**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**QUẢN LÝ DỮ LIỆU LỚN**

***Đề tài:***

**ÁP DỤNG MAPREDUCE VÀO THUẬT TOÁN KMEANS  
ĐỂ PHÂN CỤM KHÁCH HÀNG TẠI TRUNG TÂM MUA SẮM**

**Giảng viên:** Nguyễn Đắc Phương Thảo

**Nhóm thực hiện:** 02

**Lớp:** 64TTNT2

**Tên sinh viên:**

1. Ngô Quang Minh – 2251262618
2. Võ Quang Phong – 2251262624
3. Phạm Thiện Cường – 2251262583
4. Phạm Thị Kim Mây – 2251262617
5. Nguyễn Ngọc Anh – 2251262572

# **LỜI NÓI ĐẦU**

Lời đầu tiên, em xin chân thành cảm ơn Khoa Công nghệ Thông tin, Trường Đại học Thuỷ Lợi đã tạo điều kiện tốt nhất để chúng em được học môn Quản Lý Dữ Liệu Lớn. Đặc biệt chúng em xin chân thành cảm ơn cô Nguyễn Đắc Phương Thảo – Giảng viên giảng dạy bộ môn Quản Lý Dữ Liệu Lớn của lớp chúng em trong kỳ này. Chúng em vô cùng biết ơn và xin kính chúc Cô luôn mạnh khoẻ để tiếp tục cống hiến cho sự nghiệp giáo dục và đạt được nhiều thành công hơn nữa.

Với tất cả sự cố gắng của nhóm trong quá trình thực hiện đề tài, song do trình độ, hiểu biết nên đề tài khó tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý, chỉ bảo của Cô để đề tài được hoàn thiện hơn.

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 2](#_Toc181141860)

[**DANH SÁCH HÌNH VẼ** 5](#_Toc181141861)

[**DANH SÁCH CÁC BẢNG BIỂU** 6](#_Toc181141862)

[**PHẦN MỞ ĐẦU** 7](#_Toc181141863)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI** 8](#_Toc181141864)

[**1.1 CƠ SỞ LÝ THUYẾT** 8](#_Toc181141865)

[**1.1.1. Data Clustering** 8](#_Toc181141866)

[**1.1.2. Các loại phương pháp gom cụm phổ biến** 8](#_Toc181141867)

[**1.1.3. Các phương pháp đánh giá chất lượng gom cụm** 9](#_Toc181141868)

[**1.1.4. Thuật toán K-means dựa trên phân hoạch** 9](#_Toc181141869)

[**1.1.5. Áp dụng MapRuduce vào thuật toán K-means** 10](#_Toc181141870)

[**1.1.6. Các bước thực hiện MapRuduce với thuật toán K-means** 11](#_Toc181141871)

[**1.2. NỘI DUNG ĐỀ TÀI** 12](#_Toc181141872)

[**1.3. MỤC TIÊU NGHIÊN CỨU** 12](#_Toc181141873)

[**1.3.2. Nhu cầu đời sống** 12](#_Toc181141874)

[**1.3.3. Lợi ích cho doanh nghiệp** 12](#_Toc181141875)

[**1.4. ĐỐI TƯỢNG NGHIÊN CỨU** 13](#_Toc181141876)

[**1.5. KẾT QUẢ MONG ĐỢI** 13](#_Toc181141877)

[**CHƯƠNG 2: MÔ HÌNH THỰC NGHIỆM** 14](#_Toc181141878)

[**2.1. QUY TRÌNH GOM CỤM DỮ LIỆU** 14](#_Toc181141879)

[**2.2. THU THẬP DỮ LIỆU** 15](#_Toc181141880)

[**2.3. MÔ HÌNH THUẬT TOÁN K-MEANS** 16](#_Toc181141881)

[**2.3.1. Ý tưởng** 16](#_Toc181141882)

[**2.3.2. Mô hình thuật toán** 17](#_Toc181141883)

[**2.3.3. Thuật toán** 17](#_Toc181141884)

[**2.3.3. Mô hình thuật toán K-Means khi áp dụng với kỹ thuật MapReduce** 18](#_Toc181141885)

[**2.4. CÔNG THỨC TOÁN HỌC** 18](#_Toc181141886)

[**2.5. THỰC NGHIỆM VỚI DỮ LIỆU MẪU** 19](#_Toc181141887)

[**2.6. ỨNG DỤNG** 23](#_Toc181141888)

[**2.7 ĐÁNH GIÁ KẾT QUẢ** 45](#_Toc181141889)

[**2.7.1 Tri thức rút ra sau khi phân cụm** 45](#_Toc181141890)

[**2.7.2 So sánh 2 thuật toán** 46](#_Toc181141891)

[**CHƯƠNG 3: KẾT LUẬN** 47](#_Toc181141892)

[**3.1 NHỮNG KIẾN THƯC ĐÃ TIẾP THU TỪ MÔN HỌC** 47](#_Toc181141893)

[**3.2 HƯỚNG PHÁT TRIỂN** 47](#_Toc181141894)

# **DANH SÁCH HÌNH VẼ**

**Hình 1.1. Phương pháp gom cụm dữ liệu** 9

**Hình 2.1. Gom cụm dữ liệu** 14

**Hình 2.2. Flowchart giải thuật K-means** 17

**Hình 2.3. Bảng dữ liệu** 23

**Hình 2.4. So sánh tỉ lệ Male (Nam), Female (Nữ) của thuộc tính Gender** 24

**Hình 2.5. Thông tin của Age, Annual Income (k$), Spending Score (1-100)** 25

**Hình 2.6. Hệ số tương quan** 25

**Hình 2.7. Số liệu thống kê** 26

**Hình 2.8. Số liệu thống kê** 27

**Hình 2.9. Sự phân bố của dữ liệu đối với các thuộc tính** 27

**Hình 2.10. Mối quan hệ giữa Age, Annual Income (k$), Spending Score (1-100)** 28

**Hình 2.11. Chất lượng gom cụm của thuật toán K-means với 3 cụm** 30

**Hình 2.12. Tìm k cụm với phương pháp Elbow** 31

**Hình 2.13. Số cluster và tâm cụm** 32

**Hình 2.14. Tìm k cụm với phương pháp Silhouette Coefficient** 33

**Hình 2.15. Độ chính xác của thuật toán đối với cụm bằng 5** 34

**Hình 2.16. Biểu diễn dữ liệu đã được phân cụm với k = 5** 35

**Hình 2.17. Phân bố dữ liệu đã được gom cụm** 36

**Hình 2.18. Danh sách các điểm dữ liệu mang thuộc tính CustomerID** 38

**Hình 2.19. Dự đoán số cụm thuộc về dựa trên một input mới** 39

**Hình 2.20. Tìm k với thư viện yellowbrick sử dụng phương pháp Elbow** 40

**Hình 2.21. Cây phân cấp đối với tập dữ liệu với số cây bằng 5** 42

**Hình 2.22. Mô hình so sánh dữ liệu với 2 thuật toán khác nhau** 43

# **DANH SÁCH CÁC BẢNG BIỂU**

**Bảng 2.1. Thông tin khách hàng tại khu trung tâm mua sắm thu thập từ cuộc thi Kaggle** 15

**Bảng 2.2. Bảng thực nghiệm với 5 dòng từ dữ liệu gốc chưa chuẩn hóa** 19

**Bảng 2.3. Vòng lặp thứ nhất tính khoảng cách** 20

**Bảng 2.4. Vòng lặp thứ nhất phân cụm** 20

**Bảng 2.5. Vòng lặp thứ 2 tính khoảng cách** 21

**Bảng 2.6. Vòng lặp thứ 2 phân cụm** 21

**Bảng 2.7. Vòng lặp thứ 3 tính khoảng cách** 22

**Bảng 2.8. Vòng lặp thứ 3 phân cụm** 22

**Bảng 2.9. Tập dữ liệu đã được phân cụm** 44

**Bảng 2.10. Bảng so sánh 2 thuật toán** 44

# **PHẦN MỞ ĐẦU**

Quản lý dữ liệu lớn là một tập hợp các kỹ thuật được sử dụng để quản lý, khai thác và tìm kiếm các tri thức tiềm ẩn trong một tập hợp dữ liệu khổng lồ và phức tạp. Dữ liệu này có thể được thu thập từ nhiều nguồn khác nhau, như cơ sở dữ liệu, các tệp văn bản, các cảm biến, mạng xã hội, ... Nguồn dữ liệu của khách hàng và TTTM rất lớn, nếu áp dụng quản lý dữ liệu lớn trong kinh doanh TTTM sẽ mang lại rất nhiều lợi ích. Nên chúng em quyết định làm về quản lý dữ liệu khách hàng tại các khu trung tâm mua sắm.

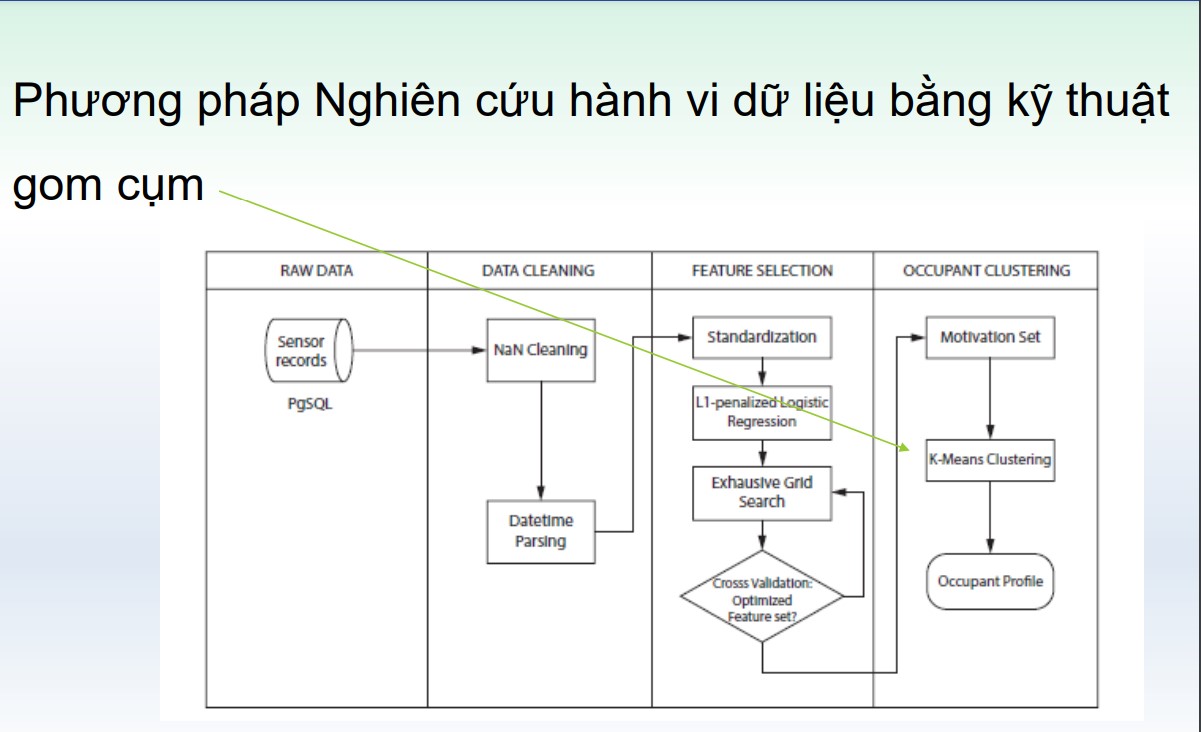
Ngày nay việc quản lý dữ liệu khách hàng tại các khu trung tâm mua sắm (TTTM) là một vấn đề quan trọng và được nhiều doanh nghiệp quan tâm. Bằng cách thu thập và phân tích dữ liệu khách hàng, các TTTM có thể hiểu rõ hơn về hành vi mua sắm, sở thích và nhu cầu của khách hàng, từ đó đưa ra các chiến lược kinh doanh hiệu quả hơn, tăng doanh thu, cải thiện trải nghiệm mua sắm, phát triển sản phẩm mới đáp ứng nhu cầu của thị trường. Trong thị trường ngày càng cạnh tranh, việc nắm bắt thông tin về khách hàng là yếu tố quyết định sự thành bại của doanh nghiệp. Ví dụ: Một TTTM có thể sử dụng dữ liệu để phân tích khách hàng thường mua sắm vào cuối tuần và mua các sản phẩm thời trang. Dựa vào thông tin này, TTTM có thể tổ chức các sự kiện đặc biệt vào cuối tuần và giới thiệu các bộ sưu tập mới.

