TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO KẾT THÚC HỌC PHẦN
KHAI KHOÁNG DỮ LIỆU

Phân cụm khách hàng tiềm năng dựa vào thuật toán gom cụm K-Medoids

Giảng viên hướng dẫn: **ThS. Huỳnh Thành Lộc**

Sinh viên thực hiện:

1. Nguyễn Trần Bảo Long 22DH112007
2. Lương Đức Khoa 22DH111647
3. Nguyễn Hoàng Bảo 22DH110271
4. Nguyễn Hữu Bình 21DH113496

Thành phố Hồ Chí Minh, tháng 7 năm 2025

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO KẾT THÚC HỌC PHẦN

KHAI KHOÁNG DỮ LIỆU

Phân cụm khách hàng tiềm năng dựa vào thuật toán gom cụm K-Medoids

Mã lớp học phần: **241123018401**

Năm học: **2024 – 2025**

Học kỳ: **3**

Sinh viên thực hiện:

1. Nguyễn Trần Bảo Long 22DH112007
2. Lương Đức Khoa 22DH111647
3. Nguyễn Hoàng Bảo 22DH110271
4. Nguyễn Hữu Bình 21DH113496

Thành phố Hồ Chí Minh, tháng 7 năm 2025

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu bài toán

Trong thời đại số hóa và cạnh tranh gay gắt giữa các doanh nghiệp hiện nay, việc thấu hiểu khách hàng trở thành yếu tố then chốt giúp doanh nghiệp phát triển bền vững. Một trong những chiến lược phổ biến trong lĩnh vực marketing là cá nhân hóa chiến dịch tiếp thị dựa trên đặc điểm hành vi và đặc điểm nhân khẩu học của khách hàng. Tuy nhiên, để làm được điều này, doanh nghiệp cần phân loại khách hàng thành các nhóm có đặc điểm tương đồng để đưa ra những chính sách phù hợp với từng nhóm.

Bài toán phân cụm khách hàng (Customer Segmentation) đóng vai trò quan trọng trong phân tích dữ liệu khách hàng. Thay vì nhìn vào từng khách hàng riêng lẻ, phương pháp phân cụm cho phép nhóm các khách hàng có đặc điểm tương đồng về độ tuổi, thu nhập, hành vi chi tiêu,... lại với nhau. Từ đó, doanh nghiệp có thể xây dựng các chiến lược marketing phù hợp như xác định khách hàng tiềm năng, tăng cường giữ chân khách hàng trung thành hoặc tiếp cận hiệu quả nhóm khách hàng mới.

Trong nghiên cứu này, chúng tôi lựa chọn thuật toán K-Medoids để thực hiện bài toán phân cụm khách hàng. Đây là một phương pháp học không giám sát thuộc họ thuật toán phân cụm (clustering), có khả năng phân nhóm dữ liệu dựa trên khoảng cách giữa các điểm dữ liệu và điểm trung tâm cụm (medoid).

Khác với K-Means, K-Medoids lựa chọn các điểm thực sự trong tập dữ liệu làm trung tâm cụm, thay vì sử dụng giá trị trung bình. Điều này giúp K-Medoids kháng nhiễu tốt hơn, đặc biệt trong những trường hợp dữ liệu có chứa outlier (giá trị ngoại lai) hoặc không tuân theo phân phối chuẩn.

Mục tiêu chính của đề tài là ứng dụng thuật toán K-Medoids để phân cụm khách hàng tiềm năng dựa trên các đặc điểm như độ tuổi, thu nhập hằng năm, điểm đánh giá chi tiêu,... trên một tập dữ liệu khách hàng thực tế. Thông qua đó, nhóm nghiên cứu mong muốn đưa ra

những phân tích định tính và định lượng giúp ích cho việc xây dựng chiến lược tiếp thị hiệu quả hơn cho doanh nghiệp.

1.2. Các công trình liên quan

Phân cụm dữ liệu là một trong những kỹ thuật cơ bản và phổ biến nhất trong khai phá dữ liệu và học máy không giám sát. Nhiều thuật toán phân cụm đã được phát triển và ứng dụng vào các bài toán như phân cụm khách hàng, phân tích thị trường, phân tích hình ảnh,... Trong đó, các thuật toán phổ biến bao gồm: K-Means**,** K-Medoids**,** DBSCAN**,** và Hierarchical Clustering**.**

Thuật toán	Mô tả	Ưu điểm	Nhược điểm
K-Means	Phân cụm dựa trên trung bình các điểm dữ liệu trong cụm	Đơn giản, hiệu quả với dữ liệu có phân phối đều và không có outlier	Nhạy cảm với outlier và khởi tạo centroid ban đầu
K-Medoids	Chọn medoid là điểm đại diện (trong dữ liệu) có khoảng cách nhỏ nhất	Ổn định hơn K-Means, chống chịu outlier tốt, kết quả dễ hiểu hơn	Tính toán chậm hơn, không hiệu quả với dữ liệu rất lớn
DBSCAN	Phân cụm dựa trên mật độ điểm lân cận	Không cần xác định số cụm trước, nhận diện tốt outlier	Không hoạt động tốt với dữ liệu có mật độ cụm khác nhau
Hierarchical	Xây dựng cây p hân cấp cụm (dendrogram)	Không cần chọn K, trực quan hóa được cấu trúc dữ liệu	Tính toán phức tạp, không phù hợp với dữ liệu lớn

Bảng 1: Các công trình liên quan.

Trong các thuật toán trên, K-Medoids nổi bật bởi một số ưu điểm quan trọng khiến nó phù hợp với bài toán phân cụm khách hàng:

- Chống chịu tốt với nhiễu và giá trị ngoại lai: Vì K-Medoids sử dụng medoid (là điểm thực tế trong cụm) thay vì trung bình như K-Means, nó ít bị ảnh hưởng bởi các giá trị ngoại lệ – điều rất thường gặp trong dữ liệu thực tế.

- Dễ hiểu và dễ diễn giải: Các medoid được chọn là điểm có ý nghĩa thực tế (thuộc tập dữ liệu gốc), giúp việc phân tích và giải thích cụm trực quan và dễ tiếp cận hơn với người dùng không chuyên.
- Linh hoạt hơn trong việc áp dụng các hàm khoảng cách phi Euclid: Có thể sử dụng Manhattan distance hoặc các hàm khác tùy yêu cầu bài toán.

Nhờ vào những lợi thế này, K-Medoids trở thành lựa chọn phù hợp cho bài toán phân cụm khách hàng, giúp doanh nghiệp đưa ra quyết định chính xác hơn trong việc xác định nhóm khách hàng tiềm năng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về K-Medoids

K-Medoids là một thuật toán phân cụm dựa trên phân vùng (partitioning clustering algorithm), tương tự như K-Means, nhưng có khả năng mạnh mẽ hơn trong việc xử lý các giá trị ngoại lai và dữ liệu có nhiễu. Điểm khác biệt cốt lõi là K-Medoids sử dụng các điểm dữ liệu thực tế trong tập dữ liệu làm tâm cụm (gọi là "medoid"), thay vì sử dụng trung bình cộng của các điểm trong cụm (centroid) như K-Means. Medoid được định nghĩa là điểm dữ liệu trong cụm có tổng khoảng cách nhỏ nhất đến tất cả các điểm khác trong cùng cụm.

2.2. Nguyên lý hoạt động của K-Medoids

Thuật toán K-Medoids (còn được gọi là PAM - Partitioning Around Medoids) hoạt động theo các bước lặp như sau:

1. **Khởi tạo:**
    - Chọn trước số lượng cụm k mong muốn.
    - Ngẫu nhiên chọn k điểm dữ liệu từ tập dữ liệu làm các medoid ban đầu.
2. **Gán cụm:**
    - Với mỗi điểm dữ liệu còn lại, gán nó vào cụm có medoid gần nhất (dựa trên một thang đo khoảng cách đã chọn).
3. **Cập nhật Medoid:**
    - Với mỗi cụm, tìm một điểm dữ liệu mới trong cụm đó mà tổng khoảng cách từ nó đến tất cả các điểm khác trong cùng cụm là nhỏ nhất. Điểm này sẽ trở thành medoid mới cho cụm đó.
    - Quá trình này được thực hiện bằng cách thử hoán đổi medoid hiện tại với một điểm không phải medoid trong cụm và tính toán chi phí (tổng khoảng cách) của cấu hình mới. Nếu chi phí giảm, việc hoán đổi được chấp nhận.
4. **Lặp lại:**
    - Lặp lại các bước 2 và 3 cho đến khi không có medoid nào thay đổi hoặc một tiêu chí hội tụ (ví dụ: số lần lặp tối đa, sự thay đổi chi phí không đáng kể) được thỏa mãn.

Mục tiêu của K-Medoids là tối thiểu hóa tổng khoảng cách (hoặc tổng độ không tương đồng) giữa các điểm dữ liệu và medoid của cụm mà chúng thuộc về.

Lý do lựa chọn thuật toán K-Medoids:

Trong bài toán phân cụm khách hàng, nhóm đã quyết định lựa chọn K-Medoids thay vì các thuật toán phân cụm phổ biến khá, dựa trên các lý do sau:

1. **Khả năng chống chịu với Outlier:** Mặc dù nhóm đã thực hiện xử lý outlier bằng phương pháp IQR Capping, dữ liệu vẫn có thể chứa các giá trị bán ngoại lai (semi-outliers) hoặc các điểm dữ liệu ở xa trung tâm phân phối, đặc biệt là với các đặc trưng như `Monetary` và `TotalQuantity` (nơi các giá trị cao được capped nhưng vẫn đại diện cho điểm cực trị). K-Medoids, với việc sử dụng các medoid là các điểm dữ liệu thực tế, ít nhạy cảm với các điểm cực đoan hơn so với K-Means (sử dụng trung bình cộng - centroid), vốn dễ bị kéo lệch bởi các outlier. Điều này giúp các tâm cụm đại diện chính xác hơn cho các nhóm khách hàng.
2. **Tính diễn giải (Interpretability):** Medoids là các điểm dữ liệu thực tế trong tập dữ liệu. Điều này có nghĩa là mỗi tâm cụm là một khách hàng "có thật", giúp dễ dàng hơn trong việc diễn giải đặc điểm của từng cụm khách hàng. Chúng ta có thể phân tích thông tin chi tiết của medoid đó để hiểu rõ hơn về hành vi tiêu biểu của cả phân khúc.
3. **Phù hợp với mục tiêu kinh doanh:** Trong phân tích khách hàng, việc xác định được những "khách hàng đại diện" (medoids) cho mỗi phân khúc có giá trị cao trong việc xây dựng chiến lược marketing và chăm sóc khách hàng mục tiêu. Ví dụ, medoid của cụm khách hàng "VIP" là một khách hàng thực sự có thể nghiên cứu để hiểu rõ hơn.

2.3. Các thang đo khoảng cách (Distance Metrics)

Việc lựa chọn thang đo khoảng cách là một yếu tố quan trọng ảnh hưởng trực tiếp đến kết quả phân cụm của K-Medoids. Thang đo xác định "sự gần nhau" hoặc "sự khác biệt" giữa hai điểm dữ liệu. Trong bài toán này, nhóm đã sử dụng các đặc trưng định lượng

đã được chuẩn hóa do đó các thang đo khoảng cách phổ biến trong không gian Euclide là phù hợp.

Khoảng cách Euclidean là thang đo khoảng cách phổ biến nhất và trực quan nhất, được sử dụng rộng rãi trong các thuật toán phân cụm dựa trên khoảng cách. Nó biểu diễn khoảng cách "đường chim bay" giữa hai điểm trong không gian N chiều.

Công thức tính khoảng cách Euclidean giữa hai điểm $p= (p_1, p_2, ..., p_n)$ và $q= (q_1, q_2, ..., q_n)$ là:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Trong đó:

- n là số lượng đặc trưng (chiều của không gian).
- p_i và q_i là giá trị của đặc trưng thứ i cho điểm p và q tương ứng.

Lý do lựa chọn khoảng cách Euclidean trong bài toán:

- Trực quan và dễ hiểu: Khoảng cách Euclidean phản ánh một cách trực tiếp sự khác biệt về "vị trí" của khách hàng trong không gian đặc trưng.
- Phù hợp với dữ liệu đã chuẩn hóa: Sau khi dữ liệu được chuẩn hóa bằng Z-score, tất cả các đặc trưng đều có cùng thang đo (trung bình 0, độ lệch chuẩn 1), giúp khoảng cách Euclidean hoạt động hiệu quả và đảm bảo mỗi đặc trưng đóng góp công bằng vào việc tính toán khoảng cách. Do đó, việc sử dụng metric='euclidean' trong triển khai K-Medoids là hoàn toàn phù hợp.

Các thang đo khác (được cân nhắc nhưng không được chọn):

Mặc dù khoảng cách Euclidean là lựa chọn chính, nhóm cũng đã cân nhắc các thang đo khác:

- Khoảng cách Manhattan (City Block Distance): Tính tổng giá trị tuyệt đối của sự khác biệt giữa các tọa độ. Khoảng cách Manhattan ít nhạy cảm hơn với các outlier

so với Euclidean, vì nó không bình phương các sai khác. Tuy nhiên, với việc dữ liệu đã được xử lý outlier bằng capping và chuẩn hóa, Euclidean vẫn là lựa chọn trực quan và đủ mạnh mẽ mà không làm mất đi tính nhạy cảm cần thiết đối với sự khác biệt tổng thể.

- Khoảng cách Cosine Similarity: Đo lường góc giữa hai vector, thường được dùng cho dữ liệu thưa hoặc khi hướng của vector quan trọng hơn độ lớn. Đối với dữ liệu hành vi khách hàng dạng số đã chuẩn hóa của chúng ta, nơi độ lớn và vị trí trong không gian đặc trưng là quan trọng, Cosine Similarity không phải là lựa chọn tối ưu.

2.4. Xác định số lượng cụm tối ưu (K)

Phương pháp Elbow

Phương pháp Elbow tìm kiếm điểm "khuỷu tay" trên biểu đồ thể hiện mối quan hệ giữa số lượng cụm (k) và một tiêu chí đánh giá nội tại của cụm (chẳng hạn như tổng bình phương khoảng cách từ mỗi điểm đến tâm cụm gần nhất - distortion). Khi tăng số lượng cụm, distortion sẽ giảm. Điểm "khuỷu tay" là nơi mà sự giảm sút của distortion bắt đầu chậm lại đáng kể, cho thấy việc tăng thêm cụm không mang lại nhiều lợi ích cải thiện đáng kể chất lượng phân cụm.

Khái niệm Distortion (Tổng khoảng cách đến medoid): Đối với thuật toán K-Medoids, Distortion là tổng khoảng cách của tất cả các điểm dữ liệu đến medoid của cụm mà chúng thuộc về. Mục tiêu của thuật toán là tối thiểu hóa giá trị này. Giả sử có k cụm C_1, C_2, ..., C_k với các medoid tương ứng là m_1, m_2, ..., m_k. Distortion được tính bằng công thức:

$$Distortion = \sum_{j=1}^{k} \sum_{x \in C_j} d(x, m_j)$$

Trong đó:

- $d(x, m_j)$ là khoảng cách (ví dụ: Euclidean) giữa điểm dữ liệu x và medoid m_j của cụm C_j mà x thuộc về.

Nguyên lý hoạt động của phương pháp Elbow: Khi tăng số lượng cụm (k), Distortion sẽ có xu hướng giảm. Tuy nhiên, tốc độ giảm sẽ thay đổi. Ban đầu, khi k còn nhỏ, việc thêm

một cụm mới thường giúp giảm Distortion đáng kể bằng cách tạo ra các cụm chặt chẽ hơn. Đến một điểm nào đó, việc tăng k sẽ chỉ mang lại sự giảm Distortion rất nhỏ, cho thấy các cụm mới được tạo ra không thực sự cải thiện nhiều cấu trúc tổng thể của dữ liệu mà chỉ đơn thuần chia nhỏ các cụm hiện có. Điểm này, nơi đường cong giảm Distortion bắt đầu "uốn cong" và trở nên gần như phẳng, được gọi là "điểm khuỷu tay". Đây chính là giá trị k được cho là tối ưu, thể hiện sự cân bằng tốt giữa việc giảm Distortion và duy trì số lượng cụm hợp lý.

2.5. Đánh giá chất lượng phân cụm

Sau khi xác định số lượng cụm tối ưu và huấn luyện mô hình K-Medoids, việc đánh giá chất lượng của các cụm là cần thiết để xác định mức độ hiệu quả của thuật toán trong việc phân tách dữ liệu. Nhóm đã sử dụng các chỉ số đánh giá nội tại (intrinsic evaluation metrics) là Silhouette Score và Davies-Bouldin Score, vốn không yêu cầu nhãn thực (ground truth) của dữ liệu, mà chỉ dựa vào cấu trúc của các cụm được tạo ra.

Silhouette Score

Khái niệm: Silhouette Score (Hệ số Silhouette) là một chỉ số đo lường mức độ tương đồng của một điểm dữ liệu với cụm của chính nó (độ kết dính - cohesion) so với các cụm lân cận (độ tách biệt - separation). Chỉ số này giúp đánh giá cả độ chặt chẽ của các cụm và khoảng cách giữa chúng.

Công thức: Đối với một điểm dữ liệu i:

1. Tính a(i): Khoảng cách trung bình giữa điểm i và tất cả các điểm khác trong cùng cụm mà i thuộc về. Giá trị a(i) càng nhỏ thì điểm i càng phù hợp với cụm của nó.
2. Tính b(i): Khoảng cách trung bình giữa điểm i và tất cả các điểm trong cụm lân cận gần nhất (cụm mà i không thuộc về nhưng gần i nhất).

Hệ số Silhouette s(i) cho điểm i được tính bằng:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Silhouette Score tổng thể của một cấu hình phân cụm là giá trị trung bình của s(i) cho tất cả các điểm dữ liệu.

