**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙟🕮🙝🙢



**MÔN HỌC: KHAI PHÁ DỮ LIỆU**

**ĐỒ ÁN CUỐI KÌ**

**ĐỀ TÀI:**

**GVHD**: Nguyễn Văn Thành

**Nhóm sinh viên thực hiện:** Nhóm 3

1. Nguyễn Trí Dũng 20133029
2. Nguyễn Khoa Quang Thắng 20133090
3. Văn Mai Thanh Nhật 20133076
4. Huỳnh Minh Phước 20133082

**Mã môn học**: DAMI330484\_22\_2\_01

Thành phố Hồ Chí Minh, Tháng 5 năm 2023

**Lời mở đầu**

Khai phá dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó. Nhận thấy được tầm quan trọng của môn học, nhóm 3 thực hiện đề tài “Khai phá tập dữ liệu giao dịch e-commerce của khách hàng”, với mục đích tìm được mối quan hệ với nhau giữa các khách hàng thông qua các giao dịch, đồng thời tìm định hướng, chiến lược phát triển bán hàng cho website. Vì ngày nay, thông qua e-commerce, các doanh nghiệp có thể tiếp cận và thiết lập sự hiện diện trên thị trường rộng lớn hơn bằng cách cung cấp các kênh phân phối rẻ hơn và hiệu quả hơn cho các sản phẩm hoặc dịch vụ của họ. E-commerce cũng đã thay đổi cách mọi người mua sắm và tiêu thụ sản phẩm và dịch vụ. Nhiều khách hàng đang chuyển sang máy tính hoặc thiết bị thông minh của họ để đặt hàng, những thứ có thể dễ dàng được giao đến tận nhà của họ.

**Mục lục**

# PHẦN 1 – GIỚI THIỆU:

Nhóm sẽ sử dụng vẽ biểu đồ và đưa ra nhận xét, thêm vào đó,thuật toán gom cụm, phương pháp số học, và thuật toán luật kết hợp sẽ được sử dụng để thực hiện đề tài.

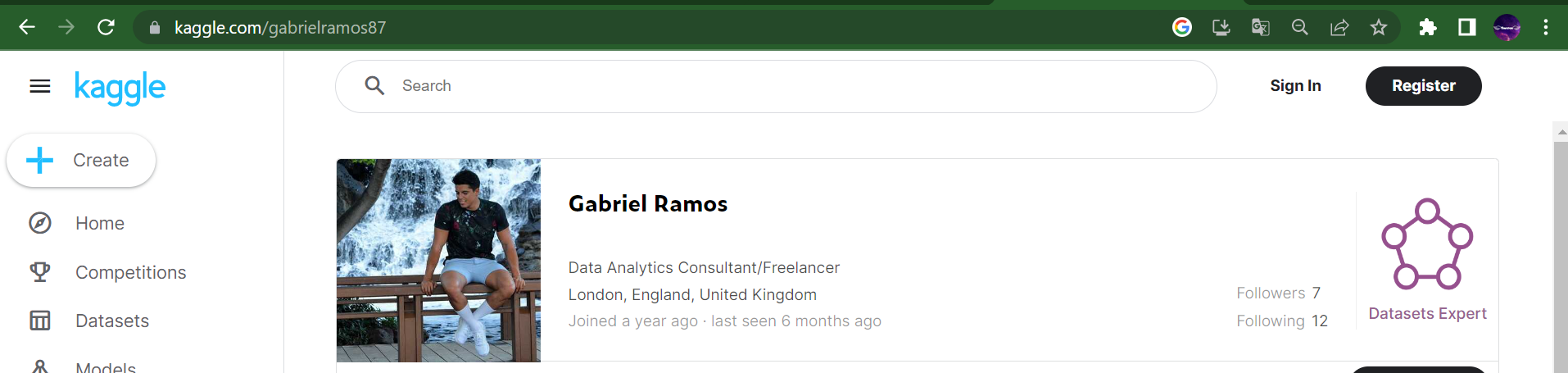
Các công cụ được dùng trong đề tài:

* Ngôn ngữ Python: Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.
* Visual studio code: Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn.
* Jupyter là một nền tảng tính toán khoa học mã nguồn mở, với khả năng nổi bật cho phép tương tác trực tiếp với từng dòng code (interactive), hỗ trợ hơn 40 ngôn ngữ lập trình, trong đó tập trung vào 3 ngôn ngữ là Julia, Python và R, cái tên Jupyter bắt nguồn từ cách chơi chữ I Python, You R, We Julia, hay một công thức ngắn gọn Jupyter = Julia + Python + R. Bên cạnh đó, Jupyter cũng là một công cụ hoàn toàn miễn phí, được tạo ra với mục đích nhắm đến khoa học dữ liệu và giáo dục, giúp mọi người cùng học lập trình dễ dàng hơn (cụ thể ở đây là Python). Jupyter có tính tương tác nên có thể sử dụng làm môi trường chạy thử và giảng dạy.

