

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**  
 MÔN HỌC: KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI:**

**Targeted Advertising**

**Giảng viên hướng dẫn: Lê Thị Tú Kiên**

**Nhóm sinh viên thực hiện: Nhóm 2**

**1.** **Nguyễn Thuỳ Linh – 2251061823**

**2.** **Nguyễn Thị Thanh Thảo - 2151061200**

**3.** **Dương Văn Đối - 2151060223**

**4.** **Đoàn Ngọc Dịu - 2151062732**

**5 Nguyễn Công Đồng - 2151060254**

**6. Hà Anh Tú - 2151062887**

**Hà Nội, Tháng 1 năm 2025**

LỜI NÓI ĐẦU

Lời đầu tiên, chúng em xin chân thành cảm ơn cô Lê Thị Tú Kiên đã luôn đồng hành, hỗ trợ và hướng dẫn chúng em trong suốt quá trình thực hiện dự án này. Sự tận tâm và nhiệt huyết của cô không chỉ giúp nhóm chúng em phát triển và hoàn thiện tiểu luận mà còn tạo điều kiện thuận lợi trong quá trình học tập tại lớp và các buổi thực hành. Những lời góp ý quý báu và sự tạo cơ hội từ cô đã giúp nhóm chúng em hoàn thành báo cáo này một cách tốt nhất.

Chúng ta đang sống trong một thời đại mà công nghệ phát triển mạnh mẽ, ảnh hưởng sâu rộng đến mọi lĩnh vực trong cuộc sống, đặc biệt là trong ngành giải trí kỹ thuật số. Tiểu luận này là một nỗ lực nhỏ nhưng đầy ý nghĩa nhằm khám phá một trong những ứng dụng mạnh mẽ của dữ liệu lớn – quảng cáo nhắm mục tiêu – và cách phân tích dữ liệu có thể tạo ra những trải nghiệm người dùng cá nhân hóa.

Chúng ta đã chứng kiến sự phát triển vượt bậc của các dịch vụ phát trực tuyến, nơi sự lựa chọn của người xem không chỉ dựa trên sở thích cá nhân mà còn được tinh chỉnh thông qua các công cụ phân tích dữ liệu, như phương pháp phân tích nhóm (cluster analysis). Việc áp dụng phương pháp này giúp các dịch vụ phát trực tuyến hiểu rõ hơn về hành vi của người dùng, từ đó đưa ra những đề xuất nội dung và quảng cáo phù hợp, mang lại hiệu quả tối ưu cho cả người xem và nhà cung cấp dịch vụ.

Chúng em hy vọng thông qua tiểu luận này, sẽ mang đến cái nhìn rõ ràng hơn về mối liên hệ giữa phân tích dữ liệu và quảng cáo mục tiêu, đồng thời làm nổi bật những tiềm năng to lớn của nó trong việc nâng cao trải nghiệm người dùng trong lĩnh vực giải trí kỹ thuật số.

Mục lục:

[**Chương 1: Tổng quan về các thuật toán phân cụm** 6](#_Toc187272908)

[**I. Phân cụm dữ liệu** 6](#_Toc187272909)

[1. Khái niệm về PCDL 6](#_Toc187272910)

[2. Ứng dụng của PCDL 6](#_Toc187272911)

[3. Các vấn đề cần giải quyết đối với PCDL 6](#_Toc187272912)

[4. Phân hoạch các thuật toán PCDL 7](#_Toc187272913)

[**II. Các thuật toán phân cụm** 7](#_Toc187272914)

[**1.CLARANS** 7](#_Toc187272915)

[1.1. Giới thiệu thuật toán 7](#_Toc187272916)

[1.2. Input và Output 7](#_Toc187272917)

[1.3. Các bước thuật toán 8](#_Toc187272918)

[1.4. Điều kiện dừng 8](#_Toc187272919)

[1.5. Nhận xét thuật toán 8](#_Toc187272920)

[**2.BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)** 9](#_Toc187272921)

[2.1. Giới thiệu thuật toán 9](#_Toc187272922)

[2.2. Input và Output 9](#_Toc187272923)

[2.3. Các bước thuật toán 10](#_Toc187272924)

[2.4. Điều kiện dừng 10](#_Toc187272925)

[2.5. Xây dựng cây CF 10](#_Toc187272926)

[2.6. Nhận xét thuật toán 13](#_Toc187272927)

[**3.OPTICS** 13](#_Toc187272928)

[3.1. Giới thiệu thuật toán 13](#_Toc187272929)

[3.2. Phát biểu bài toán phân cụm 14](#_Toc187272930)

[3.3. Các bước thuật toán OPTICS 14](#_Toc187272931)

[3.4. Lưu đồ thuật toán 16](#_Toc187272932)

[3.5 Minh họa thuật toán 18](#_Toc187272933)

[3.6 Điều kiện dừng và chất lượng phân cụm 18](#_Toc187272934)

[3.7. Các tham số và ảnh hưởng 19](#_Toc187272935)

[3.8. Nhận xét thuật toán OPTICS 20](#_Toc187272936)

[3.9. So sánh với các thuật toán khác 20](#_Toc187272937)

[1. So sánh với DBSCAN 20](#_Toc187272938)

[2. So sánh với K-means 20](#_Toc187272939)

[3. So sánh với Hierarchical Clustering 21](#_Toc187272940)

[**Chương 2 : Định nghĩa bài toán và các phương pháp giải quyết bài toán** 22](#_Toc187272941)

[**I. Định nghĩa bài toán** 22](#_Toc187272942)

[1. Lí do chọn đề tài: 22](#_Toc187272943)

[2. Tổng quan bài toán 22](#_Toc187272944)

[3. Quy trình thực hiện 23](#_Toc187272945)

[**II.Các phương pháp giải quyết bài toán** 24](#_Toc187272946)

[**1.Thuật toán CLARANS** 24](#_Toc187272947)

[**1. Chuẩn bị dữ liệu** 24](#_Toc187272948)

[**2. Áp dụng thuật toán Clarans** 24](#_Toc187272949)

[**2.Thuật toán BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)** 25](#_Toc187272950)

[**3.Thuật toán OPTICS** 26](#_Toc187272951)

[**1. Chuẩn bị dữ liệu** 26](#_Toc187272952)

[**2. Áp dụng thuật toán OPTICS** 26](#_Toc187272953)

[**3. Triển khai quảng cáo cá nhân hóa** 27](#_Toc187272954)

[**4. Ưu điểm của việc dùng OPTICS cho bài toán này** 27](#_Toc187272955)

[**5. Kết luận** 28](#_Toc187272956)

[**Chương 3: Thực nghiệm so sánh đánh giá kết quả** 28](#_Toc187272957)

[**I. Thu thập dữ liệu** 28](#_Toc187272958)

[**1. Mô tả dữ liệu (Dataset Description):** 28](#_Toc187272959)

[**2. Các trường dữ liệu trong từng tập tin:** 28](#_Toc187272960)

[**3. Quy trình thu thập và làm sạch dữ liệu:** 30](#_Toc187272961)

[**4. Các câu hỏi phân tích có thể thực hiện:** 30](#_Toc187272962)

[**II. Tìm hiểu tập dữ liệu** 31](#_Toc187272963)

[**1. Đặc trưng về người dùng** 31](#_Toc187272964)

[**2. Đặc trưng về phim** 33](#_Toc187272965)

[**3. Ý nghĩa của dữ liệu** 34](#_Toc187272966)

[**III. Tiền xử lý dữ liệu** 34](#_Toc187272967)

[**1. Xử lý và chuẩn bị dữ liệu** 34](#_Toc187272968)

[1.1. Làm sạch dữ liệu 34](#_Toc187272969)

[1.2. Chuyển đổi dữ liệu 34](#_Toc187272970)

[1.3. Kết hợp dữ liệu 34](#_Toc187272971)

[**2. Đặc trưng về người dùng** 34](#_Toc187272972)

[2.1. Rating (Đánh giá phim) 34](#_Toc187272973)

[2.2. Gender (Giới tính) 34](#_Toc187272974)

[2.3. Age (Tuổi) 35](#_Toc187272975)

[2.4. Occupation (Nghề nghiệp) 35](#_Toc187272976)

[**3. Đặc trưng về phim** 36](#_Toc187272977)

[3.1. Genres (Thể loại phim) 36](#_Toc187272978)

[**IV. Cài đặt thực nghiệm trên các phương pháp đã chọn** 37](#_Toc187272979)

[**1.Thuật toán CLARANS** 37](#_Toc187272980)

[**2.Thuật toán BIRCH** 43](#_Toc187272981)

[**3.Thuật toán OPTICS:** 45](#_Toc187272982)

[**V. So sánh và đánh giá kết quả thu được** 51](#_Toc187272983)

[**Tập dữ liệu 500 dòng:** 51](#_Toc187272984)

[**Tập dữ liệu 5000 dòng:** 52](#_Toc187272985)

[**Danh mục tài liệu tham khảo** 60](#_Toc187272986)

# **Chương 1: Tổng quan về các thuật toán phân cụm**

# **I. Phân cụm dữ liệu**

## 1. Khái niệm về PCDL

* PCDL:
* Là một kĩ thuật trong KPDL
* Nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tự nhiên, tiềm ẩn, quan trọng trong tập dữ liệu lớn => cung cấp thông tin, tri thức hữu ích cho việc ra quyết định
* Là quá trình phân chia một tập dữ liệu ban đầu thành các cụm dữ liệu sao cho các phần tử trong một cụm “tương tự” với nhau và các phần tử trong các cụm khác nhau sẽ “phi tương tự” với nhau
* Trong học máy, PCDL là vấn đề học không có giám sát
* Trong nhiều trường hợp, PCDL là một bước trong phân lớp dữ liệu: Có thể khởi tạo các lớp cho phân lớp bằng cách xác định các nhãn cho các nhóm dữ liệu

## 2. Ứng dụng của PCDL

* Có thể ứng dụng PCDL trong nhiều lĩnh vực, ví dụ:
* Thương mại: Tìm kiếm nhóm các khách hàng quan trọng có đặc trưng tương đồng và những đặc tả họ từ các bản ghi mua bán trong CSDL khách hàng
* Sinh học: Phân loại các gen với các chức năng tương đồng
* Thư viện: Phân loại sách
* Bảo hiểm: Nhận dang nhóm tham gia bảo hiểm có chi phí bồi thường cao, nhận dạng gian lận
* Quy hoạch đô thị: Nhận dạng các nhóm nhà theo kiểu và vị trí địa lý …nhằm cung cấp thông tin cho quy hoạch đô thị
* Nghiên cứu trái đất: Phân cụm để theo dõi các tâm động đất nhằm cung cấp thông tin cho nhận dạng các vùng nguy hiểm
* WWW: Có thể khám phá các nhóm tài liệu quan trọng, có nhiều ý nghĩa trong môi trường Web => Các lớp tài liệu này trợ giúp cho việc KPDL từ dữ liệu

## 3. Các vấn đề cần giải quyết đối với PCDL

* Tối thiểu lượng tri thức cần cho xác định các tham số đầu vào:
* Nhiều thuật toán phân cụm yêu cầu những tham số đầu vào nhất định trong phân tích phân cụm (như số lượng các cụm mong muốn)
* Kết quả của phân cụm thường khá nhạy cảm với các tham số đầu vào
* Nhiều tham số rất khó để xác định, nhất là với các tập dữ liệu có lượng các đối tượng lớn => người dùng khó điều chỉnh để đạt chất lượng phân cụm tốt
* Khả năng thích nghi với dữ liệu nhiễu:
* Dữ liệu ngoại lai, dữ liệu lỗi, dữ liệu chưa biết hoặc dữ liệu sai
* Nhạy cảm với dữ liệu => dẫn đến chất lượng phân cụm
* Số chiều lớn
* Ít nhạy cảm với thứ tự của các dữ liệu vào:
* Cùng một tập dữ liệu, thứ tự khác nhau, cùng một thuật toán có thể sinh ra các cụm rất khác nhau
* Phát triển các thuật toán ít nhạy cảm với thứ tự vào của dữ liệu rất quan trọng
* Phân cụm ràng buộc:
* Yêu cầu: phân cụm tốt và thỏa mãn thêm các yêu cầu ràng buộc

## 4. Phân hoạch các thuật toán PCDL

* Một số thuật toán phân cụm:
* Thuật toán K-Means
* Thuật toán AGNES
* Một số tiếp cận phân cụm khác:
* Tiếp cận dựa trên mật độ
* Tiếp cận dựa trên mô hình
* Tiếp cận dựa trên lưới

# **II. Các thuật toán phân cụm**

## **1.CLARANS**

### 1.1. Giới thiệu thuật toán

Khái niệm: **CLARANS** là một thuật toán phân cụm dựa trên kỹ thuật tìm kiếm ngẫu nhiên, được đề xuất để giải quyết các hạn chế về hiệu năng của K-Medoids khi làm việc với các tập dữ liệu lớn. Thuật toán này được đề xuất bởi **Ng và Han (1994)**.

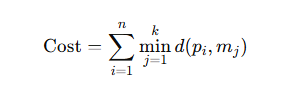
* + Mục đích
    - Giải quyết bài toán phân cụm cho dữ liệu lớn bằng cách kết hợp tìm kiếm cục bộ và ngẫu nhiên hóa.
    - Tăng hiệu quả tính toán so với K-Medoids bằng cách tìm gần đúng các trung tâm cụm tốt nhất.

### 1.2. Input và Output

* Input
  + Tập dữ liệu D chứa n điểm dữ liệu.
  + Số cụm k.
  + Số lần thử num\_local: Số lượng tối đa của các tìm kiếm cục bộ.
  + Ngưỡng max\_neighbor: Số lượng hàng xóm được duyệt trong mỗi tìm kiếm.
* Output
  + Tập các cụm C1,C2,...,Ck​ với các điểm đại diện (medoid) tối ưu.

### 1.3. Các bước thuật toán

* Bước 1: khởi tạo
  + Lấy ngẫu nhiên một tập các điểm trung tâm ban đầu (medoids).
* Bước 2: tìm kiếm cục bộ
  + Trong mỗi lần lặp, chọn ngẫu nhiên một điểm thay thế một medoid hiện tại.
  + Nếu sự thay thế này giảm chi phí tổng thể (total cost), cập nhật medoid.
    - Chi phí được tính theo công thức:



trong đó:

* pi​ là điểm dữ liệu.
* mj là medoid
* d(pi,mj) là khoảng cách giữa pi và mj
* Bước 3: kiểm tra điều kiện dừng
  + Nếu số lượng hàng xóm duyệt vượt ngưỡng max\_neighbor hoặc không tìm thấy sự cải thiện, dừng tìm kiếm cục bộ.
* Bước 4: lặp lại
  + Quay lại bước 2 và thực hiện tối đa num\_local lần tìm kiếm cục bộ để tìm giải pháp tốt nhất.

### 1.4. Điều kiện dừng

* Số lần tìm kiếm cục bộ đạt giới hạn num\_local
* Không tìm thấy hoán đổi nào có thể giảm chi phí hơn

### 1.5. Nhận xét thuật toán

* Ưu điểm:
  + Hiệu quả hơn so với K-Medoids khi áp dụng cho dữ liệu lớn.
  + Sử dụng tìm kiếm ngẫu nhiên để tránh rơi vào cực trị cục bộ.
  + Linh hoạt với các tham số numlocal và maxneighbor.
* Nhược điểm:
  + Kết quả phụ thuộc vào việc chọn các tham số ban đầu.
  + Không phù hợp với dữ liệu có cụm phức tạp hoặc không rõ ràng.

Yêu cầu tính toán lặp lại nhiều lần, tốn kém cho dữ liệu cực lớn.

## **2.BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)**

### 2.1. Giới thiệu thuật toán

- Khái niệm: BIRCH là một thuật toán phân cụm hiệu quả dành cho dữ liệu lớn, được đề xuất bởi Tian Zhang và cộng sự (Raghu Ramakrishnan, và Miron Livny) vào năm 1996. Thuật toán này dựa trên việc xây dựng cây phân cụm (Clustering Feature Tree - CF Tree) để biểu diễn dữ liệu một cách gọn gàng và giảm thiểu số lần quét dữ liệu.

- Ý tưởng thuật toán: không lưu toàn bộ các đối tượng DL của các cụm trong bộ nhớ mà chỉ lưu các đại lượng thống kê. Đó là bộ ba (n, LS, SS)

* n là số đối tượng trong cụm
* LS là tổng các giá trị thuộc tính của các đối tượng trong cụm
* SS là tổng bình phương của các giá trị thuộc tính của các đối tượng trong cụm.

-> Các bộ ba được gọi là đặc trưng của cụm(Cluster Features - CF), được lưu giữ trong một cây được gọi là cây CF (CF-tree).

* Mục đích thuật toán:
* Phân cụm dữ liệu lớn một cách hiệu quả cả về thời gian và bộ nhớ.
* Duy trì chất lượng phân cụm tốt với dữ liệu có kích thước lớn và phức tạp.

### 2.2. Input và Output

* Input:
* Tập dữ liệu D chứa n điểm dữ liệu.
* Ngưỡng T: Kích thước bán kính tối đa của các cụm trong CF Tree.
* Số cụm k (sau khi xây dựng CF Tree).
* Output: Tập các cụm C1,C2,...,Ck​.

