ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KINH TẾ LUẬT

KHOA HỆ THỐNG THÔNG TIN

PHÂN TÍCH DỮ LIỆU VỚI R/ PYTHON

TÌM HIỂU VỀ THUẬT TOÁN K-MEANS

Giảng viên hướng dẫn: **ThS. Nguyễn Phát Đạt**

Nhóm thực hiện: **Bó Đũa**

Võ Nguyên Trúc Lâm (NT)	K204110572
Nguyễn Hữu Thuận	K194060821
Lê Thảo Giang	K204110562
Lê Phước Toàn	K204110586
Nguyễn Thị Bảo Trâm	K204110588
Thẩm Thị Tú Uyên	K204111792

Thành phố Hồ Chí Minh, 01 tháng 11 năm 2022

Mục lục

Danh mục hình ảnh:

PHẦN 1: TÌM HIỂU VỀ K-MEANS

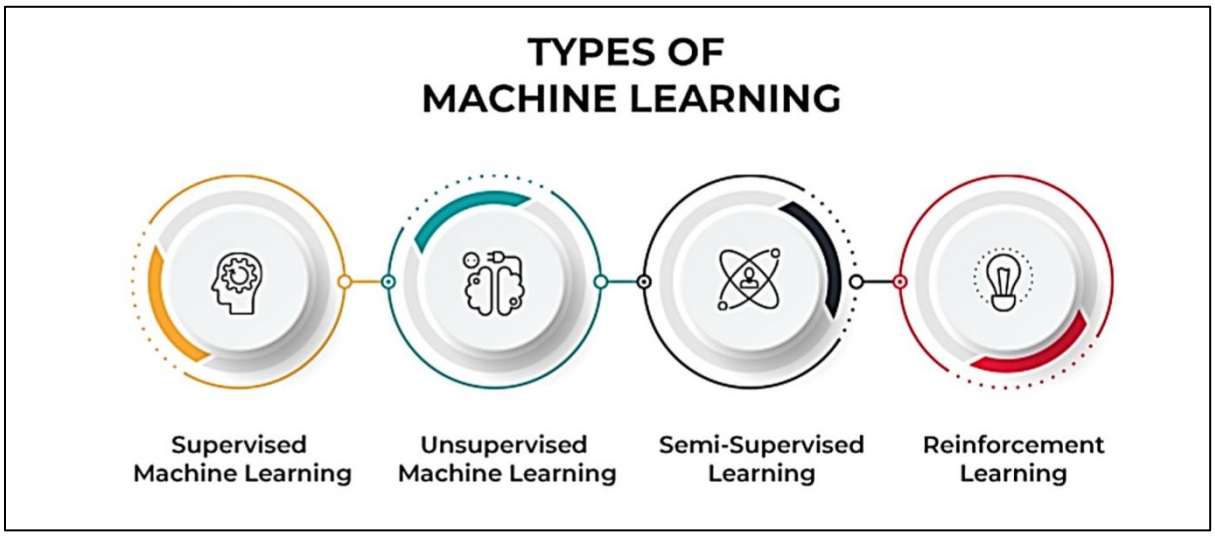
I. Giới thiệu tổng quát về Machine Learning

1. Học máy là gì?

Học máy (Machine Learning - ML) là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính. Lợi ích của ML là có thể dự đoán hoặc đưa ra quyết định mà không cần liệt kê tất cả các trường hợp cụ thể.

2. Các loại học máy:

Dựa theo trang web *Javapoint* , các loại phổ biến bao gồm:



Hình 1 1 Phân loại các phương pháp máy học

- Học có giám sát: nơi thuật toán tạo ra một hàm ánh xạ đầu vào thành đầu ra mong muốn. Một ứng dụng phổ biến của học máy có giám sát là bài toán phân loại: máy được yêu cầu học bằng cách xem một số ví dụ đầu vào-đầu ra của hàm.
- Học không giám sát: mô hình hóa một tập hợp các đầu vào: không có sẵn các ví dụ được gắn nhãn.
- Học bán giám sát: kết hợp cả các ví dụ được gắn nhãn và không được gắn nhãn để tạo ra một hàm hoặc bộ phân loại thích hợp.

– Học củng cố: Thuật toán học cách tự động xác định hành vi dựa trên hoàn cảnh, môi trường cung cấp phản hồi hướng dẫn thuật toán học tập để đạt kết quả tối ưu nhất.

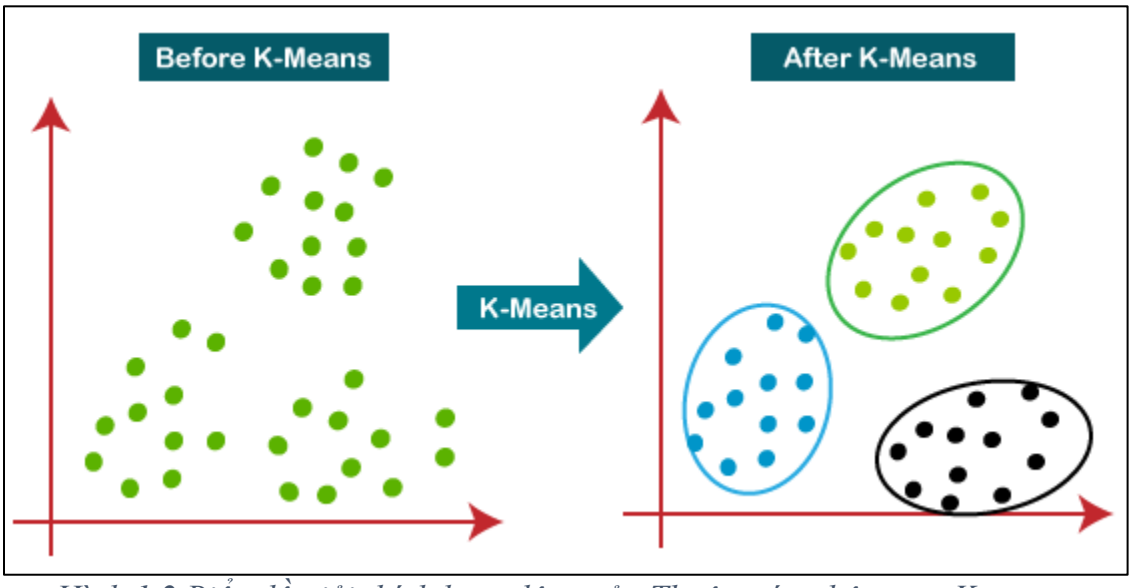
II. Giới thiệu về K-Means

1. Định nghĩa thuật toán phân cụm K-Means:

Thuật toán phân cụm K-Means là một thuật toán thuộc nhóm học máy không giám sát, nhóm các điểm dữ liệu không được gắn nhãn thành các cụm khác nhau sao cho các điểm dữ liệu mà trong cùng một cụm có các thuộc tính giống nhau.

Thuật toán phân cụm K-Means tính toán các tâm của mỗi cụm dữ liệu và lặp lại cho đến khi tâm mỗi cụm được tối ưu. Thuật toán cần giả định dữ liệu cần phân ra bao nhiêu cụm trước khi tiến hành phân cụm. K-Means còn được gọi là thuật toán phân cụm phẳng. Số lượng các cụm được tìm thấy từ dữ liệu bằng phương pháp này được ký hiệu bằng chữ 'K' trong K-Means.

Trong phương pháp này, các điểm dữ liệu được gán cho các cụm sao cho tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm mỗi cụm càng nhỏ càng tốt. Cần lưu ý rằng tính đa dạng giảm trong các cụm dẫn đến nhiều điểm dữ liệu giống hát nhau hơn trong cùng một cụm[1].



Hình 1 2 Biểu đồ giải thích hoạt động của Thuật toán phân cụm K-means

2. Các biến thể của thuật toán phân cụm K-Means:

– **K-medoids:** Tương tự thuật toán K-Means, mỗi cụm được đại diện bởi một trong các trọng số các đối tượng của cụm. Thông thường điểm gần trọng tâm sẽ được chọn làm điểm đại diện của cụm.

Tìm hiểu thêm về K-medoids: What is K-Medoids Clustering

– **Fuzzy C-means:** Có chiến lược phân cụm giống K-Means nhưng có điểm khác là một điểm dữ liệu có thể có nhiều cụm khác nhau. Fuzzy C-Means có khả năng phân cụm trong không gian đa chiều, có khả năng tối ưu hóa tâm cụm. Tuy nhiên thuật toán này có độ phức tạp tính toán lớn và tốc độ hội tụ phụ thuộc nhiều vào dữ liệu ban đầu.

Tìm hiểu thêm về Fuzzy C-means: Fuzzy C-Means Clustering

PHẦN 2: PHÂN TÍCH THUẬT TOÁN K-MEANS

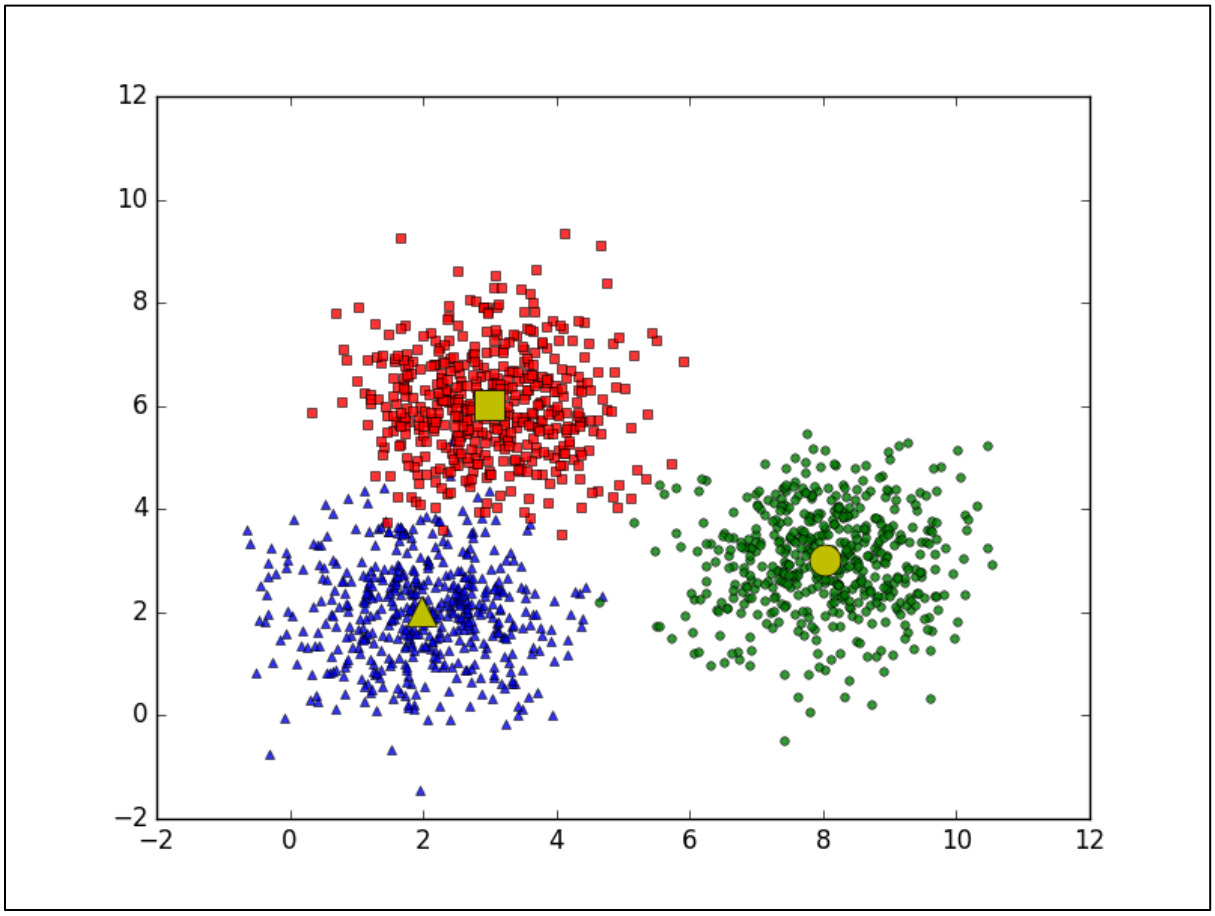
I. Mô hình giải thuật:

Thuật toán phân cụm K-Means là một trong những thuật toán phân cụm dữ liệu dựa trên học máy không giám sát được sử dụng nhiều trong các học máy nói chung và trong khai phá dữ liệu nói riêng.

Trong thuật toán phân cụm K-Means, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích của thuật toán là làm thế nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau. Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm ở gần nhau trong một không gian nào đó (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn).

Hiểu đơn giản K-Means: K là số cụm, mean là trung bình khoảng cách tới điểm nào đó. Với phương pháp phân cụm này, việc đầu tiên là chọn số cụm bất kỳ từ

tập dữ liệu - gọi là K. Giá trị K này là một số nguyên nhỏ (2, 3, 4, 5...) hoặc có thể lớn hơn[2].



