BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

BÁO CÁO MÔN: HỌC MÁY

ĐỀ TÀI: BỘ DỮ LIỆU HÀNH VI MUA SẮM CỦA KHÁCH HÀNG

KHÓA: 65 CNTT

Sinh viên thực hiện: **Võ Trường An**

Lê Thị Linh

Trần Minh Quân

GIẢNG VIÊN HƯỚNG DẪN: Th.S Vũ Thị Hạnh

TP Hồ Chí Minh 23/12/2025

LỜI MỞ ĐẦU

Lời đầu tiên, em xin chân thành cảm ơn Phân hiệu Trường Đại học Thủy Lợi, đặc biệt là cô Vũ Thị Hạnh thuộc bộ môn Công nghệ thông tin của trường. Cô đã dẫn dắt nhóm 2 chúng em gồm Võ Trường An, Lê Thị Linh và Trần Minh Quân làm đồ án này, đồng thời cô cũng trang bị những kiến thức cơ bản của Machine Learning (Học máy) để có những hiểu biết về AI (Artificial Intelligent) trong thời đại 4.0 thời đại mà AI phát triển rất mạnh mẽ và là một nghành học rất phổ biến. Trong suốt quá trình thực hiện đồ án này, nhóm 2 chúng em đã áp dụng các kiến thức đã học như các model, tiền xử lý dữ liệu, … kết hợp với tìm hiểu thêm các nội dung mới, từ đó tận dụng tối đa để hoàn thiện báo cáo đồ án này.

Mặc dù vậy, quá trình thực hiện đồ án vẫn có những hạn chế nhất định. Vì vậy, em mong muốn nhận được những lời đánh giá đúng nhất từ cô và để hoàn thiện hơn về kiến thức, bên cạnh đó chuẩn bị tốt hơn cho các đồ án khác trong tương lai!

Nhóm 2 xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Lý do chọn đề tài

- Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của thương mại điện tử và các hệ thống bán lẻ hiện đại, dữ liệu về hành vi mua sắm của khách hàng ngày càng trở nên phong phú và có giá trị cao. Việc khai thác và phân tích hiệu quả nguồn dữ liệu này giúp doanh nghiệp hiểu rõ hơn về nhu cầu, sở thích và thói quen tiêu dùng của khách hàng, từ đó nâng cao hiệu quả kinh doanh và khả năng cạnh tranh trên thị trường.

- Hành vi mua sắm của khách hàng chịu ảnh hưởng bởi nhiều yếu tố khác nhau như mức chi tiêu, tần suất mua hàng, loại sản phẩm và bối cảnh mua sắm. Nếu được phân tích đúng cách, các dữ liệu này có thể hỗ trợ doanh nghiệp trong việc **phân khúc khách hàng**, xây dựng chiến lược marketing phù hợp cho từng nhóm khách hàng và tối ưu hóa trải nghiệm người dùng.

- Dataset **Customer Shopping Behavior** cung cấp các thông tin liên quan trực tiếp đến hành vi mua sắm của khách hàng, rất phù hợp để áp dụng các kỹ thuật Machine Learning nhằm phân tích, phân cụm và dự đoán hành vi tiêu dùng. Do đó, nhóm chúng em lựa chọn đề tài này để nghiên cứu và ứng dụng các mô hình học máy vào bài toán thực tế trong lĩnh vực kinh doanh và bán lẻ.

1.2. Mục tiêu đề tài

Mục tiêu của đề tài bao gồm:

- Tìm hiểu và phân tích dữ liệu hành vi mua sắm của khách hàng.
- Thực hiện tiền xử lý dữ liệu để đảm bảo chất lượng dữ liệu đầu vào cho mô hình.
- Ứng dụng các thuật toán **Machine Learning không giám sát** để:
  - Phân cụm khách hàng dựa trên hành vi mua sắm.
  - Khám phá các nhóm khách hàng có đặc điểm tiêu dùng tương đồng.
  - Khám phá đặc điểm tiêu dùng đặc trưng của từng nhóm khách hàng.
- **Đánh giá và so sánh hiệu quả** các thuật toán phân cụm được sử dụng.
- **Phân tích ý nghĩa kinh doanh** từ các cụm khách hàng thu được.
- **Rèn luyện kỹ năng thực hành, nghiên cứu và làm việc nhóm**.

1.3. Bài toán đặt ra

- Từ dataset **Customer Shopping Behavior**, bài toán đặt ra:

  - Sử dụng các thuật toán **Machine Learning không giám sát** để phân cụm khách hàng dựa trên hành vi mua sắm, nhằm xác định các nhóm khách hàng có đặc điểm tiêu dùng tương đồng và phân tích sự khác biệt giữa các nhóm này.

- Các bước thực hiện chi tiết:

  1. **Lựa chọn thuộc tính:** Chọn các cột dữ liệu phản ánh rõ hành vi tiêu dùng (tuổi, giới tính, thu nhập, tần suất mua, loại sản phẩm…).

2. Tiền xử lý dữ liệu:
    - Xử lý giá trị thiếu, nhiễu.
    - Mã hóa biến phân loại sang dạng số.
    - Chuẩn hóa các thuộc tính số để đồng nhất thang đo.
3. Áp dụng các thuật toán phân cụm:
    - **K-Means Clustering:** phân cụm dựa trên khoảng cách Euclidean, hiệu quả với dữ liệu có dạng hình cầu.Mã hóa biến phân loại sang dạng số.
    - **DBSCAN:** phân cụm dựa trên mật độ, phát hiện các cụm có hình dạng bất thường và outliers.
    - **Hierarchical Clustering:** phân cụm theo cấu trúc cây phân cấp, quan sát trực quan mối quan hệ giữa các cụm.

1.4. Phạm vi đề tài

- **Dữ liệu sử dụng:** tập trung vào các thông tin cơ bản về khách hàng và hành vi tiêu dùng, gồm:
    - Độ tuổi và giới tính
    - Thu nhập và mức chi tiêu
    - Tần suất mua hàng
    - Loại sản phẩm được mua
- **Kỹ thuật áp dụng:** các thuật toán **phân cụm Machine Learning không giám sát**: K-Means, DBSCAN, Hierarchical Clustering.
- **Giới hạn:** không sử dụng học sâu, dữ liệu không phản ánh hành vi theo thời gian thực, chưa xem xét dữ liệu phi cấu trúc như đánh giá, bình luận.

