TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

Sinh viên: Nguyễn Duy Quảng

Lớp: K57KMT.01

MSSV: K215480106139

Giáo viên giảng dạy: TS. Nguyễn Văn Huy

Link GitHub: https://github.com/nd-wuangr26/DataScience.git

**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên*: Nguyễn Duy Quảng

*Lớp*: K57KMT.01 *Ngành:* Kỹ thuật máy tính

*Giáo viên hướng dẫn:* TS. Nguyễn Văn Huy

*Ngày giao đề:* 20/05/2025 *Ngày hoàn thành:* 25/05/2025

*Tên đề tài :* Phân loại khách hàng tiềm năng

*Yêu cầu :*

* Ứng dụng web phân loại khách hàng dựa trên dữ liệu giao dịch.
* Dữ liệu đầu vào: <https://www.kaggle.com/datasets/shwetabh123/mall-customers>
* Dữ liệu đầu ra: Phân loại khách hàng và biểu đồ phân nhóm khách hàng.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | **GIÁO VIÊN HƯỚNG DẪN** |  | |  | *(Ký và ghi rõ họ tên)* |  | |  |
|  |  |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |  |
| --- | --- |
|  | Thái Nguyên, ngày….tháng…..năm 20....  GIÁO VIÊN HƯỚNG DẪN |
|  |  |
|  | *(Ký ghi rõ họ tên)* |

MỤC LỤC

[LỜI NÓI ĐẦU 8](#_Toc199312854)

[LỜI CẢM ƠN 9](#_Toc199312855)

[CHƯƠNG I: GIỚI THIỆU MÔN HỌC VÀ ĐỀ TÀI 10](#_Toc199312856)

[1.1. Giới thiệu môn học 10](#_Toc199312857)

[1.2. Đặt vấn đề 10](#_Toc199312858)

[1.3. Hướng giải quyết 11](#_Toc199312859)

[1.4. Yêu cầu đầu vào và đầu ra 12](#_Toc199312860)

[1.5. Các tính năng chính của chương trình 12](#_Toc199312861)

[1.6. Thách thức của bài toán 13](#_Toc199312862)

[1.7. Kiến thức và công nghệ vận dụng 13](#_Toc199312863)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 15](#_Toc199312864)

[2.1. Ngôn ngữ lập trình Python 15](#_Toc199312865)

[2.1.1. Đặc điểm của Python 15](#_Toc199312866)

[2.1.2. Ứng dụng của Python 15](#_Toc199312867)

[2.2. Danh sách (List) trong Python 15](#_Toc199312868)

[2.2.1. Đặc điểm của List 16](#_Toc199312869)

[2.2.2. Các thao tác cơ bản với List 16](#_Toc199312870)

[2.3. Thư viện Pandas 16](#_Toc199312871)

[2.3.1. Cấu trúc dữ liệu chính 17](#_Toc199312872)

[2.3.2. Chức năng chính của Pandas 17](#_Toc199312873)

[2.3.3. Ví dụ sử dụng cơ bản 17](#_Toc199312874)

[2.4. Thư viện Matplotlib và Seaborn 18](#_Toc199312875)

[2.4.1. Matplotlib 18](#_Toc199312876)

[2.4.2. Seaborn 18](#_Toc199312877)

[2.4.3. Ví dụ sử dụng 18](#_Toc199312878)

[2.5. Mô hình học máy không giám sát – K-Means 19](#_Toc199312879)

[2.5.1. Khái niệm về học máy không giám sát 19](#_Toc199312880)

[2.5.2. Thuật toán K-Means 19](#_Toc199312881)

[2.5.3. Ưu điểm và nhược điểm 19](#_Toc199312882)

[2.5.4. Ví dụ sử dụng K-Means 20](#_Toc199312883)

[2.6. Phương pháp Elbow 20](#_Toc199312884)

[2.6.1. Vấn đề xác định số cụm tối ưu 20](#_Toc199312885)

[2.6.2. Nguyên lý phương pháp Elbow 21](#_Toc199312886)

[2.6.3. Cách xác định điểm Elbow 21](#_Toc199312887)

[2.6.4. Ví dụ implementation 21](#_Toc199312888)

[2.7. Streamlit – Thư viện tạo ứng dụng web 22](#_Toc199312889)

[2.7.1. Giới thiệu về Streamlit 22](#_Toc199312890)

[2.7.2. Đặc điểm nổi bật của Streamlit 22](#_Toc199312891)

[2.7.3. Các thành phần cơ bản 23](#_Toc199312892)

[2.7.4. Cấu trúc ứng dụng Streamlit cơ bản 23](#_Toc199312893)

[2.7.5. Deployment và chia sẻ 24](#_Toc199312894)

[2.7.6. Ưu điểm và hạn chế 24](#_Toc199312895)

[CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH 26](#_Toc199312896)

[3.1. Sơ đồ hệ thống 26](#_Toc199312897)

[3.1.1. Module Nhập Dữ Liệu 26](#_Toc199312898)

[3.1.2. Module Tiền Xử Lý 26](#_Toc199312899)

[3.1.3. Module Phân Cụm (Clustering) 27](#_Toc199312900)

[3.1.4. Module Trực Quan Hóa 27](#_Toc199312901)

[3.1.5. Module Giao Diện Web 27](#_Toc199312902)

[3.2. Sơ đồ khối các thuật toán chính 27](#_Toc199312903)

[3.3. Cấu trúc dữ liệu 28](#_Toc199312904)

[3.4. Chương trình 29](#_Toc199312905)

[CHƯƠNG IV: THỰC NGHIỆM VÀ KẾT LUẬN 30](#_Toc199312906)

[4.1. Thực nghiệm 30](#_Toc199312907)

[4.1.1. Kiểm thử chức năng 30](#_Toc199312908)

[4.1.1.1. Chức năng tải dữ liệu: 30](#_Toc199312909)

[4.1.1. Hình ảnh mô tả các tính năng 31](#_Toc199312910)

[4.2. Kết luận 35](#_Toc199312911)

[4.2.1. Những kết quả đã đạt được 35](#_Toc199312912)

[4.2.2. Kiến thức và kinh nghiệm thu nhận 35](#_Toc199312913)

[4.2.3. Hướng phát triển trong tương lai 35](#_Toc199312914)

[CHƯƠNG V: KẾT LUẬN 37](#_Toc199312915)

LỜI NÓI ĐẦU

Trong thời đại dữ liệu bùng nổ, việc khai thác và phân tích dữ liệu đã trở thành một yêu cầu thiết yếu đối với mọi tổ chức và doanh nghiệp. Đặc biệt, trong lĩnh vực kinh doanh và marketing, việc hiểu rõ tập khách hàng không chỉ giúp tối ưu hóa chiến lược chăm sóc khách hàng, mà còn là cơ sở để cá nhân hóa sản phẩm, nâng cao doanh thu và duy trì lợi thế cạnh tranh.