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

## **1.1 CƠ SỞ LÝ THUYẾT**

### **1.1.1. Data Clustering**

Phân cụm là nhiệm vụ nhóm một tập hợp các đối tượng lại với nhau theo cách mà các đối tượng trong cùng một cụm giống nhau hơn so với các đối tượng trong các cụm khác. Độ tương tự là một số liệu phản ánh sức mạnh của mối quan hệ giữa hai đối tượng dữ liệu. Phân cụm chủ yếu được sử dụng để khai thác dữ liệu khám phá. Nó được sử dụng đa dạng trong nhiều lĩnh vực như học máy, nhận dạng mẫu, phân tích hình ảnh, truy xuất thông tin, tin sinh học, nén dữ liệu và đồ họa máy tính.



**Hình 1.1. Phương pháp gom cụm dữ liệu**

### **1.1.2. Các loại phương pháp gom cụm phổ biến**

- **Phân hoạch (Partitioning)**: Các phân hoạch được tạo ra và đánh giá theo một tiêu chí nào đó.

- **Phân cấp (Hierarchical)**: Phân rã tập dữ liệu/đối tượng có thứ tự phân cấp theo một tiêu chí nào đó.

**- Dựa trên mật độ (Density-based)**: Dựa trên độ kết nối (connectivity) và mật độ (density).

**- Dựa trên mô hình (Model-based)**: Một mô hình giả thuyết được đưa ra cho mỗi cụm; sau đó hiệu chỉnh các thông số để mô hình phù hợp với cụm dữ liệu nhất.

***Một số phương pháp gom cụm khác:***

**- Gom cụm cứng (Hard clustering)**: Mỗi đối tượng chỉ thuộc về một cụm.

+ Mức thành viên (Degree of membership) của mỗi đối tượng với một cụm hoặc là 0 hoặc là 1.

+ Ranh giới (Boundary) giữa các cụm rõ ràng.

**- Gom cụm mờ (Fuzzy clustering)**: Mỗi đối tượng thuộc về nhiều hơn một cụm.

+ Với mức thành viên nào đó từ 0 đến 1.

+ Ranh giới giữa các cụm không rõ ràng (mờ - vague/fuzzy).

### **1.1.3. Các phương pháp đánh giá chất lượng gom cụm**

**- Đánh giá ngoại (External validation)**:

+ Đánh giá kết quả gom cụm dựa vào cấu trúc được chỉ định trước cho tập dữ liệu.

+ Độ đo: Rand statistic, Jaccard coefficient, Folkes and Mallows index, …

**- Đánh giá nội (Internal validation)**:

+ Đánh giá kết quả gom cụm theo số lượng các vector của chính tập dữ liệu (ma trận

gần – proximity matrix).

+ Độ đo: Silhouette index, Dunn’s index, …

**- Đánh giá tương đối (Relative validation)**: Đánh giá kết quả gom cụm bằng việc so sánh các kết quả gom cụm khác ứng với các bộ trị thông số khác nhau => Tiêu chí đánh giá và chọn kết quả gom cụm tối ưu:

+ Độ nén (Compactness): Các đối tượng trong cụm nên gần nhau.

+ Độ phân tách (Separation): Các cụm nên xa nhau.

### **1.1.4. Thuật toán K-means dựa trên phân hoạch**

Một trong những phương pháp nghiên cứu hành vi dữ liệu bằng kỹ thuật gom cụm. Có rất nhiều mô hình để phân cụm ngoài kia. Mặc dù đơn giản nhưng K-means được sử dụng rộng rãi để phân cụm trong nhiều ứng dụng khoa học dữ liệu, đặc biệt hữu ích nếu bạn cần nhanh chóng khám phá thông tin chi tiết từ dữ liệu chưa được gắn nhãn. Phân cụm K-mean là một loại học tập không giám sát, được sử dụng khi bạn có dữ liệu chưa được gắn nhãn (nghĩa là dữ liệu không có danh mục hoặc nhóm được xác định). Mục tiêu của thuật toán này là tìm các nhóm trong dữ liệu, với số lượng nhóm được đại diện bởi biến K. Thuật toán hoạt động lặp đi lặp lại để gán từng điểm dữ liệu cho một trong K nhóm dựa trên các tính năng được cung cấp. Các điểm dữ liệu được phân cụm dựa trên tính tương tự của tính năng. Kết quả của thuật toán phân cụm K-mean là:

1. Trọng tâm của cụm K, có thể được sử dụng để dán nhãn dữ liệu mới.

2. Phân cụm cho từng dữ liệu huấn luyện (mỗi điểm dữ liệu được gán cho một cụm).

***- Ưu điểm:***

+ Bài toán tối ưu hóa với “Cực trị cục bộ”.

+ Mỗi cụm được đặc trưng hóa bởi trung tâm của cụm (đối tượng trung bình (mean)).

+ Số cụm k nên là bao nhiêu?

+ Độ phức tạp: O (nkt), với n là số đối được, k là số cụm, t là số lần lặp (k<<n, t<<n).

***- Nhược điểm:***

+ Ảnh hưởng bởi nhiễu (các phần tử kì dị/biên gọi là outlier).

+ Không phù hợp trong việc khai phá ra các cụm có dạng không lồi (nonconvex) hay các cụm có kích thước rất khác nhau.

+ Kết quả gom cụm có dạng siêu cầu (hyperspherial).

+ Kích thước các cụm thường không đồng đều (relative uniform sizes).

### **1.1.5. Áp dụng MapRuduce vào thuật toán K-means**

MapRuduce là một mô hình lập trình song song mạnh mẽ, đặc biệt hữu ích khi xử lý khối lượng lớn dữ liệu phân tán trên nhiều nút trong hệ thống phân tán như Hadoop. Việc áp dụng MapRuduce vào thuật toán K-means mang lại nhiều lợi ích:

* **Khả năng xử lý dữ liệu lớn**: K-means thường phải xử lý nhiều điểm dữ liệu, đặc biệt trong trường hợp dữ liệu lớn, MapRuduce chia dữ liệu thành các phần nhỏ và xử lý chúng song song, giúp giảm thời gian tính toán đáng kể
* **Tăng tốc độ tính toán**: Việc thực hiện song song các bước trong thuật toán K-means giúp giảm tải cho các máy tính đơn lẻ và tận dụng sưc mạnh tính toán của nhiều máy tính cùng lúc. Công việc được thực hiện song song làm cho tốc độ hội tụ của thuật toán những giảm đáng kể
* **Khả năng mở rộng**: MapRuduce cho phép hệ thống dễ dàng mở rộng để có thể xử lý khối lượng dữ liệu lớn hơn bằng cách thêm nhiều máy tính(nút) vào hệ thống. Điều này cực kỳ quan trọng trong việc xử lý dữ liệu lớn
* **Tối ưu hoá việc sử dụng tài nguyên**: MapReduce chia nhỏ dữ liệu và gửi chúng đến các nút tính toán khác nhau, giúp tính tối ưu tìa nguyên hệ thống, từ đó giảm thiểu tình trạng thắt cổ chai khi chỉ sử dụng một máy tính đơn lẻ

### **1.1.6. Các bước thực hiện MapRuduce với thuật toán K-means**

1. **Bước 1: Khởi tạo**
   * Chọn ngẫu nhiên K tâm cụm (centroid) ban đầu
2. **Bước 2: Mapping**
   * Với mỗi điểm dữ liệu trong tập dữ liệu:
     1. Tính khoảng cách từ điểm dữ liệu đến tất cả các tâm cụm hiện tại
     2. Gán điểm dữ liệu vào cụm có tâm cụm gần nhất
     3. Tạo ra các cặp Key-Value: Key là chỉ số của cụm và Value là điểm dữ liệu
   * Mục tiêu của pha này là nhóm các điểm dữ liệu theo các cụm tương ứng
3. **Bước 3: Reducing**
   * Với mỗi Key, hệ thống sẽ thực hiện:
     1. Tính toán trung bình của tất cả các điểm dữ liệu trong cụm để cập nhật lại tâm cụm
     2. Tạo ra cặp Key-Value mới: Khoá là chỉ số của cụm và giá trị là tâm cụm mới
4. **Bước 4: Lặp lại**
   * Sau khi hoàn thành một lần MapRuduce, thuật toán sẽ tiếp tục với Mapping mới, nơi các điểm dữ liệu được gán lại vào các cụm mới dựa trên các tâm cụm vừa được cập nhật
   * Thuật toán sẽ tiếp tục lặp lại cho đến khi:
     1. Các tậm cụm không thay đổi (hoặc thay đổi không đáng kể)
     2. Đạt đến số vòng lặp tối đa đã được đặt ra
5. **Bước 5: Hội tụ**
   * Thuật toán K-means sẽ dừng lại khi các tâm cụm không thay đổi nữa (hoặc thay đổi ít hơn ngưỡng cho phép), nghĩa là các điểm dữ liệu đã được phân cụm cố định

## **1.2. NỘI DUNG ĐỀ TÀI**

**Áp dụng kĩ thuật MapRuduce vào tính toán thuật toán K-Means:**

**-** Áp dụng được kĩ thuật tính toán MapReduce vào cho tính toán của thuật toán K-Means làm tăng tốc độ tính toán và tăng độ chính xác cho thuật toán.

**Phân cụm khách hàng:**

- Phân cụm khách hàng là quá trình gom nhóm khách hàng dựa trên bộ dữ liệu đã thu thập được từ khách hàng.

- Tùy vào mục tiêu phân cụm và các đặc tính chung của khách hàng (Thói quen mua sắm, hành vi của khách hàng trong khi mua sắm, tuổi tác, giới tính, …)

## **1.3. MỤC TIÊU NGHIÊN CỨU**

**1.3.1. Mục tiêu cho kỹ thuật**  
- Tiến hành chạy được thuật toán K-Means và đánh giá chất lượng phân cụm  
- Áp dụng được kỹ thuật MapReduce cho thuật toán K-Means với bộ dữ liệu  
- So sánh thời gian chạy của thuật toán K-means khi được áp dụng kỹ thuật MapRuduce và khi không được áp dụng kỹ thuật MapReduce

### **1.3.2. Nhu cầu đời sống**

- Xã hội phát triển, nhu cầu thị yếu của họ cũng nâng cao theo, chẳng hạn như phong phú trong trang phục, điều kiện ăn uống đa dạng.

- Dựa vào lý do trên, các doanh nghiệp đã xây dựng nên các khu trung tâm mua sắm đô thị chủ yếu tập trung ở các trung tâm thành phố lớn đông dân cư, du khách qua lại với đa dạng sản phẩm thiết yếu cho đời sống con người.

- Việc áp dụng bài toán có thể chia nhỏ khách hàng theo từng mục đích khác nhau, gợi ý những sản phẩm phù hợp dựa vào đặc tính của từng khách hàng. Tương tự như các doanh nghiệp, công ty lớn như Shopee, Điện Máy Xanh, CellphoneS, …

### **1.3.3. Lợi ích cho doanh nghiệp**

- Hành động: Đưa ra các chiến lược sản xuất hàng hóa, tiếp thị, quản bá sản phẩm tùy theo nhóm khách hàng dựa trên đặc tính của họ.

=> Từ đó, các doanh nghiệp nhận được những lợi ích như:

+ Tiết kiệm chi phí sản xuất, vận chuyển.

+ Tăng độ uy tín, chất lượng của công ty.

+ Giữ chân các khách hàng quan trọng ảnh hưởng trực tiếp đến doanh thu công ty.

+ Tăng số lượng khách hàng từ phương thức tiếp thị, marketing.