Ý nghĩa giá trị:

- Giá trị nằm trong khoảng [−1,1].
- **s(i) gần +1:** Điểm i nằm rất gần và phù hợp với cụm của nó, đồng thời rất xa các cụm khác. Điều này chỉ ra cụm được phân tách rõ ràng và tốt.
- **s(i) gần 0:** Điểm i nằm gần ranh giới giữa hai cụm, có thể được gán vào cụm không rõ ràng.
- **s(i) âm:** Điểm i có thể đã được gán sai cụm, vì nó gần một cụm khác hơn là cụm của chính nó.

Mục tiêu là tối đa hóa Silhouette Score.

Davies-Bouldin Score

Khái niệm: Davies-Bouldin Score (Chỉ số Davies-Bouldin) là một chỉ số đo lường tỷ lệ giữa độ phân tán trung bình trong cụm và khoảng cách giữa các tâm cụm. Chỉ số này định nghĩa độ tương tự giữa hai cụm i và j là R_{ij}.

Công thức:

1. Tính độ phân tán của cụm: s_i là khoảng cách trung bình của tất cả các điểm trong cụm i đến tâm cụm của nó.
2. Tính khoảng cách giữa hai tâm cụm: d_{ij} là khoảng cách giữa tâm cụm i và tâm cụm j.
3. Tính độ tương tự giữa cụm i và j:

$$R_{ij} = \frac{s_i - s_j}{d_{ij}}$$

Giá trị R_{ij} càng nhỏ càng tốt, vì nó chỉ ra rằng các cụm i và j có độ phân tán thấp (chặt chẽ) và/hoặc khoảng cách giữa chúng lớn (phân tách tốt).

Davies-Bouldin Score tổng thể được tính bằng giá trị trung bình của độ tương tự tối đa mà mỗi cụm có với cụm lân cận gần nhất của nó:

$$DBS = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} R_{ij}$$

Ý nghĩa giá trị:

- Giá trị lớn hơn hoặc bằng 0.
- Giá trị càng thấp cho thấy các cụm được phân tách tốt hơn và các cụm bên trong chặt chẽ hơn.
- Mục tiêu là tối thiểu hóa Davies-Bouldin Score.

2.6. Ví dụ minh họa

Xét tập dữ liệu gồm 6 điểm hai chiều như sau:

Điểm	Tọa độ (x, y)
A	(1, 1)
B	(2, 2)
C	(6, 5)
D	(7, 6)

Ta muốn phân cụm thành k = 2

Bước 1: Khởi tạo

Chọn ngẫu nhiên 2 medoid ban đầu, giả sử chọn B và D.

Bước 2: Gán cụm

Tính khoảng cách Euclidean từ mỗi điểm đến 2 medoid:

Distance (x, y) = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Gán điểm vào cụm gần nhất (sử dụng khoảng cách Euclidean)

- **Điểm A (1, 1):**
  - Khoảng cách đến B (2,2): $\sqrt{(1-2)^2 + (1-2)^2} \approx 1.414$
  - Khoảng cách đến D(7,6): $\sqrt{(1-7)^2 + (1-6)^2} \approx 7.810$

- **A thuộc Cụm 1** (vì 1.414<7.810)

- **Điểm B (2, 2):** (Là medoid B)
  - Khoảng cách đến B(2,2): 0
  - Khoảng cách đến D(7,6): $\sqrt{(2-7)^2+(2-6)^2} \approx 6.403$
  - **B thuộc Cụm 1**
- **Điểm C (6, 5):**
  - Khoảng cách đến B (2,2): $\sqrt{(6-2)^2+(5-2)^2}$ =5
  - Khoảng cách đến D (7,6): $\sqrt{(6-7)^2+(5-6)^2} \approx 1.414$
  - **C thuộc Cụm 2** (vì 1.414<5)
- **Điểm D (7, 6):** (Là medoid D)
  - Khoảng cách đến B(2,2): $\sqrt{(7-2)^2+(6-2)^2} \approx 6.403$
  - Khoảng cách đến D(7,6): 0
  - **D thuộc Cụm 2**

Kết quả cụm ban đầu:

- **Cụm 1:** {A(1,1), B(2,2)} - Medoid hiện tại: B(2,2)
- **Cụm 2:** {C(6,5), D(7,6)} - Medoid hiện tại: D(7,6)

Tổng chi phí (Sum of Dissimilarities): Tính tổng khoảng cách từ mỗi điểm đến medoid của cụm nó.

Chi phí= $\sqrt{2}+0+\sqrt{2}+0 \approx 2.828$

Bước 3: Cập nhật Medoids

Cụm 1: {A(1,1), B(2,2)}

Medoid hiện tại là B(2,2). Chi phí cụm là d(A,B) + d(B,B) = $\sqrt{(1-2)^2+(1-2)^2}+0$ ≈1.414.

- **Thử A(1, 1) làm medoid mới:**
  - Chi phí cụm mới: d(A,A) + d(B,A) = $\sqrt{(1-2)^2+(1-2)^2}+0$ ≈1.414

- Tổng chi phí mới cho toàn bộ dữ liệu: (chi phí Cụm 1 mới) + (chi phí Cụm 2 cũ) = $\sqrt{2} + \sqrt{2} \approx 2.828$.
- Chi phí không giảm ($2\sqrt{2} = 2\sqrt{2}$), nên không thay đổi medoid cho Cụm 1. Medoid của Cụm 1 vẫn là B(2,2).

Cụm 2: {C(6,5), D(7,6)}

Medoid hiện tại là D(7,6). Chi phí cụm là d(C,D) + d(D,D) = $\sqrt{(6-7)^2 + (5-6)^2 + 0}$ ≈1.414.

- **Thử C(6, 5) làm medoid mới:**
  - Chi phí cụm mới: d(C,C) + d(D,C) = $\sqrt{(7-6)^2 + (6-5)^2} \approx 1.414$.
  - Tổng chi phí mới cho toàn bộ dữ liệu: (chi phí Cụm 1 cũ) + (chi phí Cụm 2 mới) = $\sqrt{2} + \sqrt{2} \approx 2.828$.
  - Chi phí không giảm ($2\sqrt{2} = 2\sqrt{2}$), nên không thay đổi medoid cho Cụm 2. Medoid của Cụm 2 vẫn là D(7,6).

Kết quả cuối cùng (sử dụng khoảng cách Euclidean):

- **Cụm 1:** {A(1,1), B(2,2)} với medoid là B(2,2)
- **Cụm 2:** {C(6,5), D(7,6)} với medoid là D(7,6)

CHƯƠNG 3. XÂY DỰNG MÔ HÌNH

3.1. Giới thiệu tập dữ liệu

Tập dữ liệu được sử dụng là một bộ dữ liệu giao dịch thương mại điện tử thực tế từ một nhà bán lẻ trực tuyến, bao gồm khoảng 541,909 dòng tương ứng với các hóa đơn bán hàng từ tháng 12/2010 đến tháng 12/2011.

Nguồn dữ liệu được lấy từ nền tảng Kaggle, và được sử dụng rộng rãi trong các nghiên cứu liên quan đến phân tích hành vi khách hàng, phân cụm, RFM segmentation, ...

Tên cột	Mô tả
InvoiceNo	Mã hóa đơn bán hàng (không trùng lặp)
StockCode	Mã sản phẩm được mua
Description	Tên mô tả của sản phẩm
Quantity	Số lượng mặt hàng đã mua
InvoiceDate	Thời gian giao dịch diễn ra
UnitPrice	Giá mỗi mặt hàng tại thời điểm mua
CustomerID	Mã số khách hàng (nhiều dòng có thể trùng ID nếu có nhiều đơn hàng)
Country	Quốc gia của khách hàng

Bảng 2: Bảng mô tả tập dữ liệu.

3.2. Tiền xử lý dữ liệu

3.2.1. Xử lý dữ liệu thiếu

Tập dữ liệu ban đầu chứa hơn 541.000 dòng, tuy nhiên có tới 135.080 dòng bị thiếu thông tin mã khách hàng (CustomerID) – đây là một trường quan trọng dùng để phân nhóm khách hàng.

```
missing_customer = df[df['CustomerID'].isnull()]
missing_customer.info()
```

```
Index: 135080 entries, 622 to 541540
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   InvoiceNo    135080 non-null  object
```

```
 1   StockCode    135080 non-null  object
 2   Description  133626 non-null  object
 3   Quantity     135080 non-null  int64
 4   InvoiceDate  135080 non-null  datetime64[ns]
 5   UnitPrice    135080 non-null  float64
 6   CustomerID   0 non-null       object
 7   Country      135080 non-null  object
```

Vì mục tiêu bài toán là phân cụm theo hành vi mua sắm của từng khách hàng, những dòng không xác định được danh tính sẽ không thể gom cụm đúng cách. Do đó, nhóm quyết định loại bỏ toàn bộ các dòng thiếu CustomerID

```
df = df.dropna(subset=['CustomerID'])
df.info()
```

```
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    406829 non-null  object
 1   StockCode    406829 non-null  object
 2   Description  406829 non-null  object
 3   Quantity     406829 non-null  int64
 4   InvoiceDate  406829 non-null  datetime64[ns]
 5   UnitPrice    406829 non-null  float64
 6   CustomerID   406829 non-null  object
 7   Country      406829 non-null  object
```

3.2.2. Xử lý giá trị bất thường

1. **Trong quá trình phân tích và làm sạch d ữ liệu, nhóm phát hiện hai vấn đề đáng chú ý liên quan đến tính hợp lệ của các trường số liệu quan trọng: Quantity và UnitPrice.**
   - Số lượng âm (Quantity < 0): Có tổng cộng 8.905 dòng dữ liệu có số lượng sản phẩm âm, phản ánh các trường hợp hoàn trả hoặc hủy đơn hàng. Do bài toán hướng đến phân cụm khách hàng dựa trên hành vi mua sắm thực tế, các giao dịch mang tính

chất hoàn trả này sẽ gây nhiễu và làm sai lệch đặc trưng tiêu dùng. Do đó, nhóm quyết định loại bỏ hoàn toàn các dòng có Quantity âm:

```
df = df[(df['Quantity'] > 0)]
```

- Giá sản phẩm bằng 0 (UnitPrice = 0**):** Một số sản phẩm trong tập dữ liệu được ghi nhận có đơn giá bằng 0, với tổng cộng 40 dòng bị ảnh hưởng. Đây có thể là do lỗi nhập liệu, chương trình khuyến mãi miễn phí hoặc các giao dịch nội bộ. Tuy nhiên, việc loại bỏ hoàn toàn các dòng này có thể dẫn đến mất mát dữ liệu không cần thiết.

  Để xử lý hiệu quả, nhóm sử dụng phương pháp thay thế các giá trị UnitPrice = 0 bằng giá trung bình của sản phẩm tương ứng**,** dựa trên mã sản phẩm (StockCode). Cụ thể:

```
mean_price_per_product = df[df['UnitPrice'] >
0].groupby('StockCode')['UnitPrice'].mean()

# Tạo mask các dòng có giá = 0
mask_zero_price = df['UnitPrice'] == 0

# Với mỗi dòng có giá = 0, thay bằng giá trung bình tương ứng theo
StockCode
df.loc[mask_zero_price, 'UnitPrice'] = df.loc[mask_zero_price,
'StockCode'].map(mean_price_per_product)
```

Hạng mục kiểm tra	Số dòng vi phạm	Ý nghĩa
Quantity < 0	8.905	Các dòng này là hàng bị trả lại (hoàn hàng hoặc lỗi)
UnitPrice <= 0	40	Các dòng này có giá sản phẩm không hợp lệ (lỗi nhập liệu hoặc khuyến mãi không rõ ràng)

Bảng 3: Kết quả kiểm tra dữ liệu.

Phương pháp này đảm bảo rằng giá trị bị thiếu được thay thế hợp lý, giữ nguyên được các thông tin giao dịch quan trọng và giúp tăng độ tin cậy của dữ liệu đầu vào cho mô hình phân cụm.

2. Xử lý mã sản phẩm không hợp lệ

Trong tập dữ liệu, bên cạnh các mã sản phẩm hợp lệ, nhóm phát hiện tồn tại nhiều `StockCode` không đại diện cho hàng hóa thực tế. Các mã như `POST`, `C2`, `M`,... thường biểu thị các khoản phụ phí vận chuyển, ghi chú nội bộ, hoặc các hạng mục không có giá trị tiêu dùng cụ thể. Việc giữ lại những mã này có thể gây sai lệch trong quá trình phân tích hành vi mua sắm và ảnh hưởng đến độ chính xác của mô hình phân cụm.

Để loại bỏ nhiễu và đảm bảo chất lượng dữ liệu, nhóm tiến hành **lọc và chỉ giữ lại các dòng có mã sản phẩm là số hoặc bắt đầu bằng số**, vốn là đặc điểm nhận dạng của các sản phẩm thật trong hệ thống:

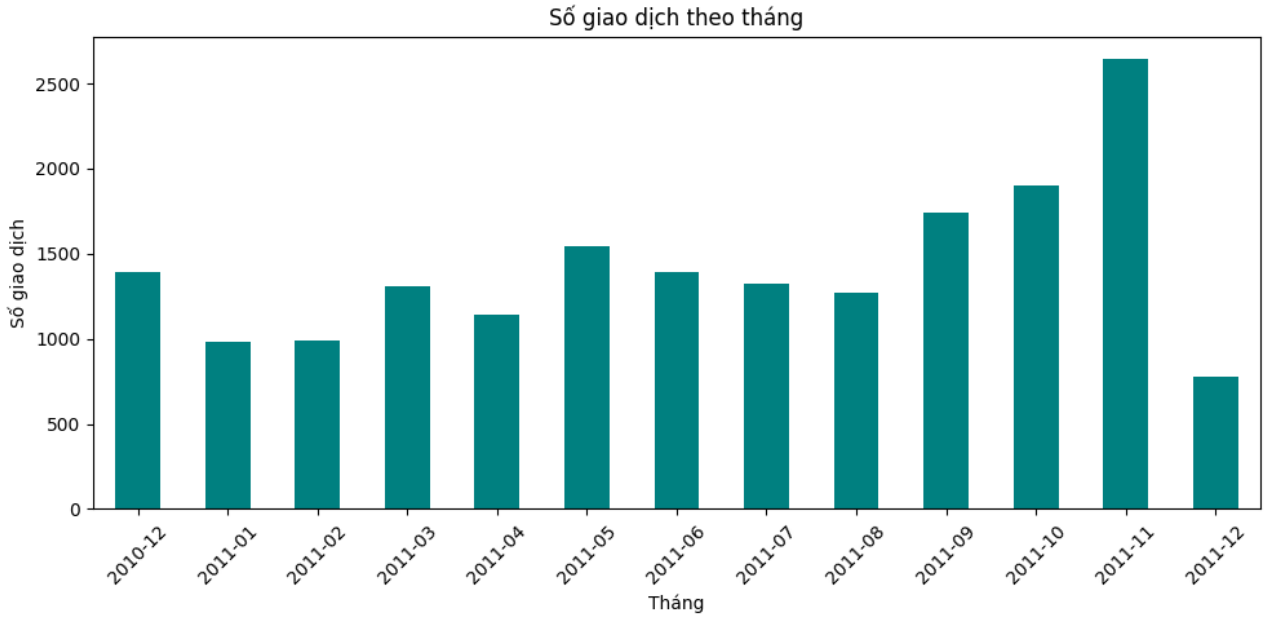
```
# Giữ lại chỉ các dòng có StockCode là số hoặc bắt đầu bằng số (sản phẩm thật)
df = df[df['StockCode'].str.match('^[0-9]+', na=False)]
```

Việc lọc này giúp loại bỏ hoàn toàn các giao dịch phụ trợ, đảm bảo rằng chỉ những sản phẩm có thật và mang tính tiêu dùng mới được đưa vào phân tích.

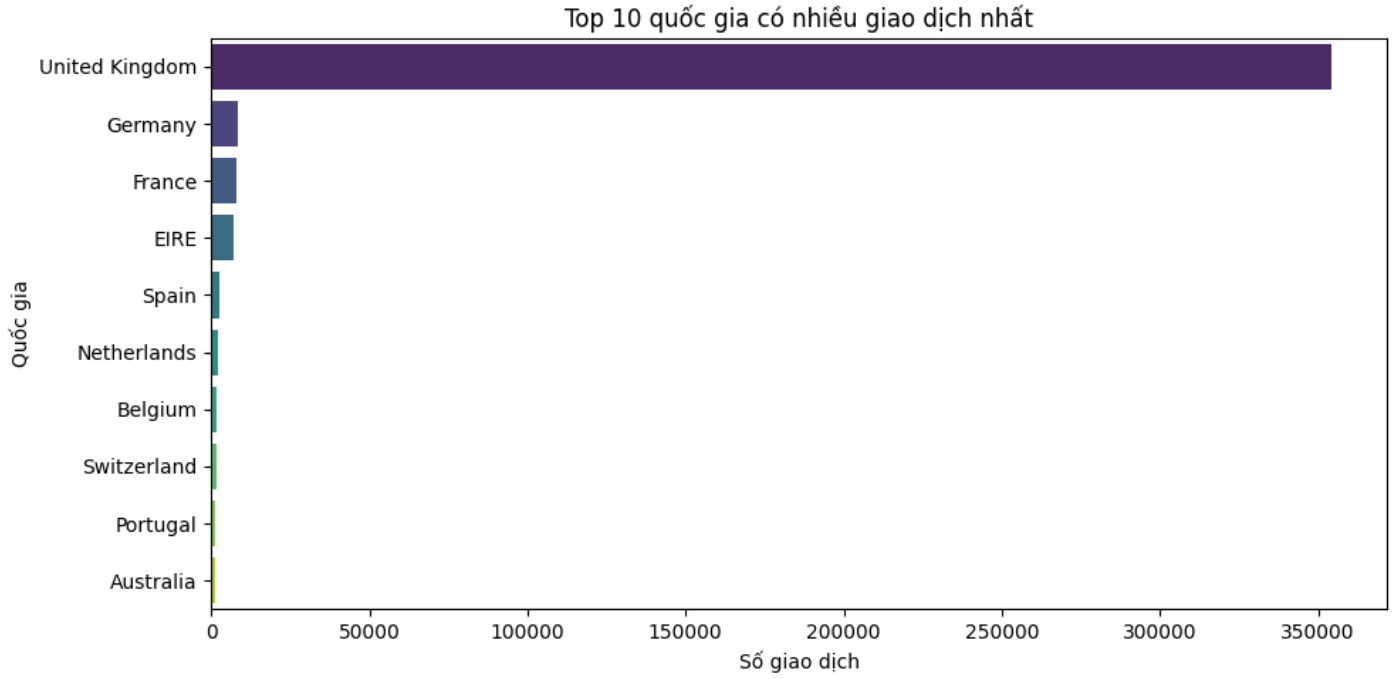
Kết quả sau xử lý:

- Tổng số dòng còn lại**:** 396.370
- Chất lượng dữ liệu đảm bảo: Tập dữ liệu hiện tại chỉ còn các giao dịch hợp lệ, đầy đủ thông tin khách hàng, và mã sản phẩm rõ ràng, hoàn toàn sẵn sàng cho các bước trích đặc trưng và phân cụm khách hàng ở các bước tiếp theo.