Bài toán **phân loại khách hàng tiềm năng** xuất phát từ nhu cầu thực tế: **doanh nghiệp cần chia tập khách hàng hiện tại thành các nhóm đặc trưng khác nhau** dựa trên hành vi tiêu dùng và đặc điểm cá nhân. Thay vì áp dụng các phương pháp phân tích thủ công hay cảm tính, việc ứng dụng các thuật toán học máy (machine learning) giúp doanh nghiệp tự động hóa quá trình phân nhóm và đưa ra các quyết định chính xác hơn.

Trong khuôn khổ môn học **Khoa học dữ liệu (Data Science)**, bài tập này giúp sinh viên tiếp cận quy trình thực tế của một dự án khoa học dữ liệu, từ khâu tiền xử lý, khám phá dữ liệu, áp dụng thuật toán phân cụm, cho đến việc trực quan hóa và giải thích kết quả. Dữ liệu được sử dụng trong bài là bộ dữ liệu **Mall Customer Segmentation** – gồm các đặc trưng như độ tuổi, thu nhập hàng năm và điểm chi tiêu của khách hàng trong một trung tâm thương mại.

Em rất mong được sự quan tâm và đóng góp ý kiến của thầy cùng các bạn để em có thể hoàn thiện bài của mình một cách chính xác và hoàn thiện nhất. Lời góp ý của thầy cùng các bạn là sự bổ sung cần thiết để bài của em thực tế hơn. Em xin chân thành cảm ơn.

LỜI CẢM ƠN

Trong quá trình học tập, em đã thấy được tầm quan trọng của việc làm việc cá nhân, cách làm những bài tập lớn hay cách để nói lên ý hiểu của mình đều nhờ cô Th.S Trần Thị Thanh trong bộ môn Công nghệ thông tin - Khoa Điện Tử - Trường đại học Kỹ thuật Công nghiệp - Đại học Thái Nguyên. Cùng với sự đóng góp ý kiến của các bạn trong lớn cũng tạo nên sự hoàn thiện bài tập lớn này. Xin trân thành cảm ơn sự giúp đỡ tận tâm, tận tình sâu sắc của cô và sự giúp đỡ cùng với những ý kiến của các bạn trong lớp.

Em xin trân thành cảm ơn!

|  |  |
| --- | --- |
|  | **Sinh viên thực hiện** |
|  |  |
|  | **Nguyễn Duy Quảng** |

CHƯƠNG I: GIỚI THIỆU MÔN HỌC VÀ ĐỀ TÀI

## Giới thiệu môn học

Trong thời đại công nghệ phát triển mạnh mẽ hiện nay, dữ liệu đã trở thành một nguồn tài nguyên có giá trị đặc biệt, đóng vai trò then chốt trong hầu hết các lĩnh vực từ kinh doanh, y tế, giáo dục, đến quản lý nhà nước. Môn học Khoa học Dữ liệu (Data Science) ra đời nhằm trang bị cho sinh viên những kiến thức và kỹ năng nền tảng để khai thác hiệu quả nguồn dữ liệu đang ngày càng phát triển. Đây là một môn học liên ngành, kết hợp giữa thống kê, khoa học máy tính, trí tuệ nhân tạo và tư duy phân tích, giúp người học có thể thu thập, xử lý, phân tích và trực quan hóa dữ liệu để đưa ra các quyết định dựa trên bằng chứng.

Môn học này hướng đến việc giúp sinh viên hiểu rõ toàn bộ quy trình thực hiện một dự án khoa học dữ liệu, bao gồm các bước như: thu thập và tiền xử lý dữ liệu, phân tích dữ liệu khám phá (EDA), xây dựng và huấn luyện các mô hình học máy (machine learning), trực quan hóa kết quả và trình bày giải pháp. Thông qua đó, sinh viên không chỉ nắm bắt được lý thuyết mà còn có cơ hội thực hành trên các công cụ hiện đại như Python, Pandas, Scikit-learn, Matplotlib và Streamlit – những công cụ phổ biến trong giới chuyên môn.

Hơn thế nữa, môn học còn giúp người học hình thành tư duy dữ liệu – một trong những kỹ năng cốt lõi trong thời đại số. Sinh viên có thể áp dụng những gì đã học vào nhiều lĩnh vực thực tiễn như marketing, tài chính, chăm sóc sức khỏe, giáo dục, và nhiều lĩnh vực khác. Với nhu cầu ngày càng cao về nhân lực có khả năng phân tích dữ liệu, môn học Khoa học Dữ liệu mang lại giá trị thiết thực cả về mặt học thuật lẫn nghề nghiệp, đóng vai trò như một bước đệm quan trọng để sinh viên phát triển trong kỷ nguyên số.

## Đặt vấn đề

Trong bối cảnh kinh doanh hiện đại, việc thấu hiểu và phân loại khách hàng ngày càng trở nên quan trọng đối với các doanh nghiệp, đặc biệt là trong lĩnh vực bán lẻ và thương mại dịch vụ. Việc xác định nhóm khách hàng tiềm năng giúp doanh nghiệp tối ưu hóa chiến lược marketing, nâng cao chất lượng dịch vụ, từ đó gia tăng doanh thu và giữ chân khách hàng hiệu quả hơn. Tuy nhiên, với số lượng khách hàng lớn và hành vi tiêu dùng ngày càng đa dạng, việc phân tích thủ công không còn phù hợp và thiếu tính chính xác. Đó chính là lý do vì sao các phương pháp phân tích dữ liệu hiện đại như phân cụm (clustering) được ứng dụng mạnh mẽ.

