TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KÌ

Môn học: KHAI PHÁ DỮ LIỆU

Đề tài:

SỬ DỤNG K-MEAN CLUSTERING TRONG VIỆC TẠO CÁC PHÂN KHÚC KHÁCH HÀNG CHO WEBSITE THƯƠNG MẠI ĐIỆN TỬ

Sinh Viên: Dư Hoàng An

MSSV: 3119411001

GVHD: TS.Trịnh Tấn Đạt

TP. Hồ Chí Minh, tháng 5 năm 2023

MỤC LỤC

BẢNG KÍ HIỆU VIẾT TẮT

Kí hiệu	Từ đầy đủ
TMĐT	Thương mại điện tử
CSDL	Cơ sở dữ liệu

CÁC BIỂU ĐỒ, HÌNH ẢNH SỬ DỤNG

LỜI MỞ ĐẦU

Trong thời đại công nghệ số ngày nay, các website thương mại điện tử đang ngày càng trở nên phổ biến và cạnh tranh dữ dội. Để tăng cường khả năng cạnh tranh và thu hút khách hàng, các website thương mại điện tử cần phải hiểu rõ nhu cầu và sở thích của khách hàng. Điều này đòi hỏi một công cụ giúp việc phân loại khách hàng một cách chính xác và hiệu quả để giúp các website thương mai điện tử nắm bắt .

Thuật toán K-Mean Clustering là một công cụ hữu ích để giải quyết vấn đề này. Nó hoạt động bằng cách phân chia dữ liệu thành các nhóm dựa trên các đặc điểm tương đồng. Ví dụ, chúng ta có thể sử dụng thuật toán K-Mean Clustering để phân loại khách hàng dựa trên lịch sử mua hàng, độ tuổi, giới tính hoặc các đặc điểm khác.

Trong tiểu luận này, chúng ta sẽ tìm hiểu chi tiết về thuật toán K-Mean Clustering và cách áp dụng nó để phân loại khách hàng cho một website thương mại điện tử. Chúng ta sẽ xem xét các bước thực hiện thuật toán và các ứng dụng thực tế của nó trong việc phân loại khách hàng.

Tiểu luận được chia làm 4 phần việc

a) Tìm hiểu về sự quan trọng của việc nghiên cứu phân khúc khách hàng đối với website TMĐT.

b) Thu thập dữ liệu

c) Tìm hiểu bài toán phân cụm trong khai phá dữ liệu, lựa chọn thuật toán phù hợp với yêu cầu bài toán và tập dữ liệu

d) Thực hiện chương trình và ghi nhận kết quả.

Chương 1: TỔNG QUAN ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại công nghệ số hiện nay, việc kinh doanh trực tuyến ngày càng phát triển và trở nên phổ biến. Tuy nhiên, để có thể thành công trong lĩnh vực này, các doanh nghiệp cần phải hiểu rõ về khách hàng của mình và đưa ra các chiến lược marketing phù hợp.

Phân loại khách hàng là một trong những cách giúp doanh nghiệp hiểu rõ hơn về khách hàng và đưa ra các chiến lược marketing phù hợp. Tuy nhiên, việc phân loại khách hàng thủ công có thể tốn nhiều thời gian và công sức.

Do đó, việc áp dụng thuật toán K-Mean Clustering để phân loại khách hàng cho website thương mại điện tử là một giải pháp hữu ích. Thuật toán này hoạt động bằng cách chia dữ liệu thành K cụm sao cho các đối tượng trong cùng một cụm có các đặc điểm tương tự nhau. Việc áp dụng thuật toán này giúp doanh nghiệp có thể đưa ra các sản phẩm, dịch vụ phù hợp với từng nhóm khách hàng và tăng cường hiệu quả trong việc thu hút và giữ chân khách hàng.

1.2 Đề xuất

Thuật toán K-Mean Clustering là một thuật toán phân cụm dữ liệu không giám sát. Nó hoạt động bằng cách chia dữ liệu thành K cụm sao cho các đối tượng trong cùng một cụm có các đặc điểm tương tự nhau.

Để áp dụng thuật toán này để phân loại khách hàng cho website thương mại điện tử, chúng ta có thể tiến hành các bước sau:

1. Chọn số lượng cụm (K) mong muốn. Số lượng cụm này sẽ ảnh hưởng đến kết quả phân loại khách hàng. Nếu K quá nhỏ, các cụm sẽ không thể phản ánh đúng đặc điểm của khách hàng. Ngược lại, nếu K quá lớn, các cụm sẽ trở nên quá nhỏ và không có ý nghĩa trong việc phân loại khách hàng.

2. Khởi tạo ngẫu nhiên K tâm cụm (centroid). Các tâm cụm này sẽ được sử dụng để tính khoảng cách giữa các đối tượng và gán mỗi đối tượng vào cụm gần nhất.
3. Tính khoảng cách giữa các đối tượng (khách hàng) đến K tâm cụm và gán mỗi đối tượng vào cụm gần nhất. Có nhiều cách để tính khoảng cách giữa các đối tượng và tâm cụm, ví dụ như khoảng cách Euclid hoặc khoảng cách Manhattan.
4. Tính lại tọa độ của các tâm cụm dựa trên các đối tượng thuộc cụm đó. Các tâm cụm mới này sẽ được tính bằng trung bình của các đối tượng thuộc cùng một cụm.
5. Lặp lại bước 3 và 4 cho đến khi các tâm cụm không thay đổi nữa.