Hình 2 1 Biểu đồ thể hiện sự phân tách giữa các cụm dữ liệu

1. Thuật toán K-Means:

- Bước 1: Chọn số cụm K;
- Bước 2: Chọn các điểm làm tâm K cụm (centroid);
- Bước 3: Tính khoảng cách các điểm so với tâm cụm và gán chúng vào các cụm gần nhất qua công thức:

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

- Bước 4: Tính lại giá trị trung tâm của từng cụm qua công thức:

$$m = \frac{(t1 + t2 + \cdots + tn)}{n}$$

– Bước 5: Lặp lại bước 3, nếu việc gán dữ liệu vào từng cụm không thay đổi so với vòng trước đó thì dừng thuật toán[3].

2. Các phương pháp tìm K:

a. Sử dụng phương pháp Khuỷu tay (Elbow Method):

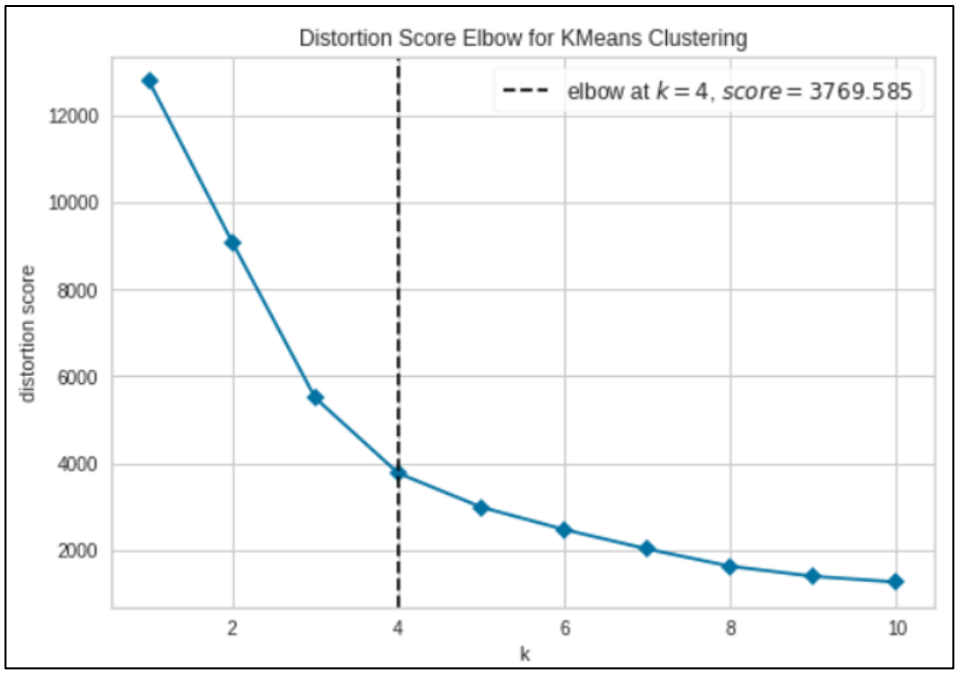
– Có nhiều cách để tìm ra con số tối ưu cho việc phân cụm, một phương pháp hữu dụng và thường được sử dụng để tìm số lượng phân cụm (số K) là phương pháp Khuỷu tay (Elbow Method), phương pháp này giúp ta lựa chọn được số lượng các cụm phù hợp dựa vào đồ thị trực quan hoá bằng cách nhìn vào sự suy giảm của hàm biến dạng và lựa chọn ra điểm khuỷu tay (elbow point).

– Ý tưởng của phương pháp Elbow về cơ bản là chạy phân cụm K-Means trên dữ liệu đầu vào cho một phạm vi giá trị của số cụm k (ví dụ: từ 1 đến 20) và cho mỗi giá trị k để sau đó tính toán tổng sai số bình phương (SSE) của cụm bên trong, là tổng khoảng cách của tất cả các điểm dữ liệu đến các trung tâm cụm tương ứng của chúng. Sau đó, giá trị SSE cho mỗi k được vẽ trong biểu đồ phân tán. Số lượng cụm tốt nhất là số mà tại đó giá trị SSE giảm, tạo ra một góc trong biểu đồ.

– Một số thuật ngữ:

+ Độ biến dạng (Distortion) hoặc SSE: Được tính bằng giá trị trung bình của các khoảng cách bình phương từ các tâm cụm của các cụm tương ứng. Thông thường việc tính khoảng cách sẽ sử dụng công thức tính khoảng cách Euclide.

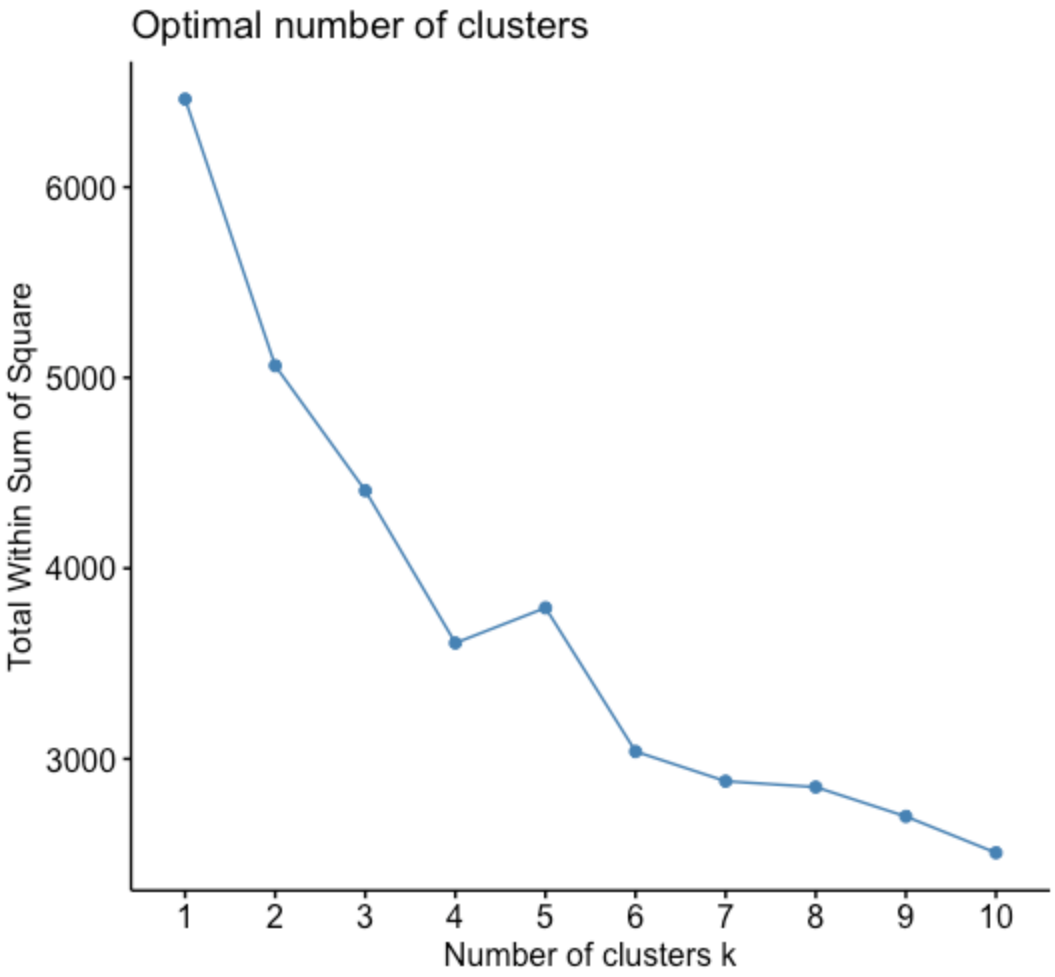
$$\boldsymbol{SSE} = \sum_{i=1}^{k} \sum_{j=0}^{n_j} Distance^2\ (x_{ij}, m_{i)}$$



Hình 2 2 Chọn K bằng phương pháp khuỷu tay (Elbow method)

+ Phương pháp Elbow được sử dụng để tìm khuỷu tay trong biểu đồ khuỷu tay. Phần khuỷu được tìm thấy khi tập dữ liệu trở nên phẳng hoặc tuyến tính sau khi áp dụng thuật toán phân tích cụm. Biểu đồ khuỷu tay cho thấy khuỷu tay tại điểm mà số lượng các cụm bắt đầu tăng lên.

– Phương pháp khuỷu tay (Elbow method) có thể không phải lúc nào cũng ứng dụng tốt cho tất cả các tập dữ liệu. Nếu không thể tìm thấy điểm K rõ ràng (elbow point) sau khi chạy thử, một cách tiếp cận khác có thể cân nhắc là Hệ số bóng (Silhouette coefficient)[4].



Hình 2 3 Phương pháp Khuỷu tay không tìm ra được điểm K trong trường hợp này

b. Sử dụng Hệ số bóng (Silhouette coefficient):

– Cách hệ số bóng hoạt động là đo mức độ gần của một điểm với các điểm lân cận gần nhất của nó, trên tất cả các cụm. Nó cung cấp thông tin về chất lượng phân cụm, thông tin này có thể được sử dụng để xác định xem liệu có nên thực hiện việc tinh chỉnh thêm bằng cách phân nhóm trên phân nhóm hiện tại hay không.

– Hệ số bóng cho một điểm dữ liệu mẫu được tính bằng công thức:

$$s = \frac{b - a}{\max(a, b)}$$

– Trong đó:

+ a: Trung bình khoảng cách từ điểm hiện tại tới tất cả điểm còn lại trong cụm

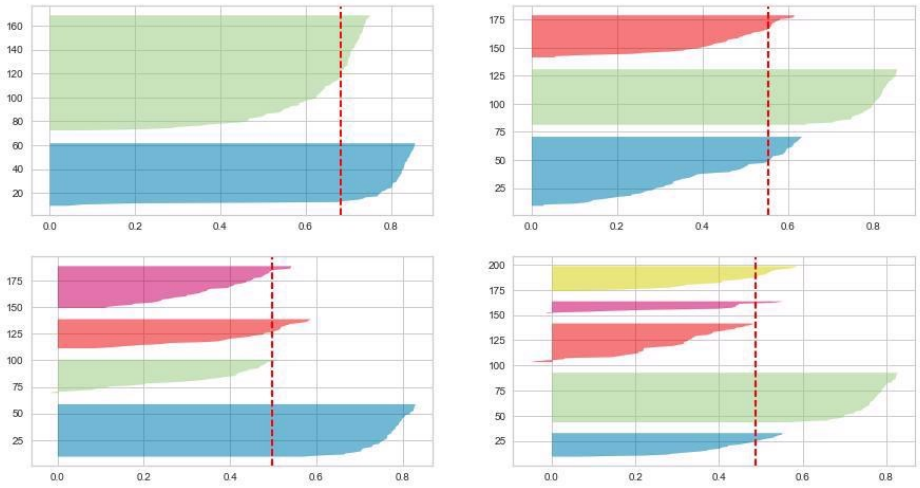
+ b: Trung bình khoảng cách điểm hiện tại tới tất cả các điểm của cụm gần nhất.

- Điểm hình bóng (Silhouette score) được tính bằng cách lấy trung bình của tất cả các dữ liệu mẫu. Điểm Silhouette cho một tập hợp các điểm dữ liệu mẫu được sử dụng để đo lường mức độ dày đặc và được phân tách rõ ràng của các cụm.
- Điểm hình bóng (Silhouette score) xem xét khoảng cách trong cụm giữa mẫu và các điểm dữ liệu khác trong cùng một cụm (a) và khoảng cách liên cụm giữa mẫu và cụm gần nhất tiếp theo (b).
- Điểm hình bóng (Silhouette score) nằm trong khoảng [-1, 1].
- Để đánh giá kết quả của điểm hình bóng (Silhouette score):

  + Điểm hình bóng càng tiến về 1 có nghĩa là các cụm rất dày đặc và được phân tách rõ ràng.

  + Điểm 0 có nghĩa là các cụm trùng nhau.

  + Điểm có giá trị âm có nghĩa là dữ liệu thuộc các cụm có thể bị sai / không chính xác.
- Ngoài phương pháp Khuỷu tay (Elbow method), đồ thị hình bóng cũng có thể được sử dụng để chọn giá trị tối ưu nhất của K (số cụm) trong phân cụm K-Means.
- Các khía cạnh cần chú ý trong biểu đồ Hình bóng là điểm số cụm thấp hơn điểm hình bóng trung bình, sự dao động lớn về kích thước của các cụm và cũng như độ dày của biểu đồ hình bóng.
- Sự khác biệt chính giữa Phương pháp Khuỷu tay và Hệ số bóng là Phương pháp Khuỷu tay chỉ tính toán khoảng cách trong khi Hệ số bóng có tính đến các biến như phương sai, độ lệch, chênh lệch cao thấp[5].



Hình 2 4 Chọn K bằng Hệ số bóng (Silhouette Coefficient)

– Đây là phân tích Silhouette được thực hiện trên các ô ở trên để chọn một giá trị tối ưu cho n clusters.

– Giá trị của n clusters là 4 và 5 có vẻ là không tối ưu đối với dữ liệu đã cho bởi các lý do sau:

+ Sự hiện diện của các cụm có điểm hình bóng dưới trung bình.

+ Biến động lớn về kích thước của các ô hình bóng.