+ Nâng cao lãi suất công ty, nâng cao giá cả mặc hàng dựa trên độ chất lượng của thương hiệu.

* Vì thế, công ty của doanh nghiệp phát triển hay xuống dốc phụ thuộc vào độ hảo cảm của khách hàng, buộc họ phải quyết định các chiến lược đúng đắn nhằm giảm thiểu kinh phí, tăng số doanh thu, tăng lượng khách hàng, tạo uy tín cho công ty. Và hơn thế nữa, việc áp dụng được kỹ thuật MapReduce vào thuật toán còn làm giảm thời gian chạy của thuật toán. Do đó, nhóm em đã chọn đề tài “**Áp dụng MapReduce vào thuật toán K-Means để phân cụm khách hàng tại trung tâm mua sắm**”.

## **1.4. ĐỐI TƯỢNG NGHIÊN CỨU**

- Đối tượng là thông tin của 200 khách hàng đến trung tâm mua sắm là một phần của cuộc khảo sát cuộc thi dữ liệu phân khúc khách hàng (Mall Customer Segementation Data) được tổ chức trên trang Kaggle.

- Nghiên cứu được thực hiện trên 200 đối tượng là các khách hàng mua sắm tại trung tâm với 4 đặc điểm về giới tính, độ tuổi, mức thu nhập và điểm chỉ tiêu dựa vào hành vi mua sắm và chi trả.

## **1.5. KẾT QUẢ MONG ĐỢI**

- Khi đã có ý tưởng xây dựng thì chắc chắn phải có kết quả mong muốn:

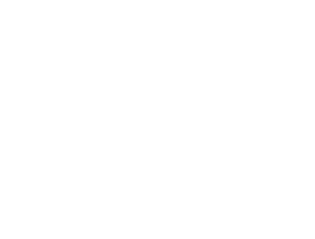
+ Gom nhóm khách hàng đạt độ tin cậy tối ưu nhất tùy vào các thuật toán gom cụm.

+ Tập dữ liệu mới các khách hàng đã được phân thành từng cụm rõ ràng.

+ Thuật toán được vận hành trơn tru với sự kết hợp của kỹ thuật MapRuduce

# **CHƯƠNG 2: MÔ HÌNH THỰC NGHIỆM**

## **2.1. QUY TRÌNH GOM CỤM DỮ LIỆU**



**CLUSTERING**

**ALGORITHMS**

**SELLECTION**



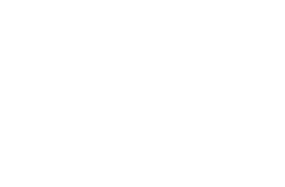
Database



Data Mart



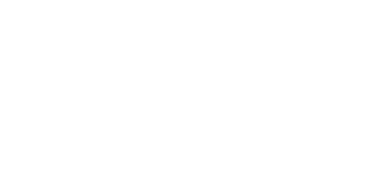
*ETL*



**INITIAL**

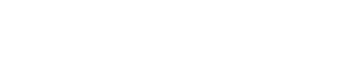
**CLUSTERING**

**RESULTS**

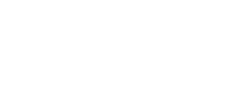


**RESULTS**

**INTERPRETATION**



***KNOWLEDGE***



*Clusters*

*Validation*

**Hình 2.1. Gom cụm dữ liệu**

*Trong đó:*

**Database**: Tập cơ sở dữ liệu

**Data Mart**: Tập dữ liệu đã qua xử lý và trích chọn thuộc tính hay còn gọi là thuật ngữ ETL (Extract-Tranform-Load).

**Clustering Algorithms Sellection**: Lựa chọn các giải thuật phân cụm để khai phá dữ liệu.

**Initial Clustering Results**: Kết quả phân cụm ban đầu với các tham số chưa tối ưu **Results Interpretation**: Biểu diễn phân cụm đã được tối ưu qua các phương pháp tìm k, tìm trung tâm cụm, khoảng cách.

**Knowledge**: Đưa ra cụm tốt nhất cho một input mới với features áp dụng giải thuật.

## **2.2. THU THẬP DỮ LIỆU**

**Bảng 2.1. Thông tin khách hàng tại khu trung tâm mua sắm thu thập từ cuộc thi Kaggle**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CustomerID** | **Gender** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| 1 | Male | 19 | 15 | 39 |
| 2 | Male | 21 | 15 | 81 |
| 3 | Female | 20 | 16 | 6 |
| 4 | Female | 23 | 16 | 77 |
| 5 | Female | 31 | 17 | 40 |
| 6 | Female | 22 | 17 | 76 |
| 7 | Female | 35 | 18 | 6 |
| 8 | Female | 23 | 18 | 94 |
| 9 | Male | 64 | 19 | 3 |
| 10 | Female | 30 | 19 | 72 |
| 11 | Male | 67 | 19 | 14 |
| 12 | Female | 35 | 19 | 99 |
| 13 | Female | 58 | 20 | 15 |
| 14 | Female | 24 | 20 | 77 |
| 15 | Male | 37 | 20 | 13 |

*Dữ liệu bao gồm những thuộc tính sau:*

**1. CustomerID**: Mô tả mã ID của khách hàng khi tham gia vào khu trung tâm mua sắm (từ 1 đến 200). Kiểu dữ liệu: số nguyên

**2. Gender**: Mô tả giới tính của mỗi khách hàng gồm hai giá trị là Male (nam) và

Female (nữ). Theo như thống kê cho thấy phái nữ chiếm phần lớn hơn phái nam.

Kiểu dữ liệu: đối tượng, chuỗi

**3. Age**: Mô tả độ tuổi của khách hàng, phân bố ở nhiều lứa tuổi khác nhau. Kiểu dữ liệu: số nguyên

**4. Annual Income (k$)**: Mô tả số lượng thu nhập hằng năm của mỗi khách hàng có đơn vị là k(nghìn)$/năm, trung bình rơi vào khoảng 50k đến 85k trên năm. Kiểu dữ liệu: số nguyên

**5. Spending Score (1-100)**: Điểm chỉ tiêu của khách hàng tại khu trung tâm mua sắm, dựa trên hành vi mua sắm, tần suất mua sản phẩm, … (Số điểm nhiều nhất của khách hàng từ 40 đến 60 điểm). Kiểu dữ liệu: số nguyên

## **2.3. MÔ HÌNH THUẬT TOÁN K-MEANS**

### **2.3.1. Ý tưởng**

Thuật toán gom cụm K-means dựa trên phân hoạch, trong đó Chia dữ liệu thành các cụm là tập hợp các điểm gần nhau trong một không gian nào đó. Mỗi cụm sẽ có một điểm trung tậm (centroid), với mỗi điểm của dữ liệu, nếu nó gần điểm trung tâm cụm nào hơn thì thuộc về cụm đó.

- **Input:**

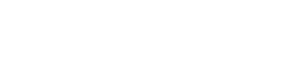
+ k: Số k cụm

+ D: Một tập dữ liệu chứa n đối tượng.

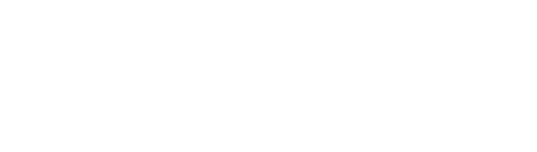
- **Output:** Tập dữ liệu khách hàng đã được phân thành số k cụm.

### **2.3.2. Mô hình thuật toán**

**Dịch chuyển các object đến các cluster**

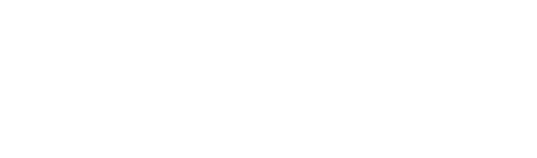


**Bắt đầu**



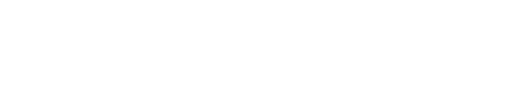
**Tìm số k clusters**

**để phân cụm**



**Phân cụm dựa trên khoảng cách**

**ngắn nhất**

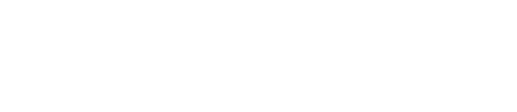


**Xác định các điểm trung tâm**

**)**

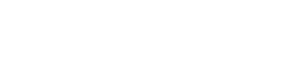
**(**

**Centroid**



**Tính khoảng cách các object**

**đến Centroid**



**Kết thúc**



**Không**



**Có**

**Hình 2.2. Flowchart giải thuật K-means**

### **2.3.3. Thuật toán**

1. Chọn k điểm bất kỳ làm các trung tâm cụm ban đầu từ D.

2. Lặp lại.

3. Phân mỗi điểm dữ liệu vào cluster có khoảng cách đến điểm trung tâm là nhỏ nhất cho đến khi nó không thay đổi

4. Cập nhật lại tâm cho từng cụm bằng cách tính trung bình cộng của tất cả các điểm dữ liệu trong từng cụm đó.

5. Lặp lại vòng lặp cho đến khi không còn thay đổi hay còn gọi lại hội tụ.

### **2.3.3. Mô hình thuật toán K-Means khi áp dụng với kỹ thuật MapReduce**

**Input**

**Thu thâp dữ liệu cần phân cụm. Khởi tạo K tâm cụm**

**Splitting**

**Chia tập dữ liệu thành 1 hoặc một nhóm các điểm**

**Mapping**

**Tính khoảng cách các điểm đến tâm cụm.   
Tạo cặp key/value, với key là trọng tâm gần nhất, value là toạ độ điểm đang xét**

**Shuffling**

**Gom các dữ liệu có cùng key thành 1 nhóm**

**Reducing**

**Tính toạ độ tâm mới bằng cách tính trung bình cộng các điểm đã được gán trọng tâm**

**Lặp lại và kiểm tra điểm dừng**

**Gán tâm mới vào tâm khởi tạo ban đầu và thực hiện vòng lặp với.  
Điều kiện dừng khi các tâm cũ không đổi so với tâm mới**

## **2.4. CÔNG THỨC TOÁN HỌC**

- Phép đo khoảng cách euclidean:

𝑑(𝑥, 𝑚𝑖) =

*Trong đó:*

+ x: Mỗi điểm giá trị trong tập dữ liệu (1<=i<=n)

+ mi: Trọng tâm cụm thứ i, i = (1,k)

- Xác định điểm trung tâm của cluster:

- Thuật toán tối ưu hàm mất mát:

argmin chính là giá trị của biến số để hàm số đó đạt giá trị nhỏ nhất.

chính là bình phương khoảng cách từ điểm xi tới tâm mj, mỗi điểm xi thuộc về cluster có trọng tâm gần nó nhất.

- Để đánh giá hiệu suất phân cụm, nhóm em sử dụng phương pháp **silhouette score:**

*Trong đó:*

+ S(i) là hệ số bống của dữ liệu điểm i.

+ a(i) là khoảng cách trung bình giữa i và tất cả các điểm dữ liệu khác trong cụm mà i thuộc về b(i).

+ b(i) là khoảng cách trung bình giữa i đến tất cả các cụm mà i không thuộc về.

* Nếu điểm S(i) đạt điểm dần về 1 nghĩa là thuật toán phân cụm tốt
* Ngược lại S(i) đạt điểm nhỏ hơn 0, dần về -1 nghĩa là thuật toán phân cụm sai
* Còn S(i) gần bằng 0, thì dữ liệu đang nằm giữa 2 cluster

Để chuẩn hóa dữ liệu, nhóm em sử dụng phương pháp thống kê dựa trên hàm StandardScaler. Chuẩn hóa các tính năng bằng cách loại bỏ giá trị trung bình và chia tỷ lệ thành phương sai đơn vị.

Công thức chuẩn hóa như sau:

*Trong đó:*

𝑣à 𝜎 lần lượt kỳ vọng và phương sai (standard deviation) của thành phần đó trên toàn bộ dữ liệu được huấn luyện.