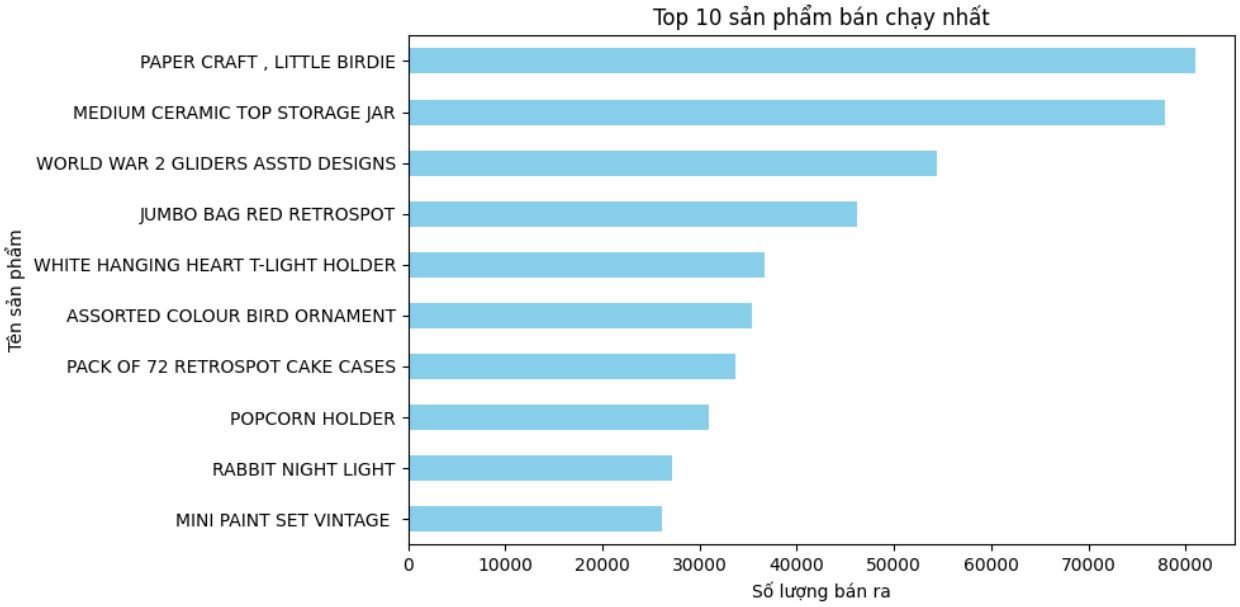
3.3. Trực quan hóa dữ liệu



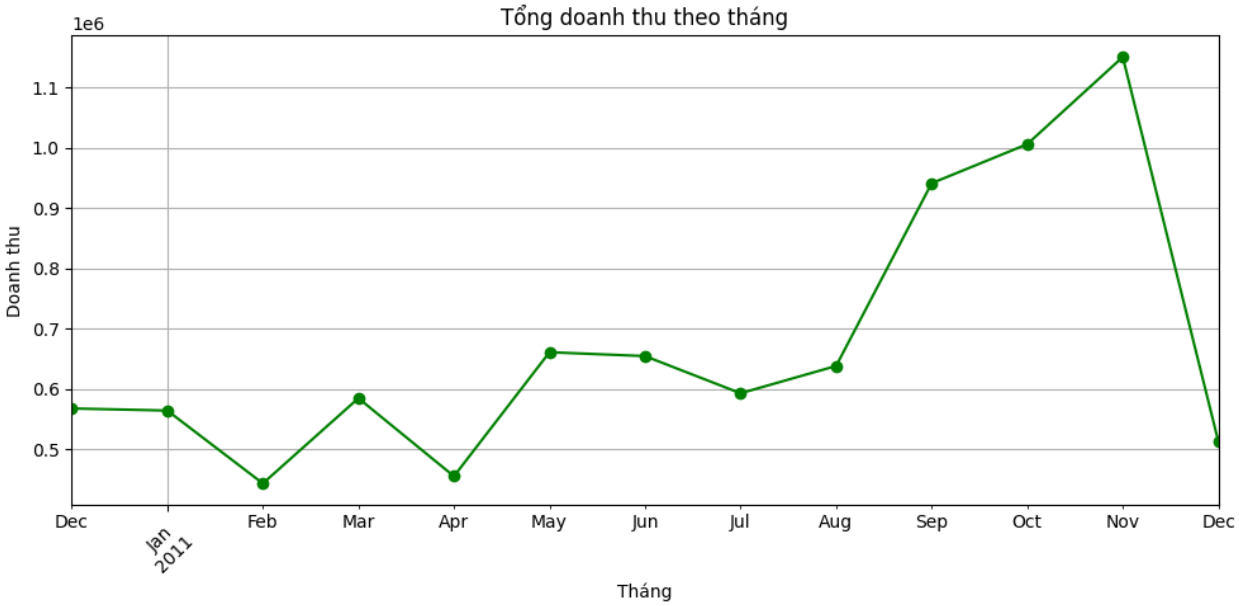
Hình 1: Số giao dịch theo tháng.



Hình 2: Số quốc gia có nhiều giao dịch



Hình 3: Top sản phẩm bán chạy



Hình 4: Tổng doanh thu theo tháng

3.4. Trích chọn đặc trưng

3.4.1. Chuẩn hóa về bảng khách hàng

Sau khi hoàn tất quá trình tiền xử lý, nhóm tiến hành trích xuất các đặc trưng đầu vào cho bài toán phân cụm. Việc lựa chọn đặc trưng không chỉ dựa vào phân tích thống kê thuần túy, mà còn được dẫn dắt bởi kiến thức chuyên môn trong lĩnh vực phân tích khách hàng.

Để xây dựng bảng đặc trưng khách hàng, nhóm đã thực hiện quy trình sau: đầu tiên, chuyển đổi dữ liệu từ bảng hóa đơn sang bảng khách hàng bằng cách thực hiện thao tác group by theo CustomerID. Sau đó, từ bảng khách hàng này, nhóm tiến hành trích chọn các đặc trưng mới dựa trên ma trận tương quan để đảm bảo các đặc trưng được lựa chọn có ý nghĩa và ít trùng lặp thông tin.

Các đặc trưng được sử dụng bao gồm:

- **Recency (R)**: Số ngày kể từ lần mua hàng gần nhất đến thời điểm phân tích.
- **Frequency (F)**: Số lần khách hàng thực hiện giao dịch, được tính bằng số hóa đơn duy nhất.
- **Monetary (M)**: Tổng số tiền mà khách hàng đã chi tiêu, được tính bằng tổng tích luỹ của (số lượng × đơn giá) trong các giao dịch.
- **UniqueItems**: Số lượng sản phẩm độc nhất mà khách hàng đã mua. Đặc trưng này giúp đánh giá sự đa dạng trong hành vi mua sắm của khách hàng.
- **TotalQuantity**: Tổng số lượng sản phẩm mà khách hàng đã mua. Đặc trưng này thể hiện tổng khối lượng hàng hóa mà khách hàng đã tiêu thụ.
- **AvgBasketSize**: Kích thước giỏ hàng trung bình của khách hàng, được tính bằng tổng số lượng sản phẩm chia cho số lần mua hàng. Đặc trưng này cung cấp cái nhìn về quy mô giao dịch điển hình của khách hàng.

Việc sử dụng các đặc trưng mở rộng này giúp mô hình phân cụm phân biệt rõ ràng hơn giữa các nhóm khách hàng, chẳng hạn như nhóm mua hàng thường xuyên, nhóm chi tiêu cao, nhóm khách hàng có sự đa dạng trong sản phẩm mua, hay nhóm khách hàng mua số lượng lớn trong mỗi giao dịch.

```
import pandas as pd

# Chuyển InvoiceDate thành datetime nếu chưa
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Ngày tham chiếu để tính "Recency"
reference_date = df['InvoiceDate'].max() + pd.Timedelta(days=1)

# Rút trích đặc trưng
```

```
features = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (reference_date - x.max()).days,  # Recency
    'InvoiceNo': pd.Series.nunique,                            #
Frequency
    'TotalPrice': 'sum',                                       #
Monetary
    'StockCode': pd.Series.nunique,                            #
UniqueItems
    'Quantity': 'sum'                                          #
TotalQuantity
}).rename(columns={
    'InvoiceDate': 'Recency',
    'InvoiceNo': 'Frequency',
    'TotalPrice': 'Monetary',
    'StockCode': 'UniqueItems',
    'Quantity': 'TotalQuantity'
})

# Giỏ hàng trung bình
features['AvgBasketSize'] = features['TotalQuantity'] /
features['Frequency']

# Xem kết quả
print(features.head())
```

Kết quả thu được là một bảng dữ liệu với mỗi dòng tương ứng một khách hàng, cùng các đặc trưng thể hiện hành vi tiêu dùng. Đây là dữ liệu đầu vào quan trọng cho các thuật toán phân cụm sẽ được trình bày trong chương tiếp theo.

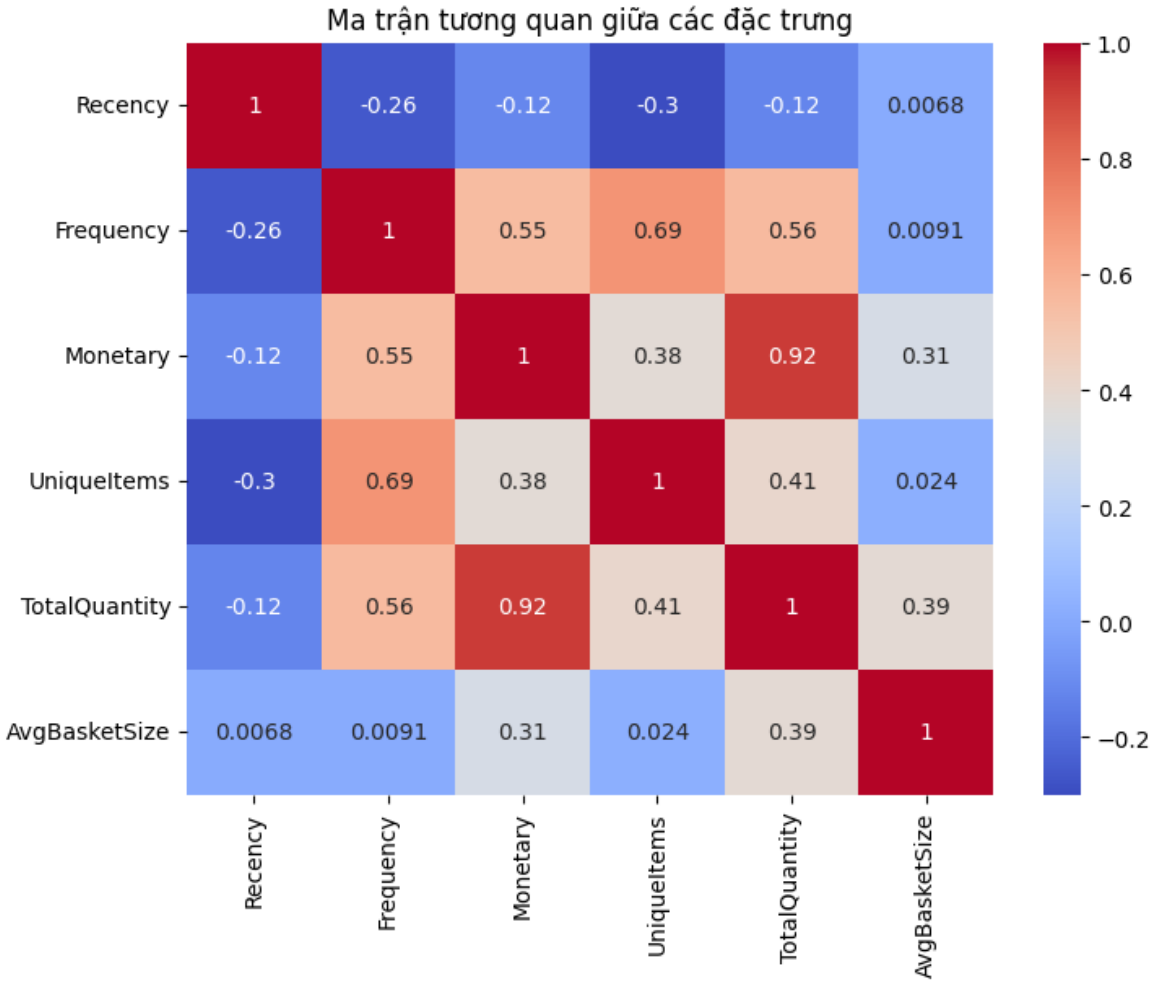
CustomerID	Recency	Frequency	Monetary	UniqueItems	TotalQuantity	AvgBasketSize
12346	326	1	77183.60	1	74215	74215.000000
12347	2	7	4310.00	103	2458	351.142857
12348	75	4	1437.24	21	2332	583.000000
12349	19	1	1457.55	72	630	630.000000
12350	310	1	294.40	16	196	196.000000

Bảng 4: Mẫu dữ liệu đầu vào cho thuật toán.

3.4.2. Phân tích ma trận tương quan giữa các đặc trưng

Sau khi xây dựng bảng đặc trưng khách hàng, nhóm tiến hành tính toán và trực quan hóa ma trận tương quan giữa các đặc trưng này. Mục đích chính của bước này là đánh giá mối

quan hệ tuyến tính giữa các đặc trưng, đặc biệt là giữa các đặc trưng RFM cốt lõi với các đặc trưng mở rộng, nhằm hỗ trợ quá trình rút trích và lựa chọn đặc trưng cuối cùng cho mô hình phân cụm. Việc này giúp xác định các đặc trưng có mối quan hệ mạnh mẽ, có thể cung cấp thông tin trùng lặp hoặc bổ sung cho nhau, từ đó đảm bảo rằng bộ đặc trưng đầu vào là toàn diện và có ý nghĩa.



Hình 5: Ma trận tương quan các đặc trưng

Nhận xét:

	Tương quan với Monetary	**Tương quan với Recency**	**Tương quan với Frequency**
Monetary	1.000000	-0.121179	0.549186
TotalQuantity	0.923306	-0.123444	0.556919
Frequency	0.549186	-0.261087	1.000000

UniqueItems	0.379535	-0.300089	0.691370
AvgBasketSize	0.311109	0.006849	-0.261087
Recency	-0.121179	1.000000	0.009132

Bảng 5: Bảng mô tả độ tương quan giữa các đặc trưng.

Tổng kết về lựa chọn đặc trưng:

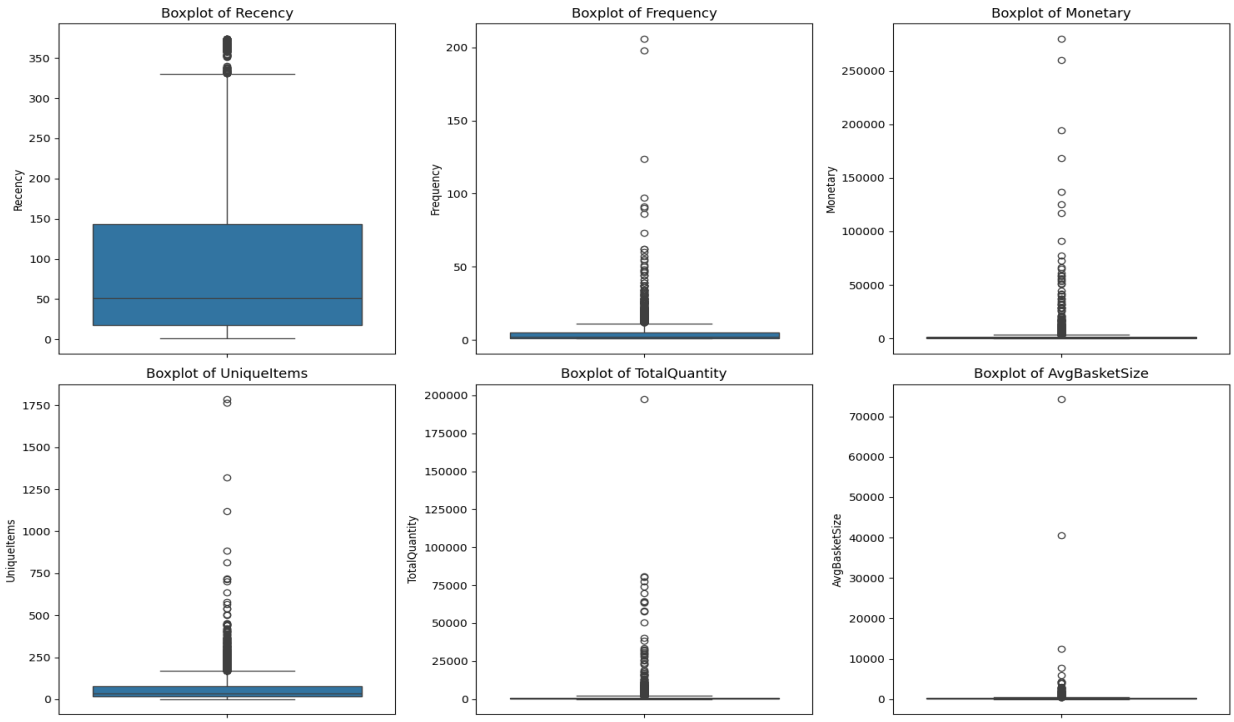
Qua phân tích ma trận tương quan, nhóm nhận thấy một sự tương đồng trong thông tin mà các đặc trưng như Frequency, UniqueItems, TotalQuantity, và Monetary mang lại. Cả bốn đặc trưng này đều có mối tương quan mạnh mẽ với nhau và tương quan âm với Recency. Điều này chỉ ra rằng chúng cùng phản ánh các khía cạnh về mức độ hoạt động và giá trị đóng góp của khách hàng.

