UEH UNIVERSITY

BÁO CÁO

Môn học: Máy Học

Giảng viên hướng dẫn: Nguyễn An Tế

Nhóm sinh viên: Nguyễn Đôn Đức
Nguyễn Trương Hoàng
Đỗ Thanh Hoa
Đặng Thị Thu Hiền
Nguyễn Huy Hoàng

Đề tài: ***Áp dụng giải thuật BANG cho bài toán phân cụm khách hàng***

TP. Hồ Chí Minh, ngày 26 tháng 11 năm 2024

THÀNH VIÊN NHÓM

STT	Họ và tên	MSSV
1	Nguyễn Đôn Đức	31221024296
2	Đặng Thị Thu Hiền	31221025556
3	Đỗ Thanh Hoa	31211025193
4	Nguyễn Huy Hoàng	31221023992
5	Nguyễn Trương Hoàng	31221025159

LỜI NÓI ĐẦU

Kính gửi quý thầy cô và các quý độc giả, đây là bài báo cáo thuộc đồ án cuối kỳ môn Máy học. Cấu trúc bài báo cáo chia làm 8 chương,

Chương 1: Chương này sẽ giúp ta đi qua về tổng quan của bài báo cáo, về mục tiêu mà nhóm muốn hướng tới cũng như phương pháp nào mà nhóm thực hiện để đạt được mục tiêu

Chương 2: Nêu ra những lý thuyết cần đến để có thể áp dụng thuật toán BANG-clustering vào bài toán phân cụm khách hàng.

Chương 3: Giới thiệu về bộ dữ liệu Cleaned Contoso.

Chương 4: Đưa ta đi qua quy trình mà nhóm sẽ xử lý dữ liệu để có thể áp dụng được với thuật toán BANG-clustering.

Chương 5: Mang đến kết quả thực nghiệm của việc áp dụng thuật toán.

Chương 6: Đánh giá kết quả thuật toán trả về dựa trên các tiêu chí nhất định.

Chương 7: Ứng dụng thuật toán vào bài toán phân cụm khách hàng để từ đó chọn ra cụm mục tiêu.

Chương 8: Tổng kết lại những gì mà nhóm đã đạt được cũng như hướng phát triển.

Toàn bộ source code của đồ án được đính kèm theo bài báo cáo và đăng tải trên GitHub. Vì gặp giới hạn về thời gian và kiến thức nên bài báo cáo không thể tránh khỏi những sai sót. Nhóm tác giả rất mong nhận được sự đóng góp ý kiến từ quý thầy cô và các quý độc giả để đồ án được hoàn thiện hơn thông qua mục Discussions.

Nhóm tác giả cũng xin chân thành cảm ơn Ts. Nguyễn An Tế đã tận tình giảng dạy, hướng dẫn để nhóm hoàn thành đồ án này.

Trân trọng,

Nhóm 03

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

1.1. Tổng quan đề tài

Trong thời đại số hóa hiện nay, thương mại điện tử đang phát triển mạnh mẽ, kéo theo sự gia tăng trong cạnh tranh giữa các doanh nghiệp. Việc hiểu rõ hành vi và nhu cầu của khách hàng trở thành một yếu tố quyết định cho sự thành công của các chiến lược kinh doanh. Đề tài này tập trung vào việc áp dụng giải thuật BANG để phân cụm khách hàng dựa trên bộ dữ liệu Cleaned Contoso. Bằng cách kết hợp các yếu tố nhân khẩu học và địa lý, nhóm sẽ sử dụng giải thuật BANG để nhận diện các nhóm khách hàng tương đồng dựa trên mật độ điểm dữ liệu trong không gian nhiều chiều, từ đó phát hiện các mẫu hành vi mua sắm đặc trưng mà các phương pháp phân cụm truyền thống có thể bỏ qua. Kết quả từ nghiên cứu này không chỉ giúp tối ưu hóa các chiến lược tiếp thị mà còn nâng cao trải nghiệm của khách hàng thông qua các dịch vụ và sản phẩm được cá nhân hóa.

1.2. Mục tiêu nghiên cứu

Tìm hiểu và nắm vững các khái niệm cơ bản liên quan đến giải thuật BANG sau đó áp dụng giải thuật BANG để phân cụm khách hàng trong bộ dữ liệu Cleaned Contoso. Cuối cùng, một trong những mục tiêu quan trọng là phân tích và trực quan hóa dữ liệu cũng như kết quả phân cụm thu được từ giải thuật BANG giúp hiểu rõ hơn về đặc điểm của từng nhóm khách hàng mà còn cung cấp cái nhìn tổng quan về cấu trúc dữ liệu.

1.3. Phương pháp nghiên cứu

Giải thuật BANG được áp dụng để phân cụm khách hàng trong bộ dữ liệu Cleaned Contoso nhằm nhận diện các nhóm khách hàng có hành vi và đặc điểm tương đồng. Những kết quả thu được từ việc phân cụm có thể được ứng dụng vào việc tối ưu hóa chiến lược tiếp thị bằng cách phát triển sản phẩm phù hợp, thiết kế các chương trình khuyến mãi nhắm đến từng nhóm khách hàng cụ thể và dự đoán xu hướng tiêu dùng trong tương lai. Cuối cùng, việc trực quan hóa kết quả phân cụm giúp dễ dàng nhận diện cấu trúc dữ liệu và mối quan hệ giữa các nhóm khách hàng.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan

2.1.1. Bài toán gom cụm

2.1.1.1. Bài toán gom cụm là gì

Theo Madhulatha [4], gom cụm (*clustering*) là một kỹ thuật phổ biến trong phân tích dữ liệu, được ứng dụng trong nhiều lĩnh vực như học máy, khai thác dữ liệu, nhận dạng mẫu, phân tích hình ảnh và các mẫu thử trong sinh học. Đây là quá trình gom nhóm các đối tượng có các đặc điểm tương tự vào các nhóm nhất định và các nhóm này cũng khác nhau, hoặc chia tập dữ liệu thành các tập con sao cho dữ liệu trong mỗi tập con có sự tương đồng dựa trên một nguyên tắc xác định về khoảng cách giữa các phần tử trong cụm với nhau. Gom cụm được coi là một trong những bài toán quan trọng nhất trong lĩnh vực không giám sát (*Unsupervised Learning*), với mục tiêu là tìm ra cấu trúc trong tập dữ liệu chưa được gán nhãn.

Bên cạnh đó, trong nghiên cứu của Fung [5], tác giả đã tổng hợp và giải thích một cách dễ hiểu về định nghĩa theo toán học của các bài toán gom cụm như sau,

Xét tập dữ liệu $X \in R^{mxn}$ trong đó m là số điểm dữ liệu và mỗi điểm x_i có n đặc trưng. Mục tiêu của quá trình gom cụm là chia tập X thành K nhóm C_k sao cho các điểm trong cùng một cụm có các đặc điểm giống nhau hơn so với các điểm ở nhóm còn lại. Nói cách khác, các điểm dữ liệu trong cùng một cụm có sự tương đồng cao hơn so với các điểm ở các cụm khác. Khi này, mỗi nhóm C_k sẽ được gọi là một cụm.

2.1.1.2. Các loại bài toán gom cụm

Theo Madhulatha [4], bài toán gom cụm có thể được chia thành hai loại: Phân cấp (*hierarchical*) và phân vùng (*partitional*). Các thuật toán phân cấp tạo ra các cụm liên tiếp dựa trên các cụm đã được hình thành trước đó. Các thuật toán này có thể được phân loại tiếp theo hai hướng nữa:

Thứ nhất, Agglomerative (từ dưới lên - bottom-up): Bắt đầu bằng cách coi mỗi phần tử là

một cụm riêng biệt, sau đó gộp dần chúng lại thành các cụm lớn hơn.

Thứ hai, Divisive (từ trên xuống - top-down): Bắt đầu với toàn bộ tập dữ liệu trong một cụm duy nhất và sau đó chia dần thành các cụm nhỏ hơn.

Đặc biệt, đặc điểm nổi bật để nhận diện các thuật toán phân cụm theo hướng phân cấp đó chính là nguyên tắc phân cụm của các thuật toán loại này luôn dựa trên một phương pháp tính khoảng cách giữa các cụm với nhau. Có hai phương pháp tính khoảng cách phổ biến trong các loại thuật toán phân cấp gọi là Manhattan distance và Euclidean distance.

Cuối cùng, thuật toán phân vùng: Xác định tất cả các cụm cùng một lúc mà không dựa vào việc hình thành dần dần như trong phân cấp. Đặc điểm nổi bật để nhận diện các loại thuật toán phân vùng chính là các thuật toán phân vùng dựa trên việc xác định trước số lượng cụm từ ban đầu trước khi bắt đầu thuật toán và sau đó sẽ lặp lại việc phân chia các phần tử trong bộ dữ liệu đang xét vào các nhóm và điều chỉnh cho đến khi các phần tử đó được xếp vào cụm dữ liệu phù hợp nhất. Thêm nữa, các thuật toán phân vùng còn được phân ra thành ba loại nhỏ nữa:

Thứ nhất, thuật toán phân vùng dựa trên mật độ điểm dữ liệu (DENSITY-BASED CLUSTERING). Các thuật toán phân vùng thuộc loại này xác định rằng chỉ các vùng có nhiều phần tử tập trung lại (tức là vùng có mật độ cao) mới được xem là cụm, ngược lại, các vùng có mật độ thấp hơn thường được coi là nhiễu hoặc là các giá trị nằm ngoài cụm.

Thứ ba, thuật toán phân vùng dựa trên các mô hình phân phối dữ liệu. Cách hoạt động của nó là cố gắng làm sao để phân loại bộ dữ liệu được cho trước thành các mô hình phân phối cụ thể nào đó, mỗi loại mô hình sẽ là một cụm dữ liệu.

Thứ hai, thuật toán phân vùng dựa trên hình dạng không gian dữ liệu của lưới (GRID-BASED CLUSTERING). Các thuật toán thuộc dạng này sử dụng cấu trúc dữ liệu lưới đa phân giải (multiresolution grid data structure) với nguyên lý chia không gian dữ liệu dạng này thành các ô dữ liệu nhỏ khác nhau gọi là cells và ô dữ liệu nào có mật độ co sẽ được coi là một cụm. Trong bài nghiên cứu này, nhóm sử dụng giải thuật phân cụm BANG và đây là một giải thuật cũng được xây dựng trên nguyên tắc này.

2.1.2. Định nghĩa về cụm

Theo Madhulatha [4], cụm (*cluster*) là tập hợp các đối tượng có sự tương đồng với nhau và khác biệt so với các đối tượng trong các cụm khác.

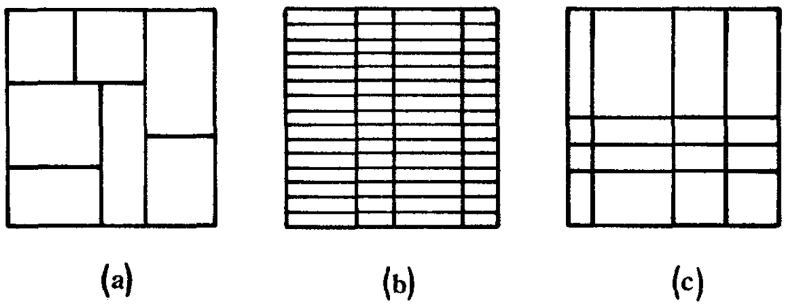
Theo Fung [5], các cụm thu được phải thể hiện sự tương đồng tự nhiên, bản chất vốn có của lĩnh vực mà dữ liệu đang thể hiện và có ý nghĩa giữa các điểm dữ liệu, không đơn giản chỉ là công việc gom nhóm các điểm dữ liệu lại với nhau một cách ngẫu nhiên.

Dựa trên nhận định về cụm của hai bài nghiên cứu trên, nhóm xác định định nghĩa về cụm như sau: Cụm là một tập hợp các nhóm dữ liệu bao gồm các điểm dữ liệu có đặc điểm giống nhau, có sự khác biệt so với các điểm dữ liệu thuộc các nhóm dữ liệu khác và các nhóm dữ liệu này đều có một đặc điểm chung chính là thể hiện được đặc điểm cụ thể của lĩnh vực mà bộ dữ liệu đang xét thể hiện.

2.2. Grid File

File là một cấu trúc để lưu trữ các bản ghi, thuở trước, các bản ghi trong file được truy cập theo cơ chế khóa đơn *(single-key access)* nhưng theo thời gian, hiệu suất của cơ chế này có chiều hướng giảm dần do sự phát triển của các file động *(highly dynamic file)*. Vì thế mà từ đó cần đến một cơ chế truy cập mới, cơ chế đa khóa *(multikey access)*. Giả sử ta có một file F, một tập A là các trường của file F, bản ghi R được gọi là truy cập theo cơ chế đa khóa khi ta dùng một tập con của A để truy cập đến bản ghi R [10].

Các chiến lược tìm kiếm có thể được phân vào hai loại: (1) các chiến lược tìm kiếm tổ chức lại tập dữ liệu và (2) các chiến lược tìm kiếm tổ chức lại không gian nhúng *(embedding space)* mà qua đó dữ liệu được truy vấn. Chẳng hạn, cây tìm kiếm nhị phân rơi vào loại đầu tiên bởi các điểm phải được sắp xếp tăng dần để phù hợp với cấu trúc dữ liệu này. Mỗi chiến lược tìm kiếm đều chia không gian tìm kiếm thành các không gian con,



Hình 1: Phân chia không gian tìm kiếm bởi các chiến lược tìm kiếm khác nhau - Nguồn: [10]

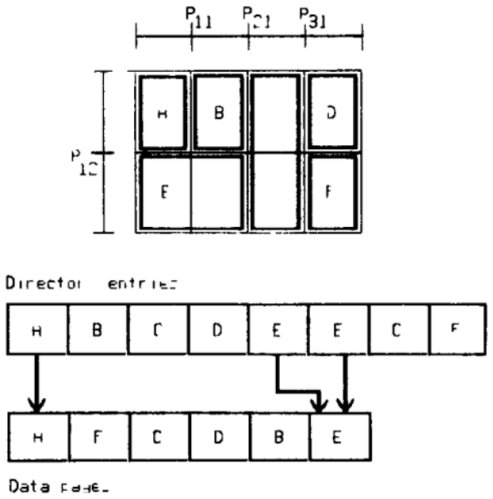
Với hình 1, ở 1a ta chia không gian tìm kiếm theo phương pháp chia để trị, không gian tìm kiếm được chia nhỏ dần dẫn đến ranh giới giữa các vùng trở nên bé hơn. Ở 1b, không gian tìm kiếm lại được chia sao cho mật độ giữa các vùng tiệm cận đồng nhất với nhau. Đối với 1c, mỗi lần phân chia, không gian tìm kiếm sẽ được phân chia thành hai vùng con, đây cũng chính là chiến lược phân chia dạng lưới *(grid partition)* làm tiền đề cho grid file và BANG file mà nhóm muốn đề cập đến.