## **2.5. THỰC NGHIỆM VỚI DỮ LIỆU MẪU**

Trong bài toán phân khúc khách hàng, tùy vào mục đích thị yếu của công ty mà mình khai phá bài toán dựa trên mục đích ấy, với bài toàn này, em chỉ phân tích hai thuộc tính Annual Income (k$) và Spending Score (1-100). Vì em lấy dữ liệu khá ít nên em sẽ tạo k cụm nhỏ hơn bài toán.

**Bảng 2.2. Bảng thực nghiệm với 5 dòng từ dữ liệu gốc chưa chuẩn hóa**

|  |  |  |
| --- | --- | --- |
|  | **Annual Income (k$)** | **Spending Score (1-100)** |
| A | 15 | 39 |
| B | 15 | 81 |
| C | 16 | 6 |
| D | 16 | 77 |
| E | 17 | 40 |

* Khởi tạo tâm cho 2 cụm:

C1(15, 39)

C2(17, 40)

* Áp dụng tính khoảng cách Euclidean:

d(A, C1) == 0

d(B, C1) = = 42.012

d(C, C1) = = 33.015

d(D, C1) = = 38.013

d(E, C1) = = 2.236

* Tương tự tính:

d(A, C2) = = 2.236

d(B, C2) = = 41.049

d(C, C2) = = 34.015

d(D, C2) = = 37.014

d(E, C2) = = 0

**Bảng 2.3. Vòng lặp thứ nhất tính khoảng cách**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iterate 1 | A | B | C | D | E |  |
| 0 | 42.012 | 33.015 | 38.013 | 2.236 | C1(15,39) |
| 2.236 | 41.049 | 34.015 | 37.014 | 0 | C2(17,40) |

* Gọi 0 là điểm không thuộc nhóm

Gọi 1 là điểm thuộc nhóm

Ta xét giá trị min của A, B, C, D, E

**Bảng 2.4. Vòng lặp thứ nhất phân cụm**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G1 |  | A |  | B |  | C |  | D |  | E |  |
|  | 1 |  | 0 |  | 1 |  | 0 |  | 0 | C1(15,39) |
|  | 0 |  | 1 |  | 0 |  | 1 |  | 1 | C2(17,40) |

* Nhóm 1 gồm: A(15, 39); C(16,6)

Nhóm 2 gồm: B(15, 81); D(16, 77); E(17, 40)

* Tính lại điểm trung tâm:

C1((15+16)/2, (39+6)/2) => C1(15.5, 22.5)

C2((15+16+17)/3, (81+77+40)/3) => C2(16, 66)

* Vòng lặp 2:

d(A, C1) = = 16.508

d(B, C1) = = 58.502

d(C, C1) = = 16.508

d(D, C1) = = 54.502

d(E, C1) = = 17.564

* Tương tự tính:

d(A, C2) = = 27.019

d(B, C2) = = 15.033

d(C, C2) = = 60

d(D, C2) = = 11

d(E, C2) = = 26.019

**Bảng 2.5. Vòng lặp thứ 2 tính khoảng cách**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iterate 2 | A | B | C | D | E |  |
| 16.508 | 58.502 | 16.508 | 54.502 | 17.564 | C1(15.5,22.5) |
| 27.019 | 15.033 | 60 | 11 | 26.019 | C2(16,66) |

* Ta xét giá trị min của A, B, C, D, E

**Bảng 2.6. Vòng lặp thứ 2 phân cụm**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G2 |  | A |  | B |  | C |  | D |  | E |  |
|  | 1 |  | 0 |  | 1 |  | 0 |  | 1 | C1(15.5,22.5) |
|  | 0 |  | 1 |  | 0 |  | 1 |  | 0 | C2(16,66) |

* Nhóm 1 gồm: A(15, 39); C(16, 6); E(17, 40)

Nhóm 2 gồm: B(15, 81); D(16, 77)

* Tính lại điểm trung tâm:

C1((15+16+17)/3, (39+6+40)/3) => C1(16, 28.333)

C2((15+16)/2, (81+77)/2) => C2(15.5, 79)

* Vòng lặp 3:

d(A, C1) = = 10.714

d(B, C1) = = 52.676

d(C, C1) = = 22.333

d(D, C1) = = 48.667

d(E, C1) = = 11.710

* Tương tự tính:

d(A, C2) = = 40.003

d(B, C2) = = 2.062

d(C, C2) = = 73.002

d(D, C2) = = 2.062

d(E, C2) = = 39.029

**Bảng 2.7. Vòng lặp thứ 3 tính khoảng cách**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iterate 3 | A | B | C | D | E |  |
| 10.714 | 52.676 | 22.333 | 48.667 | 11.710 | C1(16,28.333) |
| 40.003 | 2.062 | 73.002 | 2.062 | 39.029 | C2(15.5,79) |

* Ta xét giá trị min của A, B, C, D, E

**Bảng 2.8. Vòng lặp thứ 3 phân cụm**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G3 |  | A |  | B |  | C |  | D |  | E |  |
|  | 1 |  | 0 |  | 1 |  | 0 |  | 1 | C1(16,28.333) |
|  | 0 |  | 1 |  | 0 |  | 1 |  | 0 | C2(15.5,79) |

* Ở vòng lặp thứ 3, G2 và G3 không thay đổi.
* Nhóm 1 có 3 điểm A(15,39), C(16, 6), E(17, 40) với C1(16, 28.333) Nhóm 2 có 2 điểm B(15,81), D(16,77) với C2(15.5, 79)
* Ngừng thuật toán.

## **2.6. ỨNG DỤNG**

* *Các công cụ thư viện được dùng trong bài toán:*

**Numpy**: Hỗ trợ tính toán trên dữ liệu đã được làm sạch.

**Pandas**: Là thư viện dùng để phân tích dữ liệu, thao tác hầu như liên quan đến dữ liệu và đặc biệt là cho phép nhập dữ liệu với nhiều định dạng khác nhau.

**Matplotlib**: Biểu diễn dữ liệu đã được xử lý.

**Seaborn**: Là thư viện tạo đồ họa thống kê. Được xây dựng dựa trên thư viện matplotlib.

**Scikit-learn**: Hỗ trợ nhiều mô hình dữ liệu cần thiết trong quá trình khai phá tri thức tìm ẩn, cũng có các mô hình tiền xử lý dữ liệu khác nhau.

**Yellowbrick**: phiên bản mở rộng API Scikit-Learn để giúp lựa chọn mô hình và điều chỉnh siêu tham số dễ dàng hơn. Về cơ bản, nó đang sử dụng Matplotlib.

**Spicy**: được xây dựng dựa trên thư viện Numpy, cung cấp thao tác mảng N chiều thuận tiện và nhanh chóng. SciPy gồm các gói con (submodule) cho đại số tuyến tính, tối ưu hóa, tích hợp và thống kê bao gồm cả các thuật toán gom cụm.

* *Import thư viện cần thiết cho bài toán:*

import pandas as pd

import numpy as np

import matplotlib pyplot as plt

import seaborn as sns from sklearn

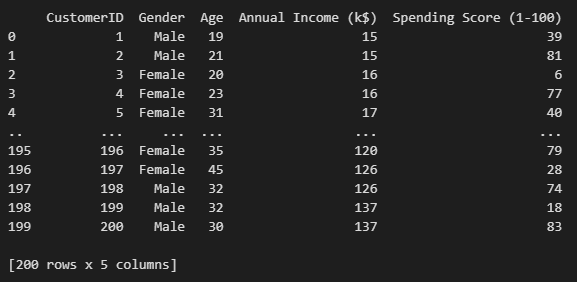
import metrics

* *Nhập dữ liệu vào bằng thư viện pandas với biến gán là dataset:*

dataset = pd.read\_csv(r"D:\BigData And DataMining\DataMining\DataMining\_Project\Mall\_Customers.csv")

print(dataset)

* *Output:*



**Hình 2.3. Bảng dữ liệu**

* *Vẽ đồ thị phân tích thuộc tính Gender:*

plt.figure(figsize=(15,5))

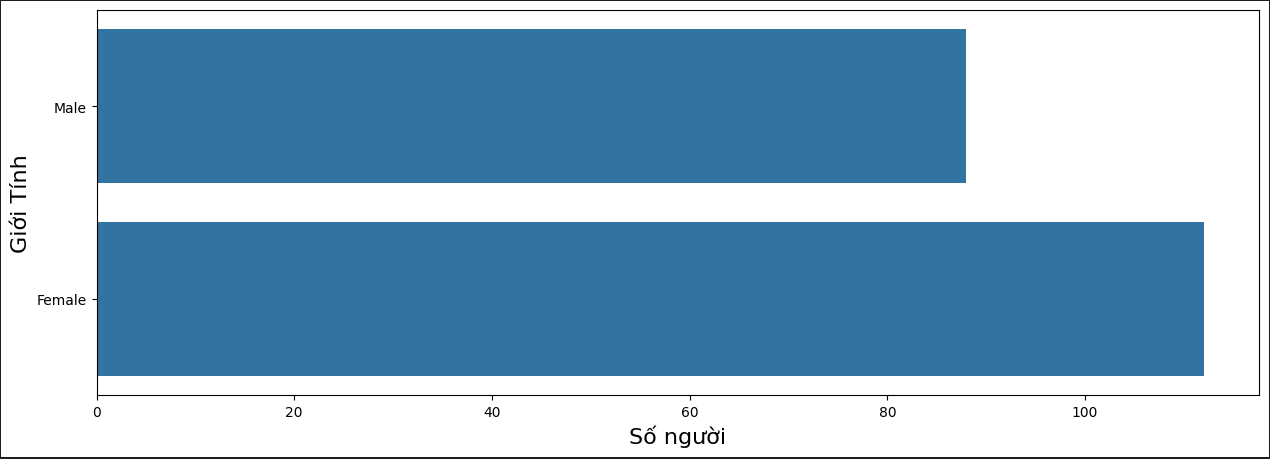
sns.countplot(y=’Gender’, data=dataset)

plt.xlabel(‘Số người’, fontsize=16)

plt.ylabel(‘Giới Tính’, fontsize=16)

plt.show()

* *Output:*



**Hình 2.4. So sánh tỉ lệ Male (Nam), Female (Nữ) của thuộc tính Gender**

* *Vẽ đồ thị phân tích mật độ của 3 thuộc tính còn lại:*

plt.figure(figsize = (20, 8))

plotnumber = 1

for col in dataset[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']]:

    if plotnumber <= 3:

        ax = plt.subplot(1, 3, plotnumber)

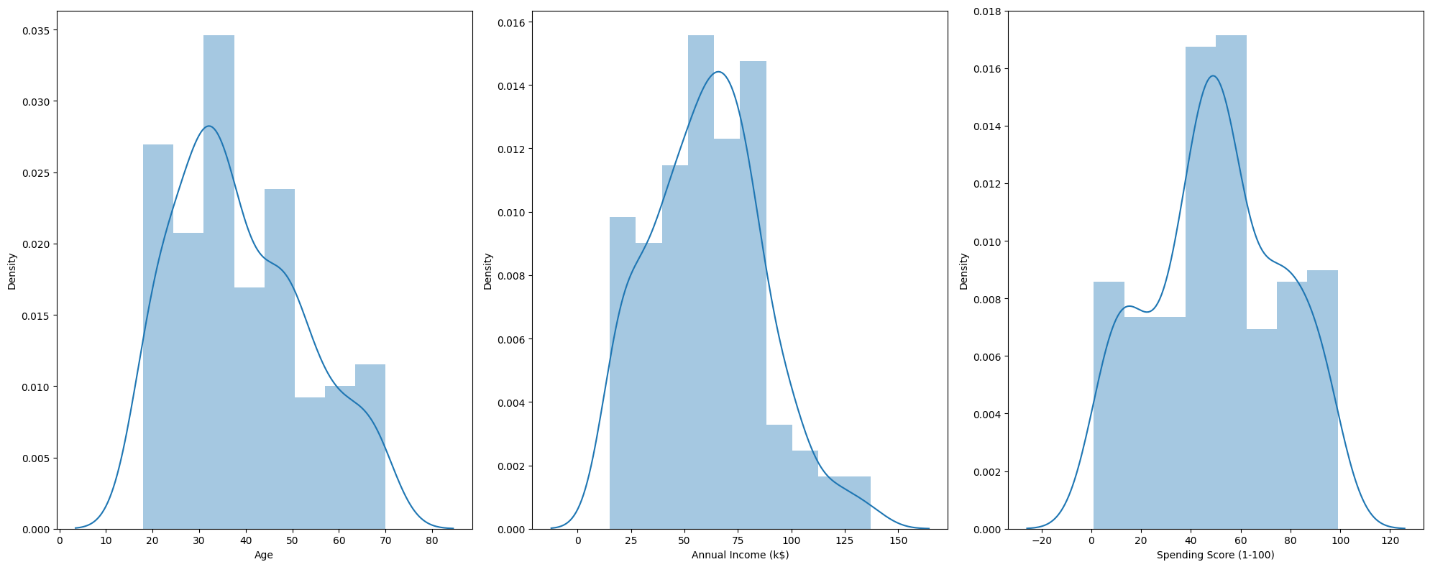
        sns.distplot(dataset[col])  #displot sẽ bị hư hình mà là distplot mới đúng

    plotnumber += 1

plt.tight\_layout() # Sắp xếp bố cục hình

plt.show()