Bài toán được đặt ra là: Làm thế nào để phân nhóm khách hàng một cách tự động, khách quan và hiệu quả dựa trên dữ liệu giao dịch thực tế? Trong khuôn khổ môn học Khoa học Dữ liệu, nhóm chúng em thực hiện bài toán phân loại khách hàng sử dụng bộ dữ liệu Mall Customer Segmentation, bao gồm các thuộc tính như độ tuổi, thu nhập hàng năm và điểm chi tiêu. Bằng cách áp dụng thuật toán KMeans, một phương pháp phân cụm không giám sát phổ biến, nhóm hướng tới việc xác định các nhóm khách hàng có đặc điểm tương đồng. Kết quả phân cụm sẽ được trực quan hóa và trình bày thông qua ứng dụng web xây dựng bằng Streamlit, giúp người dùng dễ dàng hiểu và khai thác dữ liệu hơn. Bài toán không chỉ giúp sinh viên tiếp cận quy trình phân tích dữ liệu thực tế mà còn cho thấy rõ ứng dụng thiết thực của khoa học dữ liệu trong kinh doanh.

## Hướng giải quyết

Để giải quyết bài toán phân loại khách hàng tiềm năng, nhóm lựa chọn áp dụng quy trình chuẩn của một dự án khoa học dữ liệu, kết hợp giữa phân tích thống kê và học máy không giám sát. Trước tiên, dữ liệu từ tập Mall Customer Segmentation sẽ được tiền xử lý, bao gồm kiểm tra dữ liệu thiếu, chuẩn hóa dữ liệu và lựa chọn các đặc trưng quan trọng như độ tuổi, thu nhập và điểm chi tiêu. Tiếp theo, nhóm sử dụng thuật toán KMeans để thực hiện phân cụm khách hàng thành các nhóm có hành vi tiêu dùng tương đồng. Để xác định số cụm tối ưu, nhóm áp dụng phương pháp Elbow Method thông qua việc đánh giá chỉ số nội tại (inertia) của mô hình. Sau khi phân cụm thành công, kết quả sẽ được trực quan hóa bằng biểu đồ 2D hoặc 3D để dễ dàng phân tích, đồng thời xây dựng một ứng dụng web bằng Streamlit để trình bày kết quả phân cụm một cách sinh động và thân thiện với người dùng. Ứng dụng này cho phép người dùng xem biểu đồ phân nhóm, tải dữ liệu đầu vào và quan sát đặc trưng của từng cụm khách hàng. Phương pháp tiếp cận này không chỉ đảm bảo tính khách quan trong quá trình phân tích, mà còn giúp doanh nghiệp dễ dàng áp dụng vào thực tiễn nhằm nâng cao hiệu quả chiến lược chăm sóc khách hàng.

## Yêu cầu đầu vào và đầu ra

Hệ thống phân loại khách hàng được thiết kế với đầu vào là bộ dữ liệu giao dịch có cấu trúc rõ ràng, cụ thể là tập **Mall Customer Segmentation Data**, bao gồm các thuộc tính như: mã khách hàng (CustomerID), giới tính (Gender), tuổi (Age), thu nhập hàng năm (Annual Income (k$)), và điểm chi tiêu (Spending Score (1-100)). Dữ liệu đầu vào cần đảm bảo tính đầy đủ, không có giá trị thiếu, và được chuẩn hóa trước khi đưa vào phân tích.

Đầu ra của chương trình là:

* Danh sách khách hàng được phân nhóm theo các cụm cụ thể, phản ánh các hành vi tiêu dùng tương đồng.
* Biểu đồ trực quan hóa cụm khách hàng (2D hoặc 3D) thể hiện rõ ràng các phân nhóm.
* Biểu đồ Elbow để xác định số cụm tối ưu.
* Giao diện người dùng thân thiện giúp tải dữ liệu, tùy chỉnh số cụm và hiển thị kết quả phân tích theo thời gian thực.

## Các tính năng chính của chương trình

Chương trình được xây dựng với các tính năng nổi bật như:

* **Tải và xem trước dữ liệu:** Cho phép người dùng tải lên tệp dữ liệu CSV và xem trước các thông tin cơ bản.
* **Tiền xử lý tự động:** Tự động làm sạch dữ liệu, loại bỏ dữ liệu thiếu và chuẩn hóa các giá trị số.
* **Lựa chọn số cụm phân nhóm:** Cho phép người dùng chọn số cụm hoặc để chương trình gợi ý thông qua biểu đồ Elbow.
* **Thực hiện phân cụm bằng thuật toán KMeans:** Phân tích dữ liệu để xác định các nhóm khách hàng có đặc điểm tương đồng.
* **Trực quan hóa dữ liệu:** Biểu đồ phân cụm sinh động giúp người dùng dễ dàng hiểu và so sánh các nhóm.
* **Xem thông tin cụ thể của từng cụm:** Cho phép hiển thị chi tiết đặc điểm các khách hàng trong mỗi nhóm.
* **Giao diện web dễ sử dụng:** Được xây dựng bằng Streamlit, đảm bảo khả năng tương tác tốt và thân thiện với người dùng.

## Thách thức của bài toán

Mặc dù bài toán phân loại khách hàng có vẻ đơn giản ở bề mặt, nhưng trong quá trình thực hiện, nhóm gặp phải một số thách thức điển hình như:

* **Tiền xử lý dữ liệu phức tạp:** Dữ liệu thực tế có thể chứa lỗi, thiếu giá trị hoặc định dạng không đồng nhất, đòi hỏi phải có quy trình xử lý nghiêm ngặt.
* **Lựa chọn đặc trưng phù hợp:** Không phải tất cả các trường trong dữ liệu đều hữu ích, việc chọn đúng đặc trưng ảnh hưởng lớn đến chất lượng phân cụm.
* **Xác định số cụm tối ưu:** Không có công thức cố định, đòi hỏi phải kết hợp nhiều phương pháp đánh giá và kinh nghiệm để xác định.
* **Giải thích cụm:** Sau khi phân cụm, việc đặt tên và mô tả đặc trưng mỗi cụm khách hàng một cách trực quan, có ý nghĩa cũng là một thách thức đáng kể.
* **Tối ưu trải nghiệm người dùng:** Việc xây dựng giao diện trực quan và mượt mà, đồng thời đảm bảo hiệu suất là điều cần sự phối hợp giữa kỹ thuật và thiết kế.

