tăng thì Y cũng tăng.
- Khi $\text{cov}(X, Y) < 0$: Hai biến X và Y có quan hệ tuyến tính nghịch, khi X tăng thì Y giảm và ngược lại.
- Khi $\text{cov}(X, Y) = 0$: Hai biến X và Y không có mối quan hệ tuyến tính với nhau.

2.3 Ma trận hiệp phương sai (Covariance matrix)

Ma trận hiệp phương sai là một ma trận vuông chứa các hiệp phương sai giữa các biến trong một tập dữ liệu. Nếu một tập dữ liệu có p biến ngẫu nhiên $X_1, X_2, \ldots, X_p$, thì ma trận hiệp phương sai Σ có dạng:

$$\Sigma = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \ldots & \text{Cov}(X_1, X_p) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \ldots & \text{Cov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_p, X_1) & \text{Cov}(X_p, X_2) & \ldots & \text{Var}(X_p) \end{bmatrix}$$

Trong đó:

- $\text{Var}(X_i) = \text{Cov}(X_i, X_i)$ là phương sai của biến X_i.
- $\text{Cov}(X_i, X_j)$ là hiệp phương sai giữa hai biến X_i và X_j.

Công thức tổng quát: Giả sử có một tập dữ liệu với n quan sát và p biến ngẫu nhiên được biểu diễn dưới dạng ma trận X có kích thước $n \times p$, với mỗi hàng là một quan sát và mỗi cột là một biến. Khi đó, ma trận hiệp phương sai được tính bằng công thức:

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})^T$$

Trong đó:

- X_i là vector giá trị của các biến tại quan sát thứ i.
- $\bar{X}$ là vector trung bình của từng biến.

Tính chất:

- Ma trận hiệp phương sai là **ma trận đối xứng**.
- Đường chéo chứa phương sai của từng biến.
- Nếu các biến không có tương quan (độc lập tuyến tính), thì các phần tử ngoài đường chéo bằng 0.

3 PCA, EVR, CEVR

3.1 PCA

Giới thiệu tổng quan:

Phân tích thành phần chính (PCA) là kỹ thuật giảm chiều, chuyển dữ liệu từ không gian ban đầu sang không gian mới với các thành phần chính không tương quan, giữ phần lớn phương sai. Hữu ích khi xử lý với các dữ liệu cao chiều.

Các bước tính toán:

1. **Tính vector kì vọng của toàn bộ dữ liệu:**

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N}\vec{x_i}$$

   với $\vec{x_i}$ là sample thứ i trong dữ liệu.

2. **Chuẩn hóa dữ liệu:** Với ma trận dữ liệu $\mathbf{X} \in \mathbb{R}^{n\times p}$ (n mẫu, p biến), đưa về trọng tâm:

$$\mathbf{X}_{\text{centered}} = \mathbf{X} - \overline{\mathbf{X}},$$

   với $\overline{\mathbf{X}}$ là ma trận chứa các vector trung bình của các cột (đặc trưng).

3. **Tính ma trận hiệp phương sai:**

$$\mathbf{A} = \frac{1}{N}\mathbf{X}_{\text{centered}}^{T}\mathbf{X}_{\text{centered}},$$

   trong đó $\mathbf{A} \in \mathbb{R}^{d\times d}$ là ma trận hiệp phương sai, N là số mẫu trong dữ liệu.

4. **Phân tích giá trị riêng và vector riêng:** Tìm giá trị riêng λ_i và vector riêng $\mathbf{v}_i$ của $\mathbf{A}$ sao cho:

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i, \quad i = 1, \ldots, n.$$

   Các $\mathbf{v}_i$ là các hướng của thành phần chính với λ_i là phương sai tương ứng.

5. **Lựa chọn thành phần chính:** Sắp xếp λ_i giảm dần và chọn số thành phần chính muốn giữ lại. Hoặc có thể tính tỷ lệ phương sai tích lũy như sau:

$$\text{Tỷ lệ} = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{n}\lambda_i},$$

   rồi chọn k thành phần sao cho tỷ lệ đạt 80-95% tùy nhu cầu.

6. **Chiếu dữ liệu:** Tạo ma trận $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ từ k vector riêng, chiếu dữ liệu:

$$\mathbf{Z} = \mathbf{X}_{\text{centered}}\mathbf{V}_k,$$

   với $\mathbf{Z} \in \mathbb{R}^{n\times k}$ là dữ liệu trong không gian mới.

Ý nghĩa:

- **Thành phần chính (vector riêng):** Là các hướng $\mathbf{v}_i$ tối đa hóa phương sai. Các thành phần chính trực giao với nhau, giảm chiều mà giữ thông tin chính.
- **Trị riêng:** Cho biết mức độ quan trọng (explained variance) của mỗi thành phần chính. Trị riêng càng lớn thì phương sai theo hướng đó càng lớn, dẫn đến thành phần chính đó càng quan trọng.

3.2 Explained variance ratio

Explained Variance Ratio (EVR) cho biết tỷ lệ phương sai của dữ liệu được giải thích bởi mỗi thành phần chính. EVR giúp đánh giá mức độ quan trọng tương đối của từng thành phần.