Sau khi hoàn thành các bước trên, bạn sẽ có được K cụm khách hàng với các đặc điểm tương tự nhau. Bạn có thể sử dụng kết quả này để đưa ra các sản phẩm, dịch vụ phù hợp với từng nhóm khách hàng và tăng cường hiệu quả trong việc thu hút và giữ chân khách hàng.

Đề tài được chia thành các phần:

Chương 1: Tổng quan đề tài

Chương 2: Khai phá dữ liệu

Chương 3: Kỹ thuật phân cụm và sử dụng thuật toán K-means

Chương 4: Thực nghiệm và đánh giá

CHƯƠNG 2: KHAI PHÁ DỮ LIỆU

2.1. Khái niệm về khai phá dữ liệu

Khai phá dữ liệu có thể có nhiều các để định nghĩa. Một cách ta có thể hiểu đó là ”các tri thức được khai phá hay khai thác từ dữ liệu (data)” . Hiểu theo 1 cách ngắn gọn hơn, khai phá dữ liệu có thể không nói đến việc khai phá từ 1 lượng lớn dữ liệu. “Khai phá” là 1 khái niệm đặc trưng cho quá trình tìm kiếm một lượng dữ liệu cần thiết trong 1 khối lượng dữ liệu thô lớn. Ngoài ra còn có các khái niệm khá tương đồng với khai phá dữ liệu như: “khai phá tri thức từ dữ liệu”, “trích xuất tri thức”, “phân tích dữ liệu”,…

Nhiều người cho rằng khái niệm khai phá dữ liệu tương đồng với khái niệm:

Tìm kiếm tri thức từ dữ liệu, trong khi 1 số khác cho rằng khai phá dữ liệu là 1 bước trong quá trình tim kiếm tri thức từ dữ liệu.

Khai phá dữ liệu là một bước của quá trình khai thác tri thức (Knowledge Discovery Process), bao gồm:

• Xác định vấn đề và không gian dữ liệu để giải quyết vấn đề (Problem understanding and data understanding).

• Chuẩn bị dữ liệu (Data preparation), bao gồm các quá trình làm sạch dữ liệu (data cleaning), tích hợp dữ liệu (data integration), chọn dữ liệu (data selection), biến đổi dữ liệu (data transformation).

• Khai thác dữ liệu (Data mining): xác định nhiệm vụ khai thác dữ liệu và lựa chọn kỹ thuật khai thác dữ liệu. Kết quả cho ta một nguồn tri thức thô.

• Đánh giá (Evaluation): dựa trên một số tiêu chí tiến hành kiểm tra và lọc nguồn tri thức thu được.

• Triển khai (Deployment).

Quá trình khai thác tri thức không chỉ là một quá trình tuần tự từ bước đầu tiên đến bước cuối cùng mà là một quá trình lặp và có quay trở lại các bước đã qua.

Các loại dữ liệu có thể khai phá:

1. Dữ liệu từ cơ sở dữ liệu

2. Dữ liệu từ kho chứa dữ liệu
3. Dữ liệu giao dịch
4. Các loại khác

2.2 Ứng dụng của khai phá dữ liệu

Khai phá dữ liệu giúp con người xử lý và sắp xếp các thông tin một cách gọn nhẹ và dễ dàng. Ứng dụng của khai phá dữ liệu có thể thấy trong 7 lĩnh vực sau:

1. Kinh doanh: Khai phá dữ liệu giúp doanh nghiệp xác định được các mẫu dữ liệu và dùng các thuật toán để phân tích từ đó đưa ra dự báo về xu hướng, hành vi người tiêu dùng.
2. Viễn thông: Khai phá dữ liệu giúp các nhà cung cấp viễn thông phân tích dữ liệu khách hàng để cải thiện chất lượng dịch vụ và tăng doanh thu.
3. Ngân hàng: Khai phá dữ liệu giúp các ngân hàng phát hiện gian lận, quản lý rủi ro tín dụng và phân tích hành vi khách hàng.
4. Thương mại điện tử và bán lẻ: Khai phá dữ liệu giúp các nhà bán lẻ và thương mại điện tử phân tích dữ liệu khách hàng để cải thiện trải nghiệm mua sắm và tăng doanh thu.
5. Tài chính: Khai phá dữ liệu giúp các nhà đầu tư và quản lý tài chính phân tích dữ liệu thị trường để đưa ra quyết định đầu tư thông minh hơn.
6. Y tế và chăm sóc sức khỏe: Khai phá dữ liệu giúp các nhà chăm sóc sức khỏe phân tích dữ liệu bệnh nhân để đưa ra chẩn đoán và điều trị tốt hơn.
7. An ninh, bảo mật mạng: Khai phá dữ liệu giúp các nhà an ninh mạng phát hiện và ngăn chặn các cuộc tấn công mạng.

2.3 Mô hình học không giám sát (Unsupervised learning).

Học không giám sát là một loại học máy giúp tìm ra các mô hình trong dữ liệu mà không cần phân loại hoặc gán nhãn trước. Điều này có nghĩa là thuật toán không được cung cấp bất kỳ thông tin nào về dữ liệu đại diện cho điều gì hoặc cách phân nhóm chúng. Thay vào đó, thuật toán sẽ tự tìm ra các mối quan hệ và cấu trúc trong dữ liệu.