Grid file là một cấu trúc đối xứng, linh hoạt và dữ liệu được truy xuất theo cơ chế đa khóa. Nói Grid file đối xứng là bởi mọi trường đều được coi là khóa chính, nói linh hoạt là vì cấu trúc tự tinh chỉnh tùy vào nội dung mà nó sẽ chứa sao cho mật độ giữa các vùng và thời gian truy cập là như nhau dù cho phân phối của dữ liệu có thể sẽ không đồng nhất [10]. Dựa theo bản chất nội tại của grid partition và cơ chế truy vấn đa khóa của grid file thì nghiễm nhiên rằng, các vùng được phân chia sẽ có dạng giống một hình chữ nhật.

Để định nghĩa một grid file, nhóm sẽ định nghĩa thông qua ba đối tượng chính: vùng lưới (*grid regions)*, xô *(bucket)*, thư mục lưới (*grid directory)*.

Như đã đề cập, khi dùng phân chia dạng lưới để chia không gian tìm kiếm thành các không gian con, ta sẽ được các vùng hình chữ nhật, đây chính là vùng lưới. Và nơi để lưu trữ các dữ liệu đó, lại được định nghĩa là một xô, một xô sẽ chứa tối đa *c* bản ghi. Nếu như số lượng bản ghi vượt quá *c* sẽ dẫn đến việc phân chia và hệ quả là, hai vùng có thể xuất phát từ cùng một bucket. Vùng lưới là các vùng hình chữ nhật từ phân chia mà ra, còn xô lại là nơi lưu trữ dữ liệu, vì vậy mà ta sẽ cần một cầu nối giữa hai cấu trúc này và đó là thư mục lưới. Một thư mục lưới sẽ gồm hai phần, phần thứ nhất là một mảng k chiều, phần tử của

các mảng này sẽ đóng vai trò như một con trỏ chỉ tới các bucket; phần thứ hai là các mảng một chiều, mảng một chiều này sẽ là các thang đo tuyến tính giúp cho việc phân chia không gian tìm kiếm. Và cấu trúc grid file đầu tiên minh họa cho điều này là scale-based grid file, đây cũng chính cấu trúc grid file đầu tiên được biết đến và phổ biến nhất.



Hình 2: Scale-based grid file - Nguồn: [11]

Trong SG file, thang đo tuyến tính sẽ được sử dụng để phân chia không gian tìm kiếm. Phần tử của thư mục lưới sẽ là tương ứng 1:1 với một xô, tức mỗi phần tử sẽ trỏ tới một xô. Tuy vậy, khi mà số lượng phần tử của thư mục lưới tăng lên, nếu như tăng theo hàm mũ, sẽ xảy ra trường hợp phần tử của thư mục lưới sẽ trỏ tới xô rỗng và vấn đề này sẽ càng tệ hơn với số chiều lớn [11].

Một cấu trúc khác dựa trên grid file cũng được sử dụng đó là Interpolation-based grid file. Cấu trúc này sẽ sử dụng một tập phân cấp cho các vùng lưới, mỗi vùng lưới sẽ được xác định bởi một cặp (r, l) trong đó *r* đại diện cho chỉ số vùng và *l* đại diện cho mức. Nhưng cấu trúc này cũng gặp vấn đề tương tự với SG file khi mà có trường hợp phần tử của thư mục lưới trỏ tới xô rỗng. Điểm nổi bật chính của Interpolation-based grid file đó là cho kết quả tốt đối với các dữ liệu có phân phối không đồng nhất.

Vì vậy mà để giải quyết vấn đề trên, một cấu trúc khác dựa trên grid file được ra đời, đó là BANG file.

2.3. BANG file

BANG file *(Balanced and Nested Grid File)* là Interpolation-based file nhưng có một số cải tiến nhất định. BANG file vẫn chia không gian dữ liệu thành các vùng lưới và sử dụng cơ chế đánh số (r, l) như trên Interpolation-based file.

Đối với SG file và Interpolation-based file sẽ có hai tiên đề sau ,

Thứ nhất, hợp của những không gian con được phân chia từ không gian dữ liệu phải là chính không gian dữ liệu đó và,

Thứ hai, phần giao giữa hai không gian con được phân chia sẽ không tồn tại.

BANG file chấp nhận tiên đề đầu tiên nhưng đối với tiên đề thứ hai sẽ được thay thế bằng "Nếu hai không gian con giao nhau thì không gian này sẽ chứa không gian kia, hay nói cách khác, BANG file sẽ cho phép không gian trong không gian" [11].

BANG file sẽ cho phép cân bằng lại phân phối của các điểm dữ liệu bằng cách phân phối lại các vùng, và từ đó đảm bảo rằng không có xô rỗng tồn tại, đây cũng chính là đặc điểm giúp BANG file tiếp cận được với dữ liệu có phân phối không đồng nhất. Và có lẽ cũng chính đặc điểm trên của BANG file đã đặt nền tảng cho BANG-clustering được ra đời.

2.4. Giải thuật BANG

2.2.1. Ý tưởng

Đây là một thuật toán phân lớp được dùng cho các bộ dữ liệu có dung lượng lớn. Phương pháp sử dụng một cấu trúc dữ liệu dạng lưới nhiều chiều [1]. Cấu trúc lưới cho phép dữ liệu được tổ chức thành các ô trong không gian nhiều chiều.

Theo [7], tác giả đã so sánh và tổng hợp đặc điểm của các loại thuật toán phân cụm phổ biến trong việc xử lý với các bộ dữ liệu lớn và kết quả cho thấy một thuật toán phân cụm phổ biến như K-means so với BANG thì K-means không xử lý được những bộ dữ liệu có dung lượng lớn và những dữ liệu có tính đa chiều cao, còn BANG thì có thể. Nguyên nhân của việc này chính là trong quá trình thực hiện thuật toán K-means cần phải lặp lại công việc tính khoảng cách để xác định cụm mới tối ưu. Mỗi lần thực hiện tính khoảng cách, tổng số phép tính khoảng cách trong lần lặp đó là D*k, với D là số đối tượng, k là số các

cụm. Vì lý do này nên trong bộ dữ liệu lớn sẽ tốn rất nhiều thời gian và chi phí để có thể phân cụm hoàn chỉnh [8]. Ngoài ra, nguyên nhân của vấn đề vì sao K-means không xử lý được bộ dữ liệu đa chiều là vì lời nguyền của tính đa chiều (the curse of dimensionality) trong các công thức tính khoảng cách. Quay trở lại nguyên tắc hoạt động của K-means, thuật toán phân cụm này hoạt động dựa trên nguyên tắc tính khoảng cách giữa các cụm với nhau. Song, trong bộ dữ liệu đa chiều, khi số chiều tăng lên sẽ làm tăng khoảng cách giữa các điểm dữ liệu của một bộ dữ liệu cố định cho trước. Điều này dẫn tới mật độ dữ liệu trong không gian dữ liệu sẽ thấp hơn dẫn đến tình trạng phân cụm khó có kết quả chính xác hơn [9].

Mặt khác, đối với thuật toán BANG, nguyên lý hoạt động của thuật toán không sử dụng các hàm khoảng cách để phân cụm nên không bị ảnh hưởng bởi lời nguyền của tính đa chiều (the curse of dimensionality), do đó thuật toán này hoạt động tốt đối với các bộ dữ liệu có tính đa chiều cao. Dưới đây là cách thức hoạt động của thuật toán,

Đầu tiên với một bộ dữ liệu, ta sẽ tạo một cấu trúc dữ liệu gọi là BANG, trong không gian đó, các giá trị được phân chia thành các ô (blocks) hình chữ nhật trong không gian nhiều chiều (k-dimensional value space) và tập hợp các blocks này gọi chung là Grid Regions.

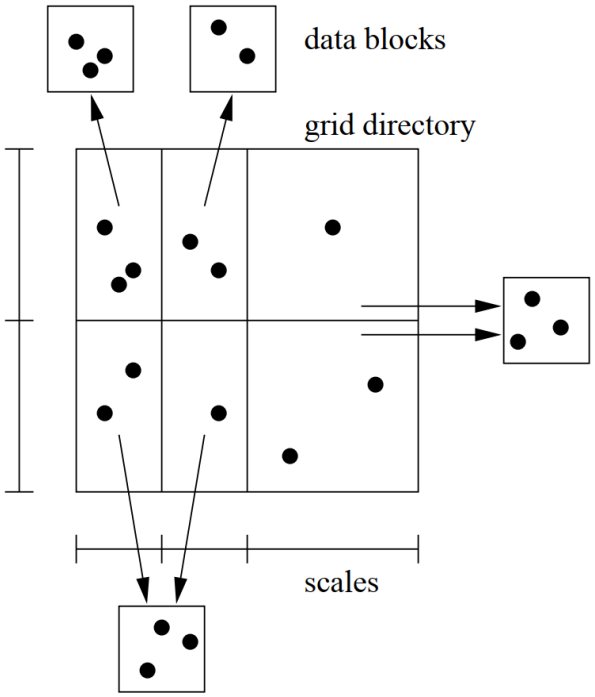
Đưa các mẫu dữ liệu của bộ dữ liệu (pattern) vào ngẫu nhiên trong không gian BANG, các pattern này có thể là các mẫu dữ liệu theo một chiều nào đó (dimensions). Các pattern này được thêm vào ngẫu nhiên trong không gian giá trị đa chiều của BANG và được lưu trữ trong không gian BANG dựa trên giá trị của chúng nhưng vẫn đảm bảo duy trì hình dạng phân phối ban đầu của dữ liệu khi thực hiện quá trình chạy thuật toán phân cụm. Thuật toán BANG phân chia không gian giá trị và quản lý các điểm dữ liệu này bằng các khối hình chữ nhật gọi là blocks và thuật toán BANG phân nhóm các mẫu dữ liệu (patterns) này dựa trên thông tin về các khối hình chữ nhật trong cấu trúc BANG để phân cụm theo các khối lân cận của chúng. Quá trình này tạo ra một sơ đồ phân cấp (dendrogram) thể hiện mối quan hệ giữa các cụm.

2.2.2. Cấu trúc

2.2.2.1. BANG-Structure

Phân cụm BANG sử dụng cấu trúc BANG-Structure để lưu các điểm dữ liệu trong không

gian giá trị bằng một cấu trúc lưới, grid-directory. Cấu trúc lưới này sẽ chia không gian giá trị k-chiều thành những vùng lưới *(grid regions)*, đó là những vùng hình chữ nhật sẽ chứa điểm dữ liệu và được phân chia dựa trên *scales*. Mỗi *scale* đại diện cho một thuộc tính của điểm dữ liệu [1].



Hình 3: BANG-Structure - Nguồn: [1]

Kết quả khi phân chia không gian giá trị sẽ là một tập phân cấp *(hierachical set)* với phần tử là các vùng lưới. Mỗi vùng này sẽ được xác định bằng một cặp (r, l), trong đó r là chỉ số vùng và l là mức. Với định nghĩa này thì vùng (0,0) sẽ là toàn bộ không gian giá trị [1].

Đặt GridRegions là tập các vùng lưới, DataBlocks là tập các khối dữ liệu và ta sẽ định nghĩa ánh xạ f: GridRegions → DataBlocks, quan hệ giữa GridRegions và DataBlocks sẽ là (1:m). Như vậy, với mỗi phần tử của GridRegions có thể có tương ứng m khối dữ liệu. Sở dĩ có điều này là do, một khối dữ liệu có thể được ánh xạ bởi nhiều vùng lưới. Hợp của những vùng lưới có tập ảnh giống nhau (ánh xạ đến cùng một khối dữ liệu) tạo thành vùng khối *(block region)*. Toán tử phân chia là một quan hệ hai ngôi, một vùng sẽ được chia thành hai vùng bằng nhau trong mỗi chiều.

Các vùng khối được định nghĩa bởi hai tiên đề sau [1],

Thứ nhất, hợp của những vùng con được phân chia sẽ bao toàn bộ không gian giá trị.

Thứ hai, nếu hai vùng con giao nhau thì một trong hai vùng này sẽ chứa vùng còn lại.

Như vậy, BANG-Structure sẽ bao gồm các thành phần sau, thư mục lưới *(grid-directory)* để lưu trữ các điểm và được sử dụng để phân chia thành các vùng dựa trên scales; vùng lưới *(grid regions)* là những vùng hình chữ nhật mà thư mục lưới phân chia không gian giá trị tạo thành; khối dữ liệu *(data block)* sẽ lưu trữ tập hợp các điểm dữ liệu; vùng khối là hợp của những vùng lưới có tập ảnh giống nhau với ánh xạ f.

2.2.2.2. Density Index

Thuật toán tính toán một chỉ số mật độ của mỗi khối thông qua số lượng mẫu và thể tích không gian của khối. Thể tích không gian V_B của một khối B (bucket) là tích Descartes của các phạm vi e của khối B trong mỗi chiều, tức là:

$$V_B = \prod e_{B_i}, \; i = 1, \ldots, k$$

Chỉ số mật độ DB của khối B được định nghĩa là tỷ lệ giữa số lượng mẫu p_B thực tế chứa trong khối B với thể tích không gian V_B của B, tức là:

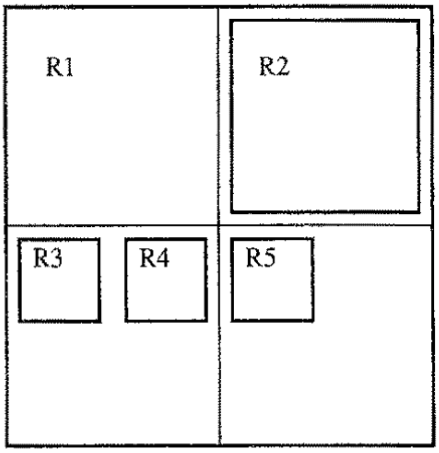
$$D_B = \frac{p_B}{V_B}$$

Các khối được sắp xếp theo chỉ số mật độ của chúng. Khối có chỉ số mật độ cao nhất (rõ ràng với độ tương quan mẫu cao nhất) trở thành tâm cụm. Các khối còn lại sau đó được phân cụm lặp đi lặp lại theo thứ tự chỉ số mật độ của chúng, từ đó xây dựng tâm cụm mới hoặc hợp nhất với các cụm hiện có. Chỉ các khối liền kề với một cụm, tức là các hàng xóm, mới có thể được hợp nhất.