– Giá trị 2 và 3 cho n clusters có vẻ là giá trị tối ưu. Điểm hình bóng cho mỗi cụm là điểm hình bóng trên trung bình. Ngoài ra, sự biến động về kích thước cũng tương tự. Độ dày của ô hình bóng đại diện cho mỗi cụm cũng là một điểm quyết định. Đối với ô có n cluster 3 (trên cùng bên phải), độ dày đồng đều hơn ô có n cluster là 2 (trên cùng bên trái) với độ dày của một cụm nhiều hơn nhiều so với độ dày khác. Do đó, chúng ta có thể chọn số lượng cụm tối ưu là 3.

II. Ứng dụng của thuật toán trong cuộc sống:

Thuật toán phân cụm K-Means là một trong những thuật toán đơn giản và nổi tiếng nhất của học máy. Nó là phương pháp tuy đơn giản nhưng đặc biệt hiệu quả trong bài toán học máy không giám sát, khi mà dữ liệu của bạn chưa được phân loại (unlabeled). Mục tiêu của thuật toán này là chia nhỏ dữ liệu của bạn

thành K nhóm dựa trên thuộc tính được cung cấp. Các điểm dữ liệu được xếp vào trong từng nhóm dựa trên sự giống nhau về đặc điểm nhận dạng.

Ứng dụng trong thực tế:

Vì khả năng cơ bản của thuật toán phân cụm K-Means là chia nhỏ dữ liệu ban đầu thành các nhóm nhỏ, tất cả hoạt động dựa trên thuật toán mà không yêu cầu bất kì kiến thức của người sử dụng về dữ liệu đã được thu thập (to - nhỏ, xấu - đẹp, méo - tròn). Nó có thể được sử dụng để xác nhận các giả thiết về việc nên phân chia làm bao nhiêu nhóm, là những nhóm nào, khi mà lượng dữ liệu thu được lớn và phức tạp. Khi mà 2 thông số trên được xác định, bất kì một sample mới sẽ dễ dàng được gán nhãn vào vị trí chính xác[6].

Đây là một thuật toán linh hoạt có thể được ứng dụng vào bất kì quy trình phân loại và chia nhóm. K-Means là một thuật toán rất đơn giản nhưng có rất nhiều ứng dụng trong thực tiễn. Một số ứng dụng của thuật toán này có thể kể đến như[7]:

- Phân khúc khách hàng trong kinh doanh.
- Phân tích gen trong y khoa.
- Sử dụng trong các bài toán Image segmentation
- Nén hình ảnh.
- Phát hiện tế bào ung thư.
- Phát hiện bất thường (anomaly detection).

PHẦN 3: ỨNG DỤNG K-MEANS VÀO BÀI TOÁN PHÂN KHÚC KHÁCH HÀNG

I. Phát biểu bài toán:

1. Lý do chọn bài toán:

Phân cụm khách hàng là quá trình phân chia khách hàng thành nhiều cụm/nhóm có chung sự tương đồng theo những tiêu chí như giới tính, tuổi tác, sở thích, thu nhập và thói quen chi tiêu, hành vi mua sắm… để doanh nghiệp có phương thức tiếp thị hiệu quả. Khi thực hiện được phân cụm khách hàng giúp đơn vị giải quyết đúng các yêu cầu của từng khách hàng, giúp tăng

lợi nhuận, giữ chân các khách hàng quan trọng, cũng như thực hiện các chiến dịch, chiến lược marketing hiệu quả hơn (Khajvand and Tarokh, 2011).

Mỗi khách hàng đều có những hành vi khác nhau và hành trình riêng biệt vậy nên một cách tiếp cận duy nhất thường không hiệu quả cho tất cả mọi người. Do đó, đòi hỏi cần có sự phân khúc khách hàng. Phân khúc khách hàng là quá trình bạn chia khách hàng thành các phân khúc dựa trên các đặc điểm chung - chẳng hạn như nhân khẩu học hoặc hành vi, vì vậy bạn có thể tiếp thị cho những khách hàng đó hiệu quả hơn.

Phân khúc khách hàng có thể được chia thành hai loại:

- Phân khúc khách hàng dựa trên họ là ai: quá trình hiểu khách hàng thường tập trung vào nhân khẩu học.
- Phân khúc khách hàng dựa trên những gì họ làm: Bạn cũng có thể phân khúc khách hàng dựa trên số tiền họ chi, tần suất và sản phẩm mà họ mua (điều này cho phép bạn xem bạn có thể tăng chi tiêu bao nhiêu). Loại phân khúc này tập trung nhiều hơn vào hành vi. Ở bài toán tiếp theo, chúng tôi tập trung vào dữ liệu này để phân khúc khách hàng[8].

2. Mô tả bài toán:

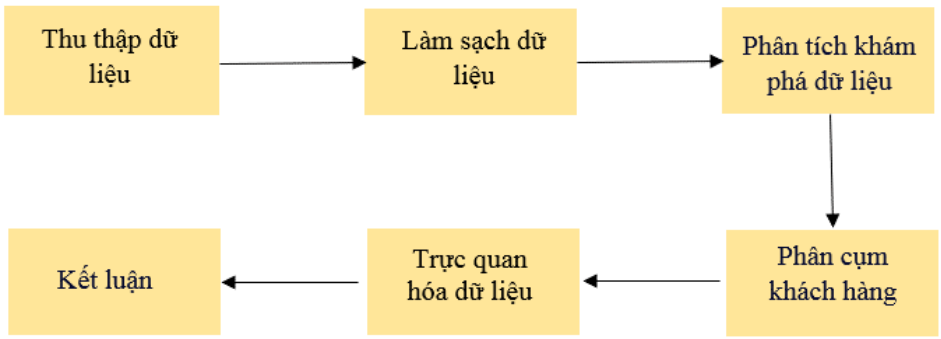
Đầu vào: Tập dữ liệu không có nhãn. Một bộ dữ liệu gồm tất cả các giao dịch xảy ra trong khoảng thời gian hơn một năm tại nhiều quốc gia của một cửa hàng bán lẻ trực tuyến có trụ sở tại Vương quốc Anh.

Mục đích của bài toán là trả lời các câu hỏi nghiên cứu dưới đây:

- Xu hướng bán hàng chung là gì?
- Giỏ hàng trung bình theo mỗi quốc gia là gì?
- Những quốc gia nào đang hoạt động tích cực nhất?
- Các sản phẩm bán chạy nhất là gì?
- Khách hàng mua sản phẩm thường xuyên như thế nào?
- Có bao nhiêu khách hàng mới mỗi tháng?
- Khi nào khách hàng có xu hướng mua sản phẩm?

Đầu ra: Các cụm dữ liệu đã được phân chia. Phân khúc khách hàng theo từng nhóm dựa trên hành vi của nhóm khách hàng.

3. Các bước thực hiện:



Hình 3 1 Các bước thực hiện bài toán

II. Mô tả bộ dữ liệu:

Bộ dữ liệu trong bài là bộ dữ liệu trên Kaggle với 516 384 dòng và 8 cột. Đây là tập dữ liệu giao dịch trực tuyến trong vòng 1 năm của một doanh nghiệp tại Anh chuyên về sản phẩm quà tặng, khách hàng của họ đa số là những người bán. Tập dữ liệu gồm 8 trường dữ liệu là:

- InvoiceNo: mã số hóa đơn.
- StockCode: mã số sản phẩm.
- Description: Tên sản phẩm.
- Quantity: Số lượng mỗi sản phẩm trên mỗi giao dịch.
- InvoiceDate: Thời gian tạo hóa đơn.
- UnitPrice: Đơn giá sản phẩm.
- CustomerID: Mã khách hàng.
- Country: Tên đất nước của khách hàng.

Các cột có kiểu dữ liệu như sau:

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
TotalPrice            float64
dtype: object
```

Hình 3 2 Các kiểu dữ liệu

=> Dựa vào những thông tin này, doanh nghiệp có thể hiểu khách hàng của mình hơn bằng cách gom thành các cụm khách hàng để phân loại khách hàng từ đó đề ra những chiến lược tiếp thị phù hợp cho từng phân khúc.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...

Hình 3 3 Mô tả tập dữ liệu

III. Sử dụng K-Means và các bước thực hiện để giải quyết bài toán:

Đối với bài toán phân khúc khách hàng này, nhóm thực hiện kết hợp thêm phân tích RFM và K-Means để giải quyết bài toán trên.

Phân tích RFM là một kỹ thuật tiếp thị được sử dụng để xếp hạng định lượng và phân nhóm khách hàng dựa trên số lần mua hàng gần đây, tần suất và tổng số tiền của các giao dịch gần đây của họ để xác định những khách hàng tốt nhất và thực hiện các chiến dịch tiếp thị cho từng tệp khách hàng mục tiêu. Hệ thống ấn định điểm số của từng khách hàng dựa trên các yếu tố này để đưa ra phân tích khách quan. Phân tích RFM dựa trên câu nói nổi tiếng trong marketing rằng "80% doanh thu của doanh nghiệp đến từ 20% khách hàng."

Phân tích RFM xếp hạng từng khách hàng dựa trên các yếu tố sau:

- Lần mua hàng gần đây (Recency). Lần mua hàng cuối cùng của khách hàng là khi nào? Những khách hàng gần đây đã mua hàng sẽ vẫn còn lưu ý đến sản phẩm và có nhiều khả năng mua hoặc sử dụng lại sản phẩm. Các doanh nghiệp thường đo lường lần truy cập gần đây theo ngày. Tuy nhiên, tùy thuộc vào sản phẩm, họ có thể đo bằng năm, vài tuần hoặc thậm chí vài giờ.
- Tần suất (Frequency). Mức độ thường xuyên mà khách hàng này mua hàng trong một khoảng thời gian nhất định? Những khách hàng đã mua một lần thường có nhiều khả năng mua lại. Ngoài ra, khách hàng lần đầu tiên có thể là mục tiêu tốt cho quảng cáo tiếp theo để họ trở thành khách hàng thường xuyên.
- Số tiền chi tiêu (Monetary): Khách hàng đã chi bao nhiêu tiền trong một khoảng thời gian nhất định? Khách hàng chi nhiều tiền thì có nhiều khả năng chi tiêu trong tương lai và có giá trị cao đối với một doanh nghiệp.

Link file google colab: Seminar K-Means customer segmentation

1. **Khai báo thư viện và đọc dữ liệu:**
   - Khai báo thư viện:

[Input]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from yellowbrick.cluster import KelbowVisualizer
from sklearn.cluster import Kmeans
from sklearn.preprocessing import StandardScaler
from scipy import stats
from mpl_toolkits import mplot3d
import datetime as dt
```

– Đọc dữ liệu:

[Input]:

```
df=pd.read_csv("/content/drive/MyDrive/SEMINAR KMEANS /data.csv", encoding = 'ISO-8859-1')
df.head()
```

[Output]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

– Mô tả dữ liệu:

[Input]:

```
df.dtypes
```

[Output]:

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
TotalPrice            float64
dtype: object
```

– Kiểm tra các hàng có giá trị trống:

[Input]:

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

[Output]:

```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

– Chuyển đổi kiểu dữ liệu InvoiceDate:

[Input]:

```
df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])
df = df[(df["InvoiceDate"].dt.year!=2011)|(df["InvoiceDate"].dt.month!=12)]
df
```

[Output]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...
516379	C579886	22197	POPCORN HOLDER	-1	2011-11-30 17:39:00	0.85	15676.0	United Kingdom
516380	C579886	23146	TRIPLE HOOK ANTIQUE IVORY ROSE	-1	2011-11-30 17:39:00	3.29	15676.0	United Kingdom
516381	C579887	84946	ANTIQUE SILVER T-LIGHT GLASS	-1	2011-11-30 17:42:00	1.25	16717.0	United Kingdom
516382	C579887	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	-1	2011-11-30 17:42:00	7.95	16717.0	United Kingdom
516383	C579887	23490	T-LIGHT HOLDER HANGING LOVE BIRD	-3	2011-11-30 17:42:00	3.75	16717.0	United Kingdom

516384 rows × 8 columns

2. Làm sạch dữ liệu:

– Xóa các hàng có giá trị trống:

[Input]:

```
df[df.isin(["NaN","missing","?","??"]).any(axis=1)].shape[0]
```

[Output]: 56

[Input]: df = df[df.isin(["NaN","missing","?","??"]).any(axis=1) == False]

– Xử lý định dạng mô tả:

[Input]:

```
df["Description"].nunique()
```

[Output]: 4211

[Input]:

```
df["Description"].str.lower().str.strip().nunique()
```

[Output]: 4183

– Một số mô tả (Description) có nội dung chỉ liên quan đến các loại phí chứ không biểu thị một mặt hàng nào, nên chúng ta có thể xóa chúng:

[Input]:

```
df = df[df["Description"].isin(["amazon fee", "samples", "postage", "packing charge","manual","discount","adjust bad debt","bank charges","cruk commission","next day carriage"]) == False]
```

– Xóa những giá trị rỗng trong các trường "Unit Price" và "Quantity". Một vài mặt hàng (items) có số lượng (quantity) và đơn giá (unit price) bằng 0. Nhóm sẽ tiến hành xóa những hàng như vậy:

[Input]:

```
df = df[(df["UnitPrice"]!=0)&(df["Quantity"]!=0)]
```

– Giải quyết những dữ liệu không nhất quán. Nhóm tiến hành kiểm tra "Stock Code" có luôn dẫn tới một mô tả "Description" cố định hay không?

