BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
HỌC MÁY CƠ BẢN

ĐỀ TÀI:

ỨNG DỤNG THUẬT TOÁN K-MEANS ĐỂ XÁC ĐỊNH VỊ TRÍ TIỀM NĂNG MỞ CỬA HÀNG MỚI

Giảng viên hướng dẫn : TS.NGUYỄN THỊ THANH TÂN

Sinh viên thực hiện : CAO QUANG LINH
NGUYỄN KHÁNH DƯƠNG
NGUYỄN MINH KHUÊ

Ngành : CÔNG NGHỆ THÔNG TIN

Chuyên ngành : CÔNG NGHỆ PHẦN MỀM

Lớp : D17CNPM3

Khóa : 2022-2027

Hà Nội, tháng 5 năm 2025

PHIẾU CHẤM ĐIỂM

STT	Họ và tên sinh viên	Nội dung thực hiện	Điểm	Chữ ký
1	Nguyễn Khánh Dương 22810310344			
2	Cao Quang Linh 22810310301			
3	Nguyễn Minh Khuê 22810310322			

Họ và tên giảng viên	Chữ ký	Ghi chú
Giảng viên chấm 1:		
Giảng viên chấm 2:		

MỤC LỤC

Mục lục **3**
DANH MỤC HÌNH ẢNH **5**
DANH MỤC TỪ VIẾT TẮT **6**
LỜI MỞ ĐẦU **1**
CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY **2**
1.1 Khái niệm về học máy 2
1.2 Các bước cơ bản trong Machine Learning 2
1.3 Phân nhóm các thuật toán học máy 3
1.3.1 Học có giám sát (Supervised Learning). 4
1.3.2 Học không giám sát 5
1.3.3 Học bán giám sát 6
1.4 Các khái niệm mà bạn cần biết trong Machine Learning 6
1.5 Ứng dụng thực tế của Machine Learning 7
CHƯƠNG 2: THUẬT TOÁN K-MEANS CLUSTERING TRONG BÀI TOÁN PHÂN CỤM **10**
2.1 Tổng quan về thuật toán K-Means Clustering 10
2.1.1 Thuật toán K-Means là gì? 10
2.1.2 Giới thiệu về K-Means 10
2.1.3 Thuật toán K-Means 13
2.2 Đánh giá thuật toán K-Means 15
2.2.1 Độ phức tạp của thuật toán K-Means 16
2.2.2 Ưu điểm của K-Means 16
2.2.3 Hạn chế của K-Means 17
2.2.4 Giải pháp khắc phục hạn chế 17
2.3 So sánh thuật toán K-Means với các thuật toán phân cụm khác trong bài toán xác định vị trí mở cửa hàng 18
2.3.1 Thuật toán K-Means 18

2.3.2 Thuật toán DBSCAN 19
2.3.3 Thuật toán Hierarchical Clustering 19
2.3.4 Thuật toán Gaussian Mixture Models (GMM) 20
2.3.5 Tính phù hợp của các thuật toán trong bài toán mở cửa hàng 21
2.3.6 Tóm lại 21
CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN 23
3.1 Giới thiệu bài toán 23
3.2 Giải quyết bài toán bằng thuật toán K-Means 23
3.3 Cài đặt chương trình. 24
3.3.1 Một số thư viện cần sử dụng. 24
3.3.2 Hàm Tiền xử lý và làm sạch dữ liệu 24
3.3.3 Hàm Áp dụng thuật toán K-Means cho dữ liệu 25
3.3.4 Hàm Đặt tên cho các cụm theo đặc trưng trung bình 25
3.3.5 Hàm Tính điểm độ phù hợp (score) cho một khu vực dựa trên Density, Income, Competitors. 26
3.3.6 Hàm Tính toán độ phù hợp cho một điểm dữ liệu dựa trên mật độ dân cư, thu nhập, và số đối thủ. 26
3.3.7 Hàm Vẽ biểu đồ Elbow để tìm số cụm tối ưu. 27
3.3.8 Hàm In thông tin chi tiết về các cụm và khu vực có điểm score cao nhất trong từng cụm. 27
3.3.9 Hàm Vẽ biểu đồ phân tán 3D của các cụm sử dụng Plotly. 28
3.3.10 Hàm Hiển thị bảng kết quả phân cụm trong GUI sử dụng Tkinter. 28
3.3.11 Hàm Xuất kết quả phân cụm ra file Excel. 29
3.3.12 Hàm Chọn cụm chứa khu vực có điểm score cao nhất. 29
3.3.13 Hàm Main 29
3.4 Kết quả 31
KẾT LUẬN 33
TÀI LIỆU THAM KHẢO 34

DANH MỤC HÌNH ẢNH

Hình 1.1: Machine Learning Workflow 2
Hình 1.2: Học có giám sát 4
Hình 1.3: Nhận dạng chữ viết tay 5
Hình 1.4: Học không giám sát 6
Hình 1.5: Machine Learning được ứng dụng trong xử lý ảnh 8
Hình 1.6: Học bán giám sát 9
Hình 2.1: Bài toán với 3 clusters. 11
Hình 2.2: Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau. 12
Hình 2.3: Hàm mất mát WCSS 14
Hình 3.1. Một số thư viên cần sử dụng 24
Hình 3.2. Hàm Tiền xử lý và làm sạch dữ liệu 24
Hình 3.3. Hàm Áp dụng thuật toán K-Means cho dữ liệu 25
Hình 3.4.Hàm Đặt tên cho các cụm theo đặc trưng trung bình 25
Hình 3.5. Tính điểm độ phù hợp (score) cho một khu vực dựa trên Density, Income, Competitors. 26
Hình 3.6. Tính toán độ phù hợp cho một điểm dữ liệu dựa trên mật độ dân cư, thu nhập, và số đối thủ. 26
Hình 3.7. Vẽ biểu đồ Elbow để tìm số cụm tối ưu. 27
Hình 3.8. In thông tin chi tiết về các cụm và khu vực có điểm score cao nhất trong từng cụm. 27
Hình 3.9. Vẽ biểu đồ phân tán 3D của các cụm sử dụng Plotly. 28
Hình 3.10. Hàm Hiển thị bảng kết quả phân cụm trong GUI sử dụng Tkinter. 28
Hình 3.11. Hàm Xuất kết quả phân cụm ra file Excel. 29
Hình 3.12.Hàm Chọn cụm chứa khu vực có điểm score cao nhất. 29
Hình 3.13.Hàm Main-1 29
Hình 3.14. Hàm Main-2 30
Hình 3.15. Biểu đồ cụm tối ưu. 31
Hình 3.16. Bảng kết quả 31
Hình 3.17.Biểu dồ 3D 32
Hình 3.18. Thông tin các cụm 32

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý Nghĩa
WCSS	Within-Cluster Sum of Squares
IQR	Interquartile Range
DBSCAN	Density-based spatial clustering of applications with noise
GMM	Gaussian Mixture Models

LỜI MỞ ĐẦU

Trong môi trường kinh doanh bán lẻ đầy cạnh tranh, việc lựa chọn vị trí mở cửa hàng đóng vai trò then chốt trong việc đảm bảo thành công và tối ưu hóa lợi nhuận. Một vị trí phù hợp không chỉ giúp doanh nghiệp tiếp cận đúng đối tượng khách hàng mà còn giảm thiểu rủi ro từ cạnh tranh và các yếu tố kinh tế-xã hội. Tuy nhiên, với khối lượng dữ liệu lớn và sự phức tạp của các đặc trưng như mật độ dân số, thu nhập bình quân, hay số lượng đối thủ cạnh tranh, việc phân tích thủ công trở nên kém hiệu quả và khó đạt được độ chính xác cao. Điều này đặt ra nhu cầu áp dụng các phương pháp phân tích dữ liệu tiên tiến, trong đó học máy nổi lên như một công cụ mạnh mẽ để hỗ trợ ra quyết định chiến lược.

Học máy, đặc biệt là các thuật toán phân cụm không giám sát như K-Means, đã chứng minh được hiệu quả trong nhiều lĩnh vực, từ phân khúc thị trường, phân tích khách hàng, đến tối ưu hóa chuỗi cung ứng. Trong bài toán xác định vị trí mở cửa hàng, K-Means cho phép phân nhóm các khu vực dựa trên các đặc trưng kinh tế-xã hội, từ đó xác định các cụm có tiềm năng cao, trung bình, hoặc thấp. Kết quả phân cụm không chỉ giúp doanh nghiệp lựa chọn vị trí tối ưu mà còn cung cấp cái nhìn sâu sắc về đặc điểm của từng phân khúc thị trường, từ đó định hình chiến lược kinh doanh phù hợp.

Nhận thấy tiềm năng của K-Means trong việc giải quyết bài toán thực tiễn, đề tài này tập trung vào việc áp dụng thuật toán K-Means để phân cụm các khu vực dựa trên các đặc trưng như mật độ dân số, thu nhập bình quân, và số đối thủ cạnh tranh. Mục tiêu chính là xây dựng một mô hình phân cụm hiệu quả, giúp xác định các khu vực tiềm năng để mở cửa hàng mới, đồng thời đánh giá kết quả thông qua các chỉ số và trực quan hóa để rút ra các nhận xét có giá trị thực tiễn. Nội dung nghiên cứu bao quát từ lý thuyết cơ bản của thuật toán K-Means, các kỹ thuật tối ưu hóa mô hình, đến việc triển khai thực tế và phân tích kết quả, nhằm mang lại một giải pháp khoa học và khả thi. Thông qua nghiên cứu này, nhóm kỳ vọng đóng góp một phương pháp phân tích dữ liệu hiện đại, hỗ trợ doanh nghiệp tối ưu hóa quy trình lựa chọn vị trí mở cửa hàng và khẳng định vai trò của học máy trong các ứng dụng thực tiễn.

CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY

1.1 Khái niệm về học máy

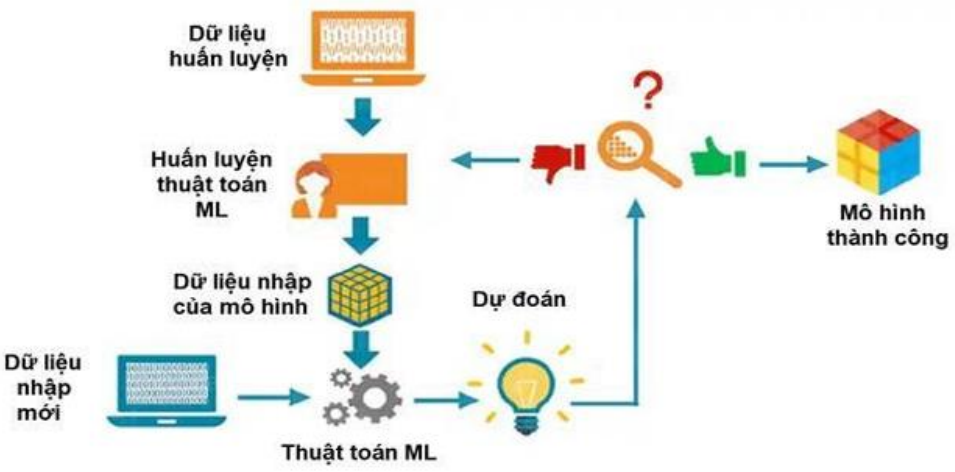
Học máy (Machine learning) là một lĩnh vực con của Trí tuệ nhân tạo (Artificial Intelligence) sử dụng các thuật toán cho phép máy tính có thể học từ dữ liệu để thực hiện các công việc thay vì được lập trình một cách rõ ràng, cung cấp cho hệ thống khả năng tự động học hỏi và cải thiện hiệu suất, độ chính xác dựa trên những kinh nghiệm từ dữ liệu đầu vào. Học máy tập trung vào việc phát triển các phần mềm, chương trình máy tính có thể truy cập vào dữ liệu và tận dụng nguồn dữ liệu đó để tự học.

Học máy vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Các mô hình học máy yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán học máy thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán học máy với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

1.2 Các bước cơ bản trong Machine Learning

Trong một Machine Learning Workflow gồm 5 bước cơ bản:



Hình 1.1: Machine Learning Workflow

– **Thu thập dữ liệu:**

Để máy tính có thể học và đưa ra các dự đoán, phân tích, lập trình viên cần cung cấp một bộ dữ liệu gọi là Dataset cho máy. Thông thường, bạn có thể thu thập các dữ liệu này hoặc sử dụng các Dataset có sẵn trên các nền tảng hỗ trợ học lập trình học máy. Cần lưu ý lựa chọn những bộ dữ liệu từ những nguồn chính thống, như vậy máy tính mới có thể học được một cách chính xác và đưa ra những kết quả đúng đắn, có tỷ lệ hiệu quả cao hơn.

– **Tiền xử lý:**

Bước tiền xử lý trong Machine Learning dùng để chuẩn hóa các dữ liệu vừa thu thập được, giúp loại bỏ các thuộc tính không cần thiết, những dữ liệu bị hỏng, thiếu. Đồng thời bước này sẽ tiến hành gán nhãn, mã hóa các đặc trưng, trích xuất những đặc trưng và rút gọn bộ dữ liệu mà vẫn đảm bảo kết quả đầu ra.

Bước Preprocessing chiếm thời gian nhất trong toàn bộ workflow, tỷ lệ thuận với độ lớn, khối lượng dữ liệu mà bạn cung cấp. Từ đó, tổng thời gian thực hiện hai bước 1 và 2 chiếm tổng thời gian khoảng 70% toàn quá trình.

– **Huấn luyện mô hình:**

Bước huấn luyện mô hình này sử dụng để cho máy học trên dữ liệu mà bạn cung cấp và tiến hành xử lý ở hai bước đầu tiên.

– **Đánh giá mô hình:**

Sau khi đã tiến hành huấn luyện mô hình, bước tiếp theo trong Machine Learning đó là đánh giá mô hình vừa tạo ra. Tùy thuộc vào từng các loại độ đo khác nhau mà mô hình vừa huấn luyện được đánh giá là tốt hay không tốt khác nhau. Về cơ bản, độ chính xác của mô hình vừa huấn luyện đạt trên 80% được cho là đảm bảo hiệu quả.

– **Cải thiện:**

Trong bước cải thiện này, những mô hình sau khi đã được đánh giá nếu không đạt chuẩn thì sẽ được tiến hành lại bước thứ 3 cho đến khi độ chính xác đạt đúng kỳ vọng cần thiết. Ba bước cuối của Machine Learning Workflow là khoảng 30% tổng quá trình.

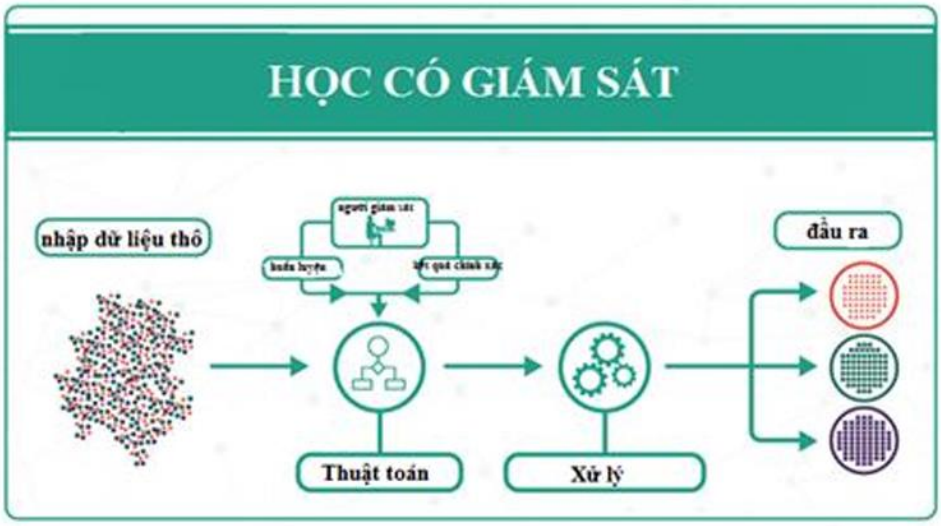
1.3 Phân nhóm các thuật toán học máy

Hiện nay có rất nhiều cách để tiến hành học máy, tuy nhiên thông thường sẽ được chia làm hai loại sau: Supervised learning (Học có giám sát), Unsupervised learning (Học không có giám sát) và Semi-supervised learning (Học bán giám sát).

Bên cạnh đó, còn có một loại học máy khác là Reinforcement Learning (học tăng cường).

1.3.1 Học có giám sát (Supervised Learning).

Học có giám sát là một hướng tiếp cận của Máy học để làm cho máy tính có khả năng "học". Trong hướng tiếp cận này, người ta "huấn luyện" máy tính dựa trên những quan sát có dán nhãn. Ta có thể hình dung những quan sát này như là những câu hỏi, và nhãn của chúng là những câu trả lời. Ý tưởng của học có giám sát là: bằng việc ghi nhớ và tổng quát hóa một số quy tắc từ một tập câu hỏi có đáp án trước, máy tính sẽ có thể trả lời được những câu hỏi dù chưa từng gặp phải, nhưng có mối liên quan.



Hình 1.2: Học có giám sát

Ví dụ ta dạy máy tính "1 + 1 = 2" và hy vọng nó sẽ học được phép tính cộng x + 1 và trả lời được là "2 + 1 = 3". Học có giám sát mô phỏng việc con người học bằng cách đưa ra dự đoán của mình cho một câu hỏi, sau đó đối chiếu với đáp án. Sau đó con người rút ra phương pháp để trả lời đúng không chỉ câu hỏi đó, mà cho những câu hỏi có dạng tương tự.

Trong học có giám sát, các quan sát bắt buộc phải được dán nhãn trước. Đây chính là một trong những nhược điểm của phương pháp này, bởi vì không phải lúc nào việc dán nhãn chính xác cho quan sát cũng dễ dàng. Ví dụ như trong dịch thuật, từ một câu của ngôn ngữ gốc có thể dịch thành rất nhiều phiên bản khác nhau trong ngôn

ngữ cần dịch sang.

Tuy nhiên, việc quan sát được dán nhãn cũng lại chính là ưu điểm của học có giám sát bởi vì một khi đã thu thập được một bộ dữ liệu lớn được dán nhãn chuẩn xác, thì việc huấn luyện trở nên dễ dàng hơn rất nhiều so với khi dữ liệu không được dán nhãn.

Ví dụ: Trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào.

Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

Hình 1.3: Nhận dạng chữ viết tay

Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

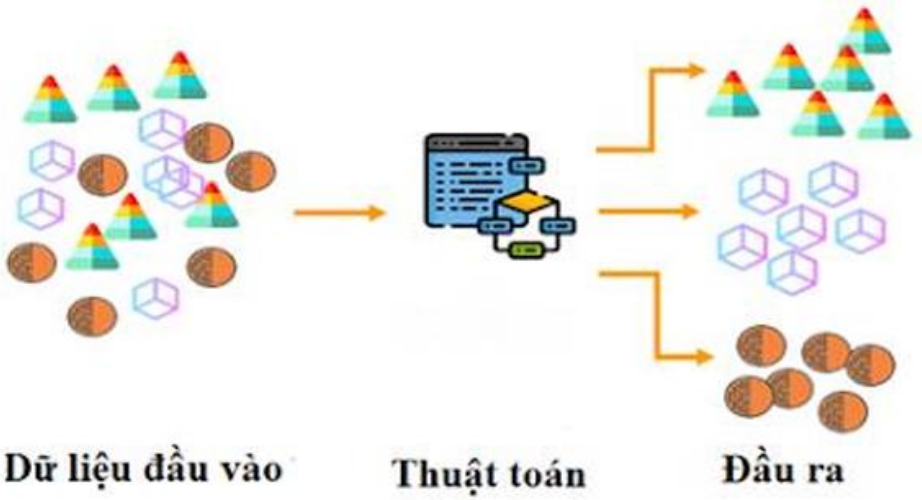
1.3.2 Học không giám sát

Trong thuật toán này, chúng ta không biết được dữ liệu đầu ra hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán Học không giám sát dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm hoặc giảm số chiều của dữ liệu để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Học không giám sát là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Những thuật toán loại này được gọi là Học không giám sát vì không giống như Học có giám sát, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào.

Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm không giám sát được đặt tên theo nghĩa này.



Hình 1.4: Học không giám sát

Ứng dụng phổ biến nhất của Học không giám sát là bài toán về phân cụm.

1.3.3 Học bán giám sát

Ranh giới giữa học có giám sát và học không giám sát đôi khi không rõ ràng. Có những thuật toán mà tập huấn luyện bao gồm các cặp (đầu vào, đầu ra) và dữ liệu khác chỉ có đầu vào. Những thuật toán này được gọi là học bán giám sát (semi-supervised learning).

Xét một bài toán phân loại mà tập huấn luyện bao gồm các bức ảnh được gán nhãn 'chó' hoặc 'mèo' và rất nhiều bức ảnh thú cưng tải từ Internet chưa có nhãn. Thực tế cho thấy ngày càng nhiều thuật toán rơi vào nhóm này vì việc thu thập nhãn cho dữ liệu có chi phí cao và tốn thời gian. Chẳng hạn, chỉ một phần nhỏ trong các bức ảnh y học có nhãn vì quá trình gán nhãn tốn thời gian và cần sự can thiệp của các chuyên gia. Một ví dụ khác, thuật toán dò tìm vật thể cho xe tự lái được xây dựng trên một lượng lớn video thu được từ camera xe hơi; tuy nhiên, chỉ một lượng nhỏ các vật thể trong các video huấn luyện đó được xác định cụ thể.