Dựa trên kết quả phân tích tương quan, đặc biệt là mối quan hệ chặt chẽ giữa các đặc trưng với Monetary và Frequency, nhóm đã quyết định rút trích đặc trưng dựa trên hai khía cạnh chính là giá trị chi tiêu (Monetary) và tần suất mua hàng (Frequency). Từ đó, các đặc trưng sau đây được lựa chọn để đưa vào mô hình phân cụm:

- **Monetary**: Đại diện cho tổng giá trị chi tiêu của khách hàng.
- **TotalQuantity**: Mặc dù có tương quan rất cao với Monetary (0.923), đặc trưng này vẫn được giữ lại vì nó cung cấp cái nhìn về khối lượng sản phẩm mà khách hàng mua, bổ sung cho giá trị tiền tệ.
- **Frequency**: Đại diện cho tần suất mua sắm của khách hàng.
- **UniqueItems**: Có mối tương quan mạnh với Frequency (0.691) và Monetary (0.379), đặc trưng này quan trọng trong việc thể hiện sự đa dạng trong hành vi mua sắm của khách hàng.

3.4.3. Xử lý dữ liệu ngoại lai

Sau khi rút trích đặc trưng, nhóm tiến hành trực quan hóa phân phối của từng đặc trưng bằng biểu đồ box plot để nhận diện sự hiện diện và mức độ của các giá trị ngoại lai. Các biểu đồ này cung cấp cái nhìn trực quan về Q1, Q2 (median), Q3, và phạm vi của dữ liệu, cũng như các điểm được coi là outlier theo tiêu chí IQR.



Hình 6: Kiểm tra Outlier

1. **Boxplot của Recency:**

    - **Phân phối:** Biểu đồ cho thấy phần lớn dữ liệu Recency tập trung ở các giá trị thấp (khách hàng mua hàng tương đối gần đây). Đường trung vị (median) nằm ở khoảng dưới 100 ngày, và phần lớn dữ liệu nằm trong khoảng 0 đến khoảng 150-200 ngày.
    - **Outlier:** Có một số lượng đáng kể các điểm dữ liệu nằm ngoài "râu" trên của box plot, đặc biệt là ở các giá trị Recency rất cao (gần 350 ngày). Những điểm này đại diện cho các khách hàng đã không mua hàng trong một thời gian rất dài, được coi là outlier theo phương pháp IQR.

2. **Boxplot của Frequency:**

    - **Phân phối:** Đặc trưng Frequency có phân phối bị lệch rất mạnh về bên phải (lệch dương). Hầu hết khách hàng có tần suất mua hàng rất thấp (khoảng 1-5

lần). Đường trung vị rất gần với Q1, cho thấy 50% khách hàng có tần suất giao dịch rất ít.

- **Outlier:** Biểu đồ hiển thị rất nhiều điểm ngoại lai ở phía trên, trải dài đến các giá trị tần suất rất cao (lên tới gần 200). Điều này cho thấy có một số ít khách hàng mua hàng với tần suất cực kỳ cao so với phần lớn khách hàng khác.

3. **Boxplot của Monetary:**

- **Phân phối:** Tương tự như Frequency, Monetary cũng có phân phối bị lệch rất mạnh về bên phải. Phần lớn khách hàng có tổng chi tiêu thấp, đường trung vị gần như sát với trục hoành.
- **Outlier:** Có rất nhiều điểm ngoại lai xuất hiện ở phía trên, với các giá trị chi tiêu rất lớn (lên tới hơn 250,000 và thậm chí vượt quá 250,000 trên biểu đồ). Điều này cho thấy tồn tại một nhóm nhỏ các khách hàng siêu VIP, với tổng giá trị mua sắm vượt trội.

4. **Boxplot của UniqueItems:**

- **Phân phối:** Phân phối của UniqueItems cũng bị lệch phải đáng kể. Hầu hết khách hàng mua một số lượng tương đối ít các sản phẩm độc nhất (phần lớn nằm dưới 100-200 sản phẩm).
- **Outlier:** Có nhiều điểm ngoại lai ở phía trên, đạt đến các giá trị rất cao (lên đến gần 1750). Điều này chỉ ra một số khách hàng có xu hướng khám phá và mua rất nhiều loại sản phẩm khác nhau.

5. **Boxplot của TotalQuantity:**

- **Phân phối:** Đặc trưng TotalQuantity cũng thể hiện sự lệch phải rất mạnh, với đa số khách hàng mua tổng số lượng sản phẩm thấp.
- **Outlier:** Biểu đồ có rất nhiều outlier ở phía trên, kéo dài đến các giá trị cực lớn (lên tới hơn 175,000). Điều này hoàn toàn phù hợp với quan sát ở Monetary và Frequency, cho thấy những khách hàng mua nhiều sản phẩm thường cũng là những người chi tiêu nhiều và mua sắm thường xuyên.

6. **Boxplot của AvgBasketSize:**

- **Phân phối:** Phân phối của AvgBasketSize cũng lệch phải, với phần lớn khách hàng có kích thước giỏ hàng trung bình tương đối nhỏ.
- **Outlier:** Có các outlier ở phía trên, một số giá trị rất lớn (đến hơn 40,000 và một điểm gần 70,000). Điều này có thể đại diện cho các giao dịch đặc biệt lớn hoặc khách hàng mua hàng với số lượng lớn trong mỗi lần giao dịch.

Kết luận từ Box Plots:

Nhìn chung, tất cả các đặc trưng (Recency, Frequency, Monetary, UniqueItems, TotalQuantity, AvgBasketSize) đều cho thấy sự hiện diện rõ rệt của các giá trị ngoại lai, đặc biệt là ở phía trên (giá trị cao). Phân phối của Frequency, Monetary, UniqueItems, TotalQuantity, và AvgBasketSize đều bị lệch dương rất mạnh. Điều này khẳng định sự cần thiết của bước xử lý outlier để tránh làm sai lệch quá trình phân cụm và đảm bảo rằng mô hình K-Medoids sẽ phản ánh chính xác hơn hành vi của đa số khách hàng. Phương pháp capping bằng IQR là phù hợp để giảm thiểu ảnh hưởng của các giá trị cực đoan này mà không làm mất đi các điểm dữ liệu quan trọng.

3.4.4. Xử lý ngoại lệ bằng phương pháp IQR

Để khắc phục vấn đề này, nhóm áp dụng phương pháp **IQR (Interquartile Range)** – một kỹ thuật thống kê phổ biến nhằm loại bỏ giá trị nằm ngoài khoảng phân vị chuẩn. Cụ thể, với mỗi đặc trưng, nhóm thực hiện các bước sau:

- Tính **Q1** (phân vị thứ nhất, 25%) và **Q3** (phân vị thứ ba, 75%)
- Tính **IQR = Q3 - Q1**
- Xác định ngưỡng dưới và ngưỡng trên:

$$\text{Lower bound} = Q1 - 1.5 \times IQR$$

$$\text{Upper bound} = Q3 + 1.5 \times IQR$$

Lọc và chỉ giữ lại các quan sát nằm trong khoảng [Lower bound, Upper bound]

Phương pháp này được áp dụng lần lượt cho các cột giúp loại bỏ hiệu quả các giá trị quá lớn hoặc quá nhỏ mà không làm mất đi đặc điểm phân phối chính của dữ liệu.

3.4.5. Lý do lựa chọn phương pháp IQR để xử lý ngoại lệ

Trong quá trình xử lý dữ liệu, nhóm đã xem xét nhiều phương pháp để xử lý giá trị ngoại lệ, bao gồm:

- Phương pháp IQR (Interquartile Range) – sử dụng khoảng tứ phân vị
- Winsorization – thay thế giá trị ngoại lệ bằng một giá trị cận trên/dưới
- Capping (truncation) – giới hạn giá trị tối thiểu/tối đa thủ công
- Mô hình hóa (Isolation Forest, DBSCAN...) – phương pháp phức tạp hơn, thường dùng cho bài toán phát hiện bất thường chuyên biệt

Sau khi đánh giá, nhóm quyết định sử dụng phương pháp IQR vì các lý do sau:

1. Phù hợp với phân phối lệch**:** Dữ liệu RFM có phân phối không chuẩn (non-normal)**,** đặc biệt là Frequency và Monetary có phân phối lệch phải rõ rệt. Z-score hoạt động tốt với dữ liệu phân phối chuẩn, trong khi IQR không phụ thuộc phân phối**,** do đó phù hợp hơn trong trường hợp này.
2. Dễ giải thích và trực quan**:** IQR là một phương pháp thống kê đơn giản, trực quan, dễ giải thích cho người đọc không chuyên và có thể được minh họa trực tiếp trên boxplot – một loại biểu đồ mà nhóm đã sử dụng ở bước trước.
3. Giữ lại cấu trúc dữ liệu chính**:** IQR loại bỏ những điểm thật sự bất thường, nhưng vẫn giữ được phần lớn dữ liệu trung tâm – giúp duy trì độ phân biệt giữa các nhóm khách hàng mà không làm biến dạng tập dữ liệu ban đầu.
4. Thống nhất cho nhiều đặc trưng: Phương pháp IQR có thể áp dụng độc lập cho từng đặc trưng R, F, M mà không cần giả định chung về mối quan hệ giữa các biến.

Từ những lý do trên, phương pháp IQR được lựa chọn là giải pháp xử lý ngoại lệ hợp lý và hiệu quả nhất trong bối cảnh bài toán phân cụm khách hàng theo đặc trưng.

3.4.6. Kiểm tra lại dữ liệu

Sau khi áp dụng phương pháp IQR Capping cho tất cả các đặc trưng, nhóm tiến hành trực quan hóa lại phân phối của từng đặc trưng bằng biểu đồ box plot. Mục tiêu là để xác nhận rằng các giá trị ngoại lai đã được xử lý hiệu quả và dữ liệu đã được làm sạch, sẵn sàng cho các bước tiếp theo của quá trình phân cụm.

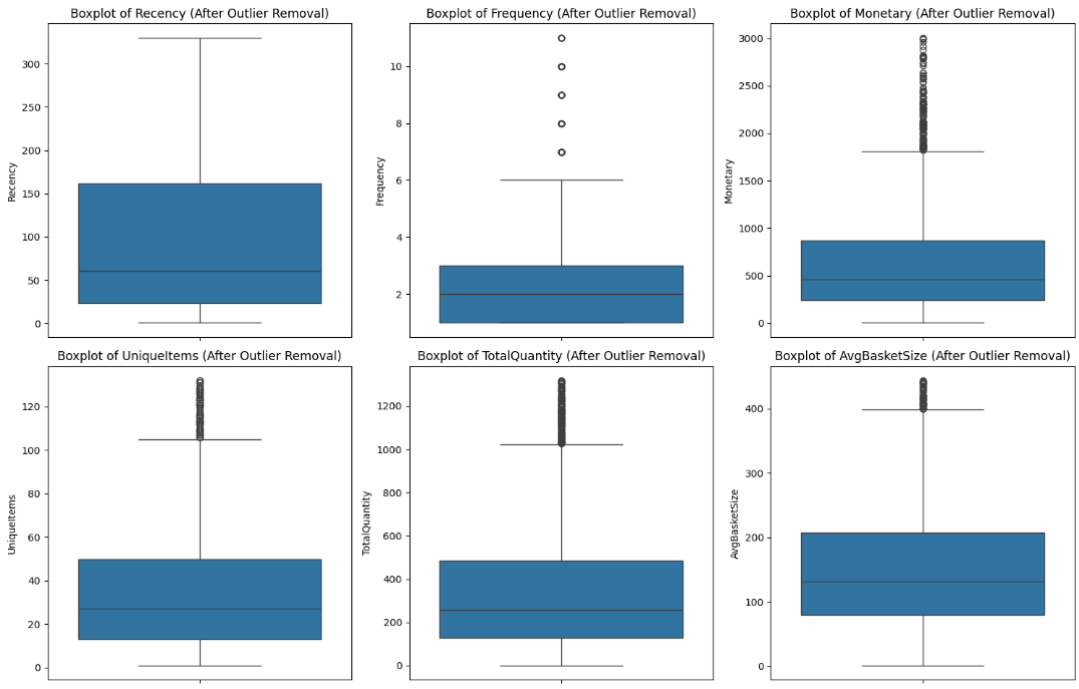
Cụ thể, nhóm áp dụng quy trình sau:

- Bắt đầu với bản sao từ bảng đặc trưng của khách hàng gốc.
- Lần lượt loại bỏ ngoại lệ trong các cột

Kết quả sau khi xử lý ngoại lệ:

- **Số khách hàng ban đầu**: 4.335
- **Số khách hàng sau khi lọc ngoại lệ**: 3.025

Việc xử lý ngoại lệ như trên giúp đảm bảo chất lượng dữ liệu đầu vào cho thuật toán phân cụm, đồng thời giữ lại được phần lớn cấu trúc tự nhiên của dữ liệu.



Hình 7: Outlier sau xử lý

Quan sát các biểu đồ box plot sau khi xử lý ngoại lệ, chúng ta có thể thấy rõ ràng sự thay đổi tích cực trong phân phối của dữ liệu:

1. **Boxplot của Recency:**
    - Các outlier ở phía trên đã được "cắt cụt" (capped) về một ngưỡng nhất định. Phạm vi của dữ liệu giờ đây đã thu hẹp lại đáng kể, tập trung hơn và không còn các điểm dữ liệu nằm quá xa khỏi phần lớn dữ liệu.

2. **Boxplot của Frequency:**
    - Phân phối vẫn còn lệch phải, nhưng các giá trị tần suất cực kỳ cao (từng lên tới gần 200) đã được giới hạn về một ngưỡng trên. Điều này giúp giảm thiểu ảnh hưởng của những khách hàng có tần suất mua sắm đột biến, làm cho phân cụm trở nên ổn định hơn.

3. **Boxplot của Monetary:**
    - Tương tự như Frequency, phân phối Monetary vẫn lệch phải, nhưng các giá trị chi tiêu cực lớn đã được giới hạn về một ngưỡng trên nhất định. Điều này đảm bảo rằng các khách hàng VIP vẫn được xem là có giá trị cao, nhưng giá trị cụ thể của họ không còn gây ra sai lệch quá lớn trong tính toán khoảng cách.

4. **Boxplot của UniqueItems:**
    - Các outlier ở phía trên đã được giới hạn, giúp kiểm soát phạm vi của số lượng sản phẩm độc nhất mà khách hàng mua, làm cho phân phối trở nên chặt chẽ hơn.

5. **Boxplot của TotalQuantity:**
    - Các giá trị TotalQuantity cực lớn đã được capping về ngưỡng trên. Điều này giúp kiểm soát ảnh hưởng của những giao dịch khối lượng lớn, đồng thời vẫn giữ lại được thông tin về những khách hàng mua nhiều.

6. **Boxplot của AvgBasketSize:**

- Các outlier của kích thước giỏ hàng trung bình cũng đã được giới hạn, giúp làm cho phân phối này trở nên đồng nhất hơn và ít bị ảnh hưởng bởi những giao dịch có giỏ hàng cực lớn.