# PHẦN 2 – DỮ LIỆU

### 3.1) Nguồn gốc của tập dữ liệu:

Trong đề tài này, dataset được nhóm 3 thống nhất sử dụng được lấy từ Kaggle và tác giả của tập dữ liệu là Gabriel Ramos.



Theo tác giả, đây là bộ dữ liệu giao dịch bán hàng của thương mại điện tử (bán lẻ trực tuyến) có trụ sở tại Vương quốc Anh trong một năm. Cửa hàng có trụ sở tại Luân Đôn này đã bán quà tặng và đồ gia dụng cho người lớn và trẻ em thông qua trang web từ năm 2007. Khách hàng của họ đến từ khắp nơi trên thế giới và thường mua hàng trực tiếp cho chính họ. Ngoài ra còn có các doanh nghiệp nhỏ mua với số lượng lớn và bán cho các khách hàng khác thông qua các kênh đại lý bán lẻ.

### 3.2) Thông tin về tập dữ liệu:

**Kích thước của tập dữ liệu:** 536350 dòng x 8 cột

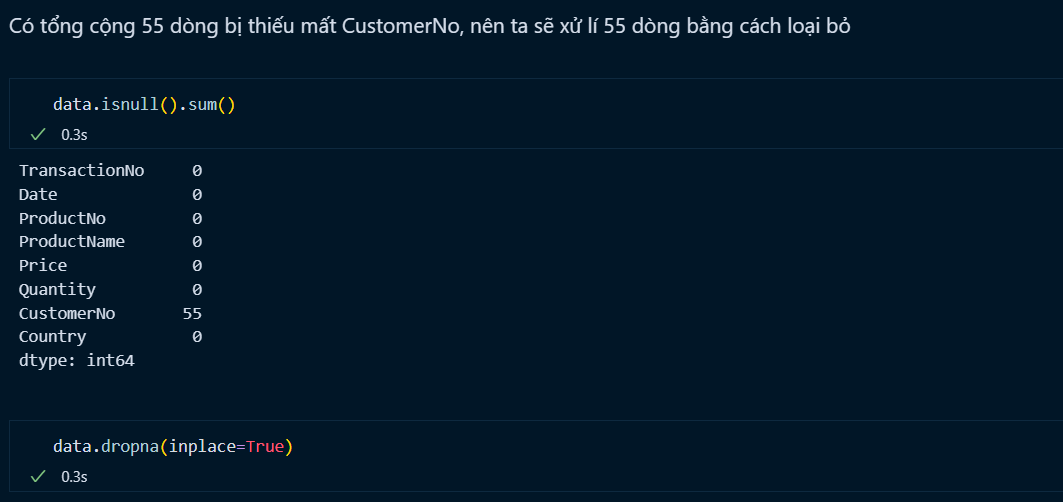
**Các biến của tập dữ liệu:**

1. ***TransactionNo (categorical):*** a six-digit unique number that defines each transaction. The letter “C” in the code indicates a cancellation.
2. ***Date (numeric):*** the date when each transaction was generated.
3. ***ProductNo (categorical):*** a five or six-digit unique character used to identify a specific product.
4. ***Product (categorical):*** product/item name.
5. Price (numeric): the price of each product per unit in pound sterling (£).
6. ***Quantity (numeric):*** the quantity of each product per transaction. Negative values related to cancelled transactions.
7. ***CustomerNo (categorical):*** a five-digit unique number that defines each customer.
8. ***Country (categorical):*** name of the country where the customer resides.