2.2.2.3. Hàng xóm (Neighbors)

Hàng xóm có thể được chia thành 2 loại, đó là *normal neighborhood* và *refined neighborhood.* Normal neighborhood là những hàng xóm tương ứng với vùng khối trong

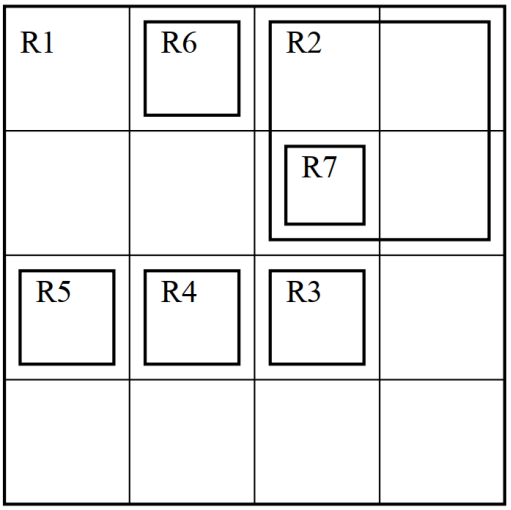
khi refined neighborhood lại là những hàng xóm tương ứng với vùng luân lý *(logical regions)*. Vùng luân lý được tính bằng cách, với một vùng cho trước ta lấy không gian giá trị của vùng trừ đi không gian giá trị của tất cả các vùng con nằm trong không gian của vùng cho trước. Khi đó mỗi vùng luân lý sẽ là một tập các vùng sao cho một vùng chứa các vùng còn lại [3].



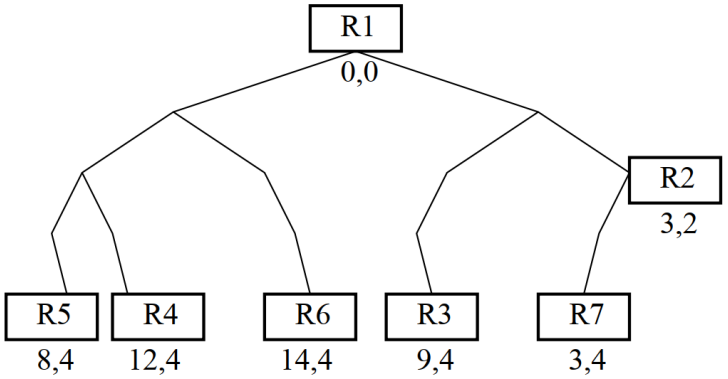
Hình 4: Vùng luân lý - Nguồn: [3]

Để có thể xác định một vùng có phải là hàng xóm với một vùng khác hay không, ta sẽ định một chỉ số thể hiện số chiều mà hai vùng có chung đó là Độ lân cận vùng *(neighbor degree)*. Độ lân cận vùng sẽ được xác định bằng chiều tiếp xúc giữa hai vùng.

Hàng xóm sẽ được xác định bằng cách so sánh scales của thư mục lưới. Nếu không có sự khác về scales giữa hai vùng thì hai vùng đó là hàng xóm và ngược lại. Trường hợp nếu hai vùng thuộc cùng một mức thì sự khác nhau về scales sẽ được xác định dễ dàng hơn. Trường hợp nếu hai vùng không thuộc cùng một mức thì vùng có mức thấp hơn sẽ được biến đổi lên mức cao. Và bởi do cơ chế đánh số của vùng lưới nên ta sẽ chọn cây nhị phân dành cho việc lưu trữ cấu trúc lưới, việc phân chia sẽ được hỗ trợ bởi cấu trúc cây này và vùng ở mức thấp hơn - vùng chứa các vùng khác sẽ được tìm bằng backtracking.



Hình 5: Cấu trúc lưới thể hiện hàng xóm - Nguồn: [1]



Hình 6: Cây nhị phân lưu trữ cấu trúc lưới - Nguồn: [1]

2.2.2.4. Dendrogram

Dendrogram được tính toán trực tiếp bởi thuật toán phân cụm. Chỉ số mật độ của tất cả các vùng được tính toán và sắp xếp theo thứ tự giảm dần. Bắt đầu từ vùng đầu tiên (có chỉ số mật độ cao nhất), tất cả các vùng lân cận được xác định và phân loại theo thứ tự giảm dần (bước 1). Việc tìm kiếm lân cận được lặp lại cho mỗi vùng đã được xử lý. Các vùng được tìm thấy được đặt vào dendrogram ở bên phải của các vùng gốc (bước 2), tuân theo các quy tắc sau:

Nếu R1 là lân cận của R2 và R2 là lân cận của R3, và R1 > R2 > R3, thì xây dựng một cụm với R1, R2 và R3 (tìm kiếm lân cận bắt đầu từ R3).

Nếu R1 là lân cận của R2 và R2 là lân cận của R3, và R1 > R2 < R3, thì xây dựng một cụm với R1, R2 và R3 (tìm kiếm lân cận bắt đầu từ R2).

Để hiểu rõ hơn về cách thức hình thành dendrogram hãy phân tích hình dưới như sau:

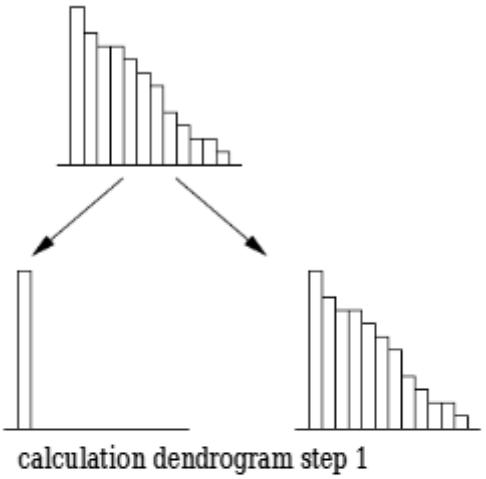


Fig. 7. Dendrogram creation

Hình 7: Sự hình thành của dendrogram - Nguồn: Erich Schikuta và Martin Erhart (1997)

Ở bước 1 (calculation dendrogram step 1) thì thuật toán tính toán chỉ số mật độ của tất cả các vùng và sắp xếp chúng theo thứ tự giảm dần. Bắt đầu với vùng có chỉ số mật độ cao nhất, thuật toán xác định các vùng lân cận và xếp chúng theo thứ tự giảm dần mật độ tạo thành cấu trúc dendrogram ban đầu, với các vùng có mật độ cao hơn ở phía bên trái.

Sang bước 2 (calculation dendrogram step 2) thì quá trình tìm kiếm vùng lân cận được lặp lại cho mỗi vùng đã được xử lý với quy tắc được nêu lên ở trên. Những vùng lân cận được tìm thấy được đặt trong dendrogram ở bên phải của các vùng gốc.

2.2.3. Quy trình phân cụm

Giải thuật phân cụm BANG dựa trên ý tưởng của Warnerka để xây dựng không gian giá trị chứa patterns sử dụng BANG-Structure [2]. Chúng được xem là các điểm trong không gian giá trị *k* chiều và được thêm ngẫu nhiên vào BANG-Structure. Những điểm này được lưu theo nhóm đặc trưng của chúng mà vẫn bảo toàn được cấu trúc và phân phối. BANG-Structure phân chia không gian và quản lý các điểm dữ liệu qua các khối (blocks), sau đó phân cụm dựa vào block information, hình thành được dendogram.

Quy trình phân cụm cụ thể như sau:

Bước 1: Tính toán *Density Index* của từng khối thông qua số lượng điểm và phần thể tích trong không gian.

Bước 2: Sắp xếp các khối theo thứ tự giảm dần Density Index. Khối có Density Index cao nhất - có nghĩa là có sự tương quan cao nhất giữa các điểm, sẽ trở thành trung tâm cụm. Các khối còn lại sau đó được phân cụm dựa theo Density Index lặp đi lặp lại, theo đó xây dựng được trung tâm cụm mới hoặc hợp nhất với những cụm hiện có. Chỉ những khối cùng thuộc về một cụm - được gọi là hàng xóm (neighbours), mới có thể hợp nhất.

Bước 3: Tìm những hàng xóm bằng cách so sánh các giá trị tỉ lệ (scale values) của cấu trúc lưới grid-directory. Nếu hai regions đồng mức, ta có thể phân chia ngay lập tức, nếu không regions ở mức thấp hơn phải chuyển đổi lên mức cao hơn sau đó tiếp tục quá trình so sánh.

Bước 4: Xây dựng dendogram, sau khi đã hoàn tất các bước tính toán Density Index và tìm Neighbours. Bắt đầu từ region đầu tiên - với Density Index cao nhất, các hàng xóm của region này được xác định và phân về cụm theo thứ tự giảm dần.

2.5. Siêu tham số

Dựa trên tài liệu của thư viện pyclustering về BANG-clustering, ta sẽ có các siêu tham số cũng như ý nghĩa của các siêu tham số ấy như sau,

Một, levels (uint) là số cấp trong cây phân cấp được dùng để phân tách khối - nghĩa là một khối nên được tách bao nhiêu lần.

Hai, ccore (bool) là đối số nếu sử dụng thư viện CCORE (C/C++). Mặc định ccore = False

Ba, density_threshold, siêu tham số này nói rằng nếu mật độ của khối nhỏ hơn giá trị này, thì dữ liệu chứa trong khối sẽ được coi là nhiễu, và các điểm trong khối sẽ được coi là các điểm ngoại lai. Mật độ của khối được định nghĩa bằng số lượng điểm trong khối chia cho thể tích của khối.

Bốn, amount_threshold (uint) là số lượng điểm trong khối khi khối chứa dữ liệu trong bang-block được coi là nhiễu và không cần phải chia nhỏ thêm cho đến cấp cuối cùng.

2.6. Đánh giá phương pháp phân cụm

Để đánh giá hiệu năng của giải thuật BANG, nhóm sẽ tiến hành đánh giá thông qua độ phức tạp không gian cũng như thời gian. Ngoài ra nhóm cũng sẽ đánh giá thông qua thời gian chạy thực tế với cấu hình sử dụng 2 chip Intel(R) Xeon(R) CPU @ 2.20GHz và 13GB RAM.

Để đánh giá kết quả phân cụm của giải thuật BANG, nhóm cũng sẽ sử dụng điểm đánh giá Silhouette (*Silhouette Score*). Chỉ số này đo lường mức độ tương đồng của một điểm dữ liệu với các điểm trong cùng cụm của nó so với các điểm ở cụm lân cận nhất.

Giá trị của Silhouette Score nằm trong khoảng [-1, 1]: Càng gần về phía 1 thì điểm dữ liệu được phân cụm đúng, cho thấy các điểm dữ liệu trong cùng một cụm tương đồng với nhau và có sự phân biệt rõ ràng với cụm khác. Giá trị gần 0 thì điểm dữ liệu nằm ở trên hoặc rất gần ranh giới giữa 2 cụm lân cận. Càng gần về phía -1 thì điểm dữ liệu có thể đã bị chồng chéo hoặc phân cụm sai vì nó gần các điểm trong cụm khác hơn là các điểm trong cụm của điểm đó. Cách tính Silhouette Score thông qua các bước,

Bước 1: Với mỗi điểm dữ liệu, tính khoảng cách trung bình a_i tới mỗi điểm dữ liệu trong cùng cụm đó. Giá trị a_i thể hiện độ tương đồng của điểm dữ liệu đang xét tới các điểm khác trong cụm.

Bước 2: Với mỗi điểm dữ liệu, tính khoảng cách trung bình b_i tới cụm lân cận nhất không chứa điểm này. Giá trị b_i thể hiện độ khác biệt của điểm dữ liệu đó so với cụm khác.

Bước 3: Tính toán chỉ số Silhouette Score cho điểm dữ liệu

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

Bước 4: Lấy trung bình Silhouette scores toàn bộ điểm dữ liệu, thu được Silhouette score tổng quát - đo lường hiệu quả của quá trình phân cụm:

$$S = \frac{1}{n} \sum_{i=1}^{n} s_i$$

CHƯƠNG III: TỔNG QUAN BỘ DỮ LIỆU

3.1. Giới thiệu bộ dữ liệu

Bộ dữ liệu Cleaned Contoso là một bộ dữ liệu mô phỏng các hoạt động kinh doanh của một công ty bán lẻ giả định tên Contoso. Đây là đường dẫn của bộ dữ liệu Cleaned Contoso trên Kaggle: [Cleaned Contoso Dataset]. Dữ liệu được làm sạch và sẵn sàng cho phân tích, bao gồm thông tin về khách hàng, sản phẩm, giao dịch bán hàng và địa lý. Mỗi bản ghi trong bộ dữ liệu là một giao dịch hoặc thông tin về một khách hàng hoặc sản phẩm cụ thể, cho phép phân tích sâu sắc về hành vi mua hàng, phân khúc khách hàng, và doanh thu theo thời gian hoặc theo khu vực.

Bộ dữ liệu gồm 25 tệp CSV với 244 cột, mỗi tệp chứa thông tin về một khía cạnh cụ thể trong hoạt động kinh doanh nhưng trong đề tài nhóm sử dụng 2 tệp,

Thứ nhất, DimCustomer.csv: Lưu trữ thông tin chi tiết về khách hàng như tên, tuổi, giới tính, và các yếu tố nhân khẩu học khác cho phép thực hiện các phân tích theo phân khúc khách hàng.

Thứ hai, DimGeography.csv: Bao gồm thông tin vị trí địa lý của khách hàng, từ đó có thể thực hiện các phân tích theo khu vực địa lý.

