

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC KINH TẾ TP. HỒ CHÍ MINH



ĐỒ ÁN MÔN HỌC
MÁY HỌC

ÁP DỤNG THUẬT TOÁN MEAN SHIFT
CHO BỘ DỮ LIỆU SHOP CUSTOMER DATA

Họ và tên	MSSV	Lớp
Nguyễn Thị Tuyết	31211027684	DS001
Vũ Nguyễn Thảo Vi	31211027686	DS001
Nguyễn Quốc Việt	31211027687	DS001
Bùi Quốc Việt	35221020290	LT27.1
Nguyễn Nhật Thảo Vy	31211025542	DS001

GVHD: TS. Nguyễn An Tê

TP. Hồ Chí Minh, Ngày 14 tháng 12 năm 2023

Mục lục

1 CHƯƠNG 1: TỔNG QUAN	4
1.1 Vai trò và ý nghĩa của Máy học	4
1.2 Giới thiệu đề tài	4
1.3 Mục đích đề tài	4
2 CHƯƠNG 2: THUẬT TOÁN MEANSHIFT	5
2.1 Bài toán phân cụm	5
2.2 Thuật toán Meanshift	6
2.2.1 Hàm mật độ xác suất	6
2.2.2 Kernel Density Estimation	6
2.2.3 Cơ sở lý thuyết của thuật toán Meanshift	9
3 CHƯƠNG 3: PHÂN TÍCH VÀ TRỰC QUAN HOÁ DỮ LIỆU	10
3.1 Tổng quan bộ dữ liệu	10
3.2 Tiền xử lý dữ liệu	12
3.2.1 Điều chỉnh các cột	12
3.2.2 Xử lý dữ liệu bị thiếu	13
3.2.3 Xử lý ngoại lai (Outliers)	14
3.3 Trực quan hóa dữ liệu	17
3.3.1 Thống kê mô tả	17
3.3.2 Tương quan giữa các biến định lượng	18
3.4 Áp dụng thuật toán Mean shift cho bộ dữ liệu Shop Customer Data . . .	29
3.4.1 Chính dạng dữ liệu	29
3.4.2 Giảm chiều dữ liệu bằng phương pháp phân tích thành phần chính PCA	30
3.4.3 Xây dựng mô hình Mean Shift	33
3.5 Lựa chọn bandwidth dựa trên phương pháp Grid Search	34
3.6 Phương pháp ELBOW	36
4 CHƯƠNG 4: ĐÁNH GIÁ THUẬT TOÁN PHÂN CỤM	39
4.1 Cơ sở lý thuyết	39
4.2 Độ đo Silhouette Score	40
4.3 So sánh Mean Shift với các thuật toán phân cụm khác	41
5 CHƯƠNG 5: KẾT LUẬN VÀ Ý NGHĨA	44

Danh sách hình vẽ

1	Hàm mật độ phân phối xác suất của phân phối Gaussian	7
2	Các loại Kernel khác. Nguồn: Wikipedia	8
3	Minh họa KDE. Nguồn: UC Davis Statistics	8
4	Box plot các biến định lượng trong bộ dữ liệu	15
5	Phân phối của từng thuộc tính trong bộ dữ liệu	15
6	Biểu đồ tương quan giữa các biến định lượng	18
7	Biểu đồ tròn thể hiện tỉ lệ giới tính	20
8	Quy trình thực hiện PCA. Nguồn: <i>Machine Learning Cơ Bản</i>	30
9	Kết quả phân cụm của thuật toán Mean Shift với các quantile khác nhau	35
10	Kết quả phân cụm dữ liệu với số cụm bằng 2 của thuật toán Mean Shift .	37
11	Kết quả phân cụm dữ liệu với số cụm bằng 2 của thuật toán Mean Shift .	39
12	Kết quả phân cụm dữ liệu của các thuật toán theo điểm Silhouette	42
13	Trực quan hóa điểm Silhouette các thuật toán phân cụm	42
14	Thời gian huấn luyện các thuật toán phân cụm	43

Danh sách bảng

1	Bảng mô tả thuộc tính khách hàng	11
---	--	----

LỜI MỞ ĐẦU

Trong thời đại hiện nay, con người không chỉ tạo ra mà còn thu thập một lượng dữ liệu lớn chưa từng có. Dữ liệu này mang lại cơ hội hiểu biết về người tiêu dùng một cách sâu sắc hơn, mở ra những triển vọng mới mẻ cho các doanh nghiệp. Từ việc chuyển đổi dữ liệu thành thông tin và từ thông tin trở thành tri thức, chúng hỗ trợ cho việc ra quyết định của doanh nghiệp. Các doanh nghiệp ngày nay sử dụng thông tin và insights từ dữ liệu để liên tục cải thiện chiến lược bán hàng và tiếp thị.

Trong bối cảnh đó, Machine Learning ngày càng trở nên quan trọng, đó là công cụ giúp chúng ta sử dụng dữ liệu hiệu quả để hỗ trợ cho việc ra quyết định. Machine Learning là một nhánh của trí tuệ nhân tạo (AI), tập trung vào việc sử dụng dữ liệu và áp dụng thuật toán để liên tục "học hỏi," từ đó cải thiện độ chính xác và đồng thời giúp chúng ta hiểu rõ hơn về thế giới xung quanh.

Đó là lý do nhóm sử dụng thuật toán Mean Shift để áp dụng vào bài toán phân cụm khách hàng dựa trên bộ dữ liệu Shop Customer Data. Mục tiêu chính là xây dựng một mô hình giúp mô tả các phân khúc khách hàng dựa trên các đặc điểm quan trọng như thu nhập, mức chi tiêu, tuổi và nhiều yếu tố khác. Trong quá trình làm đồ án môn học, nhóm chúng em khó tránh khỏi các hạn chế, sai sót. Nhóm chúng em mong sẽ nhận được lời nhận xét của thầy để cải thiện các điểm này.

Dưới sự hướng dẫn của Thầy Nguyễn An Tế, chúng em đã nhận được sự truyền cảm hứng và kiến thức quý báu cho hành trình học tập và sự nghiệp của mình trong lĩnh vực Máy Học và Khoa học Dữ liệu. Thầy luôn dành thời gian và tâm huyết để giải thích những khái niệm phức tạp một cách dễ hiểu và gần gũi, giúp chúng em hiểu sâu hơn về bản chất của mỗi vấn đề. Dù thỉnh thoảng khắt khe, chúng em hiểu rằng điều đó là cần thiết để phát triển bản thân và tiến xa hơn trong sự nghiệp. Chúng em chân thành cảm ơn thầy vì sự nhiệt tình và tận tâm giảng dạy không ngừng thầy dành cho chúng em.

1 CHƯƠNG 1: TỔNG QUAN

1.1 Vai trò và ý nghĩa của Máy học

Máy học (*Machine Learning*) là một nhánh quan trọng của lĩnh vực Trí tuệ nhân tạo, Khoa học máy tính cũng như là Khoa học dữ liệu. Hoạt động chính là nghiên cứu cách thiết kế các thuật toán và hệ thống có khả năng học hỏi từ dữ liệu và cải thiện hiệu suất của chúng. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để phân loại, dự đoán hoặc phân cụm nhằm khám phá những hiểu biết quan trọng trong quá trình khai thác dữ liệu. Những hiểu biết này sẽ thúc đẩy và hỗ trợ cho việc ra quyết định.

Máy học có vai trò và ý nghĩa quan trọng trong nhiều ứng dụng thực tế, như nhận dạng khuôn mặt, phát hiện đối tượng, hệ thống gợi ý sản phẩm, thị giác máy tính, v.v... Nó cũng giúp cho chúng ta hiểu sâu hơn về các quy luật và mẫu trong tự nhiên, xã hội và khoa học. Trong bối cảnh dữ liệu lớn tiếp tục mở rộng và phát triển, máy học càng có vai trò quan trọng và được ứng dụng vào nhiều lĩnh vực.

1.2 Giới thiệu đề tài

Với bộ dữ liệu Shop Customer Data, nhóm tìm hiểu và áp dụng thuật toán Mean Shift để phân tích cụm các khách hàng trong bộ dữ liệu này. Phân cụm khách hàng là một bài toán quan trọng trong lĩnh vực kinh doanh và tiếp thị, nhằm phân loại các khách hàng có những đặc điểm, nhu cầu và hành vi mua sắm tương tự nhau vào cùng một nhóm. Bằng cách phân cụm khách hàng, các nhà phân tích có thể đưa ra những hiểu biết có ích về thị trường, khách hàng từ đó hỗ trợ việc nghiên cứu và phát triển kinh doanh.

Mean Shift là một thuật toán phân cụm phổ biến có nhiều ưu và nhược điểm tùy vào đặc trưng của từng bộ dữ liệu được sử dụng. Bên cạnh đó Mean shift có thể áp dụng cho nhiều loại dữ liệu, như hình ảnh, âm thanh, văn bản. Trong đồ án này, thuật toán được áp dụng cho dữ liệu mô tả khách hàng với các thuộc tính nhân khẩu học và một số thông tin khác.

1.3 Mục đích đề tài

Mục tiêu của đồ án là tìm hiểu cơ chế hoạt động của thuật toán phân cụm Mean Shift thông qua việc áp dụng thuật toán lên bộ dữ liệu đã chọn. Từ đó nắm được những ưu và nhược điểm của thuật toán đối với các đặc trưng của bộ dữ liệu này. Mở rộng bằng việc so sánh hiệu quả phân cụm của thuật toán Mean Shift với một vài thuật toán phổ biến khác.

2 CHƯƠNG 2: THUẬT TOÁN MEANSHIFT

2.1 Bài toán phân cụm

Phân cụm (*Clustering*) là một phương pháp học không giám sát được sử dụng để xác định các nhóm hoặc các cụm. Ý tưởng chính là để mô tả đặc điểm của các cụm để khám phá được thông tin hữu ích cho mục đích phân tích. Phương pháp được áp dụng trong nhiều lĩnh vực, bao gồm y học, hóa học, giáo dục, xã hội học ...

Một số phương pháp phân cụm phổ biến bao gồm phương pháp phân cụm dựa trên phân hoạch (*Partitioning approach*). Phương pháp phân cụm này phân loại thông tin thành nhiều nhóm dựa trên đặc điểm và độ giống nhau của dữ liệu. Các nhà phân tích dữ liệu sẽ chỉ định số lượng cụm cho các phương pháp phân cụm. Có thể thử nhiều phân hoạch khác nhau và chọn cách tốt nhất. Một số thuật toán tiêu biểu gồm: K-means [[19]], K-Medoids [[17]], Fuzzy C-Means [[12]].

Ngoài ra, một phương pháp phân cụm khác là các phương pháp phân cụm dựa trên phân cấp (*Hierarchical approach*). Phân cụm dựa trên phân cấp Là một phương pháp phân tích cụm nhằm tìm cách xây dựng hệ thống phân cấp của các cụm, thường chia thành hai loại:

- Hợp nhất (Agglomerative): Đây là cách tiếp cận "bottom-up", bắt đầu với n cụm và lần lượt hợp nhất các cụm tương tự để thu được một cụm ở phân cấp cao hơn.
- Phân chia (Divisive): Đây là cách tiếp cận "top-down": Tất cả các quan sát bắt đầu trong một cụm gồm tất cả quan sát và thực hiện tách thành các cụm khi ở cấp thấp hơn. Một số thuật toán tiêu biểu trong phương pháp này bao gồm Diana, Agnes, BIRCH [30].

Ngoài ra, còn có các phương pháp phân cụm dựa trên mật độ (*Density-based approach*). Trong lớp các thuật toán này, các cụm được xem là các vùng có mật độ dữ liệu dày trong không gian dữ liệu, được phân tách bằng các vùng có mật độ thấp hơn. Một số thuật toán điển hình bao gồm DBSCAN [13], OPTICS [6], Mean shift [16].

Thêm vào đó, các phương pháp phân cụm dựa trên lưới (*Grid-based approach*) là một cách tiếp cận phân cụm sử dụng cấu trúc dữ liệu lưới. Thuật toán chia không gian dữ liệu thành một số ô hữu hạn, tính toán mật độ của từng ô và loại bỏ các ô có mật độ dưới ngưỡng xác định. Một số thuật toán điển hình bao gồm STING [25], WaveCluster [23], CLIQUE [4].

Ngoài ra còn có các thuật toán phân cụm dựa trên mô hình (*Model-based approach*), trong đó, điểm chung là các thuật toán này sẽ phân cụm bằng cách xác định mô hình cho mỗi

cụm. Các thuật toán tiêu biểu gồm EM [11], SOM [18], và COBWEB [14].

Một số đại lượng đặc trưng của cụm:

Trọng tâm (*centroid*): là tập hợp các giá trị trung bình của từng biến độc lập và là một vector:

$$C = \frac{1}{m} \sum_{i=1}^m x_i$$

Bán kính (*radius*) là khoảng cách trung bình từ 1 điểm bất kỳ đến trọng tâm:

$$R = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - C)^2}$$

Đường kính (*diameter*) là khoảng cách xa nhất để bao được cả tập dữ liệu:

$$D = \sqrt{\frac{1}{m(m-1)} \sum_{i \neq j} (x_i - x_j)^2}$$

2.2 Thuật toán Meanshift

2.2.1 Hàm mật độ xác suất

Hàm mật độ xác suất (*Probability Density Function*) của biến ngẫu nhiên liên tục X cho biết mức độ tập trung xác suất tại mỗi giá trị x . Cụ thể, giá trị $f(x)$ của hàm mật độ xác suất tại mỗi điểm x thể hiện xác suất một biến ngẫu nhiên liên tục rơi vào khoảng xung quanh x thoả mãn các tính chất:

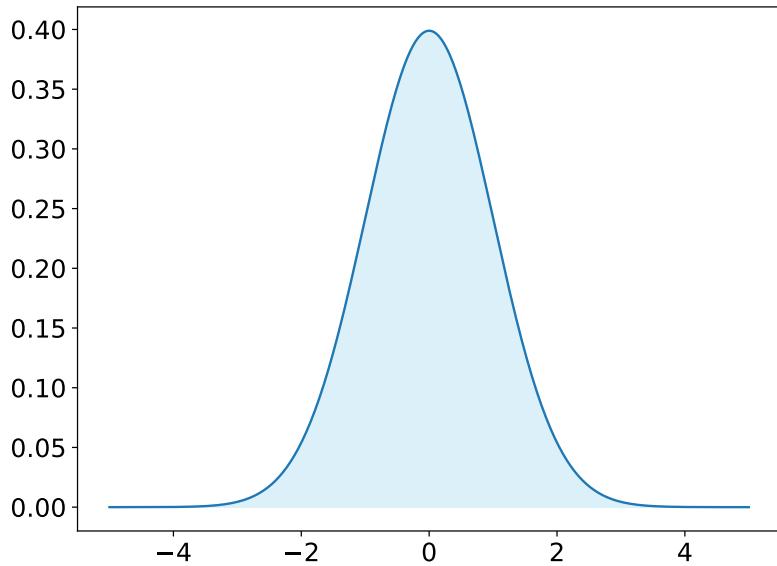
$$f(x) \geq 0, \quad \text{và} \quad \int_{-\infty}^{\infty} f(x) dx = 1$$

Với $f(x)$ là hàm mật độ xác suất. Giá trị của hàm mật độ xác suất không âm và tổng diện tích dưới đường cong của hàm mật độ xác suất bằng 1, vì xác suất tổng của tất cả các sự kiện có thể xảy ra là 1.

Thuật toán Mean Shift là thuật toán sử dụng hàm mật độ xác suất để tìm các vùng có mật độ cao, trong đồ thị PDF thì vùng có mật độ cao là đỉnh của đồ thị.

2.2.2 Kernel Density Estimation

Tuy nhiên, trong thực tế, không phải dữ liệu nào cũng tuân theo phân phối Guassian. Khi đó, ta cần sử dụng *KDE* (*kernel density estimation*) để ước lượng hàm PDF. Cho X là biến ngẫu nhiên có N quan sát, KDE được tính như sau:



Hình 1: Hàm mật độ phân phối xác suất của phân phối Gaussian

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^N K(x - x_i),$$

với K là hàm kernel.