[Input]:

```
df["Description"].nunique()
```

[Output]: 4018

[Input]:

```
df["StockCode"].nunique()
```

[Output]: 3924

[Input]:

```
df.groupby("StockCode")["Description"].nunique().sort_values(ascending = False)
```

[Output]:

```
StockCode
23236           4
23196           4
23131           3
23413           3
23370           3
               ..
22419           1
22420           1
22421           1
22422           1
gift_0001_50    1
Name: Description, Length: 3924, dtype: int64
```

– Nếu cùng là một sản phẩm nhưng được viết khác nhau, nhóm chỉ giữ lại dòng mô tả (Description) đầu tiên xuất hiện:

[Input]:

```
for stack_code in df["StockCode"].unique():
    first_description = df[df["StockCode"]==stack_code]["Description"].uniqu
e()[0]
    df.loc[df["StockCode"]==stack_code, "Description"] = first_description
```

– Kiểm tra xem mô tả có luôn được liên kết với một mã hàng tương ứng hay không:

[Input]:

```
df.groupby("Description")["StockCode"].nunique().sort_values(ascending = False)
```

[Output]:

```
Description
metal sign,cupcake single hook      6
set of 4 fairy cake placemats       4
columbian candle round              3
pink stitched wall clock            2
woven berries cushion cover         2
                                   ..
french style storage jar bonbons    1
french style storage jar cafe       1
french style storage jar jam        1
french toilet sign blue metal       1
zinc wire sweetheart letter tray    1
Name: StockCode, Length: 3791, dtype: int64
```

[Input]:

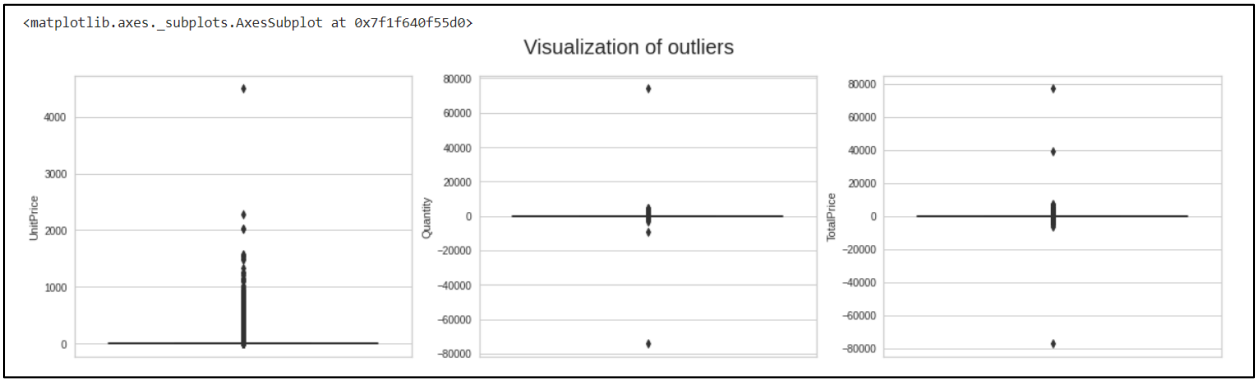
```
for description in df["Description"].unique():
        first_code = df[df["Description"]==description]["StockCode"].unique()[0]
df.loc[df["Description"]==description, "StockCode"] = first_code
```

– Xử lý các yếu tố dữ liệu ngoại lai:

[Input]:

```
df["TotalPrice"] = df["Quantity"]*df["UnitPrice"]

fig = plt.figure(figsize = (20,5))
fig.suptitle("Visualization of outliers",size=20)

axes = fig.add_subplot(1, 3, 1)
sns.boxplot(data=df,y="UnitPrice")

axes = fig.add_subplot(1, 3, 2)
sns.boxplot(data=df,y="Quantity")

axes = fig.add_subplot(1, 3, 3)
sns.boxplot(data=df,y="TotalPrice")
```

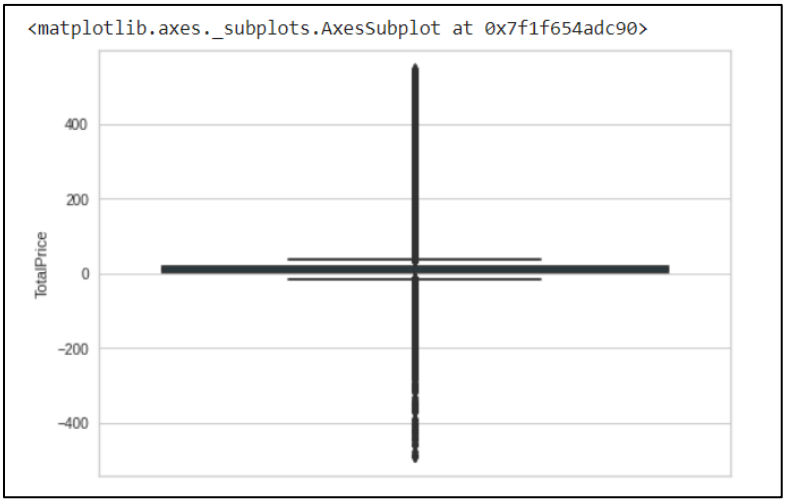
[Output]:



[Input]:

```
df = df[(np.abs(stats.zscore(df[["TotalPrice"]])) < 3).all(axis=1)]
sns.boxplot(data=df,y="TotalPrice")
```

[Output]:



3. **Phân tích khám phá dữ liệu (EDA):**
   - Xu hướng bán hàng chung là gì? Để nắm rõ được xu hướng, nhóm tiến hành lập biểu đồ số tiền được tạo ra bởi doanh số bán hàng ở mỗi ngày trong khoảng thời gian nghiên cứu:

[Input]:

```
general_trend = pd.DataFrame(data={'Date':pd.to_datetime(df.InvoiceDate).dt.date,'Total price':df.Quantity*df.UnitPrice})
general_trend = general_trend.groupby("Date")["Total price"].sum()
```

```
general_trend = pd.DataFrame(general_trend)
general_trend
```

[Input]:

```
dates = []
dates.append(pd.to_datetime("201012",format="%Y%m"))
dates += [pd.to_datetime("2011"+str(month),format="%Y%m") for month in rang
e(1,12)]
dates
```

[Output]:

```
[Timestamp('2010-12-01 00:00:00'),
 Timestamp('2011-01-01 00:00:00'),
 Timestamp('2011-02-01 00:00:00'),
 Timestamp('2011-03-01 00:00:00'),
 Timestamp('2011-04-01 00:00:00'),
 Timestamp('2011-05-01 00:00:00'),
 Timestamp('2011-06-01 00:00:00'),
 Timestamp('2011-07-01 00:00:00'),
 Timestamp('2011-08-01 00:00:00'),
 Timestamp('2011-09-01 00:00:00'),
 Timestamp('2011-10-01 00:00:00'),
 Timestamp('2011-11-01 00:00:00')]
```

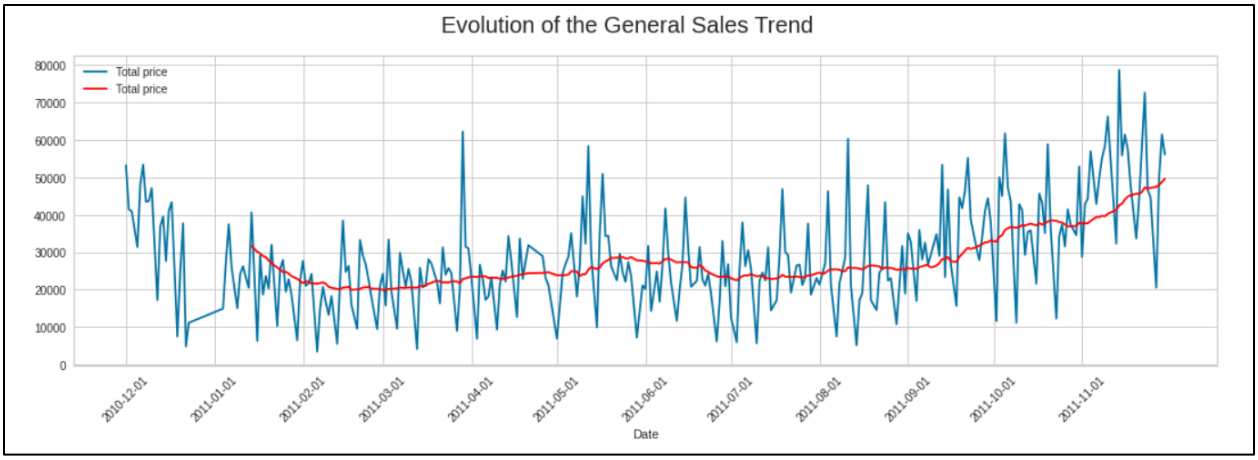
[Input]:

```
rolling_days = general_trend.copy()
rolling_days["Total price"] = rolling_days["Total price"].rolling(window=30).mea
n()
```

[Input]:

```
plt.figure(figsize = (18,5)).suptitle('Evolution of the General Sales Trend', fontsize
=20)
sns.lineplot(data=general_trend)
sns.lineplot(data=rolling_days, palette=['red'])
plt.xticks(dates,rotation = 45)
plt.show()
```

[Output]:



[Input]:

```
general_trend.index = pd.to_datetime(general_trend.index)
general_trend_months = general_trend.groupby([general_trend.index.year,general
_trend.index.month])['Total price'].sum()
general_trend_months = pd.DataFrame(general_trend_months)
general_trend_months
```

[Output]:

		Total price
Date	Date	
2010	12	694538.760
2011	1	539326.370
	2	485332.440
	3	650498.050
	4	480575.661
	5	700656.130
	6	627183.140
	7	634301.311
	8	659839.890
	9	911183.202
	10	978138.660
	11	1322569.040

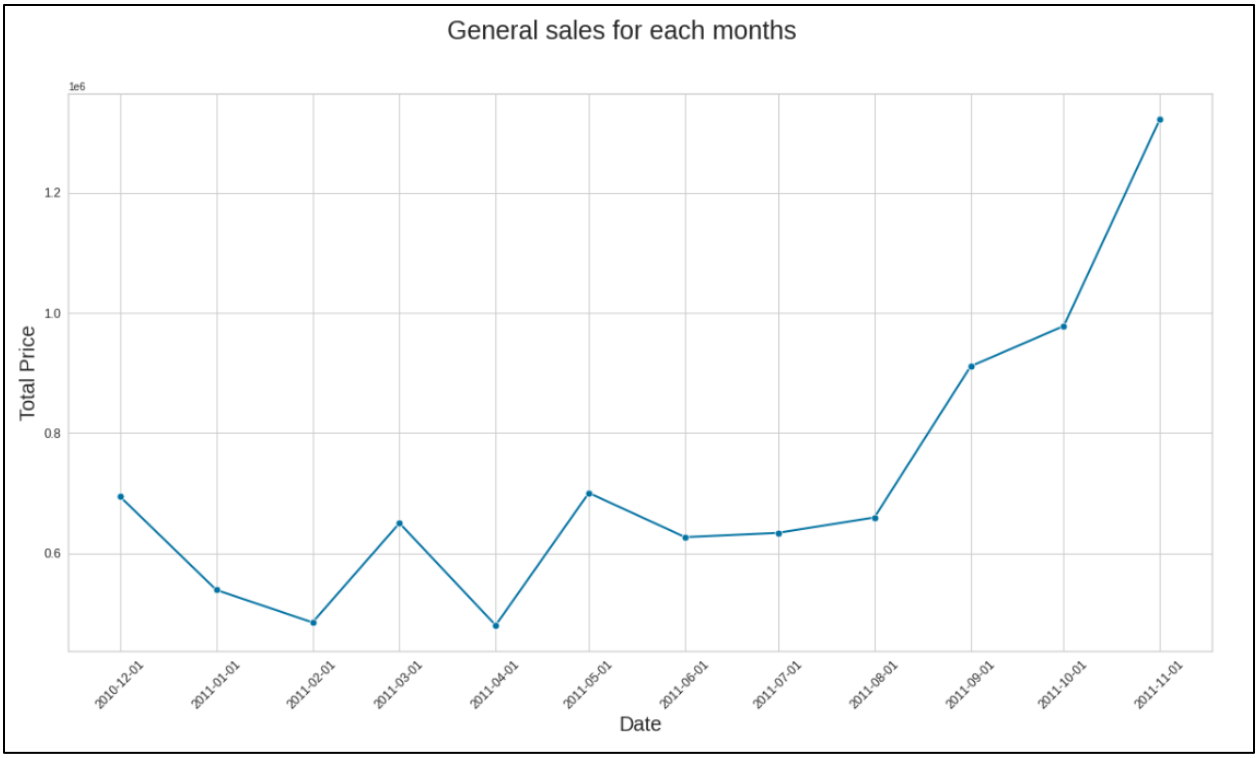
[Input]:

```
general_trend_months["Date"] = dates
```

[Input]:

```
plt.figure(figsize = (20,10)).suptitle('General sales for each months', fontsize=25)
lineplot = sns.lineplot(data=general_trend_months,x="Date", y="Total price", mar
ker="o",linewidth = 2)
lineplot.set_xlabel(xlabel="Date",size = 20)
lineplot.set_ylabel(ylabel="Total Price",size = 20)
plt.yticks(fontsize=12)
plt.xticks(dates,rotation = 45,fontsize=12)
plt.show()
```

[Output]:



[Input]:

df.Description.value_counts()[:10]