## Kiến thức và công nghệ vận dụng

Để hoàn thành bài toán này, nhóm đã vận dụng một loạt các kiến thức và công nghệ từ nhiều lĩnh vực:

* **Khoa học dữ liệu (Data Science):** Hiểu về quy trình phân tích dữ liệu, tiền xử lý, và trực quan hóa.
* **Học máy (Machine Learning):** Áp dụng thuật toán KMeans, đánh giá mô hình bằng Elbow Method.
* **Ngôn ngữ lập trình Python:** Là ngôn ngữ chính để xây dựng toàn bộ ứng dụng, nhờ sự mạnh mẽ và thư viện phong phú.
* **Thư viện xử lý dữ liệu:** Pandas, NumPy để làm việc với dữ liệu dạng bảng và mảng.
* **Thư viện trực quan hóa:** Matplotlib, Seaborn để biểu diễn dữ liệu và kết quả mô hình.
* **Công cụ xây dựng giao diện web:** Streamlit giúp phát triển nhanh ứng dụng web, tương tác trực tiếp với người dùng cuối.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình bậc cao, được phát triển bởi Guido van Rossum và lần đầu tiên được phát hành vào năm 1991. Python nổi bật với cú pháp đơn giản, dễ đọc và mạnh mẽ trong việc xử lý dữ liệu.

### Đặc điểm của Python

Python có những đặc điểm nổi bật sau:

* **Cú pháp đơn giản**: Python sử dụng cú pháp gần gũi với ngôn ngữ tự nhiên, giúp lập trình viên dễ dàng đọc hiểu và viết code.
* **Thông dịch**: Python là ngôn ngữ thông dịch, không cần biên dịch trước khi chạy.
* **Đa nền tảng**: Có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux.
* **Thư viện phong phú**: Python có hệ sinh thái thư viện rất lớn, đặc biệt mạnh trong lĩnh vực khoa học dữ liệu và học máy.

### Ứng dụng của Python

Python được ứng dụng rộng rãi trong nhiều lĩnh vực:

* Phát triển web (Django, Flask)
* Khoa học dữ liệu và phân tích dữ liệu
* Học máy và trí tuệ nhân tạo
* Tự động hóa và scripting
* Phát triển ứng dụng desktop

## Danh sách (List) trong Python

List là một trong những kiểu dữ liệu cơ bản và quan trọng nhất trong Python, cho phép lưu trữ nhiều phần tử trong một biến duy nhất.

### Đặc điểm của List

* **Có thứ tự**: Các phần tử trong list có thứ tự và thứ tự này được duy trì.
* **Có thể thay đổi**: List là mutable, có thể thêm, xóa, sửa các phần tử sau khi tạo.
* **Cho phép trùng lặp**: List có thể chứa các phần tử có giá trị giống nhau.
* **Đa kiểu dữ liệu**: Một list có thể chứa các phần tử thuộc nhiều kiểu dữ liệu khác nhau.

### Các thao tác cơ bản với List

# Tạo list

my\_list = [1, 2, 3, 'python', True]

# Truy cập phần tử

first\_element = my\_list[0] # Kết quả: 1

# Thêm phần tử

my\_list.append(4) # Thêm vào cuối

my\_list.insert(1, 'new') # Thêm vào vị trí cụ thể

# Xóa phần tử

my\_list.remove('python') # Xóa theo giá trị

del my\_list[0] # Xóa theo index

## Thư viện Pandas

Pandas là thư viện Python mạnh mẽ để xử lý và phân tích dữ liệu, được phát triển bởi Wes McKinney. Pandas cung cấp các cấu trúc dữ liệu linh hoạt và hiệu quả để làm việc với dữ liệu có cấu trúc.

### Cấu trúc dữ liệu chính

**Series**: Cấu trúc dữ liệu một chiều, tương tự như mảng nhưng có thể chứa nhiều kiểu dữ liệu khác nhau và có index.

**DataFrame**: Cấu trúc dữ liệu hai chiều, tương tự như bảng trong cơ sở dữ liệu hoặc spreadsheet, với các hàng và cột có nhãn.

### Chức năng chính của Pandas

* **Đọc và ghi dữ liệu**: Hỗ trợ nhiều định dạng như CSV, Excel, JSON, SQL, HTML.
* **Xử lý dữ liệu thiếu**: Cung cấp các phương thức để xử lý giá trị null/NaN.
* **Lọc và truy vấn dữ liệu**: Cho phép lọc dữ liệu theo điều kiện phức tạp.
* **Groupby và aggregation**: Nhóm dữ liệu và tính toán thống kê.
* **Merge và join**: Kết hợp nhiều DataFrame với nhau.

### Ví dụ sử dụng cơ bản

import pandas as pd

# Tạo DataFrame

data = {'Name': ['Alice', 'Bob', 'Charlie'],

'Age': [25, 30, 35],

'City': ['New York', 'Paris', 'London']}

df = pd.DataFrame(data)

# Đọc dữ liệu từ file CSV

df = pd.read\_csv('data.csv')

# Xem thông tin cơ bản

print(df.head()) # 5 dòng đầu

print(df.info()) # Thông tin tổng quan

print(df.describe()) # Thống kê mô tả

## Thư viện Matplotlib và Seaborn

### Matplotlib

Matplotlib là thư viện trực quan hóa dữ liệu cơ bản và mạnh mẽ nhất trong Python, được phát triển bởi John D. Hunter. Matplotlib cung cấp API cấp thấp để tạo ra các biểu đồ tùy chỉnh chi tiết.

**Đặc điểm của Matplotlib:**

* Kiểm soát chi tiết cao về mọi yếu tố của biểu đồ
* Hỗ trợ nhiều loại biểu đồ: line plot, scatter plot, bar chart, histogram, pie chart
* Có thể xuất biểu đồ với chất lượng cao cho xuất bản
* Interface tương tự MATLAB

### Seaborn

Seaborn là thư viện trực quan hóa dữ liệu được xây dựng trên nền tảng Matplotlib, cung cấp interface cấp cao hơn và thân thiện hơn với người dùng.