Cách tính toán: Với giá trị riêng λ_i $(i = 1, \ldots, n)$ của ma trận hiệp phương sai $\mathbf{A}$, tỷ lệ phương sai được giải thích của thành phần chính thứ i được tính bằng:

$$\text{EVR}_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j},$$

trong đó $\sum_{j=1}^{n} \lambda_j$ là tổng phương sai của dữ liệu. Giá trị EVR_i nằm trong khoảng $[0, 1]$ và thể hiện phần trăm phương sai mà thành phần chính thứ i giải thích.

Ý nghĩa: Giá trị EVR_i cao cho thấy thành phần chính thứ i nắm giữ nhiều thông tin của dữ liệu gốc. Trong ứng dụng thực tế, các thành phần chính với EVR_i lớn được ưu tiên giữ lại khi giảm chiều dữ liệu.

3.3 Cumulative explained variance ratio

Cumulative Explained Variance Ratio (CEVR) là tổng tích lũy của EVR, đo lường tổng phương sai được giữ lại khi sử dụng k thành phần chính đầu tiên. CEVR là công cụ quan trọng để quyết định số lượng thành phần chính cần giữ lại.

Cách tính toán: Với k thành phần chính đầu tiên, tỷ lệ phương sai tích lũy được giải thích được tính bằng:

$$\text{CEVR}_k = \sum_{i=1}^{k} \text{EVR}_i = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{n} \lambda_j},$$

trong đó λ_i là giá trị riêng tương ứng với thành phần chính thứ i (đã sắp xếp giảm dần). Giá trị CEVR_k tăng dần theo k và nằm trong khoảng $[0, 1]$.

Ý nghĩa:

- Giá trị CEVR_k cho biết tỷ lệ tổng thông tin dữ liệu được giữ lại khi sử dụng k thành phần chính đầu tiên.
- Thông thường, người ta chọn số thành phần chính k nhỏ nhất sao cho $\text{CEVR}_k \geq \alpha$, với α thường là 0.9 hoặc 0.95 tùy thuộc vào yêu cầu về mức độ bảo toàn thông tin.

4 Thuật toán phân cụm

4.1 K-means clustering

K-Means là một thuật toán phân cụm (clustering) phổ biến trong học máy, dùng để chia tập dữ liệu thành K cụm không giao nhau. Mỗi cụm được đại diện bởi một tâm (centroid). Mục tiêu của thuật toán là tối thiểu hóa tổng khoảng cách bình phương từ các điểm dữ liệu đến tâm cụm tương ứng.

Thuật toán Cho dữ liệu $X = \{x_i\}_{i=1}^n, x_i \in \mathbb{R}^d$ và số cụm k:

1. Khởi tạo ngẫu nhiên k tâm cụm $\{\mu_j^{(0)}\}_{j=1}^k$.
2. Lặp lại cho đến khi hội tụ (hoặc đạt số vòng lặp tối đa T):
   - **Bước gán nhãn (Assignment step)**: Với mỗi điểm dữ liệu x_i, gán nó vào cụm C_j có tâm $\mu_j^{(t)}$ gần nhất:
   $$c_i^{(t)} = \arg\min_{j\in\{1,\dots,k\}} \|x_i - \mu_j^{(t)}\|^2$$
   - **Bước cập nhật tâm (Update step)**: Với mỗi cụm C_j, cập nhật lại tâm cụm $\mu_j^{(t+1)}$ là trung bình của tất cả các điểm dữ liệu được gán vào cụm đó:
   $$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{x_i \in C_j^{(t)}} x_i$$
   - Nếu $\max_j \|\mu_j^{(t+1)} - \mu_j^{(t)}\| < \epsilon$ (một ngưỡng nhỏ) thì dừng.

4.2 Gaussian Mixture Model

Gaussian Mixture Model (GMM) là một mô hình xác suất dùng để biểu diễn sự phân bố dữ liệu như là sự kết hợp của nhiều phân phối chuẩn (Gaussian) đa biến. Nó giả định rằng dữ liệu được tạo ra từ một hỗn hợp của K phân phối Gaussian, mỗi phân phối có bộ tham số (trung bình μ_k, hiệp phương sai Σ_k) và trọng số π_k riêng.

Giả sử dữ liệu $\mathbf{x} \in \mathbb{R}^d$, GMM biểu diễn hàm mật độ xác suất như sau:

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

Trong đó $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ là tập hợp các tham số của mô hình.

- K là số thành phần (số Gaussian).
- π_k là trọng số trộn (mixing coefficient) của thành phần thứ k, với $\pi_k \geq 0$ và $\sum_{k=1}^K \pi_k = 1$.

- $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ là hàm mật độ xác suất của phân phối Gaussian đa biến thứ k:

$$\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k)\right) \tag{2}$$

Gaussian Mixture Model (GMM) thường được huấn luyện bằng thuật toán EM (Expectation-Maximization) để tìm các tham số Θ tối ưu hóa hàm hợp lý (likelihood) của dữ liệu.

Thuật toán huấn luyện: EM (Expectation-Maximization) Thuật toán EM là một phương pháp lặp để tìm ước lượng hợp lý tối đa (MLE) của các tham số trong các mô hình xác suất thống kê có biến ẩn.