Hai kỹ thuật được sử dụng trong học không giám sát, bao gồm:

- **Clustering (phân cụm)**

Là bài toán phân cụm X toàn bộ dữ liệu thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng giống nhau giúp công ty hiểu rõ hơn về khách hàng và phát triển các chiến lược tiếp thị phù hợp hơn.

- **Association**

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider-Man thường có xu hướng xem thêm phim Batman, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

Học không giám sát có thể hữu ích trong nhiều ứng dụng, bao gồm khám phá dữ liệu, phát hiện bất thường và hệ thống đề xuất. Nó cũng có thể được sử dụng để tiền xử lý dữ liệu cho các tác vụ học máy khác, chẳng hạn như học có giám sát.

Một số thuật toán thông dụng được sử dụng trong học không giám sát bao gồm phân cụm k-means, phân cụm phân cấp và phân tích thành phần chính (PCA). Các thuật toán này hoạt động bằng cách tìm ra các mô hình trong dữ liệu và nhóm các điểm dữ liệu tương tự lại với nhau hoặc giảm chiều của dữ liệu.

2.4 Kĩ thuật phân cụm (Clustering)

2.4.1 Định nghĩa

Kĩ thuật phân tích cụm (hay phân cụm) là quá trình phân vùng dữ liệu trong 1 tập dữ liệu lớn thành các tập dữ liệu con. Các tập dữ liệu con là 1 cụm với các dữ liệu có điểm tương đồng với nhau bên trong.

Với hai mục đích chính của khai phá dữ liệu là: dự đoán (Prediction), người ta thường sử dụng các phương pháp sau cho khai phá dữ liệu:

- Phân lớp (Classification)
- Hồi quy (Regression)
- Trực quan hóa (Visualiztion)

- Phân cụm (Clustering)
- Tổng hợp (Summarization)
- Tổng hợp ràng buộc (Dependency modeling)
- Biểu diễn mô hình (Model Evaluation)
- Phân tích sự phát triển và độ lệch (Evolution and deviation analyst)
- Luật kết hợp (Associantion rules)
- Phương pháp tìm kiếm (Search Method)

Một vài ví dụ về ý nghĩa thực tiễn của phân cụm dữ liệu như sau :

- Khám phá ra các vị trí địa lý thuận lợi cho việc xây dựng các kho hàng phục vụ mua bàn hàng của một công ty thương mại
- Xác định các cụm ảnh như ảnh của các loài động vật như loài thú, chim,... trong tập CSDL ảnh về động vật nhằm phục vụ cho việc tìm kiếm ảnh
- Xác định các nhóm người bệnh nhằm cung cấp thông tin cho việc phân phối các thuốc điều trị trong y tế
- Xác định nhóm các khách hàng trong CSDL ngân hàng có vốn các đầu tư vào bất động sản cao...

Như vậy, phân cụm dữ liệu là một phương pháp xử lý thông tin quan trọng và phổ biến, nó nhằm khám phá mối liên hệ giữa các mẫu dữ liệu bằng cách tổ chức chúng thành các cụm tương tự.

2.4.2 Các bước phân cụm

Các bước cụ thể của phân cụm sẽ phụ thuộc vào thuật toán phân cụm cụ thể mà bạn sử dụng. Tuy nhiên, có một số bước chung mà hầu hết các thuật toán phân cụm đều thực hiện:

1. **Chuẩn bị dữ liệu**: Trước khi bắt đầu phân cụm, bạn cần chuẩn bị dữ liệu của mình. Điều này có thể bao gồm việc loại bỏ các giá trị bị thiếu hoặc ngoại lệ, chuyển đổi dữ liệu về dạng chuẩn và chọn các đặc trưng phù hợp để sử dụng trong phân cụm.

2. **Chọn thuật toán phân cụm**: Có nhiều thuật toán phân cụm khác nhau và bạn cần chọn một thuật toán phù hợp với dữ liệu và mục tiêu của mình. Một số thuật toán phân cụm phổ biến bao gồm k-means, phân cụm phân cấp và DBSCAN.
3. **Thiết lập các tham số**: Hầu hết các thuật toán phân cụm đều có một số tham số mà bạn cần thiết lập trước khi chạy thuật toán. Ví dụ, với thuật toán k-means, bạn cần xác định số lượng nhóm k.
4. **Chạy thuật toán phân cụm**: Sau khi đã chuẩn bị dữ liệu, chọn thuật toán và thiết lập các tham số, bạn có thể chạy thuật toán phân cụm trên dữ liệu của mình. Thuật toán sẽ tìm ra các nhóm trong dữ liệu và gán nhãn cho từng điểm dữ liệu.
5. **Đánh giá kết quả**: Sau khi đã chạy xong thuật toán phân cụm, bạn nên đánh giá kết quả để xem liệu các nhóm có hợp lý hay không. Có một số chỉ số đánh giá khác nhau mà bạn có thể sử dụng để đo lường chất lượng của các nhóm.