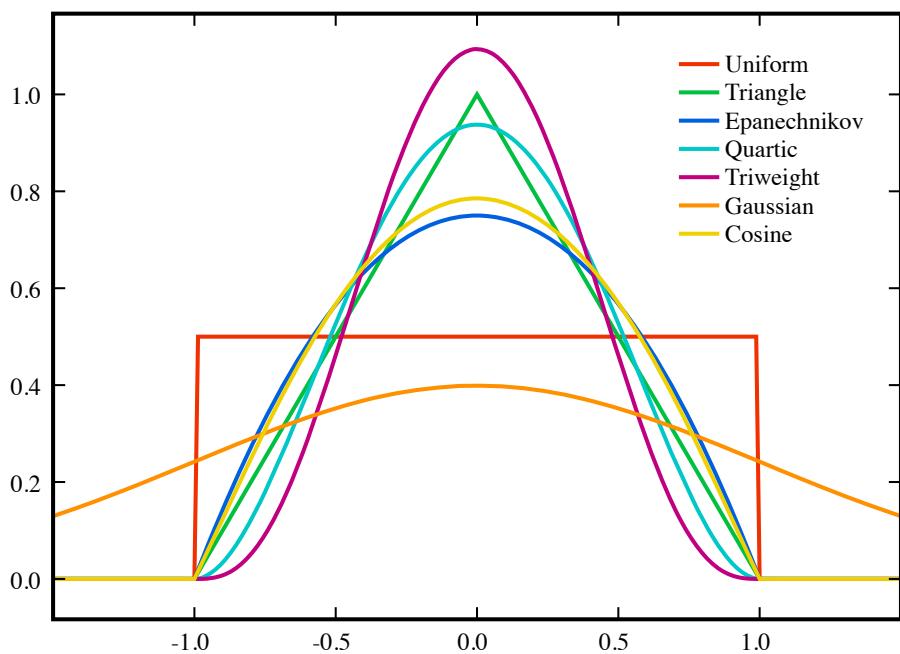
KDE được tổng hợp từ nhiều kernel. Trong KDE, kernel là hàm giúp ước lượng PDF cho từng điểm dữ liệu. Mỗi kernel trượt qua từng điểm dữ liệu trong dữ liệu mẫu rồi tạo ra một hình dạng hàm phản ánh mức độ ảnh hưởng của điểm dữ liệu tại vị trí đó lên hàm mật độ xác suất tổng thể. Có nhiều hàm kernel tuy nhiên hàm Gaussian kernel được sử dụng nhiều nhất trong thực tế, và là lựa chọn mặc định trong thư viện sklearn. Định nghĩa của hàm Gaussian kernel như sau:

$$k_{Gaussian}(x_i, x_j) = \exp\left(\frac{-1}{2\sigma^2} \|x_i - x_j\|^2\right)$$

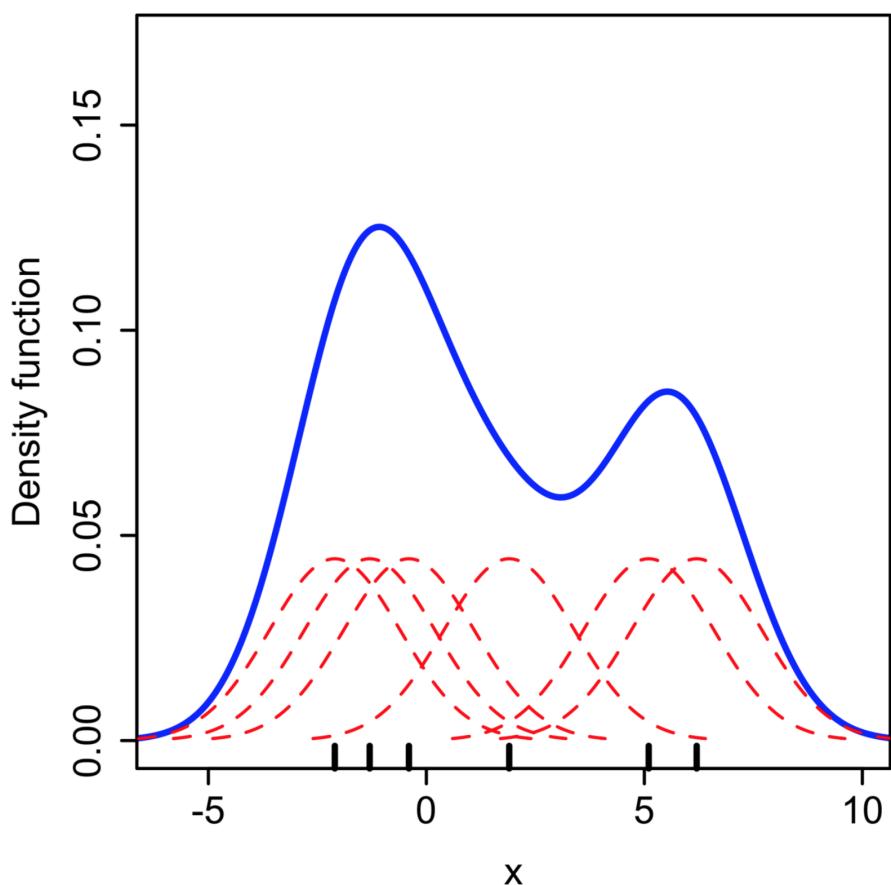
Trong đó, $\|x_i - x_j\|$ là khoảng cách giữa hai điểm dữ liệu x_i và x_j , còn σ là độ rộng của kernel hay còn gọi là bandwidth.

Ở những nơi có nhiều điểm dữ liệu tập trung thì số lượng các đường cong chồng lấn lên nhau sẽ nhiều hơn và do đó khi tính tổng cộng dồn của nó ta sẽ thu được một giá trị tích luỹ (cummulative) kernel density lớn hơn.

Bandwidth của kernel là một tham số quan trọng quyết định độ "mịn" hay "gồ ghè" của KDE vì bandwidth quyết định độ rộng của kernel. Khi bandwidth lớn, KDE trở nên mịn



Hình 2: Các loại Kernel khác. Nguồn: [Wikipedia](#)



Hình 3: Minh họa KDE. Nguồn: [UC Davis Statistics](#)

(*smooth*) hơn, nhưng đồng thời có thể làm mất mát thông tin về các đỉnh nhỏ. Khi bandwidth nhỏ, KDE trở nên gồ ghề, làm suy giảm khả năng nhận diện và phân loại

các cụm dữ liệu có cấu trúc rõ ràng. Vì vậy, việc điều chỉnh tham số bandwidth rất quan trọng.

Khi đã có KDE, việc cần làm tiếp theo là tìm các vùng có mật độ cao từ phương pháp KDE, bài toán trở thành di tìm cực trị của hàm KDE. Tuy nhiên, trong hầu hết các trường hợp, việc giải phương trình đạo hàm bằng 0 là bất khả thi. Nguyên nhân có thể đến từ sự phức tạp của dạng của đạo hàm, từ việc các điểm dữ liệu có số chiều lớn, hoặc từ việc có quá nhiều điểm dữ liệu. Hướng tiếp cận phổ biến nhất là xuất phát từ một điểm rồi dùng một phép toán lặp để tiến dần đến điểm cần tìm, tức đến khi đạo hàm gần với 0. Trong trường hợp này, ta sử dụng ước lượng Gradient. Gradient cùng hướng với Mean Shift vector [[10]] (Mean Shift vector mô tả hướng và độ lớn giữa 2 trọng tâm, điều này được đề cập ở phần tiếp theo), nghĩa là khi di chuyển từ một trọng tâm đến vị trí mới (theo Mean Shift vector), đạo hàm của KDE sẽ tiến gần tới 0, cũng là vị trí cực đại của KDE. Từ đó, ta xác định được vùng có mật độ cao.

2.2.3 Cơ sở lý thuyết của thuật toán Meanshift

Thuật toán Phân cụm Mean Shift là một thuật toán phân cụm dựa trên mật độ, không tham số, có thể được sử dụng để xác định các cụm trong tập dữ liệu. Các cụm có hình dạng tùy ý và không bị phân tách bởi các ranh giới tuyến tính (không yêu cầu trước biết về số lượng cụm và không hạn chế hình dạng của cụm).

Thuật toán Meanshift gán các điểm dữ liệu vào các cụm một cách lặp lại bằng cách di chuyển các điểm về phía mode (mode là mật độ cao nhất của các điểm dữ liệu trong khu vực, trong ngữ cảnh của Meanshift). Do đó, nó cũng được biết đến là thuật toán tìm kiếm mode (Mode-seeking algorithm). Thuật toán Mean shift thường được ứng dụng trong lĩnh vực xử lý ảnh và thị giác máy tính [[9, 26, 8]]

Ý tưởng cơ bản của phân cụm Mean-shift là di chuyển mỗi điểm dữ liệu về phía mode (tức là mật độ cao nhất) của phân phối các điểm trong một bán kính nhất định. Thuật toán lặp lại các di chuyển này cho đến khi các điểm hội tụ tới một cực đại cục bộ của hàm mật độ. Những cực đại cục bộ này đại diện cho các cụm trong dữ liệu. Cụ thể, thuật toán phân cụm Mean-Shift gồm các bước sau:

- **Bước 1:** Khởi tạo kernel cho từng điểm dữ liệu: Bắt đầu từ mỗi điểm dữ liệu, ta tạo kernel cho điểm đó. Nếu biểu diễn các điểm dữ liệu trong không gian 2 chiều, bandwidth của kernel trở thành bán kính của đường tròn với trọng tâm là điểm dữ liệu đó. Đường tròn này gọi là window giúp kiểm soát phạm vi tìm kiếm.
- **Bước 2:** Tính toán centroid: Tính trọng tâm mới từ các điểm nằm trong window. Nếu tạo ra một trọng tâm gần giống với trọng tâm vừa có được hoặc đạt đến một

điều kiện dừng thì dừng thuật toán.

- **Bước 3:** Di chuyển window đến vị trí mới, với trọng tâm mới được tính ở bước 2. Quá trình trọng tâm di chuyển đến trọng tâm mới tạo ra vector Mean Shift, vector này mô tả hướng và độ lớn giữa 2 trọng tâm, giúp window dần tiến đến các vùng dày đặc. Sau đó lặp lại bước 2.
- **Bước 4:** Trả về các trung tâm cụm cuối cùng và phân hoạch dữ liệu vào các cụm.

Ưu điểm của thuật toán Meanshift bao gồm thuật toán này không yêu cầu xác định trước số lượng cụm dữ liệu như k-Means.

Đặc biệt, Mean Shift không nhạy cảm với các điểm ngoại vi và đảm bảo tính hội tụ trong quá trình xử lý. Điều này giúp thuật toán hoạt động hiệu quả hơn khi xử lý dữ liệu có nhiễu.

Một ưu điểm khác của Mean Shift là thuật toán này không đặt ra bất kỳ giả định nào về hình dạng của các cụm dữ liệu, không giới hạn chúng trong các hình dạng cụ thể như hình cầu, hình elip, điều này tạo ra tính linh hoạt cao trong việc xác định cụm dữ liệu.

Tuy nhiên, thuật toán Mean Shift cũng có một số nhược điểm đáng lưu ý. Trước hết là việc lựa chọn bandwidth của kernel là một phần quan trọng và khó khăn. Việc bandwidth không được chọn đúng có thể dẫn đến hiệu suất kém của phân cụm và có thể dẫn đến việc các phân cụm không hiệu quả, tức là độ tương đồng giữa các cụm cao hoặc độ tương đồng giữa các điểm trong cụm thấp.

Ngoài ra, Mean Shift không hiệu quả khi áp dụng vào không gian đặc trưng có số chiều lớn. Khi số chiều của dữ liệu tăng lên, việc tính toán và xử lý trở nên phức tạp hơn, có thể làm giảm hiệu suất của thuật toán.

Những hạn chế này làm giảm tính ứng dụng của Mean Shift trong những bộ dữ liệu có số chiều cao và đòi hỏi sự điều chỉnh cẩn thận về thông số của thuật toán.

3 CHƯƠNG 3: PHÂN TÍCH VÀ TRỰC QUAN HOÁ DỮ LIỆU

3.1 Tổng quan bộ dữ liệu

Shop Customer Data là một bộ dữ liệu chứa thông tin cơ bản của khách hàng tại một cửa hàng ảo. Cửa hàng thu thập dữ liệu của khách hàng thông qua thẻ thành viên. Việc phân chia khách hàng dựa trên dữ liệu đó có thể giúp cửa hàng hiểu rõ hơn về đối tượng khách hàng của mình và đặt ra những chiến lược phù hợp đối với từng phân khúc khách hàng.

Đọc file dữ liệu dưới dạng csv:

```
1 df = pd.read_csv('/content/Customers.csv')
```

Thông tin về bộ dữ liệu:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CustomerID      2000 non-null    int64  
 1   Gender          2000 non-null    object  
 2   Age              2000 non-null    int64  
 3   Annual Income ($) 2000 non-null    int64  
 4   Spending Score (1-100) 2000 non-null    int64  
 5   Profession      1965 non-null    object  
 6   Work Experience 2000 non-null    int64  
 7   Family Size     2000 non-null    int64  
dtypes: int64(6), object(2)
memory usage: 125.1+ KB
```

Bộ dữ liệu này bao gồm 8 thuộc tính và 2000 quan sát. Các thuộc tính được mô tả cụ thể trong bảng sau:

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	CustomerID	Mã định danh cho mỗi khách hàng để phân biệt (2000 khách hàng)	int64
2	Gender	Giới tính của khách hàng: nam hoặc nữ	object
3	Age	Tuổi của khách hàng khi thu thập dữ liệu	int64
4	Annual Income	Thu nhập hàng năm của khách hàng (đơn vị: đô la)	int64
5	Spending Score	Điểm số dựa trên hành vi mua sắm của khách hàng	int64
6	Profession	Nghề nghiệp của khách hàng	object
7	Work Experience	Số năm làm việc của khách hàng	int64
8	Family Size	Số lượng thành viên trong gia đình của khách hàng	int64

Bảng 1: Bảng mô tả thuộc tính khách hàng

Quan sát giá trị của các numerical và categorical columns:

```
1 num = df.select_dtypes(exclude='O')
2 cat = df.select_dtypes(include='O')
3 print("Numerical Columns:")
4 stt = 1
5 for col in num.columns:
6     print(f'{stt})', '{:<25}'.format(col), f'({len(df[col].unique())} '
7         'values):', f'{min(df[col]):} - {max(df[col]):}')
8     stt += 1
9 print('')
10 print("Categorical Columns:")
11 stt = 1
12 for col in cat.columns:
13     print(f'{stt})', '{:<10}'.format(col), f'({len(df[col].unique())} '
14         'values):', f'{df[col].unique()}')
15     stt += 1
```

Numerical Columns:

```
1) CustomerID      (2000 values): [1 - 2000]
2) Age             (100 values): [0 - 99]
3) Annual Income ($) (1786 values): [0 - 189974]
4) Spending Score (1-100) (101 values): [0 - 100]
5) Work Experience (18 values): [0 - 17]
6) Family Size     (9 values): [1 - 9]
```

Categorical Columns:

```
1) Gender          (2 values): ['Male' 'Female']
2) Profession (10 values): ['Healthcare' 'Engineer' 'Lawyer' 'Entertainment' 'Artist' 'Executive'
   'Doctor' 'Homemaker' 'Marketing' nan]
```

Quan sát dữ liệu:

```
1 print(df.head())
2 print('-'*80)
3 print(df.tail())
```

3.2 Tiềng xử lý dữ liệu

3.2.1 Điều chỉnh các cột

Xóa cột CustomerID vì nó không có giá trị phân tích. Sau đó đổi tên cột Annual Income (\$) và Spending Score (1-100) lần lượt thành Annual Income, Spending Score để tên các cột ngắn gọn và đồng nhất về hình thức.

	CustomerID	Gender	Age	Annual Income (\$)	Spending Score (1-100)	\
0	1	Male	19	15000	39	
1	2	Male	21	35000	81	
2	3	Female	20	86000	6	
3	4	Female	23	59000	77	
4	5	Female	31	38000	40	
	Profession	Work Experience	Family Size			
0	Healthcare		1	4		
1	Engineer		3	3		
2	Engineer		1	1		
3	Lawyer		0	2		
4	Entertainment		2	6		
	CustomerID	Gender	Age	Annual Income (\$)	Spending Score (1-100)	\
1995	1996	Female	71	184387	40	
1996	1997	Female	91	73158	32	
1997	1998	Male	87	90961	14	
1998	1999	Male	77	182109	4	
1999	2000	Male	90	110610	52	
	Profession	Work Experience	Family Size			
1995	Artist		8	7		
1996	Doctor		7	7		
1997	Healthcare		9	2		
1998	Executive		7	2		
1999	Entertainment		5	2		

```

1 #Xóa cột CustomerID
2 df = df.drop(['CustomerID'], axis = 1)
3 #Thay đổi tên cột
4 df.rename(columns={
5     'Annual Income ($)' : 'Annual Income',
6     'Spending Score (1-100)' : 'Spending Score'
7 }, inplace=True)

```

3.2.2 Xử lý dữ liệu bị thiếu

Sau khi kiểm tra số lượng các dòng bị thiếu dữ liệu của từng cột. Nhận thấy chỉ có 35 dòng thuộc cột Profession bị thiếu dữ liệu. Vì số lượng các dòng này khá ít, chiếm tỉ lệ rất nhỏ trong tổng thể bộ dữ liệu nên có thể loại bỏ những dòng này mà vẫn đảm bảo sẽ không mất quá nhiều thông tin.