1.5. Giới hạn của đề tài

- Mặc dù đề tài mang tính ứng dụng cao, nhưng vẫn tồn tại một số giới hạn nhất định:
    - Đề tài chỉ sử dụng một tập dữ liệu cố định, chưa kết hợp thêm các nguồn dữ liệu khác.
    - Chưa áp dụng các mô hình học sâu (Deep Learning) hay các phương pháp nâng cao.
    - Dữ liệu chưa phản ánh hành vi mua sắm theo thời gian thực.
    - Chưa xem xét các yếu tố phi cấu trúc như đánh giá, bình luận của khách hàng.

1.6. Công cụ hỗ trợ đề tài

- Ngôn ngữ lập trình: Python.
- **Môi trường cài đặt:** Google Colab và Kaggle.

Google Colab được sử dụng để chạy và kiểm tra chương trình nhanh chóng thông qua file Jupyter Notebook (.ipynb). Kaggle được sử dụng chủ yếu trong quá trình huấn luyện và thử nghiệm mô hình, do hỗ trợ tài nguyên phần cứng ổn định và cho phép chương trình tiếp tục chạy ngay cả khi tắt máy tính, điều mà Google Colab còn hạn chế.

CHƯƠNG 2: Mô tả dữ liệu sử dụng và Tiền xử lý

2.1: Mô tả dữ liệu sử dụng:

- Trong đề tài này, nhóm sử dụng **Customer Shopping Behavior Dataset**, là tập dữ liệu mô tả hành vi mua sắm của khách hàng trong lĩnh vực bán lẻ. Dataset cung cấp các thông tin cơ bản về đặc điểm cá nhân và thói quen tiêu dùng của khách hàng, phù hợp cho việc phân tích và áp dụng các mô hình **Machine Learning**.

- Các thuộc tính chính trong tập dữ liệu bao gồm:

  - **Customer ID** – Mã định danh khách hàng
  - **Age** – Tuổi của khách hàng
  - **Gender** – Giới tính
  - **Annual Income** – Thu nhập hàng năm
  - **Purchase Amount** – Số tiền mua hàng
  - **Purchase Frequency** – Tần suất mua hàng
  - **Preferred Payment Method** – Phương thức thanh toán được ưa thích
  - **Product Category** – Loại sản phẩm được mua
  - **Review / Feedback Rating** – Xếp hạng đánh giá sau mua
  - **Shipping Type** – Kiểu giao hàng

2.2 Tiền xử lý dữ liệu:

- Trước khi đưa dữ liệu vào mô hình Machine Learning, nhóm tiến hành các bước tiền xử lý sau để đảm bảo dữ liệu sạch, thống nhất và có thể học được các mẫu tiềm ẩn:

2.2.1. Kiểm tra và xử lý giá trị thiếu:

- Nhóm kiểm tra toàn bộ các cột để xác định sự tồn tại của giá trị thiếu (missing values).
- Các dòng chứa giá trị thiếu được xử lý bằng:

  - **Thuộc tính số**: thay bằng giá trị trung bình (mean).
  - **Thuộc tính phân loại**: thay bằng giá trị xuất hiện nhiều nhất (mode).

- Nếu số lượng giá trị thiếu quá nhiều và không thể khắc phục hợp lý, dòng dữ liệu đó sẽ được loại bỏ.

2.2.2. Mã hóa biến phân loại:

- Các biến dạng chữ (categorical) như Gender, Profession, Product Category, Ever Married, Graduated được chuyển sang dạng số để mô hình Machine Learning hiểu được.
- Các phương pháp sử dụng:

  - **Label Encoding** cho biến 2 giá trị (ví dụ: Gender — Nam / Nữ)
  - **One-Hot Encoding** cho biến nhiều giá trị (ví dụ: Profession, Product_Category)

2.2.3. Chuẩn hóa dữ liệu (Scaling)

- Các biến số như Age, Annual Income, Work Experience, Family Size, Spending Score được chuẩn hóa để:
  - Tránh sự chênh lệch về thang đo giữa các thuộc tính.
  - Giúp mô hình học nhanh hơn và cải thiện hiệu quả phân cụm.
- Các phương pháp chuẩn hóa được sử dụng gồm:
  - **StandardScaler** (chuẩn hóa về trung bình 0 và độ lệch chuẩn 1).
  - **MinMaxScaler** (đưa dữ liệu về khoảng [0,1]).

2.2.4. Chia tập dữ liệu

- Sau khi hoàn tất quá trình tiền xử lý, dữ liệu được chia thành:
  - **Tập huấn luyện (Training set): 80%**
  - **Tập kiểm tra (Test set): 20%**
- Việc chia tập dữ liệu theo tỷ lệ này giúp mô hình được huấn luyện và đánh giá một cách khách quan, hạn chế hiện tượng overfitting.

2.3. Công cụ hỗ trợ xử lý dữ liệu

- **Google Colab**: được sử dụng để chạy và kiểm tra chương trình nhanh chóng qua **Jupyter Notebook (.ipynb)**, thuận tiện cho việc thử nghiệm các bước tiền xử lý và trực quan hóa dữ liệu.
- **Visual Studio Code:** được sử dụng để tổ chức và quản lý mã nguồn theo pipeline chuẩn (tiền xử lý -> phân tích dữ liệu -> xây dựng mô hình -> đánh giá), giúp mã nguồn rõ ràng, dễ bảo trì và chia thành các file tách biệt.
- **Kaggle**: được sử dụng để huấn luyện và tối ưu mô hình, nhờ **tài nguyên phần cứng ổn định** và khả năng **chạy liên tục ngay cả khi tắt máy tính**, khắc phục hạn chế của Google Colab.

CHƯƠNG 3: CÁC MÔ HÌNH HỌC MÁY SỬ DỤNG

3.1. Tổng quan

- Trong nghiên cứu này, các thuật toán học máy **không giám sát (Unsupervised Learning)** được sử dụng nhằm phân cụm khách hàng dựa trên hành vi mua sắm. Cụ thể, ba mô hình được áp dụng và so sánh bao gồm:
  - K-Means Clustering
  - DBSCAN
  - Hierarchical Clustering (Agglomerative)
- Trong đó, **K-Means** được lựa chọn làm mô hình chính do cho kết quả phân cụm rõ ràng và phù hợp với bài toán phân tích hành vi khách hàng.