- **Bước E (Expectation)**: Với các tham số hiện tại $\Theta^{(t)} = \{\pi_k^{(t)}, \mu_k^{(t)}, \Sigma_k^{(t)}\}$, tính toán xác suất hậu nghiệm (responsibility) γ_{ik} rằng điểm dữ liệu $\mathbf{x}_i$ được tạo ra bởi thành phần Gaussian thứ k:

$$\gamma_{ik}^{(t)} = \frac{\pi_k^{(t)} \cdot \mathcal{N}(\mathbf{x}_i|\mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \cdot \mathcal{N}(\mathbf{x}_i|\mu_j^{(t)}, \Sigma_j^{(t)})} \tag{3}$$

- **Bước M (Maximization)**: Cập nhật các tham số mô hình $\Theta^{(t+1)}$ dựa trên các xác suất hậu nghiệm $\gamma_{ik}^{(t)}$ đã tính ở bước E:

$$N_k^{(t+1)} = \sum_{i=1}^{N} \gamma_{ik}^{(t)} \quad \text{(số điểm hiệu dụng cho cụm } k\text{)} \tag{4}$$

$$\pi_k^{(t+1)} = \frac{N_k^{(t+1)}}{N} \tag{5}$$

$$\mu_k^{(t+1)} = \frac{1}{N_k^{(t+1)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} \cdot \mathbf{x}_i \tag{6}$$

$$\Sigma_k^{(t+1)} = \frac{1}{N_k^{(t+1)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} (\mathbf{x}_i - \mu_k^{(t+1)})(\mathbf{x}_i - \mu_k^{(t+1)})^T \tag{7}$$

Trong đó N là tổng số điểm dữ liệu.

Lặp lại các bước E và M cho đến khi hàm log-likelihood hội tụ hoặc đạt số vòng lặp tối đa.

5 Các phương pháp đánh giá mô hình

Kết quả dự đoán của mô hình phân loại (Classification) là các nhãn rời rạc, chúng ta cần các chỉ số đánh giá (evaluation metrics) để đo lường mức độ chính xác và hiệu quả trong việc phân loại các mẫu vào đúng nhãn. Các chỉ số phổ biến bao gồm **Accuracy**, **Precision**, **Recall** và **F1-score**. Mỗi chỉ số phản ánh một khía cạnh khác nhau của hiệu suất mô hình. Ngoài ra, **Confusion Matrix** cũng là một công cụ quan trọng giúp trực quan hóa chi tiết số lượng các dự đoán đúng và sai theo từng lớp, đặc biệt hữu ích trong các bài toán phân loại nhiều lớp hoặc mất cân bằng lớp.

Các ký hiệu sau được sử dụng trong các công thức đánh giá

- **TP** (True Positive): Dự đoán đúng mẫu thuộc lớp dương tính (positive class).
- **TN** (True Negative): Dự đoán đúng mẫu thuộc lớp âm tính (negative class).
- **FP** (False Positive): Dự đoán sai, mô hình dự đoán dương tính nhưng thực tế là âm tính.
- **FN** (False Negative): Dự đoán sai, mô hình dự đoán âm tính nhưng thực tế là dương tính.

5.1 Accuracy

Accuracy đo lường tỷ lệ các dự đoán chính xác trên tổng thể.Công thức của Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.2 Precision

Precision đo lường tỷ lệ dự báo chính xác các trường hợp dương tính (positive) trên tổng số trường hợp mà mô hình dự đoán là dương tính. Công thức của Precision như sau:

$$\text{Precision} = \frac{TP}{TP + FP}$$

5.3 Recall

Recall đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm Positive.Công thức của recall như sau:

$$\text{Recall} = \frac{TP}{TP + FN}$$

5.4 F1-Score

F1-score là chỉ số tổng hợp dùng để cân bằng giữa Precision và Recall. F1-score đặc biệt hữu ích khi cần sự cân bằng giữa độ chính xác và độ bao phủ trong các mô hình phân loại. Công thức của F1-score như sau:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.5 Confusion Matrix

Confusion Matrix minh họa số lượng dự đoán đúng/sai cho từng lớp. Với bài toán phân loại nhị phân, ma trận có dạng:

Actual	**Predicted**	
	Negative (No)	Positive (Yes)
Negative (No)	TN	FP
Positive (Yes)	FN	TP

Từ ma trận này, ta có thể dễ dàng tính được các chỉ số như Accuracy, Precision, Recall và F1-score.

6 Bộ dữ liệu hoa Iris

6.1 Tải dữ liệu và tổng quan về dữ liệu

Bộ dữ liệu hoa Iris hay bộ dữ liệu Iris của Fisher là một bộ dữ liệu đa biến nổi tiếng và được nhà thống kê và nhà sinh vật học người Anh Ronald Fisher sử dụng trong bài báo năm 1936 của ông: "The use of multiple measurements in taxonomic problems". [4]

Bộ dữ liệu được cung cấp bởi thư viện scikit-learn.

- **150 dòng**: thông tin về 150 mẫu hoa Iris, thuộc 3 loại: Setosa, Versicolour và Virginica.
- **5 cột**: thông tin về kích thước hoa Iris và loại hoa, trong đó:
  - `Sepal Length`: độ dài đài hoa
  - `Sepal Width`: độ rộng đài hoa
  - `Petal Length`: độ dài cánh hoa
  - `Petal Width`: độ rộng cánh hoa
  - `Target`: loại hoa (mã hóa dưới dạng 0, 1, 2)

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Hình 2: Các dòng đầu tiên trong bộ dữ liệu Iris