```

1 df.isna().sum()

```

```

Gender          0
Age            0
Annual Income  0
Spending Score 0
Profession      35
Work Experience 0
Family Size     0
dtype: int64

```

```

1 #Xóa dòng có missing data
2 df.dropna(inplace=True)
3 #Xem lại dữ liệu sau khi xóa các dòng có missing data
4 df.info()
5 #Xóa dòng có missing data
6 df.dropna(inplace=True)
7 #Xem lại dữ liệu sau khi xóa các dòng có missing data
8 df.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1965 entries, 0 to 1999
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
--- 
 0   Gender        1965 non-null   object 
 1   Age           1965 non-null   int64  
 2   Annual Income 1965 non-null   int64  
 3   Spending Score 1965 non-null   int64  
 4   Profession    1965 non-null   object 
 5   Work Experience 1965 non-null   int64  
 6   Family Size   1965 non-null   int64  
dtypes: int64(5), object(2)
memory usage: 122.8+ KB

```

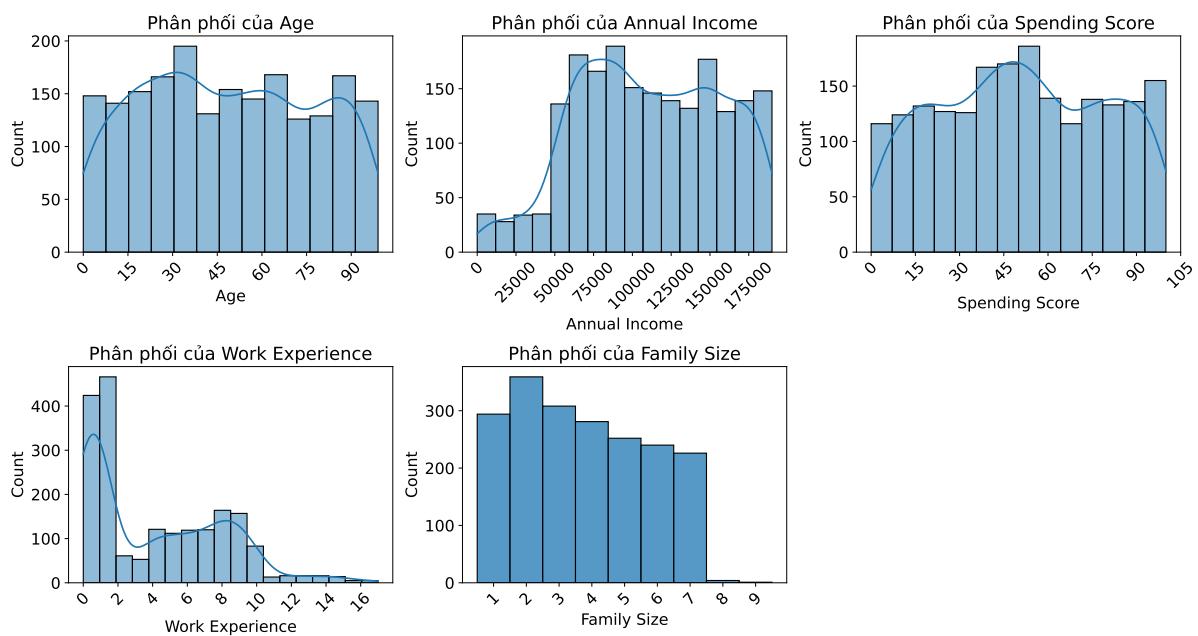
3.2.3 Xử lý ngoại lai (Outliers)

Tiến hành khảo sát outliers của biến định lượng, định tính. Đầu tiên, dùng phương pháp trực quan boxplot để xác định outliers:



Hình 4: Box plot các biến định lượng trong bộ dữ liệu

Để chọn phương pháp nhận diện outliers phù hợp với dữ liệu, cần xem xét hình dạng phân phối của từng thuộc tính:



Hình 5: Phân phối của từng thuộc tính trong bộ dữ liệu

Vì cả 5 thuộc tính đều không có phân phối chuẩn, một vài thuộc tính phân phối không đối xứng IQR được cho là phù hợp nhất để nhận diện (và xem giá trị) outliers cụ thể trong trường hợp này.

```

1 #IQR
2 def detect_outliers_iqr(data):
3     Q1 = data.quantile(0.25)
4     Q3 = data.quantile(0.75)
5     IQR = Q3 - Q1
6     outliers = data[(data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 * IQR))]
7     return outliers
8
9
10 for column in num.columns:
11     outliers = detect_outliers_iqr(num[column])
12     print(f"Outliers cột {column}: {np.sort(outliers.unique())} - Số lượng:
    ↪ {len(outliers)} ({len(outliers) / len(num[column]) * 100:.2f}%)")

```

Outliers cột Age: [] – Số lượng: 0 (0.00%)
 Outliers cột Annual Income: [] – Số lượng: 0 (0.00%)
 Outliers cột Spending Score: [] – Số lượng: 0 (0.00%)
 Outliers cột Work Experience: [17] – Số lượng: 5 (0.25%)
 Outliers cột Family Size: [] – Số lượng: 0 (0.00%)

Từ kết quả trên, giá trị 17 năm của thuộc tính Work Experience được xác định là ngoại lai. Cần khảo sát kỹ hơn các dòng dữ liệu này để quyết định phương án xử lý.

```

1 #Xem thông tin các dòng có chứa outliers
2 df[df['Work Experience']==17]

```

	Gender	Age	Annual Income	Spending Score	Profession	Work Experience	Family Size
392	Male	21	119116	30	Artist	17	4
405	Female	65	119889	11	Artist	17	6
473	Male	20	130813	92	Artist	17	5
566	Female	19	180331	14	Artist	17	5
603	Female	91	69720	78	Lawyer	17	6

Từ kết quả trên, có thể nhận thấy sự bất thường đối với giá trị cột Work Experience so với giá trị cột Age tương ứng. Cụ thể ở dòng 392, 473 và dòng 566, các đối tượng chỉ mới 19, 20 tuổi nhưng lại có 17 năm kinh nghiệm đi làm. Từ đây có thể suy ra bộ dữ liệu có thể chứa nhiều dòng có giá trị bất thường. Do đó tiến hành kiểm tra những đối tượng

bắt đầu đi làm khi dưới 16 tuổi, tức là hiệu số giữa giá trị cột Age và giá trị cột Working Experience bé hơn 16.

¹ #Xem những người bắt đầu đi làm khi dưới 16 tuổi

² df[df['Age']-df['Work Experience']<16]

	Gender	Age	Annual Income	Spending Score	Profession	Work Experience	Family Size
33	Male	18	62000	92	Homemaker	9	7
39	Female	20	69000	75	Artist	8	2
47	Female	27	71000	47	Healthcare	12	1
61	Male	19	50000	55	Artist	9	2
95	Male	24	80000	52	Artist	10	1
...
1979	Male	0	165321	93	Doctor	8	1
1980	Female	10	86925	76	Artist	7	2
1984	Female	2	153622	51	Lawyer	6	6
1986	Female	4	68094	61	Doctor	4	7
1994	Female	19	54121	89	Engineer	6	3

Vì những dòng này có ý nghĩa mâu thuẫn, có thể xem là dữ liệu không hợp lệ. Do không có đủ cơ sở để xác định nguyên nhân cũng như hướng điều chỉnh giá trị cho những dòng này, nhóm quyết định loại bỏ chúng để đảm bảo cho chất lượng của mô hình máy học.

¹ #Loại ra những người bắt đầu đi làm khi dưới 16 tuổi

² df.drop(df[df['Age']-df['Work Experience']<16].index, axis = 0, inplace=True)

3.3 Trực quan hóa dữ liệu

3.3.1 Thống kê mô tả

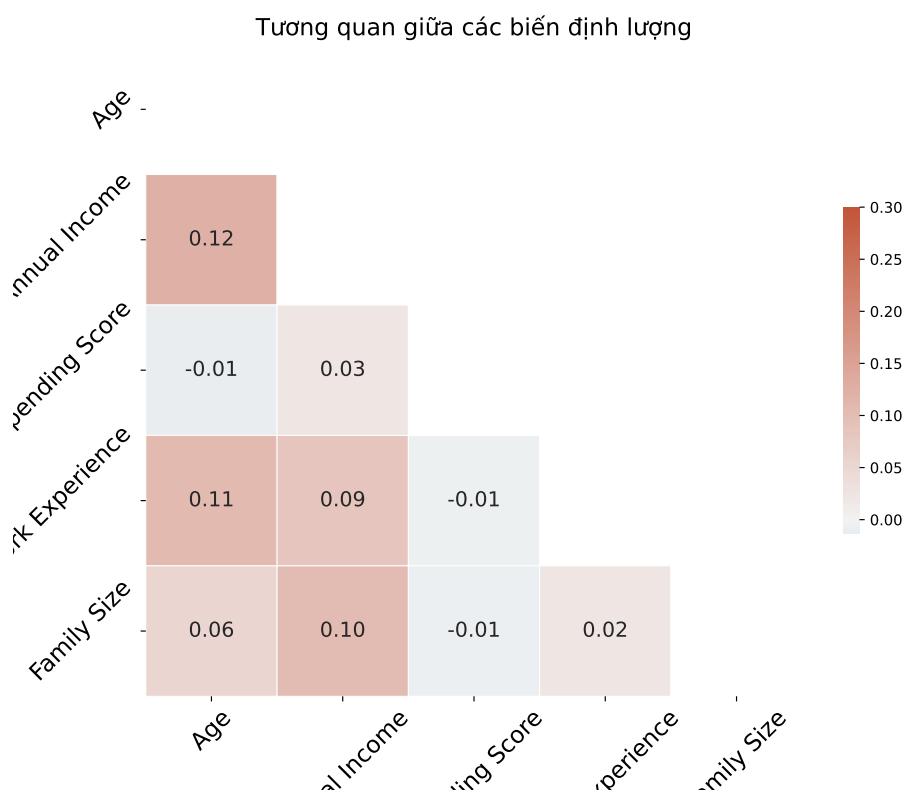
Thống kê mô tả bộ dữ liệu vừa được tiền xử lý cho thấy xu hướng trung tâm, sự phân tán và phạm vi của các biến định lượng: Age, Annual Income, Spending Score, Work Experience, Family Size.

¹ df.describe()

	Age	Annual Income	Spending Score	Work Experience	Family Size
count	1592.000000	1592.000000	1592.000000	1592.000000	1592.000000
mean	58.002513	108516.331030	50.558417	3.800879	3.752513
std	23.433713	46193.912167	27.707943	3.778756	1.956229
min	16.000000	0.000000	0.000000	0.000000	1.000000
25%	37.000000	72399.250000	28.000000	1.000000	2.000000
50%	58.000000	105278.500000	50.000000	2.000000	4.000000
75%	79.000000	147283.750000	74.000000	7.000000	5.000000
max	99.000000	189974.000000	100.000000	17.000000	9.000000

3.3.2 Tương quan giữa các biến định lượng

Để có cái nhìn tổng quan về mối quan hệ giữa các biến định lượng của bộ dữ liệu, nhóm tiến hành chọn ra các biến, tính hệ số tương quan và sử dụng heatmap để trực quan hóa:



Hình 6: Biểu đồ tương quan giữa các biến định lượng

Từ biểu đồ heatmap trên, có thể rút ra một vài nhận xét chung và tổng quát về mối tương quan giữa các biến định lượng như sau:

Mối tương quan dương:

- Kinh nghiệm làm việc (Work Experience) và Thu nhập hàng năm (Annual Income) có mối tương quan dương là 0.09, cho thấy rằng khi số năm kinh nghiệm của khách hàng tăng lên, thu nhập hàng năm của họ cũng có xu hướng tăng lên.
- Kích thước gia đình (Family Size) và Thu nhập hàng năm (Annual Income) cũng có mối tương quan dương là 0.1, cho thấy những gia đình nhiều người có xu hướng có thu nhập hàng năm cao hơn.
- Độ tuổi (Age) có mối tương quan dương với Kích thước gia đình (Family Size), Kinh nghiệm làm việc (Work Experience) và Thu nhập hàng năm (Annual Income). Có thể nhận xét rằng khi càng lớn tuổi thì kích thước gia đình càng lớn, kinh nghiệm làm việc và thu nhập hàng năm cũng tăng cao hơn.

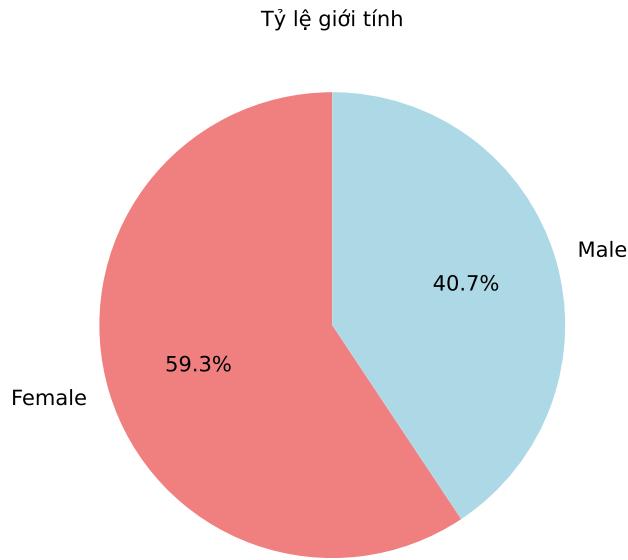
Mối tương quan âm:

- Tuổi tác (Age) và Điểm chi tiêu (Spending Score) có mối tương quan âm -0.01, ý rằng những khách hàng lớn tuổi thường có mức điểm chi tiêu thấp hơn so với những người trẻ.
- Tương tự với Kích thước gia đình (Family Size) và Mức điểm chi tiêu (Spending Score), tương quan của 2 biến này chỉ nằm ở mức -0.01. Điều này khá bất thường vì thường khi gia đình có nhiều thành viên, mức chi tiêu sẽ cao hơn.
- Số năm kinh nghiệm (Work Experience) và Điểm chi tiêu (Spending Score) cũng có mối tương quan âm là -0.01. Điều này cho thấy rằng những khách hàng có nhiều kinh nghiệm làm việc thường có mức điểm chi tiêu thấp hơn.

Tuy nhiên, nhìn chung mức tương quan giữa các biến còn thấp và không rõ ràng vì chỉ dao động trong khoảng -0.01 đến 0.12. Tiếp theo, nhóm sẽ đi sâu vào phân tích chi tiết thuộc tính Giới tính trong bộ dữ liệu.

Cụ thể, tìm hiểu về tỷ lệ giới tính khách hàng là một phần quan trọng trong việc hiểu đặc điểm khách hàng và tạo ra chiến lược kinh doanh hiệu quả. Vì nó không chỉ cung cấp cái nhìn đầu tiên về sự đa dạng của đối tượng mục tiêu mà còn giúp định hình chiến lược tiếp thị, dịch vụ, và sản phẩm.

Tìm hiểu về tỷ lệ giới tính của khách hàng có thể thấy rõ rằng khách hàng nữ đang chiếm ưu thế hơn so với khách hàng nam với tỷ lệ 59.6%. Về mặt kinh doanh, có thể coi khách hàng nữ là phân khúc khách hàng chính cần được tập trung đáp ứng nhu cầu và sở thích. Tuy nhiên, với tỷ lệ 40.4% thì khách hàng nam cũng là một phần quan trọng của thị



Hình 7: Biểu đồ tròn thể hiện tỉ lệ giới tính

trường. Từ đó, có thể cần phát triển thêm các sản phẩm hoặc dịch vụ phù hợp với nhu cầu của khách hàng nam để thu hút thêm khách hàng từ phân khúc này.

Phân bổ của nghề nghiệp của khách hàng

Ngoài tỷ lệ giới tính, để hiểu rõ về khách hàng thì việc phân tích số lượng nghề nghiệp của khách hàng cũng là một trong những yếu tố quan trọng. Tiến hành đếm số lượng khách hàng theo từng nghề nghiệp và sử dụng tabulate để in ra kết quả:

```

1 # Nghề nghiệp nào có số lượng lớn nhất?
2 from tabulate import tabulate
3 # Tính tỷ lệ của mỗi nghề nghiệp
4 profession_counts = df['Profession'].value_counts()
5
6
7 # Tạo bảng tabulate
8 table = tabulate(profession_counts.reset_index(), headers=['Nghề nghiệp', 'Số
   ↳ lượng'], tablefmt='fancy_grid', showindex=False)
9 print(table)

```

Trực quan hóa bằng biểu đồ thanh ngang:

Nghề nghiệp	Số lượng
Artist	494
Healthcare	273
Entertainment	193
Engineer	156
Doctor	126
Executive	123
Lawyer	115
Marketing	67
Homemaker	45