2.4.3 Ứng dụng

Phân cụm dữ liệu có rất nhiều ứng dụng trong các lĩnh vực khác nhau:

- Thương mại: Giúp các doanh nhân khám phá ra các nhóm khách hàng quan trọng để đưa ra các mục tiêu tiếp thị.
- Sinh học: Xác định các loài sinh vật, phân loại các Gen với chức năng tương đồng và thu được cấu trúc trong các mẫu.
- Lập quy hoạch đô thị: Nhận dạng các nhóm nhà theo kiểu và vị trí địa lý, nhằm cung cấp thông tin cho quy hoạch đô thị.
- Thư viện: Phân loại các cụm sách có nội dung và ý nghĩa tương đồng nhau để cung cấp cho độc giả.
- Bảo hiểm: Nhận dạng nhóm tham gia bảo hiểm có chi phí bồi thường cao, nhận dạng gian lận thương mại.
- Nghiên cứu đất đai: Phân cụm để theo dõi các tâm động đất nhằm cung cấp thông tin cho nhận dạng các vùng nguy hiểm.
- World Wide Web: Có thể khám phá các nhóm tài liệu quan trọng, có nhiều ý nghĩa trong môi trường web. Các lớp tài liệu này trợ giúp cho việc khai phá dữ liệu từ dữ liệu.

Chương 3: ỨNG DỤNG THUẬT TOÁN K-MEAN TRONG PHÂN LOẠI NGƯỜI DÙNG

3.1 Thuật toán phân cụm K-Mean (K-Mean Clustering)

K-means là một trong những thuật toán học tập không giám sát đơn giản nhất giải quyết vấn đề phân cụm nổi tiếng. Quy trình tuân theo một cách đơn giản và dễ dàng để phân loại một tập dữ liệu nhất định thông qua một số cụm nhất định (giả sử k cụm) cho trước cố định. Ý tưởng chính là xác định số cụm k, một trung tâm cho mỗi cụm. Các trung tâm này nên được đặt một cách khéo léo vì vị trí khác nhau gây ra kết quả khác nhau. Vì vậy, lựa chọn tốt hơn là đặt chúng càng xa nhau càng tốt. Bước tiếp theo là lấy từng điểm thuộc một tập dữ liệu nhất định và liên kết nó với trung tâm gần nhất. Khi không có điểm nào đang chờ xử lý, bước đầu tiên được hoàn thành và một nhóm tuổi sớm được thực hiện. Tại thời điểm này, chúng ta cần tính lại k trọng tâm mới là trọng tâm của các cụm từ bước trước. Sau khi chúng ta có k trọng tâm mới này, một liên kết mới phải được thực hiện giữa các điểm tập dữ liệu giống nhau và trung tâm mới gần nhất. Một vòng lặp đã được tạo ra. Kết quả của vòng lặp này, chúng ta có thể nhận thấy rằng k tâm thay đổi vị trí của chúng từng bước một cho đến khi không còn sự thay đổi nào nữa hay nói cách khác là các tâm không di chuyển nữa. Cuối cùng, thuật toán này nhằm mục đích giảm thiểu một hàm mục tiêu được gọi là sai số toàn phương trung bình được đưa ra bởi:

$$J(V) = \sum_{c=1}^{C} \sum_{j=1}^{c_i} \left(\|x_i - v_j\| \right)^2$$

Với:

$\left(\|x_i - v_j\| \right)$ là khoảng cách Euclide giữa x_i và v_j

'c_i' là số điểm dữ liệu ở phân cụm thứ i

'c' là số lượng phân cụm

Input: số k là số các cụm muốn phân loại

Tập dữ liệu D với n thực thể (objects)

Output: Tập hợp các cụm k với các thực thể có các điểm tương đồng với nhau

Các bước thực hiện thuật toán:

Bước 1: Chọn ngẫu nhiên K tâm (centroid) cho K cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm.

Bước 2: Tính khoảng cách giữa các đối tượng (objects) đến K tâm (thường dùng khoảng cách Euclidean)

Bước 3: Nhóm các đối tượng vào nhóm gần nhất.

Bước 4: Xác định lại tâm mới cho các nhóm.

Bước 5: Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng.

Ưu điểm của thuật toán:

- Nhanh, mạnh, dễ tìm hiểu
- Tương đối hiệu quả
- Cho kết quả tốt nhất khi mà tập dữ liệu đã được phân biệt hay được tách bạch khỏi các dữ liệu khác

Nhược điểm của thuật toán:

- Yêu cầu đặc tả trước về số trung tâm cụm (k).
- Chỉ dùng 1 chỉ định nhóm k duy nhất, nếu không sẽ gây Overlapping
- Các phép đo khoảng cách Euclide có thể ảnh hưởng không đồng đều đến các yếu tố cơ bản
- Không thể xử lý dữ liệu lỗi, nhiễu hoặc ngoại vi
- Việc chọn cụm trung tâm 1 cách ngẫu nhiên dẫn đến kết quả không phải là tốt nhất

3.2 Thuật toán K-mean trong phân loại người dùng

3.2.1 Phân khúc khách hàng trong thương mại điện tử

Phân khúc khách hàng hay còn gọi là **phân khúc thị trường**, là một chiến lược tiếp thị được thực hiện dựa trên việc xác định và phân chia các đối tượng mục tiêu thành các nhóm nhỏ nhằm cung cấp các nội dung, thông điệp bán hàng phù hợp nhất.