**Ưu điểm của Seaborn:**

* Cú pháp đơn giản, trực quan
* Tích hợp tốt với Pandas DataFrame
* Biểu đồ đẹp mắt với style mặc định
* Hỗ trợ tốt cho phân tích dữ liệu thống kê
* Các biểu đồ chuyên biệt: heatmap, pairplot, violin plot

### Ví dụ sử dụng

import matplotlib.pyplot as plt

import seaborn as sns

# Matplotlib

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

plt.plot(x\_data, y\_data)

plt.title('Biểu đồ đường')

plt.xlabel('Trục X')

plt.ylabel('Trục Y')

plt.show()

# Seaborn

sns.set\_style("whitegrid")

sns.scatterplot(data=df, x='column1', y='column2', hue='category')

plt.title('Biểu đồ phân tán')

plt.show()

## Mô hình học máy không giám sát – K-Means

### Khái niệm về học máy không giám sát

Học máy không giám sát (Unsupervised Learning) là một nhánh của học máy trong đó thuật toán học từ dữ liệu không có nhãn (unlabeled data). Mục tiêu là khám phá các cấu trúc ẩn, mẫu hình hoặc mối quan hệ trong dữ liệu.

### Thuật toán K-Means

K-Means là một trong những thuật toán phân cụm (clustering) phổ biến nhất trong học máy không giám sát.

**Nguyên lý hoạt động:**

* **Khởi tạo**: Chọn số lượng cụm k và khởi tạo k centroids ngẫu nhiên
* **Gán nhãn**: Gán mỗi điểm dữ liệu vào cụm có centroid gần nhất
* **Cập nhật centroids**: Tính toán lại vị trí centroids dựa trên trung bình của các điểm trong cụm
* **Lặp lại**: Lặp lại bước 2 và 3 cho đến khi centroids không thay đổi hoặc thay đổi ít

### Ưu điểm và nhược điểm

**Ưu điểm:**

* Đơn giản, dễ hiểu và implement
* Hiệu quả với dữ liệu lớn
* Hoạt động tốt khi các cụm có hình dạng cầu và kích thước tương đương

**Nhược điểm:**

* Cần xác định trước số lượng cụm k
* Nhạy cảm với việc khởi tạo centroids ban đầu
* Không hiệu quả với cụm có hình dạng phức tạp
* Bị ảnh hưởng bởi outliers

### Ví dụ sử dụng K-Means

from sklearn.cluster import KMeans

import numpy as np

# Tạo dữ liệu mẫu

X = np.random.rand(100, 2)

# Áp dụng K-Means

kmeans = KMeans(n\_clusters=3, random\_state=42)

kmeans.fit(X)

# Lấy nhãn cụm và centroids

labels = kmeans.labels\_

centroids = kmeans.cluster\_centers\_

## Phương pháp Elbow

### Vấn đề xác định số cụm tối ưu

Một trong những thách thức chính khi sử dụng K-Means là xác định số lượng cụm k tối ưu. Phương pháp Elbow là một kỹ thuật phổ biến để giải quyết vấn đề này.

### Nguyên lý phương pháp Elbow

Phương pháp Elbow dựa trên việc tính toán WCSS (Within-Cluster Sum of Squares) - tổng bình phương khoảng cách từ các điểm đến centroid của cụm chứa nó.

**Các bước thực hiện:**

* Chạy K-Means với các giá trị k khác nhau (từ 1 đến n)
* Với mỗi k, tính toán WCSS
* Vẽ biểu đồ WCSS theo k
* Tìm điểm "khuỷu tay" (elbow) trên đường cong - điểm mà tốc độ giảm WCSS chậm lại đáng kể

### Cách xác định điểm Elbow

Điểm Elbow là điểm mà việc tăng thêm số cụm không mang lại cải thiện đáng kể về WCSS. Tại điểm này, đường cong tạo thành một "khuỷu tay" - từ giảm mạnh chuyển sang giảm chậm.

### Ví dụ implementation

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

def elbow\_method(X, max\_k=10):

wcss = []

k\_range = range(1, max\_k + 1)

for k in k\_range:

kmeans = KMeans(n\_clusters=k, random\_state=42)

kmeans.fit(X)

wcss.append(kmeans.inertia\_

*# Vẽ biểu đồ Elbow*

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

plt.plot(k\_range, wcss, marker='o')

plt.title('Phương pháp Elbow')

plt.xlabel('Số cụm (k)')

plt.ylabel('WCSS')

plt.grid(True)

plt.show()

return wcss

## Streamlit – Thư viện tạo ứng dụng web

### Giới thiệu về Streamlit

Streamlit là một framework Python mã nguồn mở được thiết kế để tạo ra các ứng dụng web tương tác cho machine learning và data science một cách nhanh chóng và đơn giản. Được phát triển bởi công ty Streamlit Inc., thư viện này cho phép các nhà khoa học dữ liệu và kỹ sư machine learning tạo ra các ứng dụng web mà không cần kiến thức sâu về web development.

### Đặc điểm nổi bật của Streamlit

**Đơn giản và dễ sử dụng:**

* Chỉ cần viết Python script thuần túy
* Không cần HTML, CSS, hoặc JavaScript
* API đơn giản và trực quan

**Tương tác thời gian thực:**

* Tự động cập nhật khi code thay đổi
* Widgets tương tác như slider, button, selectbox
* Kết quả hiển thị ngay lập tức

**Tích hợp tốt:**

* Hỗ trợ tốt các thư viện data science như Pandas, Matplotlib, Plotly
* Dễ dàng hiển thị DataFrame, biểu đồ, hình ảnh
* Tích hợp với các mô hình machine learning

### Các thành phần cơ bản

**Widgets đầu vào:**

* st.text\_input(): Nhập văn bản
* st.number\_input(): Nhập số
* st.slider(): Thanh trượt
* st.selectbox(): Dropdown menu
* st.multiselect(): Chọn nhiều giá trị
* st.checkbox(): Checkbox
* st.radio(): Radio buttons

**Hiển thị dữ liệu:**

* st.write(): Hiển thị văn bản, dữ liệu
* st.dataframe(): Hiển thị DataFrame
* st.table(): Hiển thị bảng tĩnh
* st.json(): Hiển thị JSON
* st.code(): Hiển thị code

**Biểu đồ và trực quan hóa:**

* st.line\_chart(): Biểu đồ đường
* st.bar\_chart(): Biểu đồ cột
* st.pyplot(): Matplotlib plots
* st.plotly\_chart(): Plotly charts