3.2. K-Means Clustering (Mô hình chính)

3.2.1. Nguyên lý hoạt động

- K-Means là thuật toán học máy không giám sát, phân cụm dữ liệu dựa trên khoảng cách Euclidean. Thuật toán chia tập dữ liệu thành K cụm sao cho tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm là nhỏ nhất (Within-Cluster Sum of Squares – WCSS).
- Quy trình hoạt động gồm các bước:
  - Chọn K tâm cụm (centroids) ban đầu.
  - Gán mỗi điểm dữ liệu vào cụm có tâm gần nhất.
  - Cập nhật vị trí tâm cụm bằng trung bình các điểm thuộc cụm đó.
  - Lặp lại bước 2-3 cho đến khi tâm cụm ổn định hoặc đạt số vòng lặp tối đa.

3.2.2. Lý do lựa chọn:

- K-Means được lựa chọn vì:
  - Dữ liệu không có nhãn → phù hợp với **unsupervised learning.**
  - Các biến đầu vào (Purchase Amount, Frequency of Purchases) là liên tục và có ý nghĩa khoảng cách
  - Thuật toán đơn giản, dễ triển khai, hiệu quả với tập dữ liệu lớn
  - Kết quả dễ diễn giải → phù hợp cho **phân tích hành vi khách hàng (Customer Segmentation)**
  - Được sử dụng phổ biến trong các bài toán phân cụm khách hàng trong thực tế

- Ngoài ra, K-Means cho phép đánh giá chất lượng phân cụm thông qua các chỉ số như **Elbow Method** và **Silhouette Score**, giúp lựa chọn số cụm phù hợp.

3.2.3. Huấn luyện mô hình (Train model):

- Trong nghiên cứu này, hai đặc trưng được sử dụng để huấn luyện mô hình bao gồm:
  - **Purchase Amount (USD)**: mức chi tiêu của khách hàng
  - **Frequency of Purchases** (đã được chuyển đổi sang dạng số – Frequency_Num)
- Trước khi huấn luyện mô hình, dữ liệu được chuẩn hóa nhằm đưa các biến về cùng thang đo, tránh hiện tượng một biến chi phối khoảng cách trong thuật toán K-Means.

```
#Xác định tập đặc trưng đầu vào
X = df[['Purchase Amount (USD)', 'Frequency_Num']]
```

Hình 1: Tập đặc trưng đầu vào (Input Feature Set)

```
#Chuẩn hóa dữ liệu
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Hình 2: Tập đặc trưng đã được chuẩn hóa

- Sau đó, mô hình K-Means được huấn luyện với số cụm $k = 3$. Phương thức fit_predict() được sử dụng để vừa huấn luyện mô hình vừa gán nhãn cụm cho từng quan sát. Nhãn cụm này được gán trực tiếp vào cột Cluster trong DataFrame.

```
# Huấn luyện mô hình K-Means và gán nhãn cụm
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
```

Hình 3: Huấn luyện mô hình K-Means và gán nhãn cụm

- Trong đó:
  - n_clusters = 3: số cụm được lựa chọn dựa trên Elbow Method và Silhouette Score
  - random_state = 42: đảm bảo kết quả có thể tái lập

3.2.4. Hyperparameter Tuning

3.2.4.1 Xác định số cụm (n_clusters):

- Số cụm tối ưu được xác định bằng **Elbow Method**, thông qua việc tính toán WCSS với các giá trị *k* khác nhau.

```
#Xác định số cụm
wcss = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(7,5))
plt.plot(range(1, 11), wcss, marker='o')
plt.xlabel("Số cụm (k)")
plt.ylabel("WCSS")
plt.title("Elbow Method xác định số cụm")
plt.show()
```

Hình 4: Biểu đồ Elbow Method xác định số cụm tối ưu cho mô hình K-Means

- Kết quả cho thấy tại **k = 3**, đường cong WCSS bắt đầu giảm chậm lại, tạo thành điểm "khuỷu tay" trên đồ thị Elbow. Điều này cho thấy việc tăng thêm số cụm không mang lại cải thiện đáng kể về chất lượng phân cụm. Do đó, **k = 3** được lựa chọn làm số cụm tối ưu cho mô hình K-Means.

3.2.4.2. Đánh giá bằng Silhouette Score

- Chất lượng phân cụm được đánh giá bằng **Silhouette Score**, phản ánh mức độ chặt chẽ trong cụm và sự tách biệt giữa các cụm.

```
#Đánh giá
from sklearn.metrics import silhouette_score, davies_bouldin_score

sil_score = silhouette_score(X_scaled, df['Cluster'])
db_index = davies_bouldin_score(X_scaled, df['Cluster'])