* *Output:*



**Hình 2.5. Thông tin của Age, Annual Income (k$), Spending Score (1-100)**

* *So sánh mối tương quan giữa 2 thuộc tính với nhau trong tập dữ liệu:*

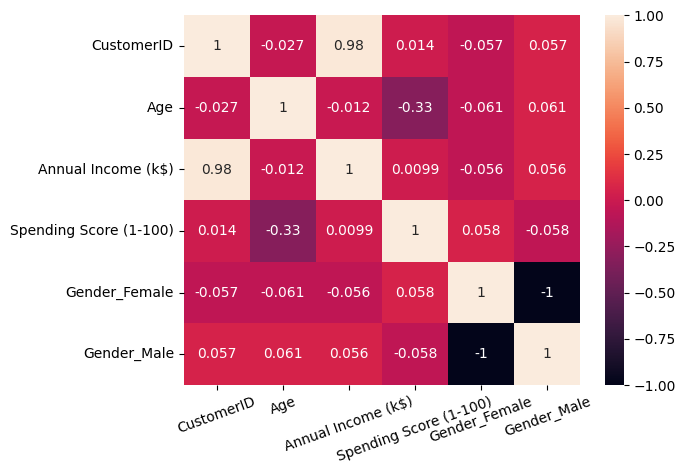
corr\_matrix = dataset.corr()

display(corr\_matrix)

sns.heatmap(corr\_matrix, annot=True)

\_ = plt.xticks(rotation=20)

* *Output:*



**Hình 2.6. Hệ số tương quan**

* *Kiểm tra dữ liệu có rỗng hay không, mô tả kiểu dữ liệu với hàm dataset.info(), liệt kê số (dòng, cột) với hàm dataset.shape:*

for i in dataset:

    missing\_value = dataset[i].isnull().sum()

    missing\_percent = missing\_value/len(dataset)\*100

    print(f"Cột {i}: có {missing\_percent}% giá trị thiếu")

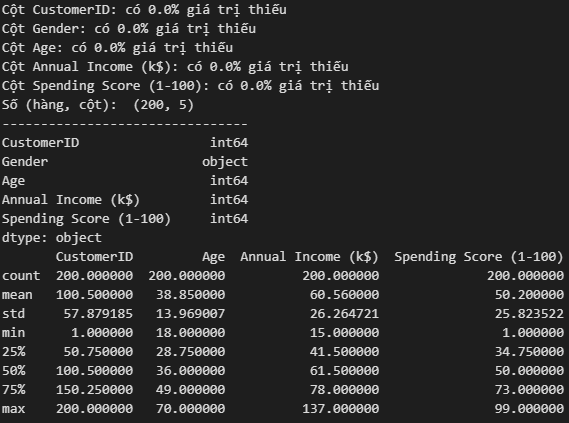
print("Số (hàng, cột): ",dataset.shape)

print('--------------------------------')

print(dataset.dtypes)

print(dataset.describe())

* *Output:*



**Hình 2.7. Số liệu thống kê**

* *Biểu diễn số liệu thống kê của các thuộc tính:*

print(dataset.describe())

* *Output:*



**Hình 2.8. Số liệu thống kê**

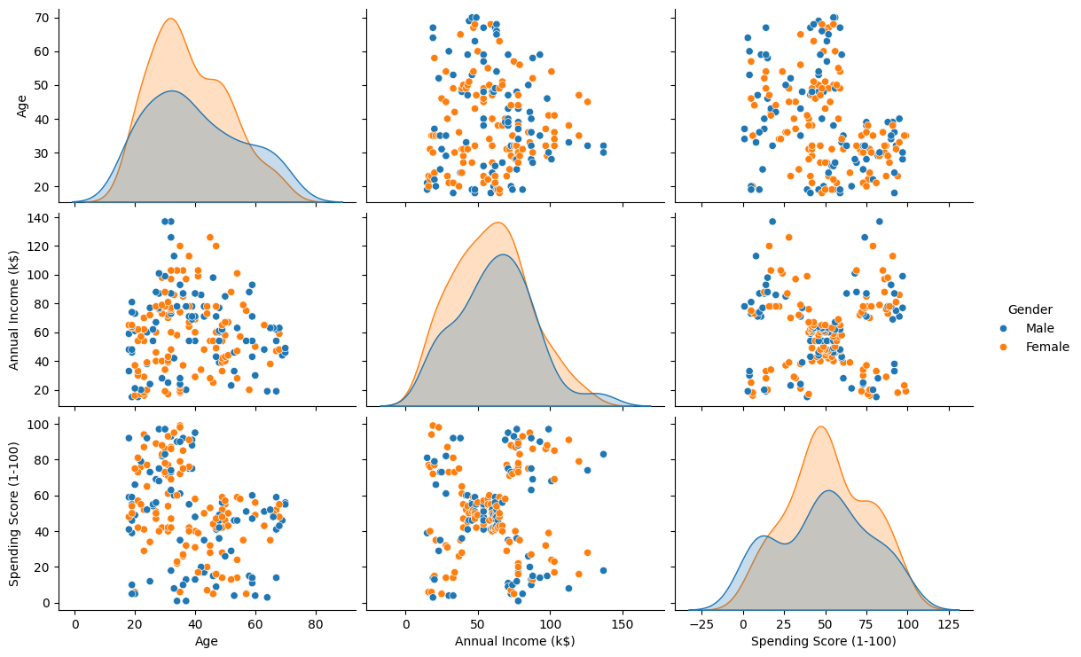
* *Mối quan hệ giữa thuộc tính Gender, Annual Income (k$), Spending Score (1-100) dựa trên thuộc tính Gender:*

vs = dataset.drop(['CustomerID', 'Gender'], axis=1)

sns.pairplot(dataset.drop(['CustomerID'], axis=1), hue='Gender', aspect=1.5)

plt.show()

* *Output:*



**Hình 2.9. Sự phân bố của dữ liệu đối với các thuộc tính**

* *Tương tự với kiểu biểu diễn khác:*

plt.figure(1 , figsize = (15 , 10))

n = 0

for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:

    for y in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:

        n += 1

        plt.subplot(3 , 3 , n)

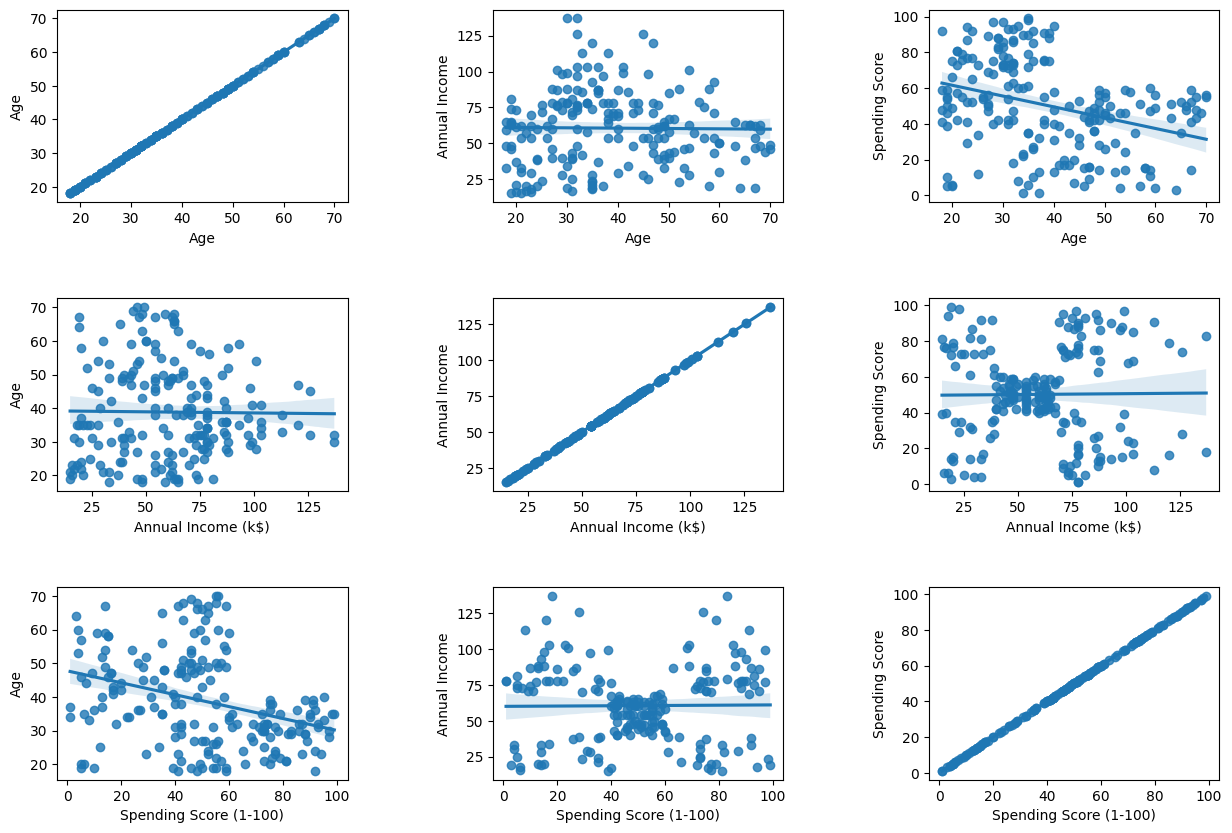
        plt.subplots\_adjust(hspace = 0.5 , wspace = 0.5)

        sns.regplot(x = x , y = y , data = dataset)

        plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1 else y )

plt.show()

* *Output:*



**Hình 2.10. Mối quan hệ giữa Age, Annual Income (k$), Spending Score (1-100)**

* *Trích chọn thuộc tính để giải quyết bài toán:*

X = dataset.iloc[:,[3,4]].values #hay .to\_numpy() đều được dùng để chuyển dataframe sang dạng numpy array cho bước khai phá dữ liệu.