3.2. Các thuộc tính

3.2.1. DimCustomer

STT	Tên thuộc tính	Mô tả	Chú thích
1	CustomerKey	Mã khách hàng	
2	GeographyKey	Mã định danh địa lý duy nhất cho từng khu vực	
3	FirstName	Tên của khách hàng	
4	LastName	Họ của khách hàng	

5	BirthDate	Ngày tháng năm sinh của khách hàng	
6	MaritalStatus	Tình trạng hôn nhân của khách hàng	“M” : Đã kết hôn “S” : Độc thân
7	Gender	Giới tính của khách hàng	“M”: Nam “F” : Nữ
8	YearlyIncome	Thu nhập hàng năm của khách hàng	
9	TotalChildren	Tổng số con của khách hàng	
10	NumberChildrenAtHome	Số con đang sống chung tại nhà	
11	Education	Trình độ học vấn của khách hàng	
12	Occupation	Nghề nghiệp của khách hàng	
13	HouseOwnerFlag	Trạng thái sở hữu nhà của khách hàng	“1” : sở hữu nhà “0” : không sở hữu nhà
14	NumberCarsOwned	Số lượng xe khách hàng sở hữu	

Bảng 1: DimCustomer

3.2.2. DimGeoraphy

STT	Tên thuộc tính	Mô tả	Chú thích
1	GeographyKey	Mã định danh địa lý duy nhất cho từng khu vực	
2	GeographyType	Loại hình địa lý	
3	ContinentName	Tên châu lục của khu vực	
4	CityName	Tên thành phố khách hàng đang sinh sống	
5	StateProvinceName	Tên tiểu bang hoặc tỉnh của khu vực	
6	RegionCountryName	Tên quốc gia của khu vực khách hàng sống	

Bảng 2: DimGeography

CHƯƠNG IV: TIỀN XỬ LÝ DỮ LIỆU

4.1. Quan sát sơ lược về bộ dữ liệu

Mục tiêu phân cụm khách hàng nhằm phục vụ cho việc đánh giá và phân tích tình hình bán hàng đối với từng nhóm khách hàng khác nhau để từ đó đưa ra những thông tin hữu ích phục vụ cho quá trình quản trị quan hệ khách hàng tại Contoso. Nếu ở Chương 3 ta đã có cái nhìn tổng quan về các thuộc tính có trong hai bảng dữ liệu được lựa chọn của bộ dữ liệu Contoso gốc là DimCustomer và DimGeography thì ở phần này, hai bảng dữ liệu trên sẽ được mô tả chi tiết kèm với tình hình hiện tại để rút ra được những vấn đề mà bộ dữ liệu đang gặp phải, giúp tiết kiệm thời gian khi xác định vấn đề của dữ cần phải xử lý hơn.

4.1.1. Bảng DimCustomer

```
customer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18869 entries, 0 to 18868
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            18869 non-null  int64  
 1   CustomerKey           18869 non-null  int64  
 2   GeographyKey          18869 non-null  int64  
 3   FirstName             18869 non-null  object 
 4   LastName              18869 non-null  object 
 5   BirthDate             18869 non-null  object 
 6   MaritalStatus         18869 non-null  object 
 7   Gender                18869 non-null  object 
 8   YearlyIncome          18869 non-null  float64
 9   TotalChildren         18869 non-null  float64
 10  NumberChildrenAtHome  18869 non-null  float64
 11  Education             18869 non-null  object 
 12  Occupation            18869 non-null  object 
 13  HouseOwnerFlag        18869 non-null  float64
 14  NumberCarsOwned       18869 non-null  float64
dtypes: float64(5), int64(3), object(7)
memory usage: 2.2+ MB
```

Hình 8: Tổng hợp thông tin của bảng DimCustomer

Dưới đây là bảng tổng hợp chi tiết các thông tin liên quan đến bảng dữ liệu DimCustomer:

Bảng DimCustomer				
Tên cột	**Kiểu dữ liệu**	**Mô tả**	**Dữ liệu mẫu**	**Ghi chú**
Unnamed: 0	int64	Số thứ tự từ 0 đến 18869	0,1,2,3,4	
CustomerKey	int64	Mã khách hàng	1,2,3,4,5	
GeographyKey	int64	Mã định danh địa lý duy nhất cho từng khu vực	680,692, 493	
FirstName	object	Tên của khách hàng	Jon, Eugene, Ruben	
LastName	object	Họ của khách hàng	Yang, Huang, Torres	
BirthDate	object	Ngày tháng năm sinh của khách hàng	1966-04-08, 1965-05-14, 1965-08-12	
MaritalStatus	object	Tình trạng hôn nhân của khách hàng	M, S	“M” : Đã kết hôn “S” : Độc thân
Gender	object	Giới tính của khách hàng	M, F	“M”: Nam “F” : Nữ
YearlyIncome	float64	Thu nhập hàng năm của khách hàng	90000.0, 60000.0, 70000.0	
TotalChildren	float64	Tổng số con của khách hàng	2.0, 3.0, 4.0	
NumberChildre	float64	Số con đang sống	0.0, 3.0, 5.0	

nAtHome		chung tại nhà		
Education	object	Trình độ học vấn của khách hàng	Bachelors, Partial College, High School	
Occupation	object	Nghề nghiệp của khách hàng	Professional, Skilled Manual, Management	
HouseOwnerFlag	float64	Trạng thái sở hữu nhà của khách hàng	0.0, 1.0	
NumberCarsOwned	float64	Số lượng xe khách hàng sở hữu	0.0, 1.0, 4.0	

Bảng 3: Dữ liệu mẫu của DimCustomer

Qua bảng trên, ta thấy, các cột: Unnamed: 0, FirstName, LastName, BirthDate, HouseOwnerFlag, NumberCarsOwned là những cột không có ý nghĩa cho mục tiêu phân cụm khách hàng. Vì thế, nhóm tiến hành loại bỏ các cột này.

4.1.2. Bảng DimGeography

```
geography.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 674 entries, 0 to 673
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   GeographyKey       674 non-null    int64 
 1   GeographyType      674 non-null    object
 2   ContinentName      674 non-null    object
 3   CityName           517 non-null    object
 4   StateProvinceName  637 non-null    object
 5   RegionCountryName  671 non-null    object
dtypes: int64(1), object(5)
memory usage: 31.7+ KB
```

Hình 9: Tổng hợp thông tin của DimGeography

Dưới đây là bảng tổng hợp chi tiết các thông tin liên quan đến bảng dữ liệu DimGeography:

Bảng DimGeography				
Tên cột	**Kiểu dữ liệu**	**Mô tả**	**Dữ liệu mẫu**	**Ghi chú**
GeographyKey	int64	Mã định danh địa lý duy nhất cho từng khu vực	1,2,3, 269, 270	
GeographyType	object	Loại hình địa lý	Continent, Country/Region	
ContinentName	object	Tên châu lục của khu vực	Asia, Europe, North America	
CityName	object	Tên thành phố khách hàng đang sinh sống	NaN	
StateProvinceName	object	Tên tiểu bang hoặc tỉnh của khu vực	NaN	
RegionCountryName	object	Tên quốc gia của khu vực khách hàng sống	NaN, Armenia, Australia	

Bảng 4: Dữ liệu mẫu của DimGeography

Quan sát bảng tổng hợp trên, nhóm nhận thấy các cột CityName, StateProvinceName, RegionCountryName, GeographyType đều có thể được suy ra từ cột GeographyKey. Thế nên, nhóm tiến hành loại bỏ các cột này với mục hợp nhất hai bảng dữ liệu là DimCustomer và DimGeography ở phần sau cho hợp lý.

4.2. Tích hợp dữ liệu

Việc tích hợp dữ liệu từ hai bảng DimCustomer và DimGeography đóng vai trò quan trọng trong việc phân cụm khách hàng, sử dụng giải thuật BANG. DimCustomer cung cấp các thuộc tính nhân khẩu học (như độ tuổi, thu nhập, và giới tính), trong khi DimGeography

bổ sung các yếu tố địa lý chẳng hạn như khu vực sinh sống. Khi kết hợp hai bảng sẽ cho ra được một bức tranh chi tiết hơn về mỗi khách hàng, bao gồm cả đặc điểm cá nhân lẫn môi trường sống của họ.

Trước khi tích hợp, loại bỏ những thuộc tính không cần thiết hoặc không liên quan như FirstName, LastName, BirthDate, HouseOwnerFlag, NumberCarsOwned của bảng DimCustomer và CityName, StateProvinceName, RegionCountryName, GeographyType của bảng DimGeoraphy để giảm chi phí tính toán. Sau đó sử dụng phương thức merge để kết hợp hai bảng dựa trên khóa chung là GeographyKey tạo ra một bảng mới chứa tất cả thông tin cần thiết từ cả hai bảng. Sau khi gộp dữ liệu, các khóa như CustomerKey và GeographyKey được loại bỏ vì không cần thiết cho phân tích tiếp theo.

4.3. Làm sạch dữ liệu

4.3.1. Giá trị khuyết (Missing value)

Việc xác định missing values trong dữ liệu rất quan trọng vì chúng có thể gây sai lệch kết quả phân tích và giảm độ chính xác của các mô hình phân cụm. Xử lý missing values giúp đảm bảo tính toàn vẹn và chất lượng của dữ liệu, từ đó cải thiện hiệu quả phân tích và hiệu suất mô hình.

Tuy nhiên, như đã nhận xét ở phần 4.1, sau khi loại bỏ các cột dữ liệu không liên quan đến mục đích phân cụm khách hàng thì hai bảng dữ liệu DimCustomer và DimGeography không còn các giá trị bị khuyết (missing values).

Song, nhóm vẫn tiến hành kiểm tra một lần nữa để đảm bảo tính chính xác của kết luận ở phần 4.1, kết quả kiểm tra cho thấy nhận xét ở phần 4.1 là chính xác.

```
customer_geo.isnull().any()
```

```
MaritalStatus           False
Gender                  False
YearlyIncome            False
TotalChildren           False
NumberChildrenAtHome    False
Education               False
Occupation              False
ContinentName           False
dtype: bool
```

```
customer_geo.isna().any()
```

```
MaritalStatus           False
Gender                  False
YearlyIncome            False
TotalChildren           False
NumberChildrenAtHome    False
Education               False
Occupation              False
ContinentName           False
dtype: bool
```

Hình 10: Kết quả kiểm tra giá trị bị thiếu lần 2

4.3.2. Giá trị bất thường (Outlier)

Như đã nhận xét ở phần 4.1, trong bảng dữ liệu DimCustomer tồn tại các giá trị ngoại lai trong ba cột dữ liệu YearlyIncome, TotalChildren, NumberChildrenAtHome.

Sau khi tích hợp dữ liệu, nhóm tiến hành xử lý các giá trị ngoại lai với quy tắc 3-sigma. Nghĩa là điểm dữ liệu nào không thuộc khoảng $(mean - 3std; mean + 3std)$ sẽ bị loại bỏ với mean và std là trung bình và độ lệch chuẩn của thuộc tính tương ứng.

```
numeric_cols = customer_geo.select_dtypes(include=np.number).columns
#Dropping outliers using z-score
def outliers_cleaning(data: pd.DataFrame, numeric_columns: list) -> pd.DataFrame:
  for column in numeric_columns:
    upper_limit = data[column].mean() + 3*data[column].std()
    lower_limit = data[column].mean() - 3*data[column].std()
    data = data[(data[column] >= lower_limit) & (data[column] <= upper_limit)]
  return data

customer_geo = outliers_cleaning(customer_geo, numeric_cols)
```

Hình 11: Xử lý giá trị ngoại lai

4.4. Biến đổi dữ liệu

Sau các bước làm sạch dữ liệu thì dữ liệu nhóm sử dụng bao gồm các thuộc tính sau: MaritalStatus, Gender, YearlyIncome, TotalChildren, NumberChildrenAtHome, Education, Occupation, ContinentName

4.4.1. Thuộc tính định danh

Bằng cách sử dụng mã hóa nhãn, thuộc tính *MaritalStatus* đại diện cho tình trạng hôn nhân của khách hàng, với các giá trị cụ thể được mã hóa như sau: "Độc thân" được gán giá trị 0 (S) và "Đã kết hôn" được gán giá trị 1 (M). Tương tự, thuộc tính *Gender* cũng được mã hóa để phân nhóm khách hàng theo giới tính, giá trị "Nữ" được gán giá trị 0 (F) và "Nam" được gán giá trị 1 (M). Việc mã hóa này giúp mô hình dễ dàng phân tích và nhận diện các yếu tố ảnh hưởng đến hành vi của khách hàng dựa trên giới tính và trạng thái của khách hàng trong mối quan hệ cá nhân.

Encode "MaritalStatus" column

```
mapping_maritalstatus = {'M': 1, 'S': 0} ##1: Married, 0: Single
customer_geo['MaritalStatus'] = [mapping_maritalstatus[marital] for marital in customer_geo['MaritalStatus']]
```

Hình 12: Mã hóa "MaritalStatus"

Encode "Gender" column

```
mapping_gender = {'F': 0, 'M': 1} ##0: Female, 1: Male
customer_geo['Gender'] = [mapping_gender[sex] for sex in customer_geo['Gender']]
```

Hình 13: Mã hóa "Gender"

ContinentName là tên lục địa nơi khách hàng sinh sống giúp xác định vị trí địa lý mà không thể đo lường bằng số. Do đây là một biến phân loại với nhiều giá trị khác nhau ("Asia", "Europe", "North America") nên cần được chuyển đổi thành các cột riêng lẻ để

biểu diễn mỗi giá trị duy nhất dưới dạng nhị phân. Sử dụng One-Hot Encoding để chuyển đổi *ContinentName* tạo ra các cột riêng cho từng lục địa, mỗi cột có giá trị 0 hoặc 1 đại diện cho việc khách hàng có hoặc không sống tại thuộc lục địa đó.

Encode "ContinentName" column

```
onehot_continent = pd.get_dummies(customer_geo['ContinentName'], dtype=np.int64)
customer_geo = pd.concat([customer_geo, onehot_continent], axis = 1)
#Drop 'ContinentName" column
customer_geo = customer_geo.drop('ContinentName', axis = 1)
```

Hình 14: Mã hóa "ContinentName"

4.4.2. Thuộc tính định tính

Education là trình độ học vấn của khách hàng mô tả đặc điểm mà không thể đo lường bằng số. Trình độ học vấn có thể ảnh hưởng đến thói quen tiêu dùng và sự lựa chọn sản phẩm của khách hàng. Vì đây là một biến định tính với nhiều giá trị khác nhau, sử dụng mã hóa nhãn - Label Encoding giúp chuyển đổi dữ liệu giáo dục thành dạng số (Partial High School: 1, High School: 2, Partial College: 3, Bachelors: 4, Graduate Degree: 5), phù hợp cho các mô hình máy học.