[Output]:

```
WHITE HANGING HEART T-LIGHT HOLDER    2284
REGENCY CAKESTAND 3 TIER              2102
JUMBO BAG RED RETROSPOT               2090
PARTY BUNTING                         1673
LUNCH BAG RED RETROSPOT               1595
ASSORTED COLOUR BIRD ORNAMENT         1451
SET OF 3 CAKE TINS PANTRY DESIGN      1422
SMALL POPCORN HOLDER                  1396
PACK OF 72 RETROSPOT CAKE CASES       1348
LUNCH BAG SUKI  DESIGN                1314
Name: Description, dtype: int64
```

- Mặt hàng bán chạy nhất là "white hanging heart T-light holder".
- Chúng ta có thể tính toán mức giao dịch trung bình hằng tháng của mỗi khách hàng trên phạm vi toàn cầu:

[Input]:

```
count_transactions_per_country = df.groupby([df.Country,df.InvoiceDate]).Description.count().reset_index()
count_transactions_per_country = count_transactions_per_country.groupby([count_transactions_per_country.Country]).Description.sum()
count_transactions_per_country = pd.DataFrame(count_transactions_per_country)

unique_per_country = df.groupby([df.Country]).CustomerID.nunique()
unique_per_country = pd.DataFrame(unique_per_country)

transactions_per_customer_per_countries = pd.concat([count_transactions_per_country,unique_per_country],axis=1).reset_index()
transactions_per_customer_per_countries["Value"] = transactions_per_customer_per_countries["Description"]/transactions_per_customer_per_countries["CustomerID"]/12

transactions_per_customer_per_countries.Description.sum()/transactions_per_customer_per_countries.CustomerID.sum()/12
```

[Output]: 9.851611908067254

- Những quốc gia đang hoạt động tích cực nhất:

[Input]:

```
general_trend_country = pd.DataFrame(data={'Date':pd.to_datetime(df.InvoiceDate),'Country':df.Country,'Total price':df.Quantity*df.UnitPrice})
general_trend_country = general_trend_country.groupby([general_trend_country.Date.dt.to_period("M"),general_trend_country.Country]).sum()
general_trend_country = pd.DataFrame(general_trend_country).reset_index()
general_trend_country.groupby([general_trend_country.Country]).sum().sort_values(by='Total price',ascending=False).head(10)
```

[Output]:

Country	Total price
United Kingdom	5819071.584
Germany	180531.500
EIRE	166001.240
France	163757.650
Netherlands	69053.040
Switzerland	46309.070
Spain	41678.440
Belgium	38631.410
Australia	32127.520
Norway	26355.570

[Input]:

```
general_trend_country.Date = general_trend_country.Date.dt.to_timestamp()
```

– Có bao nhiêu khách hàng mới mỗi tháng?

[Input]:

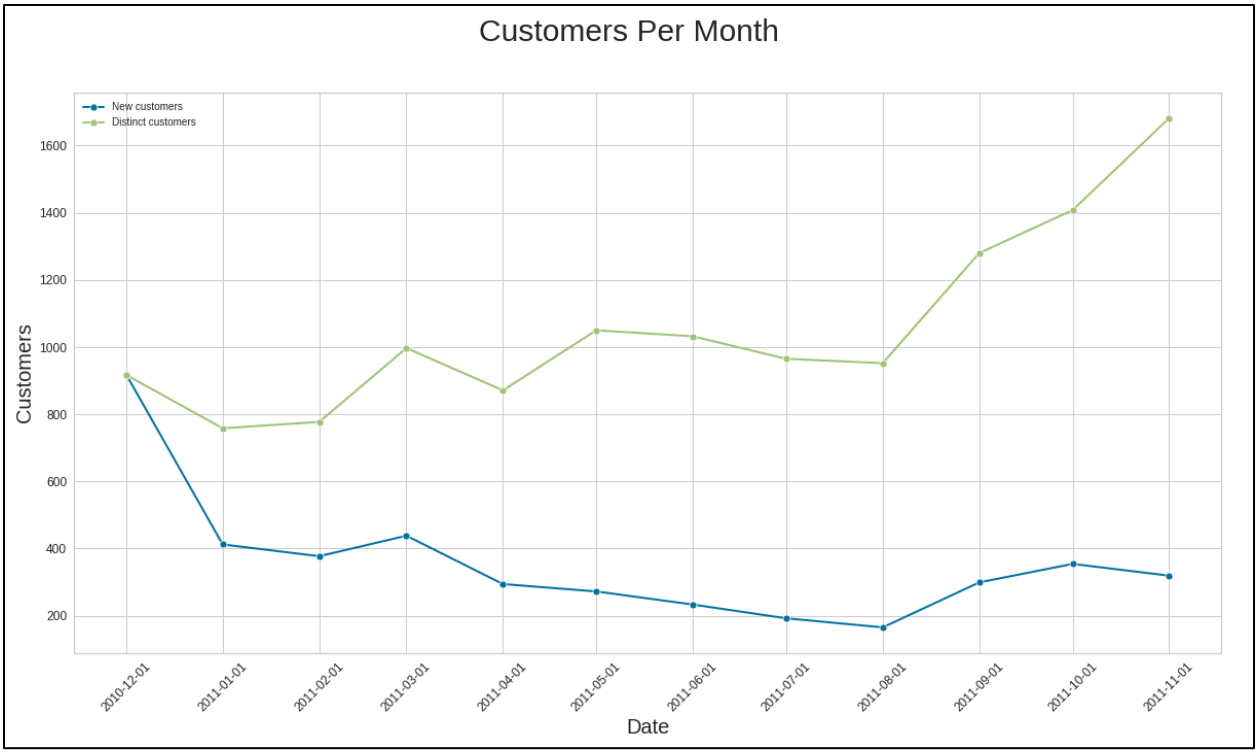
```
number_customers = df.groupby(df["InvoiceDate"].dt.to_period('M'))["CustomerID"].nunique()
number_customers = pd.DataFrame(data=number_customers).reset_index(
number_customers["Date"] = number_customers.InvoiceDate.dt.to_timestamp()
number_new_customers = []
customers_seen = []
for month in df["InvoiceDate"].dt.to_period('M').unique():
customers = df[df["InvoiceDate"].dt.to_period('M') == month].CustomerID.unique()
count=0
```

```
for customer in customers:
if customer not in customers_seen:
        count+=1
        customers_seen.append(customer)

number_new_customers.append((month,count))

number_new_customers = pd.DataFrame(number_new_customers,columns=["Dat
e","New customers"])

number_new_customers.Date = number_new_customers.Date.dt.to_timestamp()
```

[Input]:

```
fig  = plt.figure(figsize = (20,10)).suptitle('Customers Per Month', fontsize=30)
lineplot = sns.lineplot(data=number_new_customers,x="Date", y="New customers
", marker="o",linewidth = 2, label="New customers")
sns.lineplot(data=number_customers,x="Date", y="CustomerID", marker="o",line
width = 2, label="Distinct customers")
lineplot.set_xlabel("Date",fontsize=20)
lineplot.set_ylabel("Customers",fontsize=20)
plt.xticks(dates,rotation = 45,fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

[Output]:



– Tìm hiểu khách hàng có xu hướng mua các mặt hàng vào thời điểm cụ thể hơn trong ngày:

[Input]:

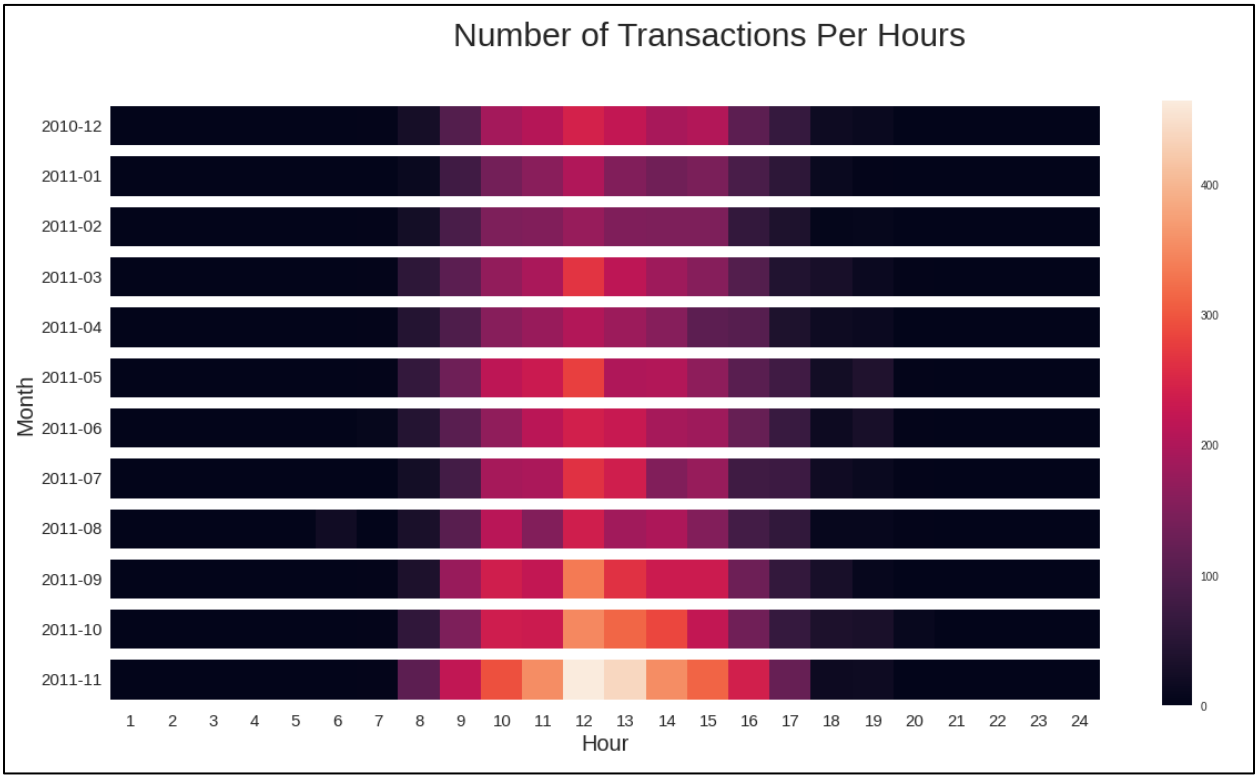
```
df_temp = df.groupby([df.CustomerID,df.InvoiceDate]).Quantity.sum()
df_temp = pd.DataFrame(df_temp).reset_index()

df_temp["Hour"]= df_temp["InvoiceDate"].dt.hour
df_temp["Month"]= df_temp["InvoiceDate"].dt.to_period('M')

count_hours = pd.DataFrame(columns=range(1,25))
count_hours["Month"] = ""

for month in sorted(df_temp["Month"].unique()):row = []
for hour in range(1,25):freq = len(df_temp[(df_temp["Month"]==month)&(df_tem
p["Hour"]==hour)])row.append(float(freq))row.append(month)
```

```
count_hours.loc[len(count_hours)] = row
count_hours = count_hours.set_index("Month")
```

[Input]:

```
fig  = plt.figure(figsize = (20,10)).suptitle('Number of Transactions Per Hours', fon
tsize=30)
heatmap = sns.heatmap(data=count_hours)
plt.yticks(rotation=0,size=15)
plt.xticks(size=15)
heatmap.set_xlabel("Hour",fontsize=20)
heatmap.set_ylabel("Month",fontsize=20)
for i in range(count_hours.shape[1] + 1):plt.axhline(i, color='white', lw=10)
plt.show()
```

[Output]:



– Để mô tả thành các buổi trong ngày, nhóm đã sử dụng Từ điển Britannica:

[Input]:

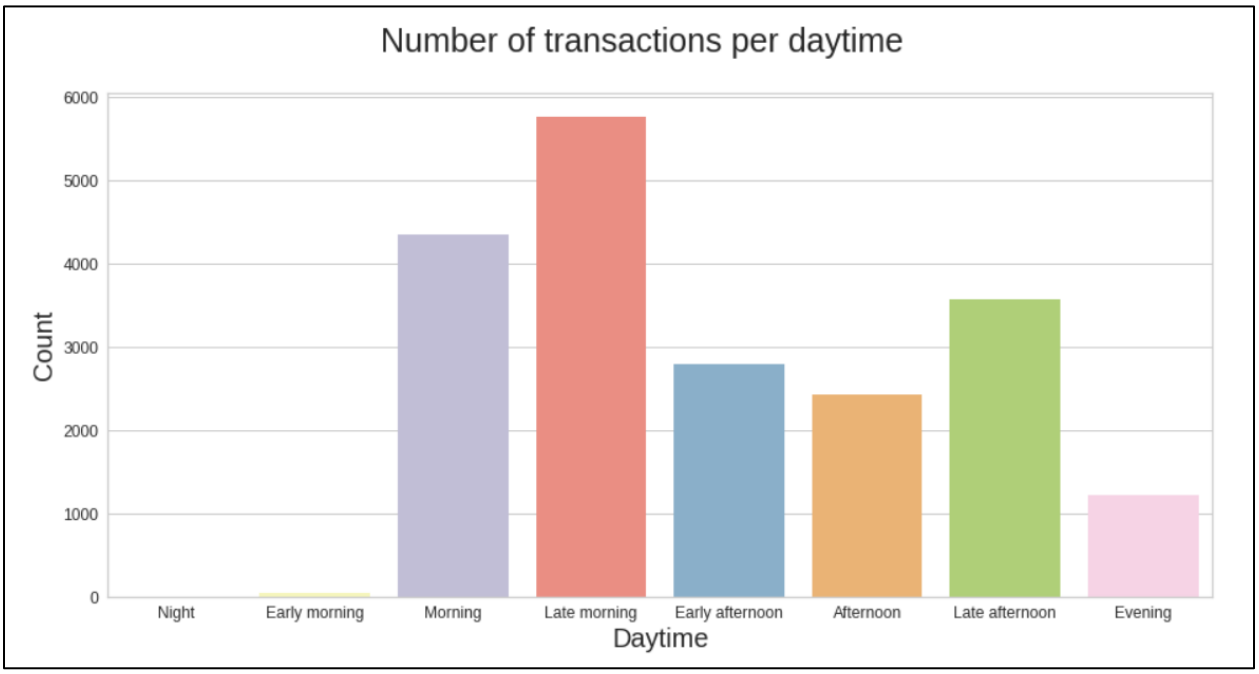
```
def daytime_encoder(date):

if (date.hour >= 5)&(date.hour < 8):
return "Early morning"
elif (date.hour >= 8)&(date.hour < 11):
return "Morning"
elif (date.hour >= 11)&(date.hour < 13):
return "Late morning"
elif (date.hour >= 13)&(date.hour < 14):
return "Early afternoon"
elif (date.hour >= 14)&(date.hour < 15):
return "Afternoon"
elif (date.hour >= 15)&(date.hour < 17):
return "Late afternoon"
elif (date.hour >= 17)&(date.hour < 21):
return "Evening"
else:
return date.hour


df_temp['InvoiceDate'] = df_temp['InvoiceDate'].map(daytime_encoder)
```

[Input]:

```
fig  = plt.figure(figsize = (15,7)).suptitle('Number of transaction
per daytime', fontsize=25)

countplot = sns.countplot(data=df_temp,x="InvoiceDate",order = ["Night","Early
morning","Morning","Late morning","Early afternoon","Afternoon","Late afterno
on","Evening"], palette="Set3")
```

```
countplot.set_xlabel("Daytime",fontsize=20)
countplot.set_ylabel("Count",fontsize=20)
plt.xticks(size=12)
plt.yticks(size=12)
plt.show()
```

[Output]:



– Có vẻ như hầu hết các giao dịch được thực hiện vào cuối buổi sáng từ 11 giờ sáng đến 12 giờ sáng. Vào sáng sớm, từ 5 giờ sáng đến 8 giờ sáng không có nhiều giao dịch được thực hiện. Và vào ban đêm, không có bất kỳ giao dịch nào.

4. Phân cụm K-Means áp dụng vào phân khúc khách hàng:

[Input]:

```
today = pd.to_datetime('today').normalize()
df_clustering = df.groupby('CustomerID').agg({'InvoiceDate': lambda InvoiceDate
: (today - InvoiceDate.max()).days, 'InvoiceNo'    : 'nunique', 'TotalPrice' : 'sum'})
```

df_clustering.columns = ['recency', 'frequency', 'monetary']

df_clustering

[Output]:

CustomerID	recency	frequency	monetary
12347.0	4016	6	3835.58
12348.0	4052	4	1407.24
12349.0	3995	1	1457.55
12350.0	4287	1	334.40
12352.0	4013	8	1545.41
...	...	...	...
18280.0	4254	1	180.60
18281.0	4157	1	80.82
18282.0	4099	2	98.76
18283.0	3986	15	1886.88
18287.0	4019	3	1837.28

4261 rows × 3 columns

[Input]:

std_scaler = StandardScaler()

df_scaled = std_scaler.fit_transform(df_clustering)

df_scaled = pd.DataFrame(df_scaled,columns=['recency', 'frequency', 'monetary'])

df_scaled["CustomerID"] = df_clustering.index

df_scaled = df_scaled.set_index("CustomerID",drop=True)

- Để xác định số lượng phân cụm tối ưu (số K), nhóm sẽ sử dụng phương pháp elbow (phương pháp khuỷu tay):

[Input]:

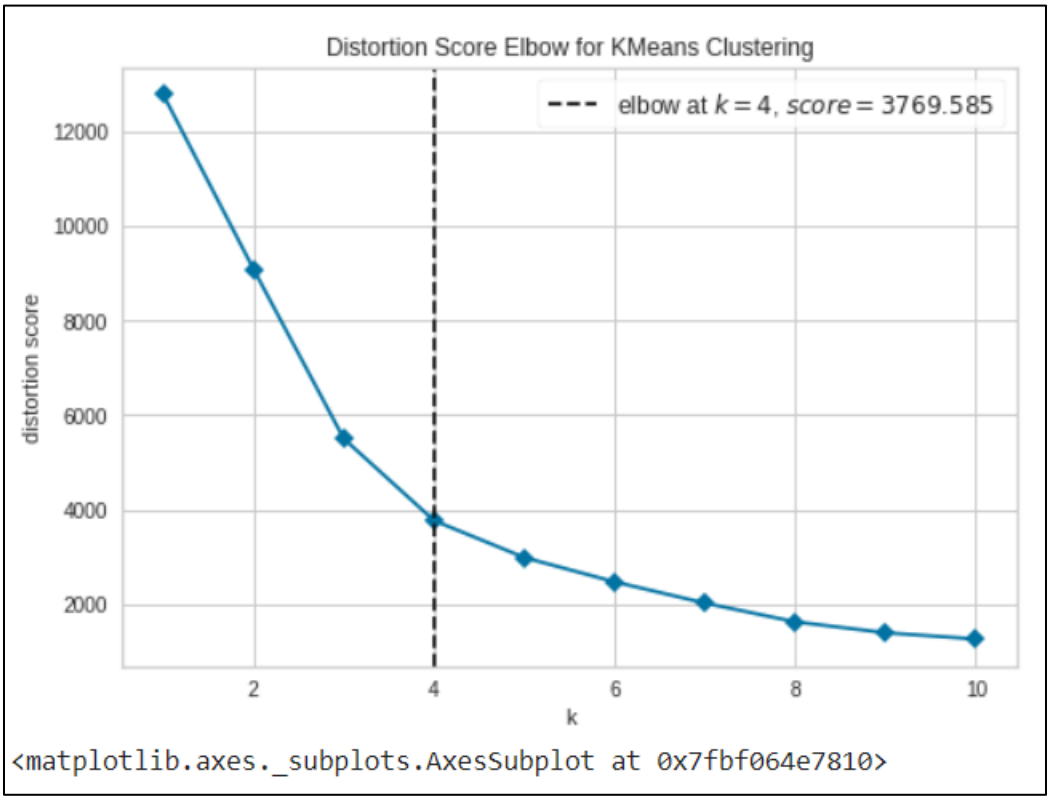
model = KMeans()

visualizer = KElbowVisualizer(model, k=(1,11), timings= False)

visualizer.fit(df_scaled)

visualizer.show()

[Output]:



[Input]:

```
kmeans = KMeans(n_clusters=4, n_init = 15, random_state=1)
kmeans.fit(df_scaled)
centroids = kmeans.cluster_centers_
centroid_df = pd.DataFrame(centroids, columns = list(df_scaled) )
centroid_df
```

[Output]:

	recency	frequency	monetary
0	-0.491976	-0.110493	-0.100629
1	1.521696	-0.344437	-0.300221
2	-0.870190	11.773733	13.732765
3	-0.774253	1.831160	1.511688

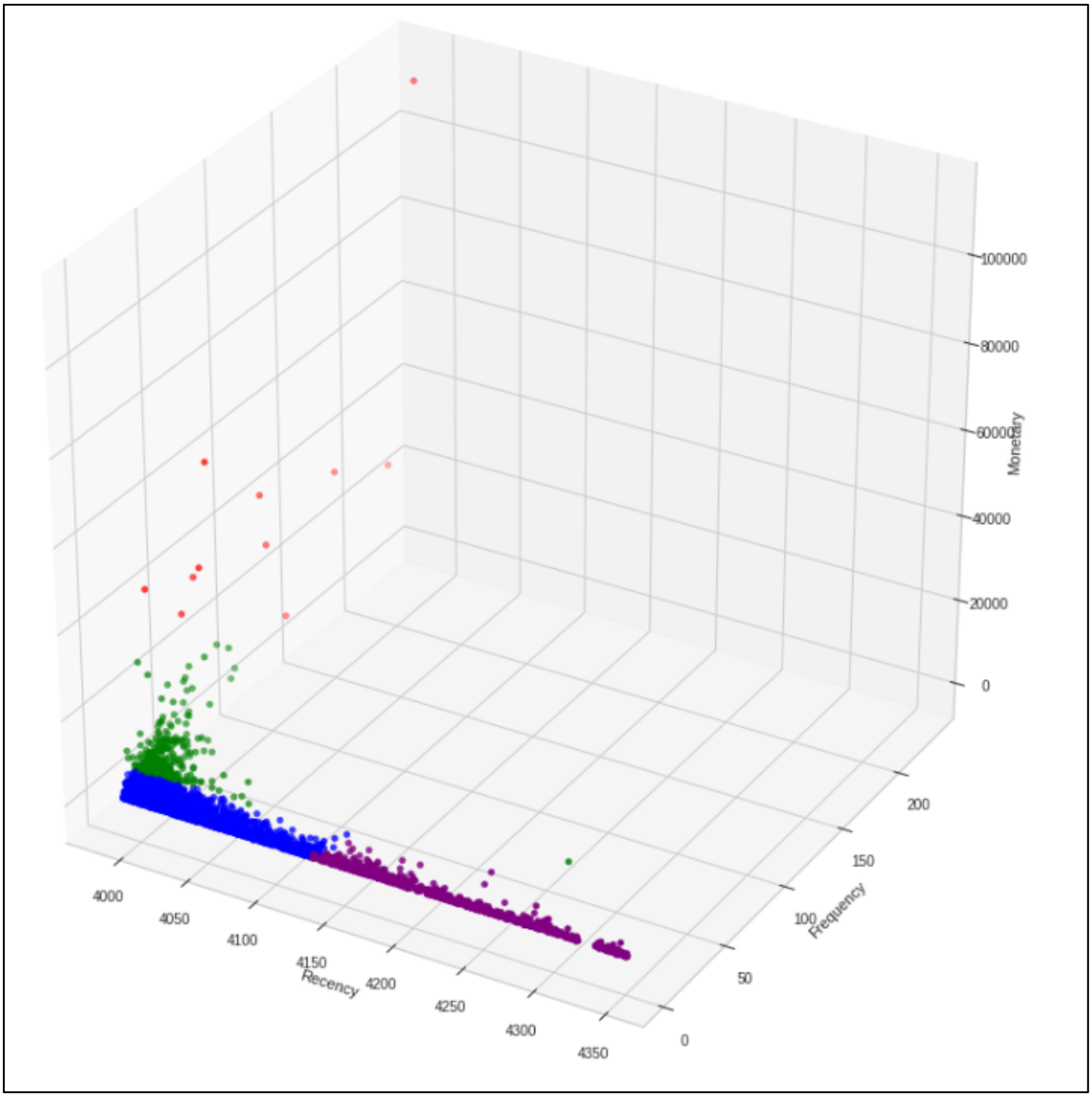
[Input]:

```
df_labels = pd.DataFrame(kmeans.labels_ , columns = list(['labels']))
df_labels['labels'] = df_labels['labels'].astype('category')
df_kmeans = df_clustering.copy()
df_kmeans['labels'] = df_labels['labels'].values
```

[Input]:

```
colors = np.array(["blue", "purple","red","green"])
fig  = plt.figure(figsize = (15,15)).suptitle('Plot of Customer\'s Distribution', fontsize=25)
ax = plt.axes(projection='3d')
ax.scatter3D(df_kmeans["recency"], df_kmeans["frequency"], df_kmeans["monetary"], marker='o', c=colors[df_kmeans["labels"].tolist()])
ax.set_xlabel('Recency')
ax.set_ylabel('Frequency')
ax.set_zlabel('Monetary')
plt.legend()
plt.show()
```

[Output]: Plot of Customer's Distribution:



[Input]:

```
agg_list=["mean", "count", "max",]
df_kmeans[["labels", "recency", "frequency", "monetary"]].groupby("labels").agg(
agg_list)
```

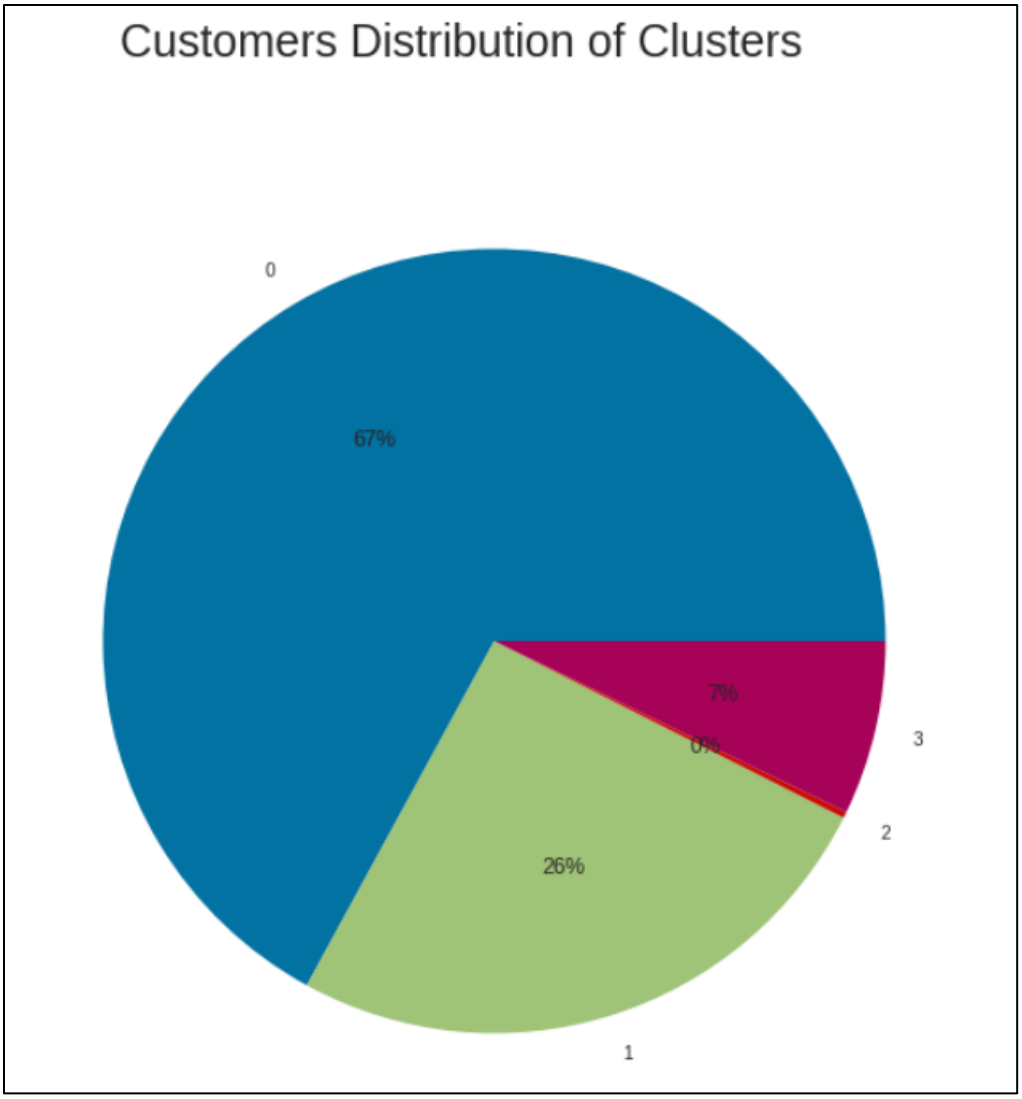
[Output]:

	recency			frequency			monetary		
	mean	count	max	mean	count	max	mean	count	max
labels									
0	4027.256913	2857	4140	3.805040	2857	16	1009.156473	2857	5979.05
1	4226.709292	1087	4350	1.787489	1087	13	380.242246	1087	5365.27
2	3989.818182	11	4007	106.272727	11	233	44581.068182	11	112382.34
3	3999.323529	306	4279	20.545752	306	85	6087.373562	306	28863.91

[Input]:

```
df_kmeans = df_kmeans.reset_index()
clusters_count = df_kmeans.groupby("labels").agg({"CustomerID": "count"})
clusters_count.reset_index(inplace=True)
clusters_count.columns = ['cluster', 'count']

fig  = plt.figure(figsize = (20,10)).suptitle('Customers Distribution of Clusters', fontsize=25)
plt.pie(clusters_count["count"], labels = clusters_count["cluster"], autopct='%.0f%%')
plt.show()
```

[Output]:



- Từ phân cụm KMeans, nhóm em có thể sắp xếp khách hàng thành 4 nhóm khác nhau theo các hành vi khác nhau.
- Phân cụm 0: "Punctual customers" - những khách hàng mua các mặt hàng đúng giờ trên trang web.
- Phân cụm 1: "Hibernating customers" - những khách hàng mua với tần suất thấp nhất, gần đây nhất và chi tiêu ít tiền nhất.
- Phân cụm 2: "Exceptional customers" - những khách hàng mà công ty này muốn giữ chân, mua hàng với tần suất cao nhất, gần đây nhất và chi tiêu nhiều tiền nhất.

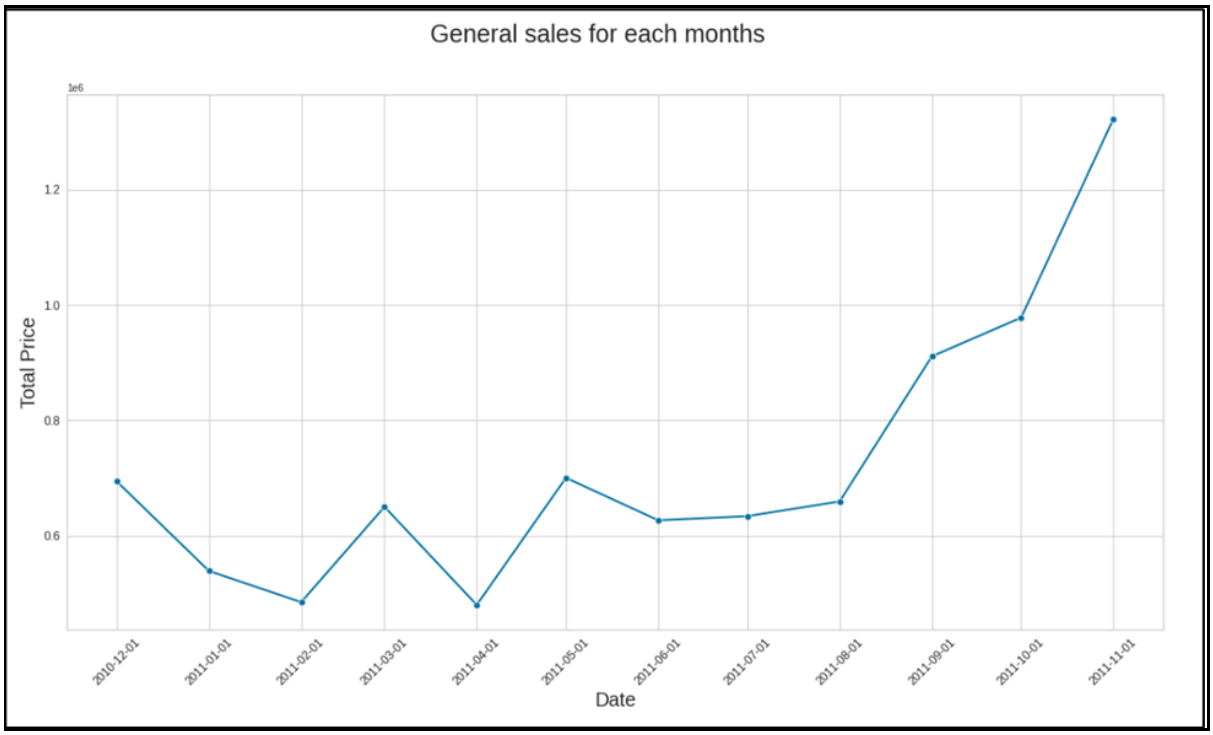
– Phân cụm 3: "Recent customers" - những khách hàng hoạt động khá tích cực gần đây có thể gây hứng thú để tiếp tục được kích thích. Tổng điểm biến dạng thu được là 4129 bằng cách sử dụng lần truy cập gần đây.

IV. Kết quả và đánh giá:

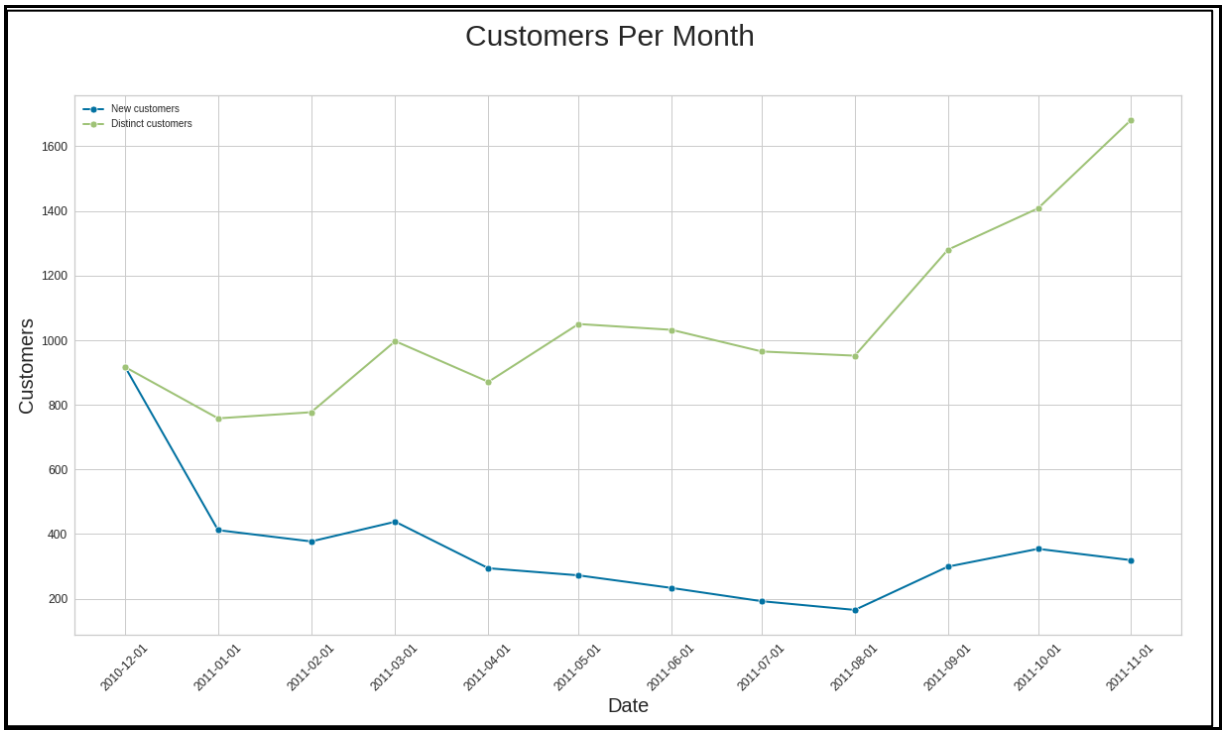
1. Kết quả:

Sau quá trình phân tích, nhóm đã tìm ra được một số insights sau từ tập dữ liệu:

a. Lượt bán hàng tăng mạnh vào tháng 8/2011, sau đó tăng cao từ tháng 8 đến tháng 11:



b. Phân tích về số lượng khách hàng trong từng tháng:



Có thể thấy, cũng từ tháng 8/2021, số lượng khách hàng tăng lên đáng kể, chủ yếu là những khách hàng mới.

c. Doanh thu và lượt bán nhiều nhất chủ yếu là tại thị trường nước Anh. Tuy nhiên, một số quốc gia khác như Pháp, Đức, Tây Ban Nha, Australia và Hà Lan cũng khá năng động. Điều này cho thấy, các doanh nghiệp nên tập trung nhiều hơn ở các thị trường này, đồng thời đẩy mạnh sản xuất để có thể tăng thêm nhiều hơn về doanh thu.

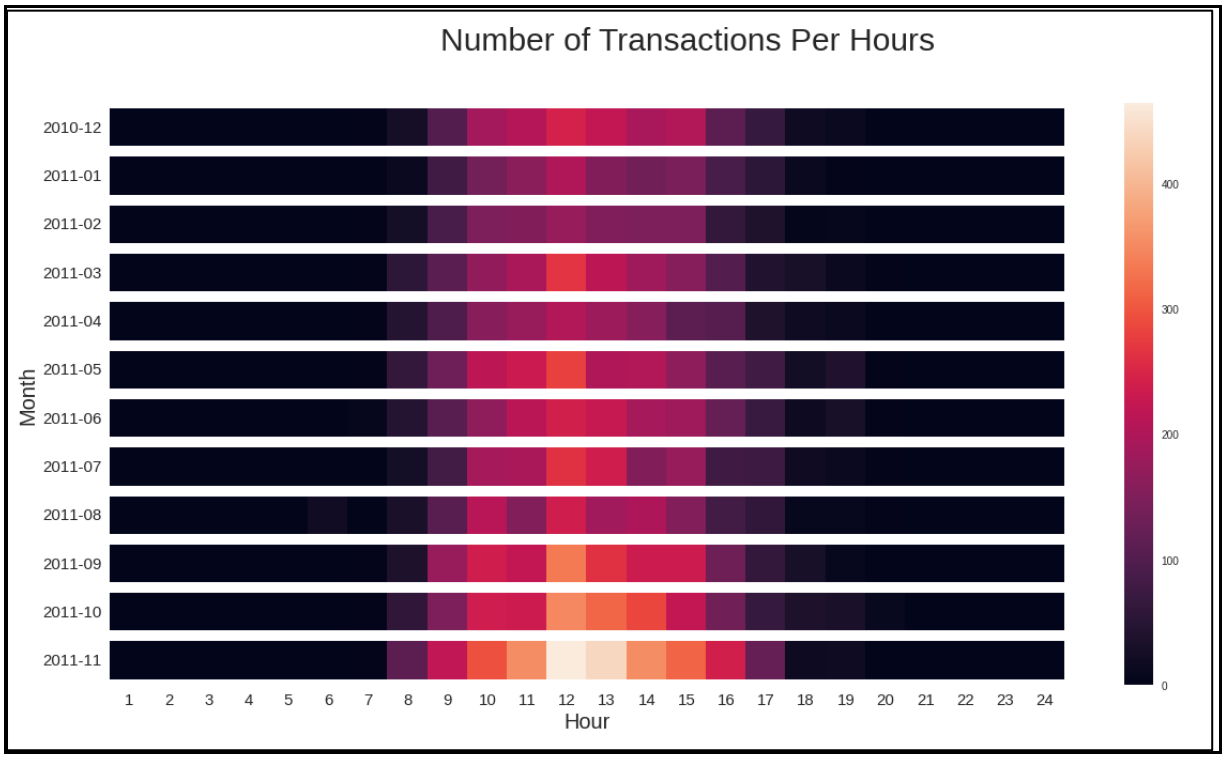
Country	Total price
United Kingdom	5819071.584
Germany	180531.500
EIRE	166001.240
France	163757.650
Netherlands	69053.040
Switzerland	46309.070
Spain	41678.440
Belgium	38631.410
Australia	32127.520
Norway	26355.570

d. Có sự khác biệt trong các mặt hàng bán chạy nhất dựa trên quốc gia. Mặt hàng bán chạy nhất là "white hanging heart T-light holder".