### Cấu trúc ứng dụng Streamlit cơ bản

import streamlit as st

import pandas as pd

import numpy as np

*# Tiêu đề ứng dụng*

st.title('Ứng dụng phân tích dữ liệu')

*# Sidebar*

st.sidebar.header('Tham số đầu vào')

user\_input = st.sidebar.slider('Chọn giá trị', 0, 100, 50)

*# Tải dữ liệu*

@st.cache\_data

def load\_data():

return pd.read\_csv('data.csv')

data = load\_data()

*# Hiển thị dữ liệu*

st.subheader('Dữ liệu thô')

st.dataframe(data)

*# Tạo biểu đồ*

st.subheader('Biểu đồ')

chart\_data = pd.DataFrame(

np.random.randn(20, 3),

columns=['A', 'B', 'C']

)

st.line\_chart(chart\_data)

### Deployment và chia sẻ

Streamlit cung cấp nhiều tùy chọn để deploy ứng dụng:

* **Streamlit Cloud**: Platform miễn phí của Streamlit
* **Heroku**: Platform cloud phổ biến
* **Docker**: Containerization
* **Local server**: Chạy trên máy local

### Ưu điểm và hạn chế

**Ưu điểm:**

* Học nhanh, sử dụng đơn giản
* Không cần kiến thức web development
* Phù hợp cho prototype và demo
* Cộng đồng lớn và tài liệu phong phú
* Miễn phí và mã nguồn mở

**Hạn chế:**

* Tùy chỉnh giao diện hạn chế
* Không phù hợp cho ứng dụng web phức tạp
* Performance có thể chậm với dữ liệu lớn
* Ít tùy chọn về state management

CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

Trong chương này, chúng tôi sẽ đi sâu vào trình bày toàn bộ quá trình thiết kế, phân tích và hiện thực hóa hệ thống phân loại khách hàng tiềm năng dựa trên dữ liệu giao dịch. Đây là một quy trình phức tạp, đòi hỏi sự phối hợp giữa các thành phần từ tiền xử lý dữ liệu, phân tích đặc trưng, lựa chọn và triển khai thuật toán học máy, đến việc trực quan hóa và cung cấp giao diện tương tác thân thiện với người dùng. Mục tiêu cuối cùng là phát triển một ứng dụng có khả năng áp dụng thực tế trong lĩnh vực phân tích hành vi khách hàng, hỗ trợ doanh nghiệp trong các quyết định kinh doanh chiến lược.

## Sơ đồ hệ thống

Hệ thống được thiết kế dưới dạng pipeline (đường ống xử lý dữ liệu), đảm bảo rằng dữ liệu được xử lý tuần tự qua các khối chức năng. Mỗi module chịu trách nhiệm một nhiệm vụ cụ thể, từ đầu vào cho đến đầu ra cuối cùng.

### Module Nhập Dữ Liệu

* Đây là điểm khởi đầu của hệ thống, nơi người dùng tải dữ liệu từ tệp định dạng CSV (Mall Customer Segmentation Data). Module sử dụng pandas.read\_csv() để nạp dữ liệu và thực hiện kiểm tra định dạng, loại bỏ những bản ghi không hợp lệ.
* Việc nhập dữ liệu được thiết kế linh hoạt, cho phép hệ thống hoạt động với các phiên bản dữ liệu khác nhau, miễn là có cùng cấu trúc.

### Module Tiền Xử Lý

* Dữ liệu đầu vào thường có thể chứa các giá trị thiếu, lỗi nhập liệu hoặc định dạng không thống nhất. Module này sẽ thực hiện việc loại bỏ hoặc thay thế các giá trị thiếu, chuẩn hóa thang đo của các đặc trưng để đưa về cùng đơn vị (z-score hoặc MinMax).
* Các đặc trưng được chọn lọc dựa trên sự liên quan đến hành vi khách hàng, tiêu biểu như: Tuổi (Age), Thu nhập hàng năm (Annual Income) và Mức độ chi tiêu (Spending Score).

### Module Phân Cụm (Clustering)

* Đây là phần cốt lõi của hệ thống, nơi thuật toán KMeans được áp dụng để chia tập khách hàng thành các cụm có đặc điểm tương đồng. Mỗi cụm đại diện cho một nhóm khách hàng có hành vi, xu hướng chi tiêu khác biệt.
* Đặc biệt, hệ thống còn tích hợp phương pháp Elbow để xác định số lượng cụm tối ưu. Điều này giúp tránh việc phân cụm quá ít hoặc quá nhiều, làm mất ý nghĩa phân tích.

### Module Trực Quan Hóa

* Một hệ thống học máy hiệu quả không thể thiếu phần trực quan hóa. Module này sử dụng các thư viện mạnh mẽ như matplotlib và seaborn để hiển thị kết quả phân cụm bằng biểu đồ scatter (2D hoặc 3D), biểu đồ Elbow và histogram để phân tích đặc điểm từng cụm.
* Trực quan hóa không chỉ giúp người dùng hiểu rõ hơn về kết quả mà còn hỗ trợ kiểm tra tính hợp lý của việc phân cụm.

### Module Giao Diện Web

* Được xây dựng bằng Streamlit – một thư viện mạnh mẽ và đơn giản dành cho khoa học dữ liệu – giao diện người dùng cho phép tải dữ liệu, chọn số cụm, hiển thị kết quả phân tích và xuất báo cáo.
* Tính tương tác cao của giao diện giúp người dùng không cần kiến thức lập trình cũng có thể sử dụng hệ thống một cách hiệu quả.

## Sơ đồ khối các thuật toán chính

Việc tích hợp và vận hành hiệu quả các thuật toán học máy là yếu tố then chốt trong chương trình.