* *Chuẩn hóa dữ liệu với hàm StandardScaler trong thư viện sklearn.preprocessing:*

from sklearn.preprocessing import StandardScaler

std = StandardScaler()

X\_std = std.fit\_transform(X)

print(X\_std[:10])

* *Import giải thuật phân cụm K-means:*

from sklearn.cluster import KMeans

* *Giả sử ta chưa thể xác định được cụ thể số k cụm là bao nhiêu nên ta sẽ cho vào vào thử là 3 cụm:*

kmean\_3 = KMeans(n\_clusters=3)

kmean\_3.fit(X\_std)

label\_3 = kmean\_3.labels\_

print(label\_3)

print(kmean\_3.cluster\_centers\_)

* *Sử dụng Silhouette Score đề đánh giá:*

from sklearn import metrics

s = metrics.silhouette\_score(X\_std, label\_3, metric='euclidean')

print("Đánh giá theo phương pháp Silhouette Score với k=3: ", s)

#rate2 = metrics.calinski\_harabasz\_score(X\_std, label\_3)

#print("Đánh giá loại 2: ", rate2)

* *Output:*



**Hình 2.11. Chất lượng gom cụm của thuật toán K-means với 3 cụm**

* *Biểu diễn phân cụm 3 lên đồ thị:*

plt.figure(figsize=(10,8))

plt.scatter(

    X\_std[label\_3 == 0, 0], X\_std[label\_3 == 0, 1],

    c = 'lightgreen', marker= 's', edgecolors='black',

    label='cluster 1')

plt.scatter(

    X\_std[label\_3 == 1, 0], X\_std[label\_3 == 1, 1],

    c = 'orange', marker= 'o', edgecolors='black',

    label='cluster 2')

plt.scatter(

X\_std[label\_3 == 2, 0], X\_std[label\_3 == 2, 1],

c = 'lightblue', marker= 'v', edgecolors='black',

label='cluster 3')

plt.scatter(

kmean\_3.cluster\_centers\_[:,0], kmean\_3.cluster\_centers\_[:,1],

c = 'red', s=300, marker= '\*', edgecolors='black',

label='centroids')

plt.xlabel('Spending Score (1-100)')

plt.ylabel('Annual Income (k$)')

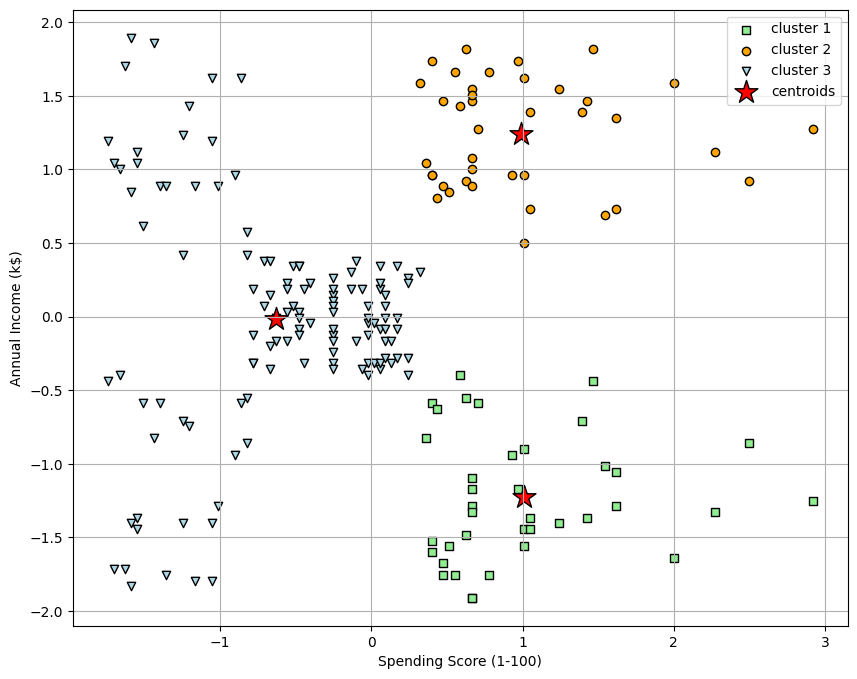
plt.legend(scatterpoints=1) # Chú thích trong biểu đồ

plt.grid() # Bỏ qua khung caro

plt.show()

print("[Trục X Trục Y]\n",kmean\_3.cluster\_centers\_)

- *Output*:



* *Tiến hành việc tìm k cụm với phương pháp Elbow bằng việc tính toán SSE (Sum Squared Error)=WSSE (Within Sum Squared Error):*

sum\_distances = []

for i in range(1, 11):

    kmeans = KMeans(n\_clusters=i, random\_state=0) #Random\_state cố định kết quả ra như nhau nếu ko sẽ cho ra kết quả trong mỗi lần chạy là khác nhau

    kmeans.fit(X\_std)

    sum\_distances.append(kmeans.inertia\_) #inertia\_ là khoảng cách giữa điểm chính giữa đến những dữ liệu xung quanh within-cluster sum-of-squares (tổng bình phương trong cụm)

* *Vẽ đường tìm k cụm:*

plt.figure(figsize=(10,8))

plt.plot(np.arange(1,11), sum\_distances, 'bs-')

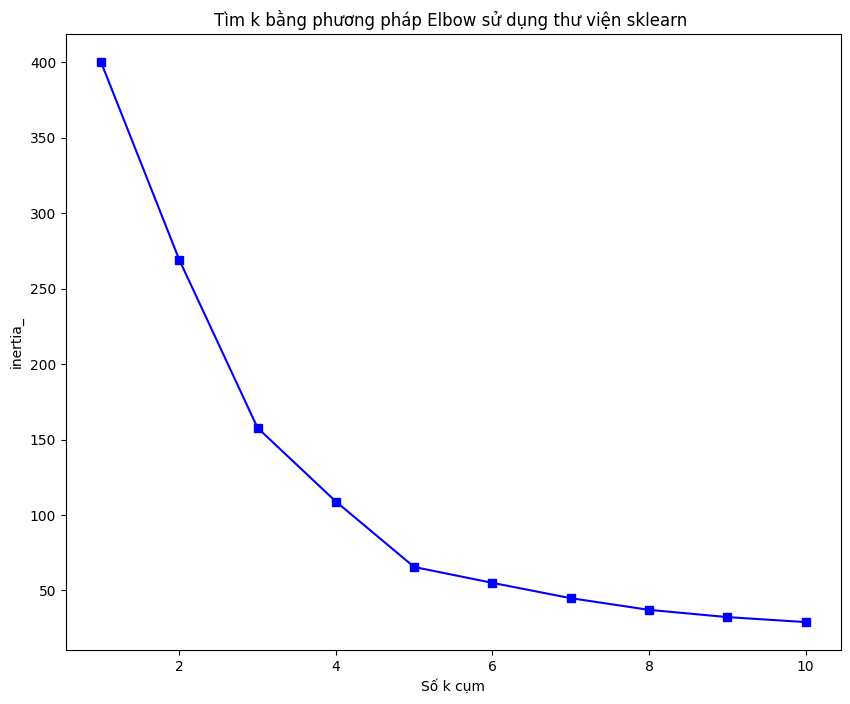
plt.title('Tìm k bằng phương pháp Elbow sử dụng thư viện sklearn ')

plt.xlabel('Số k cụm')

plt.ylabel('inertia\_')

plt.show()

* *Output:*



**Hình 2.12. Tìm k cụm với phương pháp Elbow**

* Ta có thể thấy từ điểm số cụm 5 trở đi inertia gần như không thay đổi hay còn gọi là hội tụ.
* *Sau khi đã xác định được số k cụm, việc tiếp theo là tiến hành xây dựng model thuật toán với số cụm bằng 5:*

kmean\_5 = KMeans(n\_clusters=5, init='k-means++', n\_init=20, max\_iter=200, random\_state=0, algorithm='lloyd') #init xác định điểm trung tâm dựa trên phân phối xác suất có điều kiện cho các cụm mặc định là k-means++

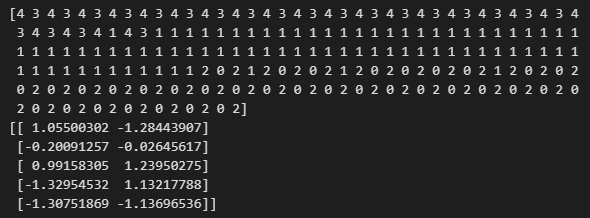
model = kmean\_5.fit(X\_std)

results = kmean\_5.predict(X\_std) # Dùng kmean\_5.label\_ đều được

print(results)

print(model.cluster\_centers\_)

* *Output:*



**Hình 2.13. Số cluster và tâm cụm**

* *Trong đó:*

**n\_clusters**: Hàm truyền tham số cụm. Mặc định là 8.

**init = ‘k-means++’**: Chọn trọng tâm cụm ban đầu bằng cách sử dụng lấy mẫu dựa trên phân phối xác suất theo kinh nghiệm của sự đóng góp của các điểm vào quán tính(inertia) tổng thể. Kỹ thuật này tăng tốc độ hội tụ. **n\_init**: Số lần thuật toán k-means được chạy với các centroid seed khác nhau. Kết quả cuối cùng là đầu ra tốt nhất của n\_init lần chạy liên tiếp theo quán tính. Mặc định là 10.

**algorithm='lloyd'**: Thuật toán được k-means sử dụng để giải quyết bài toán ngoài ra còn có elkan.

**max\_iter**: Số lần lặp tối đa của thuật toán k-means cho một lần chạy. mặc định là 300.

**random\_state**: cố định kết quả trong mỗi lần chạy. thường người ta sẽ cho là 0, 42.

* *Nếu việc tìm k cụm với phương pháp Elbow không đáng tin cậy ta sẽ thêm hàm đánh giá chất lượng gom cụm nội với phương pháp Silhouette Score:*

k\_values = []

silhouette\_score\_values = []

for i in range(2, 11):

    kmeanss = KMeans(n\_clusters=i)

    y\_pred = kmeanss.fit\_predict(X\_std)

    k\_values.append(i)

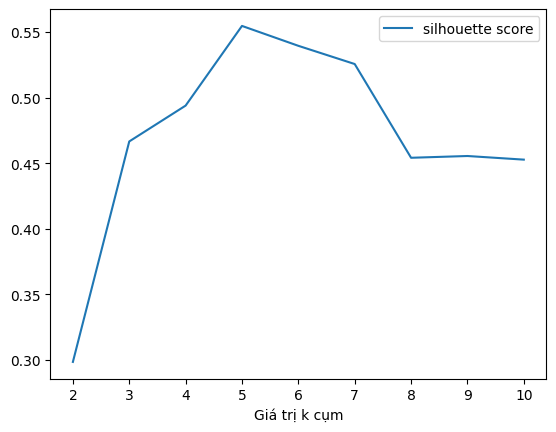
    silhouette\_score\_values.append(metrics.silhouette\_score(X\_std, y\_pred, metric='euclidean'))

res = pd.DataFrame({'Giá trị k cụm': k\_values, 'silhouette score': silhouette\_score\_values})

res.plot.line(x='Giá trị k cụm', y='silhouette score')

plt.show()

* *Output:*



**Hình 2.14. Tìm k cụm với phương pháp Silhouette Coefficient**

* *Tương tự ta đánh giá chất lượng 5 cụm:*

rate2 = metrics.silhouette\_score(X\_std, results, metric='euclidean')

print("Đánh giá theo phương pháp Silhouette Score với k=5: ", rate2)

* *Output:*



**Hình 2.15. Độ chính xác của thuật toán đối với cụm bằng 5**

* *Trực quan đồ thị:*

plt.figure(figsize=(10,8))

plt.scatter(

    X\_std[results == 0, 0], X\_std[results == 0, 1],

    c = 'lightgreen', marker= 's', edgecolors='black',

    label='cluster 1')

plt.scatter(

    X\_std[results == 1, 0], X\_std[results == 1, 1],

    c = 'orange', marker= 'o', edgecolors='black',

    label='cluster 2')

plt.scatter(

    X\_std[results == 2, 0], X\_std[results == 2, 1],

    c = 'lightblue', marker= 'v', edgecolors='black',

    label='cluster 3')

plt.scatter(

    X\_std[results == 3 , 0], X\_std[results == 3, 1],

    c = 'yellow', marker= 'd', edgecolors='black',

    label='cluster 4')

plt.scatter(

    X\_std[results == 4 , 0], X\_std[results == 4, 1],

    c = 'black', marker= 'x',

    label='cluster 5')

plt.scatter(

    model.cluster\_centers\_[:,0], model.cluster\_centers\_[:,1],

    c = 'red', s=300, marker= '\*', edgecolors='black',

    label='centroids')

plt.xlabel('Spending Score (1-100)')

plt.ylabel('Annual Income (k$)')