### 2.3. Các bước thuật toán

* Bước 1: Xây dựng cây CF (Clustering Feature Tree)
  + Mỗi nút trong cây CF chứa một **Clustering Feature (CF)** đại diện cho cụm con.
  + Khi thêm một điểm dữ liệu mới, điểm dữ liệu này được chèn vào nút lá gần nhất. Nếu vượt quá bán kính T, tách nút và cân bằng lại cây.
* Bước 2: Nếu cây CF hiện thời không có đủ bộ nhớ trong thì tiến hành xây dựng một cây CF nhỏ hơn bằng cách điều khiển bởi tham số T (vì tăng T sẽ làm hoà nhập một số các cụm con thành một cụm, điều này làm cho cây CF nhỏ hơn). Bước này không cần yêu cầu bắt đầu đọc dữ liệu lại từ đầu nhưng vẫn đảm bảo hiệu chỉnh cây dữ liệu nhỏ hơn.
* Bước 3: Thực hiện phân cụm: Các nút lá của cây CF lưu giữ các đại lượng thống kê của các cụm con. Trong bước này, BIRCH sử dụng các đại lượng thống kê này để áp dụng một số kỹ thuật phân cụm thí dụ như k-means và tạo ra một khởi tạo cho phân cụm
* Bước 4: Phân phối lại các đối tượng dữ liệu bằng cách dùng các đối tượng trọng tâm cho các cụm đã được khám phá từ bước 3: Đây là một bước tuỳ chọn để duyệt lại tập dữ liệu và gán nhãn lại cho các đối tượng dữ liệu tới các trọng tâm gần nhất. Bước này nhằm để gán nhãn cho các dữ liệu khởi tạo và loại bỏ các đối tượng ngoại lai

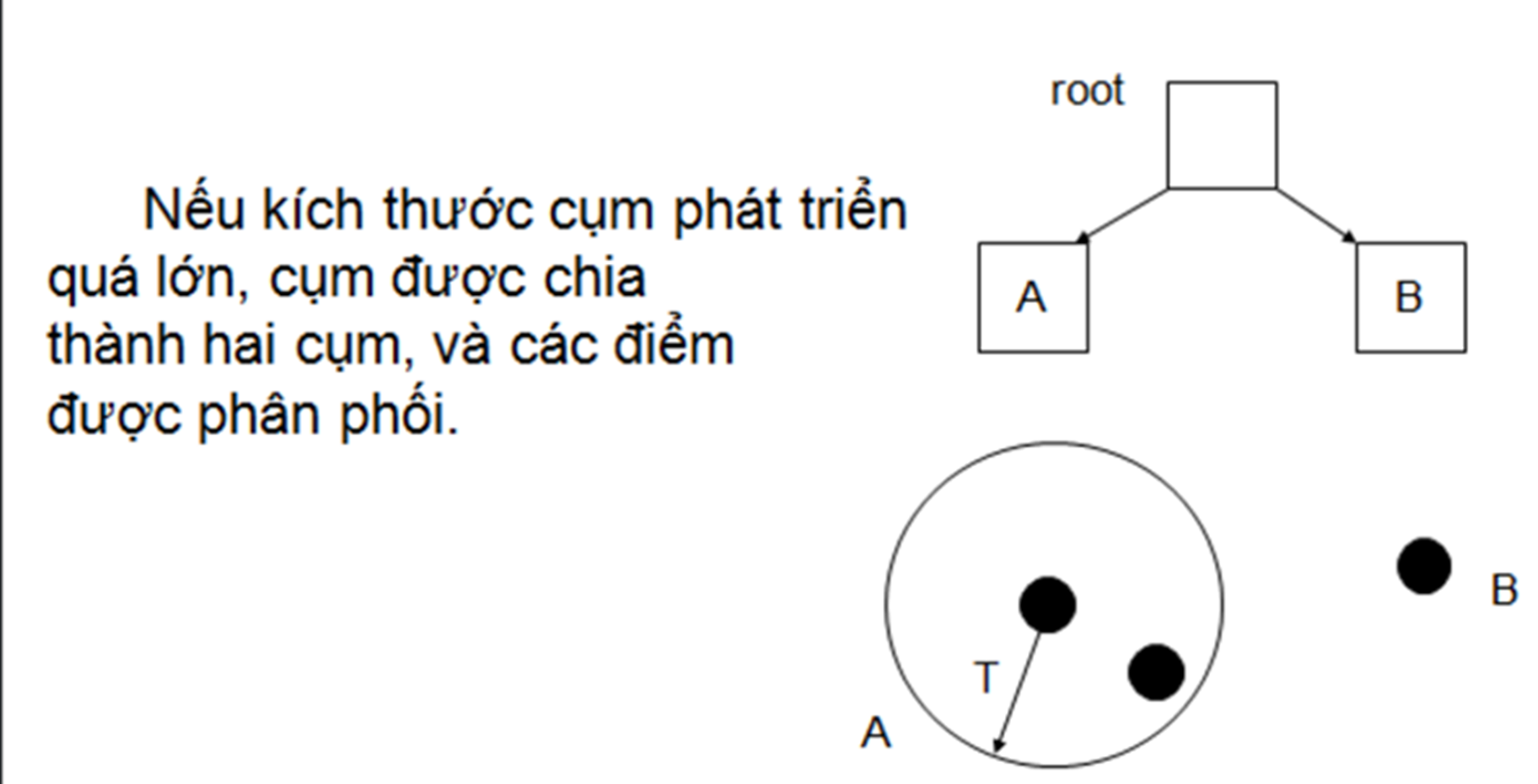
### 2.4. Điều kiện dừng

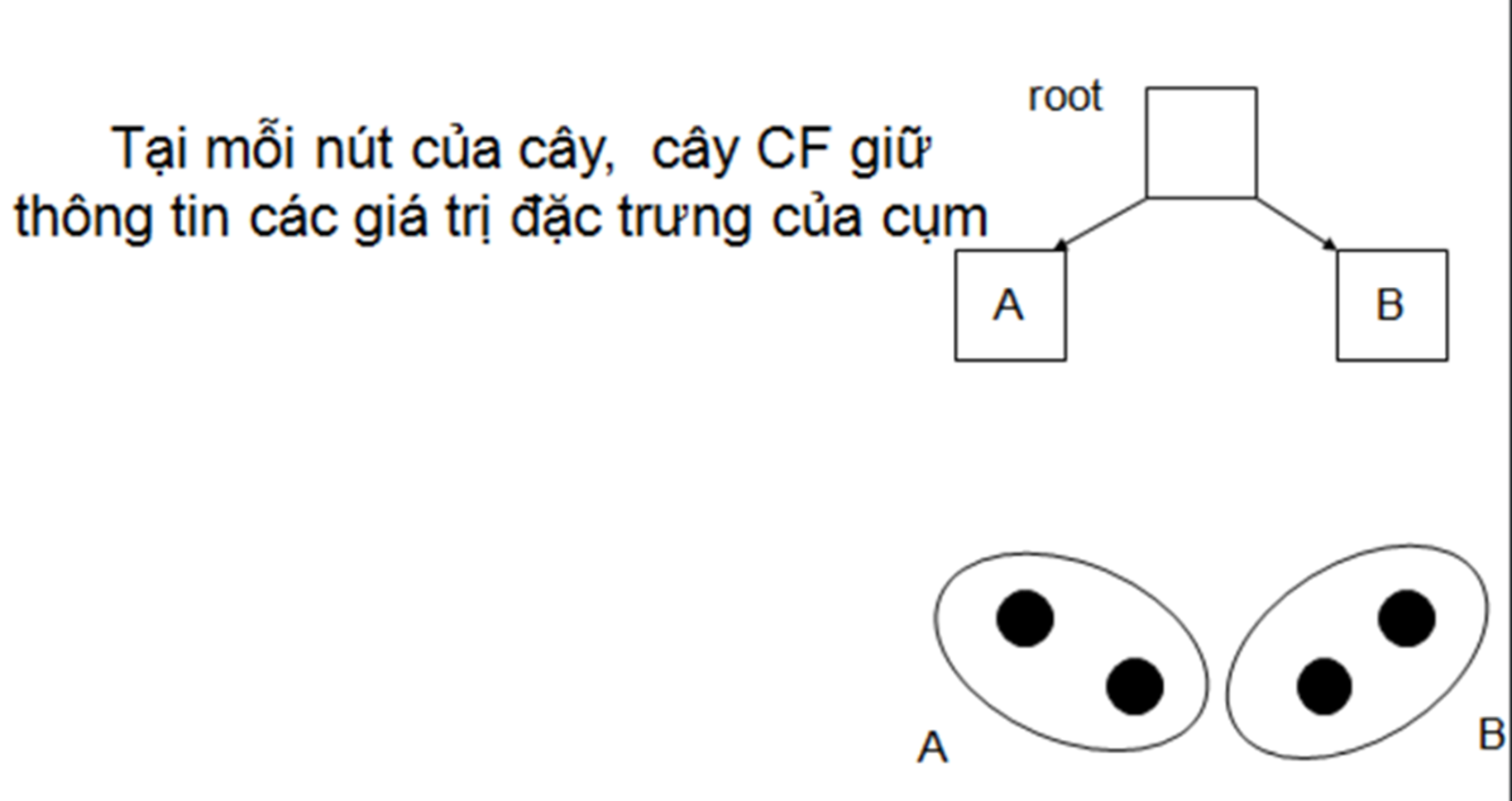
* Toàn bộ dữ liệu đã được xử lý
* Kích thước cây CF đạt giới hạn
* Số lượng cụm đã đạt yêu cầu

### 2.5. Xây dựng cây CF

Ban đầu các điểm dữ liệu nằm trong một cụm



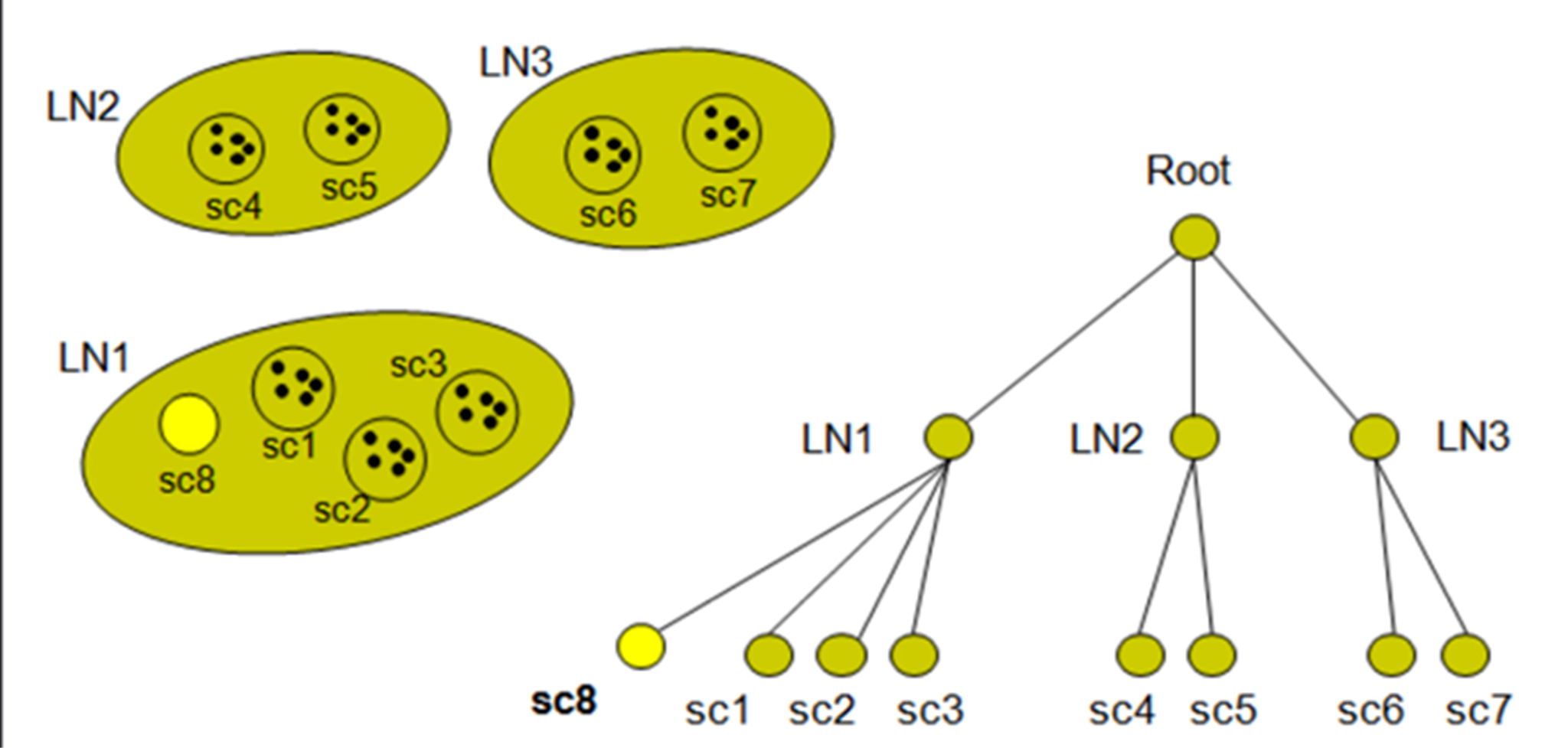


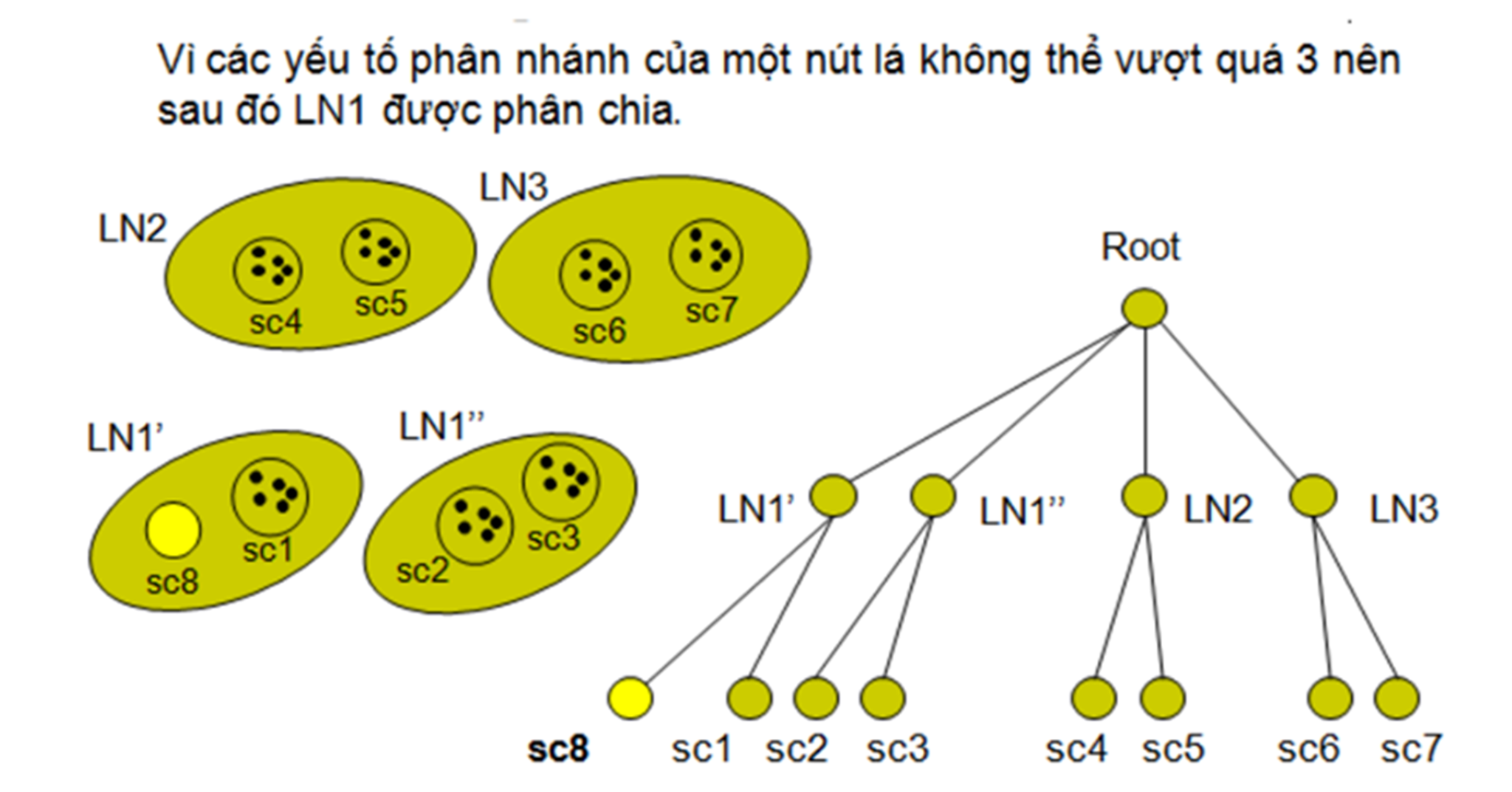


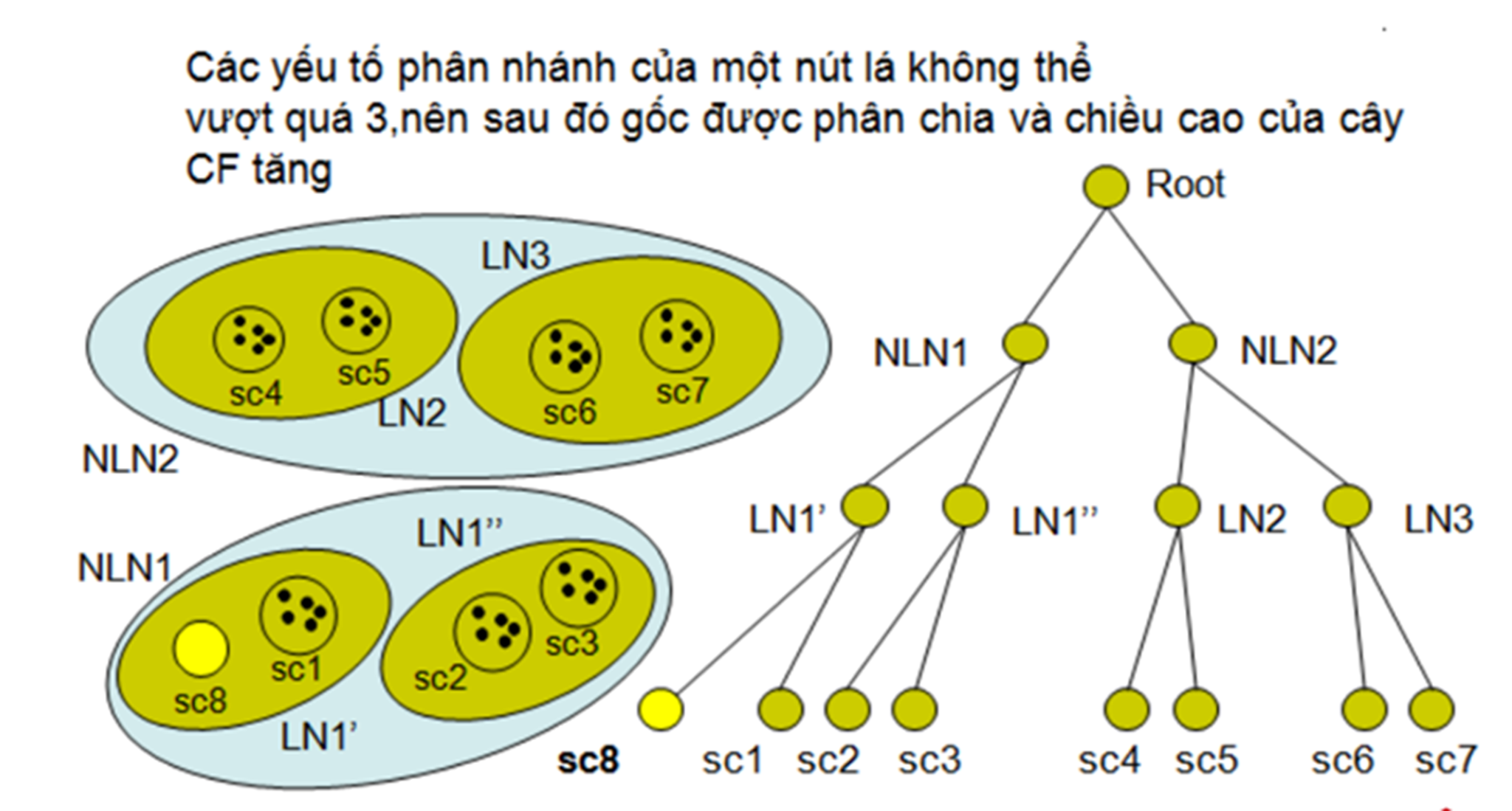
* Hoạt động chèn nút lá:

+ Có 3 cụm DL: LN1(sc1, sc2, sc3), LN2(sc4, sc5), LN3(sc6,sc7) với B=3

+ Khi xét đến điểm sc8, sc8 được phân vào cụm LN1







### 2.6. Nhận xét thuật toán

* Ưu điểm:
* Tốc độ phân cụm dữ liệu nhanh.
* Có thể áp dụng với tập dữ liệu lớn, hiệu quả hơn khi áp dụng với tập dữ liệu tăng trưởng theo thời gian.
* Duyệt toàn bộ dữ liệu một lần với một lần quét thêm tùy chọn => Độ phức tạp là O(n) (n là số đối tượng dữ liệu).
* Nhược điểm:
* Chất lượng của các cụm không được tốt
* Thực hiện tốt với dữ liệu số nếu sử dụng khoảng cách Euclide
* Không thích hợp với dữ liệu đa chiều
* Việc ép các đối tượng dữ liệu làm cho các đối tượng của một cụm có thể là đối tượng kết thúc của cụm khác, trong khi các đối tượng gần nhau có thể bị hút bởi các cụm khác nếu chúng được biểu diễn cho thuật toán theo một thứ tự khác.