print("Silhouette Score:", sil_score)
print("Davies-Bouldin Index:", db_index)
```

Hình 5: Đánh giá chất lượng mô hình K-Means

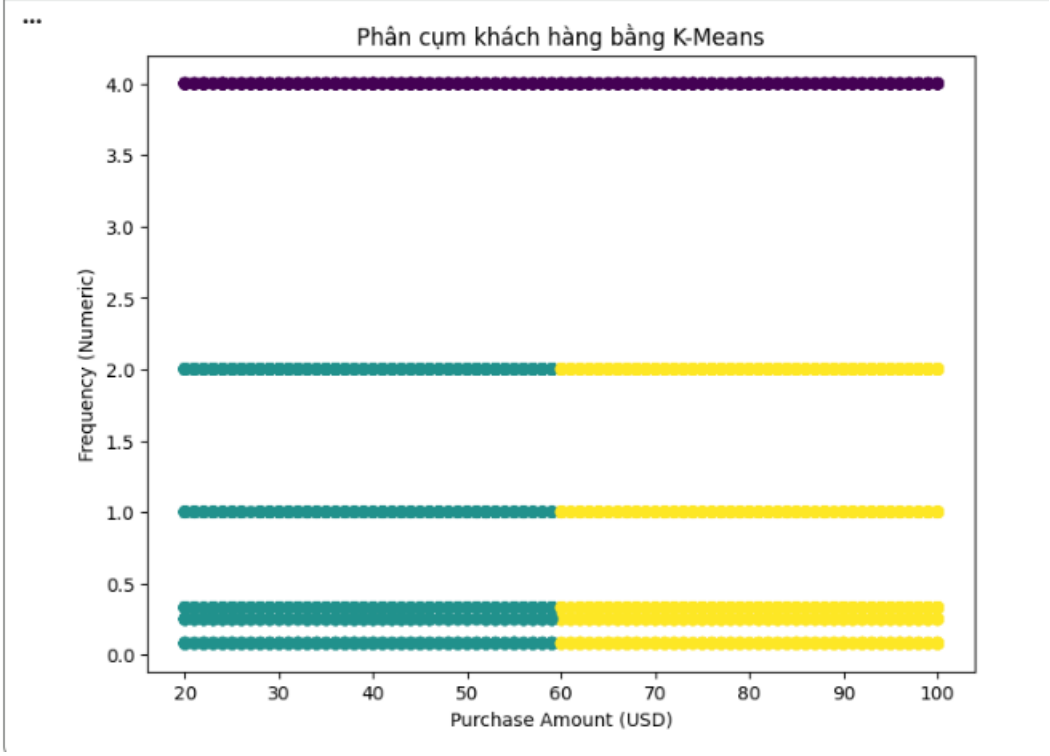
- Sau khi xác định số cụm, chất lượng phân cụm được đánh giá bằng **Silhouette Score**. Chỉ số này phản ánh mức độ chặt chẽ của các điểm dữ liệu trong cùng một cụm cũng như mức độ tách biệt giữa các cụm khác nhau.
- Silhouette Score được tính toán dựa trên nhãn cụm thu được từ mô hình K-Means với k = 3. Ngoài ra, chỉ số này cũng được so sánh với các giá trị k khác nhau nhằm đảm bảo rằng lựa chọn số cụm là hợp lý và mang lại chất lượng phân cụm tốt.

➢ Các tham số khác

- Ngoài tham số n_clusters, các tham số còn lại của mô hình K-Means được giữ ở giá trị mặc định do đã được tối ưu sẵn trong thư viện scikit-learn, bao gồm:
  - init = 'k-means++': phương pháp khởi tạo tâm cụm giúp cải thiện tốc độ hội tụ
  - max_iter = 300: số vòng lặp tối đa cho mỗi lần huấn luyện.
  - n_init = 10: số lần chạy thuật toán với các tâm khởi tạo khác nhau nhằm tránh rơi vào cực trị cục bộ

3.2.5. Trực quan hóa kết quả K-Means

```
# TRỰC QUAN HÓA KẾT QUẢ K-MEANS (Quan sát sự tách biệt giữa các cụm)
plt.figure(figsize=(8,6))
plt.scatter(
    df['Purchase Amount (USD)'],
    df['Frequency_Num'],
    c=df['Cluster']
)
plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Frequency (Numeric)")
plt.title("Phân cụm khách hàng bằng K-Means")
plt.show()
```



Hình 6: Trực quan hóa kết quả phân cụm khách hàng bằng thuật toán K-Means

- Hình trên thể hiện kết quả phân cụm bằng thuật toán K-Means dựa trên 2 đặc trưng chính là mức chi tiêu (Purchase Amount) và tần suất mua hàng (Frequency_Num).
- Các điểm dữ liệu được tô màu theo nhãn cụm cho thấy dự phân tách tương đối rõ ràng giữa các nhóm khách hàng, đặc biệt là sự khác biệt về mức chi tiêu.
- Cho thấy mô hình K-Means có khả năng phân biệt các nhóm khách hàng dựa trên hành vi mua sắm.

3.3. Hierarchical Clustering

3.3.1. Nguyên lý hoạt động

- **Hierarchical Clustering (Agglomerative)** là một thuật toán học máy **không giám sát**, thực hiện phân cụm dữ liệu theo phương pháp **từ dưới lên (bottom-up)**.
- Ban đầu, mỗi điểm dữ liệu được xem như **một cụm riêng biệt**. Sau đó, thuật toán lần lượt **gộp hai cụm gần nhau nhất** dựa trên một tiêu chí khoảng cách cho đến khi đạt được số cụm mong muốn hoặc toàn bộ dữ liệu hợp thành một cụm duy nhất.
- Các bước hoạt động chính:
  - Mỗi điểm dữ liệu là một cụm độc lập
  - Gộp hai cụm có khoảng cách nhỏ nhất
  - Lặp lại quá trình cho đến khi đạt số cụm cần thiết
- Khoảng cách giữa các cụm có thể được xác định thông qua các phương pháp liên kết (linkage), phổ biến nhất là:

3.3.2. Lý do lựa chọn mô hình

- Hierarchical Clustering được lựa chọn trong đề tài vì các lý do sau:
  - Không cần xác định trước số cụm ngay từ đầu
  - Cung cấp biểu diễn trực quan thông qua dendrogram, giúp quan sát mối quan hệ giữa các nhóm khách hàngGiúp phân tích mối quan hệ gần – xa giữa các nhóm khách hàng
  - Phù hợp để so sánh và đối chiếu kết quả với mô hình K-Means
  - Dễ giải thích trong quá trình phân tích hành vi khách hàng
- Mô hình này đặc biệt hữu ích trong giai đoạn **phân tích dữ liệu khám phá** và hỗ trợ quyết định số cụm hợp lý.

3.3.3. Huấn luyện mô hình

3.3.3.1. Xây dựng Dendrogram

- Dữ liệu đầu vào sử dụng hai đặc trưng
  - Purchase Amount (USD)
  - Purchase Amount (USD)
- Dữ liệu được chuẩn hóa trước khi áp dụng thuật toán.

```
# Phân cụm phân cấp (Hierarchical Clustering)
# Vẽ Dendrogram – biểu đồ cây của phương pháp phân cụm phân cấp
linked = linkage(X_scaled, method='ward')

plt.figure(figsize=(10,6))
dendrogram(linked)
plt.xlabel("Khách hàng")
plt.ylabel("Khoảng cách")
plt.title("Dendrogram - Hierarchical Clustering")
plt.show()
```

Hình 7: Dendrogram thể hiện cấu trúc phân cấp của dữ liệu khách hàng

3.3.3.2. Phân cụm bằng Agglomerative Clustering

- Dựa trên dendrogram, số cụm được lựa chọn là **3 cụm**, tương ứng với kết quả của K-Means.