* Thuật toán Elbow:
  + Đây là kỹ thuật xác định số cụm tối ưu trong bài toán phân cụm. Ý tưởng chính là chạy thuật toán KMeans với số cụm từ 1 đến một giá trị lớn hơn (ví dụ 10), sau đó vẽ biểu đồ tổng sai số nội tại (inertia) theo số cụm.
  + Điểm khuỷu tay (elbow) là điểm mà sau đó đường cong bắt đầu dẹt lại – biểu thị số cụm tối ưu vì sau đó việc tăng cụm không còn cải thiện đáng kể kết quả phân cụm.
* Thuật toán KMeans:
  + Một thuật toán phân cụm phổ biến thuộc nhóm học không giám sát. KMeans chia dữ liệu thành K cụm sao cho tổng bình phương khoảng cách giữa các điểm và tâm cụm là nhỏ nhất.
  + Thuật toán hoạt động bằng cách khởi tạo ngẫu nhiên K tâm cụm, sau đó lặp lại hai bước: gán điểm dữ liệu vào cụm gần nhất và tính lại tâm cụm cho đến khi hội tụ.
* Trực quan hóa kết quả:
  + Dữ liệu sau phân cụm sẽ được biểu diễn trên đồ thị nhằm cung cấp cái nhìn trực quan về cách các cụm được phân chia.
  + Trong trường hợp nhiều chiều, phương pháp PCA (Principal Component Analysis) có thể được áp dụng để giảm chiều dữ liệu trước khi trực quan hóa.

## Cấu trúc dữ liệu

Tập dữ liệu sử dụng trong chương trình là "Mall Customer Segmentation Data", gồm các trường đặc trưng cho hành vi khách hàng:

* CustomerID: Mã khách hàng – định danh duy nhất cho mỗi bản ghi.
* Gender: Giới tính – được mã hóa hoặc loại bỏ tùy thuộc vào cách xử lý đặc trưng.
* Age: Tuổi – thể hiện độ tuổi, có thể ảnh hưởng đến thói quen tiêu dùng.
* Annual Income (k$): Thu nhập – một chỉ số kinh tế phản ánh khả năng chi tiêu.
* Spending Score (1-100): Mức độ tiêu dùng – đánh giá định lượng dựa trên hành vi mua sắm.

Dữ liệu sau xử lý được lưu trữ dưới dạng DataFrame – một cấu trúc dữ liệu mạnh mẽ trong thư viện Pandas, cho phép thao tác dễ dàng và trực quan.

## Chương trình

Hệ thống được phát triển theo hướng module hóa, tức là chia nhỏ chương trình thành các phần chức năng độc lập, dễ kiểm thử và tái sử dụng:

* load\_data(path): Hàm này thực hiện việc đọc dữ liệu từ đường dẫn được chỉ định, kiểm tra định dạng và trả về DataFrame.
* preprocess\_data(df): Làm sạch dữ liệu, chuẩn hóa thang đo và lựa chọn các đặc trưng phù hợp để đưa vào mô hình học máy.
* find\_optimal\_clusters(df): Thử nghiệm nhiều giá trị K khác nhau trong KMeans, vẽ biểu đồ Elbow để xác định số cụm tối ưu.
* apply\_kmeans(df, k): Thực hiện thuật toán KMeans với số cụm K đã chọn, trả về nhãn cụm cho từng khách hàng.
* plot\_clusters(df, labels): Hiển thị kết quả phân cụm bằng biểu đồ scatter, giúp người dùng đánh giá trực quan tính phân biệt giữa các cụm.
* main(): Hàm chính sử dụng Streamlit để tạo giao diện tương tác, xử lý toàn bộ luồng dữ liệu và trả về kết quả trên web.

CHƯƠNG IV: THỰC NGHIỆM VÀ KẾT LUẬN

## Thực nghiệm

Để kiểm tra hiệu quả hoạt động và tính đúng đắn của hệ thống phân cụm khách hàng, nhóm đã tiến hành thực nghiệm toàn diện trên bộ dữ liệu **Mall Customer Segmentation** và ghi nhận kết quả theo từng giai đoạn xử lý.

### Kiểm thử chức năng

### **Chức năng tải dữ liệu:**

Người dùng có thể tải tệp .csv trực tiếp từ giao diện. Hệ thống kiểm tra định dạng hợp lệ, loại bỏ dòng thiếu giá trị và hiển thị bảng dữ liệu mẫu ngay trên web.

#### **Chức năng tiền xử lý dữ liệu:**

Hệ thống tự động chuẩn hóa các cột “Age”, “Annual Income (k$)”, và “Spending Score (1–100)”. Kết quả kiểm tra cho thấy không còn giá trị thiếu, các đặc trưng đã được đưa về cùng thang đo (scale), giúp mô hình phân cụm hoạt động hiệu quả hơn.

#### **Chức năng tiền xử lý dữ liệu:**

Khi chạy thuật toán KMeans với số cụm từ 1 đến 10, biểu đồ Elbow được hiển thị giúp người dùng trực quan xác định "điểm khuỷu tay" — nơi mà đường cong inertia bắt đầu “gãy”. Trong thực nghiệm, số cụm tối ưu rơi vào khoảng **k = 5**, điều này phản ánh sự phân hóa rõ rệt trong hành vi tiêu dùng của khách hàng.

#### Chức năng phân cụm khách hàng (KMeans):

Sau khi xác định số cụm, người dùng thực hiện phân nhóm. Kết quả được lưu trong một cột mới và hiển thị lại bảng dữ liệu kèm theo nhãn cụm. Chạy nhiều lần cho ra kết quả ổn định, chứng minh tính hội tụ của thuật toán.

#### Chức năng trực quan hóa dữ liệu:

Hệ thống sinh ra biểu đồ scatter 2D sử dụng PCA, và biểu đồ phân phối (histogram) theo từng cụm khách hàng. Các biểu đồ giúp người dùng nhanh chóng nhận ra sự khác biệt giữa các nhóm về độ tuổi, thu nhập và hành vi tiêu dùng. Giao diện đẹp, tốc độ hiển thị tốt, màu sắc rõ ràng giúp dễ phân tích.

#### Giao diện người dùng (Streamlit):

Tất cả các chức năng được tích hợp vào một giao diện web hoàn chỉnh, dễ sử dụng, trực quan. Người dùng có thể thao tác liên tục mà không cần làm mới trang. Thử nghiệm trên nhiều trình duyệt khác nhau (Chrome, Edge) đều cho kết quả ổn định.

### Hình ảnh mô tả các tính năng

A screenshot of a computer

AI-generated content may be incorrect.

Hình 1: Upload dataset

A screenshot of a computer

AI-generated content may be incorrect. Hình 2: Show dataset lên web

A graph with a line

AI-generated content may be incorrect.

Hình 3: Dự đoán phân cụm

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4: Số liệu phân cụm

A screenshot of a computer

AI-generated content may be incorrect.