Encode "Education" column

```
mapping_edu = {'Partial High School': 1, 'High School': 2, 'Partial College': 3, 'Bachelors': 4, 'Graduate Degree': 5}
customer_geo['Education'] = [mapping_edu[edu] for edu in customer_geo['Education']]
```

Hình 15: Mã hóa "Education"

Trong quá trình xử lý dữ liệu, việc nhị phân hóa (One-Hot Encoding) cũng được áp dụng cho thuộc tính *Occupation* - một biến phân loại chứa thông tin về nghề nghiệp của khách hàng ("Professional", "Skilled Manual", "Clerical", "Management", "Manual"). Vì đây là một biến định tính với nhiều giá trị khác nhau, nhị phân hóa giúp chuyển đổi dữ liệu nghề nghiệp thành dạng số, phù hợp cho các mô hình máy học. Mỗi nghề nghiệp được biểu diễn dưới dạng một cột riêng, mỗi cột có giá trị 0 hoặc 1 đại diện cho việc khách hàng có hoặc không thuộc nhóm nghề nghiệp đó.

```
onehot_occupation = pd.get_dummies(customer_geo['Occupation'], dtype=np.int64)
customer_geo = pd.concat([customer_geo, onehot_occupation], axis = 1)
#Drop 'Occupation" column
customer_geo = customer_geo.drop('Occupation', axis = 1)
```

Hình 16: Mã hóa "Occupation"

4.4.3. Thuộc tính định lượng

YearlyIncome là mức thu nhập hàng năm của khách hàng. Nhóm sẽ tiến hành tính toán độ lệch của thuộc tính này thông qua skewness score, và kết quả thu được khá bất ngờ, skewness score của YearlyIncome rơi vào khoảng 0.822. Một kết quả cho thấy rằng phân phối của YearlyIncome lệch phải. Vì vậy mà nhóm sẽ xử lý YearlyIncome với phương pháp Box-Cox và sau đó đo đường lại skewness score. Skewness score vào khoảng -0.03, độ lệch đã giảm đi rất nhiều, từ đó giúp cải thiện khả năng đóng góp của YearlyIncome vào phân cụm.

```
def transform_boxcox(data: pd.DataFrame, col: str) -> pd.Series:
  # Apply Box-Cox transformation
  transformed_data, _ = boxcox(data[col])

  # Apply MinMax scaling to bring values between 0 and 1
  scaler = MinMaxScaler()
  result = scaler.fit_transform(transformed_data.reshape(-1, 1)).flatten()

  return result.round(2)

customer_geo['YearlyIncome'] = transform_boxcox(customer_geo, 'YearlyIncome')
```

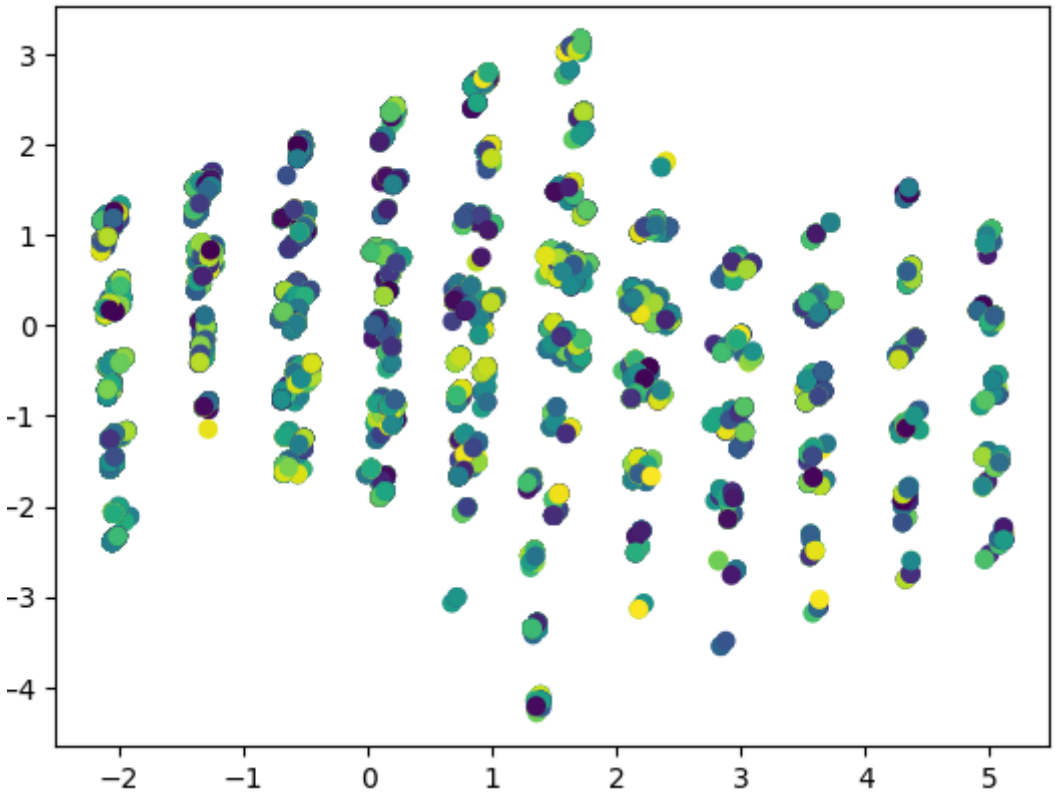
Hình 17: Xử lý skewness cao của YearlyIncome

TotalChildren cho biết tổng số con của khách hàng, trong khi *NumberChildrenAtHome* chỉ ra số con đang sống cùng khách hàng. Những thông tin này không chỉ giúp hiểu rõ hơn về quy mô gia đình mà còn ảnh hưởng đến quyết định mua sắm của khách hàng. Ví dụ, một gia đình có nhiều trẻ em có thể ưu tiên mua sắm các sản phẩm gia đình hoặc đồ chơi.

4.5. Giảm chiều dữ liệu

Nhóm sử dụng PCA (Principal component analysis) để giảm số chiều của bảng customer_geo với n_components = 2 chỉ định rằng dữ liệu sẽ được giảm xuống còn 2 thành phần chính để biểu diễn trên mặt phẳng 2D.

Biểu đồ này cung cấp một cái nhìn tổng quan về sự phân bố của dữ liệu customer_geo trong không gian 2 chiều sau khi áp dụng PCA. Các điểm dữ liệu được phân bố khá đồng đều trên mặt phẳng 2 chiều, không có cụm rõ ràng hay các khoảng cách quá lớn giữa các điểm cho thấy dữ liệu khách hàng có sự phân tán đa dạng, có thể do các đặc tính cá nhân và địa lý khác nhau. Màu sắc của các điểm được gán ngẫu nhiên nhằm tăng khả năng phân biệt các điểm nhưng không có ý nghĩa về phân cụm theo thuật toán hay các nhóm khách hàng cụ thể. Sự đa dạng về màu sắc ở đây chỉ hỗ trợ trực quan.

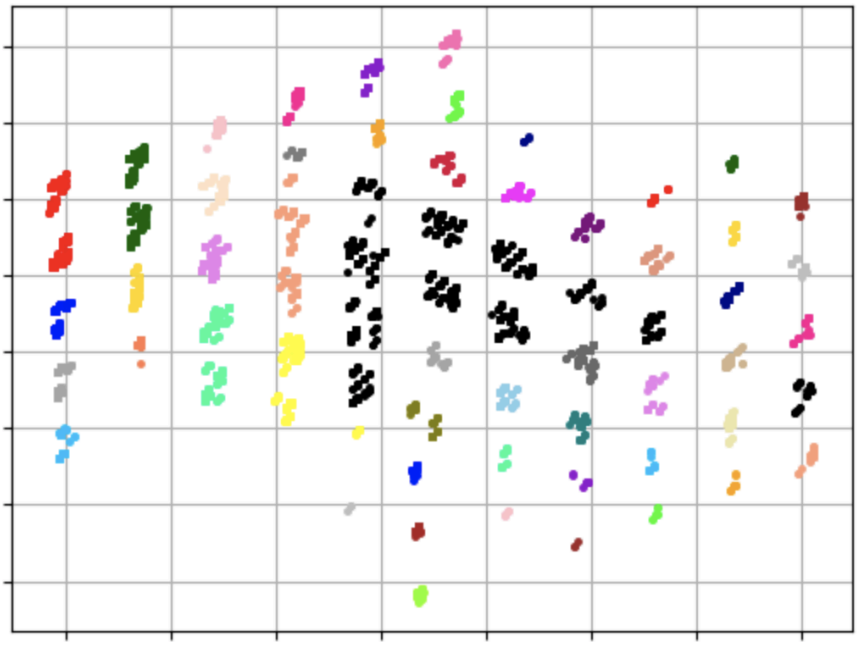


Hình 18: Kết quả giảm chiều dữ liệu với PCA

CHƯƠNG V: KẾT QUẢ THỰC NGHIỆM

5.1. Phân cụm trước hiệu chỉnh siêu tham số

Sau khi giai đoạn tiền xử lý dữ liệu đã thực hiện xong, giờ đây nhóm sẽ tiến hành phân cụm với BANG-clustering. Trước tiên, nhóm sẽ phân cụm với giá trị của tham số level = 11. Đây cũng là giá trị mặc định được cài đặt trên tài liệu của thư viện pyclustering. Khi đó, nhóm đạt được kết quả như hình sau,



Hình 19: Phân cụm trước hiệu chỉnh siêu tham số

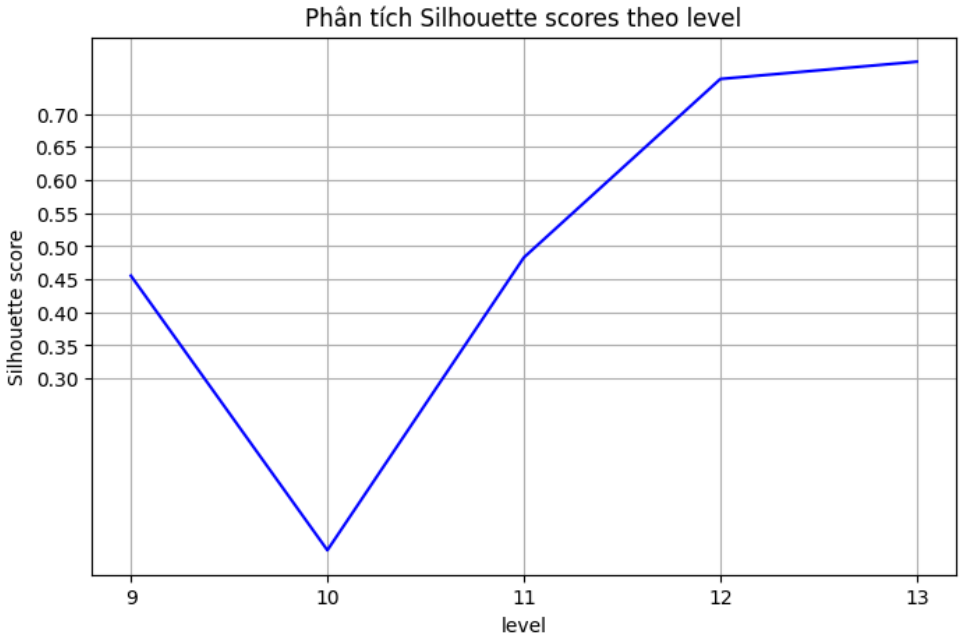
Với kết quả phân cụm có được ở hình trên, có thể nói rằng đây là một kết quả phân cụm chưa tốt khi mà có rất nhiều cụm. Xét đến mục tiêu phân cụm khách hàng, thì đây vốn dĩ không phải là một kết quả tốt bởi phân cụm khách hàng sẽ giúp cho doanh nghiệp xác định được những khách hàng có những tính cách, hành vi,... nào sẽ mua sản phẩm của doanh nghiệp trong khi đó kết quả đem lại có số lượng cụm lớn, doanh nghiệp sẽ không thể xác định được cụm nào sẽ là cụm mục tiêu mà doanh nghiệp muốn hướng đến. Do đó, nhóm sẽ tiến hành hiệu chỉnh siêu tham số.

5.2. Phân cụm sau hiệu chỉnh siêu tham số

Nhóm sẽ tiến hành hiệu chỉnh các tham số “level”, “density_threshold” và “amount_threshold”. Phương pháp mà nhóm sử dụng để hiệu chỉnh đó là phương pháp

elbow. Nhóm sẽ tiến hành chạy thuật toán nhiều lần với các giá trị khác nhau của siêu tham số, với trục hoành sẽ là giá trị của tham số còn trục tung nhóm sẽ sử dụng silhouette score. Giá trị của siêu tham số tương ứng với điểm gãy khúc sẽ là giá trị nhóm chọn để tiến hành phân cụm ở lần tiếp theo.