```
#Xây dựng và áp dụng mô hình Phân cụm phân cấp
hc = AgglomerativeClustering(
    n_clusters=3,
    linkage='ward'
)

df['Cluster_HC'] = hc.fit_predict(X_scaled)
```

Hình 8: Kết quả phân cụm khách hàng bằng Hierarchical Clustering

3.3.4. Hyperparameter tuning

3.3.4.1. Xây dựng Dendrogram

- Tham số linkage
  - ward: tối thiểu hóa phương sai trong cụm (được lựa chọn)
  - single: dễ bị nhiễu
  - single: dễ bị nhiễu
  - average: trung gian giữa single và complete
- Ward linkage cho kết quả cụm rõ ràng và ổn định nhất trong nghiên cứu này.

3.3.4.2. Kết quả phân cụm khách hàng bằng Hierarchical Clustering

- Giá trị n_clusters = 3 được xác định bằng cách:
  - Quan sát điểm cắt hợp lý trên dendrogram
  - So sánh với kết quả K-Means
  - Đánh giá bằng Silhouette Score

```
#Đánh giá mô hình Phân cụm phân cấp bằng Silhouette Score
sil_hc = silhouette_score(X_scaled, df['Cluster_HC'])
print("Silhouette Score Hierarchical:", sil_hc)
```

Hình 9: Silhouette Score của mô hình Hierarchical Clustering

3.4. DBSCAN (Mô hình so sánh)

3.4.1. Nguyên lý hoạt động

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một **thuật toán phân cụm không giám sát dựa trên mật độ**, trong đó các điểm dữ liệu được gom thành cụm dựa vào mức độ "dày đặc" của chúng trong không gian đặc trưng.
- Khác với K-Means và Hierarchical Clustering, DBSCAN **không phân cụm dựa trên khoảng cách đến tâm cụm**, mà dựa trên **mật độ phân bố của dữ liệu**.
- Ý tưởng chính của DBSCAN là:
  - Các điểm nằm gần nhau trong một vùng có mật độ cao sẽ tạo thành một cụm
  - Các điểm nằm rải rác, không thuộc vùng mật độ cao sẽ được xem là **nhiễu (outliers)**

- Thuật toán dựa trên hai tham số chính:
  - **eps**: bán kính lân cận quanh mỗi điểm
  - **min_samples**: số điểm tối thiểu trong bán kính eps để hình thành một cụm
- Quy trình hoạt động:
  - Chọn một điểm bất kỳ
  - Xác định các điểm nằm trong bán kính eps
  - Nếu số điểm ≥ min_samples → hình thành cụm
  - Mở rộng cụm cho đến khi không thể mở rộng thêm
  - Lặp lại cho đến khi duyệt hết dữ liệu

3.4.2. Lý do lựa chọn mô hình

- DBSCAN được lựa chọn trong đề tài vì:
  - Không cần xác định trước số cụm
  - Phát hiện được **outliers**, điều mà K-Means và Hierarchical Clustering không làm tốt
  - Phù hợp với dữ liệu khách hàng có hành vi mua sắm **không đồng đều**
  - Giúp phát hiện các nhóm khách hàng đặc biệt (mua rất ít hoặc rất nhiều)
- Mô hình DBSCAN được sử dụng nhằm
  - So sánh với K-Means
  - Đánh giá khả năng nhận diện khách hàng bất thường

3.4.3. DBSCAN- Mô hình so sánh (So sánh/ thử nghiệm/ kiểm tra nhiễu)

3.4.3.1. Tập đặc trưng đầu vào

- Hai đặc trưng được sử dụng:
  - **Purchase Amount (USD)**: mức chi tiêu của khách hàng
  - **Frequency of Purchases (Frequency_Num)**: tần suất mua hàng (đã mã hóa số)

```
#Xác định tập đặc trưng đầu vào
X = df[['Purchase Amount (USD)', 'Frequency_Num']]
```

Hình 10: Tập đặc trưng đầu vào cho mô hình phân cụm

3.4.3.2. Chuẩn hóa dữ liệu

Do DBSCAN dựa trên khoảng cách, dữ liệu cần được chuẩn hóa trước khi huấn luyện.

```
#Chuẩn hóa dữ liệu
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Hình 11: Tập dữ liệu sau khi chuẩn hóa

3.4.3.3.Huấn luyện mô hình DBSCAN

- Dữ liệu đầu vào gồm 2 đặc trưng:
  - **Purchase Amount (USD)**
  - **Frequency_Num**
- Trước khi huấn luyện, dữ liệu được **chuẩn hóa bằng StandardScaler** để đảm bảo các biến có cùng thang đo.
- Trước khi huấn luyện, dữ liệu được **chuẩn hóa bằng StandardScaler** để đảm bảo các biến có cùng thang đo.

```
# ================================
#  PHÂN CỤM BẰNG DBSCAN
# ================================

dbscan = DBSCAN(
    eps=0.7,         # Khoảng cách tối đa để tạo cụm
    min_samples=5    # Số điểm tối thiểu tạo thành 1 cụm
)
```

Hình 12: Phân bố khách hàng theo chi tiêu và tần suất mua bằng DBSCAN

➢ Sau khi huấn luyện:

- Mỗi khách hàng được gán một nhãn cụm
- Các điểm có nhãn -1 được xem là **nhiễu (outliers)**

3.4.3.4. Trực quan hóa kết quả phân cụm

```
# phân cụm khách hàng bằng DBSCAN
plt.figure(figsize=(8,6))
plt.scatter(
    df['Purchase Amount (USD)'],
    df['Frequency_Num'],
    c=df['Cluster_DBSCAN'],
    cmap='viridis'
)

plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Frequency (converted to numbers)")
plt.title("Phân cụm khách hàng bằng DBSCAN")
plt.show()
```

Hình 13: Kết quả phân cụm khách hàng bằng DBSCAN

3.4.4. Hyperparameter Tuning

3.4.4.1. Lựa chọn tham số eps và min_samples

➢ Trong nghiên cứu này:

- min_samples = 5 được lựa chọn theo giá trị thường dùng trong thực tế
- eps = 0.5 được lựa chọn dựa trên **thực nghiệm và quan sát kết quả phân cụm**

➢ Các giá trị này được đánh giá lại thông qua:

- Số lượng cụm tạo thành
- Số lượng điểm nhiễu (label = -1)
- Mức độ phân tách giữa các cụm