```

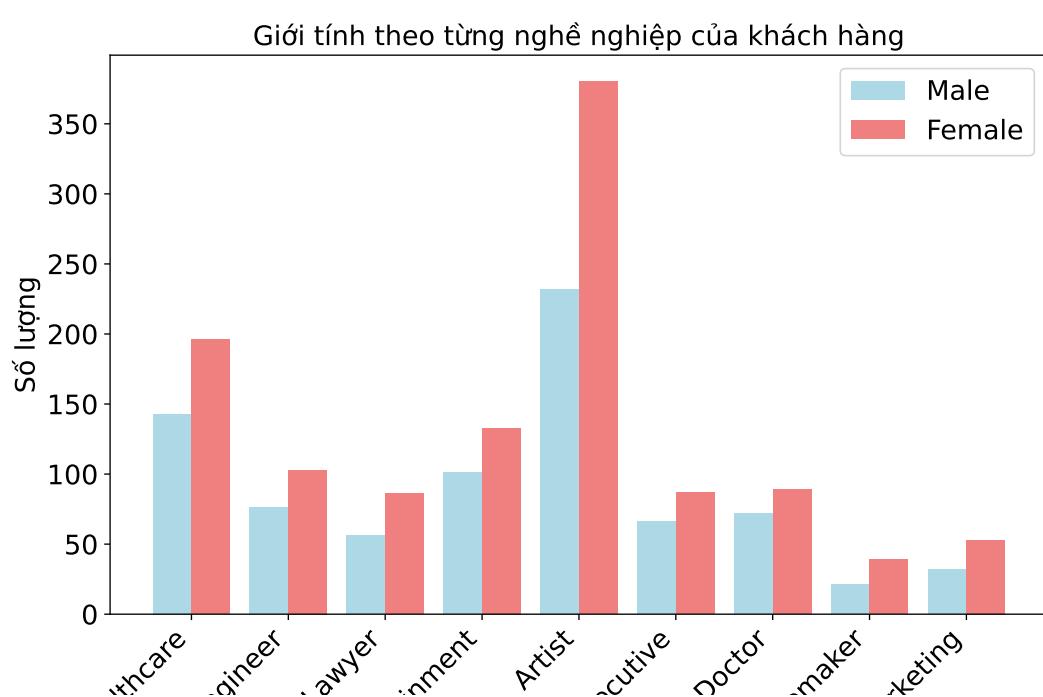
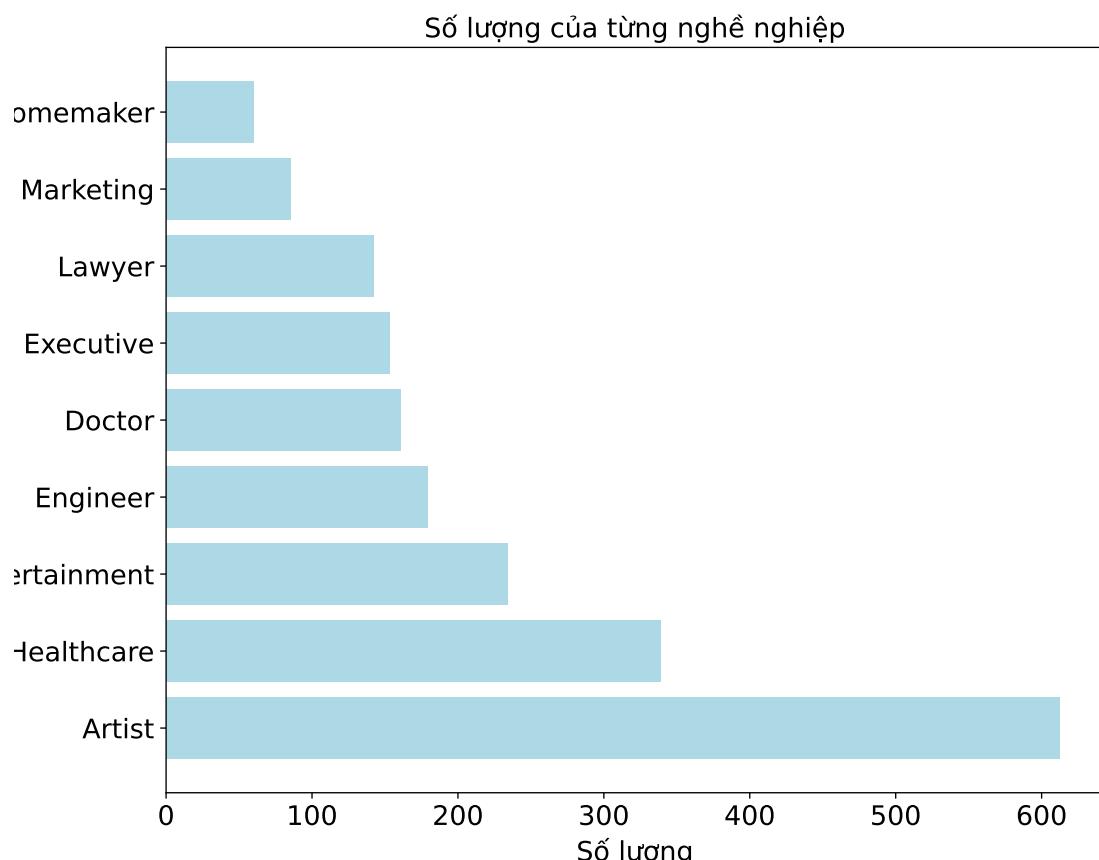
1 # Tao biểu đồ
2 plt.figure(figsize=(10, 8))
3 plt.barh(profession_counts.index, profession_counts, color='lightblue')
4 plt.title('Số lượng của từng nghề nghiệp', fontsize=16)
5 plt.xlabel('Số lượng', fontsize=16)
6 plt.ylabel('Nghề nghiệp', fontsize=16)
7 plt.xticks(fontsize=16)
8 plt.yticks(fontsize=16)
9 plt.savefig('figs/Số lượng của từng nghề nghiệp.pdf')
10 plt.show()

```

Từ bảng và biểu đồ trên có thể thấy, khách hàng là Artist (nghệ sĩ) chiếm số lượng nhiều nhất (494), gần gấp đôi so với ngành nghề xếp thứ 2 là Healthcare (chăm sóc sức khỏe) (273). Kết quả này khá bất ngờ vì thường mọi người sẽ mặc định những ngành liên quan tới kỹ thuật sẽ phổ biến hơn. Tuy nhiên có thể lý giải vì đặc thù của nghề là xuất hiện nhiều trước công chúng do đó cần mua sắm thường xuyên để thay đổi diện mạo. Cũng có thể đây là nghề nghiệp có thu nhập cao nên việc chi tiêu nhiều cho mua sắm là hợp lý. Homemaker (Nội trợ) là công việc có số lượng ít nhất. Kết quả này không quá ngạc nhiên vì trong xã hội hiện đại, mọi người thường ra ngoài làm việc và ít ở nhà để làm công việc nội trợ.

Phân bố của giới tính theo nghề nghiệp

Một điều khá thú vị có thể nhìn thấy từ biểu đồ đó là trong tất cả các ngành nghề của khách hàng, số lượng nữ giới luôn nhiều hơn nam giới. Điều này, cho thấy sự quan tâm và mức độ yêu thích của nữ giới dành cho mua sắm và tiêu dùng.

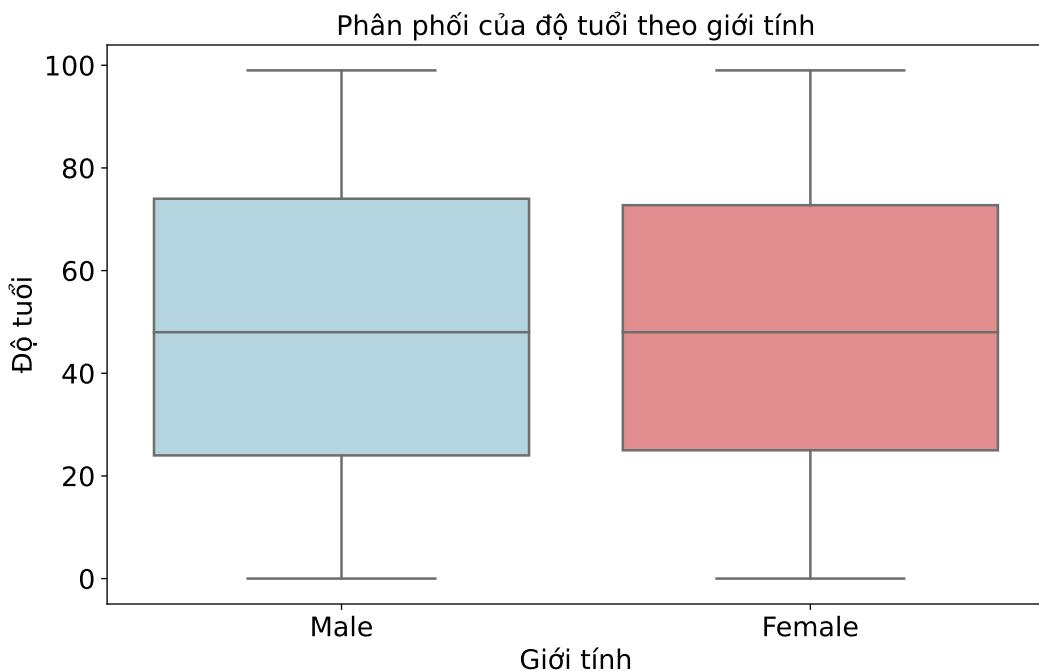


Đặc biệt, với artist (nghệ sĩ) số lượng nữ giới cao áp đảo với hơn 300 người. Từ đây, có thể thấy rằng nên có sự quan tâm và xây dựng chiến lược hợp lý để thu hút thêm khách

hàng là nam giới trong tất cả các ngành nghề.

Tìm hiểu phân phối của độ tuổi khách hàng theo giới tính

Từ boxplot có thể thấy rõ ràng giới tính dường như không có tác động đáng kể đến sự phân bố độ tuổi:



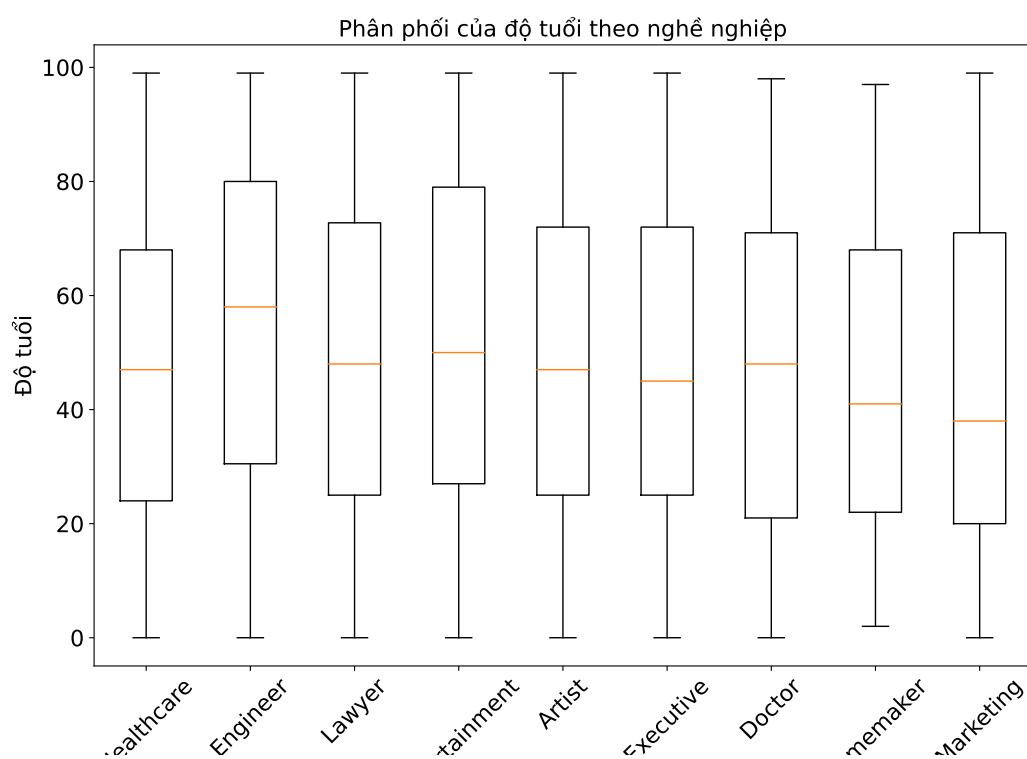
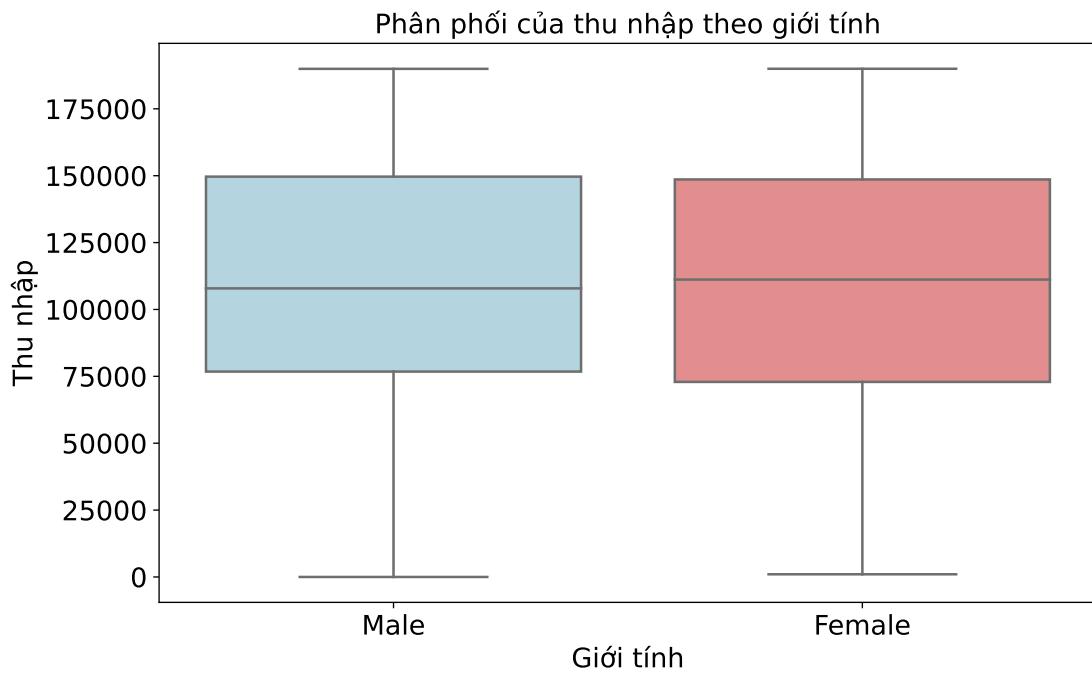
Cả giới tính nam và nữ đều có sự phân bố theo độ tuổi tương tự nhau và không có sự khác biệt đáng kể. Trong đó độ tuổi trung bình của nam và nữ đều trong khoảng 60 tuổi.

Sự phân bổ thu nhập hàng năm theo giới tính cũng tương tự như sự phân bổ theo độ tuổi theo giới tính. Giới tính không có tác động lớn đến thu nhập hàng năm. Tuy nhiên nhìn vào thu nhập trung bình, có thể thấy thu nhập của nữ giới đang nhỉnh hơn đôi chút so với nam giới.

Thay đổi của độ tuổi theo từng ngành nghề

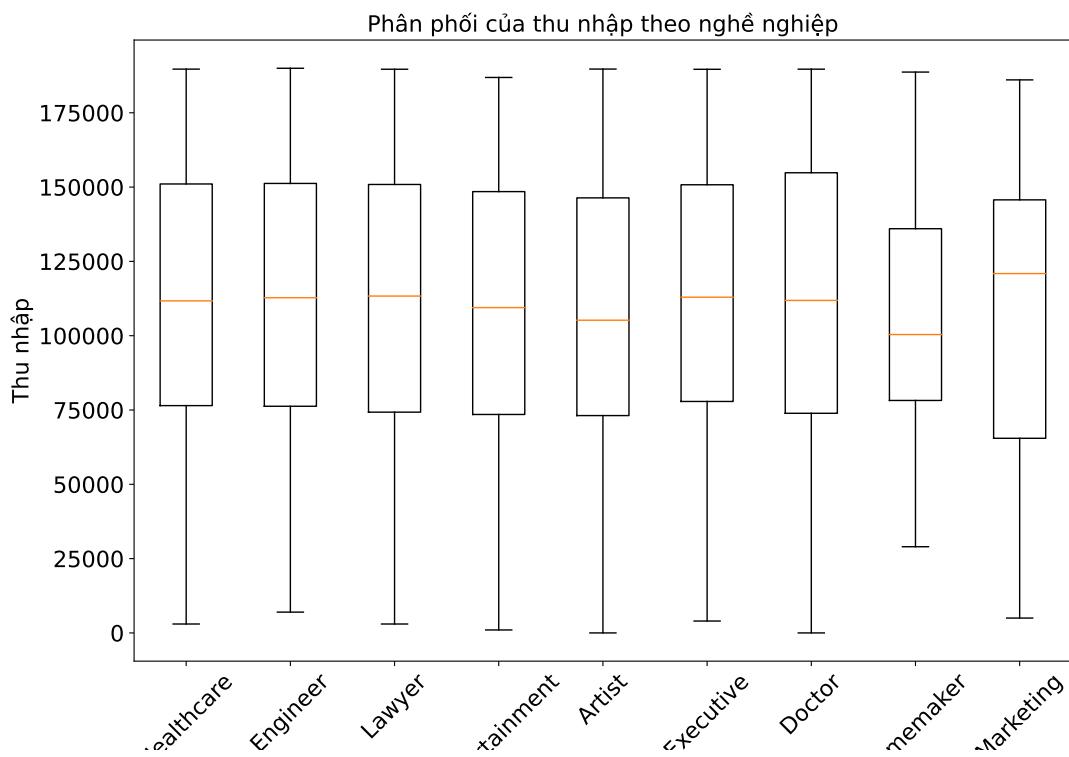
Phân tích boxplot trên, có thể suy ra rằng tuổi tác có liên quan đến nghề nghiệp. Sự phân phối độ tuổi theo các ngành nghề khác nhau không đồng đều. Những quan sát này có thể rất quan trọng trong việc dự đoán nghề nghiệp của một khách hàng dựa trên độ tuổi của họ.

Ngoài ra, điều đáng chú ý là có sự khác nhau ở mức độ phân bố độ tuổi trung bình giữa các ngành nghề. Ví dụ: độ tuổi trung bình của kỹ sư (Engineer) dịch chuyển lên cho thấy phần lớn kỹ sư có độ tuổi trong khoảng 60. Mặt khác, nội trợ (Homemaker) lại có độ tuổi trung bình dịch chuyển xuống dưới, cho thấy phần lớn những người làm nghề này



đều ở độ tuổi dưới 60. Thông tin này có thể có giá trị trong việc phát triển các chiến lược tiếp thị cho sản phẩm hoặc dịch vụ dành riêng cho một nhóm tuổi hoặc nghề nghiệp cụ thể.

Phân tích sâu hơn về mối quan hệ giữa thu nhập hàng năm và nghề nghiệp, có thể thấy sự phân bổ thu nhập của các ngành nghề có liên quan đến yếu tố nghề nghiệp.



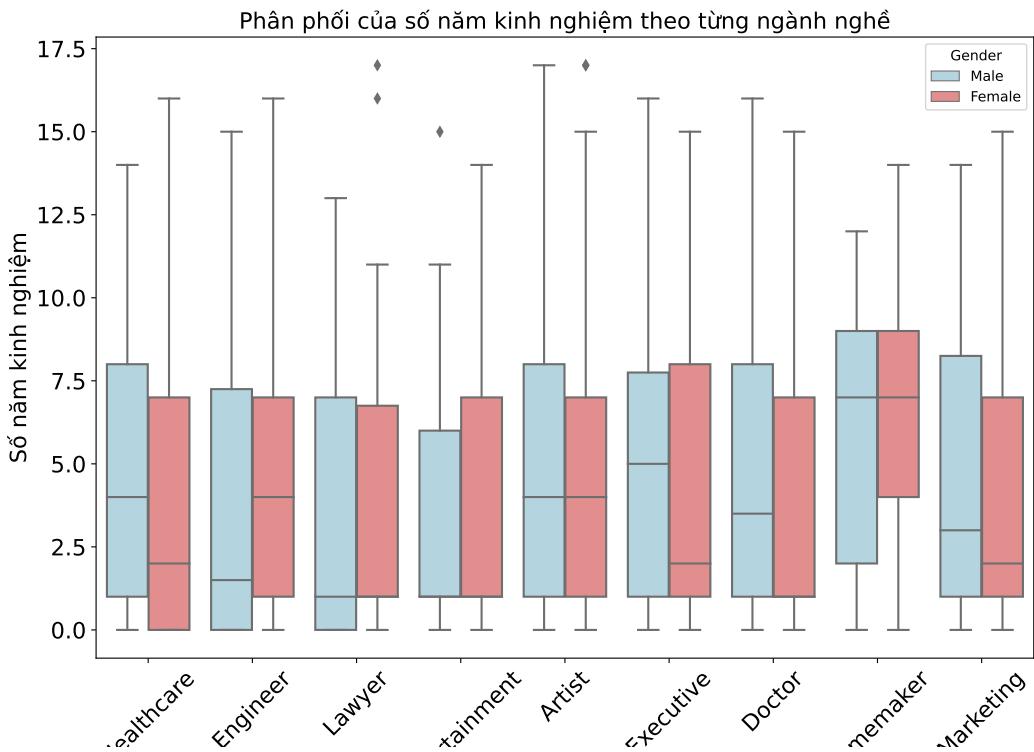
Quản lý (Executive) và kỹ sư (Engineer) là những ngành nghề có thu nhập trung bình cao nhất. Các ngành như chăm sóc sức khỏe (Healthcare), luật sư (Lawyer), nghệ sĩ (Artist) và tiếp thị (Marketing) có cùng mức thu nhập trung bình.

Tuy nhiên, có một số thay đổi đáng chú ý trong phân phối thu nhập của nghề nội trợ. Quan sát từ biểu đồ, các giá trị thấp hơn của boxplot Homemaker có xu hướng dịch chuyển khá cao lên trên.

Số năm kinh nghiệm của các ngành nghề

Qua phân tích biểu đồ về kinh nghiệm làm việc của các ngành nghề khác nhau, có thể nhận thấy một số insights quan trọng như sau:

- Các ngành nghề như chăm sóc sức khỏe (Healthcare), quản lý (Executive), bác sĩ (Doctor) và tiếp thị (Marketing) nhìn chung có kinh nghiệm làm việc cao hơn so với các ngành nghề khác.
- Đáng chú ý là ngành luật sư (Lawyer) và giải trí (Entertainment) có kinh nghiệm làm việc trung bình chỉ khoảng 1 năm, một mức tương đối thấp. Mặc dù sự phân bổ giá trị trung bình của cả hai ngành nghề là hợp lý, bắt đầu từ mức thấp nhất là 1 năm và tăng lên tới khoảng 6 năm, các giá trị trung bình của 2 ngành này vẫn chưa thực sự thỏa đáng.



- Ngược lại, kinh nghiệm làm việc trung bình của quản lý (Executive) và bác sĩ (Doctor) dao động từ 1 năm đến khoảng 8 năm, điều này là hợp lý dựa trên tính chất công việc của những ngành này.
- Hơn nữa, từ biểu đồ có thể quan sát thấy một số trường hợp ngoại lệ, chẳng hạn như khách hàng có kinh nghiệm làm việc 17 năm trong lĩnh vực luật sư (Lawyer) và nghệ sĩ (Artist).
- Nghề nội trợ (Homemaker) là nghề có kinh nghiệm làm việc cao hơn đáng kể so với các ngành nghề khác, bắt đầu từ khoảng 4 năm và có thể lên đến 9 năm. Không quá ngạc nhiên vì thường những người nội trợ sẽ có xu hướng gắn bó với công việc này lâu dài.

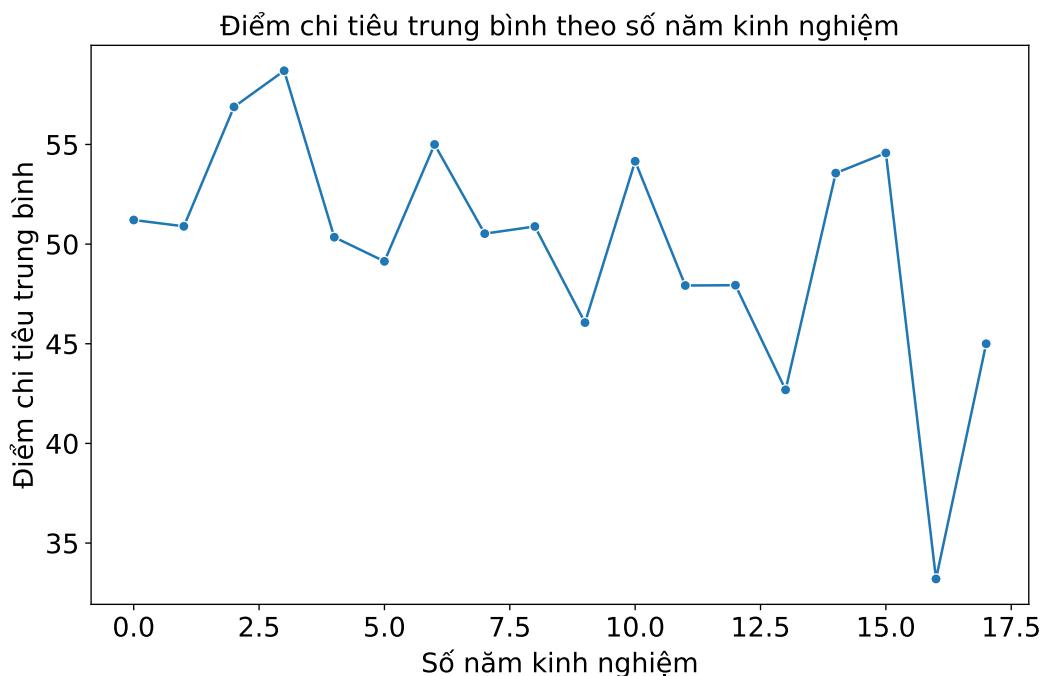
Giá trị trung bình thay đổi đáng kể đối với một số ngành nghề khi so sánh nam và nữ. Ví dụ, trong lĩnh vực chăm sóc sức khỏe (Healthcare), giá trị trung bình thấp đối với nữ và cao đối với nam. Điều này có thể do đặc thù của nghề, khi bác sĩ thường được gắn với nam và y tá gắn với nữ.

Điều thú vị là trong lĩnh vực kỹ sư (Engineer), kinh nghiệm trung bình của nữ cao hơn nhiều so với nam. Nữ giới có kinh nghiệm làm việc trung bình là 4 năm, trong khi nam giới chỉ có khoảng 1 năm. Tương tự, đối với bác sĩ (Doctor), nữ giới có kinh nghiệm làm việc là 1 năm, trong khi nam giới có 4 năm, mặc dù phạm vi tổng thể là gần như nhau.

Cuối cùng, đối với nghề nội trợ, giá trị cao nhất đối với cả nam và nữ là như nhau, nhưng giá trị thấp nhất lại khác nhau. Tuy nhiên, số năm kinh nghiệm trung bình là như nhau cho cả hai giới.

Xu hướng chi tiêu theo số năm kinh nghiệm

Có thể thấy không có 1 xu hướng nhất định của điểm chi tiêu theo số năm kinh nghiệm. Điểm chi tiêu trung bình có sự thay đổi lên xuống theo từng mốc kinh nghiệm khác nhau. Đặc biệt từ giai đoạn 13 năm có 1 sự tăng vọt của điểm chi tiêu, cao nhất là 55 điểm tại năm thứ 15.



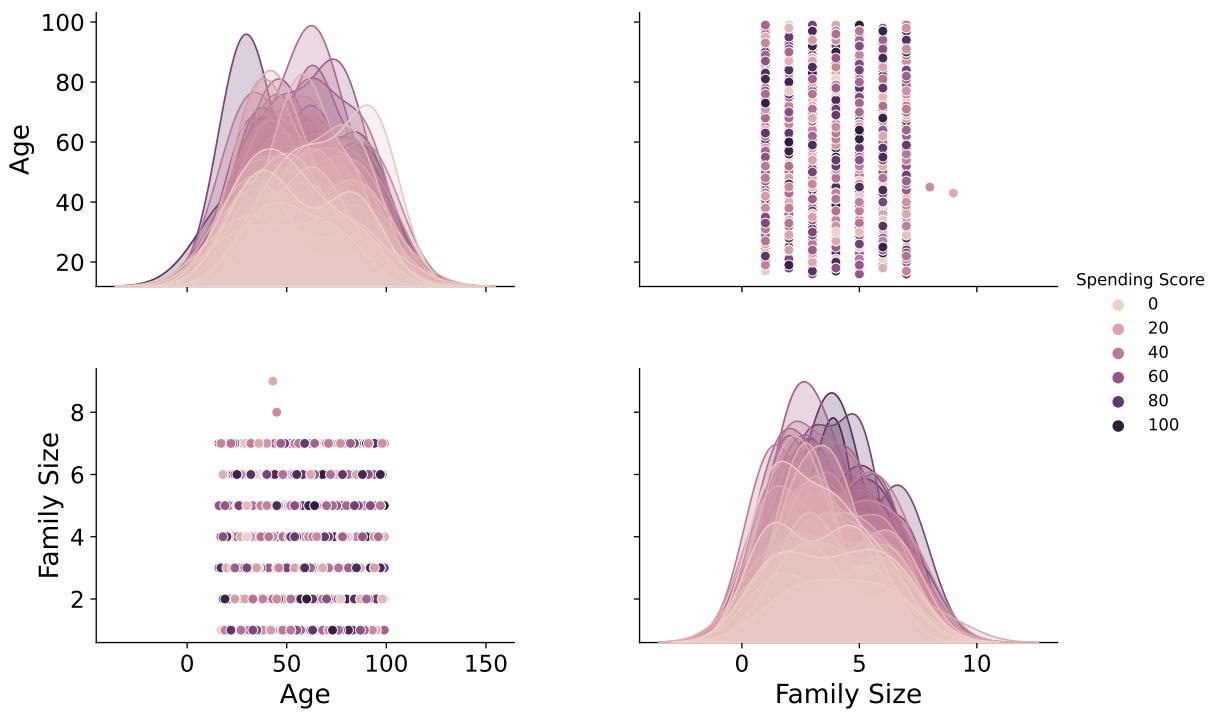
Tuy nhiên sau khi chạm đỉnh, điểm chi tiêu lại giảm đột ngột vào khoảng năm thứ 16 và tăng cao trở lại. Điều này có thể xuất phát từ việc có ít các khách hàng có số năm kinh nghiệm trong khoảng 16 năm. Điểm trung bình tăng vọt vào năm thứ 17 một phần có thể do sự xuất hiện của outliers đã được nhận xét ở biểu đồ trên.

Ảnh hưởng của độ tuổi và kích thước gia đình tới mức điểm chi tiêu

Mối tương quan giữa Age, Family Size và Spending Score cho thấy xu hướng chung là điểm chi tiêu tăng dần theo tuổi tác(thường người lớn thường có thu nhập cao hơn)

Biểu đồ cũng cho thấy rằng Spending Score có xu hướng cao hơn đối với các Family Size càng lớn (các gia đình lớn hơn có nhiều thành viên hơn để chi tiêu cho các nhu cầu, chi phí).

Nhìn chung với Tập dữ liệu như Age, Family Size thể hiện cùng xu hướng cho việc chi

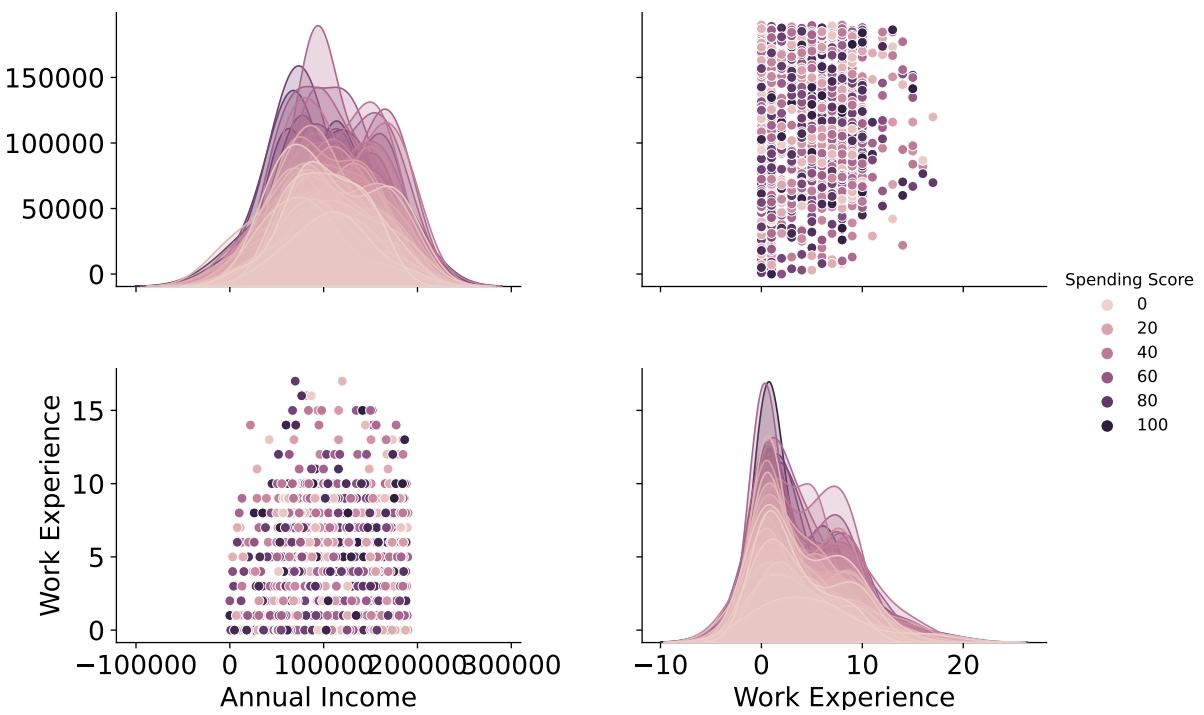


tiêu (Spending Score). Do đó có thể chọn cột Age và bỏ Family Size (do Age có range rộng hơn). Hoặc merge hai Features lại thành một.

Ảnh hưởng của thu nhập hàng năm và kinh nghiệm làm việc tới mức điểm chi tiêu

Mỗi tương quan giữa Annual Income, Work Experience và Spending Score cho thấy xu hướng chung là thu nhập hàng năm tăng dần theo kinh nghiệm làm việc (người lao động có kinh nghiệm hơn thường có kỹ năng và kiến thức tốt hơn thường có mức lương cao hơn).

Nhìn chung với Tập dữ liệu các Features như Annual Income, Work Experience không thể hiện cùng xu hướng cho việc chi tiêu (Spending Score).



3.4 Áp dụng thuật toán Mean shift cho bộ dữ liệu Shop Customer Data

3.4.1 Chính dạng dữ liệu

Trước khi xây dựng mô hình phân cụm dựa trên thuật toán Mean Shift, ta cần chuyển dạng dữ liệu các cột định danh sang định lượng bằng phương pháp Label Encoder:

```

1 label_encoder = LabelEncoder()
2
3 df['Gender'] = label_encoder.fit_transform(df['Gender'])
4 df['Profession'] = label_encoder.fit_transform(df['Profession'])

```

Biến Annual Income có giá trị lớn hơn nhiều khi so với các biến còn lại, điều này có thể khiến mô hình cho ra kết quả sai lệch, vì vậy, ta chuyển biến Annual Income về thang đo k\$ (nghìn đô la):

```

1 df['Annual Income'] = df['Annual Income']/1000

```

Kiểm tra lại dữ liệu sau khi thực hiện chuyển đổi:

```

1 df.head()

```

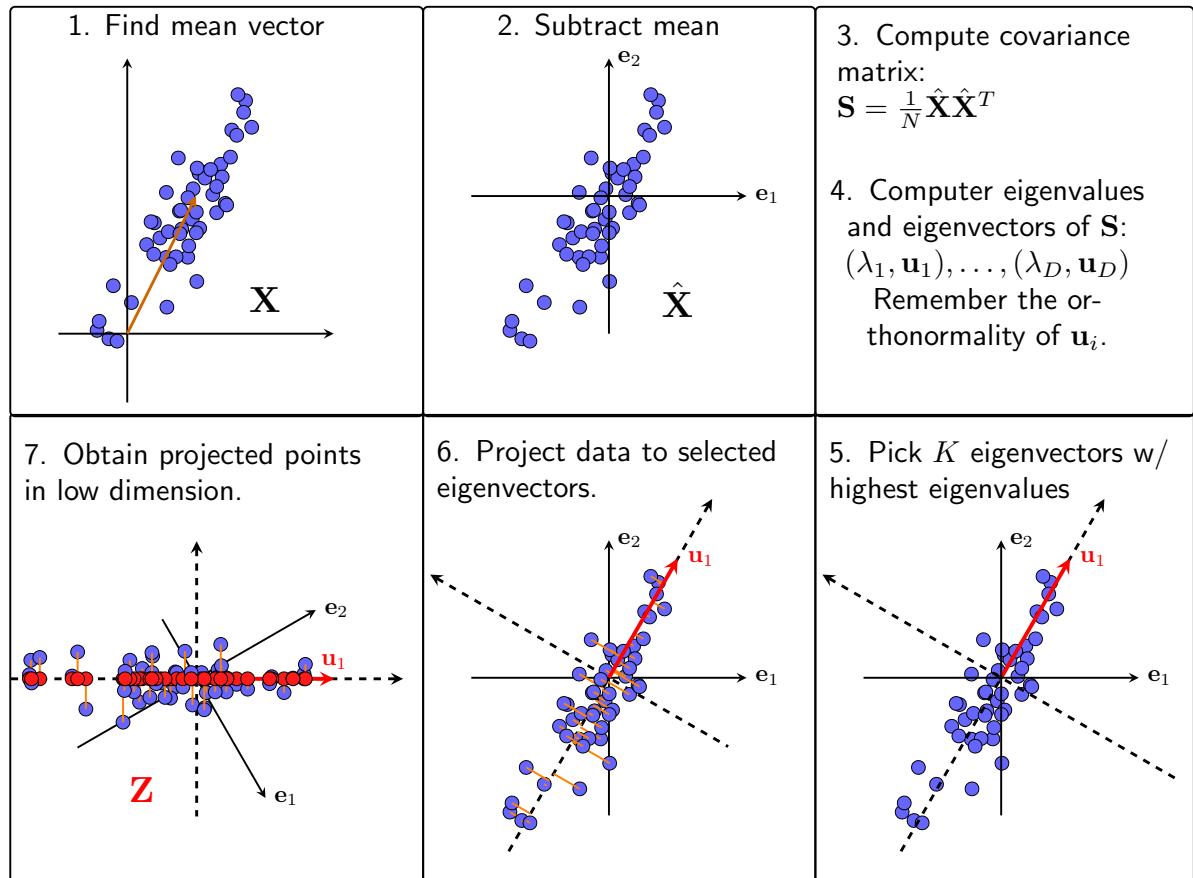
	Gender	Age	Annual Income	Spending Score	Profession	Work Experience	Family Size
0	1	19	15000	39	5	1	4
1	1	21	35000	81	2	3	3
2	0	20	86000	6	2	1	1
3	0	23	59000	77	7	0	2
4	0	31	38000	40	3	2	6

3.4.2 Giảm chiều dữ liệu bằng phương pháp phân tích thành phần chính PCA

Như đã đề cập, thuật toán Mean Shift không hiệu quả với số chiều lớn. Vì vậy, ta sẽ tiến hành giảm chiều dữ liệu bằng phương pháp PCA.

Phân tích thành phần chính (Principal Component Analysis (PCA)) là một thuật toán học máy đơn giản dựa trên các kiến thức của đại số tuyến tính. PCA đóng vai trò là một kỹ thuật cơ bản trong phân tích dữ liệu và giảm chiều dữ liệu. Việc thực hiện PCA giúp chuyển đổi dữ liệu nhiều chiều sang chiều thấp hơn trong khi vẫn giữ được thông tin cần thiết.

PCA procedure



Hình 8: Quy trình thực hiện PCA. Nguồn: Machine Learning Cơ Bản

Các bước thực hiện PCA như sau:

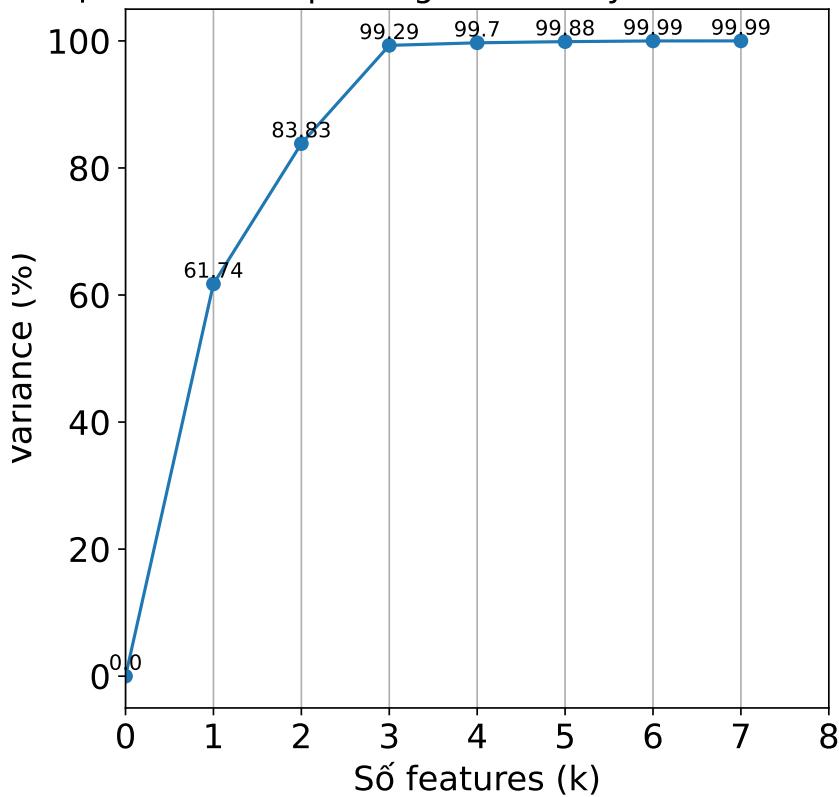
1. Tính vector trung bình $\bar{\mathbf{x}}$
2. Tính vector $\hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$
3. Tính $v = \text{covariance}(\hat{\mathbf{x}})$
4. Tính các trị riêng (eigenvalues) $\lambda_1, \dots, \lambda_n$ và vector riêng (eigenvectors) $\mathbf{u}_1, \dots, \mathbf{u}_n$ của v
5. Chọn k eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ có eigenvalues lớn nhất
6. Thực hiện phép chiếu $\mathbf{x}_1, \dots, \mathbf{x}_n$ và xuông cở sở $\mathbf{u}_1, \dots, \mathbf{u}_n$

Để có thể chọn được số k features phù hợp, ta sẽ tiến hành vẽ đồ thị biểu diễn phần trăm phương sai tích lũy theo k như sau:

```
1 from sklearn.decomposition import PCA
2 nb_features = 7
3 ## Áp dụng PCA (chưa xác định k --> giữ nguyên số chiều)
4 pca = PCA().fit(df)
5 # # Vẽ đồ thị biểu diễn % phương sai tích lũy theo số features --> chọn k theo
6 # → điểm "gãy"
7 points = np.cumsum(pca.explained_variance_ratio_) * 100 # Các điểm dữ liệu
8 points = np.insert(points, 0, 0) # Thêm điểm k = 0, variance = 0
9 x_i = np.arange(0, 8)
10 y_i = (points[-8:])//0.01/100
11
12 plt.figure(figsize = (6,6))
13 plt.plot(points, marker = 'o')
14 plt.xlabel('Số features (k)')
15 plt.ylabel('Variance (%)')
16 plt.title('Đồ thị biểu diễn % phương sai tích lũy theo số features (k)')
17 plt.xlim([0, nb_features + 1])
18 plt.grid(axis = 'x')
19 for i in x_i:
20     plt.text(i, y_i[i]+1, y_i[i], ha = 'center', va = 'baseline') # tung độ của
21     # → text cao hơn point 1 đơn vị
22 plt.show()
# %%: tổng phương sai của k cột/ tổng phương sai của tất cả các cột
```

Kể từ $k = 3$, ta nhận thấy số phần trăm phần trăm phương sai tích lũy tăng không nhiều khi k

Ô thị biểu diễn % phương sai tích lũy theo số features



tăng. Để kiểm chứng, ta sẽ tính phương sai tích lũy và % tăng của phương sai tích lũy theo k :

```

1 var = 0.0
2 for k in range(1, nb_features + 1):
3     pca = PCA(k)
4     pca.fit(df)
5
6     newVar = pca.explained_variance_ratio_.sum() * 100
7     print('    * k = %2d' %k, ': phương sai tích lũy ~ %.2f%%' %newVar,
8           '--> tăng ~ %.2f%%' %(newVar - var))
9     var = newVar

```

Từ $k = 3$ trở đi, phương sai tích lũy chỉ tăng thêm một lượng nhỏ ($<1\%$), vì vậy ta sẽ chọn $k = 3$ và thực hiện PCA cho bộ dữ liệu như sau:

```

1 pca = PCA(n_components = 3)
2 pca.fit(df)
3 X=pca.transform(df)

```

3.4.3 Xây dựng mô hình Mean Shift

Như đã đề cập ở [Phần 3.2.3](#), việc lựa chọn bandwidth để xây dựng mô hình phân cụm bằng thuật toán Mean Shift là vô cùng quan trọng nhưng đồng thời cũng rất khó để thực hiện. Trong thư viện sklearn, có một hàm hỗ trợ tính toán bandwidth đó là `sklearn.cluster.estimate_bandwidth`. Hàm này cho phép điều chỉnh tham số quantile để ước lượng bandwidth.

Quantile là một giá trị nằm trong khoảng từ 0 đến 1, xác định phần trăm dữ liệu mà ta muốn sử dụng để ước lượng bandwidth. Tham số này quyết định phạm vi dữ liệu được sử dụng để xác định KDE và từ đó tính toán bandwidth. Mỗi quantile sẽ cho ra một bandwidth khác nhau. Khi quantile được đặt là một giá trị nhỏ (gần 0), nghĩa là chỉ sử dụng một phần nhỏ dữ liệu để ước lượng, kết quả có thể cho ra bandwidth nhỏ hơn. Ngược lại, khi quantile được đặt là một giá trị lớn (gần 1), nghĩa là sử dụng một phần lớn dữ liệu để ước lượng, kết quả có thể cho ra bandwidth lớn hơn.

Vì thế, nhóm lựa chọn việc thay đổi quantile từ 0.1 đến 0.9 để tạo ra 9 giá trị bandwidth tương ứng. Sau đó lựa chọn bandwidth phù hợp dựa trên biểu đồ biểu diễn kết quả phân cụm và phương pháp Grid Search.

Lựa chọn bandwidth dựa trên biểu diễn kết quả phân cụm

Biểu diễn kết quả phân cụm dưới dạng biểu đồ 3D:

```
1 quantiles = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
2
3 fig, axes = plt.subplots(3, 3, figsize=(15, 15), subplot_kw={'projection':
4     '3d'})
5
6 # Lặp qua các giá trị quantile
7 for i, quant in enumerate(quantiles):
8     bw = estimate_bandwidth(X, quantile=quant)
9
10    # Áp dụng Mean Shift
11    ms = MeanShift(bandwidth=bw)
12    ms.fit(X)
13    labels = ms.labels_
14    cluster_centers = ms.cluster_centers_
15
16    # Biểu diễn scatter plot 3D
17    ax = axes[i // 3, i % 3]
18
19    # Biểu diễn từng điểm dữ liệu theo nhóm
```