### 3.2) Tiền xử lí:

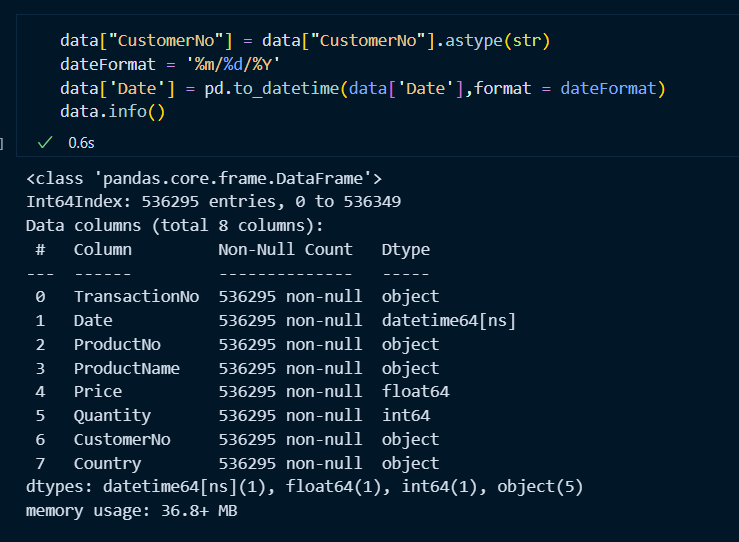
#### 3.2.1) Xử lí những giá trị bị thiếu:

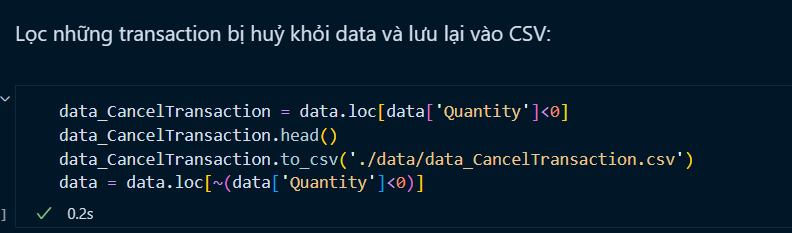
Tập dữ liệu có 55 dòng bị thiếu dữ liệu ở CustomerNo, nên do số lượng quá ít so với tập dữ liệu, nên nhóm quyết định bỏ những dòng bị thiếu dữ liệu CustomerNo.



#### 3.2.2) Chuyển đổi kiểu dữ liệu của tập dữ liệu:

Chuyển đổi dữ liệu của tập dữ liệu, CustomerNo chuyển từ int thành string và Date từ string thành datetime.



Do tập data gốc có chứa những transaction bị huỷ, nhóm sẽ loại bỏ để không ảnh hưởng tới phần tính toán.

#### 3.2.3) Xử lí outlier:

a) Giới thiệu thuật toán:

**Sử dụng phương pháp số học z-score để tìm outlier:**

Đối phương pháp này, ta cần phải giả định là dữ liệu của chúng ta tuân theo phân phối chuẩn.

Trong phân phối chuẩn, giả sử μ là kỳ vọng và σ là độ lệch chuẩn. Quy tắc 3σ cho phân phối chuẩn nói rằng:

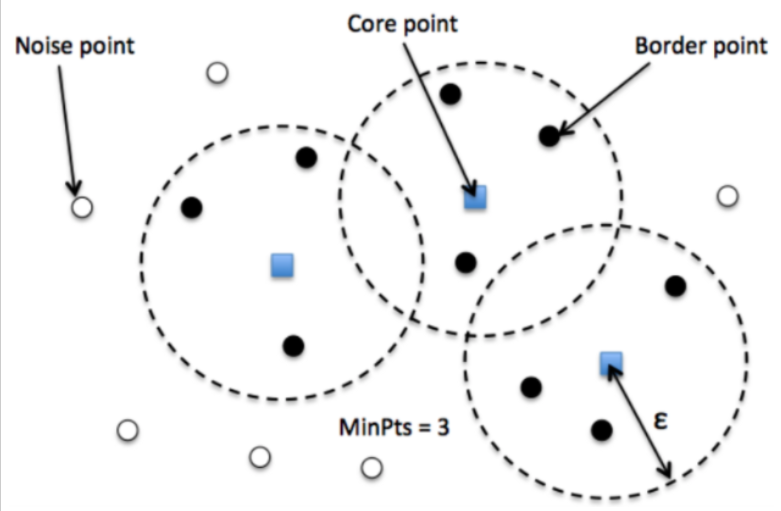
* 68% các điểm dữ liệu nằm trong khoảng μ±σ
* 95% các điểm dữ liệu nằm trong khoảng μ±2σ
* 99.7% các điểm dữ liệu nằm trong khoảng μ±3σ

Với một điểm dữ liệu x, z score của nó được tính bởi:

Vì vậy, việc các điểm nằm ngoài đoạn [μ−3σ, μ+3σ] được coi là các điểm ngoại lệ.