Phân khúc khách hàng thực chất là việc làm sáng tỏ các vấn đề sau đây khi bắt đầu

một chiến lược kinh doanh:

- Nhóm khách hàng chính mà doanh nghiệp tập trung tiếp thị là ai?
- Nhóm khách hàng nào có khả năng sinh lợi nhiều nhất và ít nhất?
- Sản phẩm/ dịch vụ của mình có điểm mạnh vượt trội nào thu hút khách hàng?
- Nhu cầu sử dụng của khách hàng là gì?
- Làm sao để các sản phẩm/ dịch vụ giải quyết được "nỗi đau" mà khách hàng đang gặp phải?
- Làm sao để cải thiện các dịch vụ để làm hài lòng khách hàng hơn?
- Các kênh tiếp thị tốt nhất là gì?
- Các kênh bán hàng hiện tại có hiệu quả không?

3.2.2 Tầm quan trọng của phân khúc khách hàng trong thương mại điện tử

Phân khúc khách hàng sẽ giúp hoạt động tiếp thị của doanh nghiệp được cá nhân hóa hơn, từ đó chiến lược kinh doanh có thể dễ dàng vạch ra và quản lý hơn. Tầm quan trọng của phân khúc khách hàng thể hiện ở các mặt lợi ích sau đây:

- Giúp cải thiện các chiến dịch tiếp thị
- Dễ dàng đề xuất các cải tiến
- Tăng khả năng mở rộng kinh doanh
- Giữ chân khách hàng
- Tối ưu về mặt giá cả
- Thúc đẩy tăng doanh thu

3.2.3 Các hình thức phân khúc khách hàng

- Phân khúc dựa trên độ tuổi.
- Phân khúc dựa trên mức thu nhập hằng tháng/hằng năm.
- Phân khúc dựa trên mức chi tiêu kì vọng.
- Phân khúc dựa trên vị trí địa lý.
- Phân khúc dựa trên mật độ dân số.
- Phân khúc dựa trên thói quen mua hàng.

3.5 Áp dụng thuật toán K-Mean Clustering

3.5.1 Các bước thực hiện

- Import các thư viện
- Khám phá dataset
- Trực quan hóa dữ liệu
- Phân cụm bằng K-Means
- Biểu diễn đường biên giữa các cụm
- Mô hình hóa các cụm đã chia

3.5.2 Tập dữ liệu

Số lượng dữ liệu: 200

5 cột thuộc tính gồm: IDCustomer, Gender, Age, Annual Income, Spending Score

Kiểu dữ liệu:

CustomerID int64

Gender object

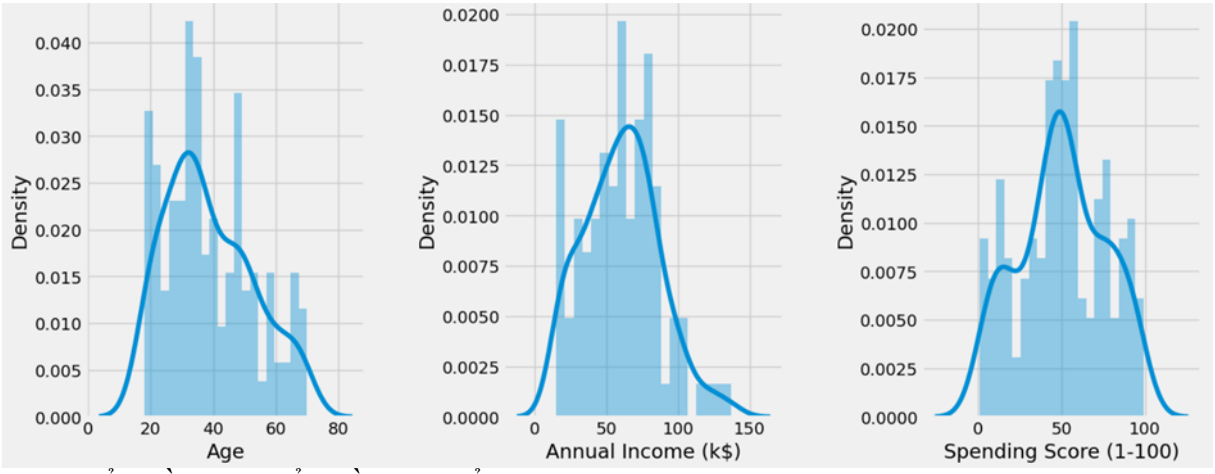
Age int64

Annual Income (k$) int64

Spending Score (1-100) int64

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Hình 3. 1: Mô tả tập dữ liệu