3.4.4.2. Đánh giá bằng Silhouette Score

Chất lượng phân cụm được đánh giá bằng **Silhouette Score**, trong trường hợp mô hình tạo ra nhiều hơn một cụm và không toàn bộ là nhiễu

```
labels = df['Cluster_DBSCAN']

if len(set(labels)) > 1 and -1 not in set(labels):
    sil_db = silhouette_score(X_scaled, labels)
    print("Silhouette Score DBSCAN:", sil_db)
else:
    print("DBSCAN không tính được Silhouette vì có nhiễu hoặc chỉ có 1 cụm")
```

Hình 14: Đánh giá chất lượng mô hình DBSCAN bằng Silhouette Score

➢ Silhouette Score phản ánh:

- Mức độ chặt chẽ trong từng cụm
- Mức độ tách biệt giữa các cụm khác nhau

3.4.5. Trực quan hóa kết quả phân cụm

➢ Kết quả DBSCAN được trực quan hóa bằng biểu đồ scatter:

```
# phân cụm khách hàng bằng DBSCAN
plt.figure(figsize=(8,6))
plt.scatter(
    df['Purchase Amount (USD)'],
    df['Frequency_Num'],
    c=df['Cluster_DBSCAN'],
    cmap='viridis'
)

plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Frequency (converted to numbers)")
plt.title("Phân cụm khách hàng bằng DBSCAN")
plt.show()
```

Hình 15: Biểu đồ phân tán thể hiện kết quả phân cụm khách hàng bằng thuật toán DBSCAN

➢ Biểu đồ giúp:

- Quan sát cấu trúc cụm
- Nhận diện rõ các điểm nhiễu (outliers)
- So sánh trực quan với K-Means và Hierarchical Clustering

CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

4.1. Nguyên tắc đánh giá trong bài toán phân cụm

- Do bài toán phân cụm **không có nhãn thật**, nên:
- Không sử dụng: Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC-AUC
- Sử dụng:

  - **Silhouette Score**
  - **Biểu đồ trực quan hóa cụm**
  - **Phân tích đặc trưng trung bình của từng cụm**
  - **So sánh kết quả giữa các mô hình**

4.2. Đánh giá mô hình K-Means

4.2.1. Đánh giá bằng Silhouette Score

```
#Silhouette
from sklearn.metrics import silhouette_score

sil_score = silhouette_score(X_scaled, df['Cluster'])
print("Silhouette Score:", sil_score)
```

```
Silhouette Score: 0.4672981695096318
```

Hình 16: Silhouette Score đánh giá chất lượng phân cụm K-Means

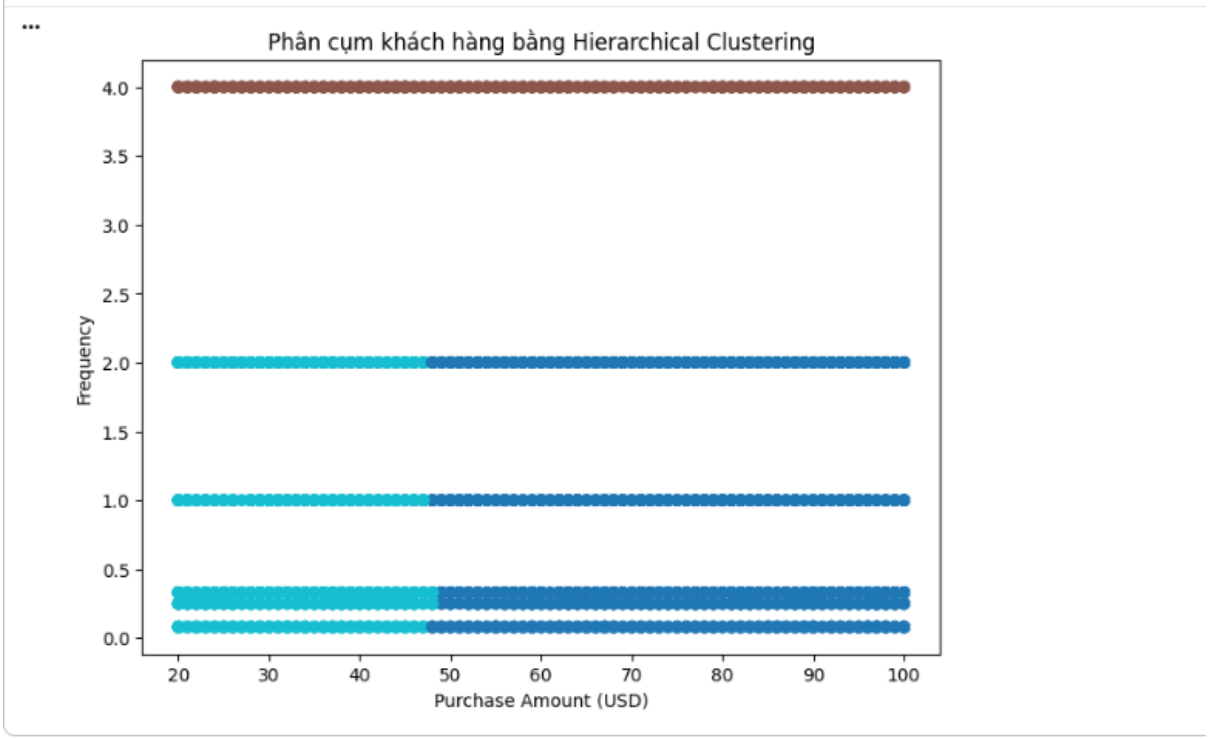
- **Giải thích:**

  - Silhouette Score ∈ [-1, 1]
  - Giá trị càng gần **1** → cụm càng rõ ràng
  - Kết quả thu được cho thấy mô hình K-Means với **k = 3** cho chất lượng phân cụm **tốt và ổn định**

4.2.2. Trực quan hóa kết quả phân cụm K-Means

```
# Trực quan hóa kết quả Phân cụm phân cấp
plt.figure(figsize=(8,6))
plt.scatter(
    df['Purchase Amount (USD)'],
    df['Frequency_Num'],
    c=df['Cluster_HC'],
    cmap='tab10'
)

plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Frequency")
plt.title("Phân cụm khách hàng bằng Hierarchical Clustering")
plt.show()
```