```

19     for cluster_label in np.unique(labels):
20         cluster_points = X[labels == cluster_label]
21         ax.scatter(cluster_points[:, 0], cluster_points[:, 2],
22                    ↳ cluster_points[:, 1],
23                    label=f'Cluster {cluster_label}', alpha=0.5)
24
25     # Dánh dấu trung tâm cụm
26     ax.scatter(cluster_centers[:, 0], cluster_centers[:, 2], cluster_centers[:, 1],
27                marker='x', s=200, c='red', label='Cluster Centers')
28
29     ax.set_xlabel('Component 1')
30     ax.set_ylabel('Component 3')
31     ax.set_zlabel('Component 2')
32     ax.set_title(f'Quantile = {quant}, Bandwidth = {bw}')
33     ax.legend()
34
35 plt.tight_layout()
36 plt.show()

```

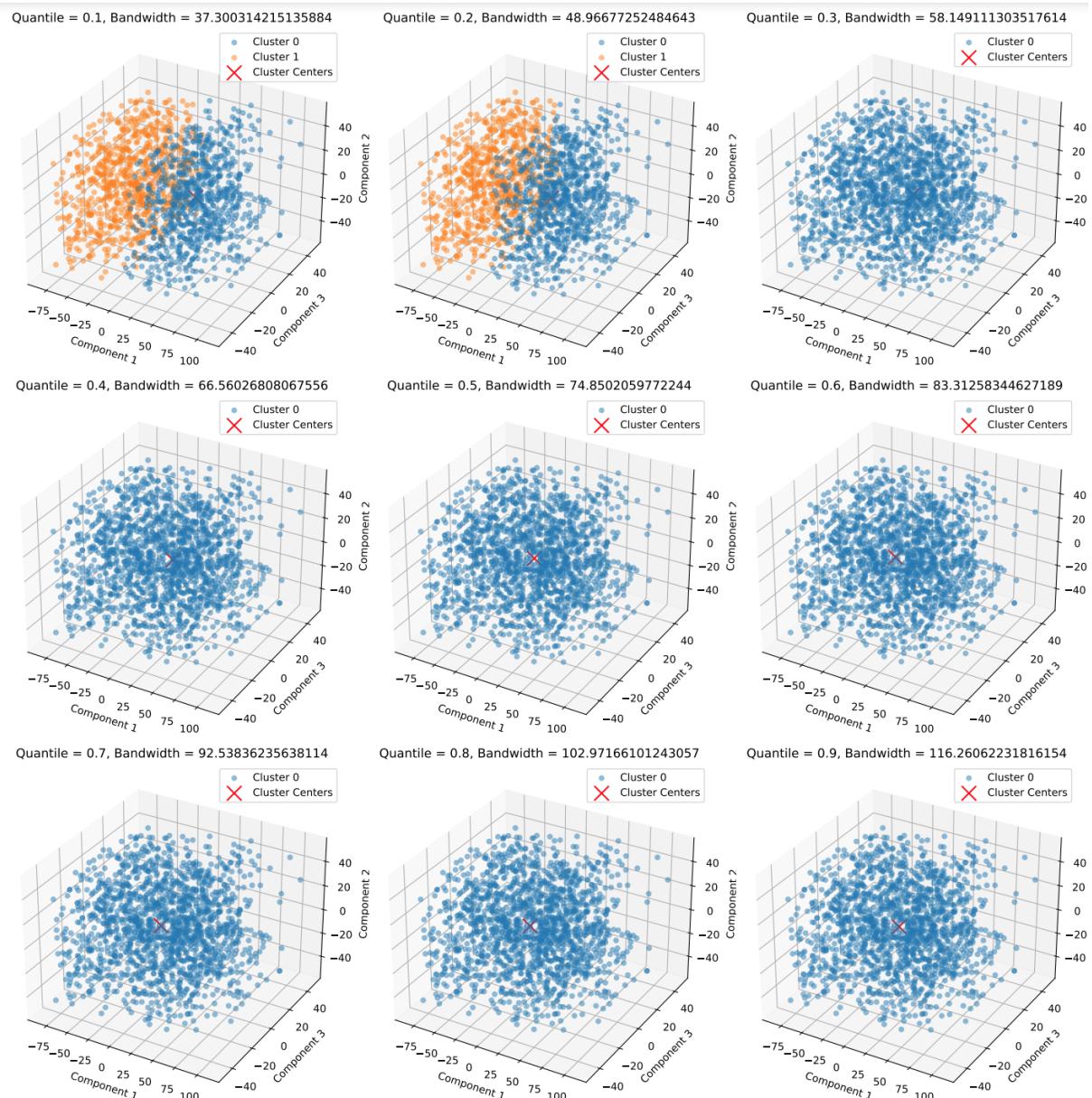
Sau khi quan sát biểu đồ, ta thấy rằng với quantile bằng 0.1 (tương ứng với bandwidth xấp xỉ 37.3) và quantile bằng 0.2 (tương ứng với bandwidth xấp xỉ 48.97), dữ liệu đều phân ra được 2 cụm. Các biểu đồ với quantile lớn hơn 0.2 đều chỉ cho ra được 1 cụm. Có thể thấy rằng, kết quả phân cụm tốt hơn khi quantile bằng 0.1 hoặc 0.2.

3.5 Lựa chọn bandwidth dựa trên phương pháp Grid Search

Grid search là một phương pháp tìm kiếm siêu tham số (*hyperparameter*). Siêu tham số là các thông số mà ta không thể học từ dữ liệu mà phải đặt trước trước khi huấn luyện mô hình. Grid search thường được áp dụng để tìm ra giá trị tốt nhất cho các siêu tham số bằng cách kiểm tra tất cả các tổ hợp có thể của chúng trong một phạm vi đã định sẵn.

Khi áp dụng grid search vào thuật toán phân cụm Mean Shift, quy trình sẽ tương tự. Đầu tiên, ta có thể lựa chọn các siêu tham số như bán kính kernel (kernel bandwidth) hoặc các tham số khác liên quan đến quá trình di chuyển các điểm dữ liệu. Grid search sẽ thử nghiệm một loạt các giá trị cho các siêu tham số này trên một lưới giá trị đã xác định trước, và sau đó đánh giá hiệu suất của mô hình với mỗi tổ hợp siêu tham số.

Ví dụ, nếu ta muốn tối ưu hóa bán kính kernel trong Mean Shift, ta có thể đặt một loạt giá trị cho bán kính, mô hình sẽ được huấn luyện và đánh giá bằng một phép đo hiệu suất nhất định (ví dụ như độ tách biệt giữa các cụm). Từ đó, Grid search sẽ tìm ra tổ hợp



Hình 9: Kết quả phân cụm của thuật toán Mean Shift với các quantile khác nhau

siêu tham số tốt nhất cho mô hình Mean Shift.

Để kiểm tra cụ thể quantile nào tốt nhất, ta sử dụng phương pháp Grid Search.

```

1  from sklearn.cluster import MeanShift
2  from sklearn.metrics import silhouette_score
3  from sklearn.model_selection import GridSearchCV
4
5  bw_list=[]
6  for quant in quantiles:
7      bw = estimate_bandwidth(X,quantile = quant)
8      bw_list.append(bw)
9

```

```

10 meanshift = MeanShift(bandwidth=estimate_bandwidth(X, quantile=0.2),
11     ↳ bin_seeding=True)
12
13 # Định nghĩa tham số grid search
14 param_grid = {'bandwidth': bw_list}
15
16 grid_search = GridSearchCV(meanshift, param_grid, scoring=silhouette_score)
17 grid_search.fit(X)
18
19 best_model = grid_search.best_estimator_
20
21 labels = best_model.predict(X)
22 best_model = grid_search.best_estimator_
23 best_params = grid_search.best_params_
24 silhouette_avg = silhouette_score(X, labels)
25
26 print("Best Params:", best_params)
27 print("Avg Score - Silhouette Score:", silhouette_avg)

```

Phương pháp Grid Search cho kết quả bandwidth bằng 37.3 là tốt nhất trong 9 bandwidth, với Silhouette score xấp xỉ 0.37. Như vậy, với quantile bằng 0.1, thuật toán Mean Shift cho kết quả phân cụm tốt nhất.

3.6 Phương pháp ELBOW

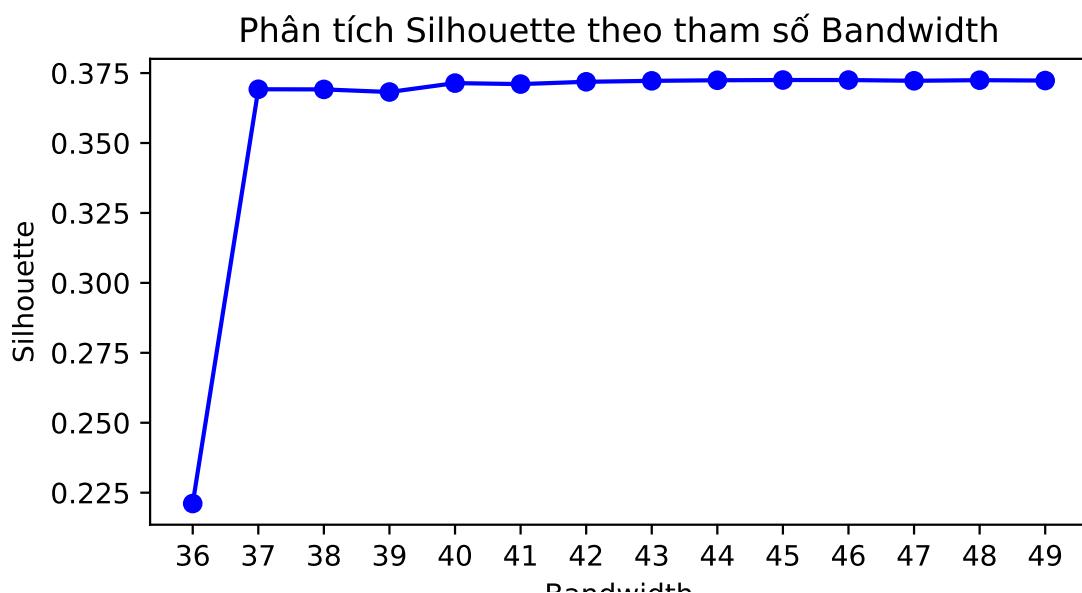
Phương pháp Elbow là phương pháp thường được dùng để phỏng đoán, xác định số lượng k cụm trong tập dữ liệu đối với các thuật toán phân cụm cần phải chỉ định trước tham số này ví dụ như K-Means. Ngoài xác định số cụm, phương pháp này có thể được sử dụng để xác định số thành phần chính trong bài toán PCA hay dùng để chọn số lượng tham số trong các mô hình dựa trên dữ liệu khác. ELBOW là “điểm gãy” của đồ thị nơi mà khi tham số k tăng lên thì hiệu quả tương ứng của mô hình không có quá nhiều cải thiện. Tại đây, lợi ích từ việc tăng giá trị tham số k không tương xứng với chi phí, do đó điểm này thường được xem là giá trị tối ưu cho tham số cần tìm. Đối với thuật toán Mean Shift, chúng ta không cần chỉ định số lượng cụm, thay vào đó tham số bandwidth là yếu tố quan trọng quyết định hiệu quả của mô hình phân cụm. Vì vậy, áp dụng phương pháp ELBOW để đánh giá hiệu quả phân cụm với các bandwidth khác nhau bằng điểm số Silhouette.

Từ kết quả lựa chọn bandwidth ở phần trên, nhóm chọn khoảng giá trị của bandwidth dùng để khảo sát là [36;49].

```

1 bandwidth = [float(i) for i in list(range(36,50))]
2 scores = []
3 labels_list = []
4 for i, b in enumerate(bandwidth):
5     ms2 = MeanShift(bandwidth=b, bin_seeding=True)
6     ms2.fit(X)
7     labels_list.append(ms2.labels_)
8     scores.append(silhouette_score(X, labels_list[i]))
9 ## Biểu diễn trực quan Inertia --> xác định elbow
10 plt.figure(figsize = (6, 3))
11 plt.plot(bandwidth, scores, 'bo-')
12 plt.xlabel('Bandwidth')
13 plt.xticks(bandwidth)
14 plt.ylabel('Silhouette')
15 plt.title('Phân tích Silhouette theo tham số Bandwidth')
16 plt.savefig('figs/elbow_bandwidth.pdf')
17 plt.show()

```



Hình 10: Kết quả phân cụm dữ liệu với số cụm bằng 2 của thuật toán Mean Shift

Có thể thấy, tương tự như kết quả của các phương pháp chọn bandwidth phía trên, “điểm gãy” cũng chính là bandwidth phù hợp là khoảng 37 (biểu đồ vẽ theo bandwidth đã làm tròn thành số nguyên, thực tế tham số này có thể là số thực), tương ứng với quantile bằng 0.1. Từ giá trị bandwidth 37 trở đi tuy điểm Silhouette có tăng nhưng không đáng kể. Vậy kết quả của phương pháp ELBOW cũng cỗ cho lựa chọn tham số bandwidth ở trên.

Biểu diễn kết quả phân cụm dưới dạng biểu đồ 2D

Sau khi đã chọn được bandwidth phù hợp, ta sẽ tiến hành phân cụm bằng thuật toán Mean Shift với bandwidth được ước lượng từ quantile bằng 0.1:

```
1 ms = MeanShift(bandwidth = estimate_bandwidth(X,quantile = 0.1))
2 ms.fit(X)
3 labels = ms.labels_
4 cluster_centers = ms.cluster_centers_
5
6 labels_unique = np.unique(labels)
7 n_clusters_ = len(labels_unique)
8
9 print(f"Số cụm : {n_clusters_}")
```

Số cụm : 2

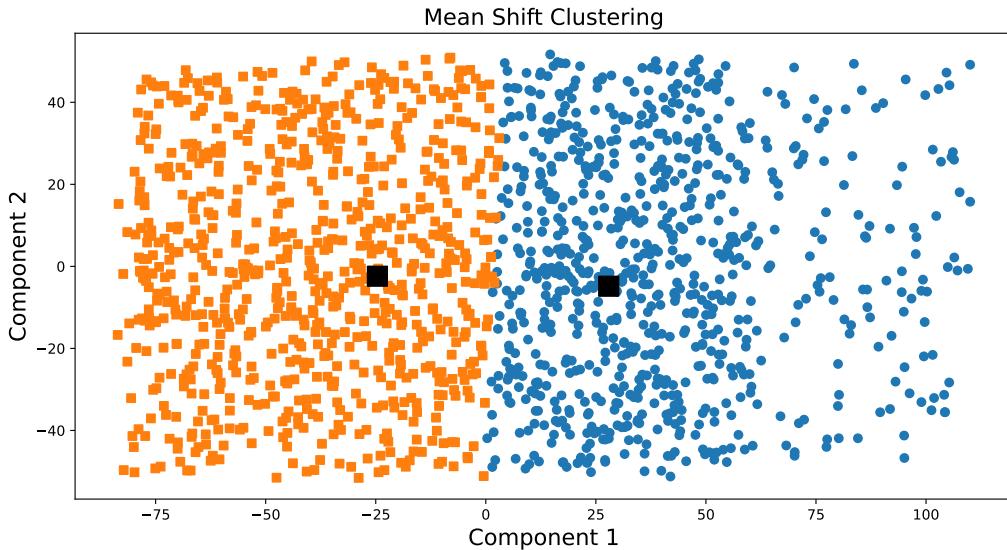
Tiếp theo ta sẽ tiến hành trực quan hóa dữ liệu đã được phân cụm. Kết quả phân cụm từ thuật toán Mean Shift có thể được quan sát như ở [Hình 10](#).

```
1 plt.figure(figsize=(12, 6))
2
3 markers = ['o', 's', '^', '*']
4
5 for cluster_label, marker in zip(range(n_clusters_), markers):
6     cluster_points = X[labels == cluster_label]
7     plt.scatter(
8         cluster_points[:, 0],
9         cluster_points[:, 1],
10        label=f'Cluster {cluster_label}',
11        marker=marker,
12    )
13
14 plt.scatter(
15     cluster_centers[:, 0],
16     cluster_centers[:, 1],
17     color='black',
18     marker=marker,
19     s=200,
20     label='Cluster Centers',
21 )
22
23 plt.xlabel('Component 1', fontsize=16)
24 plt.ylabel('Component 2', fontsize=16)
25 plt.title('Mean Shift Clustering', fontsize=16)
```