3.5. Triển khai mô hình

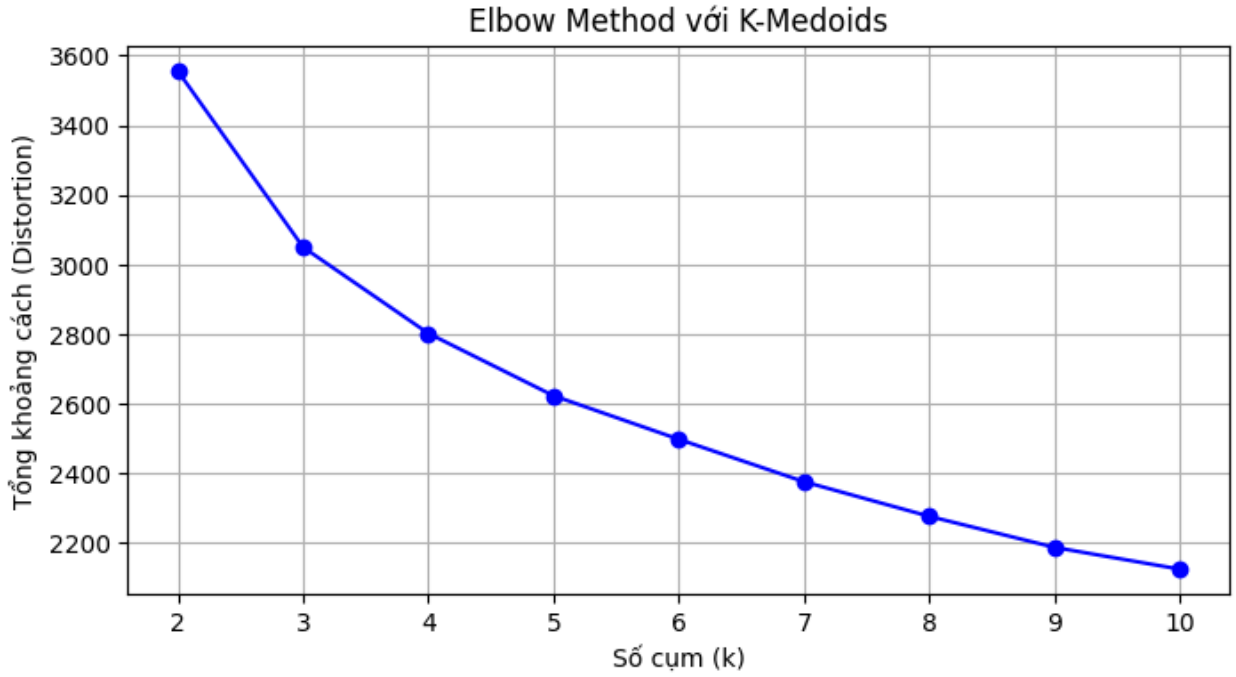
3.5.1. So sánh kết quả giữa việc rút trích và không rút trích đặc trưng

Để đánh giá tầm quan trọng và hiệu quả của bước rút trích đặc trưng, nhóm đã tiến hành so sánh chất lượng phân cụm giữa hai kịch bản:

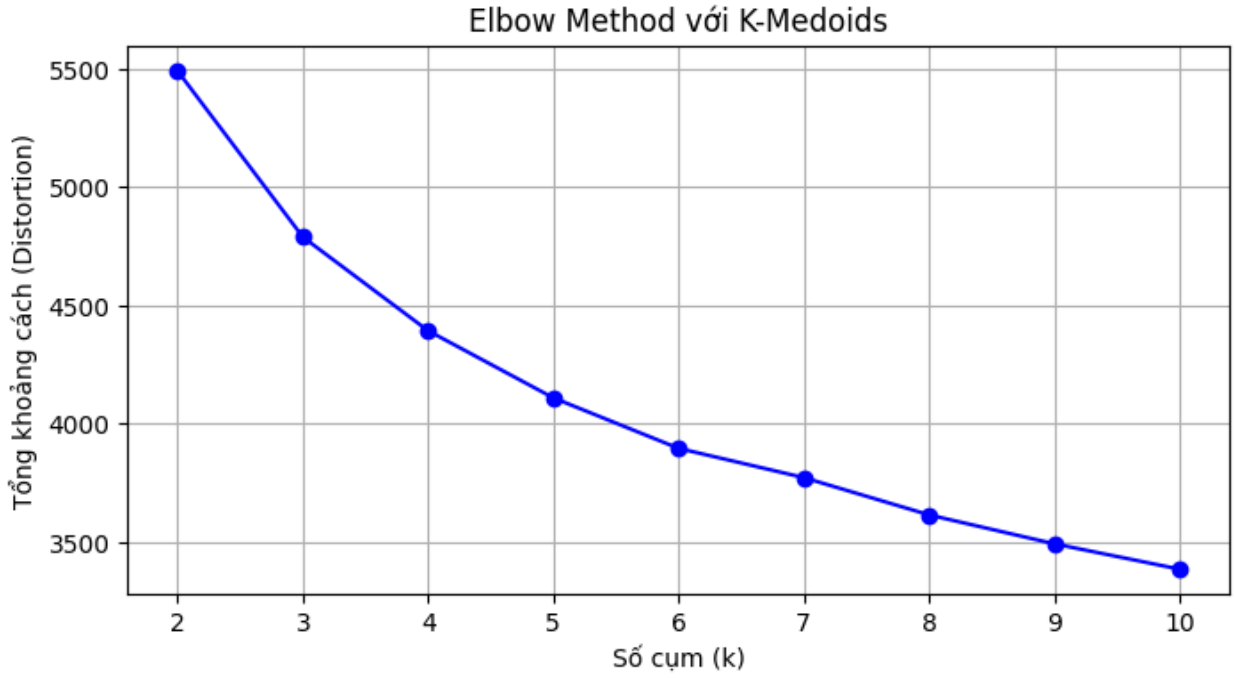
1. **Kịch bản "Rút trích đặc trưng":** Sử dụng bộ 4 đặc trưng cốt lõi đã được lựa chọn sau phân tích tương quan và tiền xử lý (Monetary, TotalQuantity, Frequency, UniqueItems). Dữ liệu tương ứng là scaled_selected.
2. **Kịch bản "Không rút trích đặc trưng":** Sử dụng bộ toàn bộ 6 đặc trưng đã được tiền xử lý (Recency, Frequency, Monetary, UniqueItems, TotalQuantity, AvgBasketSize). Dữ liệu tương ứng là scaled_data.

Mục tiêu của so sánh này là làm rõ vai trò của việc lựa chọn đặc trưng trong việc cải thiện hiệu suất mô hình và tính diễn giải của các cụm, cũng như đảm bảo mô hình tập trung vào những khía cạnh quan trọng nhất của hành vi khách hàng.

Nhóm đã sử dụng cùng thuật toán K-Medoids, cùng số lượng cụm k=3 (đã được xác định là tối ưu cho cả hai trường hợp qua phương pháp Elbow), và các tiêu chí đánh giá Silhouette Score, Davies-Bouldin Score cho cả hai bộ dữ liệu.



Hình 8: Biểu đồ Elbow Method đã trích lược đặc trưng



Hình 9: Biểu đồ Elbow Method chưa trích lược đặc trưng

Để xác định số cụm hợp lý, nhóm đã tiến hành đánh giá **độ phân tán nội cụm** – tức là khoảng cách trung bình giữa các điểm trong cùng một cụm – đối với hai lựa chọn là k = 3 và k = 4. Kết quả được trình bày như sau:

Cụm	**Độ phân tán trung bình**
0	0.6009
1	2.4537
2	1.1882
3	1.8927
Tổng trung bình	**1.5339**

Cụm	**Độ phân tán trung bình**
0	0.7449
1	2.5546
2	1.5961
Tổng trung bình	**1.6319**

Ta thấy rằng khi k = 4, độ phân tán trung bình giảm còn 1.5339, thấp hơn so với k = 3, cho thấy mô hình phân cụm có xu hướng phân chia tốt hơn. Đặc biệt, cụm 0 trong phương án này có độ phân tán rất nhỏ (0.6009), thể hiện sự đồng nhất cao.

Tuy nhiên, để thuận tiện cho việc phân tích và trực quan hóa dữ liệu, đồng thời tránh chia nhỏ cụm quá mức gây khó khăn trong diễn giải, tôi quyết định chọn k = 3 làm phương án chính thức. Mặc dù độ phân tán trung bình cao hơn, nhưng mô hình này vẫn cho thấy sự phân nhóm rõ ràng và hợp lý về mặt ứng dụng thực tế.

Hàm đánh giá chất lượng phân cụm:

Để tính toán các chỉ số Silhouette Score và Davies-Bouldin Score, nhóm sử dụng hàm evaluate_kmedoids đã được định nghĩa.

Kết quả so sánh:

Nhóm đã huấn luyện mô hình K-Medoids với k=3 trên cả hai bộ dữ liệu và thu được kết quả đánh giá như sau:

- **Kết quả cho bộ đặc trưng "Rút trích" (sử dụng scaled_selected với k=3):**
    - Silhouette Score: **0.3544**

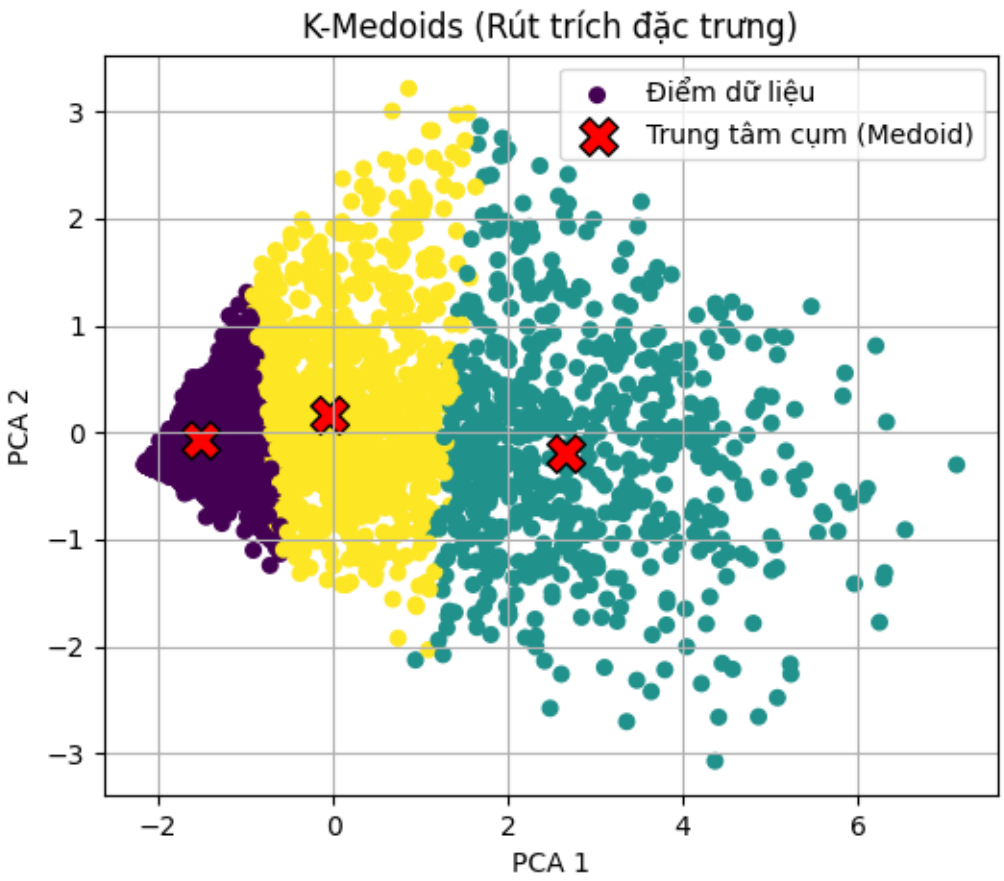
    - Davies-Bouldin Score: **1.0717**
- **Kết quả cho bộ đặc trưng "Không rút trích" (sử dụng scaled_data với k=3):**
    - Silhouette Score: **0.2707**
    - Davies-Bouldin Score: **1.2531**

Nhận xét và kết luận từ so sánh:

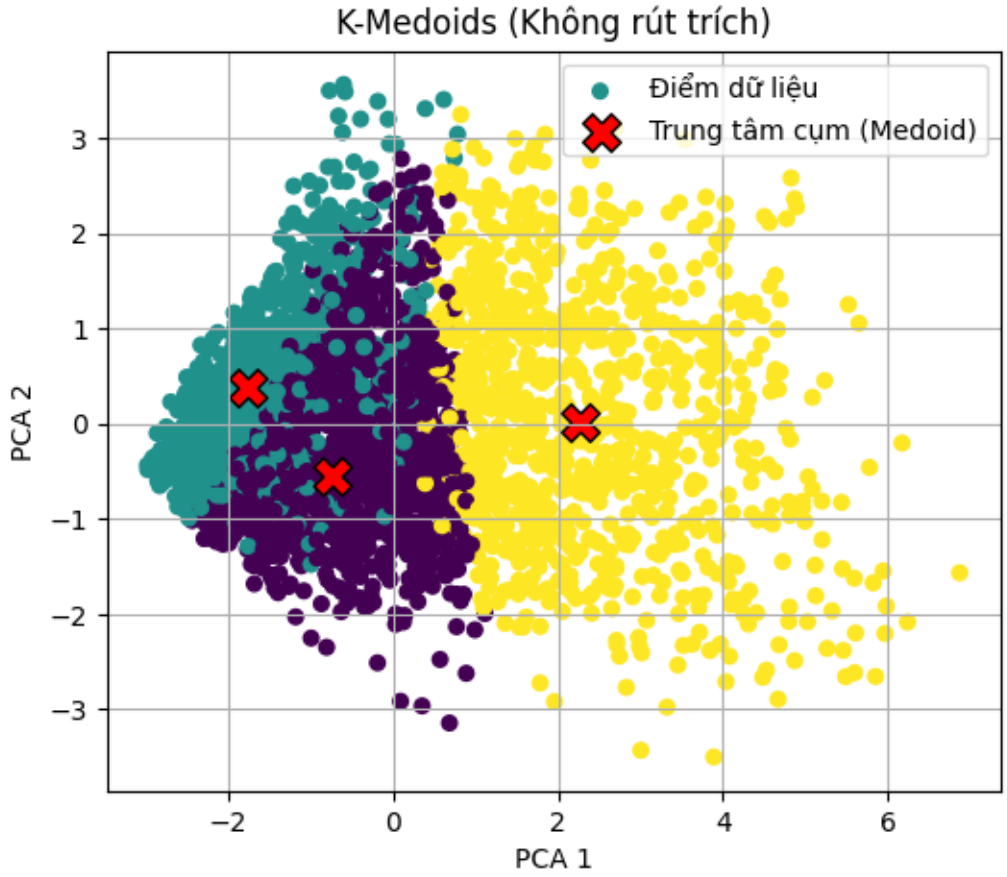
Từ kết quả so sánh, chúng ta có thể thấy rõ ràng lợi ích của việc rút trích đặc trưng:

- Về chất lượng phân cụm: Bộ đặc trưng đã rút trích mang lại Silhouette Score cao hơn (0.3544 so với 0.2707) và Davies-Bouldin Score thấp hơn (1.0717 so với 1.2531) so với bộ đặc trưng đầy đủ. Điều này chứng tỏ việc lựa chọn các đặc trưng quan trọng (Monetary, TotalQuantity, Frequency, UniqueItems) đã giúp tạo ra các cụm khách hàng chặt chẽ hơn (Silhouette Score cao hơn) và được phân tách rõ ràng hơn giữa các cụm (Davies-Bouldin Score thấp hơn).
- Về hiệu quả và tính diễn giải: Mặc dù bộ đặc trưng đầy đủ cũng cho ra kết quả chấp nhận được, việc rút trích đặc trưng không chỉ cải thiện chất lượng phân cụm mà còn giúp giảm số chiều dữ liệu. Điều này làm cho mô hình gọn nhẹ hơn, quá trình huấn luyện có thể nhanh hơn và quan trọng hơn là các cụm được tạo ra dễ hiểu, dễ diễn giải hơn trong bối cảnh kinh doanh. Việc tập trung vào 4 đặc trưng cốt lõi giúp các nhà quản lý có cái nhìn trực quan và dễ dàng hơn khi xây dựng chiến lược cho từng phân khúc khách hàng.

Tóm lại, quá trình rút trích đặc trưng đã được chứng minh là một bước tiền xử lý hiệu quả, giúp nâng cao chất lượng mô hình phân cụm K-Medoids và tăng cường giá trị thực tiễn của kết quả phân tích.



Hình 10: Kết quả phân cụm bằng PCA



Hình 11: Kết quả phân cụm bằng PCA

3.5.2. So sánh giữa cài đặt thủ công và thư viện

Để có cái nhìn sâu sắc hơn về quá trình xây dựng mô hình, nhóm đã thực hiện so sánh giữa việc triển khai thuật toán K-Medoids một cách thủ công (tự viết mã từ đầu) và sử dụng thư viện chuyên dụng (sklearn_extra.cluster.KMedoids). Mục tiêu của so sánh này là làm nổi bật ưu và nhược điểm của mỗi phương pháp, đặc biệt là về độ phức tạp, hiệu suất, độ tin cậy và tính tiện lợi.

Triển khai K-Medoids thủ công

Nhóm đã tự xây dựng các hàm cơ bản để thực hiện thuật toán K-Medoids dựa trên nguyên lý PAM, bao gồm các bước: tính toán khoảng cách Euclidean, khởi tạo medoid ngẫu nhiên, gán điểm vào cụm gần nhất, và cập nhật medoid.