## **3.OPTICS**

### 3.1. Giới thiệu thuật toán

* OPTICS hay sắp xếp các điểm để xác định cấu trúc cụm là một thuật toán phân cụm dựa trên mật độ được sử dụng trong học máy. Không giống như các phương pháp phân cụm truyền thống, OPTICS có thể khám phá các cụm có nhiều hình dạng và kích thước khác nhau. Nó đánh giá mật độ của các điểm dữ liệu trong không gian đặc điểm, xác định các vùng có mật độ cao hơn là các cụm tiềm năng.
* OPTICS Là mở rộng của DBSCAN, bằng cách giảm bớt các tham số đầu vào.
* OPTICS (Ordering Points To Identify the Clustering Structure) sắp xếp các cụm theo thứ tự tăng dần nhằm tự động phân cụm dữ liệu.
* Thuật toán tạo ra một biểu đồ khả năng tiếp cận, mô tả thứ tự các điểm được xử lý. Điều này cho phép trích xuất cụm linh hoạt và xác định các điểm nhiễu.
* OPTICS cung cấp một phương pháp tiếp cận mạnh mẽ để khám phá các cấu trúc phức tạp trong các tập dữ liệu, làm cho nó có giá trị trong các tình huống mà các phương pháp truyền thống có thể gặp khó khăn, đặc biệt là khi xử lý các cụm có hình dạng bất thường hoặc mật độ cụm thay đổi.
* Nó được trình bày bởi Mihael Ankerst, Markus M. Breunig, [Hans-Peter Kriegel](https://en.wikipedia.org/wiki/Hans-Peter_Kriegel) và Jörg Sander

### 3.2. Phát biểu bài toán phân cụm

* Thuật toán OPTICS là cần thiết lập các tham số cần thiết trước khi bắt đầu xử lý dữ liệu. Hai tham số chính là:

1. **𝜀 (epsilon)**: Bán kính để xác định vùng lân cận của mỗi điểm. Các điểm nằm trong vùng 𝜀 của một điểm được coi là hàng xóm của điểm đó.
2. **MinPts**: Số lượng điểm tối thiểu cần thiết trong vùng lân cận (bao gồm cả điểm trung tâm) để một điểm được coi là **đối tượng lõi (core object)**.

* Input :

1. Tập dữ liệu D={p1,p2,...,pn} chứa n điểm
2. Hai tham số:

* ε: Bán kính để xác định lân cận của một điểm.
* MinPts: Số lượng điểm tối thiểu cần thiết trong lân cận để xác định một cụm
* Output:

1. Một thứ tự phân cụm các điểm dữ liệu (cluster ordering).
2. Các cụm dữ liệu và các điểm nhiễu

### 3.3. Các bước thuật toán OPTICS

* Phương pháp cơ bản của OPTICS tương tự như DBSCAN , nhưng thay vì duy trì các thành viên cụm đã biết nhưng chưa được xử lý trong một tập hợp, chúng được duy trì trong hàng đợi ưu tiên

**Bước 1: Khởi tạo**:

* Chọn ngẫu nhiên một điểm p từ tập dữ liệu.
* **khoảng cách lõi (core-distance)** và **khoảng cách tiếp cận (reachability-distance)** của p ban đầu là "CHƯA XÁC ĐỊNH" (undefined).

**Bước 2: Duyệt tập dữ liệu**:

* Thêm các điểm trong lân cận (ε-neighborhood). của p vào danh sách thứ tự (**OrderSeeds**) nếu chưa được xử lý.
* Sắp xếp các điểm trong danh sách dựa trên khoảng cách tiếp cận nhỏ nhất.

**Bước 3: Cập nhật danh sách OrderSeeds**:

* Với các điểm lân cận, tính khoảng cách tiếp cận mới và cập nhật giá trị nếu khoảng cách này nhỏ hơn khoảng cách đã biết.

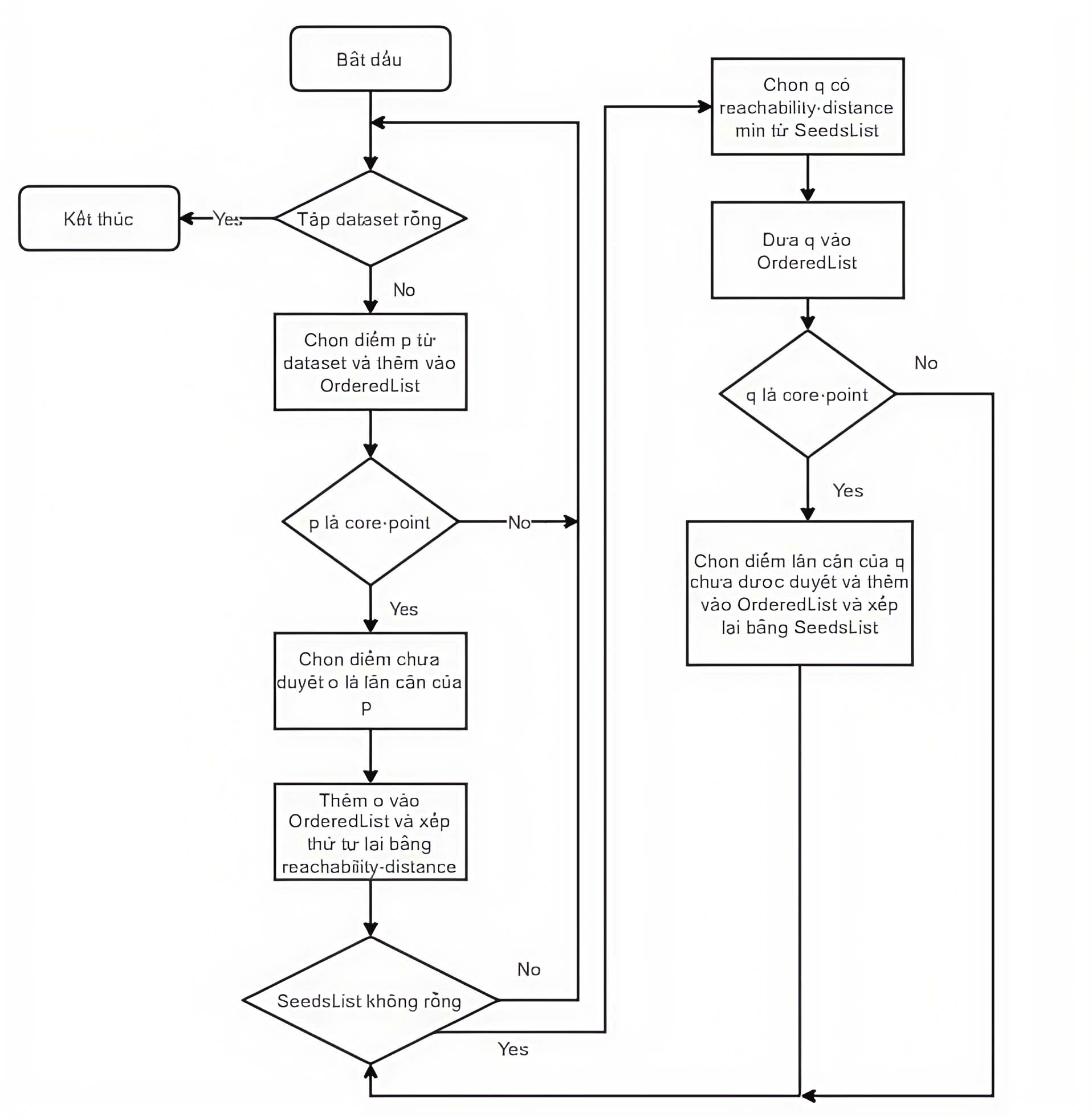
**Bước 4: Lặp lại với hàng đợi:**

* Duyệt qua từng điểm trong Seeds, tiếp tục cập nhật và xử lý các điểm cho đến khi hàng đợi rỗng.

**Bước 5 : Xuất kết quả**:

* Xây dựng biểu đồ khả năng tiếp cận.
* Trích xuất các cụm từ biểu đồ bằng cách xác định các đoạn có giá trị thấp (biểu thị cụm).

### 3.4. Lưu đồ thuật toán



1. Khởi tạo

* Mục đích: Bắt đầu với tập dữ liệu và thiết lập các tham số.
* Hoạt động:
  + Nhập tập dữ liệu D.
  + Đặt các tham số:
    - ϵ: Bán kính tìm lân cận.
    - MinPts: Số lượng điểm tối thiểu để xác định một cụm.
  + Tạo danh sách trống để lưu trữ thứ tự của các điểm đã xử lý (Cluster Order).
  + Đánh dấu tất cả các điểm là "chưa được xử lý".

2. Lặp qua từng điểm

* Mục đích: Xử lý từng điểm trong dữ liệu.
* Hoạt động:
  + Chọn một điểm chưa được xử lý p.
  + Kiểm tra các điểm lân cận trong phạm vi ϵ:
    - Nếu số điểm lân cận ≥MinPts, xác định p là core point (điểm lõi).
    - Nếu không, điểm p không phải là điểm lõi.
  + Thêm p vào danh sách Cluster Order.

3. Tính toán core distance

* Mục đích: Xác định độ mật độ của điểm p.
* Hoạt động:
  + Core Distance:
    - Nếu p là điểm lõi, tính khoảng cách từ p tới điểm xa nhất trong MinPts điểm lân cận gần nhất.
    - Nếu p không phải là điểm lõi, không có core distance.

4. Tính toán reachability distance

* Mục đích: Đánh giá khả năng tiếp cận của các điểm lân cận từ điểm lõi.
* Hoạt động:
  + Với mỗi điểm q trong vùng lân cận của p, tính:

Reachability Distance(p,q)=max⁡(Core Distance(p),Distance(p,q))

* + Giá trị này giúp xác định điểm nào có thể được kết nối với p dựa trên mật độ.

5. Sắp xếp các điểm theo mật độ

* Mục đích: Tạo thứ tự các điểm theo mức độ mật độ.
* Hoạt động:
  + Đánh dấu điểm p là "đã xử lý".
  + Thêm các điểm lân cận của p vào hàng đợi (queue), sắp xếp theo thứ tự tăng dần của reachability distance.
  + Lặp lại quá trình cho các điểm trong hàng đợi cho đến khi tất cả các điểm trong cụm được xử lý.

6. Phân cụm

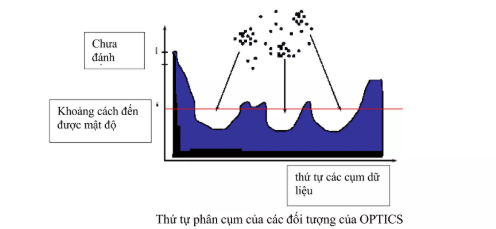
* Mục đích: Xác định các cụm và vùng nhiễu.
* Hoạt động:
  + Dựa trên danh sách Cluster Order và reachability distance, sử dụng Reachability Plot để:
    - Tìm các đoạn liên tục có reachability distance thấp (các cụm).
    - Các điểm có reachability distance cao được xem là nhiễu.

7. Kết thúc

* Mục đích: Hoàn thành quá trình phân cụm.
* Hoạt động:
  + Xuất danh sách Cluster Order.
  + Xuất Reachability Plot để phân tích hoặc trực quan hóa.

## 

### 3.5 Minh họa thuật toán



***Hình ảnh. Thí dụ PCDL của thuật toán OPTICS***

### 3.6 Điều kiện dừng và chất lượng phân cụm

* **Điều kiện dừng của OPTICS:**Thuật toán dừng khi tất cả các điểm trong cơ sở dữ liệu đã được xử lý. Đặc biệt, OPTICS không yêu cầu người dùng chỉ định số lượng cụm hay ngưỡng mật độ cụ thể, mà chỉ tạo ra một thứ tự phân cụm theo mức độ tiếp cận của các điểm.
* **Chất lượng phân cụm:**Chất lượng phân cụm trong OPTICS có thể được đánh giá bằng cách phân tích thứ tự phân cụm và biểu đồ "reachability plot". Các cụm có thể được xác định dựa trên sự thay đổi đột ngột của khoảng cách tiếp cận.

### 3.7. Các tham số và ảnh hưởng

1. Phân tích tham số ϵ (Epsilon):

* **Vai trò:** ϵ xác định bán kính lân cận trong không gian dữ liệu để kiểm tra mật độ của các điểm xung quanh.
* **Ảnh hưởng khi thay đổi giá trị:**
  + **Giá trị nhỏ:**
    - Phát hiện cụm nhỏ hoặc các cụm có mật độ cao một cách chính xác.
    - Có nguy cơ bỏ sót các cụm lớn hơn hoặc không phát hiện được cụm trong dữ liệu thưa.
  + **Giá trị lớn:**
    - Dễ dàng gộp các cụm gần nhau thành một cụm duy nhất, dẫn đến mất chi tiết trong phân cụm.
    - Có thể khiến các điểm nhiễu bị nhận diện nhầm thành một phần của cụm.

2. Phân tích tham số MinPts (Minimum Points):

* **Vai trò:** MinPts là số lượng điểm tối thiểu trong vùng bán kính ϵ\epsilonϵ để xác định một cụm.
* **Ảnh hưởng khi thay đổi giá trị:**
  + **Giá trị cao:**
    - Chỉ nhận diện các cụm lớn và dày đặc.
    - Dữ liệu thưa hoặc cụm nhỏ có thể bị bỏ sót, dẫn đến nhiều điểm bị coi là nhiễu.
  + **Giá trị thấp:**
    - Nhạy hơn với cụm nhỏ, giúp nhận diện cả các cụm có mật độ thấp.
    - Dễ gây nhầm lẫn khi các điểm nhiễu bị nhận diện nhầm thành cụm.

3.Mối quan hệ giữa ϵ và MinPts:

* Hai tham số này cần được cân chỉnh phù hợp với đặc điểm của dữ liệu:
  + ϵ quá lớn cùng với MinPts thấp có thể gây ra hiện tượng “quá khớp,” gộp các điểm không liên quan vào cụm.
  + ϵ nhỏ cùng với MinPts cao có thể dẫn đến việc bỏ sót cụm và nhận diện quá nhiều điểm nhiễu.
* Sự kết hợp hợp lý giữa ϵ và MinPts là yếu tố quyết định hiệu quả phân cụm, thường được xác định thông qua thử nghiệm trên tập dữ liệu thực tế.

### 3.8. Nhận xét thuật toán OPTICS

- OPTICS có độ phức tạp: O(nLogn) (n là kích thước của tập DL)

**- Ưu điểm của OPTICS:**

1. **Không cần ngưỡng mật độ cố định:** Thuật toán có thể phát hiện các cụm với mật độ khác nhau mà không cần tham số epsilon cố định.
2. **Xử lý các cụm có hình dạng phức tạp:** OPTICS có thể phát hiện các cụm có hình dạng bất thường.
3. **Khả năng xác định cấu trúc phân cụm linh hoạt:** Tạo ra một thứ tự phân cụm có thể giúp xem xét các cấu trúc phân cụm ở các mức độ chi tiết khác nhau.
4. Cho ra kết quả tốt và thực thi hiệu quả trên nhiều tập dữ liệu

**- Nhược điểm của OPTICS:**

1. **Phức tạp tính toán:** Dù thuật toán không yêu cầu tái chạy nhiều lần, nhưng tính toán có thể tốn kém, đặc biệt với tập dữ liệu lớn.
2. **Khó đánh giá trực tiếp:** Không có một cách rõ ràng để xác định số lượng cụm như trong DBSCAN, do thuật toán chỉ cung cấp thứ tự phân cụm.
3. Có vài đối tượng biên có thể bị thiếu khi trích rút cụm nếu chúng được xử lý bởi OPTICS trước khi một đối tượng nhân của cụm tương ứng được tìm thấy.

### 3.9. So sánh với các thuật toán khác

### 1. So sánh với DBSCAN

* **Điểm tương đồng:**
  + Cả hai thuật toán đều dựa trên khái niệm mật độ, sử dụng chung các tham số ϵ\epsilonϵ (bán kính lân cận) và MinPts (số điểm tối thiểu để xác định cụm).
  + Đều có khả năng phát hiện điểm nhiễu và không yêu cầu số cụm đầu vào.
* **Điểm khác biệt:**
  + DBSCAN cố định cụm dựa trên một giá trị ϵ\epsilonϵ, dẫn đến hạn chế khi các cụm có mật độ khác nhau.
  + OPTICS không cố định cụm mà sắp xếp các điểm dựa trên mật độ lân cận, từ đó xử lý tốt hơn với các cụm có mật độ thay đổi hoặc hình dạng phức tạp.