**Sử dụng density-based method (DBSCAN):**

Kĩ thuật này này đề cập đến các phương pháp học không giám sát nhằm xác định các cụm phân biệt trong phân phối của dữ liệu, dựa trên ý tưởng rằng một cụm trong không gian dữ liệu là một vùng có mật độ điểm cao được ngăn cách với các cụm khác bằng các vùng liền kề có mật độ điểm thấp. DBSCAN là một thuật toán cơ sở để phân nhóm dựa trên mật độ. Nó có thể phát hiện ra các cụm có hình dạng và kích thước khác nhau từ một lượng lớn dữ liệu chứa outlier.

Phân loại dạng điểm trong DBSCAN: Ta sẽ chia chúng thành ba loại: Đối với các điểm nằm sâu bên trong cụm chúng ta xem chúng là điểm lõi. Các điểm biên nằm ở phần ngoài cùng của cụm và điểm outlier không thuộc bất kì một cụm nào. Bên dưới là hình vẽ mô phỏng thể hiện ba loại điểm tương ứng nêu trên.

* ***minPts:*** Là một ngưỡng số điểm dữ liệu tối thiểu được nhóm lại với nhau nhằm xác định một vùng lân cận epsilon có mật độ cao. Số lượng minPts không bao gồm điểm ở tâm.
* ***epsilon (kí hiệu ε):*** Một giá trị khoảng cách được sử dụng để xác định vùng lân cận epsilon của bất kỳ điểm dữ liệu nào.

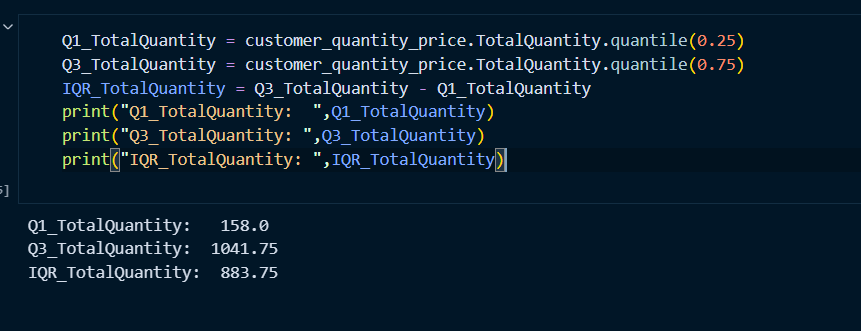
**Sử dụng Elbow method để chọn epsilon và k:**

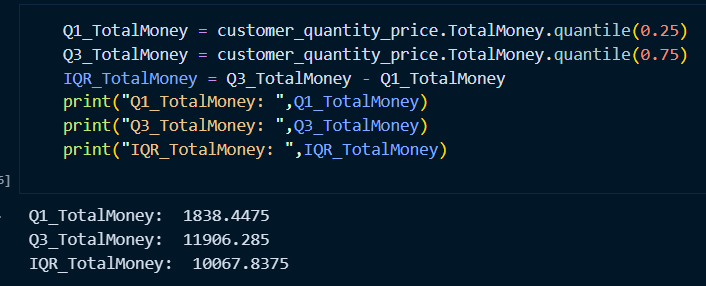
Ta sử dụng phương pháp Elbow, để tìm một giá trị phù hợp cho eps, chúng ta có thể vẽ đồ thị khoảng cách kNN của các điểm (nghĩa là khoảng cách của mỗi điểm đến điểm lân cận gần thứ k của nó) theo thứ tự giảm dần và tìm kiếm một elbow trong đồ thị. Ý tưởng đằng sau là các điểm nằm bên trong các cụm sẽ có khoảng cách k-láng giềng gần nhất nhỏ, bởi vì chúng ở gần các điểm khác trong cùng một cụm.

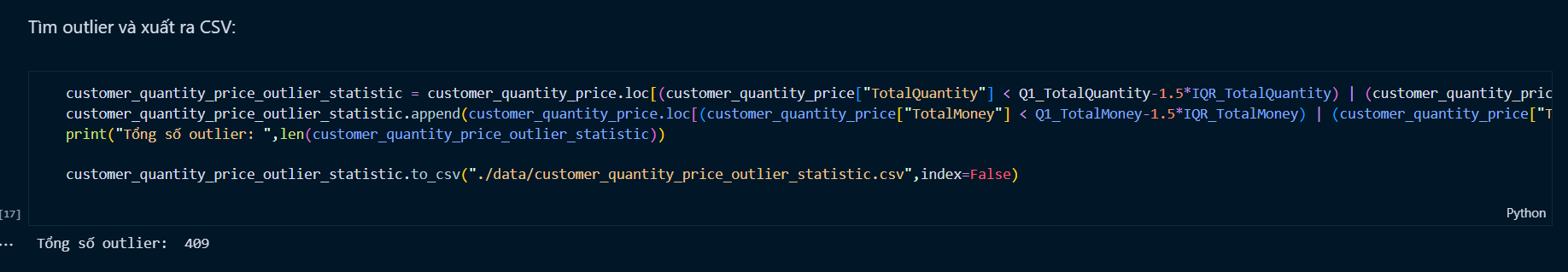
b) Thực hiện trên tập dữ liệu:

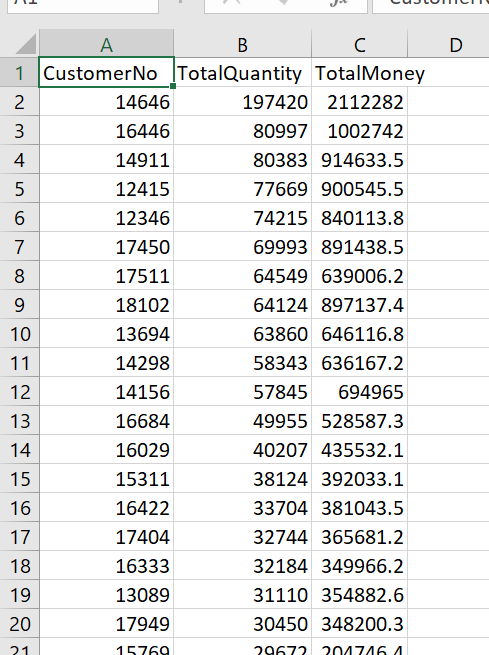
***Đối với outlier của các khách hàng:***

*Sử dụng phương pháp số học z-score để tìm outlier:*

* *Bước 1:* Tính tổng lượng hàng và tổng tiền của khách hàng đã chi tiêu:
* *Bước 2:* Tính Q1, Q3, IQR.



* *Bước 3:Lọc outlier ra khỏi tập dữ liệu và xuất ra csv:*



|  |  |
| --- | --- |
| Trước khi bỏ outlier: | Sau khi bỏ outlier: |
|  |  |

*Sử dụng density-based method (DBSCAN):* 

* Bước 1: Chọn thông số cho DBSCAN:
  + Để chọn minPts, ta có công thức minPts >= Dimension + 1, ta có minPts = 4.
  + Để chọn epsilon: Nhóm sẽ chạy elbow method với n\_neighbor = 2 và n\_neighbor = 100 để so sánh.

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

* Bước 2: Chạy thuật toán:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

Trước khi bỏ outlier:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

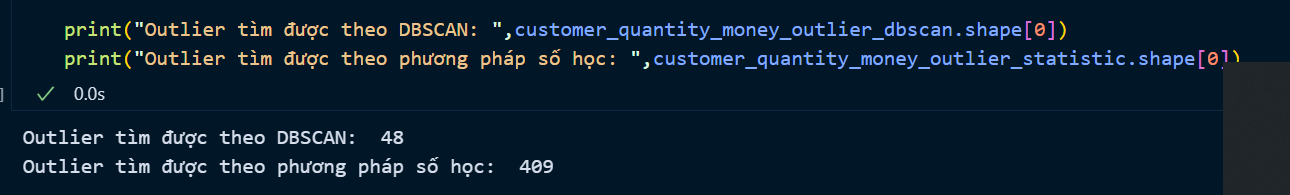
Sau khi bỏ outlier:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

Nhận xét DBSCAN với n\_neighbor có giá trị là 2 và 100:

* XYZ

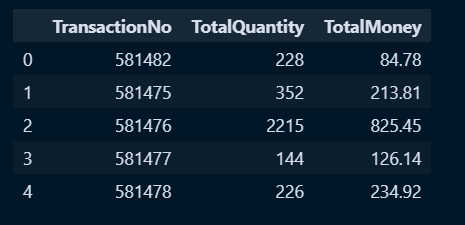
Nhận xét 2 phương pháp phát hiện outlier (DBSCAN và Statistic):