Hình 5: Trung bình của từng cụm

A chart with colorful dots

AI-generated content may be incorrect.

Hình 6: Biểu đồ scatter

A colorful pie chart with numbers and a few percentages

AI-generated content may be incorrect.

Hình 7: Biểu đồ tròn

A screenshot of a computer

AI-generated content may be incorrect.

Hình 8: Hình ảnh tải dữ liệu về sau khi phân cụm

## Kết luận

### Những kết quả đã đạt được

Sau quá trình thực hiện, nhóm đã hoàn thành một hệ thống phân cụm khách hàng hoạt động hiệu quả với các đặc điểm nổi bật:

* Phân tích được hành vi khách hàng thông qua dữ liệu định lượng: tuổi, thu nhập, điểm tiêu dùng.
* Tự động xác định số lượng cụm tối ưu bằng phương pháp Elbow.
* Phân cụm khách hàng bằng thuật toán KMeans và trực quan hóa kết quả chi tiết, dễ hiểu.
* Xây dựng giao diện web hoàn chỉnh sử dụng Streamlit, cho phép người dùng dễ dàng tương tác và khám phá dữ liệu.

Hệ thống không chỉ hỗ trợ người dùng hiểu rõ hơn về đặc điểm từng nhóm khách hàng, mà còn có thể trở thành công cụ tiền đề cho các chiến dịch tiếp thị mục tiêu hoặc tư vấn chiến lược kinh doanh.

### Kiến thức và kinh nghiệm thu nhận

Thông qua đề tài này, nhóm đã tích lũy được nhiều kỹ năng quý báu, bao gồm:

* Khả năng phân tích và xử lý dữ liệu thực tế bằng Python.
* Hiểu sâu hơn về quy trình xây dựng một hệ thống khoa học dữ liệu: từ tiền xử lý, phân tích mô hình đến triển khai và trực quan hóa.
* Nắm vững thuật toán KMeans, Elbow Method và cách áp dụng PCA để giảm chiều dữ liệu cho trực quan.
* Rèn luyện khả năng làm việc nhóm, quản lý tiến độ dự án, phát triển phần mềm có tính thực tiễn.

### Hướng phát triển trong tương lai

Dự án có nhiều tiềm năng mở rộng và cải tiến:

* **Mở rộng thuộc tính dữ liệu**: Thêm các yếu tố như tần suất mua hàng, loại sản phẩm ưa thích, thời gian tương tác,... để phân tích đa chiều hơn.
* **So sánh nhiều thuật toán phân cụm khác nhau** như DBSCAN, Gaussian Mixture để tìm phương pháp tối ưu hơn KMeans.
* **Ứng dụng vào thực tế doanh nghiệp**: Kết nối hệ thống với cơ sở dữ liệu CRM để cập nhật khách hàng theo thời gian thực.
* **Tích hợp dashboard tương tác nâng cao** bằng Plotly hoặc Dash, cho phép lọc, tìm kiếm và phân tích sâu từng nhóm khách hàng.

CHƯƠNG V: KẾT LUẬN

Dự án " Phân loại khách hàng tiềm năng" đã được nhóm thực hiện nhằm giải quyết bài toán thực tế về phân khúc khách hàng dựa trên các đặc trưng như thu nhập và điểm chi tiêu. Qua đó, nhóm không chỉ rèn luyện kiến thức về khoa học dữ liệu mà còn phát triển các kỹ năng thực hành như tiền xử lý dữ liệu, xây dựng mô hình học máy không giám sát, đánh giá mô hình và trình bày kết quả thông qua giao diện người dùng.

Bộ dữ liệu sử dụng là "Mall Customer Segmentation", với ba thuộc tính chính: tuổi, thu nhập và điểm chi tiêu. Sau bước tiền xử lý, nhóm áp dụng thuật toán KMeans để phân cụm khách hàng. Để lựa chọn số cụm tối ưu, nhóm sử dụng phương pháp Elbow, từ đó xác định được 5 cụm đại diện cho các nhóm khách hàng có hành vi tương đồng. Kết quả phân cụm được trực quan hóa bằng biểu đồ scatter 2D, giúp người dùng dễ dàng nhận diện và phân tích đặc điểm từng nhóm.

Ứng dụng web được xây dựng bằng Streamlit cho phép người dùng tải dữ liệu mới, chọn số cụm, hiển thị biểu đồ phân nhóm và xem thông tin từng cụm một cách trực quan. Đây là bước quan trọng giúp chuyển giao kết quả phân tích cho người dùng không chuyên về kỹ thuật, đặc biệt là các bộ phận marketing hoặc chăm sóc khách hàng.

Mặc dù đạt được kết quả khả quan, dự án vẫn còn một số hạn chế. KMeans giả định các cụm có hình dạng hình cầu và nhạy với nhiễu, do đó có thể chưa phù hợp với các tập dữ liệu phức tạp hơn. Bộ dữ liệu hiện tại cũng còn đơn giản, chưa phản ánh đầy đủ các yếu tố hành vi người tiêu dùng. Ngoài ra, ứng dụng web mới chỉ hỗ trợ hiển thị kết quả, chưa có các chức năng nâng cao như lưu trữ, xuất báo cáo, hoặc gợi ý chiến lược kinh doanh.

Từ dự án này, nhóm đã hiểu rõ hơn về toàn bộ quy trình triển khai một bài toán khoa học dữ liệu thực tế — từ phân tích yêu cầu, xử lý dữ liệu, huấn luyện mô hình, đến trực quan hóa và đóng gói sản phẩm. Kỹ năng lập trình Python, thao tác với Pandas, Scikit-learn, và xây dựng giao diện với Streamlit cũng được củng cố vững chắc.

Trong tương lai, nhóm mong muốn phát triển hệ thống theo hướng mở rộng dữ liệu, thử nghiệm thêm các thuật toán như DBSCAN hoặc GMM, và bổ sung các tính năng cao cấp cho ứng dụng như xuất báo cáo tự động, lưu lịch sử người dùng, hoặc kết nối với cơ sở dữ liệu doanh nghiệp. Dự án là bước đệm quan trọng giúp nhóm hiểu và ứng dụng hiệu quả khoa học dữ liệu trong thực tiễn, đặc biệt trong lĩnh vực phân tích và chăm sóc khách hàng.