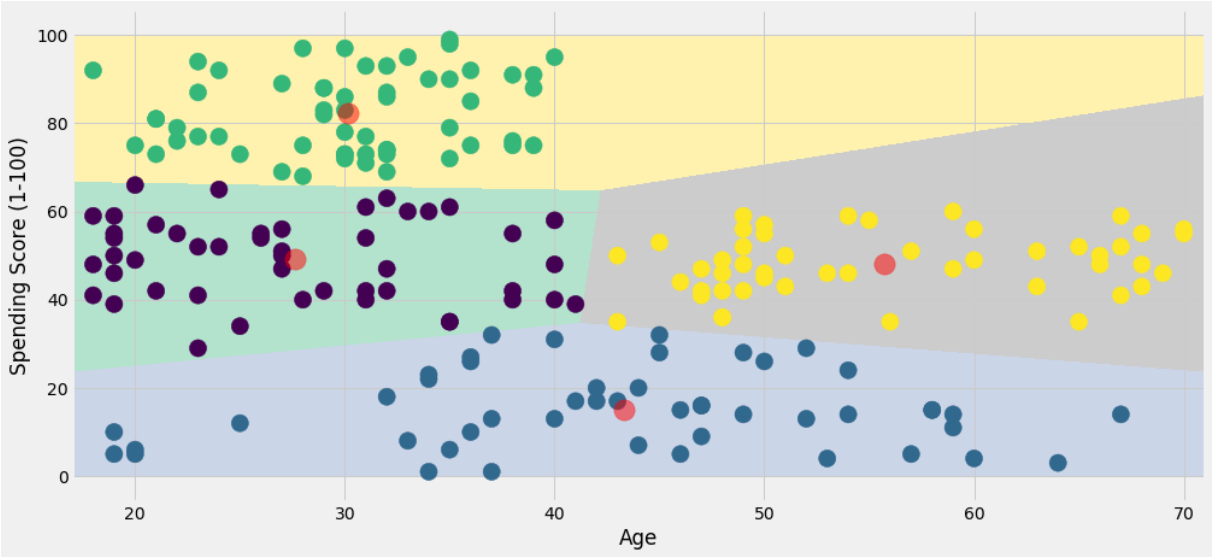


Biểu đồ 3. 1: Biểu đồ phân bổ dựa trên Age, Annual Income, Spending Score

3.5.3 Kết quả

Dựa trên tuổi (Age) và mức chi tiêu (Spending Score)

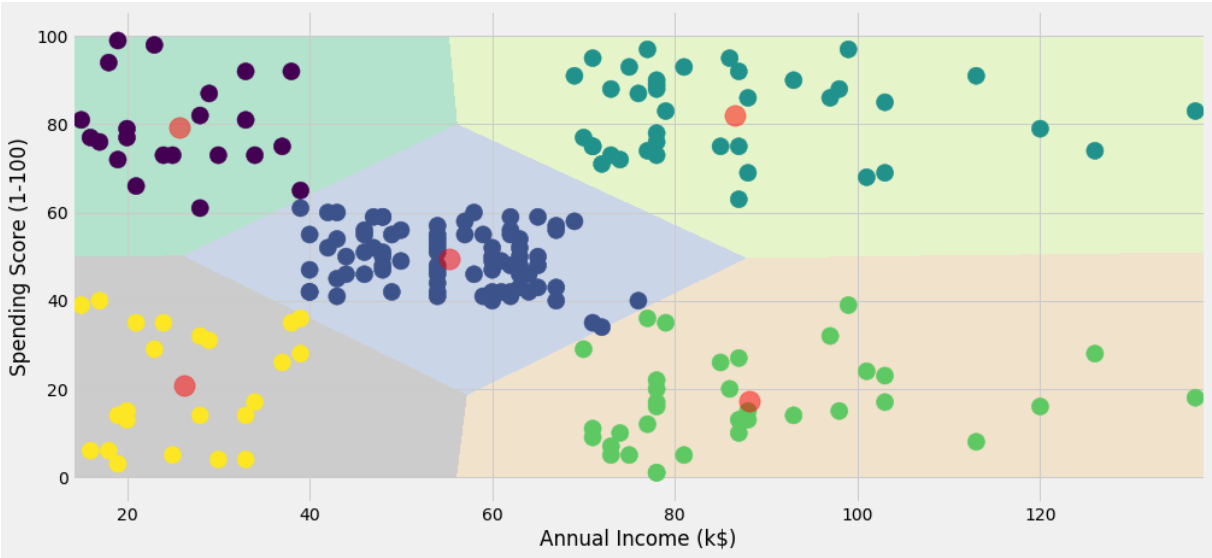
Hình 3. 2: Lựa chọn số cụm dựa trên dữ liệu lỗi của tuổi (Age) và mức chi tiêu (Spending Score)



Hình 3. 3: Phân cụm dựa trên tuổi (Age) và mức chi tiêu (Spending Score)

Dựa trên mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)

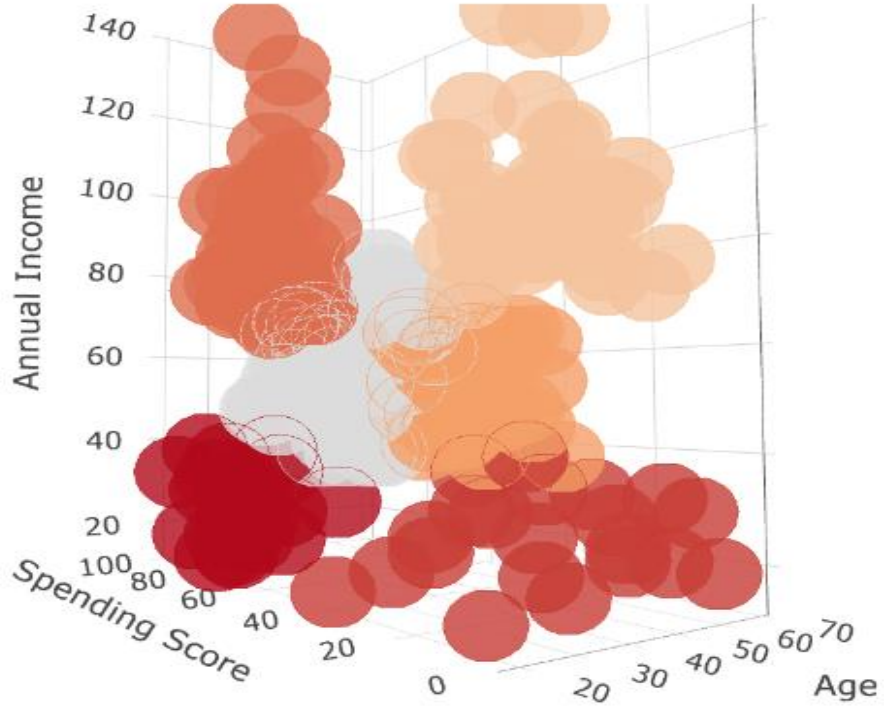
Hình 3. 4: Lựa chọn số cụm dựa trên dữ liệu lỗi của mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)