1.4 Các khái niệm mà bạn cần biết trong Machine Learning

Dataset (bộ dữ liệu): Đây là tập dữ liệu ở dạng nguyên thủy, chưa được xử lý

mà lập trình viên thu thập được ở bước đầu tiên (Data collection).

Data point (điểm dữ liệu): Đây là một phần của Dataset, dùng biểu thị cho một quan sát. Từng data point có nhiều thuộc tính hoặc đặc trưng khác nhau, được chia làm dữ liệu số và dữ liệu không phải số. Data point được biểu diễn thành từng dòng, mỗi dòng có thể có 1 hay nhiều đặc trưng dữ liệu.

Training data (dữ liệu học) và Test data (dữ liệu kiểm tra): Training data sử dụng để cho máy huấn luyện mô hình, test data dùng để dự đoán các kết quả đồng thời cũng đánh giá mô hình. Tỷ lệ giữa hai loại dữ liệu này thường là 8/2 (train/ test).

Model (mô hình): Là những mô hình dùng để train trên một training data dựa theo thuật toán mà mô hình đó đang sử dụng. Từ đó, mô hình sẽ đưa ra kết quả, quyết định dựa trên những kiến thức đã được học.

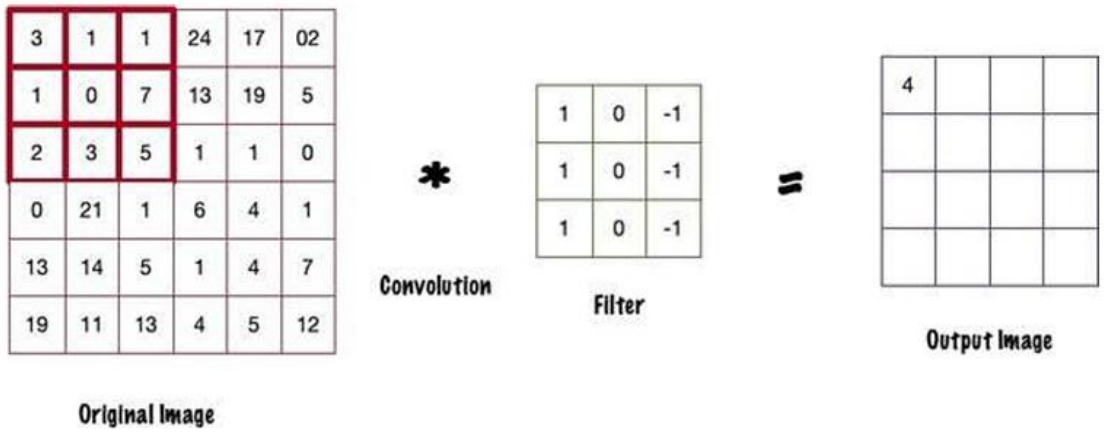
1.5 Ứng dụng thực tế của Machine Learning

Machine Learning được ứng dụng khá phổ biến trong các lĩnh vực khác nhau như:

Xử lý ảnh

Bài toán xử lý ảnh hay image processing sử dụng để giải quyết những vấn đề liên quan đến hình ảnh hay thực hiện các phép biến đổi trên dữ liệu dạng ảnh, ví dụ như:

- **Gắn thẻ ảnh:** Tương tự như Facebook, thuật toán có thể tự động phát hiện khuôn mặt người dùng, từ đó gắn thẻ những người dùng mà bạn kết bạn có trong hình ảnh. Thực chất, thuật toán này được triển khai từ những bức ảnh bạn tự gắn thẻ bản thân trước đó.
- **Nhận dạng ký tự:** Thuật toán sử dụng chuyển đổi dữ liệu dạng văn bản trên giấy tờ thành dữ liệu số.



Hình 1.5: Machine Learning được ứng dụng trong xử lý ảnh

Phân tích văn bản

Phân tích văn bản hay Text analysis sử dụng trong việc trích xuất, phân loại những thông tin từ văn bản (có thể là bài đăng Facebook, thư điện tử, tài liệu,…) với các ứng dụng:

- **Lọc tin nhắn rác**: Đây là ứng dụng phổ biến nhất của Machine Learning trong phân tích văn bản. Bộ lọc sẽ tiến hành phân loại xem thư điện tử, tin nhắn có ohair là spam không dự trên nội dung mà tin nhắn gửi đến.

- **Khai thác thông tin:** Từ một văn bản, máy có thể phân tích để trích xuất ra những thông tin quan trọng, cần thiết như tên người, các keyword,…

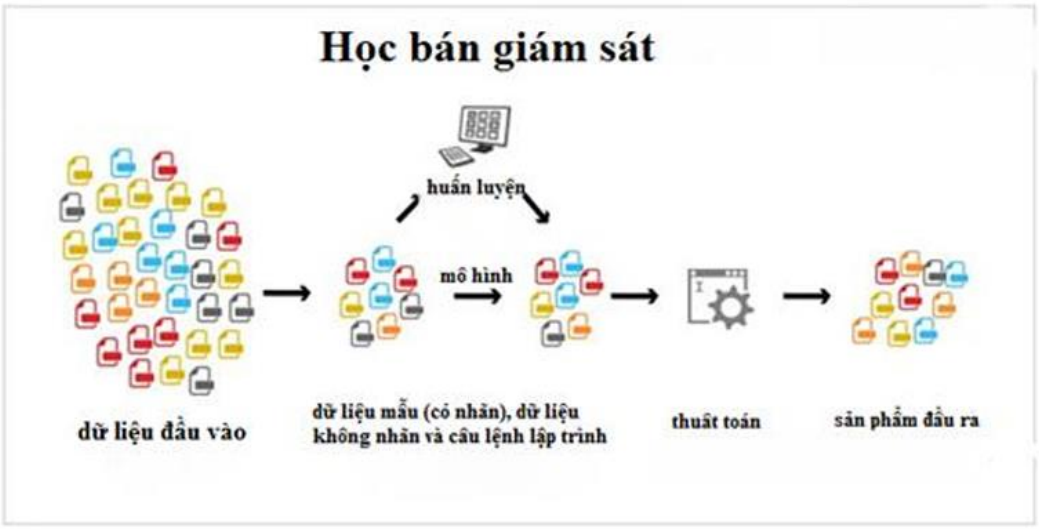
Khai phá dữ liệu

Khai phá dữ liệu hay Data mining là quá trình tìm ra những thông tin, dữ liệu có giá trị, có ích hoặc tiến hành đưa ra những dự đoán thông qua dữ liệu được cung cấp. Từ một bảng dữ liệu lớn gồm nhiều bản ghi với mỗi bản ghi là một đối tượng mà máy cần thực hiện học, tương ứng mỗi đặc trưng là một cột. Máy có thể dự đoán giá trị của bản ghi mới dựa vào nội dung bản ghi trước đó đã được học. Một vài ứng dụng phổ biến của Machine Learning trong khai phá như:

- **Phát hiện những bất thường:** Sử dụng để tìm ra những ngoại lệ (có thể là phát hiện gian lận khi sử dụng thẻ tín dụng).

Phân cụm

- **Dự đoán:** Dự đoán giá nhà, giá xe,… thông qua việc điền những dữ liệu cần vào các bản ghi để từ đó máy tiến hành dự đoán sau khi đã học những dữ liệu cơ bản trước đó.



Hình 1.6: Học bán giám sát

CHƯƠNG 2: THUẬT TOÁN K-MEANS CLUSTERING TRONG BÀI TOÁN PHÂN CỤM

2.1 Tổng quan về thuật toán K-Means Clustering

2.1.1 Thuật toán K-Means là gì?

Thuật toán phân cụm K-Means là một phương pháp được sử dụng trong phân tích tính chất cụm của dữ liệu. Nó đặc biệt được sử dụng nhiều trong khai phá dữ liệu và thống kê. Nó phân vùng dữ liệu thành k cụm khác nhau. Giải thuật này giúp chúng ta xác định được dữ liệu của chúng ta nó thực sự thuộc về nhóm nào.

Để các bạn dễ hình dung ứng dụng của thuật toán. Chúng ta hãy quan sát một ví dụ thực tế như sau:

Trong các mô hình kinh doanh, doanh nghiệp sẽ chia nhỏ tệp khách hàng ra thành những nhóm đối tượng khác nhau để có thể áp dụng những chiến lược kinh doanh cụ thể cho từng nhóm đối tượng. Điều này giúp cho khách hàng được tiếp cận với các sản phẩm thật sự phù hợp với bản thân họ. Sự phù hợp đó sẽ kéo doanh số của chúng ta tăng lên. Vấn đề đặt ra là làm sao có thể chia nhỏ tệp khách hàng đó ra khi mà số lượng hóa đơn là rất lớn và chúng ta không thể ngồi để phân tích từng vị khách.

Và mục tiêu của các thuật toán phân cụm là từ tập dữ liệu khổng lồ đó. Làm sao chúng ta biết có những nhóm dữ liệu đặc trưng nào trong đó? Từng dữ liệu trong đó thuộc vào nhóm nào? Đó là cái mà thuật toán phân cụm của chúng ta cần đi tìm câu trả lời.

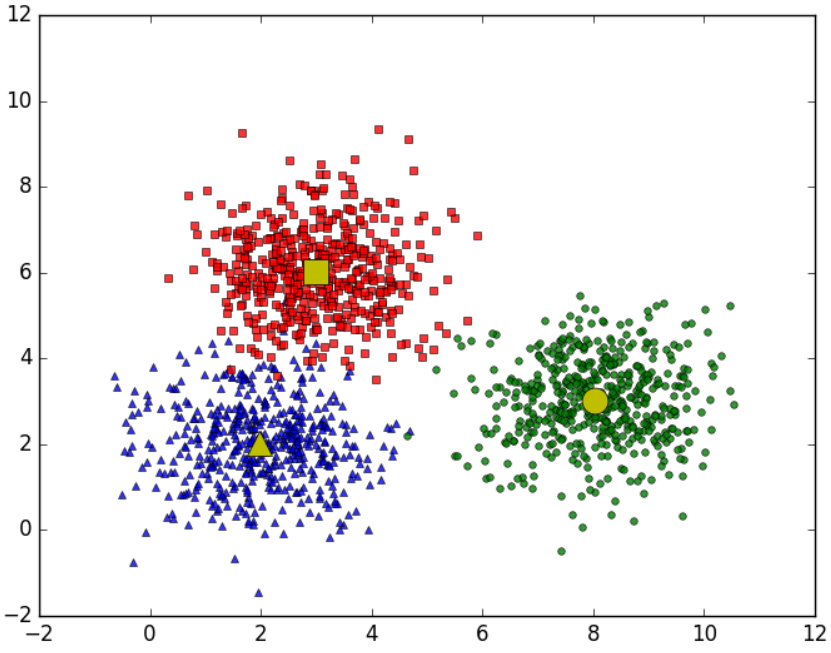
2.1.2 Giới thiệu về K-Means

Trong thuật toán K-Means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho *dữ liệu trong cùng một cụm có tính chất giống nhau*.