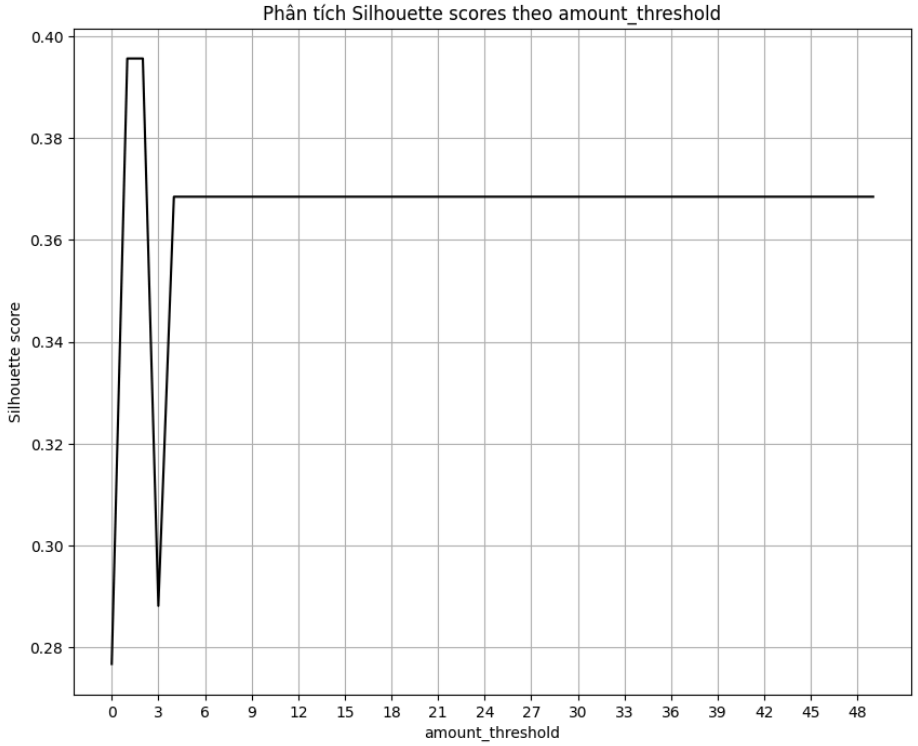
Thứ nhất, đối với "level" ta có kết quả,



Hình 20: Hiệu chỉnh "level"

Dễ thấy rằng, tại level = 10 giá trị silhouette score bắt đầu tăng trở lại, thoạt nhìn thì ta có thể chọn giá trị cho level ở 11, 12,... và thậm chí cao hơn do silhouette score ở các level này cao. Nhưng nhóm nhận thấy rằng, ở level càng cao thì số cụm sẽ tăng lên, số cụm tăng lên sẽ làm giảm khoảng cách từ một điểm giá trị đến cụm mà nó được gán, từ đó dẫn đến silhouette score tăng lên, tuy silhouette score tăng lên dẫu là dấu hiệu tốt nhưng sẽ không đúng trong mọi trường hợp, vì như vậy khả năng cao mô hình của nhóm sẽ rơi vào trường hợp quá khớp *(overfitting)* do đó mà nhóm sẽ không chọn ở level cao hơn. Đối với các level thấp hơn, ở mức level thấp thì số cụm gần như là bằng 1, nghĩa là chỉ có một cụm cho tất cả các điểm dữ liệu. Sở dĩ có điều đó là do level quy định số lần phân chia của một khối. Vậy nếu khối đó phân chia ít thì tất nhiên rằng, sẽ không có nhiều cụm vì vậy mà silhouette score cũng sẽ cao. Và với lập luận trên, nhóm sẽ chọn level = 10.

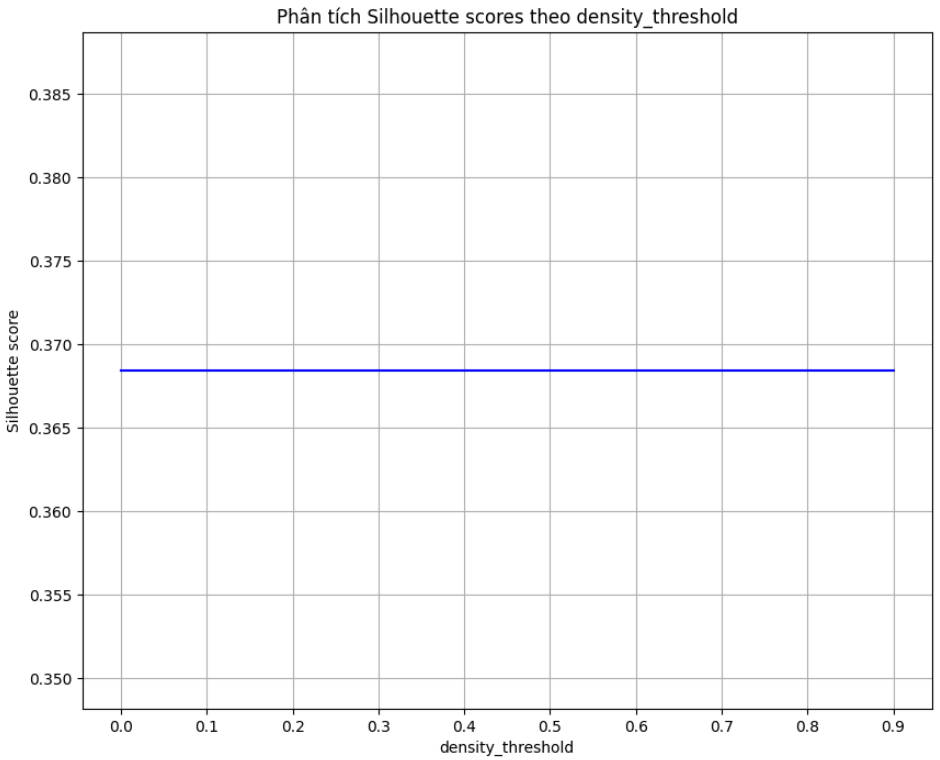
Thứ hai, đối với "amount_threshold",



Hình 21: Hiệu chỉnh "amount_threshold"

Ở đây, ở giá trị amount_threshold = 1 và amount_threshold = 2 cho ta giá trị silhouette score cao hơn so với các giá trị amount_threshold còn lại, hơn nữa ở giá trị amount_threshold = 3 thì silhouette score có giảm nhưng sau đó lại tăng và không đổi ở các giá trị lớn hơn. Vậy nên, ta sẽ chọn amount_threshold = 2 bởi nếu ta chọn các giá trị lớn hơn cũng sẽ không cải thiện nhiều so với giá trị 2 này.

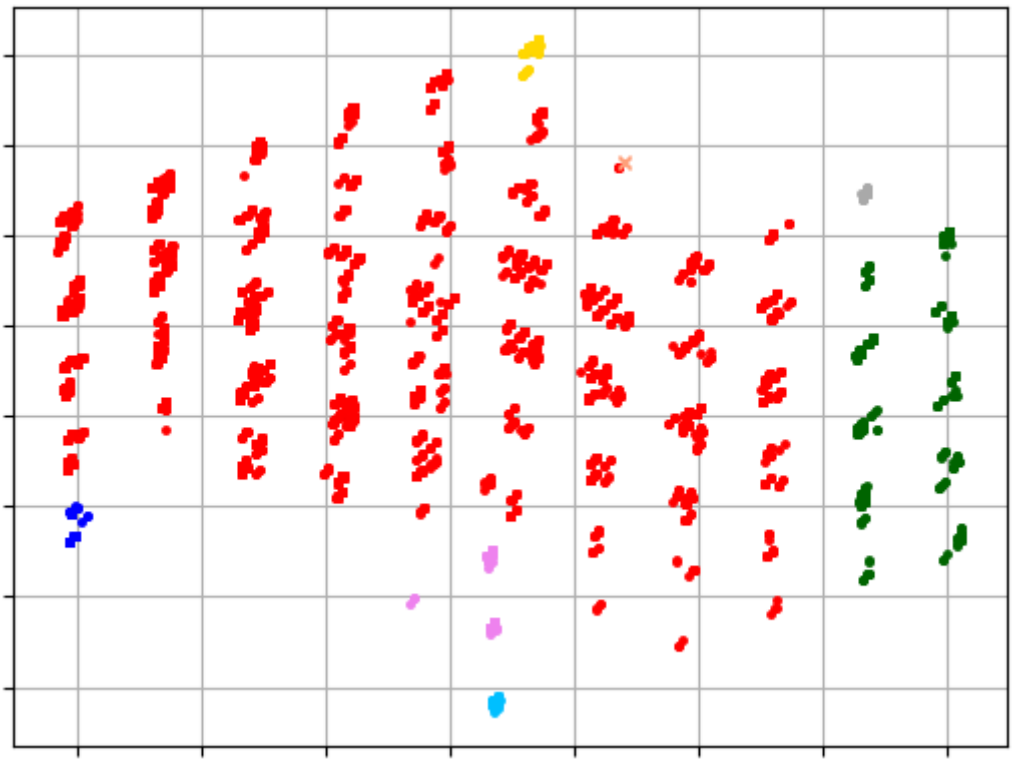
Thứ ba, đối với "density_threshold",



Hình 22: hiệu chỉnh "density_threshold"

Giá trị của silhouette score không đổi với các giá trị của density_threshold và như vậy, ta sẽ chọn density_threshold = 0.0, đây cũng chính là giá trị mặc định của thư viện pyclustering.

Như vậy, sau quá trình hiệu chỉnh siêu tham số, ta đạt được kết quả với các siêu tham số đó là level = 10, amount_threshold = 2 và density_threshold = 0.0. Với các giá trị này ta một lần nữa tiến hành phân cụm với BANG-clustering,



Hình 23: Phân cụm sau hiệu chỉnh siêu tham số

Ta có thể thấy rằng, số cụm đã giảm so với trước giai đoạn hiệu chỉnh siêu tham số. Nhưng liệu kết quả này có thể được dùng cho phân cụm khách hàng hay không. Ta sẽ tiến hành đánh giá.

CHƯƠNG VI: ĐÁNH GIÁ

6.1. Độ phức tạp

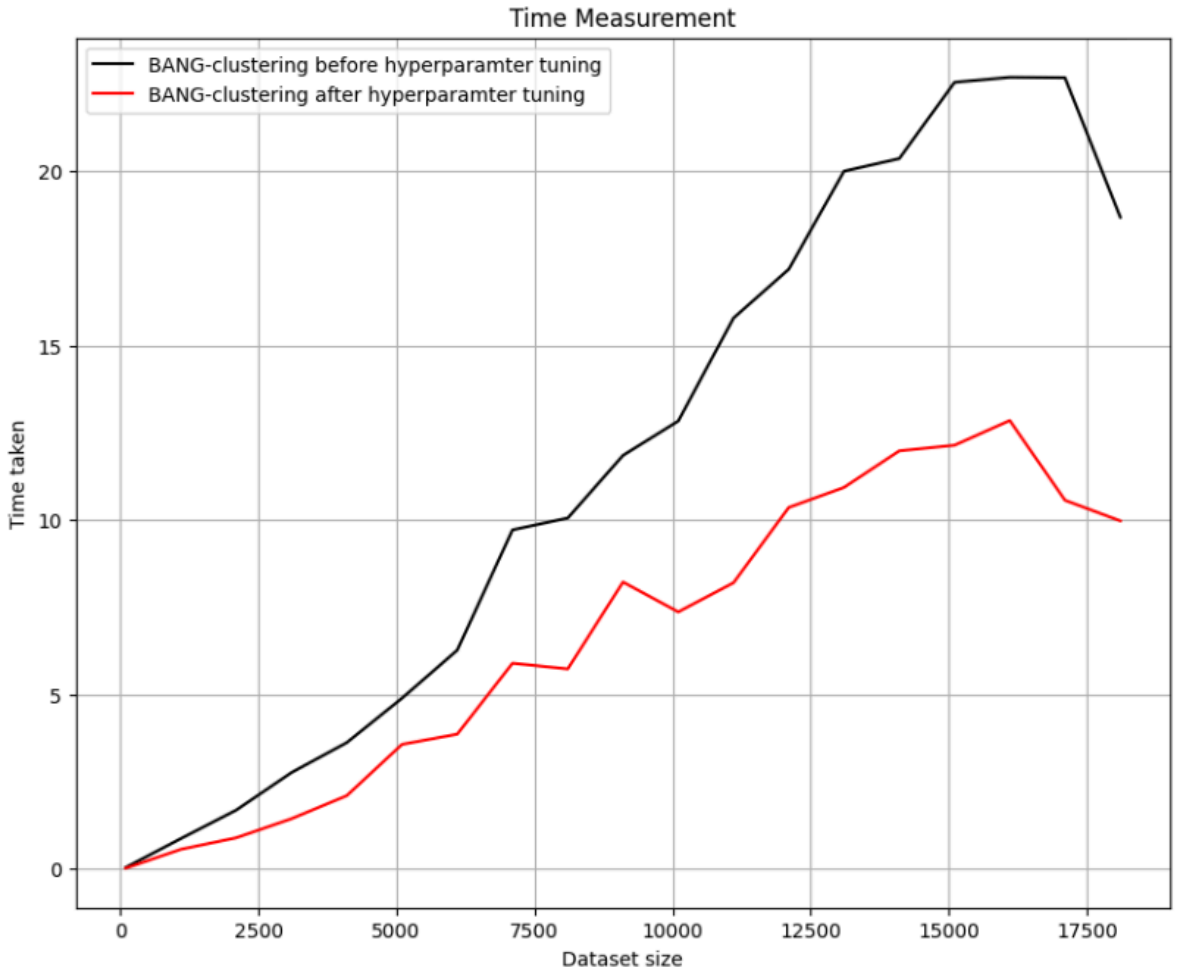
Ở đây, ta sẽ tính toán độ phức tạp về không gian và thời gian. Trước nhất đó là độ phức tạp về không gian, BANG-clustering sử dụng BANG-structure để lưu trữ các điểm trong không gian thông qua một thư mục lưới, thư mục lưới này gồm hai thành phần, đó là một mảng k chiều và một mảng 1 chiều. Đối với thành phần đầu tiên của BANG-structure, nếu gọi len() là hàm trả về số phần tử trong một chiều bất kỳ của mảng này, khi đó ta sẽ có độ phức tạp khi lưu mảng k chiều là $O(len(k)len(k-1)len(k-2)\dots len(1)len(0))$, đặt $size = len(k)len(k-1)len(k-2)\dots len(1)len(0)$, vậy ta có O(size). Thành phần thứ hai đó là một mảng một chiều, mảng này có các phần tử đóng vai trò như con trỏ tới khối dữ liệu, nếu số phần tử của mảng này là p, vậy ta có O(p). Hơn nữa, để lưu trữ cấu trúc lưới của các vùng thể hiện hàng xóm như ở hình 6, thuật toán dùng một cây để làm điều này, nếu số phần tử của cây là T thì độ phức tạp sẽ là O(T). Như vậy, độ phức tạp về không gian là $O(size) + O(p) + O(T) = O(size)$.

Về độ phức tạp thời gian, ta sẽ đo lường thông qua các bước mà BANG-clustering thực hiện để phân cụm. Đầu tiên, nếu gọi số lần phân chia tối đa một khối là *level*, số khối là *m*, vậy ở bước phân chia không gian giá trị, ta có số khối tối đa sinh ra là $level \times m$, vậy $O(level \times m)$. Ở bước tiếp theo, BANG-clustering sẽ tiến hành gán mỗi điểm dữ liệu vào khối tương ứng, gọi số lượng điểm dữ liệu là *n*, bước này ta có O(n). Sau đó, thuật toán sẽ tính toán Density Index cho mỗi khối, vậy ta đạt được O(m). Khi đã tính Density Index cho các khối, BANG-clustering sẽ sắp xếp các khối giảm dần theo Density Index, giả sử thuật toán chọn cách sắp xếp các phần tử có độ phức tạp lớn nhất, khi đó ta có $O(n^2)$ [12]. Sắp xếp các khối theo Density Index hoàn tất, thì BANG-clustering sẽ bắt đầu từ khối có Density Index cao nhất, tìm hàng xóm và sau đó hợp nhất để tạo cụm, quá trình sẽ được lặp lại và sẽ cần $O(m^2)$. Vậy, độ phức tạp về thời gian là $O(level \times m) + O(n) + O(m) + O(n^2) + O(m^2) = O(n^2 + m^2)$.