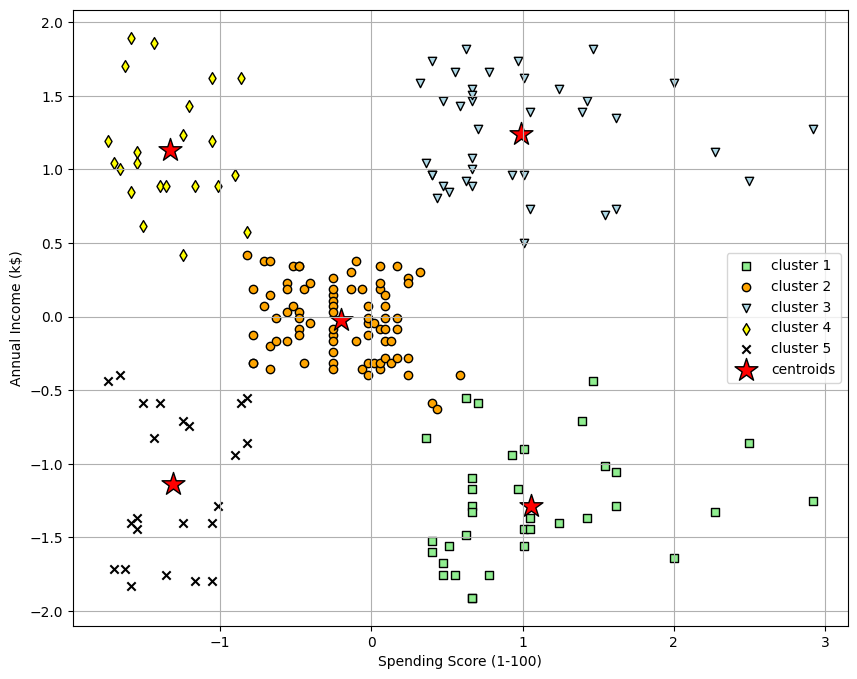
plt.legend(scatterpoints=1) # Chú thích trong biểu đồ

plt.grid() # Bỏ qua khung caro

plt.show()

print("[Trục X      Trục Y]\n",kmean\_5.cluster\_centers\_)

* *Output:*



**Hình 2.16. Biểu diễn dữ liệu đã được phân cụm với k = 5**

* *Tạo ra một tập dữ liệu tương tự:*

df = dataset[['CustomerID', 'Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']]

print(df)

* *Gán các cụm đã được phân cụm vào tập dữ liệu:*

df['Cluster'] = results

data = df.to\_csv('Results\_Clustering\_K-means.csv')

* *Biểu diễn sự phân bố dữ liệu dựa trên dữ liệu bản sao đã tạo:*

plt.figure(figsize=(10,6))

sns.scatterplot(x = 'Annual Income (k$)', y = 'Spending Score (1-100)', hue='Cluster', palette=['green', 'orange','brown','dodgerblue','red'], legend='full', data=df, s=60)

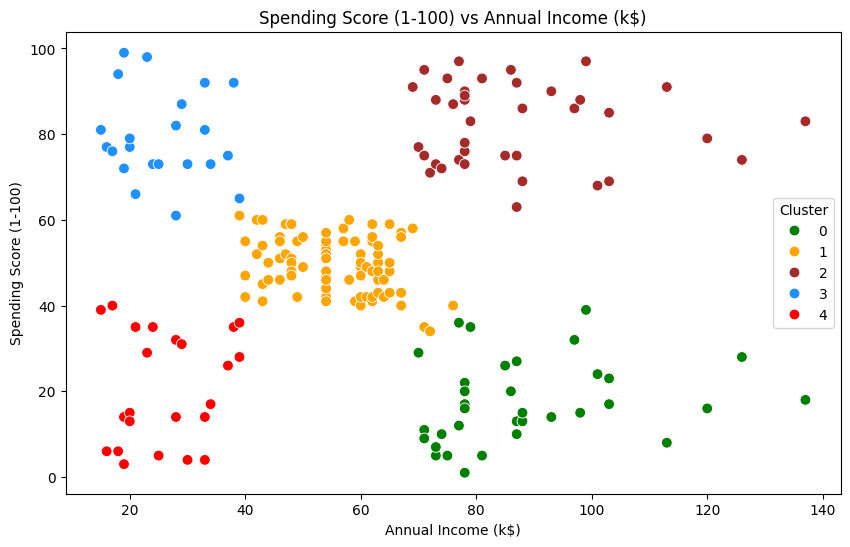
plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Spending Score (1-100) vs Annual Income (k$)')

plt.show()

* *Output:*



**Hình 2.17. Phân bố dữ liệu đã được gom cụm**

* *Tạo danh sách tổng các điểm trong cụm và mã số khách hàng cụ thể trong từng cụm riêng lẻ:*

C1 = df[df['Cluster']==0]

print('Số lượng khách hàng trong cụm 0 là: ', len(C1))

print('Mã số khách hàng lần lượt= ', C1['CustomerID'].values)

print('---------------------------------------------')

C2 = df[df['Cluster']==1]

print('Số lượng khách hàng trong cụm 1 là: ', len(C2))

print('Mã số khách hàng lần lượt= ', C2['CustomerID'].values)

print('---------------------------------------------')

C3 = df[df['Cluster']==2]

print('Số lượng khách hàng trong cụm 2 là: ', len(C3))

print('Mã số khách hàng lần lượt= ', C3['CustomerID'].values)

print('---------------------------------------------')

C4 = df[df['Cluster']==3]

print('Số lượng khách hàng trong cụm 3 là: ', len(C4))

print('Mã số khách hàng lần lượt= ', C4['CustomerID'].values)

print('---------------------------------------------')

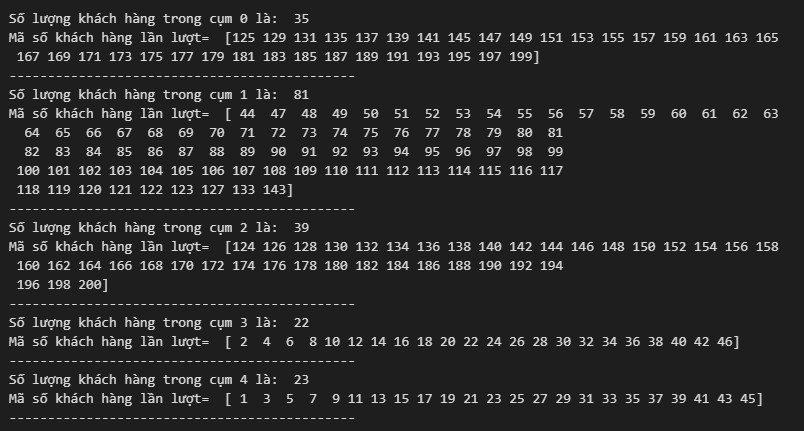
C5 = df[df['Cluster']==4]

print('Số lượng khách hàng trong cụm 4 là: ', len(C5))

print('Mã số khách hàng lần lượt= ', C5['CustomerID'].values)

print('---------------------------------------------')

* *Output:*



**Hình 2.18. Danh sách các điểm dữ liệu mang thuộc tính CustomerID**

* *Để dự đoán kết quả cụm dựa trên một input mới ta sẽ tạo ra một hàm Predict\_clusters:*

def Predict\_clusters(model, thunhap, diemso):

    arr = np.array([[thunhap, diemso]])

    pred = model.predict(arr)

    if (diemso <= 3 and diemso >= -1 & thunhap <= 2 and thunhap >=-2):

        return df[df['Cluster'] == pred[0]].sample(5)

    else:

        return print("Sai số liệu nhập!")

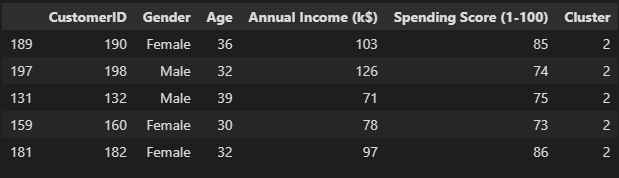
* Vì dữ liệu tích chọn của Annual Income và Spending Score của ta đã được chuẩn hóa dữ liệu nên giá trị khác đi nhiều với Annual Income - thunhap cao nhất là 3 và nhỏ nhất là -1 còn Spending Score - diemso cao nhất là 2 nhỏ nhất là -2. Hàm trả về 5 dòng dữ liệu đầu tương ứng với giá trị Annual Income với Spending Score.

thunhap = 2

diemso = 3

Predict\_clusters(model, thunhap, diemso)

* *Output:*



**Hình 2.19. Dự đoán số cụm thuộc về dựa trên một input mới**

* *Trong đó:*

+ Cụm 0 gồm những khách hàng có mức thu nhập cao và điểm chỉ tiêu thấp

(high - low)

+ Cụm 1 gồm những khách hàng có mức thu nhập trung bình và điểm chỉ tiêu trung bình (normal - normal)

+ Cụm 2 gồm những khách hàng có mức thu nhập cao và điểm chỉ tiêu cao

(high - high)

+ Cụm 3 gồm những khách hàng có mức thu nhập thấp nhưng điểm chỉ tiêu cao (low - high)

+ Cụm 4 gồm những khách hàng có mức thu nhập thấp và điểm chỉ tiêu thấp

(low - low)

* *Thực hiện việc so sánh thuật toán K-MEANS với một thuật toán gom cụm khác dựa trên phân cấp là Agglomerative:*

+ Cài đặt thư viện biểu diễn kết quả tìm số k cụm bằng phương pháp elbow tương tự K-MEANS

+ Vì là sử dụng jupyter notebook trên visual code nên cách thức cài đặt thư viện có chút khác biệt.

!pip install yellowbrick

* Thuật toán phân cụm đa phần đều dựa trên số k cụm để quyết định kết quả gom cụm nên việc tiên quyết trong mỗi bài toán là ta phải xác định một phương pháp bất kì nào hoặc nhiều hơn để tìm số k cụm. Trong bài toán này, nhóm em sử dụng phương pháp (Elbow):

from sklearn.cluster import AgglomerativeClustering

from yellowbrick.cluster import KElbowVisualizer

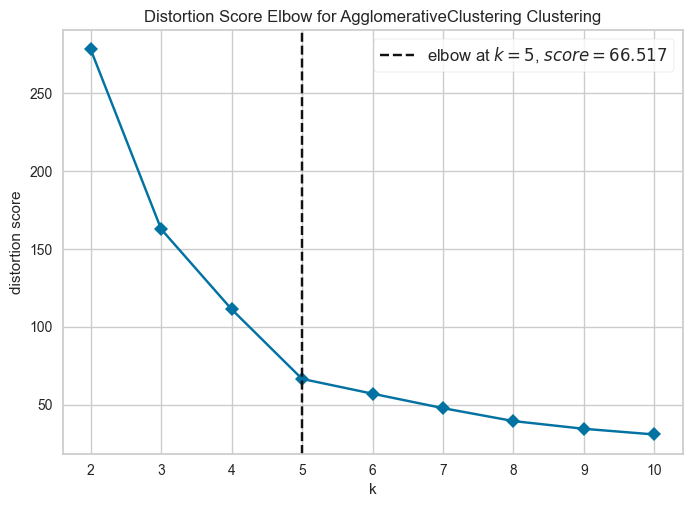
model = AgglomerativeClustering()

visualizer = KElbowVisualizer(model, k=(2,11), timings=False)

visualizer.fit(X\_std)

\_ = visualizer.show()

* *Output:*



**Hình 2.20. Tìm k với thư viện yellowbrick sử dụng phương pháp Elbow**

* Sau khi đã xác định số cụm tối ưu nhất bằng 5 với điểm tin cậy là 67% ta tiến hành ứng dụng giải thuật vào tập dữ liệu đã được trích chọn và chuẩn hóa trước đó:

algorithms = AgglomerativeClustering(n\_clusters=5, affinity='euclidean', linkage='ward')

k\_5 = algorithms.fit(X\_std)

model2 = k\_5.labels\_

* Affinity là hàm tính khoảng cách cụm nó mặc định là euclidean, linkage=’ward’ là hàm tiêu chí liên kết xác định khoảng cách sử dụng giữa các nhóm quan sát. Thuật toán sẽ hợp nhất các cặp cụm tối thiểu hóa tiêu chí này, phương pháp ‘ward’ giảm thiểu phương sai của cụm được hợp nhất. Thuật toán ward linkage cũng chỉ được sử dụng trong điều kiện giả định các quan sát nằm trong không gian euclidean. Biểu diện phân cụm đối với cây phân cấp:

import scipy.cluster.hierarchy as sch

plt.figure(1, figsize = (16 ,8))

plt.grid(b=None)