Ví dụ: Một công ty muốn tạo ra những chính sách ưu đãi cho những nhóm khách hàng khác nhau dựa trên sự tương tác giữa mỗi khách hàng với công ty đó (số năm là khách hàng; số tiền khách hàng đã chi trả cho công ty; độ tuổi; giới tính; thành

phố; nghề nghiệp; …). Giả sử công ty đó có rất nhiều dữ liệu của rất nhiều khách hàng nhưng chưa có cách nào chia toàn bộ khách hàng đó thành một số nhóm/cụm khác nhau. Nếu một người biết Machine Learning được đặt câu hỏi này, phương pháp đầu tiên anh (chị) ta nghĩ đến sẽ là K-Means Clustering. Vì nó là một trong những thuật toán đầu tiên mà anh ấy tìm được trong các cuốn sách, khóa học về Machine Learning. Và tôi cũng chắc rằng anh ấy đã đọc blog Machine Learning cơ bản. Sau khi đã phân ra được từng nhóm, nhân viên công ty đó có thể lựa chọn ra một vài khách hàng trong mỗi nhóm để quyết định xem mỗi nhóm tương ứng với nhóm khách hàng nào. Phần việc cuối cùng này cần sự can thiệp của con người, nhưng lượng công việc đã được rút gọn đi rất nhiều.

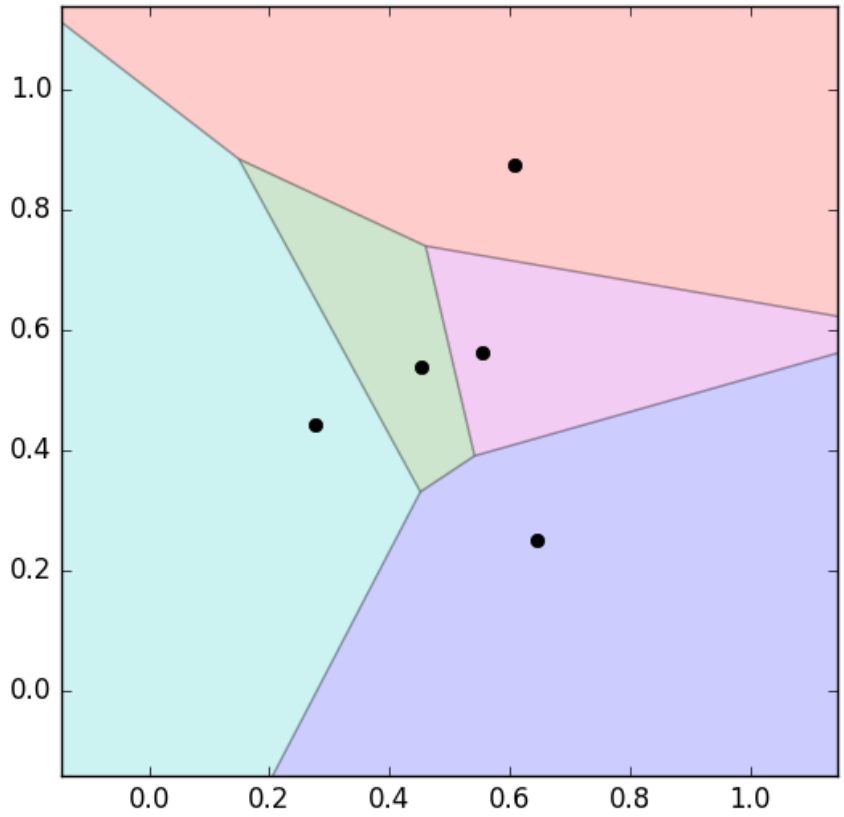
Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm *ở gần nhau trong một không gian nào đó* (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về 3 cụm dữ liệu (từ giờ tôi sẽ viết gọn là *cluster*).



Hình 2.1: Bài toán với 3 clusters.

Giả sử mỗi cluster có một điểm đại diện (*center*) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó. Tới đây, chúng ta có một bài toán thú vị: *Trên một vùng biển hình vuông lớn có ba đảo hình vuông, tam giác, và tròn màu vàng như hình trên. Một điểm trên biển được gọi là thuộc lãnh hải của một đảo nếu nó nằm gần đảo này hơn so với hai đảo kia . Hãy xác định ranh giới lãnh hải của các đảo.*

Hình dưới đây là một hình minh họa cho việc phân chia lãnh hải nếu có 5 đảo khác nhau được biểu diễn bằng các hình tròn màu đen:



Hình 2.2: Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau.

Chúng ta thấy rằng đường phân định giữa các lãnh hải là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, lãnh hải của một đảo sẽ là một hình đa giác.

Cách phân chia này trong toán học được gọi là Voronoi Diagram.

2.1.3 Thuật toán K-Means

a. Tóm tắt thuật toán.

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K.

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

– Bước 1: Chọn K điểm bất kỳ làm các center ban đầu.

– Bước 2: Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.

– Bước 3: Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

– Bước 4: Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

– Bước 5: Quay lại bước 2.

Chúng ta có thể đảm bảo rằng thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp. Thật vậy, vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Theo kiến thức về dãy số trong chương trình cấp 3: *nếu một dãy số giảm và bị chặn dưới thì nó hội tụ!* Hơn nữa, số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và chúng ta có thể dừng thuật toán tại đây.

b. Chi tiết về hàm mất mát (WCSS)

Hàm mất mát trong K-Means được gọi là **Within-Cluster Sum of Squares (WCSS)**, được định nghĩa như sau:

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

trong đó:

- C_i là cụm thứ i,
- μ_i là tâm của cụm i,
- $\|x - \mu_i\|^2$ là bình phương khoảng cách Euclidean từ điểm x đến tâm cụm μ_i.

Hình 2.3: Hàm mất mát WCSS

Mục tiêu của thuật toán là tối thiểu hóa WCSS, tức là đảm bảo các điểm trong cùng cụm càng gần tâm cụm càng tốt. Trong bài toán xác định vị trí mở cửa hàng, WCSS giúp đánh giá mức độ tương đồng của các khu vực (dựa trên mật độ dân số, thu nhập bình quân, số đối thủ cạnh tranh) trong cùng một cụm [Tiep Vu, 2017].

c. Áp dụng vào bài toán thực tế

Trong bối cảnh xác định vị trí tiềm năng mở cửa hàng, thuật toán K-Means được sử dụng để phân cụm các khu vực dựa trên các thuộc tính như:

Mật độ dân số (người/km²),

Thu nhập bình quân (VND/tháng),

Số đối thủ cạnh tranh.

Ví dụ minh họa: Giả sử có dữ liệu từ 5 khu vực với các giá trị:

- Khu vực A: (12,000, 12,000,000, 3),
- Khu vực B: (8,000, 10,000,000, 5),
- Khu vực C: (15,000, 14,000,000, 2),
- Khu vực D: (10,000, 11,000,000, 4),
- Khu vực E: (6,000, 9,000,000, 6).

Khi chạy K-Means với K=3 K = 3 K=3:

- Bước 1: Chọn 3 điểm ngẫu nhiên làm tâm ban đầu (ví dụ: A, C, E).

– Bước 2: Gán các khu vực vào cụm gần nhất dựa trên khoảng cách Euclidean.

– Bước 3: Cập nhật tâm mới bằng trung bình các điểm trong cụm.

Kết quả này giúp doanh nghiệp chọn khu vực phù hợp dựa trên đặc điểm kinh doanh.

d. Tối ưu hóa thuật toán

Để cải thiện hiệu quả, có thể sử dụng **K-Means++** để chọn tâm ban đầu thông minh hơn:

– Thay vì chọn ngẫu nhiên, K-Means++ ưu tiên các điểm xa nhau làm tâm ban đầu, giảm nguy cơ hội tụ vào nghiệm không tối ưu.

– Quy trình:

- Bước 1: Chọn ngẫu nhiên một điểm làm tâm đầu tiên.
- Bước 2: Chọn điểm tiếp theo với xác suất tỷ lệ với bình phương khoảng cách đến tâm gần nhất.
- Bước 3: Lặp lại đến khi đủ K tâm.

Ngoài ra, việc chuẩn hóa dữ liệu (sử dụng StandardScaler) là cần thiết để đưa các thuộc tính về cùng thang đo, tránh thiên vị thuộc tính có giá trị lớn (như thu nhập)

e. Lưu ý khi triển khai

– **Chọn K**: Số cụm K cần được xác định trước, có thể dùng phương pháp Elbow hoặc Silhouette để chọn K tối ưu.

– **Nhạy cảm với ngoại lai**: Các khu vực có dữ liệu bất thường (như thu nhập cực cao) có thể làm lệch tâm cụm, nên cần xử lý ngoại lai trước.

– **Khởi tạo ban đầu**: Nếu chọn tâm ban đầu không tốt, thuật toán có thể hội tụ chậm hoặc cho kết quả không ổn định.

2.2 Đánh giá thuật toán K-Means

Thuật toán K-Means là một trong những phương pháp phân cụm phổ biến nhất trong học máy nhờ tính đơn giản và hiệu quả. Tuy nhiên, để áp dụng thuật toán này một cách hiệu quả, cần hiểu rõ độ phức tạp tính toán, các ưu điểm nổi bật, và những

hạn chế của nó. Phần này sẽ trình bày chi tiết các khía cạnh đánh giá thuật toán K-Means, bao gồm độ phức tạp, ưu điểm, hạn chế, và các giải pháp khắc phục hạn chế.

2.2.1 Độ phức tạp của thuật toán K-Means

Độ phức tạp tính toán của thuật toán K-Means phụ thuộc vào số lượng điểm dữ liệu, số cụm, và số lần lặp của thuật toán. Cụ thể, độ phức tạp được biểu diễn như sau:

Độ phức tạp: $O_{(K.N.l)}$ với l: số lần lặp

2.2.2 Ưu điểm của K-Means

Thuật toán K-Means có nhiều ưu điểm khiến nó trở thành lựa chọn phổ biến trong các bài toán phân cụm:

Khả năng mở rộng: K-Means có thể dễ dàng xử lý dữ liệu mới bằng cách cập nhật các tâm cụm khi có thêm điểm dữ liệu. Điều này đặc biệt hữu ích trong các ứng dụng thời gian thực, như phân đoạn khách hàng trực tuyến.

Bảo đảm hội tụ: Thuật toán luôn hội tụ sau một số bước lặp hữu hạn, vì hàm mất mát WCSS (Within-Cluster Sum of Squares) giảm dần sau mỗi vòng lặp và bị chặn dưới bởi 0.