6.2. Thời gian thực chạy

Về tiêu chí thời gian, ta sẽ đánh giá giải thuật phân cụm khi tăng dần kích cỡ tập dữ liệu.

Thời gian của quá trình phân cụm trước và sau khi hiệu chỉnh siêu tham số được trực quan trên biểu đồ dưới:



Hình 24: Đồ thị thời gian phân cụm theo kích cỡ tập dữ liệu.

Quan sát đồ thị Hình 24, khi thực hiện phân cụm với tập tham số mặc định, thời gian chạy được biểu diễn bằng đường màu đen trên đồ thị, ta thấy rằng giải thuật có thời gian chạy thay đổi đồng biến với kích cỡ tập dữ liệu tuy nhiên đến một ngưỡng nhất định, thời gian chạy có xu hướng giảm đi. Cụ thể, khi tăng kích cỡ bộ dữ liệu, thời gian chạy cũng tăng dần có thể lên đến hơn 20 giây, tuy nhiên khi lên đến khoảng 17000 mẫu, thời gian chạy bắt đầu có xu hướng giảm. Tương tự như vậy, sau khi hiệu chỉnh siêu tham số, khi đánh giá thời gian chạy của giải thuật cũng có dáng điệu đồ thị được biểu diễn bằng đường màu đỏ trên đồ thị. Ta thấy rằng thời gian chạy sau khi hiệu chỉnh cũng có xu hướng đồng biến với kích cỡ tập dữ liệu nhưng thời gian đã giảm đi đáng kể so với trước khi hiệu chỉnh siêu tham số, chỉ còn khoảng dưới 15 giây cho tập dữ liệu 17000 mẫu.

Để lý giải vì sao cải thiện được thời gian chạy, ta sẽ dựa vào ý nghĩa của sự thay đổi các siêu tham số. Nhắc lại rằng các siêu tham số sau khi hiệu chỉnh có giá trị: level = 10, amount_threshold = 2 và density_threshold = 0.0. Đối với tham số level, khi giảm từ 11 xuống 10 tức giảm số lần phân chia một khối từ 11 còn 10 lần, do đó một khối sẽ được phân chia ít hơn, từ đó phân cụm sẽ nhanh hơn và độ dày đặc của mỗi cụm sẽ tốt hơn. Đối với tham số density_threshold = 0.0, nghĩa rằng những khối có mật độ = 0.0, hay khối rỗng sẽ bị coi là nhiễu. Đối với tham số amount_threshold = 2, tức những block có 2 điểm dữ liệu sẽ bị coi là nhiễu và do đó không cần tiếp tục phân chia, giảm bớt quá trình tính toán. Nhìn chung, thời gian thực hiện của giải thuật đã cải thiện đáng kể.

6.3. Silhouette score

Thuật toán	Silhouette score
Trước khi hiệu chỉnh siêu tham số	0.4828
Sau khi hiệu chỉnh siêu tham số	0.0330

Bảng 5: Kết quả Silhouette score trước và sau khi hiệu chỉnh siêu tham số

Quan sát kết quả phân cụm thông qua hệ số Silhouette ở bảng 5 thì trước khi hiệu chỉnh siêu tham số, giá trị đạt được là 0.4828, cho thấy mức độ phân tách giữa các cụm ở mức tương đối tốt. Tuy nhiên, sau khi hiệu chỉnh siêu tham số, hệ số Silhouette giảm mạnh chỉ còn 0.0330, phản ánh rằng các cụm trở nên kém phân biệt hơn và có xu hướng chồng lấn nhau đáng kể.

Để lý giải cho việc vì sao sau khi hiệu chỉnh tham số thì hệ số lại giảm, cần xem xét ý nghĩa của những thay đổi trong các siêu tham số đã hiệu chỉnh. Cụ thể, tham số level trước khi hiệu chỉnh là 11 và sau khi hiệu chỉnh là 10. Dựa vào nghiên cứu ở phần trên, nhóm nhận thấy rằng, ở level càng cao thì số cụm sẽ tăng lên và silhouette score tăng lên, tuy silhouette score tăng lên dẫu là dấu hiệu tốt nhưng sẽ không đúng trong mọi trường hợp, vì như vậy khả năng cao mô hình của nhóm sẽ rơi vào trường hợp quá khớp. Đối với các level thấp hơn, ở mức level thấp thì số cụm gần như là bằng 1, nghĩa là chỉ có một cụm cho tất cả các điểm dữ liệu. Và với lập luận trên, nhóm sẽ chọn level = 10. Chỉ có level = 10 thì kết quả chia cụm nó mới mang ý nghĩa kinh tế để phân tích.

6.4. So sánh với Kmeans và HAC

Ở đây, đối với HAC thì nhóm sẽ chọn hướng tiếp cận Agglomerative Clustering để so sánh với BANG-clustering.

6.4.1. Về thời gian chạy thuật toán

Hình 25: Kết quả đánh giá thời gian chạy của ba thuật toán Kmeans, BANG và HAC

Theo như hình trên, ta thấy một cách tổng quan rằng khi độ lớn của dữ liệu tăng lên thì thời gian chạy của cả hai thuật toán BANG, HAC cũng tăng theo, duy chỉ có Kmeans là thuật toán không thay đổi thời gian chạy khi số điểm dữ liệu trong bộ dữ liệu tăng lên.

Đối với BANG, khi độ dữ liệu tăng lên hơn 17500 điểm dữ liệu thì BANG là thuật toán có thời gian chạy lâu nhất - khoảng 26 giây. Đây là kết quả thấp hơn hẳn so với hai thuật toán còn lại là Kmeans và HAC với thời gian chạy lần lượt là xấp xỉ 0 giây và khoảng 10 giây. Mặt khác, so với Kmeans thì thời gian chạy của BANG lại tiệm cận không thay đổi tuyến

tính khi số điểm dữ liệu từ khoảng 4000 trở lên. Điều này cũng xảy ra tương tự khi thực hiện chạy thuật toán HAC, tuy nhiên, với HAC, hình dạng đồ thị thay đổi ổn định hơn ở khác khoảng từ 7000 đến 12000 điểm dữ liệu.

Để lý giải cho việc BANG có thời gian chạy lâu hơn so với Kmeans và HAC thì có lẽ ta phải dựa vào cách thức mà BANG phân cụm. BANG phân cụm bằng cách chia không gian giá trị để tạo thành các khối, sau đó lại trải qua quá trình tính toán Density Index cho các khối, tiếp đến sắp xếp các khối theo Density Index, tìm hàng xóm và sau đó hợp nhất để tạo thành các cụm. Quá trình này có thể nói phức tạp và cần thời gian tính toán hơn và có lẽ vì thế mà BANG cần thời gian nhiều hơn so với hai thuật toán còn lại.

6.4.2. Silhouette Score

Thuật toán	Silhouette score
Kmeans	0.45
HAC	0.40
BANG - sau khi hiệu chỉnh tham số	0.03

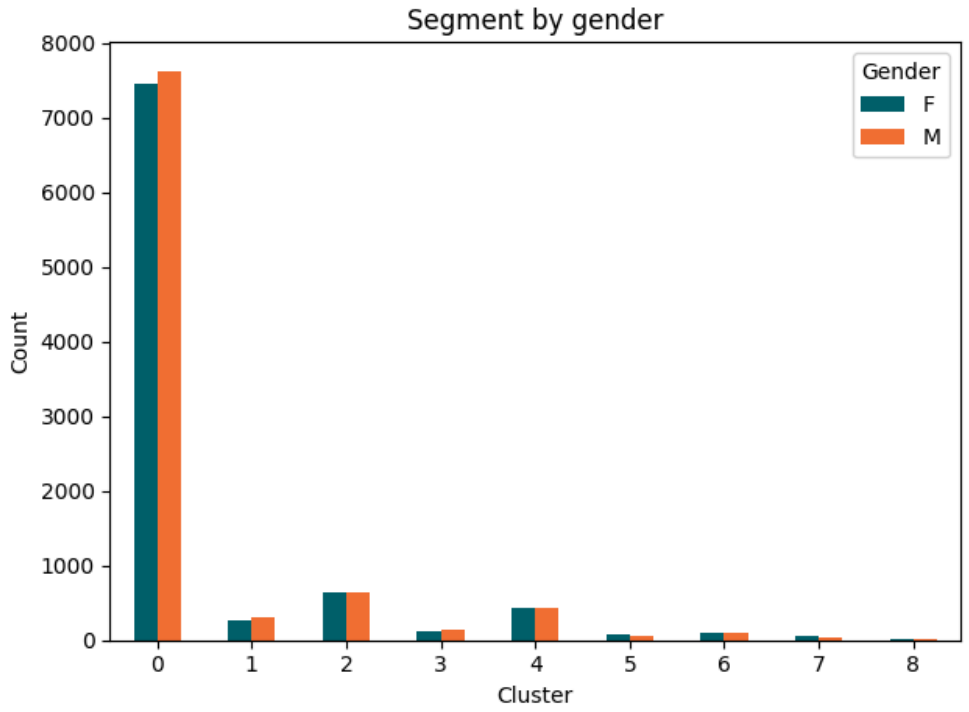
Bảng 6: Kết quả Silhouette Score của ba thuật toán Kmeans, HAC và BANG

Qua kết quả của bảng trên, ta thấy BANG là thuật toán có điểm Silhouette Score thấp nhất trong ba thuật toán. Điều này có nghĩa là BANG là thuật toán phân cụm không tốt khi áp dụng trên bộ dữ liệu có độ phân tán cao. Cách hoạt động của BANG vốn dĩ là kết nối các vùng hàng xóm gần nhau để tạo thành một cụm, mà khi dữ liệu bị phân tán mạnh, nghĩa là các điểm dữ liệu nằm rải rác, không tập trung rõ ràng thành từng nhóm cụ thể thì thuật toán BANG có xu hướng kết nối các vùng lân cận gần nhau, ngay cả khi những vùng này không thực sự thuộc về cùng một cụm, điều này dẫn tới việc phân cụm không phù hợp, phân cụm sai, làm giảm hiệu quả phân cụm và dẫn đến điểm Silhouette Score thấp.

CHƯƠNG VII: ỨNG DỤNG

7.1. Phân cụm khách hàng

Với kết quả đạt được ở các chương trước, ta sẽ áp dụng thuật toán BANG-clustering vào phân cụm khách hàng. Kết quả đạt được sẽ nằm trong file "application.csv", ở đây ta sẽ có thêm một thuộc tính đó là "cluster_final". Thuộc tính này sẽ cho ta biết kết quả phân cụm đối với mỗi điểm dữ liệu. Trước tiên, ta hãy phân cụm khách hàng theo giới tính,



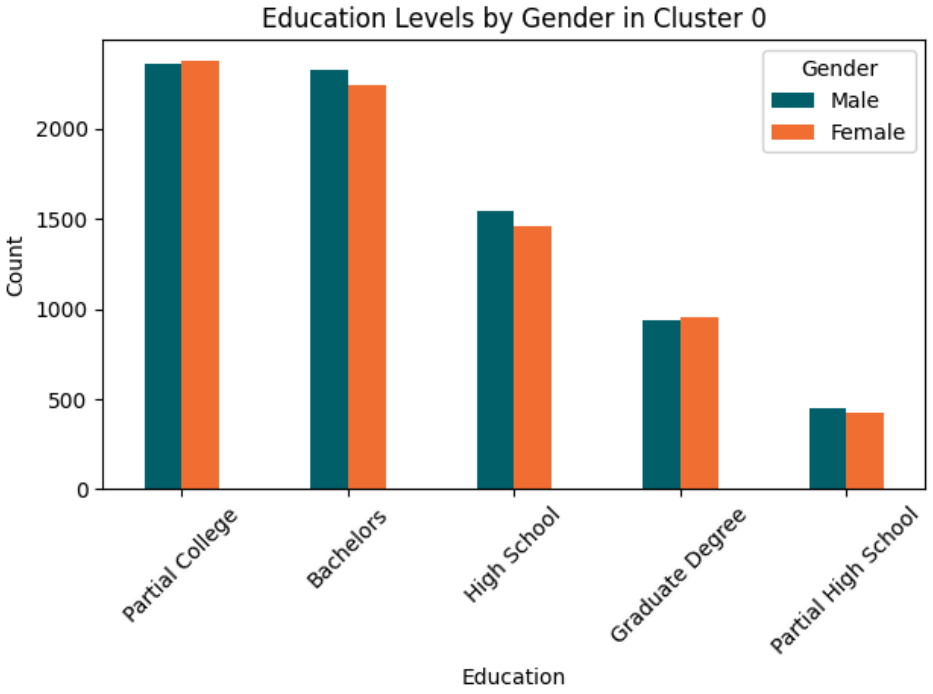
Hình 26: Kết quả phân cụm theo giới tính

Với kết quả ở hình trên, dễ thấy rằng cụm chiếm số lượng nhiều nhất trong tất cả các cụm đó là cụm 0 (Cluster = 0). Vì vậy, ta sẽ chọn cụm này để tiến hành phân tích sâu hơn.