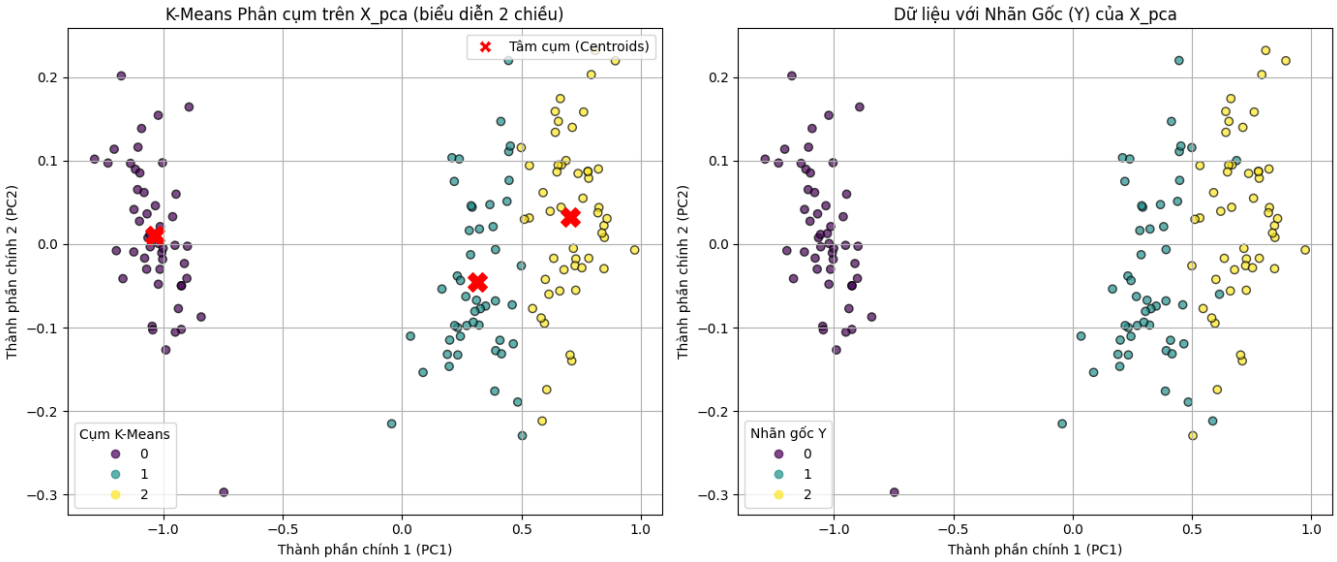
6.2 Phân tích và dự đoán loại hoa Iris

Sau khi dùng chuẩn hóa **Z-score** và phân tích thành phần chính (**PCA**) trên bộ dữ liệu Iris, chúng ta thu được:

- **EVR** (Explained variance ratio) được nắm giữ bởi thành phần chính thứ nhất (PC1) và thành phần chính thứ hai (PC2) lần lượt là `0.97343527` và `0.01707156`.
- **CEVR** (Cumulative explained variance ratio) của PC1 và PC2 là `0.990506835254475`.

Điều này thể hiện chỉ với hai thành phần chính đầu tiên, 99% tỉ lệ phương sai của dữ liệu được nắm giữ bởi PC1 và PC2.

Tiếp theo, ta áp dụng thuật toán k-means với k = 3 và thu được kết quả sau:



Hình 3: Kết quả phân cụm trên bộ dữ liệu Iris

- Kết quả phân cụm chính xác tới **96%** sau khi nối nhãn với cụm phù hợp (bằng thuật toán Hungary).
- Ba cụm điểm dữ liệu được phân biệt rõ ràng trên mặt phẳng 2 chiều và có thể nhận biết bằng mắt thường.

7 Bộ dữ liệu ABIDE II

7.1 Tải dữ liệu và tổng quan về dữ liệu

Bộ cơ sở dữ liệu **ABIDE II** [5] được giới thiệu ở NeuroHackademy 2020 bởi giáo sư Tal Yarkoni. Bộ dữ liệu đã được thay đổi một ít để phù hợp với lab này, cụ thể sẽ là phân cụm bệnh nhân có bị ung thư (cancer) hay không (normal).

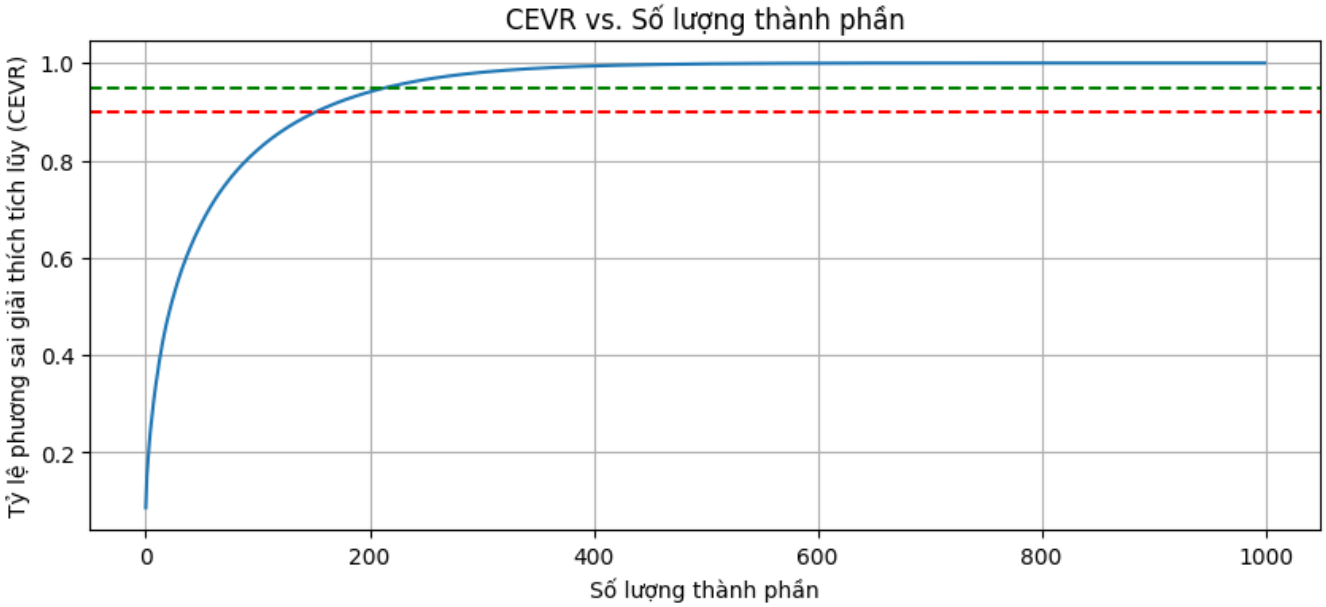
Dữ liệu được cung cấp trong file `ABIDE2(updated).csv`, trong đó:

- Số dòng: 1004. Trong đó có 463 bệnh nhân bị ung thư và 541 bệnh nhân không bị ung thư.
- Số cột: 1444
  - `Unnamed: 0`: Chỉ số dòng (index)
  - `Site`: Nơi thu thập dữ liệu (ví dụ: `ABIDEII-KKI_1`)
  - `Subject`: ID của đối tượng nghiên cứu
  - `Age`: Tuổi của đối tượng
  - Các chỉ số còn lại là dữ liệu đặc trưng về não bộ
    * Tiền tố `fsArea_` (FreeSurfer Area): Diện tích bề mặt của một vùng vỏ não (mm^2).
    * Tiền tố `fsVol_` (FreeSurfer Volume): Thể tích của một vùng vỏ não hoặc cấu trúc dưới vỏ não (mm^3).
    * Tiền tố `fsLGI_` (FreeSurfer Local Gyrification Index): Là một thước đo mức độ gấp cuộn của vỏ não tại một vùng cụ thể. Chỉ số này phản ánh mức độ phức tạp của các nếp cuộn não (gyri) và rãnh não (sulci) ở quy mô cục bộ.
    * Tiền tố `fsCT_` (FreeSurfer Cortical Thickness): Độ dày của vỏ não (mm).
      `Chữ cái L và R` trong tên cột: biểu thị bán cầu não trái (*Left*) hoặc phải (*Right*).
    * **ROI**: *Region of Interest* – Vùng quan tâm trong nghiên cứu ảnh não.
  - `Group`. Cột này gồm 2 giá trị là: `Cancer` và `Normal`.

7.2 Phân tích và dự đoán bệnh nhân

Sau khi dùng chuẩn hóa **Z-score** và phân tích thành phần chính (**PCA**) trên bộ dữ liệu ABIDE II, chúng ta thu được:

- **EVR** (Explained variance ratio) được nắm giữ bởi thành phần chính thứ nhất (PC1) và thành phần chính thứ hai (PC2) lần lượt là `0.08663063` và `0.06472462`.
- **CEVR** (Cumulative explained variance ratio) của PC1 và PC2 là `0.15135524985487855`.
- **CEVR** của 200 thành phần chính đầu tiên thì đạt gần tới `0.95`.