Đảm bảo số lượng cụm: K-Means luôn tạo ra đúng K cụm, giúp doanh nghiệp dễ dàng kiểm soát số nhóm cần phân tích.

Cụm không rỗng: Mỗi cụm luôn có ít nhất một điểm dữ liệu, tránh tình trạng cụm rỗng gây khó khăn trong diễn giải.

Cụm không phân cấp và không chồng chéo: Các cụm được tạo ra độc lập, không có cấu trúc phân cấp, và mỗi điểm dữ liệu chỉ thuộc một cụm, giúp kết quả dễ hiểu và áp dụng.

Tính gần gũi trong cụm: Mọi điểm dữ liệu trong một cụm luôn gần tâm cụm của nó hơn so với tâm của bất kỳ cụm nào khác, đảm bảo tính đồng nhất trong cụm.

Ví dụ minh họa: Trong bài toán phân cụm khu vực để mở cửa hàng, K-Means đảm bảo rằng mỗi khu vực được gán vào một cụm duy nhất (ví dụ: khu vực đông

dân, thu nhập cao), và các khu vực trong cùng cụm có đặc điểm tương đồng, giúp doanh nghiệp dễ dàng đưa ra chiến lược kinh doanh riêng cho từng cụm.

2.2.3 Hạn chế của K-Means

Mặc dù có nhiều ưu điểm, K-Means cũng tồn tại một số hạn chế cần lưu ý:

Không xử lý được cụm không lồi hoặc phức tạp: K-Means giả định các cụm có dạng hình cầu và sử dụng khoảng cách Euclidean, do đó không thể phát hiện các cụm có hình dạng bất thường (ví dụ: cụm hình lưỡi liềm hoặc vòng tròn đồng tâm).

Khó khăn trong khởi tạo tâm cụm:

– Việc chọn ngẫu nhiên các tâm cụm ban đầu có thể dẫn đến kết quả không tối ưu, vì thuật toán có thể hội tụ vào cực tiểu địa phương.

– Độ hội tụ của thuật toán phụ thuộc mạnh vào các tâm cụm khởi tạo, khiến kết quả có thể khác nhau giữa các lần chạy.

Khó chọn số cụm tối ưu: Việc xác định K đòi hỏi phải thử nghiệm nhiều lần, sử dụng các phương pháp như Elbow hoặc Silhouette, làm tăng thời gian phân tích.

Nhạy cảm với nhiễu và ngoại lai: Các điểm dữ liệu bất thường (ví dụ: một khu vực có thu nhập cực cao) có thể làm lệch tâm cụm, ảnh hưởng đến chất lượng phân cụm.

Giới hạn ở biên cụm rõ ràng: K-Means giả định mỗi điểm dữ liệu chỉ thuộc một cụm, không phù hợp với các bài toán mà một đối tượng có thể thuộc nhiều cụm hoặc biên giữa các cụm mờ (fuzzy).

Ví dụ minh họa: Trong bài toán phân cụm khách hàng, nếu dữ liệu chứa một số khách hàng có hành vi mua sắm bất thường (ví dụ: chi tiêu hàng tỷ đồng mỗi tháng), các tâm cụm có thể bị kéo lệch, khiến các cụm khác không phản ánh đúng đặc điểm của phần lớn khách hàng. Tương tự, nếu khách hàng có hành vi thuộc cả hai nhóm (ví dụ: vừa mua quần áo vừa mua điện tử), K-Means không thể gán họ vào cả hai cụm.

2.2.4 Giải pháp khắc phục hạn chế

Để khắc phục các hạn chế của K-Means, có thể áp dụng các giải pháp sau:

Sử dụng K-Means++: Thay vì chọn tâm ngẫu nhiên, K-Means++ chọn các tâm ban đầu sao cho chúng cách xa nhau, giảm nguy cơ hội tụ vào cực tiểu địa phương và cải thiện tốc độ hội tụ.

Tiền xử lý dữ liệu: Loại bỏ ngoại lai bằng các phương pháp như Z-score hoặc IQR(độ trải giữa), và chuẩn hóa dữ liệu để giảm tác động của nhiễu. Ví dụ: trước khi chạy K-Means, doanh nghiệp có thể lọc bỏ các khu vực có mật độ dân số vượt quá ngưỡng bất thường.

Sử dụng các phương pháp chọn K: Áp dụng phương pháp Elbow hoặc chỉ số Silhouette để xác định số cụm tối ưu một cách khoa học, thay vì thử nghiệm ngẫu nhiên.

Kết hợp với các thuật toán khác: Đối với các cụm không lồi hoặc phức tạp, có thể sử dụng các thuật toán như DBSCAN hoặc Gaussian Mixture Models (GMM) thay vì K-Means.

Áp dụng phân cụm mờ (Fuzzy Clustering): Nếu cần gán một điểm dữ liệu vào nhiều cụm, có thể sử dụng các thuật toán như Fuzzy C-Means, cho phép mỗi điểm có mức độ thuộc về nhiều cụm

2.3 So sánh thuật toán K-Means với các thuật toán phân cụm khác trong bài toán xác định vị trí mở cửa hàng

2.3.1 Thuật toán K-Means

Cách hoạt động: K-Means là một thuật toán phân cụm dựa trên khoảng cách, chia dữ liệu thành K K K cụm bằng cách tối thiểu hóa tổng bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm gần nhất (WCSS). Thuật toán lặp qua hai bước chính: gán mỗi điểm dữ liệu vào cụm có tâm gần nhất (dựa trên khoảng cách Euclidean) và cập nhật tâm cụm bằng trung bình của các điểm trong cụm. Quá trình lặp lại cho đến khi các tâm cụm không thay đổi đáng kể hoặc đạt số vòng lặp tối đa.

Áp dụng vào bài toán mở cửa hàng: Trong bài toán này, K-Means phân chia các khu vực thành K K K cụm, mỗi cụm đại diện cho một nhóm khu vực có đặc điểm tương đồng (ví dụ: khu vực đông dân và thu nhập cao, khu vực trung bình, hoặc khu vực ít tiềm năng). Ví dụ, với dữ liệu gồm 100 khu vực có mật độ dân số (5,000–20,000 người/km²), thu nhập bình quân (8,000,000–15,000,000 VND/tháng), và số đối thủ cạnh tranh (1–10), K-Means có thể nhóm các khu vực thành 3 cụm. Doanh nghiệp có thể sử dụng kết quả để ưu tiên các khu vực thuộc cụm tiềm năng cao (mật độ dân số cao, thu nhập cao, ít đối thủ).

Đặc điểm nổi bật:

- Yêu cầu xác định trước số cụm K, thường được chọn bằng các phương pháp như Elbow hoặc Silhouette.
- Giả định các cụm có dạng hình cầu và kích thước tương đương.
- Phù hợp với dữ liệu có cấu trúc rõ ràng và các cụm tách biệt.

2.3.2 Thuật toán DBSCAN

Cách hoạt động: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán phân cụm dựa trên mật độ, nhóm các điểm dữ liệu nằm gần nhau trong không gian mật độ cao thành một cụm và coi các điểm nằm ở vùng mật độ thấp là ngoại lai (noise). Thuật toán sử dụng hai tham số chính: khoảng cách tối đa ϵ \epsilon ϵ (xác định vùng lân cận của một điểm) và số điểm tối thiểu minPts \text{minPts} minPts (yêu cầu để một điểm được coi là "điểm lõi"). Các điểm lõi cùng vùng lân cận được gộp thành cụm, trong khi các điểm không thuộc cụm nào được xem là ngoại lai.

Áp dụng vào bài toán mở cửa hàng: DBSCAN có thể được sử dụng để phân cụm các khu vực dựa trên mật độ dân số và thu nhập bình quân, đồng thời phát hiện các khu vực bất thường (ví dụ: khu vực có thu nhập cực cao hoặc mật độ dân số cực thấp). Ví dụ, trong tập dữ liệu 100 khu vực, DBSCAN có thể nhóm các khu vực có đặc điểm tương đồng vào các cụm có hình dạng bất kỳ (không nhất thiết là hình cầu như K-Means) và loại bỏ các khu vực không phù hợp (ngoại lai) khỏi danh sách tiềm năng. Điều này hữu ích khi dữ liệu có các khu vực đặc biệt không thuộc nhóm nào rõ ràng.

Đặc điểm nổi bật:

- Không yêu cầu xác định trước số cụm, phù hợp khi số lượng cụm không rõ ràng.
- Có khả năng xử lý các cụm có hình dạng bất kỳ và phát hiện ngoại lai.
- Phụ thuộc vào việc chọn ϵ \epsilon ϵ và minPts \text{minPts} minPts, đòi hỏi thử nghiệm kỹ lưỡng.

So sánh với K-Means: K-Means yêu cầu số cụm K cố định và giả định cụm hình cầu, phù hợp với dữ liệu có cấu trúc đều. DBSCAN linh hoạt hơn trong việc xử lý cụm không đều và ngoại lai, nhưng khó áp dụng nếu dữ liệu có mật độ không đồng nhất (ví dụ: các khu vực có mật độ dân số thay đổi mạnh).

2.3.3 Thuật toán Hierarchical Clustering

Cách hoạt động: Hierarchical Clustering (phân cụm phân cấp) xây dựng một cây phân cấp (dendrogram) bằng cách gộp hoặc chia các điểm dữ liệu dựa trên khoảng cách giữa chúng. Có hai phương pháp chính:

- **Gộp (agglomerative)**: Bắt đầu với mỗi điểm dữ liệu là một cụm riêng, sau đó gộp các cặp cụm gần nhau nhất (dựa trên khoảng cách, thường là Euclidean) cho đến khi tất cả điểm thuộc một cụm duy nhất.
- **Chia (divisive)**: Bắt đầu với toàn bộ dữ liệu trong một cụm, sau đó chia thành các cụm nhỏ hơn. Phương pháp gộp phổ biến hơn.

Số cụm được xác định bằng cách cắt dendrogram ở một mức độ phù hợp, dựa trên khoảng cách hoặc số cụm mong muốn.

Áp dụng vào bài toán mở cửa hàng: Hierarchical Clustering có thể được sử dụng để phân cụm các khu vực và cung cấp một cái nhìn phân cấp về mối quan hệ giữa chúng. Ví dụ, với dữ liệu 100 khu vực, thuật toán có thể tạo một dendrogram cho thấy cách các khu vực được gộp thành các nhóm lớn (ví dụ: khu vực đô thị vs. khu vực nông thôn), sau đó chia nhỏ thành các nhóm chi tiết hơn (ví dụ: khu vực đô thị tiềm năng cao vs. tiềm năng thấp). Doanh nghiệp có thể chọn số cụm bằng cách cắt dendrogram, ví dụ, tại mức tạo ra 3 cụm tương tự như K-Means.