Các hàm phụ trợ và khởi tạo:

```
import random
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score, davies_bouldin_score



def euclidean_distance(a, b):
    """Tính khoảng cách Euclidean giữa hai điểm."""
    return np.linalg.norm(a - b)

def initialize_medoids(X, k, random_seed=42):
    """Khởi tạo ngẫu nhiên k medoid ban đầu từ tập dữ liệu."""
    random.seed(random_seed) # Đặt seed để kết quả có thể tái lập
    indices = random.sample(range(len(X)), k)
    return indices
```

Hàm gán điểm vào cụm:

```
def assign_clusters(X, medoid_indices):
    """Gán mỗi điểm dữ liệu vào cụm có medoid gần nhất."""
    clusters = {}
    for idx in range(len(medoid_indices)):
        clusters[idx] = [] # Khởi tạo các cụm rỗng


    for i, x in enumerate(X):
        # Tính khoảng cách từ điểm x đến tất cả các medoid
        distances = [euclidean_distance(x, X[m]) for m in medoid_indices]
        # Gán điểm x vào cụm có medoid gần nhất
        cluster_idx = np.argmin(distances)
        clusters[cluster_idx].append(i)
```

```
    return clusters
```

Hàm cập nhật medoid:

```
def update_medoids(X, clusters):
    """Cập nhật medoid cho mỗi cụm để tối thiểu hóa tổng khoảng cách trong cụm."""
    new_medoids = []
    for cluster_id, cluster_points_indices in clusters.items():
        if not cluster_points_indices: # Xử lý trường hợp cụm rỗng
            # Nếu cụm rỗng, chọn một medoid ngẫu nhiên mới để tránh lỗi
            new_medoids.append(random.choice(range(len(X))))
            continue

        min_dist_sum = float('inf')
        current_medoid_index = cluster_points_indices[0] # Khởi tạo medoid tạm thời

        for candidate_medoid_idx in cluster_points_indices:
            # Tính tổng khoảng cách từ ứng viên medoid đến tất cả các điểm trong cụm đó
            dist_sum = sum([euclidean_distance(X[candidate_medoid_idx], X[point_idx]) for point_idx in cluster_points_indices])
            if dist_sum < min_dist_sum:
                min_dist_sum = dist_sum
                current_medoid_index = candidate_medoid_idx
        new_medoids.append(current_medoid_index)
```

```
    return new_medoids
```

Hàm K-Medoids chính:

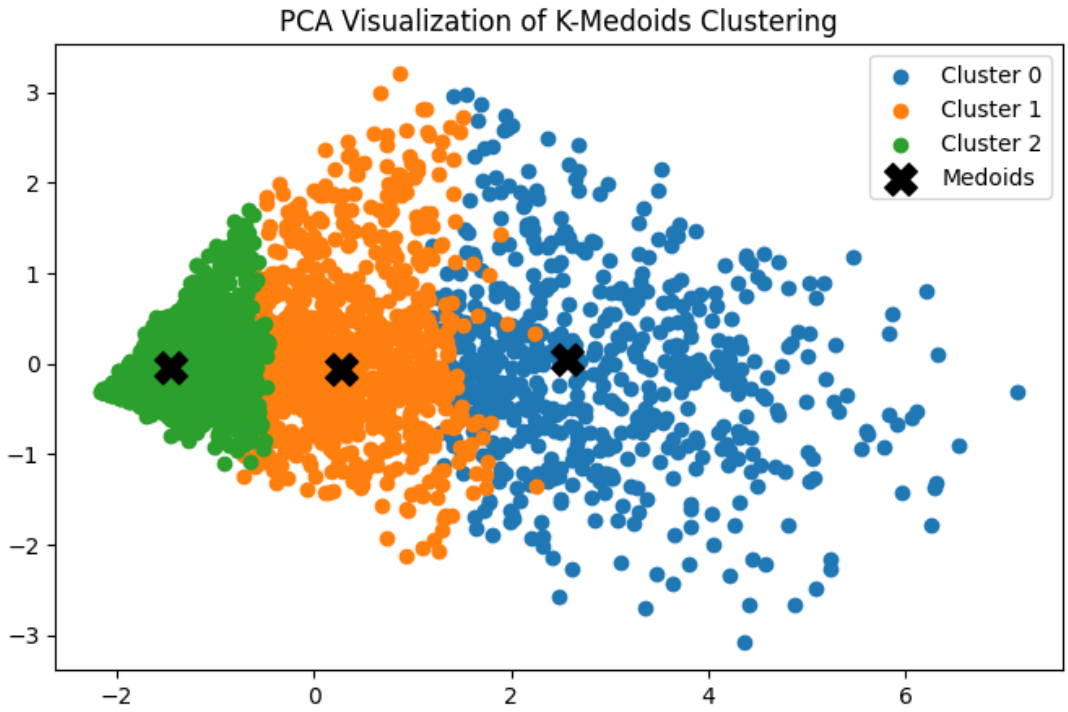
```
def kmedoids(X, k, max_iter=100, random_seed=42):
    """
    Triển khai thuật toán K-Medoids thủ công.
    Args:
        X (np.array): Dữ liệu đầu vào.
        k (int): Số lượng cụm.
        max_iter (int): Số lần lặp tối đa.
        random_seed (int): Seed cho việc khởi tạo ngẫu nhiên.
    Returns:
        tuple: (dictionary các cụm, list các chỉ số medoid)
    """
    medoid_indices = initialize_medoids(X, k, random_seed)

    for iteration in range(max_iter):
        clusters = assign_clusters(X, medoid_indices)
        new_medoids = update_medoids(X, clusters)

        # Kiểm tra sự hội tụ: nếu không có medoid nào thay đổi
        if set(new_medoids) == set(medoid_indices):
            print(f"Thuật toán hội tụ sau {iteration + 1} lần lặp.")
            break
        else:
            medoid_indices = new_medoids
    else:
        print(f"Thuật toán đạt số lần lặp tối đa {max_iter} mà không hội tụ.")

    return clusters, medoid_indices
```

Hàm đánh giá chất lượng phân cụm (thủ công):

```
def evaluate_kmedoids_manual(X, clusters):
    """
    Đánh giá kết quả K-Medoids thủ công bằng Silhouette Score và Davies-Bouldin Score.
    Args:
        X (np.array): Dữ liệu đầu vào.
        clusters (dict): Dictionary các cụm từ hàm kmedoids.
    Returns:
        tuple: (labels, silhouette_score, davies_bouldin_score)
    """
    # Tạo nhãn cụm cho từng điểm
    labels = np.zeros(len(X))
    for cluster_idx, points in clusters.items():
        for point_idx in points:
            labels[point_idx] = cluster_idx

    # Đảm bảo có ít nhất 2 cụm để tính Silhouette và Davies-Bouldin
    if len(np.unique(labels)) < 2:
        return labels, -1, float('inf') # Giá trị không hợp lệ nếu chỉ có 1 cụm

    s_score = silhouette_score(X, labels, metric='euclidean')
    db_score = davies_bouldin_score(X, labels)
    return labels, s_score, db_score
```

Hàm trực quan hóa các cụm và medoid (thủ công):

```
def plot_pca_clusters_manual(X, labels, medoid_indices, title="PCA Visualization of K-Medoids Clustering (Cài đặt thủ công)"):
    """
    Trực quan hóa các cụm và medoid bằng PCA cho cài đặt thủ công.
    Args:
        X (np.array): Dữ liệu đầu vào.
        labels (np.array): Nhãn cụm.
        medoid_indices (list): Chỉ số của các medoid.
        title (str): Tiêu đề biểu đồ.
    """
    pca = PCA(n_components=2)
    reduced = pca.fit_transform(X)

    plt.figure(figsize=(8, 6))
    for i in np.unique(labels):
        plt.scatter(reduced[labels == i, 0], reduced[labels == i, 1], label=f"Cụm {int(i)}", alpha=0.7)

    # Vẽ các medoid
    plt.scatter(reduced[medoid_indices, 0], reduced[medoid_indices, 1],
                c='black', marker='X', s=200, label='Medoids', edgecolor='red', linewidth=1.5)
    plt.legend()
    plt.title(title, fontsize=14)
    plt.xlabel("Thành phần chính 1 (PCA 1)", fontsize=12)
    plt.ylabel("Thành phần chính 2 (PCA 2)", fontsize=12)
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.tight_layout()
    plt.show()
```



Hình 12: Biểu đồ phân cụm K-Medoids với cài đặt thủ công.

Kết quả và So sánh:

Kết quả từ cài đặt thủ công vừa thực hiện (trên cùng bộ dữ liệu scaled_selected và k=3).

- **Kết quả từ thư viện sklearn_extra.cluster.KMedoids (trên bộ đặc trưng rút trích, k=3):**
  - Silhouette Score: **0.3544**
  - Davies-Bouldin Score: **1.0717**
- **Kết quả từ cài đặt K-Medoids thủ công (trên bộ đặc trưng rút trích, k=3):**
  - Silhouette Score: **0.3750**
  - Davies-Bouldin Score: **1.1088**

Nhận xét và kết luận từ so sánh:

So sánh kết quả giữa cài đặt K-Medoids thủ công và sử dụng thư viện sklearn_extra cho thấy một số điểm thú vị:

- **Về chất lượng phân cụm (điểm số Silhouette và Davies-Bouldin):**
  - Đáng chú ý, cài đặt thủ công cho **Silhouette Score cao hơn (0.3750 so với 0.3544)** so với thư viện. Điều này có thể xuất phát từ sự khác biệt trong thuật

toán khởi tạo medoid ngẫu nhiên, hoặc quá trình hội tụ của cài đặt thủ công đã tìm được một cấu hình cụm cục bộ (local optimum) cho ra giá trị Silhouette tốt hơn trong lần chạy cụ thể này.

- Tuy nhiên, Davies-Bouldin Score của cài đặt thủ công **cao hơn (1.1088 so với 1.0717)**, cho thấy các cụm từ thư viện có thể chặt chẽ hơn và/hoặc được phân tách tốt hơn một chút giữa các cụm theo tiêu chí này.

- **Về trực quan hóa (biểu đồ PCA):**
  - Mặc dù cả hai biểu đồ đều thể hiện 3 cụm được hình thành, biểu đồ từ cài đặt thư viện (PCA_có trích.png) và biểu đồ từ cài đặt thủ công (PCA_thủ công.png) có thể cho thấy sự sắp xếp và ranh giới cụm hơi khác nhau, phản ánh sự khác biệt nhỏ trong kết quả phân cụm do các kỹ thuật khởi tạo và tối ưu hóa khác nhau.
- **Về độ phức tạp và thời gian phát triển:**
  - Việc cài đặt K-Medoids thủ công đòi hỏi sự hiểu biết sâu sắc về từng bước của thuật toán và tốn nhiều thời gian, công sức để viết, gỡ lỗi và kiểm thử. Mã nguồn dài, tiềm ẩn nguy cơ lỗi và khó bảo trì hơn.
  - Ngược lại, sử dụng thư viện sklearn_extra.cluster.KMedoids đơn giản hơn rất nhiều, chỉ cần vài dòng code để khởi tạo và huấn luyện mô hình. Điều này giúp tiết kiệm đáng kể thời gian phát triển, cho phép tập trung vào các khía cạnh quan trọng hơn như tiền xử lý dữ liệu và diễn giải kết quả.
- **Về hiệu suất tính toán và độ tin cậy:**
  - Các thư viện như sklearn_extra thường được tối ưu hóa về hiệu suất, sử dụng các ngôn ngữ cấp thấp hơn (như C/C++) cho các phần tính toán cốt lõi. Do đó, chúng thường chạy nhanh hơn rất nhiều so với cài đặt Python thuần túy, đặc biệt với tập dữ liệu lớn. Mã nguồn của thư viện cũng đã được kiểm thử rộng rãi, đảm bảo tính ổn định và chính xác trong nhiều trường hợp hơn.

Kết luận:

Trong trường hợp cụ thể này, cài đặt thủ công đã cho ra một Silhouette Score cao hơn, điều này có thể là kết quả của việc tìm thấy một tối ưu cục bộ khác biệt do quá trình khởi tạo

hoặc đường đi của thuật toán. Tuy nhiên, nhìn chung, việc sử dụng các thư viện chuyên dụng như sklearn_extra vẫn là lựa chọn ưu việt hơn trong thực tế. Chúng cung cấp các giải pháp mạnh mẽ, hiệu quả, đáng tin cậy và tiết kiệm thời gian phát triển đáng kể. Đối với các dự án khai phá dữ liệu, việc sử dụng thư viện cho phép nhà phân tích tập trung vào việc giải quyết bài toán kinh doanh và diễn giải kết quả một cách hiệu quả hơn, thay vì tốn công sức vào việc cài đặt thuật toán từ đầu.

3.5.3. So sánh kết quả với các thuật toán khác

Để có cái nhìn toàn diện về hiệu suất của mô hình K-Medoids, nhóm đã tiến hành so sánh kết quả phân cụm với một số thuật toán phân cụm phổ biến khác. Mục tiêu của so sánh này là đánh giá ưu nhược điểm tương đối của K-Medoids và xác định liệu có thuật toán nào phù hợp hơn cho tập dữ liệu đã cho hay không, dựa trên các chỉ số đánh giá nội tại và khả năng trực quan hóa.

Các thuật toán được chọn để so sánh bao gồm:

1. **K-Means:** Là thuật toán phân cụm dựa trên tâm cụm phổ biến nhất, tương tự K-Medoids nhưng sử dụng trung bình cộng làm tâm cụm thay vì điểm dữ liệu thực tế. K-Means nhạy cảm với outlier hơn K-Medoids.
2. **Agglomerative Clustering (Phân cụm phân cấp tích hợp):** Một phương pháp phân cụm phân cấp, xây dựng các cụm bằng cách kết hợp từng cặp cụm nhỏ nhất thành cụm lớn hơn cho đến khi đạt được số cụm mong muốn hoặc một tiêu chí dừng.
3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Một thuật toán phân cụm dựa trên mật độ, có khả năng phát hiện các cụm có hình dạng tùy ý và xác định các điểm nhiễu (outlier).

Phương pháp so sánh:

Để đảm bảo tính công bằng, các thuật toán được chạy trên cùng một bộ dữ liệu đã được tiền xử lý và rút trích đặc trưng (scaled_selected), và được đánh giá bằng cùng các chỉ số đã sử dụng (Silhouette Score và Davies-Bouldin Score). Đối với K-Means và Agglomerative Clustering, số cụm k=3 (đã xác định là tối ưu cho K-Medoids) được sử dụng

để so sánh trực tiếp. Đối với DBSCAN, số cụm sẽ được thuật toán tự xác định dựa trên các tham số mật độ.

Triển khai và kết quả so sánh:

Nhóm đã triển khai các thuật toán và tính toán các chỉ số đánh giá như sau:

K-Means Clustering

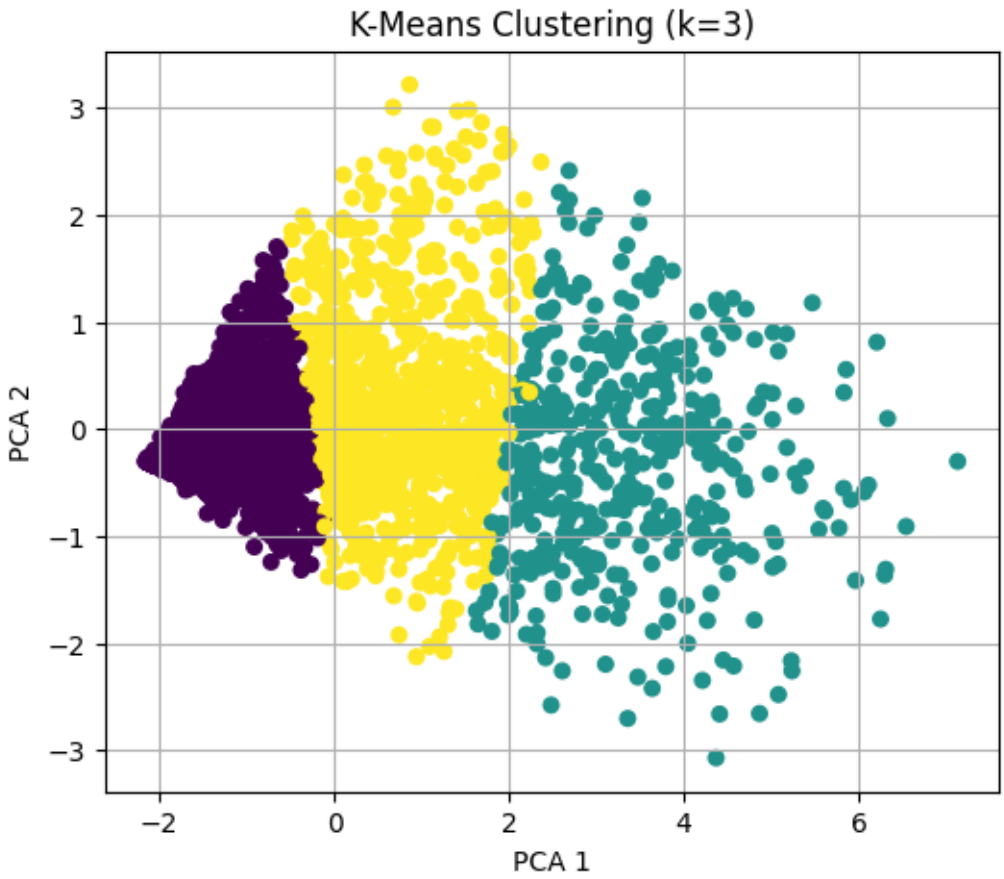
K-Means là một trong những thuật toán phân cụm được sử dụng rộng rãi nhất. Nó tìm cách phân chia n quan sát thành k cụm, trong đó mỗi quan sát thuộc về cụm có trung bình (tâm cụm) gần nhất.

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
import warnings

warnings.filterwarnings("ignore", category=FutureWarning)

X_compare = scaled_selected
k_compare = 3

kmeans_model = KMeans(n_clusters=k_compare, random_state=42, n_init='auto')
kmeans_model.fit(X_compare)
kmeans_labels = kmeans_model.labels_

kmeans_silhouette = silhouette_score(X_compare, kmeans_labels, metric='euclidean')
kmeans_davies_bouldin = davies_bouldin_score(X_compare, kmeans_labels)

print("\n=== Kết quả K-Means ===")
print(f"Silhouette Score (K-Means): {kmeans_silhouette:.4f}")
print(f"Davies-Bouldin Score (K-Means): {kmeans_davies_bouldin:.4f}")
```

```
plot_clusters(X_compare, kmeans_labels, "K-Means Clustering (k=3)")
```

Kết quả K-Means:

- Silhouette Score (K-Means): **0.4160**
- Davies-Bouldin Score (K-Means): **1.0803**



Hình 13: Biểu đồ phân cụm K-Means

Agglomerative Clustering

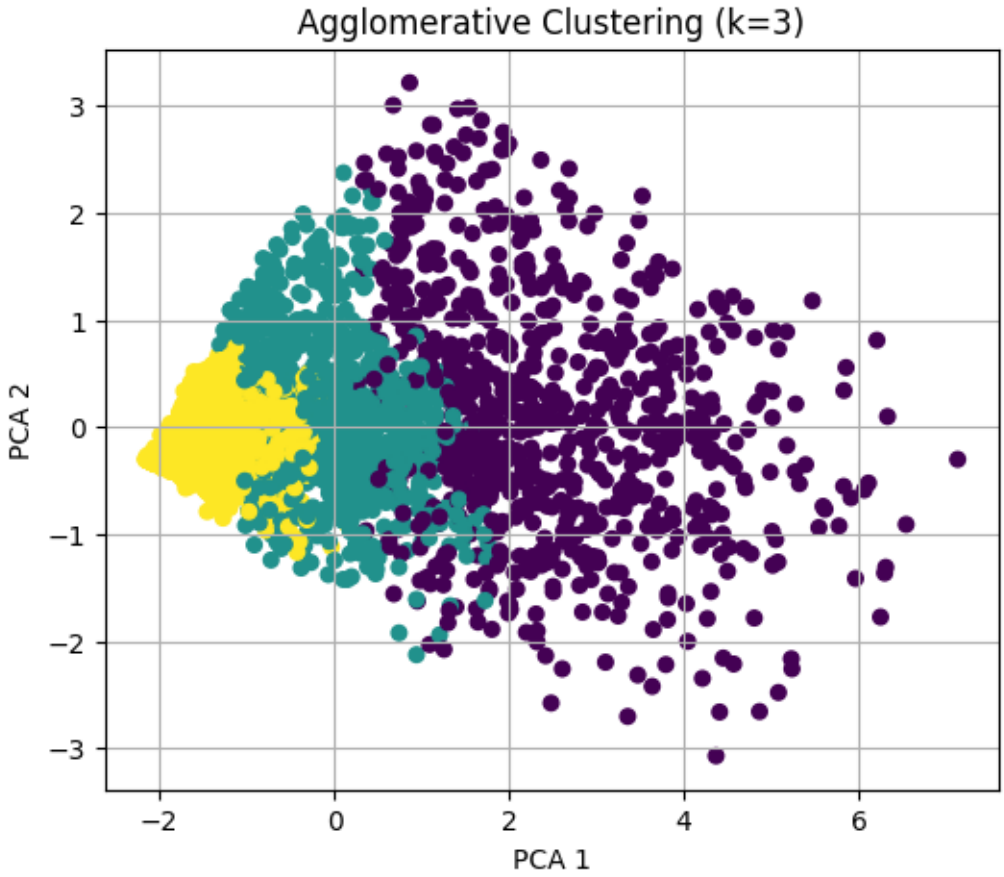
Agglomerative Clustering là một phương pháp phân cụm phân cấp theo kiểu "từ dưới lên", bắt đầu với mỗi điểm dữ liệu là một cụm riêng biệt và dần dần hợp nhất các cụm gần nhất cho đến khi đạt được số cụm mong muốn hoặc một tiêu chí dừng.