Hình 3. 5: Phân cụm dựa trên mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)

Dựa trên tuổi (Age), mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)

Hình 3. 6: Lựa chọn số cụm dựa trên dữ liệu lỗi của tuổi (Age), mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)



Hình 3. 7: Phân cụm của tuổi (Age), mức thu nhập (Annual Income) và mức chi tiêu (Spending Score)

KẾT LUẬN

Trong bài tiểu luận này, em đã trình bày cách sử dụng k-means clustering để tạo các phân khúc khách hàng cho website thương mại điện tử. Em đã thu thập sau đó áp dụng thuật toán k-means để phân cụm khách hàng theo các đặc trưng như tuổi (Age), mức thu nhập (Annual Income), mức chi tiêu (Spending Score). Em đã đề xuất các chiến lược tiếp thị phù hợp với từng phân khúc khách hàng để tăng doanh số bán hàng và khách hàng trung thành.

Bài tiểu luận của em đã cho thấy k-means clustering là một công cụ hữu ích trong việc phân tích dữ liệu khách hàng và tạo các phân khúc khách hàng có ý nghĩa. Tuy nhiên, bài tiểu luận Một số hướng phát triển có thể được thực hiện trong tương lai là:

- Sử dụng các nguồn dữ liệu khác nhau để có được thông tin về sở thích và nhu cầu của khách hàng, ví dụ như dữ liệu về lịch sử tìm kiếm, xem sản phẩm, đánh giá sản phẩm, phản hồi khách hàng, mạng xã hội, v.v.
- Áp dụng các thuật toán phân cụm khác nhau để so sánh kết quả và chọn ra thuật toán phù hợp nhất với dữ liệu và mục tiêu nghiên cứu, ví dụ như hierarchical clustering, DBSCAN, spectral clustering, v.v.

Cũng có một số hạn chế như: dữ liệu chỉ bao gồm một số đặc trưng cơ bản của khách hàng, không có thông tin về sở thích và nhu cầu của khách hàng; thuật toán k-means chỉ phù hợp với các dữ liệu có phân bố đều và không có nhiễu; số cụm được chọn có thể không phản ánh được sự đa dạng của khách hàng. Do đó, trong tương lai, em mong muốn có thể mở rộng nghiên cứu bằng cách sử dụng các nguồn dữ liệu khác nhau, áp dụng các thuật toán phân cụm khác nhau và kiểm tra hiệu quả của các chiến lược tiếp thị trên các phân khúc khách hàng.

TÀI LIỆU THAM KHẢO

Jiawei Han, Micheline Kamber, *Jian* Pei*; Data mining* : *concepts* and techniques, 3rd edition, published in 2012

Phân khúc thị trường E-Commerce B2C tại Việt Nam, https://izzi.asia/vi-VN/bai-viet/phan-khuc-thi-truong-e-commerce-b2c-tai-viet-nam, truy cập cuối 15/05/2023

K-Means Clustering in Python: A Practical Guide, K-Means Clustering in Python: A Practical Guide – Real Python, lần truy cập cuối 16/05/2023

PHỤ LỤC

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
from google.colab import drive
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os
warnings.filterwarnings("ignore")
py.offline.init_notebook_mode(connected = True)
```

```
[3] drive.mount('/content/drive')
```

```
[4] df = pd.read_csv('/content/drive/MyDrive/Mall_Customers.csv')
    df.head()
```

```
df.shape
```

```
df.describe()
```

```
df.dtypes
```

```
[ ] plt.style.use('fivethirtyeight')
```

```
df.isnull().sum()
```

```
[ ]  plt.figure(1 , figsize = (15 , 6))
     n = 0
     for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
         n += 1
         plt.subplot(1 , 3 , n)
         plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
         sns.distplot(df[x] , bins = 20)
         plt.title('Distplot of {}'.format(x))
     plt.show()
```

```
[ ]  plt.figure(1 , figsize = (15 , 5))
     sns.countplot(y = 'Gender' , data = df)
     plt.show()
```

```
[ ]  plt.figure(1 , figsize = (15 , 7))
     n = 0
     for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
         for y in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
             n += 1
             plt.subplot(3 , 3 , n)
             plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
             sns.regplot(x = x , y = y , data = df)
             plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1 else y )
     plt.show()
```