Đặc điểm nổi bật:

- Tạo ra cấu trúc phân cấp, cho phép phân tích dữ liệu ở nhiều mức độ chi tiết.
- Không yêu cầu xác định trước số cụm, nhưng cần chọn điểm cắt dendrogram.
- Phù hợp với dữ liệu nhỏ hoặc trung bình, nhưng tốn tài nguyên tính toán với dữ liệu lớn.

So sánh với K-Means: K-Means nhanh hơn và phù hợp với dữ liệu lớn, trong khi Hierarchical Clustering cung cấp cái nhìn phân cấp, hữu ích khi doanh nghiệp muốn hiểu mối quan hệ giữa các khu vực ở các cấp độ khác nhau. Tuy nhiên, K-Means dễ triển khai hơn và ít tốn tài nguyên tính toán.

2.3.4 Thuật toán Gaussian Mixture Models (GMM)

Cách hoạt động: Gaussian Mixture Models (GMM) là một thuật toán phân cụm xác suất, giả định rằng dữ liệu được tạo ra từ một hỗn hợp của các phân phối Gaussian. Mỗi cụm được mô tả bởi một phân phối Gaussian với trung bình, hiệp phương sai, và trọng số riêng. Thuật toán sử dụng phương pháp Expectation-Maximization (EM) để ước lượng các tham số của các phân phối Gaussian, từ đó gán xác suất cho mỗi điểm dữ liệu thuộc vào một cụm.

Áp dụng vào bài toán mở cửa hàng: GMM có thể được sử dụng để phân cụm các khu vực với khả năng gán "mềm" (soft clustering), nghĩa là mỗi khu vực có xác suất thuộc vào nhiều cụm thay vì chỉ thuộc một cụm duy nhất như K-Means. Ví dụ, một khu vực có thể có 70% xác suất thuộc cụm tiềm năng cao và 30% thuộc cụm trung bình. Điều này hữu ích khi các khu vực có đặc điểm không hoàn toàn tách biệt, ví dụ, một khu vực có mật độ dân số cao nhưng thu nhập trung bình. GMM có thể mô tả các cụm có hình dạng elip, linh hoạt hơn giả định hình cầu của K-Means.

Đặc điểm nổi bật:

- Hỗ trợ gán xác suất, cho phép phân cụm mềm.
- Có thể mô tả các cụm có hình dạng elip hoặc kích thước khác nhau.

- Yêu cầu xác định trước số cụm, tương tự K-Means, nhưng phức tạp hơn trong tính toán.

So sánh với K-Means: K-Means sử dụng phân cụm cứng (mỗi điểm chỉ thuộc một cụm) và giả định cụm hình cầu, trong khi GMM linh hoạt hơn với phân cụm mềm và cụm không đồng nhất. Tuy nhiên, GMM phức tạp hơn và tốn tài nguyên tính toán, đặc biệt với dữ liệu lớn, trong khi K-Means đơn giản và nhanh hơn.

2.3.5 Tính phù hợp của các thuật toán trong bài toán mở cửa hàng

Trong bài toán xác định vị trí mở cửa hàng, lựa chọn thuật toán phụ thuộc vào đặc điểm của dữ liệu và mục tiêu kinh doanh:

- **K-Means**: Phù hợp khi dữ liệu có cấu trúc rõ ràng, các khu vực có đặc điểm tách biệt, và doanh nghiệp muốn một giải pháp nhanh, dễ triển khai. Ví dụ, nếu dữ liệu 100 khu vực có các cụm rõ ràng (đông dân vs. thưa dân), K-Means là lựa chọn tốt với chi phí tính toán thấp.
- **DBSCAN**: Thích hợp khi dữ liệu có ngoại lai (như khu vực có dữ liệu bất thường) hoặc các cụm có hình dạng không đều. Tuy nhiên, doanh nghiệp cần thử nghiệm để chọn ϵ \epsilon ϵ và minPts \text{minPts} minPts, điều này có thể tốn thời gian.
- **Hierarchical Clustering**: Hữu ích khi doanh nghiệp muốn phân tích phân cấp, ví dụ, hiểu cách các khu vực được chia thành nhóm lớn trước khi chia nhỏ. Tuy nhiên, nó kém hiệu quả với dữ liệu lớn.
- **GMM**: Tốt khi các khu vực có đặc điểm không tách biệt rõ ràng và doanh nghiệp muốn biết xác suất một khu vực thuộc vào các nhóm khác nhau. GMM phù hợp với dữ liệu phức tạp hơn, nhưng đòi hỏi tài nguyên tính toán lớn.

Ví dụ minh họa: Với dữ liệu 100 khu vực, nếu doanh nghiệp muốn phân tích nhanh với 3 cụm cố định, K-Means là lựa chọn tối ưu. Nếu dữ liệu có nhiều khu vực bất thường (như khu vực có thu nhập cực cao), DBSCAN có thể được dùng để loại bỏ ngoại lai. Nếu doanh nghiệp muốn một phân tích chi tiết với cấu trúc phân cấp, Hierarchical Clustering là phù hợp. Cuối cùng, nếu cần xác suất để đánh giá mức độ tiềm năng, GMM sẽ cung cấp thông tin chi tiết hơn.

2.3.6 Tóm lại

Việc so sánh K-Means với DBSCAN, Hierarchical Clustering, và GMM cho thấy mỗi thuật toán có cách tiếp cận riêng, phù hợp với các đặc điểm dữ liệu và mục tiêu khác nhau. Trong bài toán xác định vị trí mở cửa hàng, K-Means thường được ưu tiên nhờ tính đơn giản, hiệu quả, và khả năng xử lý dữ liệu lớn. Tuy nhiên, doanh nghiệp nên cân nhắc sử dụng DBSCAN cho dữ liệu có ngoại lai, Hierarchical Clustering cho phân tích phân cấp, hoặc GMM cho các trường hợp cần phân cụm

mềm. Việc lựa chọn thuật toán cần dựa trên phân tích dữ liệu ban đầu và thử nghiệm thực tế để đảm bảo kết quả phân cụm phù hợp với chiến lược kinh doanh

CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN

3.1 Giới thiệu bài toán

Trong lĩnh vực kinh doanh bán lẻ, việc lựa chọn vị trí mở cửa hàng đóng vai trò quan trọng trong việc đảm bảo thành công và tối ưu hóa doanh thu. Bài toán xác định vị trí tiềm năng yêu cầu phân tích các khu vực dựa trên các đặc trưng kinh tế-xã hội để xác định các nhóm khu vực có đặc điểm tương đồng, từ đó lựa chọn vị trí phù hợp nhất. Thuật toán K-Means, một phương pháp phân cụm không giám sát, được sử dụng để nhóm các khu vực thành các cụm dựa trên các đặc trưng như mật độ dân số, thu nhập bình quân, và số đối thủ cạnh tranh.

Tập dữ liệu giả lập được sử dụng trong bài toán này chứa 15 bản ghi, mỗi bản ghi đại diện cho một khu vực với 3 đặc trưng chính:

- **Mật độ dân số**: Số người trên mỗi km^2
- **Thu nhập bình quân**: Thu nhập trung bình hàng tháng của cư dân khu vực
- **Số đối thủ cạnh tranh**: Số cửa hàng hoặc doanh nghiệp tương tự trong khu vực

Mục tiêu là sử dụng K-Means để phân cụm các khu vực thành các nhóm, ví dụ: khu vực tiềm năng cao (đông dân, thu nhập cao, ít đối thủ), khu vực trung bình, hoặc khu vực ít tiềm năng. Kết quả phân cụm sẽ hỗ trợ doanh nghiệp đưa ra quyết định chiến lược về vị trí mở cửa hàng, ưu tiên các khu vực có tiềm năng kinh doanh cao.

3.2 Giải quyết bài toán bằng thuật toán K-Means

Để áp dụng K-Means, dữ liệu cần được tiền xử lý để đảm bảo tính chính xác và hiệu quả của thuật toán:

- **Kiểm tra dữ liệu**: Đảm bảo không có giá trị thiếu hoặc bất thường.
- **Chuẩn hóa dữ liệu**: Sử dụng StandardScaler để đưa các đặc trưng (mật độ dân số, thu nhập bình quân, số đối thủ cạnh tranh) về cùng thang đo, tránh hiện tượng đặc trưng có giá trị lớn chi phối khoảng cách Euclidean.

3.3 Cài đặt chương trình.

3.3.1 Một số thư viện cần sử dụng.

```
import pandas as pd
import numpy as np
import tkinter as tk
from tkinter import ttk
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.io as pio
import os
import warnings
import sys
```

Hình 3.1. Một số thư viên cần sử dụng

3.3.2 Hàm Tiền xử lý và làm sạch dữ liệu

```
def load_and_clean_data(file_path: str) -> pd.DataFrame | None:

    try:
        file_extension = os.path.splitext(file_path)[1].lower()
        if file_extension == '.csv':
            data = pd.read_csv(file_path)
        elif file_extension == '.xlsx':
            data = pd.read_excel(file_path, engine='openpyxl')
        else:
            raise ValueError("Định dạng file không được hỗ trợ. Chỉ chấp nhận CSV hoặc XLSX.")

        required_columns = ['District', 'Density', 'Income', 'Competitors']
        if not all(col in data.columns for col in required_columns):
            raise ValueError("File thiếu một hoặc nhiều cột yêu cầu: District, Density, Income, Competitors")

        # Loại bỏ các hàng có giá trị thiếu
        if data[required_columns].isnull().any().any():
            data = data.dropna(subset=required_columns)
            print("Đã loại bỏ các hàng có giá trị thiếu.")

        return data[required_columns]
    except Exception as e:
        print(f"Lỗi khi tải dữ liệu: {e}")
        return None
```

Hình 3.2. Hàm Tiền xử lý và làm sạch dữ liệu

3.3.3 Hàm Áp dụng thuật toán K-Means cho dữ liệu

```python
def apply_kmeans(data: pd.DataFrame, n_clusters: int = 3) -> tuple[pd.DataFrame, StandardScaler, KMeans]:
    scaler = StandardScaler()
    features = ['Density', 'Income', 'Competitors']
    data_scaled = scaler.fit_transform(data[features])

    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    data['Cluster'] = kmeans.fit_predict(data_scaled)

    return data, scaler, kmeans
```

Hình 3.3. Hàm Áp dụng thuật toán K-Means cho dữ liệu

3.3.4 Hàm Đặt tên cho các cụm theo đặc trưng trung bình

```python
def name_clusters(cluster_means: pd.DataFrame) -> dict:
    cluster_names = {}
    mean_density = cluster_means['Density'].mean()
    mean_income = cluster_means['Income'].mean()
    mean_competitors = cluster_means['Competitors'].mean()

    sorted_clusters = cluster_means.sort_values(['Competitors', 'Density', 'Income']).index

    for i, cluster in enumerate(sorted_clusters):
        density = cluster_means.loc[cluster, 'Density']
        income = cluster_means.loc[cluster, 'Income']

        if i == 0:
            cluster_names[cluster] = "Khu vực tiềm năng cao"
        elif i == 1 and (density > mean_density or income > mean_income):
            cluster_names[cluster] = "Khu vực đông đúc"
        else:
            cluster_names[cluster] = "Khu vực cạnh tranh cao"

    return cluster_names
```

Hình 3.4.Hàm Đặt tên cho các cụm theo đặc trưng trung bình

3.3.5 Hàm Tính điểm độ phù hợp (score) cho một khu vực dựa trên Density, Income, Competitors.

```python
def compute_suitability_score(row: pd.Series, scaler: StandardScaler) -> float:

    features = pd.DataFrame([[row['Density'], row['Income'], row['Competitors']]],
                            columns=['Density', 'Income', 'Competitors'])
    scaled_features = scaler.transform(features)[0]
    scaled_density, scaled_income, scaled_competitors = scaled_features

    weights = {'density': 2.0, 'income': 1.5, 'competitors': 2.0}
    score = (weights['density'] * scaled_density +
             weights['income'] * scaled_income -
             weights['competitors'] * scaled_competitors)

    return score
```

Hình 3.5. Tính điểm độ phù hợp (score) cho một khu vực dựa trên Density, Income, Competitors.