```

26 plt.savefig('figs/cluster_result_1.pdf')
27 plt.xticks(fontsize=16)
28 plt.yticks(fontsize=16)
29 plt.legend(fontsize=16, bbox_to_anchor=(1.02, 1), loc='upper left')
30 plt.show()

```



Hình 11: Kết quả phân cụm dữ liệu với số cụm bằng 2 của thuật toán Mean Shift

4 CHƯƠNG 4: ĐÁNH GIÁ THUẬT TOÁN PHÂN CỤM

4.1 Cơ sở lý thuyết

Các tiêu chí chung đối với một kết quả phân cụm tốt bao gồm *Intra-class*, thể hiện độ tương đồng của các điểm dữ liệu trong cùng một cụm cao và *extra-class (inter-class)*, cho biết độ tương đồng giữa các cụm thấp.

Có thể dùng các chỉ số sau để đo sự tương đồng giữa các đối tượng: Đối với đối tượng gồm n thuộc tính định lượng: giả sử $x = (x_1, x_2, \dots, x_m)$ và $y = (y_1, y_2, \dots, y_n)$ là hai đối tượng cần so sánh

Cosine: Giá trị này sẽ nằm trong khoảng từ -1 đến 1, trong đó 1 có nghĩa là hai đối tượng hoàn toàn giống nhau, -1 có nghĩa là hai đối tượng hoàn toàn khác nhau, và 0 có nghĩa là không có sự tương quan nào giữa hai đối tượng.

$$sim(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Trong đó x_i và y_i là các thuộc tính định lượng của đối tượng x và y . Tỷ số là tổng tích vô hướng các vector thuộc tính, mẫu số là căn bậc hai của tổng bình phương các thuộc tính cũng chính là độ dài của các vector. Việc chia cho mẫu số này cho phép đánh giá tốt hơn sự tương đồng giữa các vector có độ dài khác nhau. Ngoài ra còn có các đại lượng về khoảng cách cũng có thể dùng để kiểm tra sự tương đồng như: khoảng cách Minkowski, khoảng cách Manhattan, khoảng cách Euclid, khoảng cách Chebyshev, khoảng cách Canberra.

Đối với đối tượng gồm n thuộc tính định danh:

$$\text{sim}(x, y) = \frac{|x \cap y|}{n} = \frac{\sum_{i=1}^n e(x_i, y_i)}{n}$$

Trong đó $e(x_i, y_i)$ là một hàm chỉ ra mức độ tương tự giữa x_i và y_i trên một thuộc tính cụ thể. Nếu $x_i = y_i$, $e(x_i, y_i) = 1$; nếu $x_i \neq y_i$, $e(x_i, y_i) = 0$. Kết quả cuối cùng của công thức này sẽ là tổng của tất cả các giá trị hàm $e(x_i, y_i)$ chia cho số lượng thuộc tính, tạo ra một giá trị từ 0 đến 1 biểu thị mức độ tương tự giữa hai đối tượng. Giá trị càng gần 1, hai đối tượng càng giống nhau. Giá trị càng gần 0, hai đối tượng càng khác nhau.

Ngoài ra, ta cũng có thể dùng trọng số áp dụng cho công thức trên:

$$\text{sim}(x, y) = \frac{\sum_{i=1}^n w_i e(x_i, y_i)}{n} \quad w_i = |\text{DOM}(A_i)|$$

Trong đó w là trọng số của thuộc tính định danh. Trọng số này cho phép chúng ta đánh giá mức độ quan trọng của mỗi thuộc tính khi tính toán độ tương tự. Trọng số cao hơn sẽ cho thấy thuộc tính đó quan trọng hơn trong việc xác định mức độ tương tự giữa hai đối tượng. Trọng số thấp hơn sẽ cho thấy thuộc tính đó ít quan trọng hơn. Cần có kiến thức chuyên môn hoặc thông qua quá trình học máy để xác định trọng số.

Đánh giá hiệu suất của phân cụm (clustering) là một phần quan trọng trong quá trình phân tích dữ liệu. Có nhiều phương pháp đánh giá khác nhau, sau đây là một số phương pháp phổ biến:

4.2 Độ đo Silhouette Score

Silhouette là một thang đo hỗ trợ đánh giá hiệu quả phân cụm, được đề xuất bởi [22]. Điểm Silhouette đo lường độ né (compactness/cohesion) và độ phân tách (separation) các điểm dữ liệu trong các cụm. Tức là đo lường mức độ phù hợp của một điểm dữ liệu với cụm của chính nó và mức độ khác biệt của điểm dữ liệu này với các cụm khác gần nó. Phân cụm tốt là khi các cụm được phân tách tốt và đồng nhất bên trong.

Giá trị của Silhouette dao động từ -1 đến 1, giá trị càng cao cho biết đối tượng được kết hợp tốt với cụm của chính nó và không khớp với các cụm lân cận. Nếu hầu hết các điểm dữ liệu có giá trị silhouette cao thì mô hình phân cụm là tương đối phù hợp. Nếu nhiều điểm có giá trị thấp hoặc âm thì mô hình phân cụm có thể có quá nhiều hoặc quá ít cụm. Cụ thể nếu giá trị tiến về 1 thì phân cụm càng tốt; nếu giá trị bằng 0 tức là có sự chồng chéo (overlapping) giữa các cụm; giá trị tiến về -1 có nghĩa dữ liệu được phân cụm không phù hợp.

Điểm Silhouette được tính bằng công thức sau:

$$Silhouette_coef = \frac{(b_x - a_x)}{\max(a_x, b_x)}$$

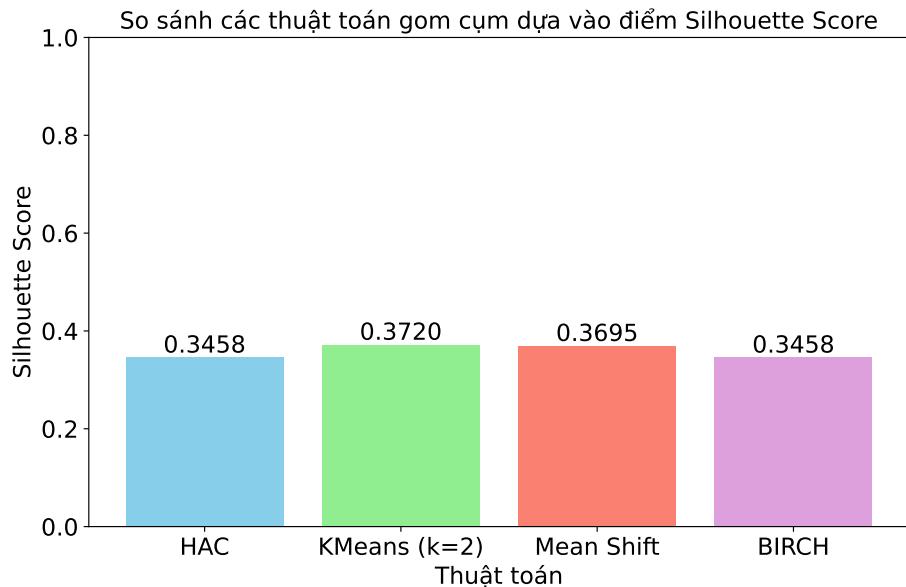
Trong đó: a_x là khoảng cách trung bình từ điểm dữ liệu x đến các quan sát cùng cluster. b_x là khoảng cách trung bình từ điểm dữ liệu x đến các quan sát thuộc cluster gần nhất.

Silhouette sẽ có giá trị cao hơn đối với các cụm dữ liệu có hình dạng lồi (*convex clusters*). Do đó, điểm Silhouette của các thuật toán phân cụm dựa trên mật độ có thể thấp nhưng không có nghĩa rằng mô hình phân cụm không tốt.

4.3 So sánh Mean Shift với các thuật toán phân cụm khác

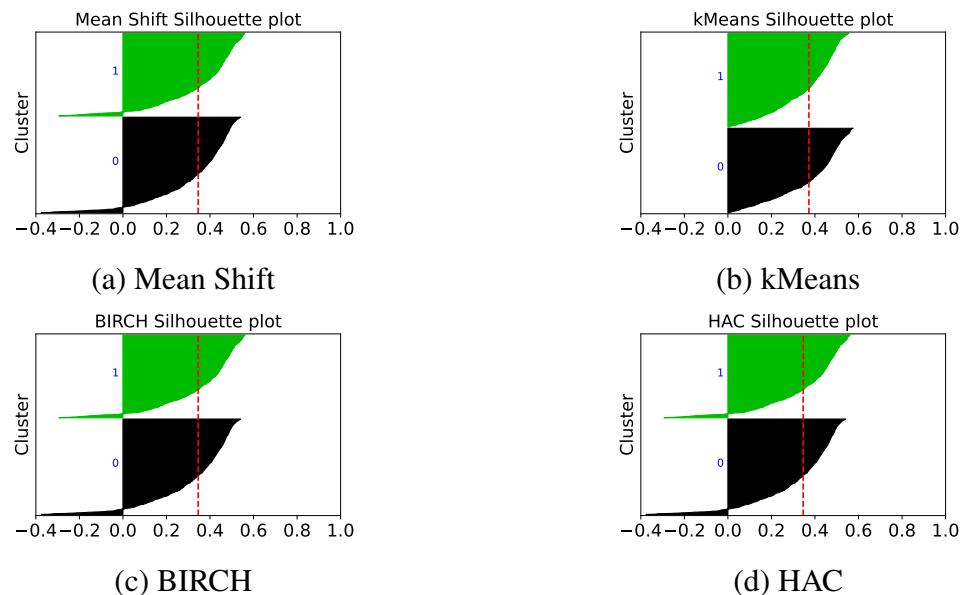
Ngoài Mean Shift, các thuật toán phân cụm phổ biến khác bao gồm HAC [20], kMeans [19], BIRCH [30]. Để xem xét hiệu quả của các thuật toán phân cụm khác nhau trên bộ dữ liệu Shop Customer Data, nhóm thực hiện huấn luyện các thuật toán khác và đánh giá thông qua Silhouette score và thu được kết quả như sau:

Điểm Silhouette chỉ ra rằng KMeans với số cụm bằng 2 và Mean Shift có kết quả tương đương và hiệu quả nhất trong việc phân cụm dữ liệu Shop Customer Data, với điểm số khoảng 0.37. Theo sau là HAC và BIRCH là 2 thuật toán cho ra kết quả phân cụm có điểm Silhouette thấp hơn với 0.3458. Để kiểm chứng thêm và trực quan hóa kết quả



Hình 12: Kết quả phân cụm dữ liệu của các thuật toán theo điểm Silhouette

Silhouette score, ta có thể vẽ biểu đồ như sau:

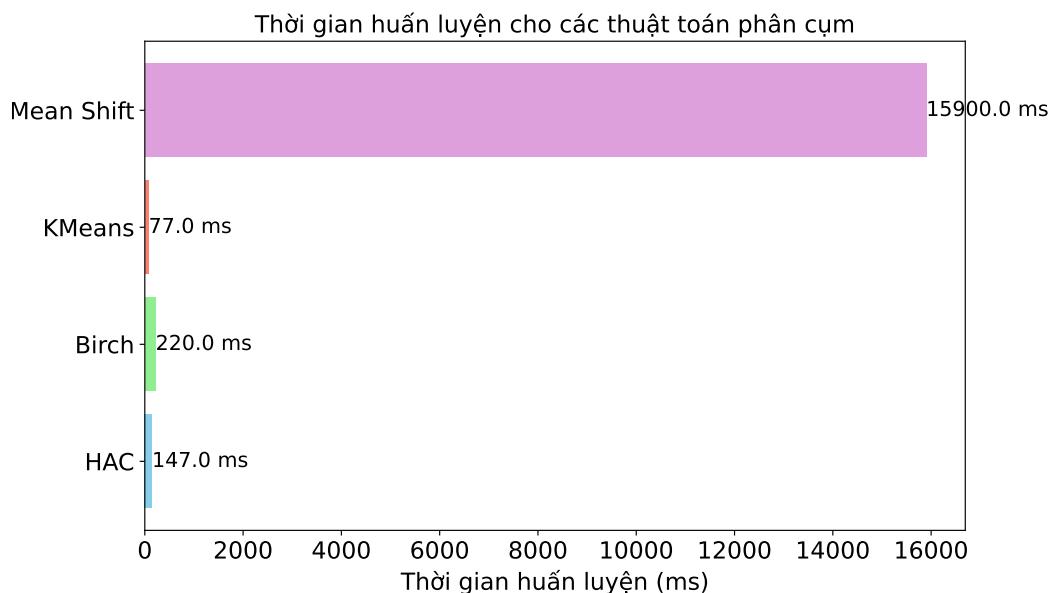


Hình 13: Trực quan hóa điểm Silhouette các thuật toán phân cụm

Từ biểu đồ trên, có thể thấy thuật toán Mean Shift với tham số bandwidth được chọn phân dữ liệu thành 2 cụm. Kết quả phân cụm đạt được điểm Silhouette trung bình (đường đứt đoạn màu đỏ) bé hơn 0.4 (cụ thể là 0.369). Giá trị này không âm có nghĩa là dữ liệu được phân cụm phù hợp. Tuy nhiên như đã trình bày, điểm Silhouette gần với 0 biểu thị có sự chồng chéo giữa các cụm. Thực tế biểu đồ cho thấy một số ít điểm dữ liệu thuộc cụm 1 (màu xanh lá) có điểm Silhouette âm, tức là chúng được phân cụm không phù hợp. Khi

so sánh cả 4 thuật toán có thể thấy không có quá nhiều chênh lệch về điểm số Silhouette giữa các thuật toán. Tuy nhiên KMeans cho ra kết quả phân cụm tốt nhất khi áp dụng với bộ dữ liệu này vì không xuất hiện điểm âm ở tất cả các điểm dữ liệu của 2 cụm.

Hơn nữa, dù có độ chính xác cao nhưng Mean Shift là thuật toán có thời gian huấn luyện rất chậm khi so sánh với các thuật toán phân cụm khác. Cụ thể thời gian huấn luyện của từng thuật toán trên bộ dữ liệu Shop Customer Data có thể được biểu diễn như biểu đồ sau:



Hình 14: Thời gian huấn luyện các thuật toán phân cụm

Có thể thấy sự khác biệt lớn về thời gian huấn luyện của thuật toán Mean Shift so với các thuật toán phân cụm còn lại. Điều này cho thấy dù Mean Shift có độ chính xác cao nhưng nếu xét thêm yếu tố thời gian, KMeans vẫn là thuật toán phân cụm hiệu quả nhất đối với bộ dữ liệu này.

5 CHƯƠNG 5: KẾT LUẬN VÀ Ý NGHĨA

Với đề tài **Áp dụng thuật toán Mean Shift cho bộ dữ liệu Shop Customer Data**, đầu tiên, nhóm đã tìm hiểu về bản chất, cách hoạt động của thuật toán Meanshift. Sau đó, nhóm sử dụng bộ dữ liệu “Shop Customer Data” để tiến hành xây dựng mô hình MeanShift phân cụm các khách hàng, từ đó tìm hiểu về phân khúc khách hàng cũng như đặc điểm của các nhóm khách hàng này. Cuối cùng, nhóm thực hiện đánh giá mô hình để đảm bảo tính hiệu quả và độ chính xác của quá trình phân cụm.

Trong quá trình phân tích và xây dựng mô hình, nhóm đã sử dụng đa dạng các phương pháp để tìm ra được băng thông (bandwidth) tốt nhất, phù hợp với mục tiêu xây dựng mô hình phân cụm các nhóm khách hàng hiệu quả. Sau khi đánh giá kết quả, để có sự so sánh và tìm hiểu thêm về hiệu quả của các thuật toán khác đối với mục tiêu phân cụm bộ dữ liệu, nhóm đã sử dụng thêm các thuật toán HAC, k-Means và BIRCH dựa trên Silhouette score và thời gian huấn luyện. Bên cạnh đó, đồ án còn sử dụng các kỹ thuật Label Encoding, PCA (giảm chiều dữ liệu) trước khi phân cụm để đảm bảo độ chính xác và tính hiệu quả của mô hình.

Thông qua việc xây dựng mô hình, đồ án đã cung cấp cái nhìn tổng quan và chi tiết hơn về các nhóm khách hàng cũng như những xu hướng của các thuộc tính. Từ đó đưa ra các nhận định hữu ích cho các chiến dịch quảng cáo và tri ân khách hàng. Điều này có thể đóng vai trò quan trọng trong quá trình đưa ra các quyết định kinh doanh và đề xuất cải thiện chất lượng bán hàng.

Tuy nhiên, đồ án vẫn tồn tại một số hạn chế. Trong quá trình phát triển đề tài, nhóm phải chấp nhận một rủi ro nhất định để bỏ đi các dữ liệu của các khách hàng bắt đầu đi làm khi dưới 16 tuổi. Sau khi đánh giá mô hình, nhóm nhận thấy điểm Silhouette còn thấp. Theo nhận định của nhóm, kết quả này có thể do Mean Shift chưa phải phương pháp tốt nhất để phân cụm cho bộ dữ liệu, hoặc vấn đề có thể xuất phát từ tính chính xác của các giá trị trong bộ dữ liệu.

Một điểm tích cực đó là mô hình phân cụm khách hàng đã đạt được những kết quả nhất định. Mô hình đã phân cụm thành công các khách hàng thành các nhóm có đặc điểm tương đồng về hành vi mua sắm. Đây là kết quả hữu ích để đơn vị kinh doanh hiểu rõ hơn về khách hàng, từ đó đưa ra các chiến lược kinh doanh, marketing và bán hàng hiệu quả.

BẢNG PHÂN CÔNG

Thành viên	Phân công	Đánh giá
Nguyễn Thị Tuyết	Tìm hiểu lý thuyết Mean Shift EDA Kết luận và ý nghĩa	100%
Vũ Nguyễn Thảo Vi	Tìm hiểu lý thuyết Mean Shift Code Grid Search thuật toán Mean Shift Đánh giá và trực quan hóa kết quả phân cụm	100%
Nguyễn Quốc Việt	Tìm hiểu lý thuyết Mean Shift Trình bày lý thuyết Mean Shift Code thuật toán Mean Shift So sánh các thuật toán phân cụm	100%
Bùi Quốc Việt	EDA Code Grid Search thuật toán Mean Shift	80%
Nguyễn Nhật Thảo Vy	Tổng quan Tiền xử lý dữ liệu Tìm hiểu lý thuyết Mean Shift Lý thuyết các thuật toán phân cụm	100%

Tài liệu

- [1] 2.3. Clustering — scikit-learn 1.3.2 documentation. <https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient>. Accessed: December 12, 2023.
- [2] sklearn.metrics.silhouette_score — scikit-learn 1.3.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. Accessed: December 12, 2023.
- [ibm] What is machine learning? <https://www.ibm.com/topics/machine-learning>. Accessed: December 14, 2023.
- [4] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998a). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98*, page 94–105, New York, NY, USA. Association for Computing Machinery.
- [5] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998b). Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105.
- [6] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999a). Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD '99*, page 49–60, New York, NY, USA. Association for Computing Machinery.
- [7] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999b). Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60.
- [8] Babaiean, A., Rastegar, S., Bandarabadi, M., and Rezaei, M. (2009). Mean shift-based object tracking with multiple features. In *2009 41st Southeastern Symposium on System Theory*, pages 68–72.
- [9] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.
- [10] Demirović, D. (2019). An Implementation of the Mean Shift Algorithm. *Image Processing On Line*, 9:251–268. <https://doi.org/10.5201/itol.2019.255>.
- [11] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from

incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

- [12] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.
- [13] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 226–231. AAAI Press.
- [14] Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, 2(2):139–172.
- [Frost] Frost, J. Interquartile range (iqr): How to find and use it. <https://statisticsbyjim.com/basics/interquartile-range/>. Accessed: December 12, 2023.
- [16] Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40.
- [17] Kaufmann, L. and Rousseeuw, P. (1987). Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416.
- [18] Kohonen, T. (2004). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.
- [19] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations.
- [20] Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms.
- [21] Patnaik, A. K., Bhuyan, P. K., and Krishna Rao, K. (2016). Divisive analysis (diana) of hierarchical clustering and gps data for level of service criteria of urban streets. *Alexandria Engineering Journal*, 55(1):407–418.
- [22] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.
- [23] Sheikholeslami, G., Chatterjee, S., and Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the*

24rd International Conference on Very Large Data Bases, VLDB '98, page 428–439, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- [24] Shmueli, G. et al. (2017). *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*. Wiley.
- [25] Wang, W., Yang, J., and Muntz, R. R. (1997). Sting: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, page 186–195, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [26] Wen, Z.-q. and Cai, Z.-x. (2006). Mean shift algorithm and its application in tracking of objects. In *2006 International Conference on Machine Learning and Cybernetics*, pages 4024–4028.
- [27] Wu, K.-L. and Yang, M.-S. (2007). Mean shift-based clustering. *Pattern Recognition*, 40(11):3035–3052.
- [28] Xu, R. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*.
- [29] Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.
- [30] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114.