### 2. So sánh với K-means

* **Điểm tương đồng:**
  + Cả hai đều là phương pháp phân cụm dữ liệu, tập trung vào việc nhóm các điểm gần nhau thành cụm.
* **Điểm khác biệt:**
  + **Số cụm:**
    - K-means yêu cầu xác định trước số lượng cụm (kkk), trong khi OPTICS tự động khám phá số lượng cụm dựa trên mật độ.
  + **Hình dạng cụm:**
    - K-means phù hợp với cụm hình cầu và các cụm có kích thước tương đương, nhưng không xử lý tốt cụm có hình dạng phức tạp.
    - OPTICS linh hoạt hơn, dễ dàng nhận diện cụm với hình dạng bất kỳ và mật độ thay đổi.
  + **Nhạy cảm với điểm nhiễu:**
    - K-means dễ bị ảnh hưởng bởi điểm nhiễu hoặc giá trị ngoại lai, trong khi OPTICS có khả năng phát hiện và loại bỏ nhiễu.

### 3. So sánh với Hierarchical Clustering

* **Điểm tương đồng:**
  + Cả hai đều không yêu cầu xác định trước số cụm.
  + Đều cung cấp một cách nhìn tổng quát về cấu trúc dữ liệu.
* **Điểm khác biệt:**
  + Hierarchical Clustering tạo cây phân cấp (dendrogram) để phân cụm dữ liệu ở các cấp độ khác nhau, còn OPTICS tạo ra cấu trúc cụm dựa trên mật độ theo thứ tự sắp xếp điểm.
  + OPTICS xử lý dữ liệu lớn hiệu quả hơn, trong khi Hierarchical Clustering thường yêu cầu nhiều bộ nhớ và thời gian tính toán khi kích thước dữ liệu tăng.
  + Hierarchical Clustering không có cơ chế phát hiện điểm nhiễu rõ ràng, còn OPTICS có khả năng nhận diện các điểm nhiễu dựa trên khoảng cách mật độ.

# **Chương 2 : Định nghĩa bài toán và các phương pháp giải quyết bài toán**

# **I. Định nghĩa bài toán**

## 1. Lí do chọn đề tài:

* Tính ứng dụng cao và thực tiễn trong ngành giải trí số:
* Sự bùng nổ của các nền tảng phát trực tuyến đã khiến cá nhân hóa nội dung trở thành yếu tố quan trọng giúp duy trì sự quan tâm của người dùng và tạo lợi thế cạnh tranh.
* Phân tích cụm đóng vai trò quan trọng trong việc hiểu rõ sở thích của khách hàng, từ đó xây dựng chiến lược tiếp thị phù hợp, nâng cao trải nghiệm người dùng và tối ưu hóa hiệu quả kinh doanh.
* Thúc đẩy đổi mới và cá nhân hóa trong kinh doanh:
* Đề tài không chỉ tập trung vào ngành giải trí mà còn mở rộng ra các lĩnh vực khác như thương mại điện tử, giáo dục, và y tế, mở ra cơ hội khám phá các giải pháp công nghệ tiên tiến.
* Thông qua nghiên cứu, sinh viên có thể áp dụng các phương pháp phân tích dữ liệu như phân cụm vào thực tế, rèn luyện kỹ năng tư duy phân tích và đóng góp vào việc thúc đẩy xu hướng cá nhân hóa trong kỷ nguyên số hóa.

## 2. Tổng quan bài toán

Dataset bao gồm 22 thuộc tính (features), phần lớn là các đặc tính về người dùng và các thể loại phim mà họ yêu thích. Mỗi dòng trong dataset đại diện cho hành vi của một người dùng đối với một bộ phim cụ thể, và Rating là đánh giá của người dùng đối với bộ phim đó.

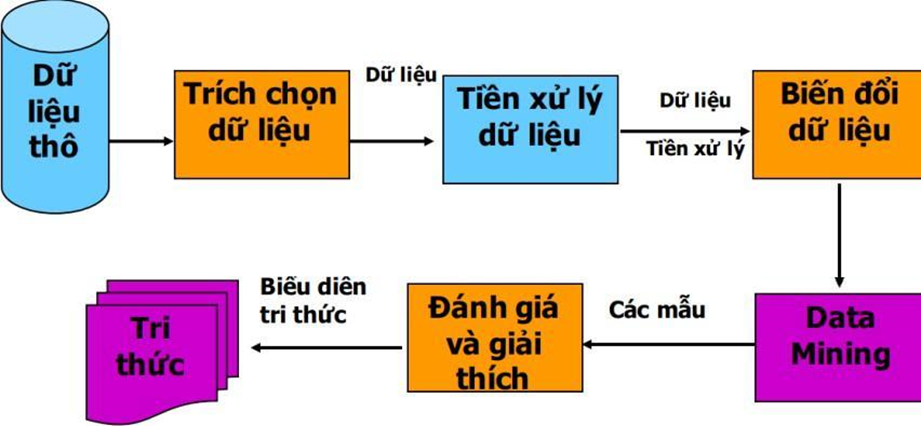
* Rating: Mức đánh giá của người dùng đối với bộ phim (giá trị từ 1 đến 5).
* Gender: Giới tính của người dùng (0 hoặc 1).
* Age: Độ tuổi của người dùng.
* Occupation: Nghề nghiệp của người dùng.
* War, Horror, Musical, Crime, Mystery, Film-Noir, Animation, Thriller, Documentary, Comedy, Fantasy, Western, Adventure, Action, Sci-Fi, Romance, Childrens, Drama: Các thể loại phim mà người dùng yêu thích. Mỗi thể loại là một thuộc tính nhị phân (0 hoặc 1), trong đó "1" có nghĩa là người dùng yêu thích thể loại phim đó, và "0" có nghĩa là họ không quan tâm đến thể loại đó.

Input:

* Các thuộc tính về người dùng: Gender, Age, Occupation.
* Các thể loại phim mà người dùng yêu thích: War, Horror, Musical, Crime, Mystery, Film-Noir, Animation, Thriller, Documentary, Comedy, Fantasy, Western, Adventure, Action, Sci-Fi, Romance, Childrens, Drama.
* Rating: Mức độ yêu thích của người dùng đối với bộ phim (giá trị từ 1 đến 5).

Output: Kết quả phân cụm dựa trên Rating của người dùng.

## 3. Quy trình thực hiện



Quy trình thực hiện khai phá bao gồm 6 bước:

- Bước 1: Tạo tập tin dữ liệu đầu vào

- Bước 2: Tiền xử lý, làm sạch tập dữ liệu

- Bước 3: Chọn tác vụ khai phá dữ liệu (phân lớp)

- Bước 4: Khai phá dữ liệu: tìm kiếm tri thức

- Bước 5: Đánh giá mẫu tìm được

- Bước 6: Biểu diễn tri thức

Ở trong bài toán này chúng em đã tóm tắt các bước thành những mục sau:

* Thu thập dữ liệu
* Tiền xử lý dữ liệu
* Chọn tác vụ khai phá dữ liệu ( phân cụm)
* Khai phá dữ liệu: tìm kiếm tri thức
* Biểu diễn tri thức
* Đánh giá kết quả

# **II.Các phương pháp giải quyết bài toán**

## **1.Thuật toán CLARANS**

### **1. Chuẩn bị dữ liệu**

Xử lý dữ liệu đảm bảo phù hợp với đầu vào của thuật toán:

1.1. Chuẩn hóa giá trị số

* Áp dụng chuẩn hóa Z-score (Standardization). Phương pháp này đưa các giá trị của đặc trưng (feature) về dạng có trung bình (mean) bằng 0 và độ lệch chuẩn (standard deviation) bằng 1

1.2. Chọn đặc trưng phù hợp

* **Lọc đặc trưng quan trọng**: Giữ lại các cột như Rating, các thể loại phim (War, Horror, Drama,...) và Age để phân cụm dựa trên sở thích người dùng.
* **Loại bỏ thông tin không cần thiết**: Ví dụ, thông tin như Gender có thể không cần thiết trong bài toán này.

1.3. Xử lý dữ liệu danh mục

* **Gender**: Dữ liệu danh mục như Gender đã được mã hóa dạng nhị phân (0, 1), không cần xử lý thêm.
* **Thể loại phim**: Các cột thể loại phim giữ nguyên dưới dạng nhị phân (0 hoặc 1) để đánh dấu cho một phim có thuộc thể loại tương ứng hay không.

### **2. Áp dụng thuật toán Clarans**

2.1. Xác định tham số

* **K** : số cụm được phân ra. Ví dụ: k = 5
* **num\_local:** số lần lặp lại thuật toán, sau mỗi lần lặp sẽ giữ lại các medoids đại diện của lần chạy thuật toán trước đó để so sánh để xem giá trị nào tốt hơn. Ví dụ: num\_local = 2
* **max\_neighbor**: số hàng xóm tối đa trong mỗi lần tìm kiếm cục bộ. Ví dụ: k = 10

2.2. Thực hiện phân cụm

1. **Lấy ngẫu nhiên k điểm làm medoids khởi đầu**:
   * Chọn ngẫu nhiên k điểm làm medoids (tâm cụm) khởi tạo, tiến hành tính toán và nhóm dữ liệu vào các medoids gần nó nhất, ta được các cụm khởi đầu, đồng thời tính toán được chi phí phân cụm.
2. **Duyệt cho đến max\_neighbor để tìm ra medoids tốt nhất**:
   * Duyệt ngẫu nhiên k các điểm trong bộ dữ liệu, tính toán chi phí và so sánh với chi phí phân cụm tốt nhất trước đó.
     1. Nếu chi phí giảm, cập nhật medoids mới.
     2. Nếu chi phí tăng hoặc không đổi, giữ nguyên medoids cũ.
3. **Đưa ra kết quả phân cụm:**
   * Sau num\_local vòng lặp, ta tìm được medoids tối ưu nhất và các cụm tương ứng kèm theo.

## **2.Thuật toán BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)**

Thuật toán BIRCH sẽ phân cụm người dùng dựa trên hành vi xem phim của họ bằng cách sử dụng một cấu trúc dữ liệu gọi là CF-Tree (Clustering Feature Tree).

* Bước 1: Xây dựng CF-Tree
  + Mỗi nút trong cây CF sẽ chứa thông tin về một cụm dữ liệu, bao gồm:
    - n là số đối tượng trong cụm (Số lượng người dùng trong cụm)
    - LS là tổng các giá trị thuộc tính của các đối tượng trong cụm
    - SS là tổng bình phương của các giá trị thuộc tính của các đối tượng trong cụm.
  + Quá trình xây dựng CF-Tree sẽ thực hiện việc phân chia người dùng vào các cụm. Nếu một điểm dữ liệu không thể được chèn vào bất kỳ cụm nào mà không vượt quá giới hạn dung lượng, thì nó sẽ tạo ra một cụm mới.
* Bước 2: Giảm kích thước dữ liệu
  + CF-Tree cho phép giảm thiểu số lượng cụm cần thiết bằng cách sử dụng kỹ thuật phân nhóm phân cấp (hierarchical clustering). Điều này giúp giảm số lượng trung tâm cụm và giúp dễ dàng xác định các nhóm người dùng có hành vi tương tự nhau.
* Bước 3: Duy trì cấu trúc CF-Tree
  + Thuật toán tiếp tục cập nhật cấu trúc cây khi các điểm dữ liệu mới (người dùng mới) được thêm vào hoặc khi các nhóm người dùng thay đổi hành vi xem phim.
* Bước 4: Định nghĩa các cụm người dùng
  + Cuối cùng, các nhóm người dùng sẽ được phân thành các cụm. Mỗi cụm sẽ chứa các người dùng có hành vi xem phim giống nhau, ví dụ: một nhóm người dùng yêu thích hài lãng mạn, nhóm người yêu thích kinh dị, nhóm người yêu thích phim hành động.

## **3.Thuật toán OPTICS**

### **1. Chuẩn bị dữ liệu**

Để áp dụng thuật toán OPTICS, trước tiên cần xử lý dữ liệu đảm bảo phù hợp với đầu vào của thuật toán:

1.1. Chuẩn hóa giá trị số

* Các đặc trưng như Rating, Age được chuẩn hóa về khoảng [0,1] bằng phương pháp Min-Max Scaling để đảm bảo rằng các giá trị không làm ảnh hưởng đến khoảng cách mật độ.

1.2. Chọn đặc trưng phù hợp

* **Lọc đặc trưng quan trọng**: Giữ lại các cột như Rating, các thể loại phim (War, Horror, Drama,...) và Age để phân cụm dựa trên sở thích người dùng.
* **Loại bỏ thông tin không cần thiết**: Ví dụ, thông tin như Occupation có thể không cần thiết trong bài toán này.

1.3. Xử lý dữ liệu danh mục

* **Gender**: Dữ liệu danh mục như Gender đã được mã hóa dạng nhị phân (0, 1), không cần xử lý thêm.
* **Thể loại phim**: Các cột thể loại phim giữ nguyên dưới dạng nhị phân (0 hoặc 1) để đại diện cho sở thích của người dùng.

### **2. Áp dụng thuật toán OPTICS**

Khác với K-means, OPTICS không yêu cầu biết trước số cụm. Các bước cụ thể như sau:

2.1. Xác định tham số

* **MinPts**: Số lượng điểm tối thiểu cần có trong vùng lân cận để xác định một điểm lõi. Giá trị này thường được chọn dựa trên ngữ cảnh dữ liệu, ví dụ: MinPts=5
* **ε**: Bán kính tìm kiếm các điểm lân cận. OPTICS không yêu cầu cố định ε, nhưng một giá trị tối đa εmax ​ có thể được đặt để giảm thời gian tính toán.

2.2. Thực hiện phân cụm

**Tính toán khoảng cách và mật độ**:

* + Xác định khoảng cách tiếp cận (*Reachability Distance*) và khoảng cách lõi (*Core Distance*) cho từng điểm.

**Sắp xếp thứ tự điểm**:

* + Dữ liệu được sắp xếp theo thứ tự mật độ, hình thành biểu đồ Reachability Plot.

**Xác định cụm từ đồ thị Reachability**:

* + Phát hiện các vùng có mật độ cao dựa trên biến động của khoảng cách tiếp cận (tạo thành các cụm).

2.3. Diễn giải cụm

Từ kết quả OPTICS, ta sẽ thu được các cụm có mật độ tương đồng. Ví dụ:

* **Cụm 1**: Người dùng thích phim hành động và khoa học viễn tưởng.
* **Cụm 2**: Người dùng yêu thích phim tình cảm và hài.
* **Cụm 3**: Người dùng thường xem phim tài liệu và chính kịch.

### **3. Triển khai quảng cáo cá nhân hóa**

Dựa trên kết quả phân cụm từ thuật toán OPTICS:

3.1. Phân tích cụm

* **Cụm 1**: Người dùng có sở thích với phim hành động và khoa học viễn tưởng.
  + **Chiến dịch**: Gợi ý các bộ phim hành động sắp ra mắt hoặc giới thiệu phim khoa học viễn tưởng có đánh giá cao.
* **Cụm 2**: Người yêu thích phim lãng mạn và hài.
  + **Chiến dịch**: Gợi ý các bộ phim hài nhẹ nhàng hoặc tình cảm nổi bật trên nền tảng.
* **Cụm 3**: Người dùng quan tâm đến phim tài liệu và chính kịch.
  + **Chiến dịch**: Đề xuất các bộ phim tài liệu chất lượng cao về các chủ đề như lịch sử, khoa học, hoặc xã hội.

3.2. Tăng trải nghiệm người dùng

* Hiển thị danh mục phim theo thứ tự ưu tiên dựa trên cụm của người dùng.
* Tạo email marketing hoặc thông báo đẩy (*push notification*) với nội dung gợi ý cá nhân hóa.

### **4. Ưu điểm của việc dùng OPTICS cho bài toán này**

**So với K-means**

1. **Không cần xác định số cụm trước**: OPTICS tự động phát hiện cụm dựa trên mật độ.
2. **Xử lý tốt cụm có hình dạng bất thường**: Thích hợp với dữ liệu phức tạp, không đồng nhất.
3. **Xử lý nhiễu hiệu quả**: OPTICS có thể xác định các điểm nhiễu và loại bỏ chúng khỏi cụm.

**Hạn chế**

* Thuật toán OPTICS có thể chậm hơn K-means với tập dữ liệu lớn, do tính toán khoảng cách mật độ cho từng điểm.

### **5. Kết luận**

Sử dụng OPTICS trong bài toán phân cụm gợi ý nội dung cá nhân hóa mang lại khả năng phát hiện cụm, đặc biệt với dữ liệu phức tạp. Kết quả giúp doanh nghiệp triển khai quảng cáo và gợi ý nội dung phù hợp, nâng cao trải nghiệm người dùng và tối ưu hóa chiến lược tiếp thị.

# **Chương 3: Thực nghiệm so sánh đánh giá kết quả**

# **I. Thu thập dữ liệu**

### **1. Mô tả dữ liệu (Dataset Description):**

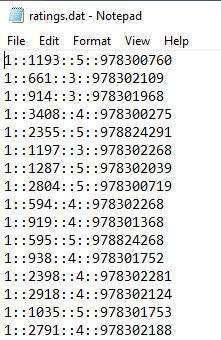
* **MovieLens 1M Dataset** chứa 1.000.209 đánh giá ẩn danh của khoảng 3.900 bộ phim do 6.040 người dùng MovieLens tham gia MovieLens vào năm 2000 thực hiện. Dữ liệu bao gồm ba tập tin chính:
  + **movies.dat**: Chứa thông tin về các bộ phim, bao gồm ID phim, tên phim và thể loại của phim.
  + **ratings.dat**: Chứa thông tin về các đánh giá phim từ người dùng, bao gồm ID người dùng, ID phim, đánh giá và thời gian đánh giá.
  + **users.dat**: Chứa thông tin về người dùng, bao gồm ID người dùng, giới tính, độ tuổi, nghề nghiệp, và mã bưu chính.

### **2. Các trường dữ liệu trong từng tập tin:**

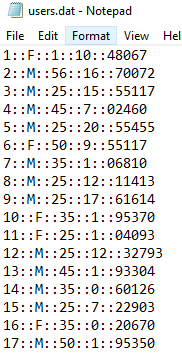
* **movies.dat:**

****

* + movieId: ID của bộ phim.
  + title: Tên bộ phim.
  + genres: Thể loại phim, có thể là nhiều thể loại phân cách bằng dấu "|".
* **ratings.dat:**

****

* + userId: ID của người dùng.
  + movieId: ID của bộ phim.
  + rating: Điểm đánh giá của người dùng, từ 1 đến 5.
  + timestamp: Thời gian đánh giá, tính theo giây kể từ ngày 01/01/1970.
* **users.dat:**

****

* + userId: ID của người dùng.
  + gender: Giới tính người dùng (M hoặc F).
  + age: Độ tuổi người dùng.
  + occupation: Nghề nghiệp của người dùng.
  + zip code: Mã bưu chính của người dùng.