3.3.6 Hàm Tính toán độ phù hợp cho một điểm dữ liệu dựa trên mật độ dân cư, thu nhập, và số đối thủ.

```python
def calculate_suitability(row: pd.Series, scaler: StandardScaler) -> str:

    score = compute_suitability_score(row, scaler)

    if score > 1.0:
        return "Rất phù hợp"
    elif score > 0.0:
        return "Phù hợp"
    elif score > -1.0:
        return "Ít phù hợp"
    return "Không phù hợp"
```

Hình 3.6. Tính toán độ phù hợp cho một điểm dữ liệu dựa trên mật độ dân cư, thu nhập, và số đối thủ.

3.3.7 Hàm Vẽ biểu đồ Elbow để tìm số cụm tối ưu.

```python
def plot_elbow_method(data: pd.DataFrame, max_k: int = 10) -> None:
    """Vẽ biểu đồ Elbow để tìm số cụm tối ưu.

    Args:
        data (pd.DataFrame): DataFrame chứa dữ liệu đầu vào.
        max_k (int): Số cụm tối đa để kiểm tra. Mặc định là 10.
    """
    scaler = StandardScaler()
    data_scaled = scaler.fit_transform(data[['Density', 'Income', 'Competitors']])

    # Tính inertia cho các giá trị k từ 1 đến max_k
    inertia_values = [KMeans(n_clusters=k, random_state=42).fit(data_scaled).inertia_
                      for k in range(1, max_k + 1)]

    # Vẽ biểu đồ Elbow
    plt.figure(figsize=(8, 6))
    plt.plot(range(1, max_k + 1), inertia_values, marker='o')
    plt.title('Elbow Method For Optimal k')
    plt.xlabel('Số lượng cụm (k)')
    plt.ylabel('Inertia')
    plt.xticks(range(1, max_k + 1))
    plt.grid(True)
    #lưu biểu đồ vào file
    plt.savefig('elbow_plot.png')
    plt.show()
    plt.close()
```

Hình 3.7. Vẽ biểu đồ Elbow để tìm số cụm tối ưu.

3.3.8 Hàm In thông tin chi tiết về các cụm và khu vực có điểm score cao nhất trong từng cụm.

```python
def print_cluster_information(data: pd.DataFrame, cluster_names: dict, scaler: StandardScaler) -> None:

    print("\nThông tin các cụm và khu vực có điểm score cao nhất:")
    for cluster in sorted(cluster_names.keys()):
        cluster_data = data[data['Cluster'] == cluster].copy()
        if cluster_data.empty:
            continue

        # Tính score cho từng khu vực trong cụm
        cluster_data['Score'] = cluster_data.apply(lambda row: compute_suitability_score(row, scaler), axis=1)

        # Chọn khu vực có score cao nhất
        best_district_in_cluster = cluster_data.loc[cluster_data['Score'].idxmax()]
        best_score = best_district_in_cluster['Score']
        suitability = calculate_suitability(best_district_in_cluster, scaler)

        print(f"Cụm {cluster}: {cluster_names[cluster]}")
        print(f"  Địa điểm có score cao nhất: {best_district_in_cluster['District']}")
        print(f"  Thông tin: Density={best_district_in_cluster['Density']:.2f}, "
              f"Income={best_district_in_cluster['Income']:.2f}, "
              f"Competitors={best_district_in_cluster['Competitors']:.2f}")
        print(f"  Score: {best_score:.2f}")
        print(f"  Suitability: {suitability}\n")
```

Hình 3.8. In thông tin chi tiết về các cụm và khu vực có điểm score cao nhất trong từng cụm.

3.3.9 Hàm Vẽ biểu đồ phân tán 3D của các cụm sử dụng Plotly.

```
def plot_clusters_3d(data: pd.DataFrame, cluster_names: dict, scaler: StandardScaler) -> None:
    traces = []
    for cluster in sorted(data['Cluster'].unique()):
        cluster_data = data[data['Cluster'] == cluster]
        suitabilities = [calculate_suitability(row, scaler) for _, row in cluster_data.iterrows()]

        trace (variable) cluster_data: Any
            x=cluster_data['Density'],
            y=cluster_data['Income'],
            z=cluster_data['Competitors'],
            mode='markers',
            name=cluster_names[cluster],
            text=cluster_data['District'],
            customdata=suitabilities,
            hovertemplate=(
                'District: %{text}<br>'
                'Density: %{x}<br>'
                'Income: %{y}<br>'
                'Competitors: %{z}<br>'
                'Suitability: %{customdata}<br>'
            ),
            marker=dict(size=8)
        )
        traces.append(trace)
    layout = go.Layout(
        title='Phân cụm các khu vực theo đặc trưng',
        scene=dict(xaxis_title='Mật độ dân cư', yaxis_title='Mức thu nhập', zaxis_title='Số đối thủ'),
        legend=dict(title=dict(text='Cụm')),
```

Hình 3.9. Vẽ biểu đồ phân tán 3D của các cụm sử dụng Plotly.

3.3.10 Hàm Hiển thị bảng kết quả phân cụm trong GUI sử dụng Tkinter.

```
def show_in_gui(data: pd.DataFrame, cluster_names: dict, scaler: StandardScaler) -> None:

    root = tk.Tk()
    root.title("Kết Quả Phân Cụm")
    root.geometry("1000x400")

    columns = ("District", "Density", "Income", "Competitors", "Cluster", "Suitability")
    tree = ttk.Treeview(root, columns=columns, show="headings")

    for col in columns:
        tree.heading(col, text=col)
        tree.column(col, width=150, anchor='center')

    for _, row in data.iterrows():
        suitability = calculate_suitability(row, scaler)
        tree.insert("", "end", values=(
            row['District'],
            f"{row['Density']:.2f}",
            f"{row['Income']:.2f}",
            f"{row['Competitors']:.2f}",
            cluster_names[row['Cluster']],
            suitability
        ))

    scrollbar = ttk.Scrollbar(root, orient='vertical', command=tree.yview)
    tree.configure(yscroll=scrollbar.set)
    scrollbar.pack(side='right', fill='y')
```

Hình 3.10. Hàm Hiển thị bảng kết quả phân cụm trong GUI sử dụng Tkinter.

3.3.11 Hàm Xuất kết quả phân cụm ra file Excel.

```
def export_to_excel(data: pd.DataFrame, cluster_names: dict, scaler: StandardScaler,
                    file_path: str = 'cluster_results.xlsx') -> None:

    df_export = data.copy()
    df_export['Cluster_Name'] = df_export['Cluster'].map(cluster_names)
    df_export['Suitability'] = df_export.apply(lambda row: calculate_suitability(row, scaler), axis=1)

    df_export.to_excel(file_path, index=False, engine='openpyxl')
    print(f"Kết quả đã được xuất ra file: {file_path}")
```

Hình 3.11. Hàm Xuất kết quả phân cụm ra file Excel.

3.3.12 Hàm Chọn cụm chứa khu vực có điểm score cao nhất.

```
def select_best_cluster(data: pd.DataFrame, cluster_names: dict, scaler: StandardScaler) -> None:

    # Tính score cho tất cả khu vực
    data_with_score = data.copy()
    data_with_score['Score'] = data_with_score.apply(lambda row: compute_suitability_score(row, scaler), axis=1)

    # Tìm khu vực có score cao nhất
    best_row = data_with_score.loc[data_with_score['Score'].idxmax()]
    best_cluster = best_row['Cluster']
    best_district = best_row['District']
    best_score = best_row['Score']
    suitability = calculate_suitability(best_row, scaler)

    print(f"Cụm tiềm năng cao nhất: {cluster_names[best_cluster]} (Cụm {best_cluster})")
    print(f"Địa điểm có score cao nhất: {best_district}")
    print(f"  Thông tin: Density={best_row['Density']:.2f}, "
          f"Income={best_row['Income']:.2f}, "
          f"Competitors={best_row['Competitors']:.2f}")
    print(f"  Score: {best_score:.2f}")
    print(f"  Suitability: {suitability}")
```

Hình 3.12.Hàm Chọn cụm chứa khu vực có điểm score cao nhất.