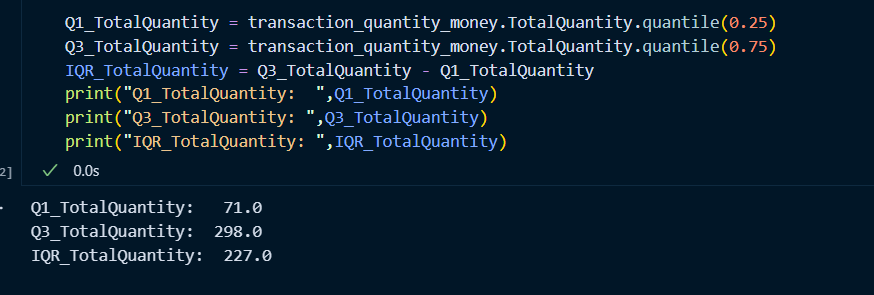
* Phương pháp DBSCAN không hiệu quả vì những giá trị tổng tiền chi và tổng lượng hàng mua không tuân theo qui luật nào mà có thể tìm ra bằng phương pháp DBSCAN với khoảng cách Euclide.
* Phương pháp statistic tìm được nhiều outlier hơn so với DBSCAN, nhưng điều này không có nghĩa là cứ tìm nhiều outlier hơn thì sẽ là tốt, nhưng đối với tập dữ liệu khách hàng này thì Statistic sẽ tốt hơn.

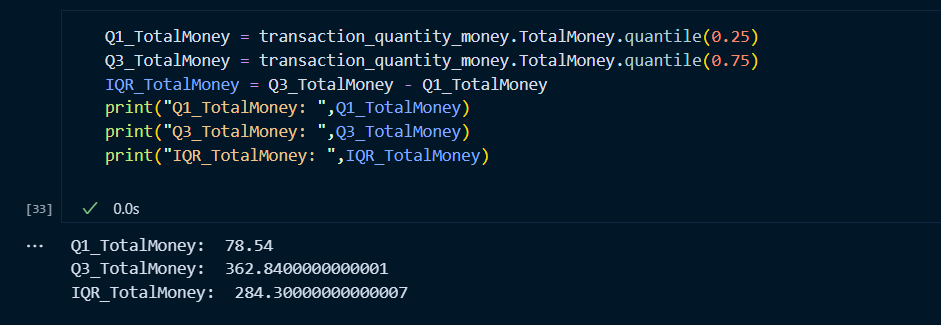
**Đối với tìm outlier của các transaction:**

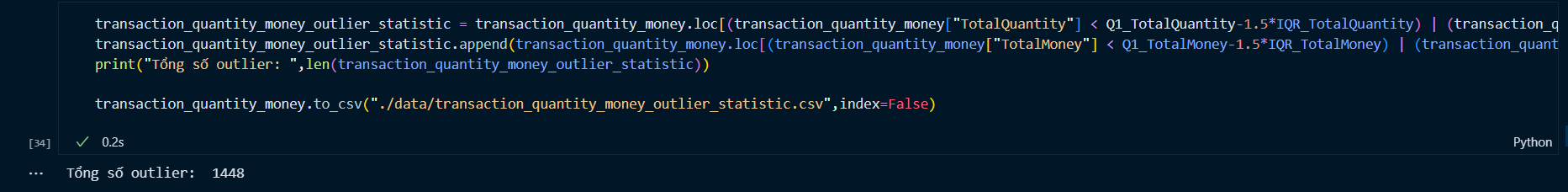
*Sử dụng phương pháp số học z-score để tìm outlier:*

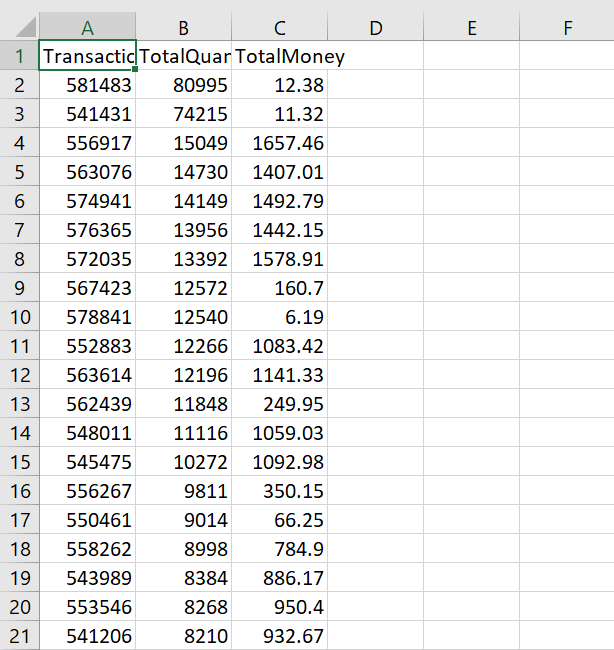
* *Bước 1:* Tính tổng lượng hàng và tổng tiền theo transaction:

**

* *Bước 2: Tính Q1, Q3 và IQR:*

****

* *****Bước 3:Lọc outlier ra khỏi tập dữ liệu và xuất ra csv:*



|  |  |
| --- | --- |
| Trước khi bỏ outlier: | Sau khi bỏ outlier: |
|  |  |

*Sử dụng density-based method (DBSCAN):*

* *Bước 1:*
  + Để chọn minPts, ta có công thức minPts >= Dimension + 1, ta có minPts = 4.
  + Để chọn epsilon: Nhóm sẽ chạy elbow method với n\_neighbor = 2 và n\_neighbor = 100 để so sánh.