### **3. Quy trình thu thập và làm sạch dữ liệu:**

* **Bước 1**: Duyệt qua các tập tin dữ liệu (movies.dat, ratings.dat, users.dat).
* **Bước 2:** Nối các tập tin dữ liệu lại với nhau qua việc sử dụng khóa ngoại: userId và movieId.
* **Bước 3**: Làm sạch dữ liệu bằng cách xử lý các giá trị thiếu (missing values), loại bỏ các bản ghi không hợp lệ.
* **Bước 4**: Tiến hành chuẩn hóa các cột dữ liệu, ví dụ như chuyển đổi các thể loại phim thành danh sách các thể loại có cấu trúc rõ ràng.
* **Bước 5**: Phân tích các mối quan hệ giữa các cột dữ liệu, ví dụ như phân tích mối quan hệ giữa điểm số đánh giá và thể loại phim.

### **4. Các câu hỏi phân tích có thể thực hiện:**

* **Phân tích thói quen người dùng:** Người dùng có xu hướng đánh giá cao các thể loại phim nào? (Phân tích điểm đánh giá theo thể loại).
* **Phân tích theo độ tuổi và giới tính:** Mối quan hệ giữa độ tuổi, giới tính và sở thích phim.
* **Đề xuất phim:** Sử dụng các phương pháp học máy để đề xuất phim cho người dùng dựa trên các đánh giá trước đó.
* **Phân tích nghề nghiệp:** Các nghề nghiệp có liên quan đến thể loại phim yêu thích hoặc điểm đánh giá.

Khi thu thập dữ liệu này, mục tiêu là phân tích hành vi của người dùng về sở thích phim, tạo hệ thống đề xuất nghiên cứu thói quen xem phim của các nhóm người dùng khác nhau.

# **II. Tìm hiểu tập dữ liệu**

## **1. Đặc trưng về người dùng**

1.1. Rating (Đánh giá phim)

* Điểm số người dùng đưa ra cho mỗi bộ phim.
* **Kiểu dữ liệu**: Số thực (float)
* **Ý nghĩa**: Thể hiện mức độ yêu thích của người dùng đối với bộ phim.
* **Ứng dụng**: Là một trong những yếu tố quan trọng giúp nhóm người dùng có sở thích tương đồng.

1.2. Gender (Giới tính)

* Thông tin về giới tính của người dùng.
* **Kiểu dữ liệu**: Chuỗi kí tự (String) bao gồm 2 giới tính nam và nữ.
* **Ý nghĩa**: Có thể ảnh hưởng đến sở thích phim, ví dụ: phim hành động có thể phổ biến hơn với nam giới.

1.3. Age (Tuổi)

* Tuổi của người dùng.
* **Kiểu dữ liệu**: Số nguyên (integer)

1: "Dưới 18 tuổi"

18: "18-24"

25: "25-34"

35: "35-44"

45: "45-49"

50: "50-55"

56: "56+"

* **Ý nghĩa**: Người dùng thuộc các độ tuổi khác nhau thường có sở thích phim khác nhau, ví dụ: người trẻ tuổi thích phim hành động, khoa học viễn tưởng; người lớn tuổi thích phim chính kịch, tài liệu.

1.4. Occupation (Nghề nghiệp)

* Nghề nghiệp của người dùng.
* **Kiểu dữ liệu**: Danh mục (Category).

o Học thuật/giáo dục

o Nghệ sĩ

o Nhân viên văn phòng/hành chính

o Sinh viên cao đẳng/sau đại học

o Dịch vụ khách hàng

o Bác sĩ/chăm sóc sức khỏe

o Giám đốc điều hành/quản lý

o Nông dân

o Nội trợ

o Học sinh K-12

o Luật sư

o Lập trình viên

o Đã nghỉ hưu

o Bán hàng/tiếp thị

o Nhà khoa học

o Tự kinh doanh

o Kỹ thuật viên/kỹ sư

o Thợ thủ công/thợ thủ công

o Thất nghiệp

o Nhà văn

* **Ý nghĩa**: Nghề nghiệp có thể gợi ý về thời gian rảnh rỗi và loại nội dung phù hợp, ví dụ: sinh viên có thể thích phim giải trí hoặc hành động.

## **2. Đặc trưng về phim**

2.1. Genres (Thể loại phim)

* Các thể loại mà bộ phim thuộc về.
* **Kiểu dữ liệu**: Danh mục (Category). Bao gồm các thể loại sau:

o War

o Horror

o Musical

o Crime

o Mystery

o Film-Noir

o Animation

o Thriller

o Documentary

o Comedy

o Fantasy

o Western

o Adventure

o Action

o Sci-Fi

o Romance

o Childrens

o Drama

* **Ý nghĩa**: Các thể loại giúp xác định sở thích của người dùng dựa trên lịch sử xem phim.

## **3. Ý nghĩa của dữ liệu**

Dữ liệu này giúp trả lời các câu hỏi sau:

* Người dùng thuộc nhóm nào có xu hướng thích thể loại phim nào?
* Độ tuổi và giới tính ảnh hưởng như thế nào đến sở thích phim?
* Dựa trên các nhóm (cụm) được phân tích, có thể cá nhân hóa gợi ý phim phù hợp hơn.

# **III. Tiền xử lý dữ liệu**

## **1. Xử lý và chuẩn bị dữ liệu**

### 1.1. Làm sạch dữ liệu

* Loại bỏ các bản ghi không đầy đủ hoặc lỗi (ví dụ: người dùng không có đánh giá hoặc lịch sử xem phim).
* Xử lý dữ liệu thiếu: Điền giá trị trung bình hoặc các kỹ thuật khác.

### 1.2. Chuyển đổi dữ liệu

* Mã hóa các đặc trưng danh mục (như Nghề nghiệp hoặc Thể loại phim) thành số hoặc dạng nhị phân.
* Chuẩn hóa các đặc trưng số và chữ.

### 1.3. Kết hợp dữ liệu

* Kết nối dữ liệu người dùng và dữ liệu phim dựa trên các đặc trưng như Lịch sử xem phim hoặc Đánh giá phim.

## **2. Đặc trưng về người dùng**

### 2.1. Rating (Đánh giá phim)

* Điểm số người dùng đưa ra cho mỗi bộ phim.
* **Kiểu dữ liệu**: Số thực (float), nằm trong khoảng từ 1 đến 5
* **Ý nghĩa**: Thể hiện mức độ yêu thích của người dùng đối với bộ phim.
* **Ứng dụng**: Là một trong những yếu tố quan trọng giúp nhóm người dùng có sở thích tương đồng.

### 2.2. Gender (Giới tính)

* Thông tin về giới tính của người dùng.
* **Kiểu dữ liệu**: Danh mục nhị phân (1: Nam, 0: Nữ).
* **Ý nghĩa**: Có thể ảnh hưởng đến sở thích phim, ví dụ: phim hành động có thể phổ biến hơn với nam giới.

### 2.3. Age (Tuổi)

* Tuổi của người dùng.
* **Kiểu dữ liệu**: Danh mục (Category) gồm 4 mục sau:
* 1: Nhỏ hơn 18
* 2: 18-34
* 3: 35-54
* 4: 55+
* **Ý nghĩa**: Người dùng thuộc các độ tuổi khác nhau thường có sở thích phim khác nhau, ví dụ: người trẻ tuổi thích phim hành động, khoa học viễn tưởng; người lớn tuổi thích phim chính kịch, tài liệu.

### 2.4. Occupation (Nghề nghiệp)

* Nghề nghiệp của người dùng.
* **Kiểu dữ liệu**: Danh mục (Category) gồm 21 danh mục được kí hiệu như sau :

0: "khác" hoặc không xác định

1: "học thuật/giáo dục"

2: "nghệ sĩ"

3: "nhân viên văn phòng/hành chính"

4: "sinh viên cao đẳng/sau đại học"

5: "dịch vụ khách hàng"

6: "bác sĩ/chăm sóc sức khỏe"

7: "giám đốc điều hành/quản lý"

8: "nông dân"

9: "nội trợ"

10: "học sinh K-12"

11: "luật sư"

12: "lập trình viên"

13: "đã nghỉ hưu"

14: "bán hàng/tiếp thị"

15: "nhà khoa học"

16: "tự kinh doanh"

17: "kỹ thuật viên/kỹ sư"

18: "thợ thủ công/thợ thủ công"

19: "thất nghiệp"

20: "nhà văn"

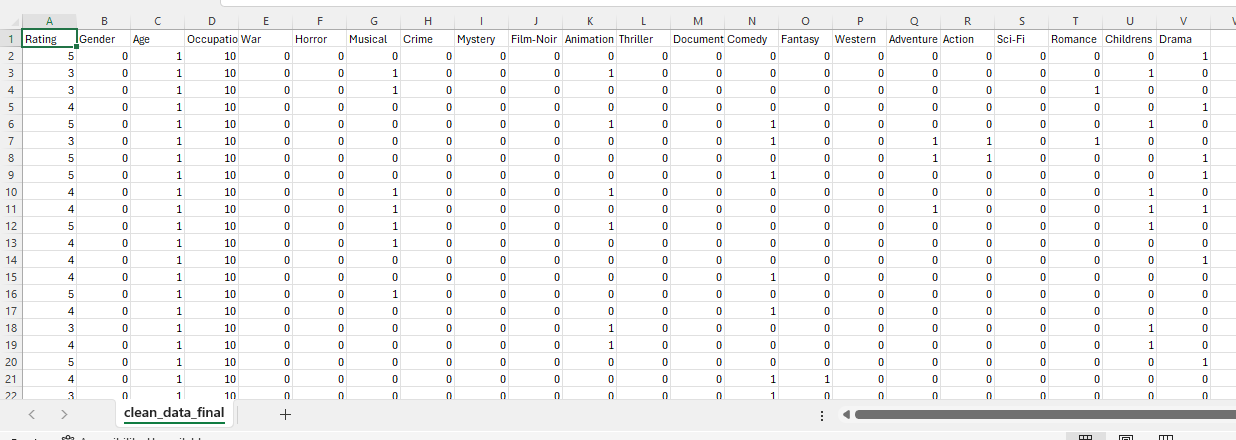
* **Ý nghĩa**: Nghề nghiệp có thể gợi ý về thời gian rảnh rỗi và loại nội dung phù hợp, ví dụ: sinh viên có thể thích phim giải trí hoặc hành động.

## **3. Đặc trưng về phim**

### 3.1. Genres (Thể loại phim)

* Các thể loại mà bộ phim thuộc về.
* **Kiểu dữ liệu**: Danh mục nhị phân nếu có thì đánh dấu là 1, ngược lại là 0.
* **Ý nghĩa**: Các thể loại giúp xác định sở thích của người dùng dựa trên lịch sử xem phim.

=> Sau khi tiền xử lý dữ liệu xong, ta thu được file dữ liệu: *clean\_data\_final.csv*

**

# **IV. Cài đặt thực nghiệm trên các phương pháp đã chọn**

## **1.Thuật toán CLARANS**

* Các thư viện sử dụng:
* Thư viện pandas: Đọc dữ liệu từ tệp CSV chuyển dữ liệu thành dạng bảng (DataFrame) và thực hiện các thao tác như loại bỏ các cột không cần thiết và lấy mẫu dữ liệu.
* Thư viện numpy: Xử lý các phép toán với mảng numpy như tính toán khoảng cách giữa các cụm hoặc thao tác với mảng trong các chỉ số đánh giá.
* Thư viện matplotlib: vẽ đồ thị, biểu đồ trong Python. Pylot là module con của thư viện này, chuyên dùng để tạo biểu đồ trực quan.
* Thư viện sklearn: hỗ trợ chuẩn hóa dữ liệu, giảm chiều, phân cụm và đánh giá chất lượng phân cụm.
* Thư viện pyclustering: hỗ trợ chạy thuật toán Clarans
* Các thư viện khác như os, datetime: hỗ trợ lưu lại các chỉ số đánh giá và biểu đồ phân cụm
* Các bước thực hiện:
* Bước 1: Đọc tệp dữ liệu và kiểm tra dữ liệu mẫu bằng cách in ra các giá trị:

# Đọc dữ liệu

df = pd.read\_csv("clean\_data\_final.csv")

data = df.drop(columns = ['Gender']).head(500).values

#data = df[['Occupation','Rating','Age']].head(5000).values

print(data[:20])

* Bước 2: Chuẩn hóa dữ liệu

#Hàm chuẩn hóa dữ liệu

def preprocess\_data(data):

scaler = StandardScaler()

return scaler.fit\_transform(data)

#Chuẩn hóa dữ liệu:

scaler\_data = preprocess\_data(data)

* Bước 3: Giảm chiều dữ liệu

#Hàm giảm chiều dữ liệu

def perform\_pca(data, n\_components=2):

pca = PCA(n\_components=n\_components)

return pca.fit\_transform(data)

#Giảm chiều dữ liệu

perform\_pca\_data = perform\_pca(scaler\_data)

* Bước 4: Chạy thuật toán CLARANS với đầu vào là:

k = 5 #số cụm (tương ứng với 5 mức độ rating 1 2 3 4 5)

num\_local = 2 #Số lần khởi tạo lại thuật toán

max\_neighbor = 10 #Số hàng xóm tìm kiếm mỗi lần lặp

clarans\_instance = clarans(perform\_pca\_data, k, num\_local, max\_neighbor) # Cấu hình clarans

clarans\_instance.process() #Chạy thuật toán

* Bước 5: Lấy kết quả sau khi phân cụm

# Lấy kết quả

clusters = clarans\_instance.get\_clusters() #trả về kết quả các cụm

medoids = clarans\_instance.get\_medoids() #trả về kết quả các điểm medoids

* Bước 6: Đánh giá mô hình

labels = np.zeros(perform\_pca\_data.shape[0], dtype=int)

for cluster\_id, cluster in enumerate(clusters):

for index in cluster:

labels[index] = cluster\_id

#Các chỉ số đánh giá thuật toán

silhouette\_avg = silhouette\_score(perform\_pca\_data, labels)

davies\_bouldin = davies\_bouldin\_score(perform\_pca\_data, labels)

calinski\_harabasz = calinski\_harabasz\_score(perform\_pca\_data, labels)

# Tạo DataFrame

metrics\_df = pd.DataFrame({

'Metric': ['Silhouette Score', 'Davies-Bouldin Index','Calinski\_harabasz\_score'],

'Value': [silhouette\_avg, davies\_bouldin,calinski\_harabasz]

})

# Hiển thị nội dung DataFrame

print(metrics\_df)

* Bước 7: Trực quan hóa dữ liệu

plt.figure(figsize=(10, 7))

for idx, cluster in enumerate(clusters):

cluster\_points = np.array([perform\_pca\_data[i] for i in cluster])

plt.scatter(cluster\_points[:, 0], cluster\_points[:, 1], label=f'Cluster {idx+1}')

# Vẽ Medoids

medoids\_points = np.array([perform\_pca\_data[i] for i in medoids])

plt.scatter(medoids\_points[:, 0], medoids\_points[:, 1], s=200, c='yellow', marker='X', label='Medoids')

plt.title("CLARANS Clustering")

plt.legend()

plt.grid(True)

plt.show()

* Bước 8: lưu kết quả:

# Tạo timestamp

timestamp = datetime.now().strftime("%Y%m%d\_%H%M%S")

# Đường dẫn lưu biểu đồ

output\_dir1 = './output\_BTL/png/'

output\_file1 = os.path.join(output\_dir1, f'plot\_{timestamp}.png')

# Lưu biểu đồ

plt.savefig(output\_file1, format='png', dpi=300)

print(f"Biểu đồ đã được lưu tại: {output\_file1}")

# Đường dẫn lưu file CSV

output\_dir = './output\_BTL/metrics/'

output\_file = os.path.join(output\_dir, 'clarans\_evaluation\_metrics.csv')

# Kiểm tra xem file đã tồn tại chưa

if os.path.exists(output\_file):

# Nếu file tồn tại, thêm dữ liệu vào

metrics\_df.to\_csv(output\_file, mode='a', index=False, header=False)

print(f"Đã thêm dữ liệu mới vào file: {output\_file}")

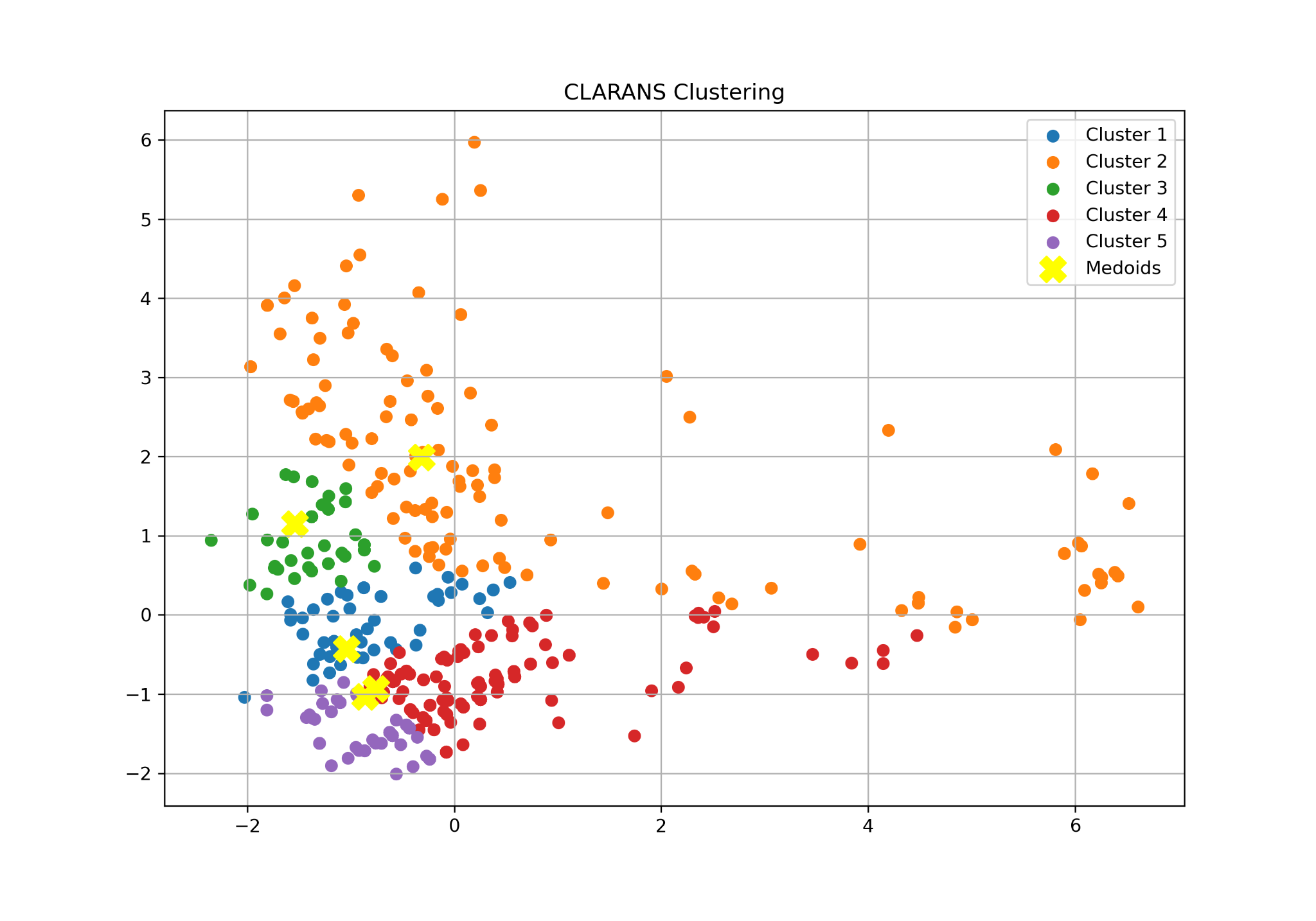
else:

# Nếu file chưa tồn tại, tạo file mới và thêm dữ liệu

metrics\_df.to\_csv(output\_file, mode='w', index=False, header=True)

print(f"File chưa tồn tại. Đã tạo mới và thêm dữ liệu vào file: {output\_file}")

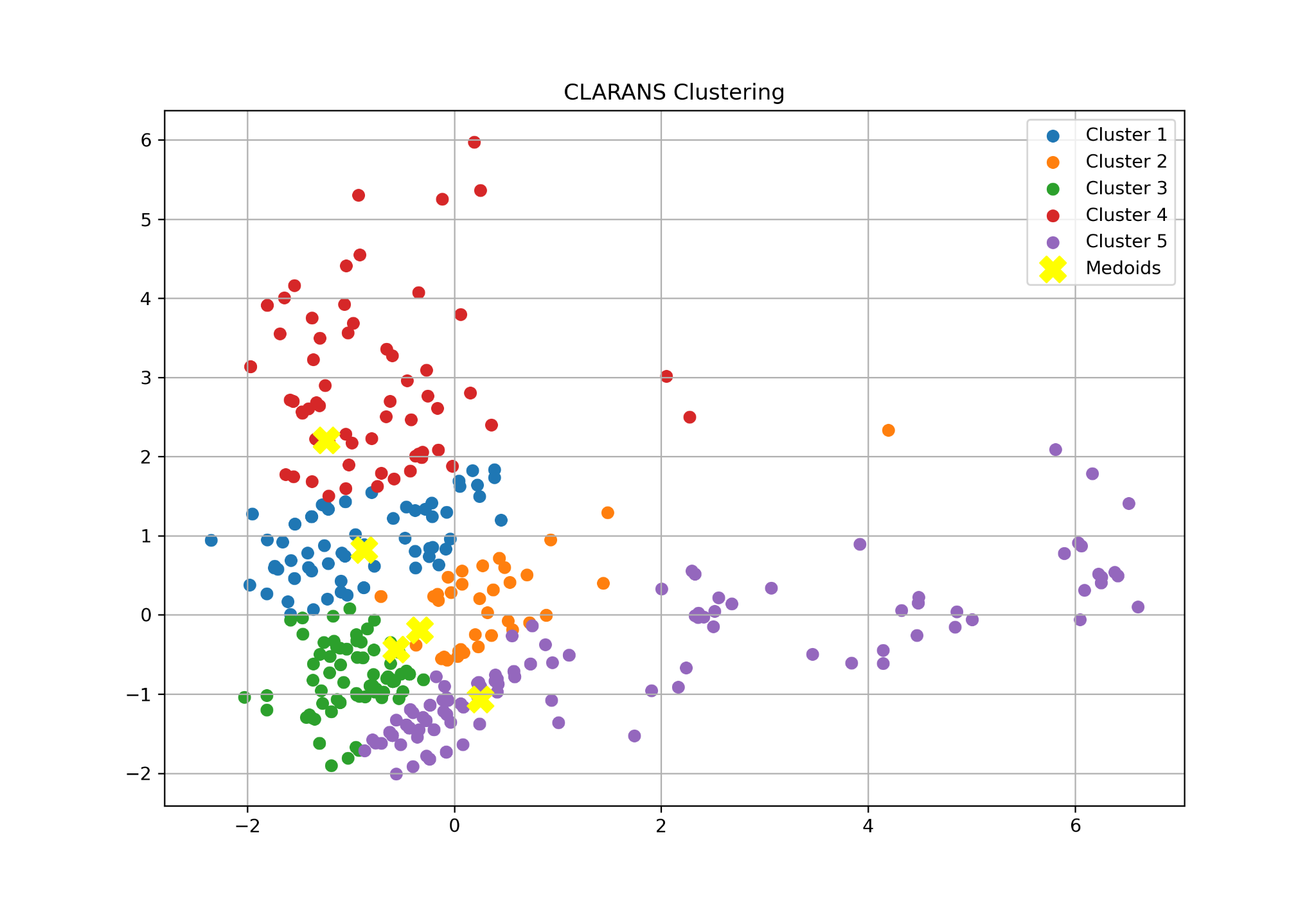
* Kết quả:
  + - Khi chạy thuật toán với 500 bộ dữ liệu:



Silhouette Score: 0.14467831508122447

Davies-Bouldin Index: 1.147838390343091

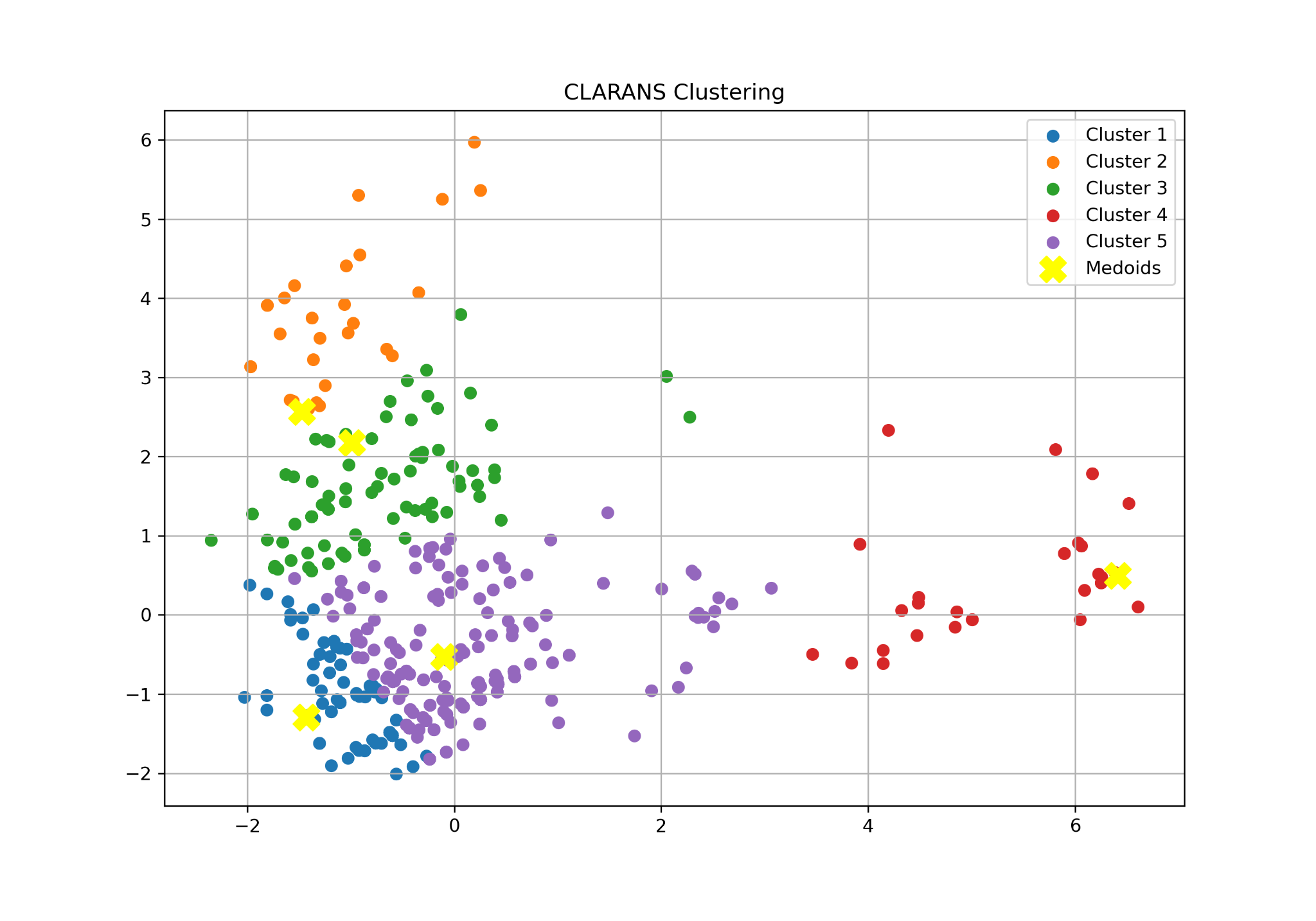
Calinski\_harabasz\_score: 101.06075816224968



Silhouette Score: 0.11391868783860794

Davies-Bouldin Index: 1.443155829181748

Calinski\_harabasz\_score: 123.76404315616409

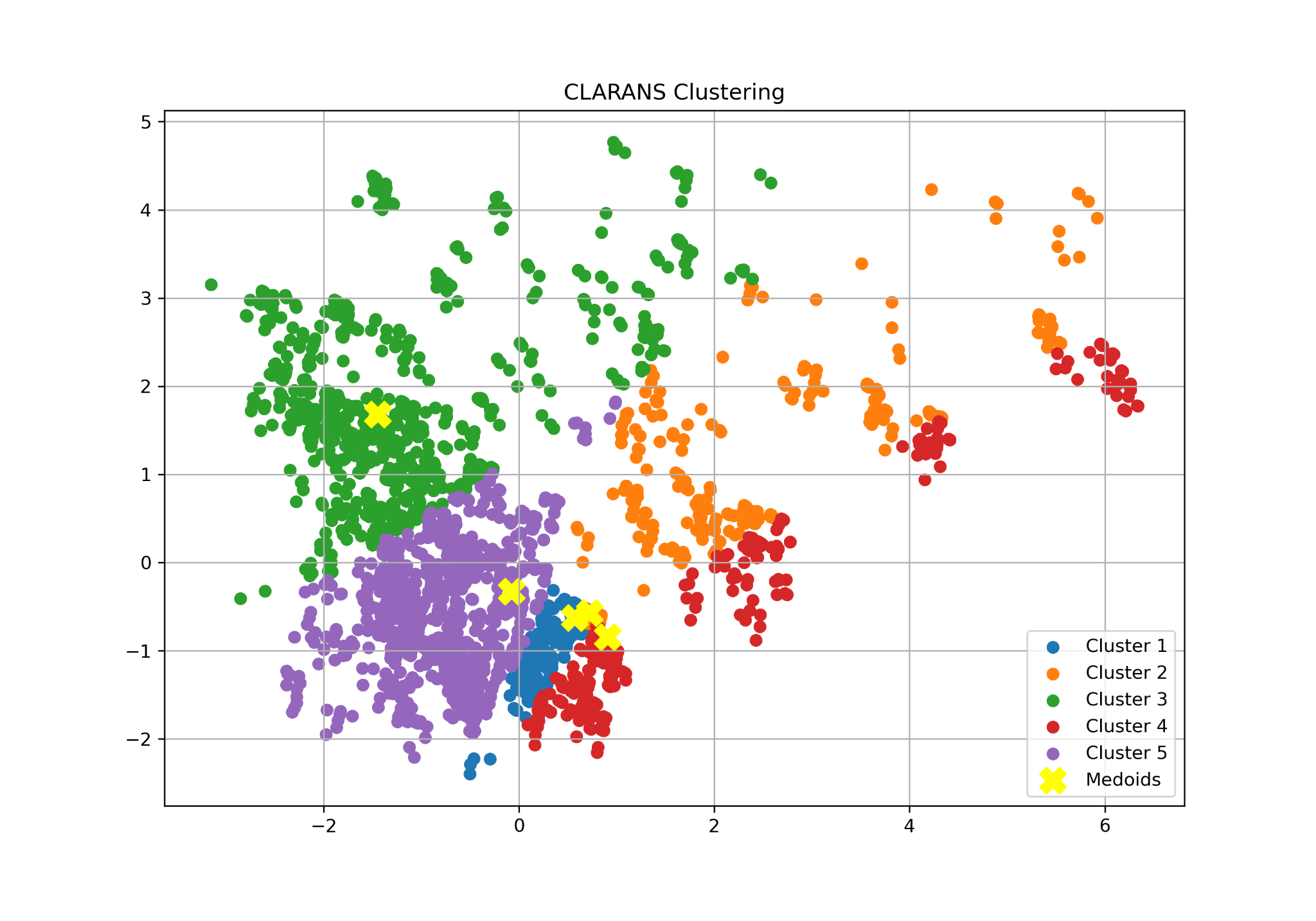


Silhouette Score: 0.263433642272918

Davies-Bouldin Index: 0.9116208330779789

Calinski\_harabasz\_score: 510.9105330144628

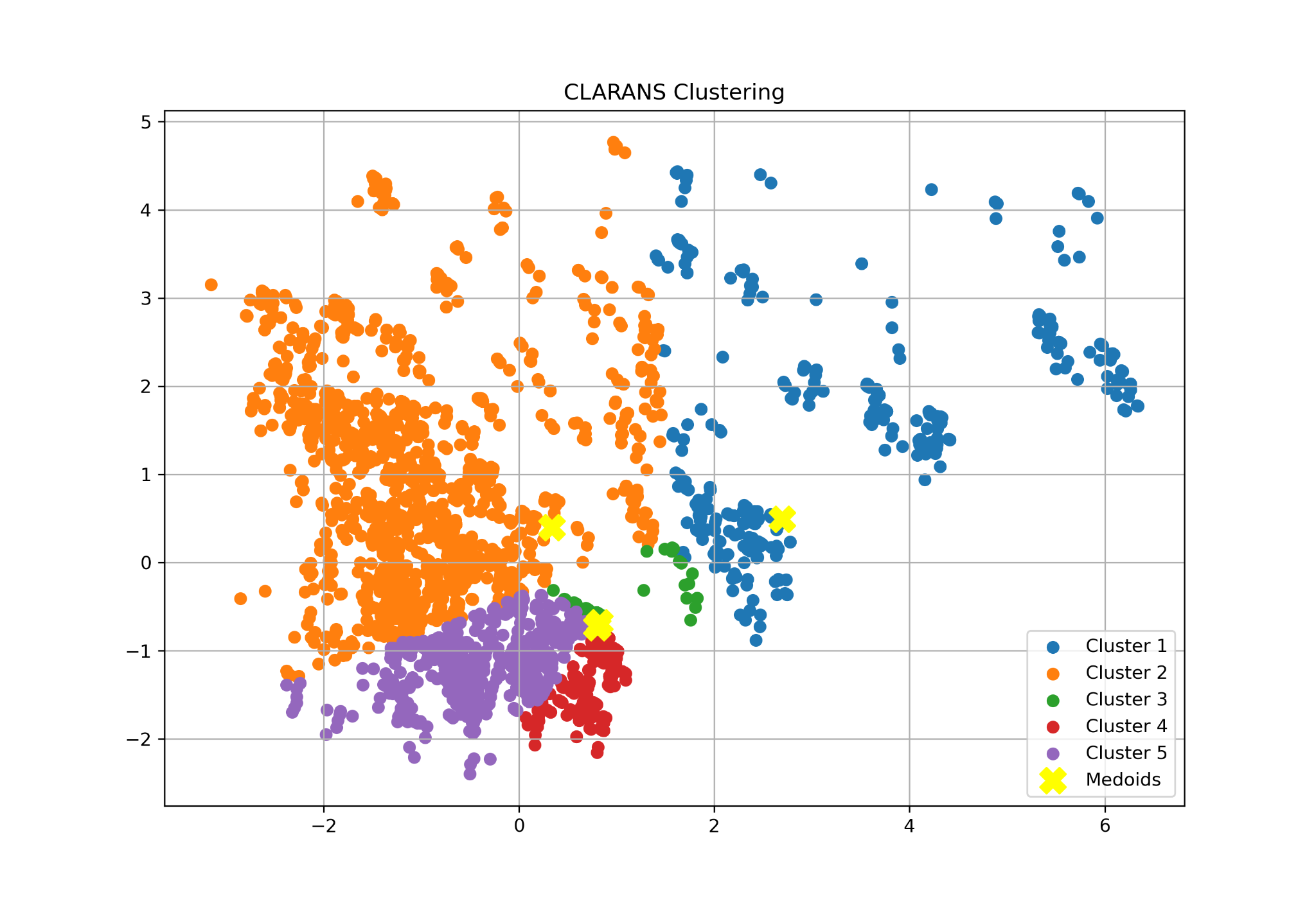
* + - Khi chạy thuật toán với 5000 bộ dữ liệu:



Silhouette Score: 0.11251544768863878

Davies-Bouldin Index: 1.3318649683909982

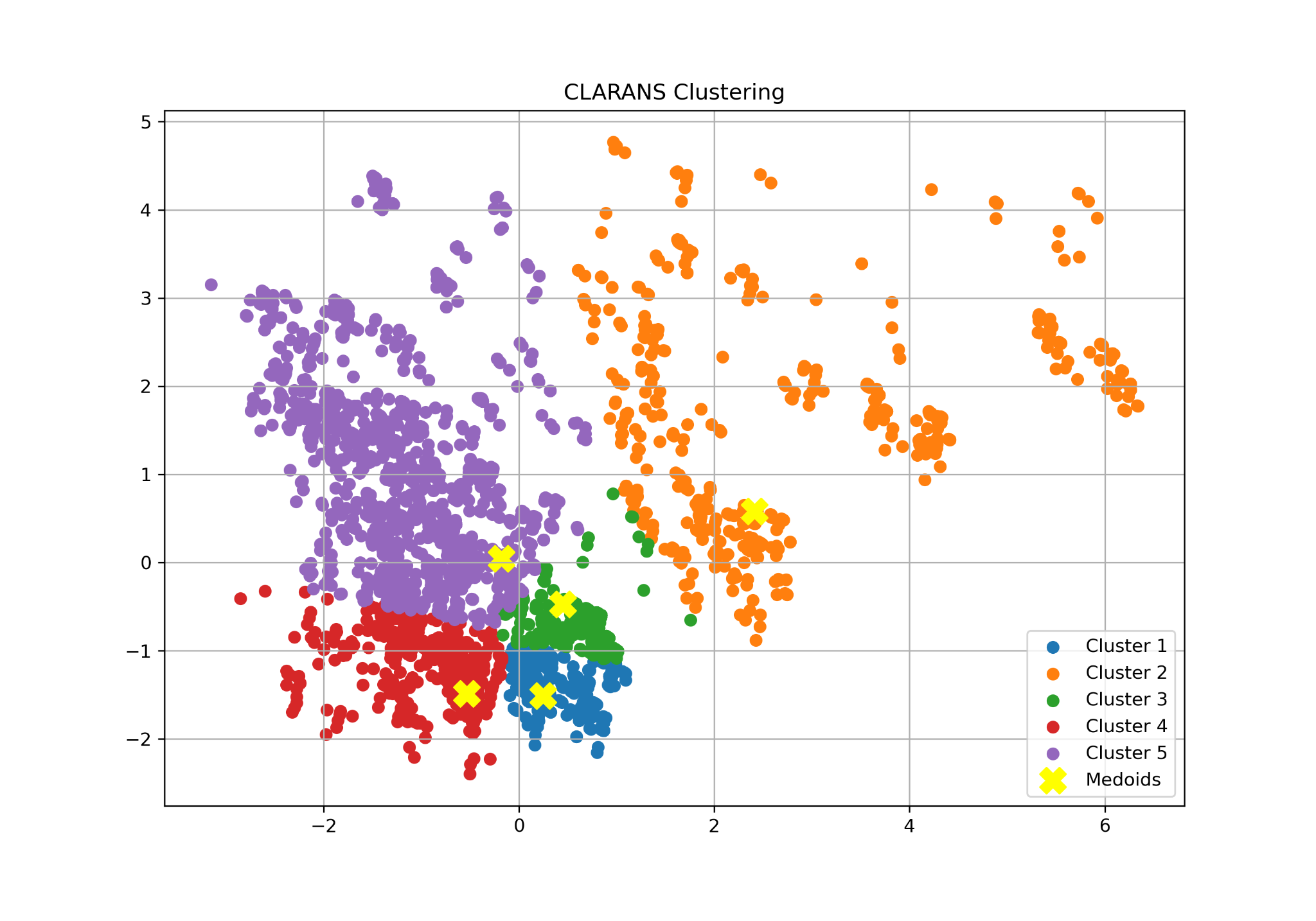
Calinski\_harabasz\_score: 1812.7608002927232



Silhouette Score: 0.20539725599714548

Davies-Bouldin Index: 0.8489004057513224

Calinski\_harabasz\_score: 2587.5297179894224



Silhouette Score: 0.310574692933526

Davies-Bouldin Index: 0.8260655919290686

Calinski\_harabasz\_score: 3107.9766800685115

## **2.Thuật toán BIRCH**

* Các thư viện sử dụng:
* Thư viện pandas: Đọc dữ liệu từ tệp CSV chuyển dữ liệu thành dạng bảng (DataFrame) và thực hiện các thao tác như loại bỏ các cột không cần thiết và lấy mẫu dữ liệu.
* Thư viện numpy: Xử lý các phép toán với mảng numpy như tính toán khoảng cách giữa các cụm hoặc thao tác với mảng trong các chỉ số đánh giá.
* Thư viện matplotlib: vẽ đồ thị, biểu đồ trong Python. Pylot là module con của thư viện này, chuyên dùng để tạo biểu đồ trực quan.
* Thư viện sklearn: hỗ trợ chuẩn hóa dữ liệu, giảm chiều, phân cụm và đánh giá chất lượng phân cụm.
* Thư viện scipy: được sử dụng để tính khoảng cách giữa các điểm trong cùng một cụm nhằm tính toán Dunn’s Index (chỉ số đánh giá phân tách các cụm).
* Các bước thực hiện:
* Bước 1: Đọc tệp dữ liệu CSV và tách các cột đặc trưng (loại bỏ Gender vì nó có thể không liên quan đến phân cụm)

df = pd.read\_csv('clean\_data\_final.csv')

data = df.drop(columns=['Gender']).head(5000).values

* Bước 2: Chuẩn hóa dữ liệu để đảm bảo rằng tất cả các đặc trưng có cùng thang đo trước khi áp dụng phân cụm.

scaler = StandardScaler()

scaler\_data = scaler.fit\_transform(data)

* Bước 3: Giảm chiều dữ liệu từ 21 chiều xuống 2 chiều.

pca = PCA(n\_components=2)

pca\_data = pca.fit\_transform(scaler\_data)

Mặc dù thuật toán BIRCH có thể hoạt động trên dữ liệu lớn và có nhiều chiều nhưng khi số chiều quá lớn, dữ liệu có thể bị loãng, khiến việc phân cụm không chính xác. PCA giúp khắc phục điều này. Hơn nữa, giảm chiều dữ liệu còn hỗ trợ biểu diễn kết quả dễ dàng hơn.

* Bước 4: Chạy thuật toán BIRCH với đầu vào là số cụm mong muốn n\_clusters=5, ngưỡng T=0.5 và yếu tố phân nhánh B=50

birch\_model = Birch(n\_clusters=5, threshold=0.5, branching\_factor=50)

birch\_model.fit(pca\_data)

* Bước 5: Lấy kết quả sau khi thực hiện phân cụm

labels = birch\_model.labels\_

centroids = birch\_model.subcluster\_centers\_

* Bước 6: Đánh giá mô hình

Sử dụng các chỉ số đánh giá sau:

* Silhouette Score: Đánh giá độ phân tách của các cụm. Giá trị gần 1 nghĩa là phân cụm tốt, giá trị thấp hoặc âm là các cụm chồng lấn hoặc không phân biệt rõ.

silhouette = silhouette\_score(pca\_data, labels)

* Davies-Bouldin Index (DBI): Đo lường sự tương tự giữa các cụm. Giá trị càng nhỏ, phân cụm càng tốt. DBI = 0 là phân cụm hoàn hảo.

dbi = davies\_bouldin\_score(pca\_data, labels)

* Bước 7: Trực quan hóa dữ liệu

# Vẽ biểu đồ

plt.figure(figsize=(10, 7))

colors = plt.cm.get\_cmap("tab10", len(set(labels)))

for cluster\_id in set(labels):

cluster\_points = pca\_data[np.array(labels) == cluster\_id]

plt.scatter(cluster\_points[:, 0], cluster\_points[:, 1],

label=f'Cluster {cluster\_id+1}', color=colors(cluster\_id))

# Vẽ centroids

plt.scatter(centroids[:, 0], centroids[:, 1], s=60, c='yellow',

marker='X', label='Centroids')

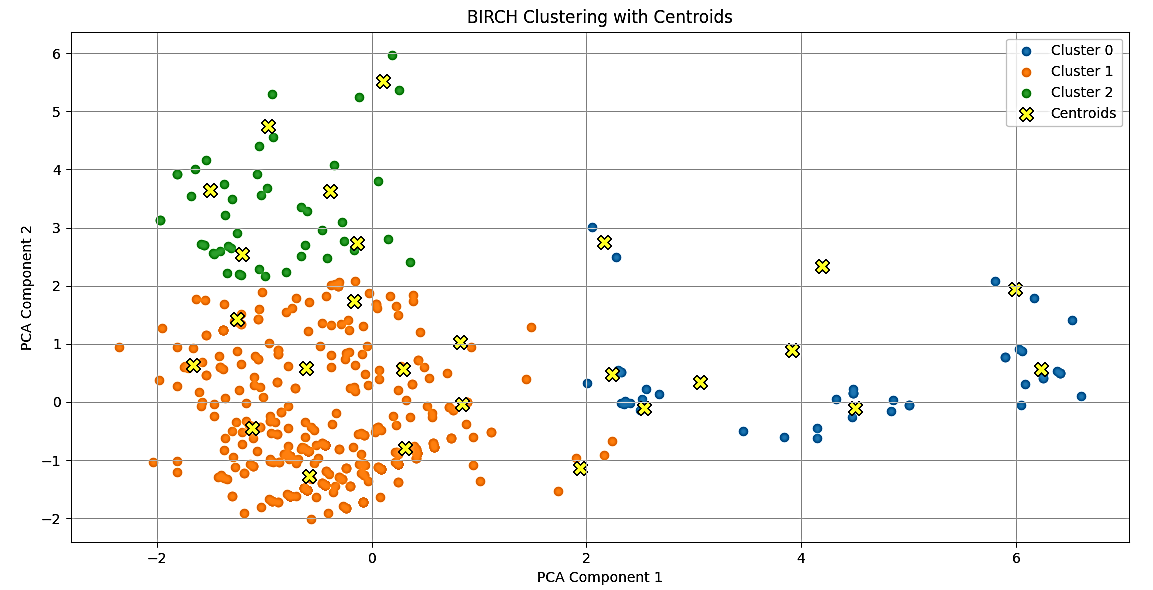
plt.title("BIRCH Clustering")

plt.legend()

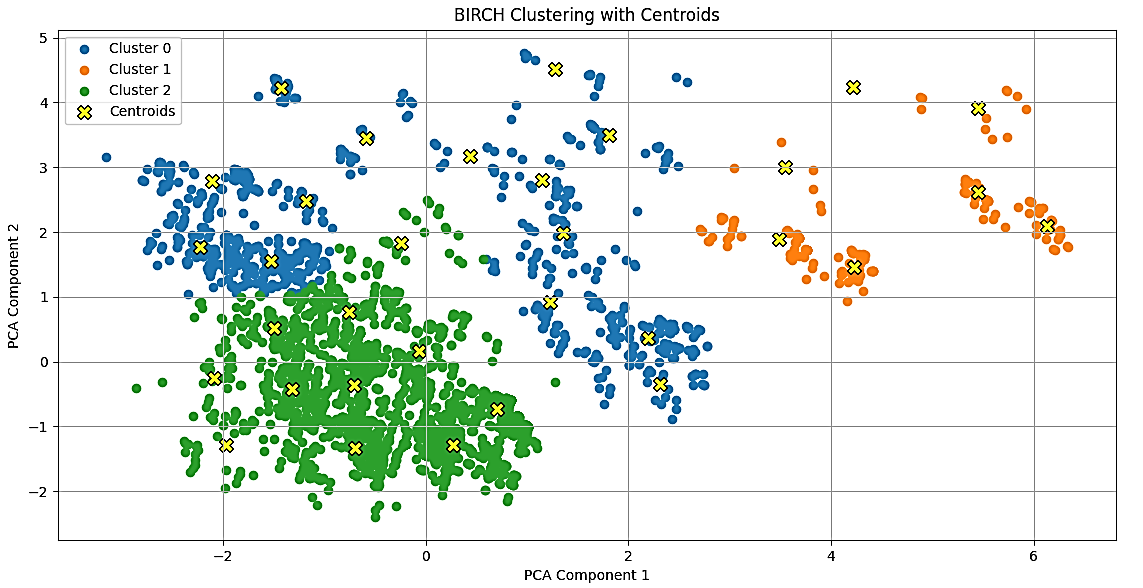
plt.grid(True)

plt.show()

* Kết quả:
  + Với tập dữ liệu 500:



* Silhouette Score: 0.5471
* Calinski-Harabasz Index: 496.6146
* Davies-Bouldin Index (DBI): 0.5581
* Với tập dữ liệu 5000 :



* Silhouette Score: 0.4939
* Calinski-Harabasz Index: 3091.0008
* Davies-Bouldin Index (DBI): 0.9640

## **3.Thuật toán OPTICS:**

* Thư viện sử dụng:

**a) pandas và numpy**

* **pandas**: Thư viện mạnh mẽ để thao tác và phân tích dữ liệu dạng bảng (DataFrame). Thường dùng để nhập, xử lý, và phân tích dữ liệu.
* **numpy**: Cung cấp các công cụ toán học và mảng đa chiều (arrays) hiệu quả, hỗ trợ xử lý dữ liệu lớn.

**b) sklearn.cluster.OPTICS**

* **OPTICS**: Một thuật toán phân cụm (clustering) có khả năng tìm các cụm có hình dạng phức tạp và không đồng nhất về mật độ.
  + Thích hợp cho dữ liệu lớn và phân cụm không đều.
  + Khác với K-means, OPTICS không yêu cầu phải xác định trước số lượng cụm.

**c) sklearn.metrics**

* **silhouette\_score**: Đánh giá độ chặt chẽ và tách biệt của các cụm. Giá trị nằm trong khoảng từ -1 đến 1:
  + **1**: Các cụm được phân biệt rõ ràng.
  + **0**: Các cụm chồng lấn nhau.
  + **-1**: Dữ liệu bị gán sai cụm.
* **calinski\_harabasz\_score**: Đánh giá tỷ lệ giữa tổng phương sai giữa các cụm và tổng phương sai trong cụm. Giá trị càng cao, chất lượng cụm càng tốt.
* **davies\_bouldin\_score**: Đánh giá mức độ tương đồng giữa các cụm. Giá trị càng thấp, cụm càng tốt.

**d) matplotlib.pyplot**

* Thư viện vẽ đồ thị, dùng để trực quan hóa dữ liệu, kết quả phân cụm, hoặc các biểu đồ khác.

**e) sklearn.decomposition.PCA**

* **PCA (Principal Component Analysis)**: Kỹ thuật giảm số chiều của dữ liệu, giúp biểu diễn dữ liệu trong không gian ít chiều hơn (ví dụ: từ 10 chiều về 2 chiều) nhưng vẫn giữ được nhiều thông tin quan trọng.
  + Thường dùng để trực quan hóa dữ liệu phân cụm trong không gian 2D hoặc 3D.

**f) sklearn.preprocessing.StandardScaler**

* Dùng để chuẩn hóa dữ liệu (scaling), tức là biến dữ liệu về cùng một thang đo (mean = 0 và standard deviation = 1).
  + Rất quan trọng khi dữ liệu có các cột với đơn vị đo lường khác nhau.
* Các bước thực hiện:

1. Đọc và tiền xử lý dữ liệu

print("==> Bước 1: Đọc dữ liệu")

df = pd.read\_csv('data\\clean\_data\_final.csv').head(5000) # Lấy 5000 bản ghi đầu tiên

print("Dữ liệu ban đầu (5 dòng đầu):")

print(df.head())

# Loại bỏ cột 'Gender' không cần thiết cho phân cụm

data = df.drop(columns=['Gender'])

2. Chuẩn hóa dữ liệu

print("\n==> Bước 2: Chuẩn hóa dữ liệu")

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(data)

3. Giảm chiều dữ liệu bằng PCA

print("\n==> Bước 3: Giảm chiều dữ liệu với PCA")

pca = PCA(n\_components=2)

pca\_data = pca.fit\_transform(scaled\_data)

print("Dữ liệu sau khi giảm chiều (5 dòng đầu):")

print(pca\_data[:5])

4. Áp dụng thuật toán OPTICS

print("\n==> Bước 4: Phân cụm dữ liệu với OPTICS")

optics\_model = OPTICS(min\_samples=5, xi=0.05, min\_cluster\_size=0.05)

optics\_model.fit(pca\_data)

# Lấy nhãn cụm

labels = optics\_model.labels\_

# Thêm nhãn cụm vào DataFrame gốc

df['Cluster'] = labels

5. Đánh giá mô hình

print("\n==> Bước 5: Đánh giá mô hình phân cụm")

silhouette = silhouette\_score(pca\_data, labels)

calinski\_harabasz = calinski\_harabasz\_score(pca\_data, labels)

dbi = davies\_bouldin\_score(pca\_data, labels)

print(f"Silhouette Score: {silhouette:.4f}")

print(f"Calinski-Harabasz Index: {calinski\_harabasz:.4f}")

print(f"Davies-Bouldin Index (DBI): {dbi:.4f}")

# Lưu kết quả đánh giá vào tệp CSV

metrics\_df = pd.DataFrame({

'Metric': ['Silhouette Score', 'Calinski-Harabasz Index', 'Davies-Bouldin Index'],

'Value': [silhouette, calinski\_harabasz, dbi]

})

metrics\_df.to\_csv('results\\optics\_clustering\_metrics.csv', index=False)

6. Phân tích đặc điểm từng cụm

print("\n==> Bước 6: Phân tích đặc điểm cụm")

genres\_columns = ['War', 'Horror', 'Musical', 'Crime', 'Mystery', 'Film-Noir', 'Animation', 'Thriller',

'Documentary', 'Comedy', 'Fantasy', 'Western', 'Adventure', 'Action', 'Sci-Fi', 'Romance',

'Childrens', 'Drama']

for cluster\_id in np.unique(labels):

if cluster\_id != -1: # Loại bỏ các điểm nhiễu (noise) có nhãn là -1

cluster\_data = df[df['Cluster'] == cluster\_id]

print(f"\nCụm {cluster\_id}:")

print(f"- Số lượng bản ghi: {len(cluster\_data)}")

print(f"- Độ tuổi trung bình: {cluster\_data['Age'].mean():.2f}")

print(f"- Điểm đánh giá trung bình: {cluster\_data['Rating'].mean():.2f}")

# Tính tần suất các thể loại phim

genre\_frequencies = cluster\_data[genres\_columns].mean()

print("\n- Tần suất các thể loại phim:")

print(genre\_frequencies.to\_string(index=True))

# Thể loại phim phổ biến nhất

favorite\_genre = genre\_frequencies.idxmax()

print(f"- Thể loại phổ biến nhất: {favorite\_genre} ({genre\_frequencies[favorite\_genre]:.4f})")

7. Tính toán tọa độ tâm cụm

print("\n==> Bước 7: Tính toán tọa độ tâm cụm")

centroids = {}

for cluster\_id in np.unique(labels):

if cluster\_id != -1: # Loại bỏ các điểm nhiễu

cluster\_points = pca\_data[labels == cluster\_id]

centroid = cluster\_points.mean(axis=0) # Tính toán tọa độ trung bình (tâm cụm)

centroids[cluster\_id] = centroid

print(f"Cụm {cluster\_id} - Tọa độ tâm cụm: {centroid}")

8. Vẽ biểu đồ phân cụm và tâm cụm

print("\n==> Bước 8: Vẽ biểu đồ phân cụm và tâm cụm")

plt.figure(figsize=(10, 7))

colors = plt.colormaps["tab10"].colors

# Vẽ từng cụm

for cluster\_id in np.unique(labels):

if cluster\_id != -1: # Loại bỏ các điểm nhiễu

cluster\_points = pca\_data[labels == cluster\_id]

plt.scatter(cluster\_points[:, 0], cluster\_points[:, 1], label=f'Cluster {cluster\_id}')