Nhóm thắc mắc rằng liệu có sự khác nhau về thu nhập giữa nam và nữ trong cluster 0 hay không, kết quả thu được khá thú vị,

Hình 27: Thu nhập giữa nam và nữ trong cluster 0

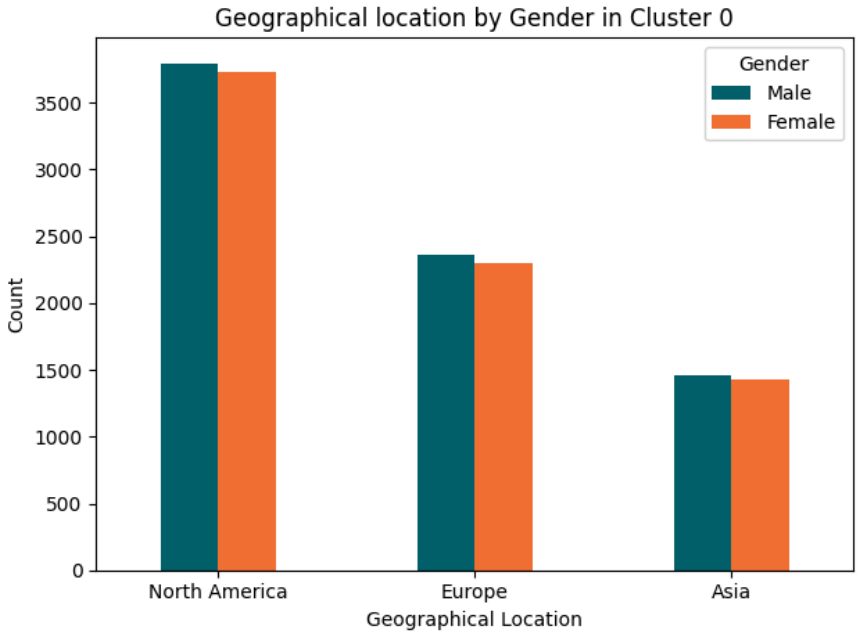
Lý do mà nhóm vẽ biểu đồ hộp là bởi: (i) nhóm muốn xem xét liệu có sự khác nhau về thu nhập giữa nam và nữ hay không, (ii) nhóm muốn biết phân phối của các phần tử ngoại lai nếu có giữa nam và nữ. Với câu hỏi đầu tiên, dường như không có sự khác biệt giữa nam và nữ, chênh lệch về khoảng thu nhập của hai nhóm khá nhỏ. Các phần tử ngoại lai đại diện cho những mẫu có mức thu nhập cao, nếu một nhóm tồn tại nhiều phần tử ngoại lai hơn so với nhóm còn lại thì ta sẽ cân nhắc đến việc phân biệt giữa hai nhóm do thu nhập cao báo hiệu khả năng tài chính cao và từ đó mức chi trả cũng cao hơn. Nhưng, biểu đồ hộp của hai nhóm khá tương đồng, có lẽ giữa hai nhóm này không có sự khác biệt về thu nhập. Nếu như đã không có sự khác biệt, thì nhóm sẽ xác định một khoảng thu nhập chung cho cả hai nhóm đó là [17000, 130000]. Liệu giữa nam và nữ với thu nhập trong đoạn [17000, 130000] ta sẽ nên hướng đến nhóm người có trình độ học vấn như thế nào,



Hình 28: Trình độ học vấn giữa nam và nữ

Đối trình độ học vấn mà nói, "Partial College" và "Bachelors" có quy mô gần như là giống nhau. Và không chỉ riêng hai trình độ này mà các trình độ học vấn cũng có kết quả tương tự, và có lẽ điều này sẽ giải thích nguyên nhân đằng sau việc thu nhập giữa nam và nữ không có chênh lệch đáng kể. Nhưng ở đây, mục tiêu cốt lõi vẫn là phân cụm khách hàng, nên đặc điểm về trình độ học vấn mà ta chọn sẽ là "Partial College" và "Bachelors" do hai nhóm có quy mô lớn nhất so với những nhóm khác.

Một điều mà có thể sẽ tác động đến khả năng chi tiêu cho một món hàng nữa đó là khu vực mà một khách hàng sinh sống. Mỗi khu vực sẽ có văn hóa khác nhau, văn hóa tác động đến con người nơi họ sống, vậy nên mà văn hóa chi tiêu cũng sẽ khác. Do vậy mà đây cũng là một yếu tố mà ta nên xem xét đến,



Hình 29: Lục địa sinh sống giữa nam và nữ

Có vẻ tệp khách hàng đang tập trung vào những người sinh sống ở Bắc Mỹ (North America). Tuy châu Âu (Europe) và châu Á (Asia) tuy số lượng lớn nhưng khu vực Bắc Mỹ có phần nhỉnh hơn hai khu vực này. Do đó, ta sẽ chọn tệp khách hàng đang sinh số ở Bắc Mỹ.

7.2. Cụm mục tiêu

Thông qua phần 7.1, ta đã cùng nhau đi tìm cụm khách hàng cho doanh nghiệp. Ta xem xét về giới tính giữa các khách hàng liệu có sự khác nhau hay không, thu nhập giữa họ ra sao, trình độ học vấn như thế nào, khu vực sinh sống có gì khác. Với kết quả và thảo luận ở phần trên. Ta sẽ có kết luận về cụm khách hàng như sau, cụm khách hàng mà ta sẽ hướng đến sẽ không có sự khác nhau giữa nam và nữ, đồng thời cũng không có sự khác biệt về thu nhập. Nhưng ta sẽ hướng đến những người có trình độ học vấn ở bậc cử nhân hoặc hiện tham gia vào một phần của chương trình đại học, có thể là học bán thời gian, hoặc chưa hoàn thành toàn bộ quá trình học để nhận bằng đại học. Cụ thể hơn, đó là những người hiện đang sinh sống ở khu vực Bắc Mỹ.

CHƯƠNG VIII: KẾT LUẬN

8.1. Về BANG-clustering - Model Deployment

Qua bài báo cáo trên, nhóm tác giả đã xây dựng thành công mô hình phân cụm khách hàng dựa trên giải thuật BANG-clustering. Trải qua các lần thực nghiệm và kiểm thử trên tập dữ liệu Cleaned Contoso, kết quả nghiên cứu là mô hình phân cụm khách hàng BANG với các siêu tham số level = 10, density_threshold = 0.0 và amount_threshold = 2. Đồng thời dựa vào tiêu chí Silhouette score để đánh giá chất lượng phân cụm, mô hình cũng cho ra kết quả 0.03, tuy đây dường như là một kết quả có thể nói là chưa tốt nhưng đối với tập dữ liệu Cleaned Contoso và BANG-clustering thì có lẽ đây là một trong những kết quả chấp nhận được. Nhóm tác giả cũng nhận thấy rằng, BANG-clustering sẽ phù hợp với tập dữ liệu có phân phối tập trung và tạo thành một hình dạng cụ thể nào đó hơn.

8.2. Về kết quả đạt được

Bài báo cáo này đã nghiên cứu thuật toán phân cụm BANG và ứng dụng thuật toán vào phân loại nhóm khách hàng. Mục đích của bài đồ án là để hiểu rõ hơn về thuật toán phân cụm BANG cũng như cách ứng dụng vào trong các bài toán phân cụm. Bài báo cáo đã đạt được một số kết quả như sau, thuật toán phân cụm BANG đã phân chia khách hàng thành chín nhóm khách hàng chính, trong đó nhóm nghiên cứu xác định cụm mục tiêu là cụm "Cluster 0". Trong cụm này, nhóm xác định cụm khách hàng đó là những người có trình độ học vấn là "Partial College", "College"; hiện đang sinh sống ở khu vực Bắc Mỹ và giữa những khách hàng này sẽ không có sự khác biệt giữa nam và nữ về thu nhập.

Dựa vào kết quả đạt được trong phần nghiên cứu ứng dụng của thuật toán thì nhóm tác giả cũng đưa ra một số nhận xét về thuật toán như sau: Đối với bộ dữ liệu mà điểm dữ liệu tạo thành hình dạng nào thì kết quả phân cụm của thuật toán BANG là tốt. Tuy nhiên khi gặp các bộ dữ liệu với các đặc điểm như mật độ khác nhau, rải đều trong không gian thì kết quả phân cụm không tốt bằng Kmeans với HAC (Agglomeratie Clustering).

8.3. Về hướng phát triển

Tổng kết, giải thuật BANG-clustering là phương pháp phân cụm hiệu quả đối với tập dữ liệu kích cỡ lớn, bởi cách tiếp cận của giải thuật dựa trên việc chia nhỏ không gian dữ liệu

thành các khối, hơn nữa các siêu tham số cho phép người dùng điều chỉnh các tính chất của giải thuật để linh hoạt phân chia khối, phát hiện các khối có mật độ thấp và nhận diện nhiễu trong dữ liệu, từ đó nâng cao kết quả phân cụm và giảm bớt thời gian tính toán.

Tuy nhiên, giải thuật vẫn tồn tại một số hạn chế nếu xét tới thư viện hỗ trợ giải thuật và khả năng biểu diễn trực quan giải thuật. Thứ nhất, khi xử lý các tập dữ liệu đa chiều cần phải biến đổi về không gian phù hợp sao cho số chiều dữ liệu phản ánh đúng và đầy đủ thông tin để có thể biểu diễn trực quan sao cho rõ ràng nhất, thứ hai, các siêu tham số của pyclustering cần phải lựa chọn kĩ lưỡng phù hợp với tập dữ liệu và mục đích phân cụm để tránh sai lệch trong việc nhận diện các khối và nhiễu, ngoài ra trong một số trường hợp cụm dữ liệu có hình dạng phức tạp, việc phân chia theo khối hình chữ nhật sẽ trở nên không tối ưu. Hơn nữa, nhóm cũng chưa đánh giá được mô hình phân cụm trên các chỉ số khác như Rand Index, Jaccard Index,...

Trong tương lai, nếu có cơ hội tiếp tục nghiên cứu đề tài liên quan về giải thuật BANG-clustering, nhóm tác giả đề xuất tiếp tục xây dựng, cải thiện các hạn chế đã nêu trên của giải thuật, đồng thời cải thiện cách đánh kết quả phân cụm của mô hình trên nhiều chỉ số hơn và nghiên cứu mở rộng ứng dụng giải thuật BANG vào các vấn đề thực tiễn khác.

TÀI LIỆU THAM KHẢO

[1] Schikuta, E., & Erhart, M. *The BANG-clustering system: Grid-based data analysis.* Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0052867

[2] Warnekar, C. S., & Krishna, G. (1979). *A heuristic clustering algorithm using union of overlapping pattern-cells.* Pattern Recognition, 11(2), 85 - 93. https://doi.org/10.1016/0031-3203(79)90054-2

[3] Schikuta, E., & Erhart, M. (1998). BANG-Clustering: A novel grid-clustering algorithm for huge data sets. In *Lecture Notes in Computer Science* (pp. 867–874). Springer Berlin Heidelberg. http://dx.doi.org/10.1007/bfb0033313

[4] Madhulatha, T. S. (2012). An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

[5] Fung, G. (2001). A comprehensive overview of basic clustering algorithms.

[6] Djouzi, K., & Beghdad-Bey, K. (2019, June). A review of clustering algorithms for big data. In *2019 International Conference on Networking and Advanced Systems (ICNAS)* (pp. 1-6). IEEE.

[7] Mahdi, M. A., Hosny, K. M., & Elhenawy, I. (2021). Scalable clustering algorithms for big data: A review. *IEEE Access*, *9*, 80015-80027.

[8] Trần, H. V., Nguyễn, T. T., & Trần, T. T. (2017). Một cải tiến thuật toán K-Means song song sử dụng phương pháp lấy mẫu.

[9] Dash, B., Mishra, D., Rath, A., & Acharya, M. (2010). A hybridized K-means clustering approach for high dimensional dataset. *International Journal of Engineering, Science and Technology*, *2*(2), 59-66

[10] Nievergelt, J., Hinterberger, H., & Sevcik, K. C. (1984b). The grid file. *ACM Transactions on Database Systems*, *9*(1), 38–71. https://doi.org/10.1145/348.318586

[11] Freeston, M. (1987). The BANG file: A new kind of grid file. *Proceedings of the*

1987 ACM SIGMOD International Conference on Management of Data - SIGMOD '87, 260–269. http://dx.doi.org/10.1145/38713.38743

[12] Chauhan, Y., & Duggal, A. (2020, September 17). *Different Sorting Algorithms comparison based upon the Time Complexity*. Unknown. https://www.researchgate.net/publication/344280789_Different_Sorting_Algorithms_comparison_based_upon_the_Time_Complexity

ĐÁNH GIÁ CÔNG VIỆC

1. Nguyễn Đôn Đức

Công việc	Mức độ hoàn thành
Tìm hiểu thuật toán, phương pháp đánh giá và tìm tập dữ liệu	100%
Tiền xử lý dữ liệu và code thuật toán BANG-clustering	100%
Viết luận: Chương II (2.2.3, 2.4), Chương VI (6.2), Chương VIII (8.1, 8.3)	100%
Code đánh giá mô hình trước khi hiệu chỉnh siêu tham số với thời gian và silhouette score	100%
Code Hyperparameter Tuning và đánh giá mô hình sau khi hiệu chỉnh siêu tham số với thời gian và silhouette score	100%
Check chính tả - Chương I đến IV	100%

2. Đặng Thị Thu Hiền

Công việc	Mức độ hoàn thành
Tìm hiểu thuật toán, phương pháp đánh giá và tìm tập dữ liệu	100%
Tiền xử lý dữ liệu và code thuật toán BANG-clustering	100%
Viết luận: Chương I, III , IV (4.3, 4.4)	100%
Slides trình chiếu: I, II, III, IV, VI	100%

3. Đỗ Thanh Hoa

Công việc	Mức độ hoàn thành
Tìm hiểu thuật toán, phương pháp đánh giá và tìm tập dữ liệu	100%
Tiền xử lý dữ liệu và code thuật toán BANG-clustering	100%
Viết luận: Chương II (2.1, 2.2.1), Chương IV (4.1, 4.2), Chương VI (6.4)	100%
Code Kmeans và HAC (Agglomerative Clustering)	100%
Slides trình chiếu: Chương V, VII, VIII	100%

4. Nguyễn Trương Hoàng

Công việc	Mức độ hoàn thành
Tìm hiểu thuật toán, phương pháp đánh giá và tìm tập dữ liệu	100%
Tiền xử lý dữ liệu và code thuật toán BANG-clustering	100%
Viết luận: Chương II (2.2.2.3, 2.2.2.4, 2.3), Chương VI (6.3), Chương VIII (8.2)	100%
Code Hyperparameter Tuning và đánh giá mô hình sau khi hiệu chỉnh siêu tham số với thời gian và silhouette score	100%

Check chính tả - Chương V đến VIII	100%

5. Nguyễn Huy Hoàng

Công việc	Mức độ hoàn thành
Tìm hiểu thuật toán, phương pháp đánh giá và tìm tập dữ liệu	100%
Tiền xử lý dữ liệu và code thuật toán BANG-clustering	100%
Viết luận: Chương II (2.2.2.1, 2.2.2.3), Chương V, Chương VI (6.1), Chương VII	100%
Code đánh giá mô hình trước khi hiệu chỉnh siêu tham số với thời gian và silhouette score	100%
Format báo cáo	100%