\_ = dend = sch.dendrogram(sch.linkage(X\_std, method='ward'))

# theroshold

plt.axhline(y=8, linestyle='--', color='orange')

# ploting graphabs

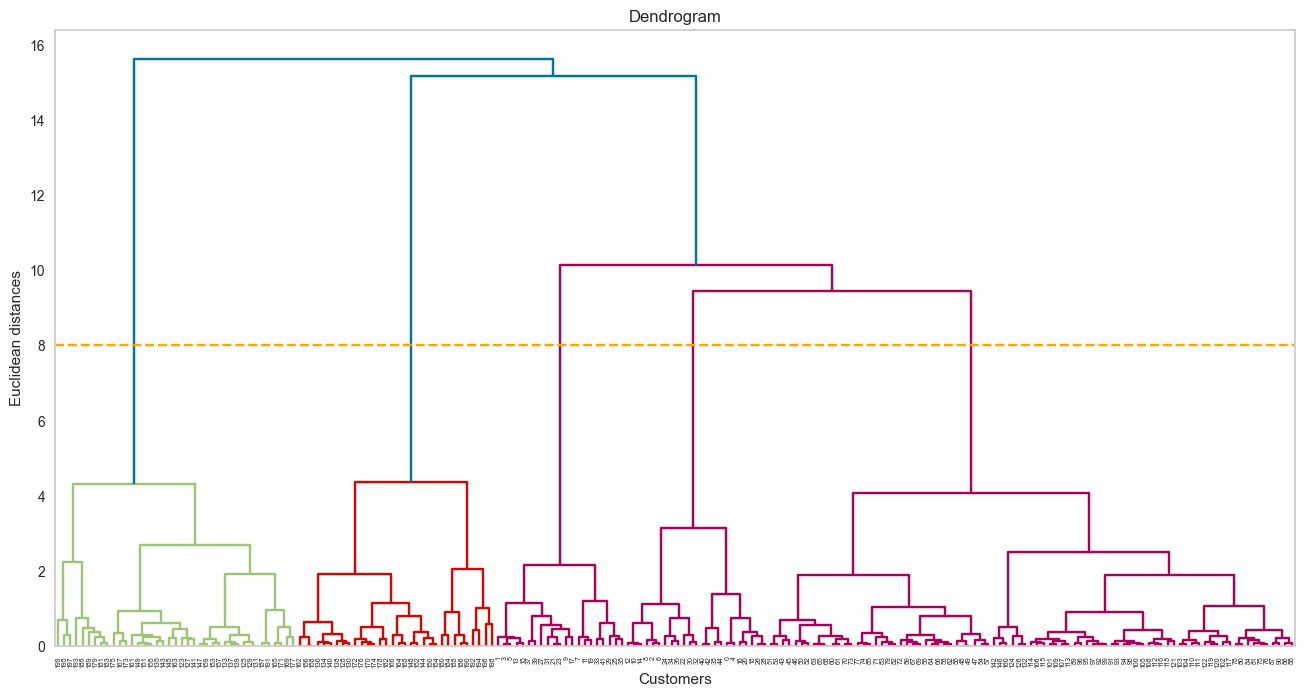
plt.title('Dendrogram')

plt.xlabel('Customers')

plt.ylabel('Euclidean distances')

plt.show()

* *Output:*



**Hình 2.21. Cây phân cấp đối với tập dữ liệu với số cây bằng 5**

* Cây các cụm: Phân cấp cụm thường tạo cây các cụm hay còn được gọi là dendrogram.

+ Các lá của cây biểu diễn các đối tượng riêng lẻ.

+ Các nút trong cây biểu diễn các cụm.

* Lưu dữ liệu lại thành một dataset mới với thuộc tính thêm vào là Clusters đã gom cụm sử dụng thuật toán Agglomerative dựa trên phân cấp:

df2 = dataset[['CustomerID', 'Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']]

df2['Clusters'] = model2

df2.to\_csv('Results\_Clustering\_Aggolomerative.csv')

* Cuối cùng là ta biểu diễn so sánh 2 tập dữ liệu với 2 thuật toán K-MEANS và AGGLOMERATIVE với nhau xem xét sự thay đổi như thế nào:

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15,5))

sns.scatterplot(ax=axes[0], data=df, x = 'Annual Income (k$)', y = 'Spending Score (1-100)', hue='Cluster',

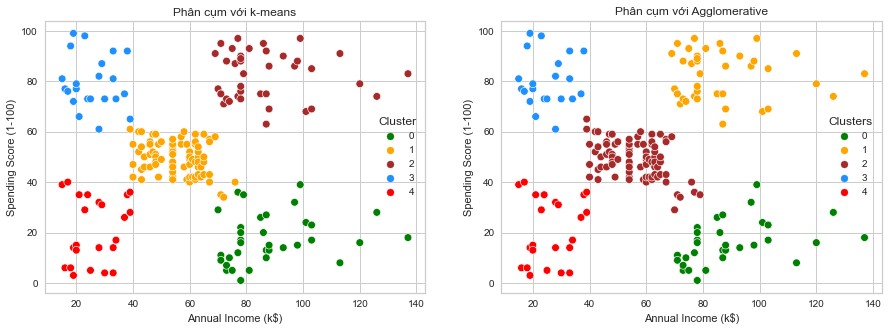
                legend='full', s=60, palette=['green', 'orange','brown','dodgerblue','red']).set\_title('Phân cụm với k-means')

sns.scatterplot(ax=axes[1], data=df2, x = 'Annual Income (k$)', y = 'Spending Score (1-100)', hue='Clusters',

                legend='full', s=60, palette=['green', 'orange','brown','dodgerblue','red']).set\_title('Phân cụm với Agglomerative')

plt.show()

* *Output:*



**Hình 2.22. Mô hình so sánh dữ liệu với 2 thuật toán khác nhau**

* Có thể thấy thuật toán Agglomerative gom cụm dữ liệu đạt chất lượng tốt hơn Kmeans khá nhiều nhưng chưa phải là giải thuật tốt nhất để đánh giá.
* Tập dữ liệu mới được lưu sau khi áp dụng thuật toán Agglomerativ:

**Bảng 2.9. Tập dữ liệu đã được phân cụm**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **CustomerID** | **Gender** | **Age** | **Annual Income**  **(k$)** | **Spending Score**  **(1 – 100)** | **Clusters** |
| 139 | 140 | Female | 35 | 74 | 72 | 1 |
| 140 | 141 | Female | 57 | 75 | 5 | 0 |
| 141 | 142 | Male | 32 | 75 | 93 | 1 |
| 142 | 143 | Female | 28 | 76 | 40 | 2 |
| 143 | 144 | Female | 32 | 76 | 87 | 1 |
| 144 | 145 | Male | 25 | 77 | 12 | 0 |
| 145 | 146 | Male | 28 | 77 | 97 | 1 |
| 146 | 147 | Male | 48 | 77 | 36 | 2 |
| 147 | 148 | Female | 32 | 77 | 74 | 1 |
| 148 | 149 | Female | 34 | 78 | 22 | 0 |

## **2.7 ĐÁNH GIÁ KẾT QUẢ**

### **2.7.1 Tri thức rút ra sau khi phân cụm**

1. Cụm 0: Khách hàng có thu nhập cao, chi tiêu thấp
   * Đặc điểm: Những khách hàng này có thu nhập cao nhưng lại chi tiêu ít tại trung tâm mua sắm
   * Ý nghĩa: Đây là nhóm khách hàng tiềm năng chưa khai thác hết. Cần có chiến lược thu hút nhóm này mua sắm nhiều hơn, ví dụ như cung cấp các sản phẩm cao cấp hoặc chiến lược marketing nhắm đến sở thích của họ
2. Cụm 1: Khách hàng có thu nhập vừa, chi tiêu vừa
   * Đặc điểm: Nhóm này có thu nhập trung bình và có chi tiêu ở mức vừa phải
   * Ý nghĩa: Đây là nhóm khách hàng phổ biến. Việc cung cấp các ưu đãi hoặc các chương trình giảm giá có thể khiến họ chi tiêu nhiều hơn
3. Cụm 2: Khách hàng có thu nhập cao, chi tiêu cao
   * Đặc điểm: Khách hàng có thu nhập cao và thường xuyên chi tiêu lớn tại trung tâm mua sắm
   * Ý nghĩa: Đây là nhóm khách hàng rất quan trọn và có lợi nhuận cao. Do đó, nên tạp trung duy trì sự hài lồng của họ, cung cấp dịch vụ chất lượng, và có chương trình khách hàng thân thiết để giữ chân họ.
4. Cụm 3: Khách hàng thu nhập thấp, chi tiêu cao
   * Đặc điểm: Khách hàng này có xu hướng chi tiêu nhiều, mặc dù có thể thu nhập không quá cao
   * Ý nghĩa: Đây là nhóm khách hàng có tiềm năng cao cho việc tiêu thị các sản phẩm thời trang, công nghệ, và các sản phẩm theo xu hướng. Việc nắm bắt đúng sở thích và xu hướng của nhóm này sẽ rất quan trọng để tăng doanh thu
5. Cụm 4: Khách hàng có thu nhập thấp, chi tiêu thấp
   * Đặc điểm: Khách hàng này có thu nhập thấp, chi tiêu thấp
   * Ý nghĩa: Đây có thể là không phải nhóm khách hàng mục tiêu chính, nhưng vẫn có thể khai thác nếu có các chương trình ưu đãi phù hợp với túi tiền của họ hoặc chiến lược tiếp cận sản phẩm giá rẻ

### **2.7.2 So sánh 2 thuật toán**

**Bảng 2.10. Bảng so sánh 2 thuật toán**

|  |  |  |
| --- | --- | --- |
|  | **K-MEANS** | **AGGLOMERATIVE** |
| **Loại** | Phân hoạch | Phân cấp |
| **Số k cụm** | 5 cụm | 5 cụm |
| **Tính khoảng cách** | Euclidean | Euclidean |
| **Độ phức tạp** | 𝑂((3𝑛𝑘𝑑)𝑡𝑇𝑓𝑙𝑜𝑝) | 𝑂((𝑛3𝑘𝑑)𝑡𝑇𝑓𝑙𝑜𝑝) |
| **Điểm tin cậy** | 55% | 67% |

* Dựa trên dữ liệu phân khúc khách hàng với hai thuộc tính Annual Income (k$) và Spending Score (1-100) đã được chuẩn hóa dữ liệu dựa trên thống kê sử dụng hàm StandardScaler của sklearn, thuật toán phân cụm dựa trên phân cấp Agglomerative cho ra kết quả gom cụm tối ưu hơn nhưng chưa thể khẳng định được đây là thuật toán tối ưu nhất. Việc này đòi hỏi ta phải kiên nhẫn và thời gian để thử nghiệm nhiều phương pháp và các giải thuật phân cụm khác nhau để đưa ra quyết định cuối cùng cho bài toán này.

# **CHƯƠNG 3: KẾT LUẬN**

## **3.1 NHỮNG KIẾN THƯC ĐÃ TIẾP THU TỪ MÔN HỌC**

Trong quá trình học hỏi và nghiên cứu ứng dụng của môn KHAI PHÁ DỮ LIỆU nhóm chúng em đã tiếp thu được phần nào những kiến thức cần tiếu cho tương lai sau này:

* Hiểu biết về quy trình khai phá tri thức cho dữ liệu có giám sát và không có giám sát
* Biết cách thu thập, phân tích và xử lý dữ liệu thô
* Vận dụng toán học để khai phá tri thức tiềm ẩn của dữ liệu

## **3.2 HƯỚNG PHÁT TRIỂN**

* So sánh nhiều thuật toán phân cụm hơn để đưa ra được kết quả gom cụm tốt nhất
* Thử với nhiều phương pháp tính khoảng cách khác ngoài Euclidean và phương pháp đánh giá chất lượng gom cụm khác