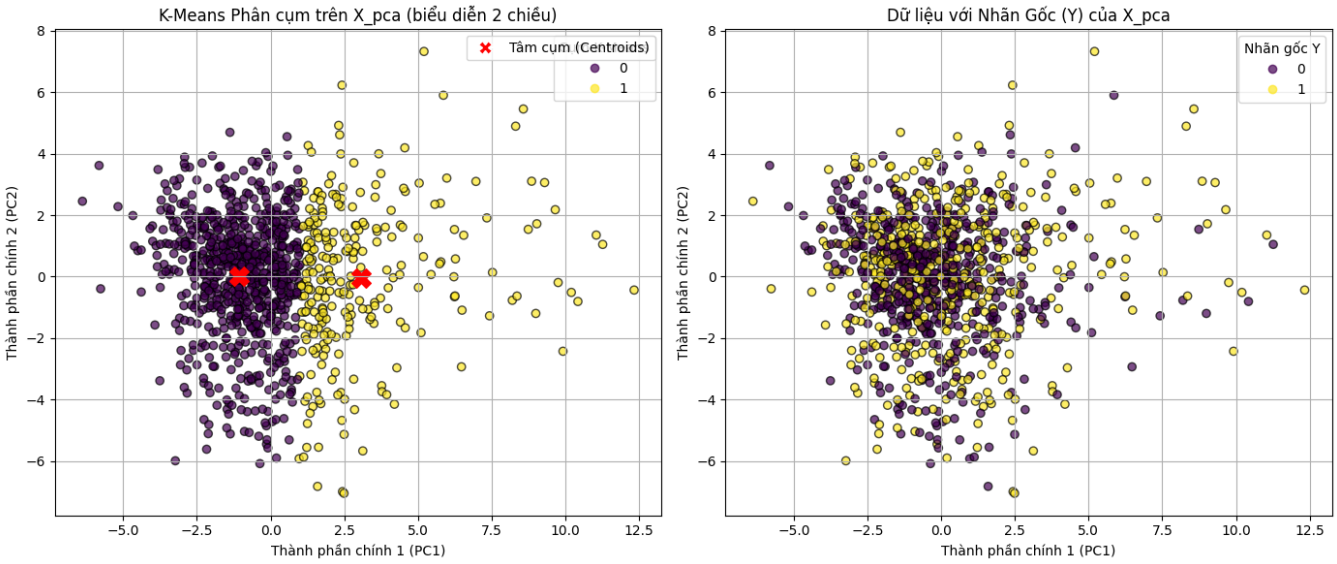


Hình 4: CEVR tương ứng với số lượng thành phần chính

Chúng tôi chọn `n-components` = 2 cho PCA vì khả năng diễn giải tốt trên mặt phẳng hai chiều và kết quả chấp nhận được với hai thuật toán phân cụm.

7.3 Sử dụng thuật toán K-Means

Tiếp theo, ta áp dụng thuật toán k-means với k = 2 và thu được kết quả sau:

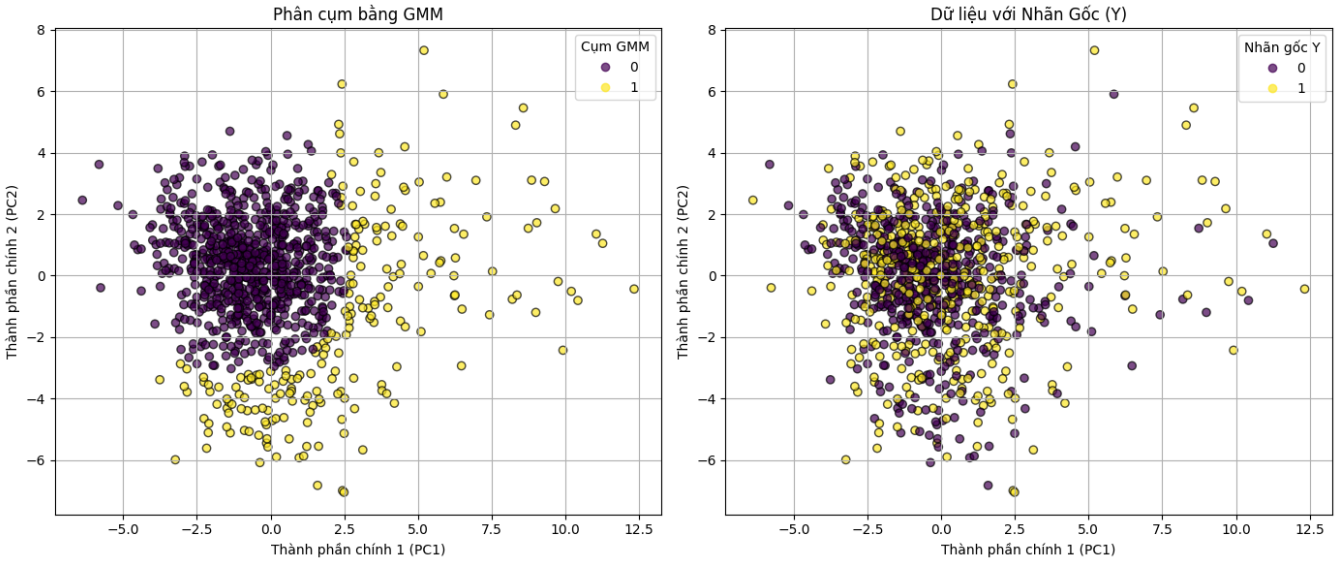


Hình 5: Kết quả phân cụm bằng KMeans trên bộ dữ liệu ABIDE II

- Kết quả phân cụm chính xác **58.1%** sau khi nối nhãn với cụm phù hợp (bằng thuật toán Hungary). `Accuracy: 0.581`. `Recall: 0.307`. `Precision: 0.587`. `F1-score: 0.403`.

- Hai cụm dữ liệu thật trên mặt phẳng hai chiều không thể phân biệt rõ được. Tuy nhiên nhãn `Normal` so với nhãn `Cancer` có mật độ cao hơn ở trung tâm.

7.4 Sử dụng thuật toán GMM



Hình 6: Kết quả phân cụm bằng GMM trên bộ dữ liệu ABIDE II

- Kết quả phân cụm chính xác **56.7%** sau khi nối nhãn với cụm phù hợp (bằng thuật toán Hungary). `Accuracy: 0.567`. `Recall: 0.266`. `Precision: 0.564`. `F1-score: 0.361`.

7.5 Nhận xét kết quả phân cụm

Khi áp dụng cả hai thuật toán K-Means và GMM lên bộ dữ liệu ABIDE II (sau khi giảm chiều bằng PCA xuống còn 2 thành phần chính), chúng ta có thể đưa ra một số nhận xét sau:

- Với giá trị F1-score khá thấp nên cả hai thuật toán K-Means và GMM đều cho thấy hiệu suất phân cụm chưa cao khi đánh giá dựa trên nhãn thực tế, với độ chính xác (Accuracy) chỉ nhỉnh hơn một chút so với mức ngẫu nhiên. Điều này cho thấy rằng cấu trúc tự nhiên của dữ liệu không dễ dàng tách thành hai cụm tương ứng hoàn toàn với hai nhóm Cancel và Normal.