# Vẽ tâm cụm

for cluster\_id, centroid in centroids.items():

plt.scatter(centroid[0], centroid[1], marker='X', color='black', s=20, label=f'Centroid {cluster\_id}')

# Vẽ các điểm nhiễu (nếu có)

noise\_points = pca\_data[labels == -1]

plt.scatter(noise\_points[:, 0], noise\_points[:, 1], color='gray', label='Noise', alpha=0.5)

# Trang trí biểu đồ

plt.title("OPTICS Clustering with Centroids")

plt.xlabel("PCA Component 1")

plt.ylabel("PCA Component 2")

plt.legend()

plt.grid(True)

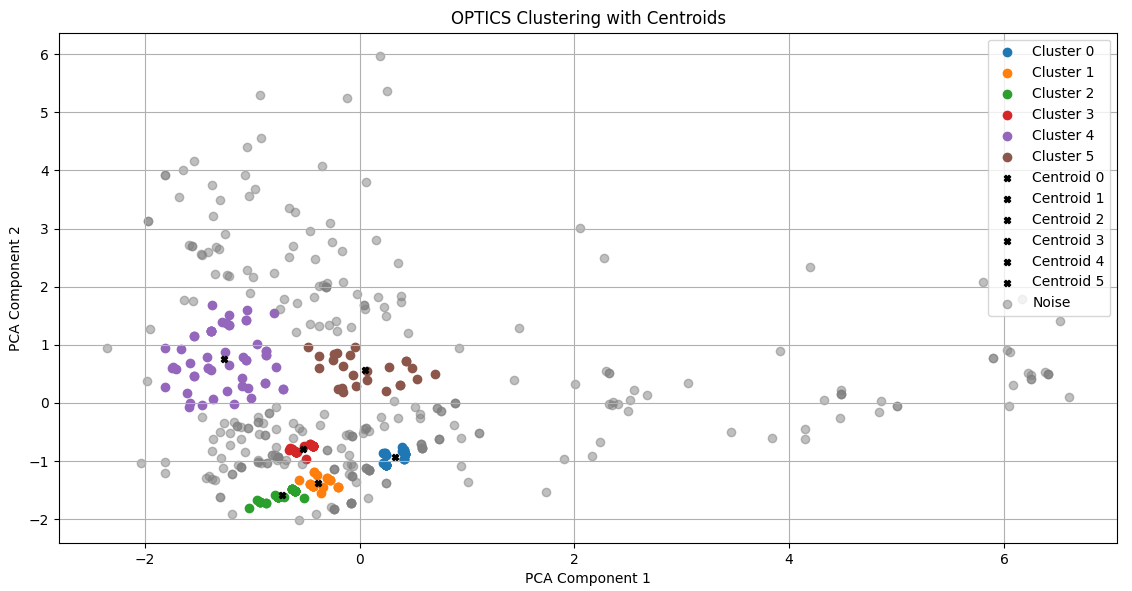
# Lưu biểu đồ vào tệp

plt.savefig('results\\optics\_clustering\_with\_centroids.png')

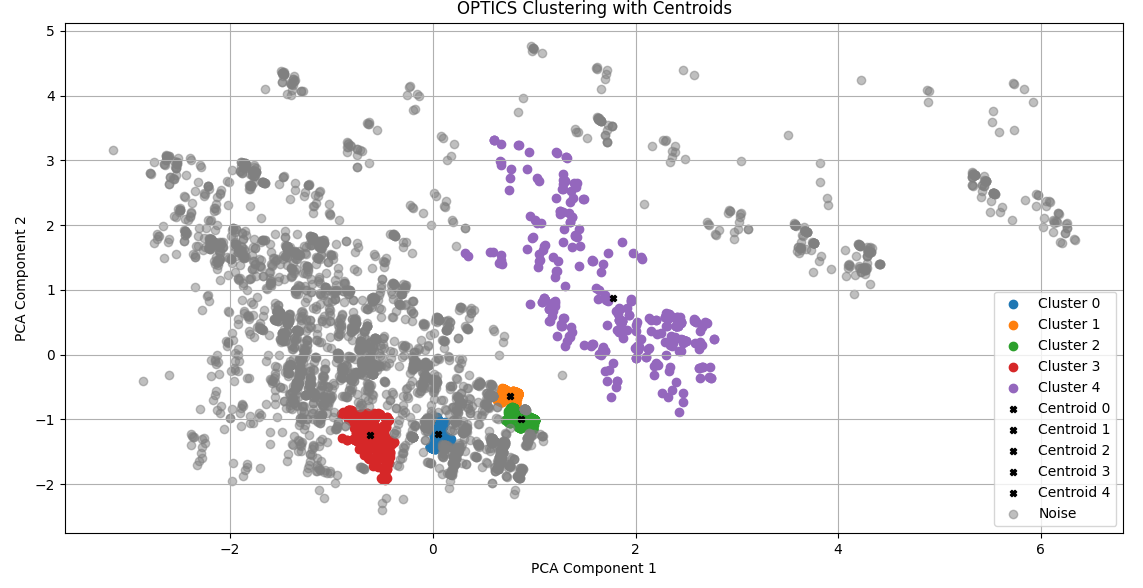
plt.show()

**Kết quả**:

* Với tập dữ liệu 500:



* Silhouette Score: **-0.0632**
* Calinski-Harabasz Index: **18.0931**
* Davies-Bouldin Index (DBI): **2.9335**
* Với tập dữ liệu 5000:



* Silhouette Score: **-0.0629**
* Calinski-Harabasz Index: **221.1231**
* Davies-Bouldin Index (DBI): **1.2988**

# **V. So sánh và đánh giá kết quả thu được**

### **Tập dữ liệu 500 dòng:**

#### 1. CLARANS:

* Silhouette Score dao động từ **0.263** (khá thấp, cụm không rõ ràng).
* Davies-Bouldin Index tốt nhất là **0.911** (cụm chặt chẽ và rõ ràng nhất trong các giá trị của CLARANS).
* Calinski-Harabasz cao nhất là **510.91** (phân cụm tương đối tốt).

#### 2. BIRCH:

* Silhouette Score: **0.5471** (cụm tốt hơn trong tập dữ liệu 500 dòng).
* Calinski-Harabasz Index: **496.6146** (cụm tương đối rõ).
* Davies-Bouldin Index (DBI): **0.5581** (cụm phân biệt rõ hơn).

#### 3. OPTICS:

* Silhouette Score: **-0.0632** (cụm yếu).
* Davies-Bouldin Index: **2.9335** (phân biệt kém).
* Calinski-Harabasz: **18.0931** (cụm không rõ ràng).

**=> Tập 500 dòng: BIRCH là thuật toán tốt nhất.**

### **Tập dữ liệu 5000 dòng:**

#### 1. CLARANS:

* Silhouette Score dao động từ **0.310** (Silhouette cao nhất là **0.310**, cho thấy cụm khá rõ ràng ở một số trường hợp).
* Davies-Bouldin Index tốt nhất là **0.826** (cụm chặt chẽ).
* Calinski-Harabasz cao nhất là **3107.98**, rất tốt.

#### 2. BIRCH:

* Silhouette Score: **0.4939** (cải thiện rõ rệt, cụm tốt hơn so với CLARANS).
* Calinski-Harabasz Index: **3091.0008** (rất tốt cụm rõ ràng).
* Davies-Bouldin Index (DBI): **0.9640** (cụm phân biệt ổn).

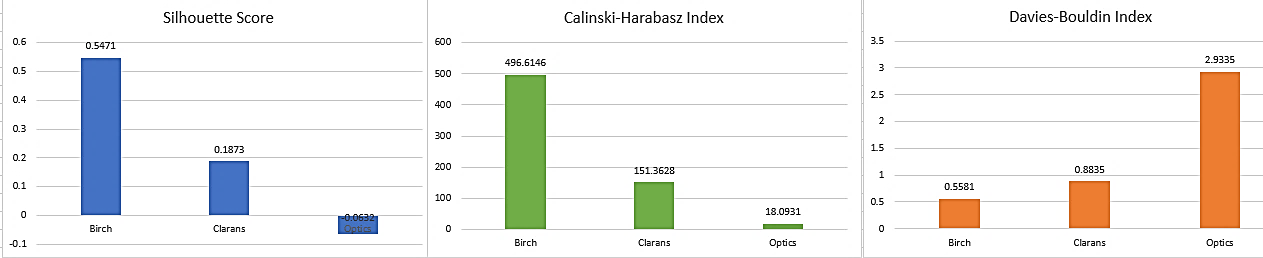
#### 3. OPTICS:

* Silhouette Score: **-0.0961** (cụm yếu).
* Davies-Bouldin Index: **1.4162** (phân biệt kém).
* Calinski-Harabasz: **211.8504** (rất thấp, cụm không rõ ràng).

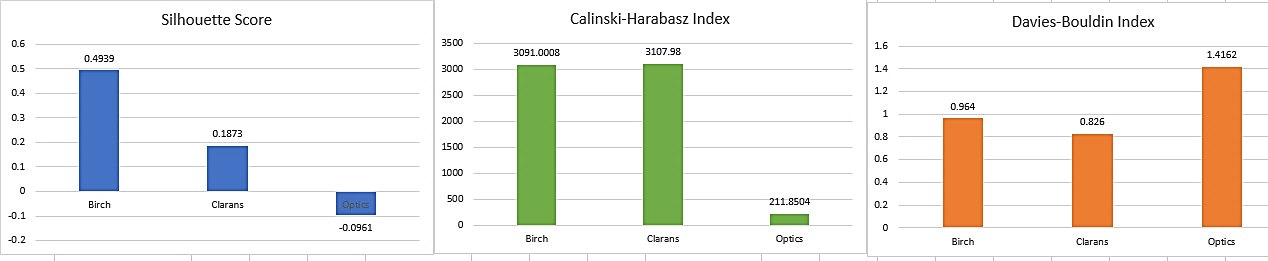
**=> Tập 5000 dòng: BIRCH tiếp tục là thuật toán tốt nhất.**

**Kết luận**

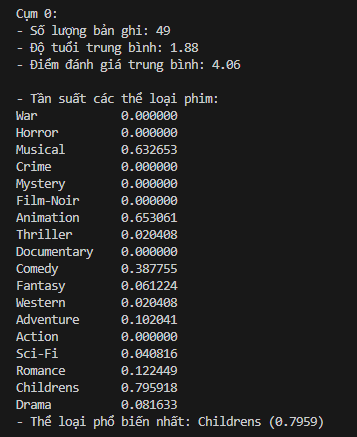
* Với tập dữ liệu 500 dòng :

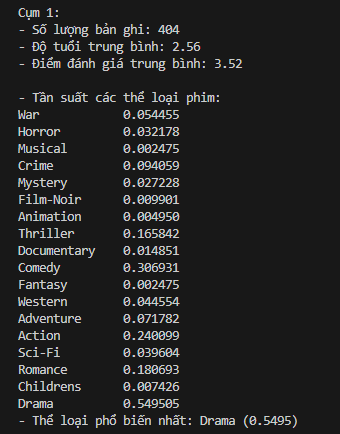


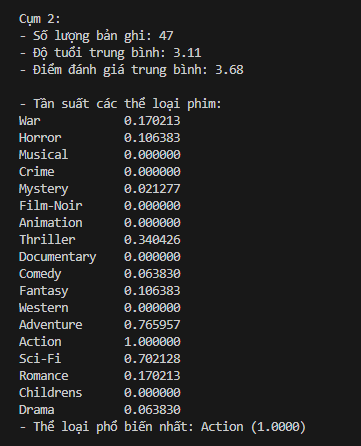
* **BIRCH** là thuật toán hoạt động tốt nhất trên cả hai tập dữ liệu. Nó có các chỉ số Silhouette cao nhất, Davies-Bouldin nhỏ nhất, và Calinski-Harabasz lớn nhất.
* **CLARANS** có thể cạnh tranh ở một số trường hợp, nhưng không ổn định bằng BIRCH.
* **OPTICS** không phù hợp với tập dữ liệu này do các chỉ số đánh giá đều rất kém.
* Với tập dữ liệu 5000 dòng:



* **BIRCH** là thuật toán hiệu quả nhất với Silhouette Score cao nhất và chỉ số Calinski-Harabasz cho thấy phân cụm rõ ràng. Mặc dù DBI tăng nhẹ, BIRCH vẫn là lựa chọn tối ưu.
* **CLARANS** là lựa chọn thứ hai tốt nhất, với chỉ số Calinski-Harabasz cao nhưng Silhouette Score thấp hơn, cho thấy sự phân cụm chưa hoàn toàn rõ ràng.
* **OPTICS** có hiệu suất kém nhất, với tất cả các chỉ số đều cho thấy sự phân cụm rất yếu và không rõ ràng.
* Chọn BIRCH để thực hiện với dataset:
* Với dataset 500 dòng ta có:







* Cụm 0

Số lượng bản ghi: 49

Độ tuổi trung bình: 1.88 (18-34)

Điểm đánh giá trung bình: 4.06 (khá cao)

Tần suất các thể loại phim:

* Thể loại phổ biến nhất là Childrens (0.7959), cho thấy nhóm này chủ yếu yêu thích các bộ phim thiếu nhi.
* Các thể loại khác như Musical, Animation và Comedy cũng có tần suất cao, phản ánh sở thích giải trí nhẹ nhàng và vui nhộn.
* Action, Sci-Fi và War không được yêu thích trong nhóm này.

Nhận xét: Cụm này có xu hướng yêu thích các bộ phim dành cho trẻ em và phim hoạt hình. Đặc điểm của nhóm này có thể là các gia đình hoặc những người có sở thích hướng tới sự giải trí nhẹ nhàng, vui tươi, với đánh giá cao về chất lượng phim.

* Cụm 1

Số lượng bản ghi: 404

Độ tuổi trung bình: 2.56 (35-54)

Điểm đánh giá trung bình: 3.52 (khá cao, nhưng thấp hơn so với Cụm 0)

Tần suất các thể loại phim:

* Thể loại phổ biến nhất là Drama (0.5495), cho thấy nhóm này chủ yếu yêu thích các bộ phim tâm lý và kịch tính.
* Các thể loại khác có tần suất cao bao gồm Comedy, Action, Romance và Thriller, phản ánh sự đa dạng trong sở thích phim ảnh của nhóm này.
* Các thể loại như Musical, Fantasy, Documentary, và Childrens ít được ưa chuộng.

Nhận xét: Cụm này đại diện cho nhóm người xem có sở thích đa dạng, nhưng với sự ưa chuộng mạnh mẽ đối với các bộ phim tâm lý (Drama). Đây có thể là nhóm người trưởng thành, có sở thích xem các bộ phim phức tạp, nhiều cảm xúc và kịch tính. Mặc dù điểm đánh giá trung bình không quá cao, nhưng vẫn nằm trong mức khá.

* Cụm 2

Số lượng bản ghi: 47

Độ tuổi trung bình: 3.11 (35-54)

Điểm đánh giá trung bình: 3.68 (cao hơn Cụm 1)

Tần suất các thể loại phim:

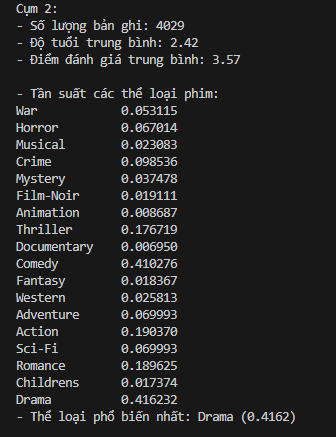
* Thể loại phổ biến nhất là Action (1.0000), với tần suất cực kỳ cao, cho thấy nhóm này chỉ yêu thích phim hành động.
* Các thể loại khác như Sci-Fi, Adventure, Thriller và War cũng có sự yêu thích đáng kể, phản ánh sự ưa chuộng các bộ phim thể thao, phiêu lưu và hành động mạnh mẽ.
* Các thể loại như Comedy, Drama, Childrens không được chú trọng.

Nhận xét: Cụm này đại diện cho nhóm người xem có sở thích rất mạnh mẽ đối với các bộ phim hành động và phiêu lưu. Với độ tuổi trung bình cao hơn và điểm đánh giá cao, nhóm này có thể là những người trưởng thành, yêu thích sự kịch tính và mạnh mẽ của các bộ phim hành động.

* Với dataset 5000 dòng ta có:







* Cụm 0:

Số lượng bản ghi: 731

Độ tuổi trung bình: 2.52 (18-34)

Điểm đánh giá trung bình: 3.52

Nhận xét:

* Thể loại phổ biến nhất là Action, chiếm tỉ lệ cao nhất (0.8153), cho thấy nhóm người xem này chủ yếu thiên về các bộ phim hành động.
* Ngoài hành động, các thể loại như Adventure, Sci-Fi, Thriller, và Fantasy cũng được xem nhiều, chứng tỏ sự quan tâm đến những bộ phim có yếu tố phiêu lưu, khoa học viễn tưởng và kịch tính.
* Cụm 1:

Số lượng bản ghi: 240

Độ tuổi trung bình: 2.17(18-34)

Điểm đánh giá trung bình: 3.80

Nhận xét:

* Thể loại phổ biến nhất là Childrens (0.9750), cho thấy nhóm này chủ yếu xem các bộ phim dành cho trẻ em.
* Các thể loại khác như Animation, Comedy, Musical cũng xuất hiện với tần suất cao, phản ánh sở thích đa dạng, nhưng vẫn chủ yếu tập trung vào những thể loại mang tính giải trí và hoạt hình.
* Cụm 2:

Số lượng bản ghi: 4029

Độ tuổi trung bình: 2.42(18-34)

Điểm đánh giá trung bình: 3.57

Nhận xét:

* Thể loại phổ biến nhất là Drama (0.4162), cho thấy người xem trong cụm này chủ yếu yêu thích các bộ phim tâm lý, kịch tính.
* Các thể loại khác như Comedy, Action, Romance, Sci-Fi cũng được ưa chuộng, chứng tỏ sự quan tâm đến cả các bộ phim giải trí và các câu chuyện tình cảm.

# **Danh mục tài liệu tham khảo**

1.<https://analyticsindiamag.com/developers-corner/comprehensive-guide-to-clarans-clustering-algorithm/>

2.<https://fr.slideshare.net/slideshow/phn-cm-d-liu-trong-datamingpdf/258056391>

3.<https://www.jaist.ac.jp/~bao/VNAlectures/Clustering_VNA_Hien(M6M7).pdf>

4.<https://cs.ecu.edu/dingq/CSCI6905/readings/CLARANS.pdf>