Hình 17: Trực quan hóa kết quả phân cụm khách hàng bằng K-Means

➢ **Phân tích hình:**

- Các cụm khách hàng được tách biệt tương đối rõ ràng
- Mỗi cụm đại diện cho một nhóm hành vi mua sắm khác nhau:
  - Chi tiêu cao – mua thường xuyên
  - Chi tiêu trung bình
  - Chi tiêu thấp – mua ít

4.2.3. Phân tích đặc trưng trung bình của các cụm

```
#Thống kê hành vi trung bình mỗi cụm
cluster_summary = df.groupby('Cluster')[[
    'Purchase Amount (USD)',
    'Frequency_Num'
]].mean()

cluster_summary
```

Cluster	Purchase Amount (USD)	Frequency_Num
0	58.972171	4.000000
1	38.940217	0.919432
2	80.240469	0.931173

Hình 18: Thống kê trung bình theo cụm khách hàng

➢ **Ý nghĩa:**

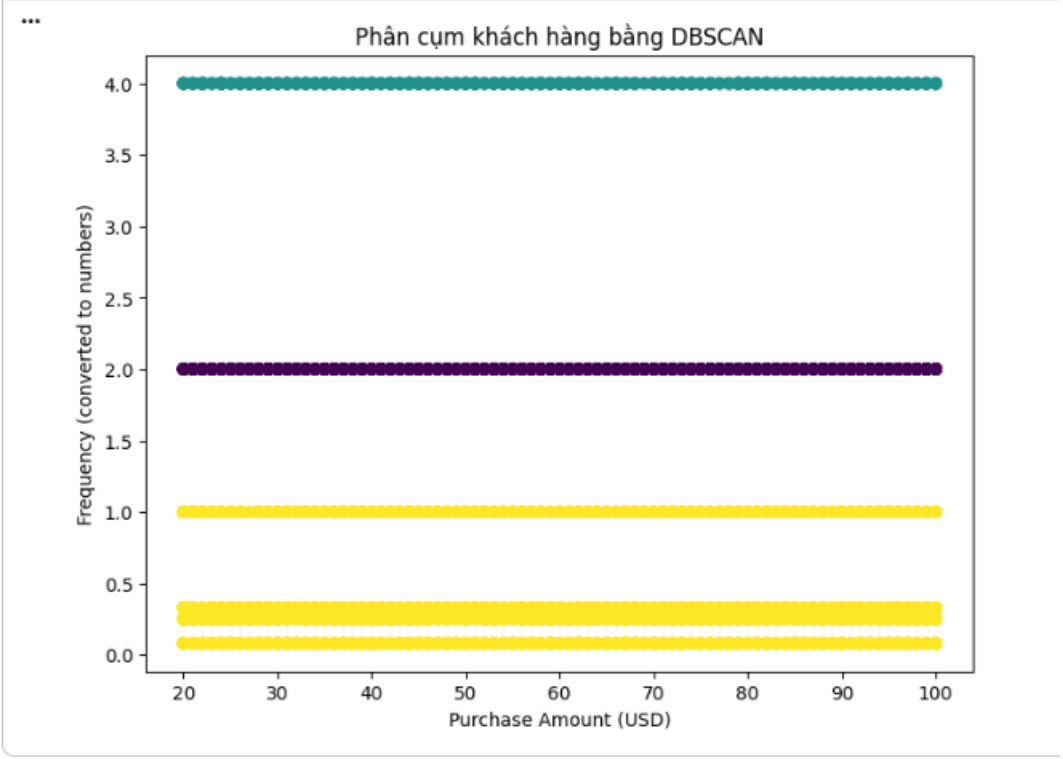
- Cho phép hiểu rõ **đặc trưng hành vi** của từng nhóm khách hàng
- Hỗ trợ doanh nghiệp xây dựng **chiến lược marketing phù hợp cho từng cụm**

4.3. Đánh giá mô hình DBSCAN

4.3.1. Kết quả phân cụm DBSCAN

```
# phân cụm khách hàng bằng DBSCAN
plt.figure(figsize=(8,6))
plt.scatter(
    df['Purchase Amount (USD)'],
    df['Frequency_Num'],
    c=df['Cluster_DBSCAN'],
    cmap='viridis'
)

plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Frequency (converted to numbers)")
plt.title("Phân cụm khách hàng bằng DBSCAN")
plt.show()
```



Hình 19: Kết quả phân cụm khách hàng bằng DBSCAN

➢ **Phân tích:**

- DBSCAN phát hiện được:
  - Các cụm mật độ cao
  - Các điểm nhiễu (nhãn -1)
- Phù hợp để phát hiện **khách hàng bất thường**

4.3.2. Đánh giá bằng Silhouette Score

```
labels = df['Cluster_DBSCAN']

if len(set(labels)) > 1 and -1 not in set(labels):
    sil_db = silhouette_score(X_scaled, labels)
    print("Silhouette Score DBSCAN:", sil_db)
else:
    print("DBSCAN không tính được Silhouette vì có nhiễu hoặc chỉ có 1 cụm")
```

```
Silhouette Score DBSCAN: 0.338393604221279
```

Hình 20: Đánh giá chất lượng mô hình DBSCAN

➢ **Giải thích khi vấn đáp:**

- DBSCAN **không luôn tính được Silhouette**
- Vì:
  - Có **nhiễu (-1)**
  - Hoặc chỉ tạo **1 cụm**