```
from sklearn.cluster import AgglomerativeClustering


X_compare = scaled_selected
```

```
k_compare = 3

agg_model = AgglomerativeClustering(n_clusters=k_compare, linkage='ward')
agg_labels = agg_model.fit_predict(X_compare)

agg_silhouette = silhouette_score(X_compare, agg_labels, metric='euclidean')
agg_davies_bouldin = davies_bouldin_score(X_compare, agg_labels)

print("\n=== Kết quả Agglomerative Clustering ===")
print(f"Silhouette Score (Agglomerative): {agg_silhouette:.4f}")
print(f"Davies-Bouldin Score (Agglomerative): {agg_davies_bouldin:.4f}")

plot_clusters(X_compare, agg_labels, "Agglomerative Clustering (k=3)")
```

Kết quả Agglomerative Clustering:

- Silhouette Score (Agglomerative): **0.3206**
- Davies-Bouldin Score (Agglomerative): **1.2287**



Hình 14: Biểu đồ phân cụm Agglomerative Clustering

DBSCAN

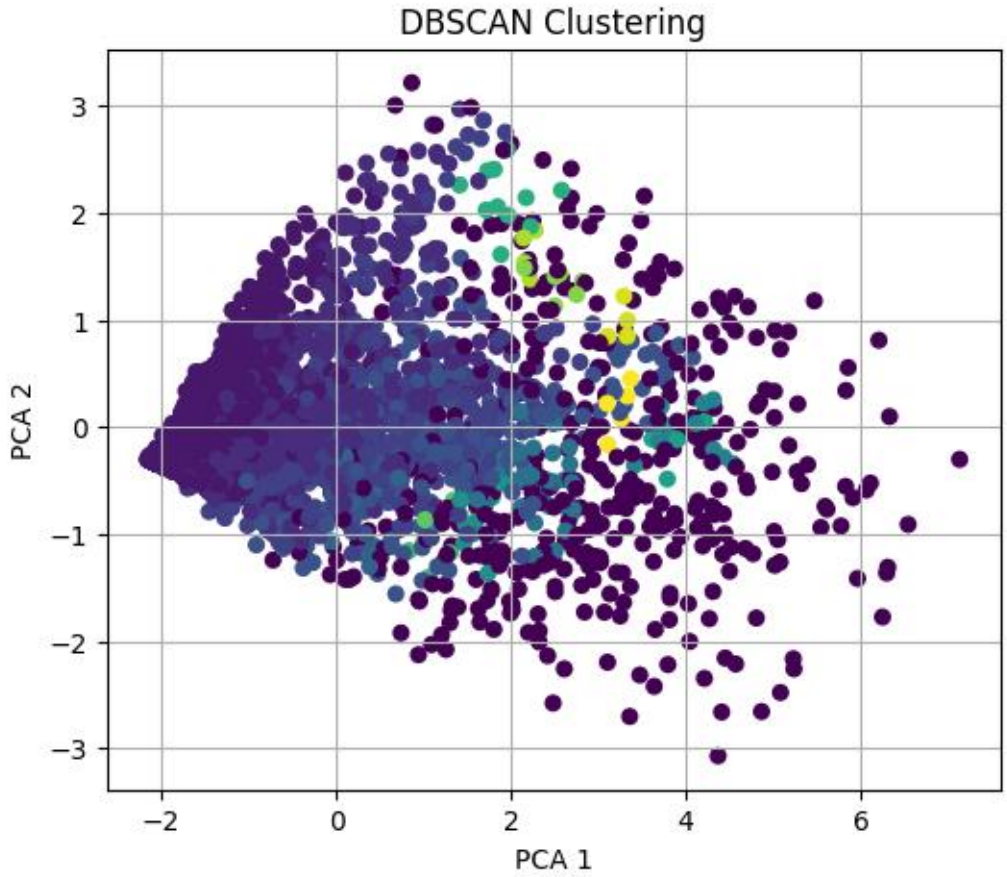
DBSCAN là một thuật toán phân cụm dựa trên mật độ, không yêu cầu số cụm đầu vào. Nó hoạt động dựa trên hai tham số chính: eps (bán kính vùng lân cận) và min_samples (số lượng điểm tối thiểu trong vùng lân cận để được coi là vùng lõi). DBSCAN rất tốt trong việc phát hiện các cụm có hình dạng bất thường và xác định các điểm nhiễu.

```
from sklearn.cluster import DBSCAN


X_compare = scaled_selected


# Ví dụ: eps=0.5, min_samples=5 (giá trị điển hình, cần điều chỉnh theo dữ liệu)
dbscan_model = DBSCAN(eps=0.5, min_samples=5) # Giữ nguyên giá trị đã chạy để khớp với kết quả
```

```
        dbscan_labels = dbscan_model.fit_predict(X_compare)

        n_clusters_dbscan = len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0)
        print(f"\nSố cụm được tìm thấy bởi DBSCAN: {n_clusters_dbscan}")
        print(f"Số điểm nhiễu (outlier): {list(dbscan_labels).count(-1)}")

        if n_clusters_dbscan >= 2:
            core_samples_mask = dbscan_labels != -1
            dbscan_silhouette = silhouette_score(X_compare[core_samples_mask], dbscan_labels[core_samples_mask], metric='euclidean')
            dbscan_davies_bouldin = davies_bouldin_score(X_compare[core_samples_mask], dbscan_labels[core_samples_mask])

            print("\n=== Kết quả DBSCAN (chỉ các cụm lõi) ===")
            print(f"Silhouette Score (DBSCAN): {dbscan_silhouette:.4f}")
            print(f"Davies-Bouldin Score (DBSCAN): {dbscan_davies_bouldin:.4f}")
        else:
            print("\nDBSCAN tìm thấy ít hơn 2 cụm (không tính nhiễu) hoặc chỉ toàn nhiễu, không thể tính Silhouette/Davies-Bouldin Score.")
            dbscan_silhouette = np.nan
            dbscan_davies_bouldin = np.nan