- Đặc biệt, chỉ số Recall ở cả hai thuật toán đều khá thấp, cho thấy mô hình gặp khó khăn trong việc xác định đúng các trường hợp thuộc nhóm "Cancer" (nếu giả định "Cancer" là lớp positive). Precision tuy cao hơn Recall nhưng vẫn ở mức trung bình.

- Quan sát biểu đồ trực quan (Hình 5 và Hình 6), cả K-Means và GMM đều cho thấy sự chồng lấn đáng kể giữa các cụm dữ liệu. Không có một ranh giới rõ ràng nào có thể tách biệt hoàn toàn hai nhóm bệnh nhân. Điều này lý giải tại sao độ chính xác đạt được không cao. Mặc dù có ghi nhận rằng nhãn "Normal" có mật độ cao hơn ở trung tâm, sự phân bố tổng thể vẫn rất phức tạp.

8 Kết luận

8.1 Bộ dữ liệu hoa Iris

Sử dụng phương pháp **PCA** và thuật toán **k-means** cho kết quả phân cụm chính xác đến **96%** trên bộ dữ liệu hoa Iris. Điều đó cho thấy chỉ cần PCA và k-means - một thuật toán học không giám sát là đủ để phân loại trên bộ dữ liệu hoa Iris.

8.2 Bộ dữ liệu ABIDE II

Sử dụng phương pháp **PCA** và thuật toán **k-means** cho kết quả phân cụm chính xác **58.1%** trên bộ dữ liệu ABIDE II (đã qua chỉnh sửa). Các thông số khác: `Recall: 0.307`, `Precision: 0.587`, `F1-score: 0.403`.

Sử dụng phương pháp **PCA** và thuật toán **GMM** cho kết quả phân cụm chính xác **56.7%** trên bộ dữ liệu ABIDE II (đã qua chỉnh sửa). Các thông số khác: `Recall: 0.266`, `Precision: 0.564`, `F1-score: 0.361`.

Qua các độ đo trên, đặc biệt là `Accuracy: 0.576`, ta thấy chỉ dùng các phương pháp biến đổi số học như PCA và mô hình học máy học không giám sát như K-Means và GMM khó phân loại được bệnh nhân ung thư (cancer) và người bình thường (normal) trên bộ dữ liệu ABIDE II (đã qua chỉnh sửa).

Nếu được sử dụng các kĩ thuật học sâu (deep learning), chúng tôi đề xuất dùng Generative Adversarial Network (GAN) và Graph Convolution Network (GCN) [6] cùng với học có giám sát để có thể cải thiện hơn trong tác vụ dự đoán người bệnh.

Tài liệu

[1] Ha Duong, Hoang Trung, Duc Tai, and Minh Trung. Math for AI - MTH00056, AI23@HCMUS. https://github.com/ductai05/Math-For-AI/, 2025.

[2] NTM Ngoc, NV Thin, NTH Nhung, and ND Minh. *Giáo trình bài tập Xác suất thống kê*. Nhà xuất bản Đại học Quốc gia TP.HCM, HCMC, 1st edition, 2022.

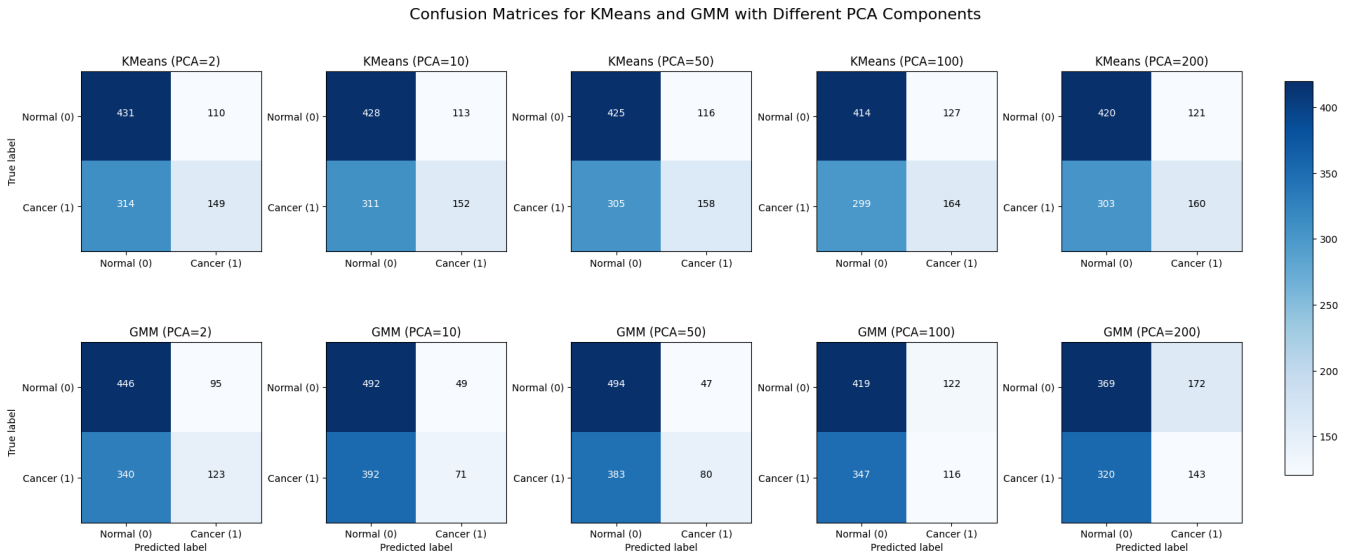
[3] Trung tâm Thông tin Khoa học và Công nghệ TP.HCM. Thống kê mô tả trong nghiên cứu. https://thongke.cesti.gov.vn/dich-vu-thong-ke/tai-lieu-phan-tich-thong-ke/861-thong-ke-mo-ta-trong-nghien-cuu-dai-luong-tuong-quan.