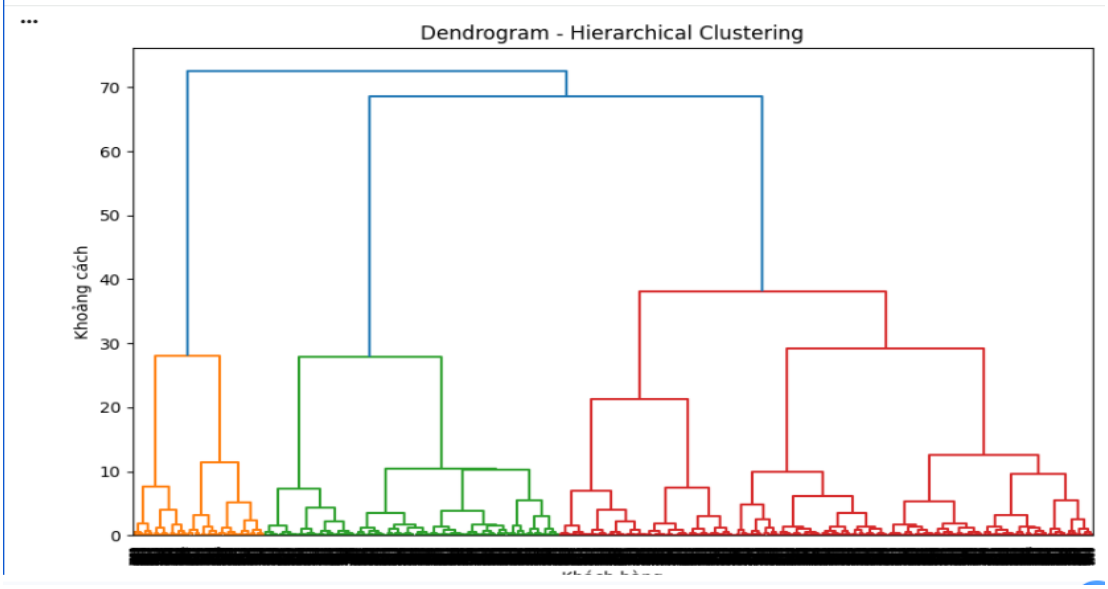
=>Đó là lý do cần **kiểm tra điều kiện trước khi đánh giá**

4.4. Đánh giá mô hình Hierarchical Clustering

4.4.1. Dendrogram – đánh giá cấu trúc cụm

```
# Phân cụm phân cấp (Hierarchical Clustering)
# Vẽ Dendrogram – biểu đồ cây của phương pháp phân cụm phân cấp
linked = linkage(X_scaled, method='ward')

plt.figure(figsize=(10,6))
dendrogram(linked)
plt.xlabel("Khách hàng")
plt.ylabel("Khoảng cách")
plt.title("Dendrogram - Hierarchical Clustering")
plt.show()
```



Hình 21: Dendrogram thể hiện cấu trúc phân cấp của dữ liệu khách hàng

➢ **Ý nghĩa:**

- Quan sát mối quan hệ gần – xa giữa các điểm dữ liệu
- Hỗ trợ lựa chọn **số cụm phù hợp**

4.4.2. Silhouette Score Hierarchica

```
#Đánh giá mô hình Phân cụm phân cấp bằng Silhouette Score
sil_hc = silhouette_score(X_scaled, df['Cluster_HC'])
print("Silhouette Score Hierarchical:", sil_hc)
```

```
Silhouette Score Hierarchical: 0.4396882415102732
```

Hình 22: Đánh giá chất lượng mô hình Hierarchical Clustering bằng Silhouette Score

- **Nhận xét:**
  - Silhouette Score cho thấy mô hình Hierarchical với **n_clusters = 3** cho kết quả hợp lý
  - Tuy nhiên, độ rõ ràng cụm thấp hơn K-Means

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

- Trong đề tài này, nhóm đã thực hiện phân tích hành vi mua sắm của khách hàng dựa trên tập dữ liệu **Customer Shopping Behavior** bằng các thuật toán học máy không giám sát. Thông qua quá trình tiền xử lý dữ liệu, lựa chọn đặc trưng và chuẩn hóa dữ liệu, nhóm đã xây dựng và đánh giá ba mô hình phân cụm gồm **K-Means, DBSCAN và Hierarchical Clustering**.
- Kết quả thực nghiệm cho thấy:

    - **K-Means Clustering** cho kết quả phân cụm rõ ràng, dễ diễn giải và đạt **Silhouette Score cao nhất**, phù hợp cho bài toán phân khúc khách hàng.
    - **DBSCAN chỉ là mô hình so sánh** có khả năng phát hiện các điểm nhiễu và các khách hàng có hành vi mua sắm bất thường, tuy nhiên nhạy cảm với việc lựa chọn tham số.
    - **Hierarchical Clustering** cung cấp cái nhìn trực quan thông qua dendrogram, hỗ trợ phân tích mối quan hệ giữa các nhóm khách hàng, nhưng chi phí tính toán cao hơn.

- Việc so sánh các mô hình cho thấy **K-Means là mô hình phù hợp nhất** để làm mô hình chính trong nghiên cứu này, đáp ứng tốt mục tiêu phân cụm khách hàng dựa trên mức chi tiêu và tần suất mua sắm.
- Việc so sánh các mô hình cho thấy **K-Means là mô hình phù hợp nhất** để làm mô hình chính trong nghiên cứu này, đáp ứng tốt mục tiêu phân cụm khách hàng dựa trên mức chi tiêu và tần suất mua sắm.

    - Xây dựng chiến lược marketing phù hợp cho từng nhóm khách hàng
    - Cá nhân hóa trải nghiệm mua sắm
    - Nâng cao hiệu quả kinh doanh và chăm sóc khách hàng

6.2. Hướng phát triển

- Mặc dù đề tài đã đạt được các mục tiêu đề ra, vẫn còn một số hướng phát triển trong tương lai như sau:
- **Mở rộng tập đặc trưng**

    - Bổ sung thêm các thuộc tính như: đánh giá sản phẩm, phương thức thanh toán, loại sản phẩm, thời gian mua sắm
    - Kết hợp dữ liệu theo thời gian để phân tích xu hướng hành vi mua sắm

- **Áp dụng các thuật toán nâng cao hơn**

    - Thử nghiệm các thuật toán phân cụm khác như **Gaussian Mixture Model (GMM)**, **Spectral Clustering**
    - So sánh hiệu quả với các mô hình hiện tại

- **Tối ưu hyperparameter tự động**

- Áp dụng Grid Search hoặc các phương pháp tối ưu hóa tự động để lựa chọn tham số tốt hơn cho DBSCAN và Hierarchical Clustering

➢ **Ứng dụng thực tế**

- Triển khai mô hình vào hệ thống gợi ý sản phẩm
- Ứng dụng trong hệ thống CRM để phân nhóm khách hàng và hỗ trợ ra quyết định kinh doanh

➢ **Kết hợp học có giám sát**

- Sử dụng kết quả phân cụm làm nhãn giả (pseudo-labels) cho các mô hình học có giám sát nhằm dự đoán hành vi mua sắm trong tương lai

CHƯƠNG 6: TÀI LIỆU THAM KHẢO

https://www.kaggle.com/datasets/ayeshasiddiqa123/customer-shopping-behavior-dataset