```
[ ]  plt.figure(1 , figsize = (15 , 6))
     for gender in ['Male' , 'Female']:
         plt.scatter(x = 'Age' , y = 'Annual Income (k$)' , data = df[df['Gender'] == gender] ,
                     s = 200 , alpha = 0.5 , label = gender)
     plt.xlabel('Age'), plt.ylabel('Annual Income (k$)')
     plt.title('Age vs Annual Income w.r.t Gender')
     plt.legend()
     plt.show()
```

```
[ ] plt.figure(1 , figsize = (15 , 6))
    for gender in ['Male' , 'Female']:
        plt.scatter(x = 'Annual Income (k$)',y = 'Spending Score (1-100)' ,
                    data = df[df['Gender'] == gender] ,s = 200 , alpha = 0.5 , label = gender)
    plt.xlabel('Annual Income (k$)'), plt.ylabel('Spending Score (1-100)')
    plt.title('Annual Income vs Spending Score w.r.t Gender')
    plt.legend()
    plt.show()
```

```
[ ] X1 = df[['Age' , 'Spending Score (1-100)']].iloc[: , :].values
    inertia = []
    for n in range(1 , 11):
        algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
        algorithm.fit(X1)
        inertia.append(algorithm.inertia_)
```

```
[ ] plt.figure(1 , figsize = (15 ,6))
    plt.plot(np.arange(1 , 11) , inertia , 'o')
    plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
    plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
    plt.show()
```

```
[ ] algorithm = (KMeans(n_clusters = 4 ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X1)
    labels1 = algorithm.labels_
    centroids1 = algorithm.cluster_centers_
```

```
[ ] h = 0.02
    x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
    y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
    Z = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
```

```
[ ] plt.figure(1 , figsize = (15 , 7) )
    plt.clf()
    Z = Z.reshape(xx.shape)
    plt.imshow(Z , interpolation='nearest',
               extent=(xx.min(), xx.max(), yy.min(), yy.max()),
               cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')

    plt.scatter( x = 'Age' ,y = 'Spending Score (1-100)' , data = df , c = labels1 ,
                s = 200 )
    plt.scatter(x = centroids1[: , 0] , y =  centroids1[: , 1] , s = 300 , c = 'red' , alpha = 0.5)
    plt.ylabel('Spending Score (1-100)') , plt.xlabel('Age')
    plt.show()
```

```
[ ] X2 = df[['Annual Income (k$)' , 'Spending Score (1-100)']].iloc[: , :].values
    inertia = []
    for n in range(1 , 11):
        algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
        algorithm.fit(X2)
        inertia.append(algorithm.inertia_)
```

```
    plt.figure(1 , figsize = (15 ,6))
    plt.plot(np.arange(1 , 11) , inertia , 'o')
    plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
    plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
    plt.show()
```

```
[ ] algorithm = (KMeans(n_clusters = 5 ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X2)
    labels2 = algorithm.labels_
    centroids2 = algorithm.cluster_centers_
```

```
[ ] h = 0.02
    x_min, x_max = X2[:, 0].min() - 1, X2[:, 0].max() + 1
    y_min, y_max = X2[:, 1].min() - 1, X2[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
    Z2 = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
```

```
[ ] plt.figure(1 , figsize = (15 , 7) )
    plt.clf()
    Z2 = Z2.reshape(xx.shape)
    plt.imshow(Z2 , interpolation='nearest',
               extent=(xx.min(), xx.max(), yy.min(), yy.max()),
               cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')

    plt.scatter( x = 'Annual Income (k$)' ,y = 'Spending Score (1-100)' , data = df , c = labels2 ,
                s = 200 )
    plt.scatter(x = centroids2[: , 0] , y =  centroids2[: , 1] , s = 300 , c = 'red' , alpha = 0.5)
    plt.ylabel('Spending Score (1-100)') , plt.xlabel('Annual Income (k$)')
    plt.show()
```

```
[ ] X3 = df[['Age' , 'Annual Income (k$)' ,'Spending Score (1-100)']].iloc[: , :].values
    inertia = []
    for n in range(1 , 11):
        algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
        algorithm.fit(X3)
        inertia.append(algorithm.inertia_)
```

```
[ ] plt.figure(1 , figsize = (15 ,6))
    plt.plot(np.arange(1 , 11) , inertia , 'o')
    plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
    plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
    plt.show()
```

```
[ ] algorithm = (KMeans(n_clusters = 6 ,init='k-means++', n_init = 10 ,max_iter=300,
                            tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X3)
    labels3 = algorithm.labels_
    centroids3 = algorithm.cluster_centers_
```

```
[ ] df['label3'] =  labels3
    trace1 = go.Scatter3d(
        x= df['Age'],
        y= df['Spending Score (1-100)'],
        z= df['Annual Income (k$)'],
        mode='markers',
         marker=dict(
            color = df['label3'],
            size= 20,
            line=dict(
                color= df['label3'],
                width= 12
            ),
            opacity=0.8
         )
    )
    data = [trace1]
    layout = go.Layout(
    #     margin=dict(
    #         l=0,
    #         r=0,
    #         b=0,
    #         t=0
    #     )
        title= 'Clusters',
        scene = dict(
                xaxis = dict(title  = 'Age'),
                yaxis = dict(title  = 'Spending Score'),
                zaxis = dict(title  = 'Annual Income')
            )
    )
    fig = go.Figure(data=data, layout=layout)
    py.offline.iplot(fig)
```