[4] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

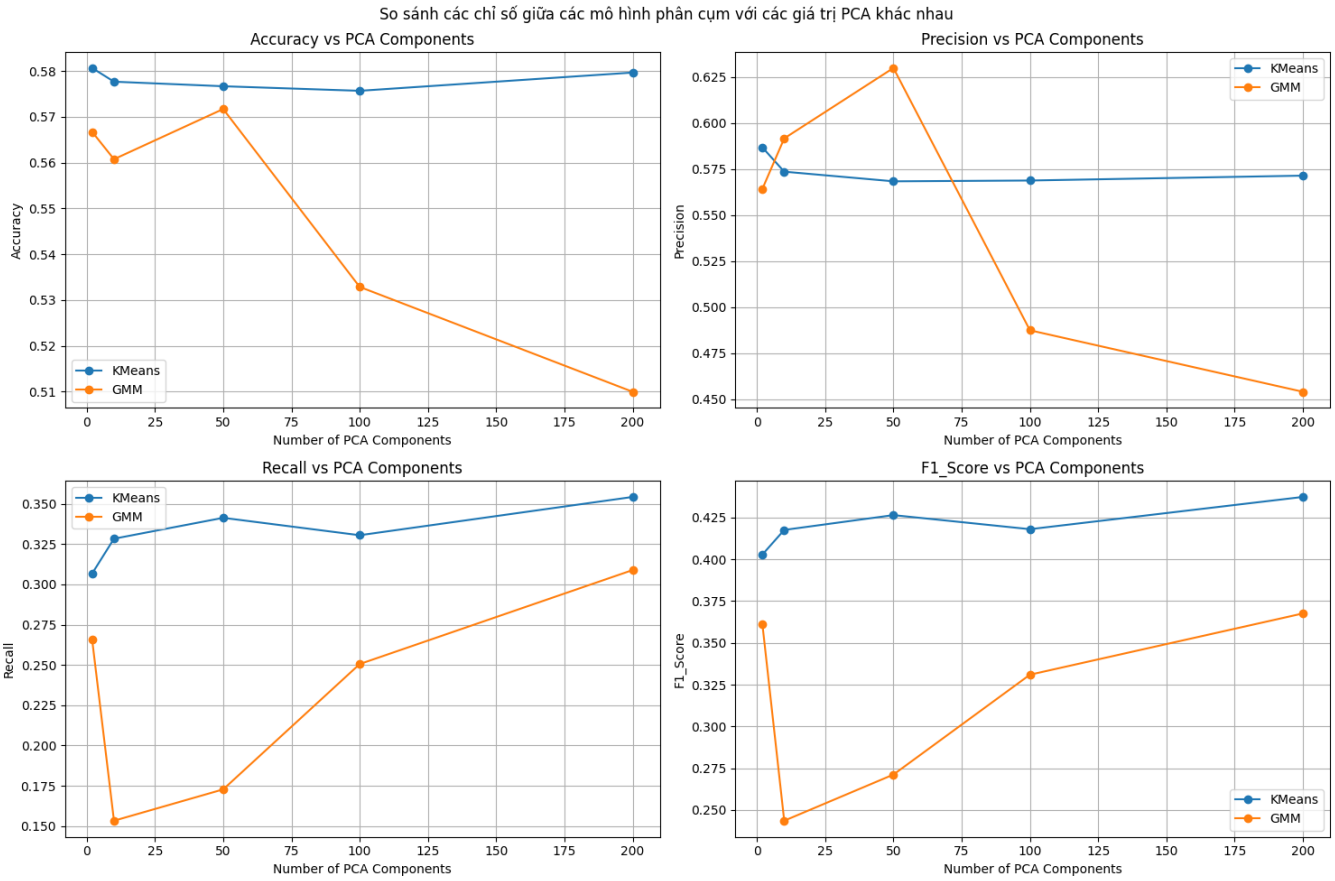
[5] Adriana Di Martino, Daniel O'Connor, Betty Chen, Kaat Alaerts, Jeffrey S. Anderson, Michal Assaf, Joshua H. Balsters, Leslie Baxter, Boris C. Bernhardt, Laura M.E. Blanken, and others. Enhancing studies of the connectome in autism using the autism brain imaging data exchange ii. *Scientific Data*, 4(1):1–13, 2017.

[6] Nguyen Huynh, Da Yan, Yueen Ma, Shengbin Wu, Cheng Long, Mirza Tanzim Sami, Abdullateef Almudaifer, Zhe Jiang, Haiquan Chen, Michael N Dretsch, et al. The use of generative adversarial network and graph convolution network for neuroimaging-based diagnostic classification. *Brain Sciences*, 14(5):456, 2024.

A Phụ lục



Hình 7: Confusion matrix cho thuật toán KMeans và GMM với các số thành phần chính PCA khác nhau



Hình 8: So sánh các chỉ số đánh giá giữa các mô hình phân cụm với các số thành phần chính PCA khác nhau