```
WHITE HANGING HEART T-LIGHT HOLDER    2284
REGENCY CAKESTAND 3 TIER              2102
JUMBO BAG RED RETROSPOT               2090
PARTY BUNTING                         1673
LUNCH BAG RED RETROSPOT               1595
ASSORTED COLOUR BIRD ORNAMENT         1451
SET OF 3 CAKE TINS PANTRY DESIGN      1422
SMALL POPCORN HOLDER                  1396
PACK OF 72 RETROSPOT CAKE CASES       1348
LUNCH BAG SUKI  DESIGN                1314
Name: Description, dtype: int64
```

Từ việc phân tích những mặt hàng nào bán chạy hơn ở các quốc gia nào, có thể thấy được sở thích, xu hướng và các mặt hàng được ưa chuộng. Doanh nghiệp có thể đẩy mạnh việc sản xuất và kinh doanh mặt hàng đó tại các thị trường tương ứng.

e. Thời điểm hoạt động tích cực nhất trong ngày là vào khoảng buổi trưa, hầu hết khách hàng mua hàng vào khoảng thời gian này trong ngày.



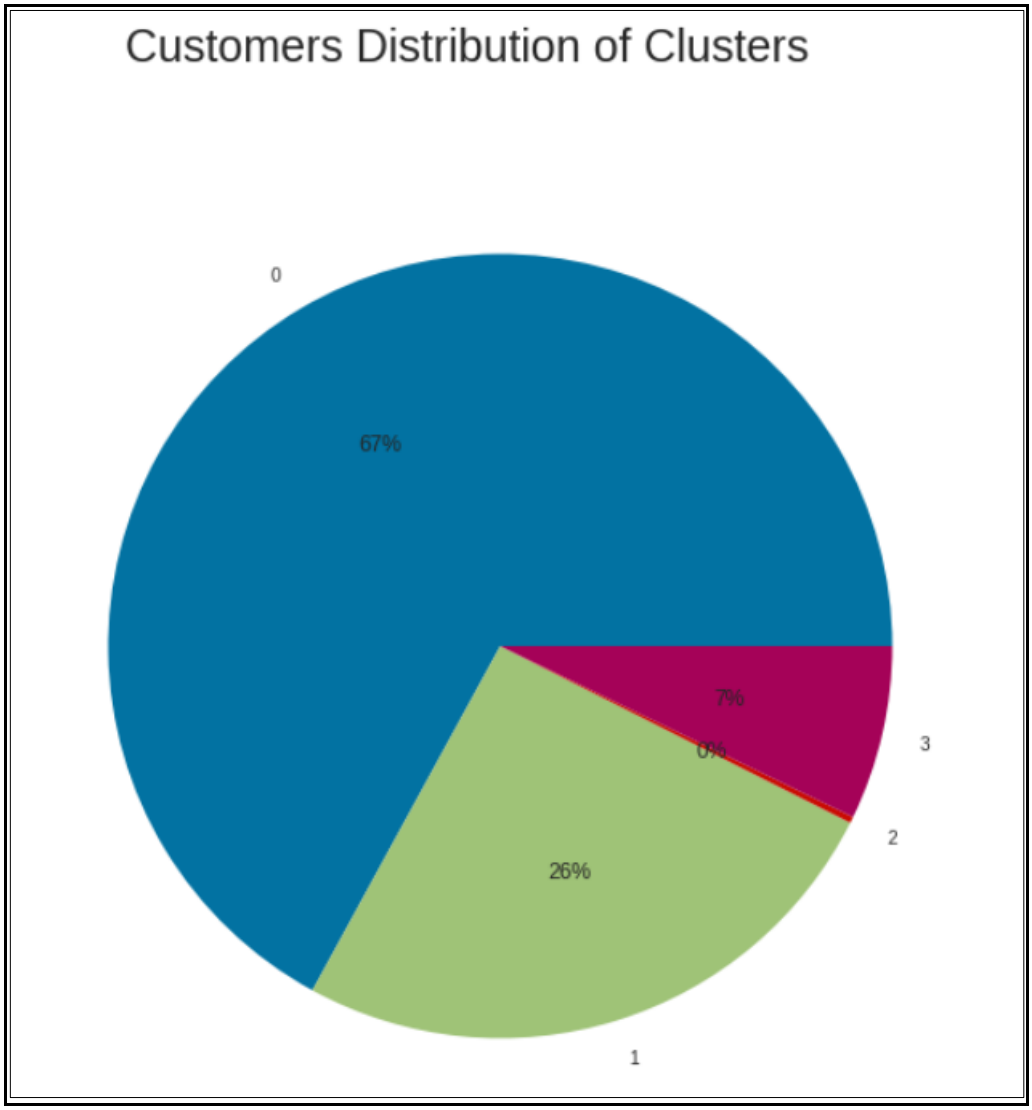
Qua kết quả phân tích này, doanh nghiệp nên đẩy mạnh thêm các hoạt động marketing, truyền thông mạnh mẽ hơn vào các thời gian này và mở thêm những chương trình ưu đãi, khuyến mãi vào các thời gian hợp lý để có thể tối ưu hóa lợi nhuận.

f. Về việc phân khúc khách hàng, sau khi thực nghiệm với thuật toán K-Means, nhóm đã sắp xếp khách hàng thành 4 nhóm khác nhau theo các hành vi khác nhau.

- Phân cụm 0: "Punctual customers" - những khách hàng mua các mặt hàng đúng giờ của doanh nghiệp.
- Phân cụm 1: "Hibernating customers" - những khách hàng mua với tần suất thấp nhất, gần đây nhất và chi tiêu ít tiền nhất.
- Phân cụm 2: "Exceptional customers" - những khách hàng mà công ty này muốn giữ chân, mua hàng với tần suất cao nhất, gần đây nhất và chi tiêu nhiều tiền nhất.

– Phân cụm 3: "Recent customers" - những khách hàng hoạt động khá tích cực gần đây có thể gây hứng thú để tiếp tục kích thích nhu cầu mua hàng của nhóm khách này.

Qua kết quả phân khúc khách hàng, chúng ta thấy được nhóm 0 chiếm số lượng lớn nhất, là những khách hàng mua sản phẩm đúng giờ, và thời gian họ hay mua nhất trong ngày vào các khoảng thời gian buổi trưa. Từ đó, doanh nghiệp có thể đẩy mạnh chiến lược Marketing vào các thời gian từ 11 - 13h trưa phù hợp với nhóm khách hàng này để có thể tiếp cận được nhiều khách hàng hơn, họ cũng sẽ có xu hướng mua hàng nhiều hơn, từ đó, tăng doanh thu tối đa cho doanh nghiệp.



2. Đánh giá mô hình:

a. Ưu điểm:

– Thuật toán phân cụm có độ chính xác cao, ngoại lệ chỉ chiếm 4.129 trường hợp, tương đương 0,76%.
– Với số lượng gần 550.000 dòng dữ liệu, thuật toán có thể xử lý mà không cần xác định trước đầu ra (nhãn)
– Có khả năng học tập không cần giám sát.

b. Nhược điểm:

– Tuy số lượng ngoại lệ thấp nhưng độ lệch lớn (cụm bị phân tán) nên gây ảnh hưởng đến độ chính xác của thuật toán. Chủ yếu là do Phân cụm 1: "Hibernating customers"
– Khi số cụm càng nhỏ so với số bộ dữ liệu, thuật toán càng dễ đi đến kết quả chưa phải tối ưu. Điều này phụ thuộc vào cách chọn K trung tâm cụm ban đầu. Để khắc phục điều này, ta cần lặp lại thuật toán nhiều lần và chọn phương án có giá trị hàm mất mát nhỏ nhất[9].
– Trong thuật toán K-Means, chúng ta cần biết số lượng clusters (cụm). Nhưng thực tế, chúng ta khó xác định được giá trị này.

c. Cải tiến thuật toán:

– Cải tiến thuật toán K-Means: thay vì chọn số điểm (k) làm trọng tâm, chúng ta tăng số cụm từ 1 lên k cụm bằng cách đưa trung tâm cụm mới vào cụm có mức độ biến dạng lớn nhất và tính lại trọng tâm các cụm.
– Với thuật toán K-Means bắt đầu bằng cách chọn K cụm và chọn ngẫu nhiên K điểm làm trung tâm cụm, hoặc chọn phân hoạch ngẫu nhiên K cụm và tính trọng tâm của từng cụm này. Việc chọn ngẫu nhiên K điểm làm trung tâm cụm như đã nói ở trên có thể cho ra các kết quả khác nhau tùy vào số K được chọn này[10].

PHẦN 4: TỔNG KẾT

I. Ưu - nhược điểm của thuật toán:

1. Ưu điểm:

– Đơn giản, dễ hiểu, dễ thực hiện, K-Means phù hợp với các tập dữ liệu lớn

– Đặc biệt, tập dữ liệu không nhất thiết phải có các labels, K-Means vẫn có thể phân cụm, ứng dụng tốt với những cụm siêu hình, cụm hình cầu.
– Những kết quả sau khi phân cụm bằng K-Means rất rõ ràng và dễ hiểu, tạo ra các mô tả về các cụm đơn giản và dễ hiểu được tập dữ liệu.

2. Nhược điểm:

– K-Means chỉ thực hiện được với dữ liệu dạng số
– Cần xác định trước đại lượng K và số lượng cluster. Tuy nhiên trong thực tế, nhiều trường hợp, chúng ta không thể xác định được giá trị này. Cũng rất khó để xác định K bao nhiêu là chính xác và khó để so sánh chất lượng của các cụm. Để cải thiện nhược điểm này, có thể ứng dụng phương pháp khuỷu tay (Phương pháp Elbow) hoặc sử dụng Hệ số bóng (Silhouette Coefficient).
– Thứ tự của dữ liệu có ảnh hưởng đến kết quả cuối cùng.
– Các Cluster cần có số lượng điểm gần bằng nhau: vì nếu chênh lệch quá lớn, mức độ chính xác sẽ giảm đi đáng kể.
– Bên cạnh đó, các cluster cần có dạng hình tròn, cụ thể hơn, các cluster cần tuân theo phân phối chuẩn và ma trận hiệp phương sai là ma trận đường chéo có các điểm trên đường chéo giống nhau.
– K-Means cho những kết quả khác nhau nếu cách xác định số K và cách phân cụm thực hiện khác nhau, nếu lấy một mẫu trong tập dữ liệu cũng có thể cho một kết quả khác, điều này dẫn đến sự thiếu nhất quán.
– Trong một số trường hợp, K-Means không đạt được độ chính xác: khi một cluster nằm phía trong 1 cluster khác.

Mặc dù còn tồn tại một số hạn chế, không thể phủ nhận, K-Means là một thuật toán cực kỳ quan trọng và phổ biến, đây là một nền tảng cho nhiều thuật toán phức tạp khác nhau sau này.

TÀI LIỆU THAM KHẢO

[1] Trích dẫn từ nguồn: *K-Means Clustering Algorithm - Javatpoint*

[2] Tham khảo từ nguồn: K-Means Clustering - Machine Learning co ban

[3] Tham khảo từ nguồn: Integration K-Means Clustering Method and Elbow Method

[4] Trích dẫn từ nguồn: Các phương pháp đánh giá trong thuật toán Clustering

[5] Tham khảo từ nguồn: KMeans Silhouette Score With Python Examples - DZone AI

[6] Tham khảo từ nguồn: Tài liệu Machine Learing for Vietnameses

[7] Tham khảo từ nguồn: K-Means clustering - Deep AI Khanh Blog

[8] Tham khảo từ nguồn: Analyzing customer sentiments using K-Means algorithm

[9] Tham khảo từ nguồn: Thuật Toán K-Means Clustering - Dat Hoang's Blog

[10] Tham khảo từ nguồn: Cải tiến thuật toán K-means và ứng dụng phân cụm dữ liệu tự động