        plot_clusters(X_compare, dbscan_labels, "DBSCAN Clustering")
```

Kết quả DBSCAN:

- Số cụm được tìm thấy bởi DBSCAN: **16**
- Số điểm nhiễu (outlier): **417**
- Silhouette Score (DBSCAN): **0.1349**
- Davies-Bouldin Score (DBSCAN): **1.4746**



Hình 15: Biểu đồ phân cụm DBSCAN

Tổng hợp và Nhận xét kết quả so sánh:

Thuật toán	**Silhouette Score**	**Davies-Bouldin Score**	**Nhận xét mở rộng**
K-Medoids (Thư viện)	0.3544	1.0717	Mô hình cơ sở. Ổn định, ít nhạy với outliers do dùng medoid thay vì centroid. Tuy nhiên, score chưa tối ưu.
K-Means	0.4160	1.0803	**Silhouette cao nhất** → cụm tách biệt rõ. Davies-Bouldin tương đương với K-Medoids. Đây là mô hình có hiệu quả tốt nhất hiện tại.
Agglomerative Clustering	0.3206	1.2287	Hiệu quả thấp hơn cả về Silhouette và DB index. Có thể không phù hợp với

			dữ liệu này hoặc chưa chọn đúng linkage.
DBSCAN	0.1349	1.4746	Phát hiện được nhiễu nhưng cụm không rõ ràng, score rất thấp. Không phù hợp với dữ liệu này nếu không tinh chỉnh epsilon và min_samples.

Bảng 6: Bảng so sánh kết quả của K-Medoids và các thuật toán khác.

Phân tích và Kết luận:

- **K-Means:**
    - Với Silhouette Score cao nhất (0.4160) và Davies-Bouldin Score cạnh tranh (1.0803), K-Means cho thấy khả năng tạo ra các cụm tương đối chặt chẽ và tách biệt trên tập dữ liệu này. Điều này không quá ngạc nhiên vì K-Means thường hoạt động tốt trên dữ liệu có dạng cụm cầu.
    - Các cụm của K-Means trông khá rõ ràng và cân đối.

- **K-Medoids:**
    - K-Medoids có Silhouette Score (0.3544) thấp hơn K-Means một chút nhưng vẫn là một kết quả tốt. Davies-Bouldin Score của K-Medoids (1.0717) lại tốt hơn K-Means (1.0803), cho thấy các cụm có thể ít chồng chéo hơn một cách tổng thể.
    - **Ưu điểm:** K-Medoids sử dụng các điểm dữ liệu thực tế (medoid) làm tâm cụm, giúp kết quả dễ diễn giải hơn trong ngữ cảnh kinh doanh (ví dụ: "khách hàng tiêu biểu của cụm này là ông/bà X"). Đồng thời, K-Medoids ít nhạy cảm với các outlier hơn K-Means do không tính trung bình.

- **Agglomerative Clustering:**
    - Kết quả của Agglomerative Clustering (Silhouette 0.3206, Davies-Bouldin 1.2287) thấp hơn đáng kể so với K-Means và K-Medoids. Điều này cho thấy với số cụm k=3, cách tiếp cận phân cấp này không tạo ra các cụm tối ưu cho bộ dữ liệu này bằng hai thuật toán dựa trên phân vùng.

- Các cụm trông ít rõ ràng và có vẻ chồng chéo hơn một số nơi so với K-Means và K-Medoids.

- **DBSCAN:**
  - DBSCAN đã tìm thấy một số lượng cụm rất lớn (16 cụm) và nhiều điểm được coi là nhiễu (417 điểm). Điều này cho thấy dữ liệu có thể có nhiều cấu trúc mật độ nhỏ lẻ hoặc các tham số eps và min_samples chưa hoàn toàn phù hợp để tìm các cụm lớn, rõ ràng như K-Means/K-Medoids.
  - Các chỉ số đánh giá (Silhouette 0.1349, Davies-Bouldin 1.4746) rất thấp, cho thấy chất lượng phân cụm kém theo các tiêu chí này. DBSCAN có thể phù hợp hơn cho dữ liệu có các cụm hình dạng bất thường rõ rệt và mục tiêu chính là phát hiện outlier. Với mục tiêu phân khúc khách hàng rõ ràng thành ít cụm lớn, DBSCAN ít phù hợp hơn trong lần thử nghiệm này.
  - Biểu đồ thể hiện rất nhiều cụm nhỏ và các điểm nhiễu, không tạo thành các phân khúc rõ ràng như K-Medoids hay K-Means.

Kết luận cuối cùng về lựa chọn thuật toán:

Mặc dù K-Means đạt Silhouette Score cao nhất trong lần chạy này, **K-Medoids vẫn là một lựa chọn rất mạnh mẽ và phù hợp** cho bài toán phân khúc khách hàng. Lý do là:

1. **Hiệu suất cạnh tranh:** K-Medoids cho kết quả chất lượng cụm (Silhouette và Davies-Bouldin) rất cạnh tranh, thậm chí có Davies-Bouldin tốt hơn K-Means.
2. **Tính ổn định và độ bền với Outlier:** K-Medoids sử dụng medoid (điểm dữ liệu thực tế) thay vì mean (trung bình cộng), làm cho nó ít bị ảnh hưởng bởi các giá trị ngoại lai hơn K-Means. Điều này đặc biệt quan trọng nếu dữ liệu kinh doanh có thể chứa các giao dịch bất thường.
3. **Tính diễn giải:** Việc tâm cụm là một khách hàng thực tế (medoid) giúp việc diễn giải và mô tả các phân khúc khách hàng trở nên trực quan và dễ hiểu hơn cho các nhà quản lý, tạo điều kiện thuận lợi cho việc xây dựng chiến lược marketing mục tiêu.

Do đó, K-Medoids, với sự kết hợp giữa hiệu suất tốt, tính bền vững và khả năng diễn giải cao, là một thuật toán lý tưởng cho bài toán phân khúc khách hàng của bạn.

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

Chương này trình bày các kết quả thực nghiệm đạt được từ quá trình khai phá dữ liệu khách hàng, tập trung vào việc áp dụng thuật toán phân cụm K-Medoids dựa trên các đặc trưng **'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'**.

Thuật toán	**Silhouette Score**	**Davies-Bouldin Score**	**Nhận xét mở rộng**
K-Medoids (Thư viện)	0.3544	1.0717	Mô hình cơ sở. Ổn định, ít nhạy với outliers do dùng medoid thay vì centroid. Tuy nhiên, score chưa tối ưu.
K-Means	0.4160	1.0803	**Silhouette cao nhất** → cụm tách biệt rõ. Davies-Bouldin tương đương với K-Medoids. Đây là mô hình có hiệu quả tốt nhất hiện tại.
Agglomerative Clustering	0.3206	1.2287	Hiệu quả thấp hơn cả về Silhouette và DB index. Có thể không phù hợp với dữ liệu này hoặc chưa chọn đúng linkage.
DBSCAN	0.1349	1.4746	Phát hiện được nhiễu nhưng cụm không rõ ràng, score rất thấp. Không phù hợp với dữ liệu này nếu không tinh chỉnh epsilon và min_samples.

Bảng 7: Bảng so sánh và nhận xét kết quả của K-Medoids và các thuật toán khác.

Nhận xét chung: Các kết quả thực nghiệm cho thấy mô hình K-Medoids đã hoạt động hiệu quả trong việc phân khúc khách hàng dựa trên các đặc trưng hành vi mua sắm đã được tiền xử lý. Việc lựa chọn số cụm tối ưu (k=3) và sử dụng kỹ thuật rút trích đặc trưng đã đóng

góp đáng kể vào chất lượng phân cụm. Khi so sánh với các thuật toán khác, K-Medoids thể hiện hiệu suất cạnh tranh, củng cố tính phù hợp của nó cho bài toán này.

4.1.1. Tiền xử lý dữ liệu và Rút trích đặc trưng

Kết quả thực nghiệm cho thấy tầm quan trọng của việc tiền xử lý dữ liệu để chuẩn hóa và biến đổi các đặc trưng 'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'. Việc áp dụng rút trích đặc trưng đã mang lại hiệu quả đáng kể.

- **So sánh hiệu quả phân cụm (có và không rút trích đặc trưng):**
    - **Không rút trích đặc trưng:** K-Medoids trên dữ liệu gốc đã được chuẩn hóa cho Silhouette Score là **0.2078** và Davies-Bouldin Score là **1.3283**.
    - **Có rút trích đặc trưng (PCA):** Sau khi rút trích đặc trưng bằng PCA, K-Medoids đạt Silhouette Score **0.3544** và Davies-Bouldin Score **1.0717**.

Biện luận: Sự cải thiện rõ rệt về cả hai chỉ số (Silhouette tăng, Davies-Bouldin giảm) cho thấy không chỉ giảm chiều dữ liệu mà còn giúp loại bỏ nhiễu và tăng cường khả năng phân tách cụm, tạo ra các cụm chặt chẽ hơn và ít chồng chéo hơn. Điều này chứng minh rằng việc rút trích đặc trưng là một bước thiết yếu để nâng cao chất lượng phân cụm trên bộ dữ liệu này.

4.1.2. Xác định số cụm tối ưu (k)

Thông qua việc sử dụng các phương pháp Elbow Method, Silhouette Score và Davies-Bouldin Score, nhóm đã xác định số cụm tối ưu cho bài toán phân khúc khách hàng là k=3.

- **Biểu đồ Elbow Method:** Cho thấy một "khuỷu tay" rõ ràng tại k=3, nơi sự giảm của tổng bình phương khoảng cách trong cụm (Inertia) bắt đầu chậm lại.
- **Biểu đồ Silhouette Score:** Đạt giá trị cao nhất tại k=3, cho thấy các cụm có độ kết dính cao và được phân tách tốt nhất ở số cụm này.
- **Biểu đồ Davies-Bouldin Score:** Đạt giá trị thấp nhất tại k=3, củng cố thêm rằng đây là số cụm mà các cụm được phân tách rõ ràng nhất và ít chồng chéo nhất.

4.1.3. Phân tích kết quả phân cụm K-Medoids

Với k=3 cụm và sử dụng các đặc trưng đã rút trích, thuật toán K-Medoids đã phân loại khách hàng thành ba phân khúc chính. Bằng cách xem xét giá trị trung bình của các đặc

trưng **'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'** cho mỗi cụm, chúng ta có thể hiểu rõ hơn về hành vi của từng nhóm khách hàng.

- **Cụm 0 (Khách hàng Chi tiêu thấp và Ít mua):**
  - **Monetary:** $244.31
  - **TotalQuantity:** $132.55
  - **Frequency:** $1.27
  - **UniqueItems:** $14.42
  - **Số lượng:** 1336 khách hàng
  - **Nhận xét:** Cụm này chiếm phần lớn khách hàng. Họ có mức chi tiêu (Monetary) rất thấp, mua số lượng sản phẩm (TotalQuantity) ít, tần suất mua sắm (Frequency) rất thấp (trung bình chỉ hơn 1 lần), và số lượng mặt hàng khác nhau (UniqueItems) cũng ít. Đây có thể là nhóm khách hàng mới, khách hàng chỉ mua một lần, hoặc khách hàng có sự gắn kết thấp với doanh nghiệp. Doanh nghiệp có thể tập trung vào các chiến dịch kích thích mua hàng lần hai hoặc tăng cường tần suất mua sắm cho nhóm này.

- **Cụm 1 (Khách hàng Giá trị cao và Thường xuyên):**
  - **Monetary:** $1471.49
  - **TotalQuantity:** $813.69
  - **Frequency:** $4.95
  - **UniqueItems:** $66.93
  - **Số lượng:** 614 khách hàng
  - **Nhận xét:** Đây là nhóm khách hàng có giá trị cao nhất và tần suất mua sắm cao nhất. Họ chi tiêu rất nhiều (Monetary cao nhất), mua số lượng sản phẩm lớn (TotalQuantity cao nhất), có tần suất mua sắm cao (Frequency gần 5 lần), và đa dạng các mặt hàng (UniqueItems cao nhất). Đây chính là nhóm khách hàng trung thành, mang lại doanh thu lớn cho doanh nghiệp. Cần có các chính sách chăm sóc đặc biệt, ưu đãi dành riêng và giữ chân nhóm khách hàng này.

- **Cụm 2 (Khách hàng Chi tiêu trung bình, ít thường xuyên):**

- **Monetary:** $648.97
- **TotalQuantity:** $360.52
- **Frequency:** $2.42
- **UniqueItems:** $42.53
- **Số lượng:** 1075 khách hàng
- **Nhận xét:** Cụm này đại diện cho nhóm khách hàng có mức chi tiêu và tần suất mua sắm ở mức trung bình, cao hơn Cụm 0 nhưng thấp hơn đáng kể so với Cụm 1. Họ mua một số lượng và đa dạng mặt hàng ở mức vừa phải. Đây có thể là những khách hàng có tiềm năng để trở thành khách hàng giá trị cao nếu được khuyến khích đúng cách, hoặc có thể là những khách hàng đã từng hoạt động nhưng đang dần ít tương tác hơn. Doanh nghiệp có thể áp dụng các chương trình khuyến mãi để tăng cường tần suất và giá trị chi tiêu của nhóm này.

4.1.4. So sánh giữa cài đặt thủ công và thư viện

Việc triển khai thuật toán K-Medoids một cách thủ công đã cung cấp một cái nhìn sâu sắc về cơ chế hoạt động bên trong thuật toán. Tuy nhiên, khi so sánh với việc sử dụng thư viện sklearn_extra.cluster.KMedoids, có những điểm khác biệt đáng chú ý:

- **K-Medoids thủ công:** Silhouette Score: **0.3750**, Davies-Bouldin Score: **1.1088**.
- **K-Medoids thư viện:** Silhouette Score: **0.3544**, Davies-Bouldin Score: **1.0717**.

Biện luận: Mặc dù trong lần chạy cụ thể này, cài đặt thủ công đạt được Silhouette Score cao hơn một chút, nhưng Davies-Bouldin Score lại kém hơn so với thư viện. Sự khác biệt này có thể do phương pháp khởi tạo ngẫu nhiên hoặc đường đi hội tụ khác nhau của thuật toán, dẫn đến việc tìm thấy các tối ưu cục bộ khác nhau. Tuy nhiên, xét về tổng thể, thư viện cung cấp một giải pháp ổn định hơn, đã được tối ưu hóa về hiệu suất và tính tin cậy. Việc sử dụng thư viện cũng giúp tiết kiệm đáng kể thời gian phát triển và bảo trì mã nguồn, cho phép tập trung vào việc diễn giải và áp dụng kết quả kinh doanh.

4.1.5. So sánh kết quả với các thuật toán phân cụm khác

Để đánh giá toàn diện hơn, K-Medoids đã được so sánh với K-Means, Agglomerative Clustering và DBSCAN trên cùng bộ dữ liệu đã rút trích đặc trưng.

Thuật toán	Silhouette Score	Davies-Bouldin Score
K-Medoids (Thư viện)	0.3544	**1.0717**
K-Means	**0.4160**	1.0803
Agglomerative Clustering	0.3206	1.2287
DBSCAN	0.1349	1.4746

Bảng 8: Bảng so sánh kết quả thuật toán.

Biện luận:

- **K-Means** đạt Silhouette Score cao nhất, cho thấy khả năng phân cụm hiệu quả trên dữ liệu có dạng cụm cầu.
- **K-Medoids** cho thấy hiệu suất rất cạnh tranh với K-Means, đặc biệt là về Davies-Bouldin Score (thấp nhất), cho thấy các cụm được phân tách tốt. K-Medoids cũng vượt trội về khả năng chống chịu outlier và tính diễn giải (medoid là điểm dữ liệu thực tế).
- **Agglomerative Clustering** cho hiệu suất thấp hơn rõ rệt so với K-Means và K-Medoids trên bộ dữ liệu này với k=3 cụm.
- **DBSCAN** tìm thấy rất nhiều cụm nhỏ và một lượng lớn điểm nhiễu, đồng thời có các chỉ số đánh giá rất thấp. Điều này cho thấy DBSCAN không phù hợp để tạo ra các phân khúc khách hàng lớn, rõ ràng trên bộ dữ liệu này với các tham số đã chọn.

Kết luận về lựa chọn thuật toán: Dựa trên các kết quả thực nghiệm, mặc dù K-Means cho Silhouette Score cao nhất, **K-Medoids vẫn được đánh giá là lựa chọn tối ưu** cho bài toán phân khúc khách hàng này. K-Medoids không chỉ mang lại hiệu suất chất lượng cụm rất tốt và cạnh tranh mà còn cung cấp khả năng diễn giải trực quan (medoid là khách hàng thực tế) và có tính bền vững hơn trước các giá trị ngoại lai, điều này rất quan trọng trong dữ liệu kinh doanh thực tế.

4.1.6. Điểm mạnh của mô hình

Dựa trên quá trình triển khai và kết quả thu được, mô hình phân cụm K-Medoids của chúng ta có những điểm mạnh nổi bật sau:

- **Phân khúc khách hàng rõ ràng và dễ diễn giải:** Mô hình đã thành công trong việc xác định ba phân khúc khách hàng riêng biệt. Các phân khúc này có thể được diễn giải rõ ràng dựa trên các đặc trưng hành vi mua sắm ('Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'), giúp doanh nghiệp dễ dàng xây dựng chiến lược marketing mục tiêu.
- **Tính bền vững với giá trị ngoại lai (Outlier Robustness):** K-Medoids ít nhạy cảm hơn với các điểm dữ liệu ngoại lai vì nó sử dụng các điểm dữ liệu thực tế (medoid) làm trung tâm cụm thay vì trung bình cộng. Điều này đặc biệt quan trọng trong dữ liệu kinh doanh có thể chứa các giao dịch bất thường.
- **Tăng cường hiệu suất nhờ rút trích đặc trưng (PCA):** Việc áp dụng PCA không chỉ giúp giảm chiều dữ liệu mà còn cải thiện đáng kể chất lượng phân cụm, chứng minh tầm quan trọng của tiền xử lý dữ liệu.
- **Điểm cụm là dữ liệu thực tế:** Các medoid là những khách hàng hiện có trong tập dữ liệu, giúp việc mô tả và hình dung "khách hàng tiêu biểu" của từng phân khúc trở nên trực quan và dễ hiểu hơn đối với những người không chuyên về kỹ thuật.
- **Hiệu suất cạnh tranh:** Mặc dù K-Means đạt Silhouette Score cao nhất trong so sánh, K-Medoids vẫn cho thấy hiệu suất rất cạnh tranh với Davies-Bouldin Score tốt hơn, khẳng định đây là một lựa chọn mạnh mẽ cho bài toán này.

4.1.7. Điểm yếu của mô hình

Bên cạnh những điểm mạnh, mô hình cũng bộc lộ một số hạn chế và điểm yếu:

- **Độ nhạy cảm với khởi tạo ban đầu:** Giống như K-Means, K-Medoids có thể nhạy cảm với việc lựa chọn các medoid ban đầu.
- **Yêu cầu xác định số cụm k trước:** Mô hình K-Medoids yêu cầu phải xác định trước số cụm (k).
- **Chi phí tính toán cao hơn K-Means:** Đối với các tập dữ liệu rất lớn, K-Medoids có thể tốn kém về mặt tính toán hơn K-Means.

- **Hạn chế với các cụm có hình dạng phức tạp:** K-Medoids giả định các cụm có hình dạng tương đối cầu. Nó có thể không hiệu quả trong việc phát hiện các cụm có hình dạng phức tạp hoặc không theo cấu trúc mật độ đều.
- **DBSCAN không hiệu quả trong trường hợp này:** DBSCAN đã cho thấy hiệu suất kém, cho thấy dữ liệu có thể không có các cụm dựa trên mật độ rõ ràng hoặc cần tinh chỉnh tham số rất kỹ lưỡng.

4.1.8. Các yếu tố ảnh hưởng đến hiệu quả mô hình

Một số yếu tố đã được xác định có ảnh hưởng đáng kể đến hiệu quả của mô hình phân cụm:

- **Chất lượng tiền xử lý dữ liệu:** Việc chuẩn hóa và biến đổi các đặc trưng là rất quan trọng để đưa chúng về cùng một thang đo và giảm thiểu ảnh hưởng của các giá trị cực đoan.
- **Rút trích đặc trưng (PCA):** PCA đóng vai trò then chốt trong việc cải thiện chất lượng cụm bằng cách loại bỏ nhiễu và tập trung vào các phương sai chính trong dữ liệu.
- **Lựa chọn số cụm tối ưu:** Việc xác định chính xác số cụm k=3 đã đảm bảo rằng các phân khúc được tạo ra có ý nghĩa và tối ưu về mặt thống kê.
- **Lựa chọn thuật toán:** Mỗi thuật toán phân cụm có những giả định và cách thức hoạt động riêng, ảnh hưởng trực tiếp đến kết quả.
- **Đặc trưng đầu vào:** Mô hình chỉ dựa trên các đặc trưng 'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'. Việc thiếu các đặc trưng khác (ví dụ: nhân khẩu học, sở thích sản phẩm, kênh tương tác) có thể hạn chế khả năng phân loại khách hàng một cách toàn diện hơn.

4.1.9. Kết luận chung

Mô hình K-Medoids đã thể hiện hiệu quả rõ rệt trong việc phân khúc khách hàng dựa trên dữ liệu 'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems', cung cấp cái nhìn sâu sắc về các nhóm khách hàng khác nhau và hành vi của họ. Các kết quả thực nghiệm chứng minh rằng việc tiền xử lý dữ liệu cẩn thận, đặc biệt là rút trích đặc trưng bằng PCA, là yếu tố then chốt để đạt được chất lượng phân cụm cao.

Mặc dù mô hình K-Medoids có những điểm yếu nhất định, nhưng những điểm mạnh về tính bền vững với outlier, khả năng diễn giải trực quan và hiệu suất cạnh tranh đã làm cho nó trở thành một lựa chọn phù hợp và giá trị cho bài toán phân khúc khách hàng này.

Trong tương lai, để cải thiện hơn nữa độ chính xác và tính toàn diện của các phân khúc, có thể xem xét:

- Bổ sung thêm các đặc trưng khác ngoài 'Monetary', 'TotalQuantity', 'Frequency', 'UniqueItems'.
- Thử nghiệm các kỹ thuật phân cụm nâng cao hơn hoặc kết hợp.
- Thực hiện phân tích độ nhạy cảm đối với các tham số khởi tạo để đảm bảo tính ổn định của kết quả.

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Đề tài đã hoàn thành xuất sắc mục tiêu đề ra là xây dựng một mô hình phân khúc khách hàng hiệu quả dựa trên dữ liệu hành vi mua sắm, sử dụng thuật toán phân cụm K-Medoids. Những kết quả nổi bật bao gồm:

- **Xác định các phân khúc khách hàng rõ ràng:** Mô hình đã thành công trong việc phân chia tập khách hàng thành **ba cụm riêng biệt** và có ý nghĩa kinh doanh sâu sắc. Mỗi cụm được đặc trưng bởi các hành vi mua sắm khác nhau về chi tiêu (Monetary), tổng số lượng sản phẩm mua (TotalQuantity), tần suất mua hàng (Frequency) và số lượng mặt hàng độc đáo (UniqueItems). Các phân khúc này bao gồm:
    - **Cụm khách hàng Chi tiêu thấp và Ít mua:** Nhóm khách hàng lớn nhất, có mức chi tiêu và tần suất mua sắm rất hạn chế.
    - **Cụm khách hàng Giá trị cao và Thường xuyên:** Nhóm khách hàng cốt lõi, chi tiêu cao nhất và mua sắm thường xuyên nhất, mang lại doanh thu chính.
    - **Cụm khách hàng Chi tiêu trung bình, có tiềm năng:** Nhóm khách hàng ở giữa, có thể được phát triển thành khách hàng giá trị cao hơn thông qua các chiến lược phù hợp.
- **Xác định phương pháp tối ưu:** Nghiên cứu đã chỉ ra rằng việc kết hợp **tiền xử lý dữ liệu cẩn thận, rút trích đặc trưng bằng PCA và thuật toán K-Medoids** (với k=3 cụm tối ưu) mang lại hiệu quả phân cụm vượt trội. PCA không chỉ giảm chiều dữ liệu mà còn cải thiện đáng kể chất lượng cụm.
- **Đánh giá và so sánh mô hình:** Mô hình K-Medoids đã được đánh giá thông qua các chỉ số Silhouette Score và Davies-Bouldin Score, cho thấy hiệu suất cạnh tranh mạnh mẽ so với các thuật toán phân cụm khác như K-Means, Agglomerative Clustering và DBSCAN. Đặc biệt, K-Medoids nổi bật với khả năng chống chịu outlier và tính diễn giải trực quan (medoid là khách hàng thực tế).
- **Cơ sở cho chiến lược kinh doanh:** Các phân khúc khách hàng được xác định là cơ sở vững chắc để doanh nghiệp phát triển các chiến lược marketing mục tiêu, cá nhân

hóa trải nghiệm khách hàng, và tối ưu hóa các chương trình khuyến mãi, chăm sóc khách hàng.

5.2. Những khó khăn, hạn chế

Trong quá trình thực hiện đề tài, nhóm cũng đã đối mặt với một số khó khăn và nhận thấy các hạn chế cần được cải thiện trong các nghiên cứu tương lai:

- **Phụ thuộc vào việc xác định số cụm k:** Giống như nhiều thuật toán phân cụm khác, K-Medoids yêu cầu phải xác định trước số cụm tối ưu. Mặc dù đã sử dụng các phương pháp như Elbow và Silhouette để hỗ trợ, việc lựa chọn này vẫn có thể mang tính chủ quan ở một mức độ nhất định.
- **Tính nhạy cảm với khởi tạo ban đầu:** Kết quả của K-Medoids có thể bị ảnh hưởng bởi việc lựa chọn các medoid ban đầu. Dù đã sử dụng init='k-medoids++' để cải thiện, nhưng khả năng đạt được tối ưu toàn cục vẫn là một thách thức.
- **Chi phí tính toán:** Đối với các tập dữ liệu cực lớn, K-Medoids có thể có chi phí tính toán cao hơn so với K-Means do quá trình tìm kiếm medoid liên quan đến việc tính toán khoảng cách giữa tất cả các cặp điểm.
- **Khả năng diễn giải dữ liệu:** Một thách thức không nhỏ trong quá trình thực hiện là việc ban đầu chưa có được các giá trị cụ thể của từng cụm (như giá trị trung bình của Monetary, Frequency, v.v.) từ kết quả của mô hình. Điều này đã gây khó khăn trong việc diễn giải ý nghĩa kinh doanh của từng phân khúc một cách chi tiết và cụ thể. Việc khắc phục đã đòi hỏi nhiều lần tương tác và điều chỉnh để thu thập đủ thông tin.
- **Hạn chế của dữ liệu đầu vào:** Mô hình chỉ dựa trên các đặc trưng hành vi mua sắm. Việc thiếu các đặc trưng khác như thông tin nhân khẩu học (độ tuổi, giới tính), sở thích sản phẩm cụ thể, hoặc kênh tương tác có thể hạn chế khả năng phân loại khách hàng một cách toàn diện và sâu sắc hơn.

Giả định hình dạng cụm: K-Medoids, tương tự K-Means, hoạt động hiệu quả nhất với các cụm có hình dạng tương đối cầu. Nó có thể gặp khó khăn với các cụm có hình dạng phức tạp, không đều hoặc chồng chéo.

TÀI LIỆU THAM KHẢO

[[1] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc, 2019.

[2] G. Bonaccorso, *Machine Learning Algorithms*. Packt Publishing Ltd., 2017.

[3] D. Sarkar, R. Bali, and T. Sharma, *Practical Machine Learning with Python*. Apress Media, LLC, 2018.

[4] V. H. Tiệp, *Machine Learning cơ bản*, 2018

Bảng phân công công việc của các thành viên trong nhóm

Họ và tên	MSSV	Viết báo cáo	Tìm hiểu nội dung	Code và xây dựng model	Ý thức làm việc
Nguyễn Trần Bảo Long	22DH112007	100%	100%	100%	100%
Lương Đức Khoa	22DH111647	70%	100%	100%	100%
Nguyễn Hoàng Bảo	22DH110271	100%	100%	100%	100%
Nguyễn Hữu Bình	21DH113496	100%	100%	70%	100%