3.3.13 Hàm Main

```
def main(file_path: str = 'TestData.xlsx') -> None:
    # Tải và làm sạch dữ liệu
    data = load_and_clean_data(file_path)
    if data is None:
        return

    # Áp dụng k-Means
    n_clusters = 3
    data, scaler, kmeans = apply_kmeans(data, n_clusters)

    # Tính trung bình đặc trưng cho các cụm và đặt tên
    cluster_means = data.groupby('Cluster')[['Density', 'Income', 'Competitors']].mean()
    cluster_names = name_clusters(cluster_means)

    # In thông tin cụm và chọn cụm tốt nhất
    print_cluster_information(
    select_best_cluster(data, cluster_names, scaler)

    # Xuất dữ liệu ra Excel
    export_to_excel(data, cluster_names, scaler)
```

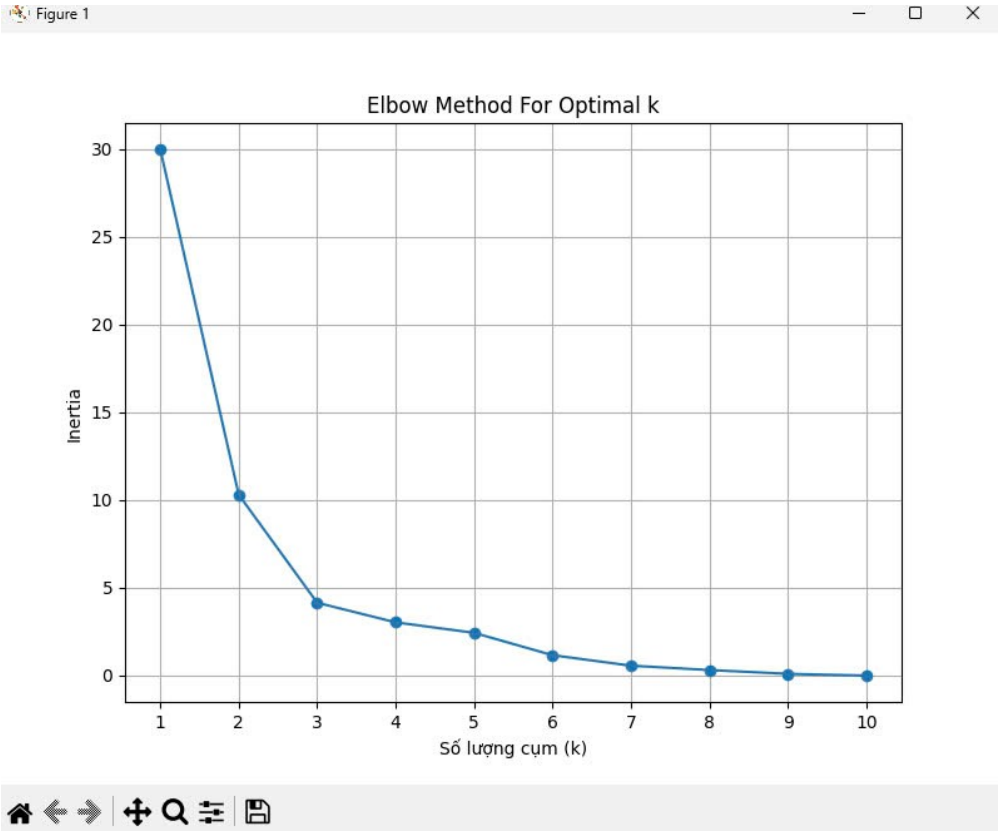
(variable) cluster_names: dict

Hình 3.13.Hàm Main-1

```
    # In thông tin cụm và chọn cụm tốt nhất
    print_cluster_information(data, cluster_names, scaler)
    select_best_cluster(data, cluster_names, scaler)

    # Xuất dữ liệu ra Excel
    export_to_excel(data, cluster_names, scaler)

    # Vẽ biểu đồ Elbow và biểu đồ 3D
    plot_elbow_method(data)
    plot_clusters_3d(data, cluster_names, scaler)

    # Hiển thị bảng kết quả trong GUI
    show_in_gui(data, cluster_names, scaler)

if __name__ == "__main__":
    # Kiểm tra xem có đường dẫn file được cung cấp qua dòng lệnh không
    file_path = 'datatest3.xlsx'
    if len(sys.argv) > 1:
        file_path = sys.argv[1]
    main(file_path)
```

Hình 3.14. Hàm Main-2

3.4 Kết quả

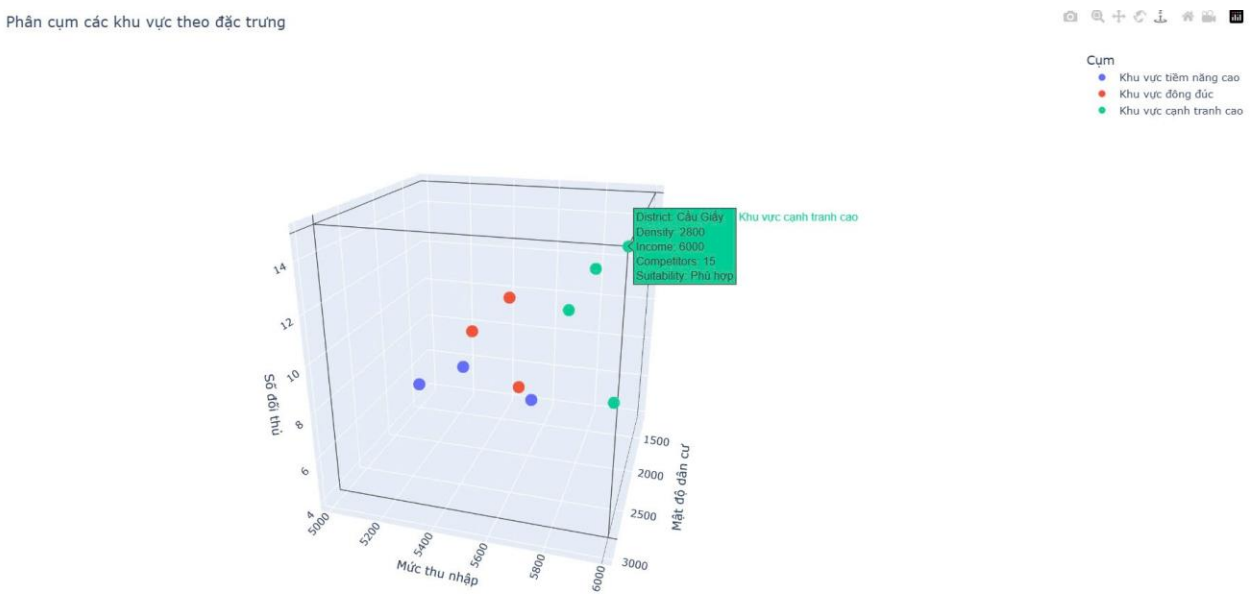


Hình 3.15. Biểu đồ cụm tối ưu.

Kết Quả Phân Cụm

District	Density	Income	Competitors	Cluster	Suitability
Hoan Kiem	25000.00	800.00	20.00	Khu vực cạnh tranh cao	Phù hợp
Ba Dinh	20000.00	750.00	15.00	Khu vực cạnh tranh cao	Phù hợp
Dong Da	45760.00	700.00	18.00	Khu vực đông đúc	Rất phù hợp
Hai Ba Trung	30000.00	720.00	16.00	Khu vực đông đúc	Rất phù hợp
Cau Giay	15783.00	680.00	14.00	Khu vực đông đúc	Không phù hợp
Thanh Xuan	30500.00	650.00	12.00	Khu vực đông đúc	Rất phù hợp
Hoang Mai	8000.00	600.00	8.00	Khu vực tiềm năng cao	Không phù hợp
Long Bien	6000.00	620.00	7.00	Khu vực tiềm năng cao	Không phù hợp
Nam Tu Liem	12000.00	670.00	10.00	Khu vực tiềm năng cao	Ít phù hợp
Tay Ho	15000.00	740.00	13.00	Khu vực đông đúc	Phù hợp
Bac Tu Liem	15000.00	680.00	11.00	Khu vực đông đúc	Phù hợp
Ha Dong	12000.00	650.00	12.00	Khu vực tiềm năng cao	Không phù hợp
Gia Lam	5000.00	600.00	6.00	Khu vực tiềm năng cao	Không phù hợp
Dong Anh	4500.00	590.00	5.00	Khu vực tiềm năng cao	Không phù hợp
Thanh Tri	6000.00	610.00	7.00	Khu vực tiềm năng cao	Không phù hợp

Hình 3.16. Bảng kết quả



Hình 3.17.Biểu dồ 3D

```
Thông tin các cụm và khu vực có điểm score cao nhất:
Cụm 0: Khu vực đông đúc
  Địa điểm có score cao nhất: Dong Da
  Thông tin: Density=45760.00, Income=700.00, Competitors=18.00
  Score: 2.90
  Suitability: Rất phù hợp

Cụm 1: Khu vực tiềm năng cao
  Địa điểm có score cao nhất: Nam Tu Liem
  Thông tin: Density=12000.00, Income=670.00, Competitors=10.00
  Score: -0.11
  Suitability: Ít phù hợp

Cụm 2: Khu vực cạnh tranh cao
  Địa điểm có score cao nhất: Ba Dinh
  Thông tin: Density=20000.00, Income=750.00, Competitors=15.00
  Score: 0.98
  Suitability: Phù hợp

Cụm tiềm năng cao nhất: Khu vực đông đúc (Cụm 0)
Địa điểm có score cao nhất: Dong Da
  Thông tin: Density=45760.00, Income=700.00, Competitors=18.00
  Score: 2.90
  Suitability: Rất phù hợp
Kết quả đã được xuất ra file: cluster_results.xlsx
```

Hình 3.18. Thông tin các cụm

KẾT LUẬN

Đề tài đã áp dụng thuật toán K-Means để phân cụm các khu vực, hỗ trợ xác định vị trí tiềm năng mở cửa hàng mới trong lĩnh vực bán lẻ, mang lại những kết quả đáng ghi nhận. K-Means đã phân chia các khu vực thành các cụm dựa trên các đặc trưng kinh tế-xã hội như mật độ dân số, thu nhập bình quân, và số đối thủ cạnh tranh, tạo ra các nhóm từ khu vực tiềm năng cao với đông dân, thu nhập cao, ít đối thủ, đến khu vực ít tiềm năng với mật độ dân số thấp và cạnh tranh cao. Kết quả phân cụm cung cấp cơ sở khoa học để doanh nghiệp lựa chọn vị trí mở cửa hàng và xây dựng chiến lược kinh doanh phù hợp với từng phân khúc thị trường.

Việc tiền xử lý dữ liệu, bao gồm chuẩn hóa và giảm chiều bằng PCA, cùng với kỹ thuật K-Means++ để khởi tạo tâm cụm, đã nâng cao hiệu quả của mô hình. Biểu đồ trực quan hóa cho thấy các cụm được tách biệt rõ ràng, chứng tỏ khả năng của K-Means trong việc xử lý dữ liệu phức tạp. Thử nghiệm với dữ liệu mới khẳng định tính tổng quát, khi các khu vực mới được phân cụm chính xác, đảm bảo tính ứng dụng thực tiễn.

Tuy nhiên, dữ liệu giả lập chưa phản ánh đầy đủ các yếu tố thực tế như hạ tầng giao thông hay hành vi tiêu dùng. Việc chọn số cụm cố định cũng cần tối ưu hóa bằng các phương pháp như Elbow hoặc Silhouette. Trong tương lai, nghiên cứu có thể thu thập dữ liệu thực tế, bổ sung đặc trưng như lưu lượng giao thông, và thử nghiệm các thuật toán phân cụm khác để tăng độ chính xác. Đề tài đã chứng minh tiềm năng của K-Means trong kinh doanh, đặt nền tảng cho các nghiên cứu tiếp theo về ứng dụng học máy, hỗ trợ doanh nghiệp tối ưu hóa chiến lược và đạt được thành công.

TÀI LIỆU THAM KHẢO

[1].VŨ VĂN TIỆP.(2020). *Machine Learning cơ bản*.
[2]. Aurélien Géron.(2017) . Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
[3]. https://scikit-learn.org/stable/modules/clustering.html#k-means