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

*Bước 2:* Chạy thuật toán:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

Trước khi bỏ outlier:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

Sau khi bỏ outlier:

|  |  |
| --- | --- |
| n\_neighbor = 2 | n\_neighbor = 100 |
|  |  |

Nhận xét DBSCAN với n\_neighbor có giá trị là 2 và 100:

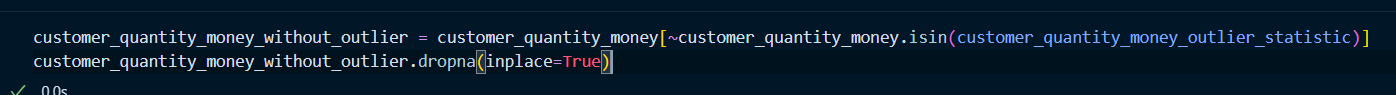
* XYZ

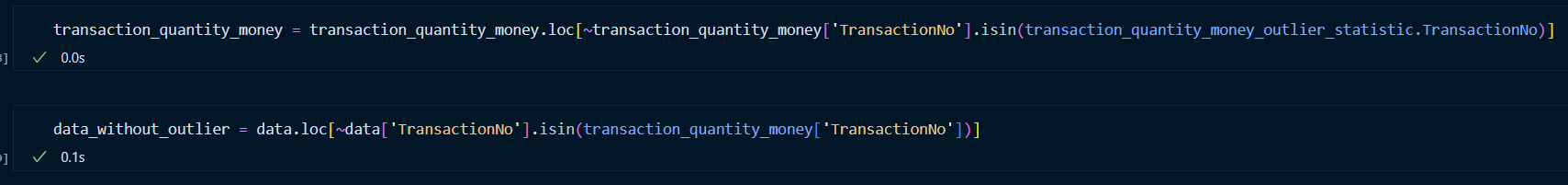
Nhận xét 2 thuật toán phát hiện outlier (DBSCAN và Statistic):

* Phương pháp DBSCAN không hiệu quả vì những giá trị tổng tiền chi và tổng lượng hàng mua không tuân theo qui luật nào mà có thể tìm ra bằng phương pháp DBSCAN với khoảng cách Euclide.
* Phương pháp statistic tìm được nhiều outlier hơn so với DBSCAN, nhưng điều này không có nghĩa là cứ tìm nhiều outlier hơn thì sẽ là tốt, nhưng đối với tập dữ liệu transaction này thì Statistic sẽ tốt hơn.



c) Loại bỏ outlier sau khi chọn thuật toán:





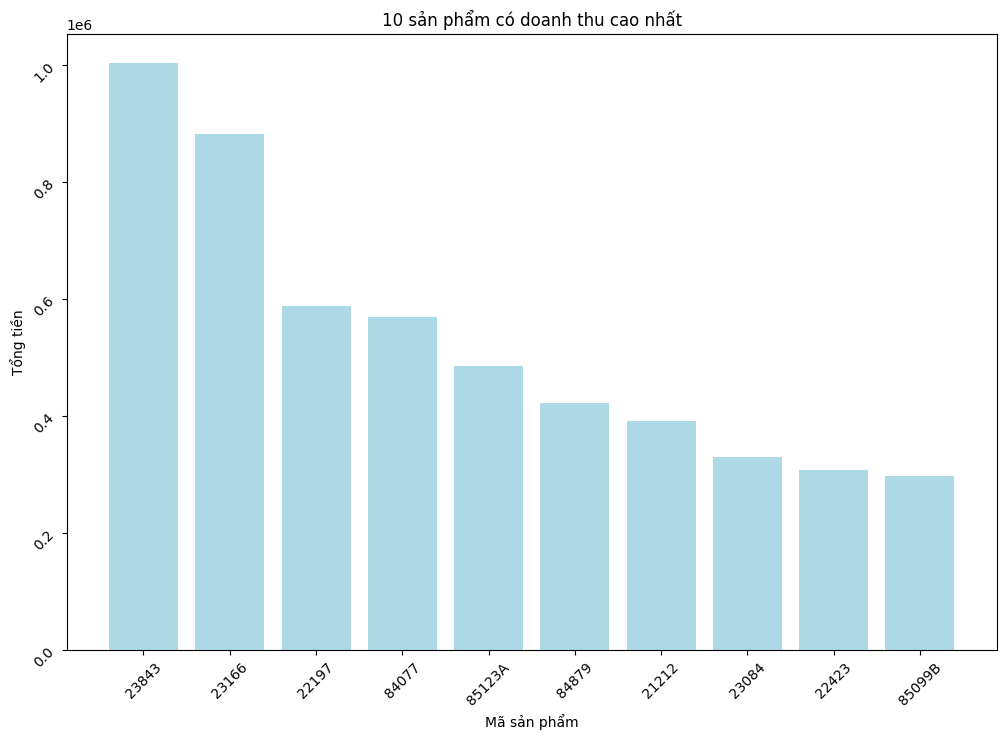
# PHẦN 3 - TRỰC QUAN HOÁ DỮ LIỆU

## 4.1) Câu hỏi đặt ra:

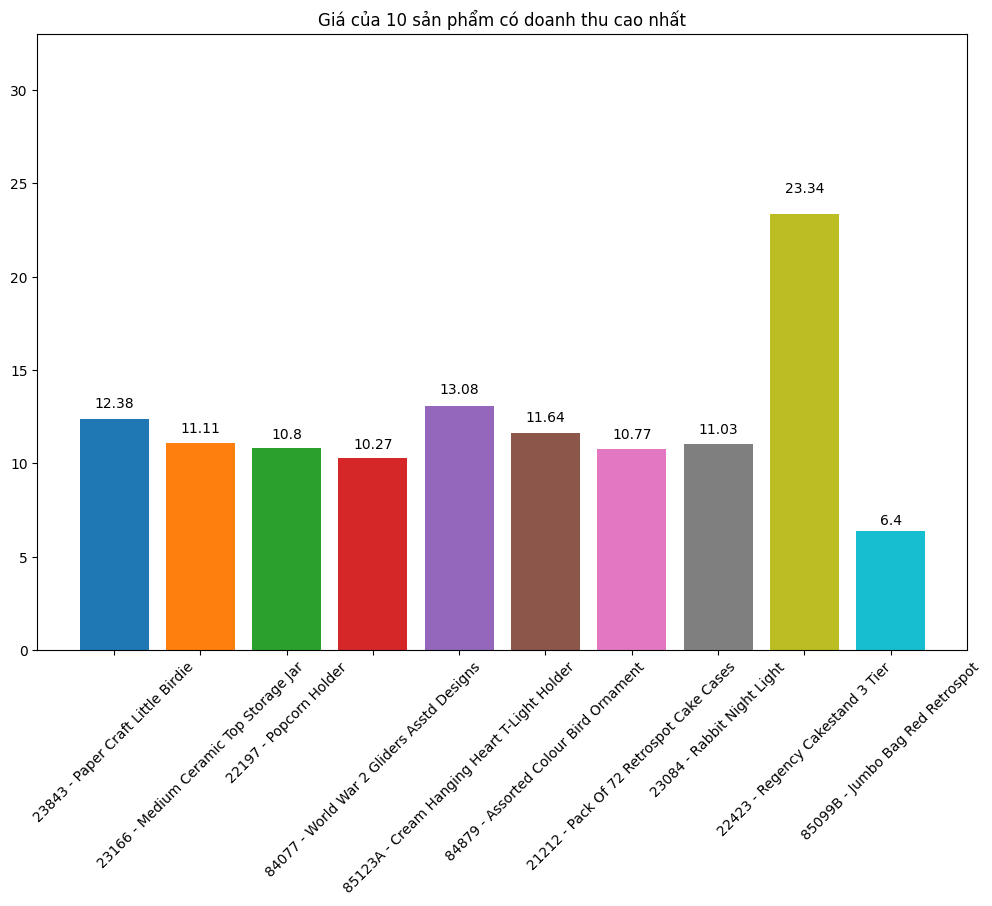
1. Xu hướng bán hàng thay đổi qua từng tháng như thế nào?
2. Những sản phẩm được mua thường xuyên nhất là gì?
3. Mỗi giao dịch khách hàng thường mua bao nhiêu sản phẩm?
4. Đâu là phân khúc khách hàng có lợi nhuận cao nhất?
5. Dựa trên những câu hỏi trên, có thể đề xuất chiến lược nào cho doanh nghiệp để thu được nhiều lợi nhuận hơn?

## 4.2) 10 giao dịch có tổng thanh toán cao nhất:

## 4.3) 10 sản phẩm có tổng doanh thu cao nhất:

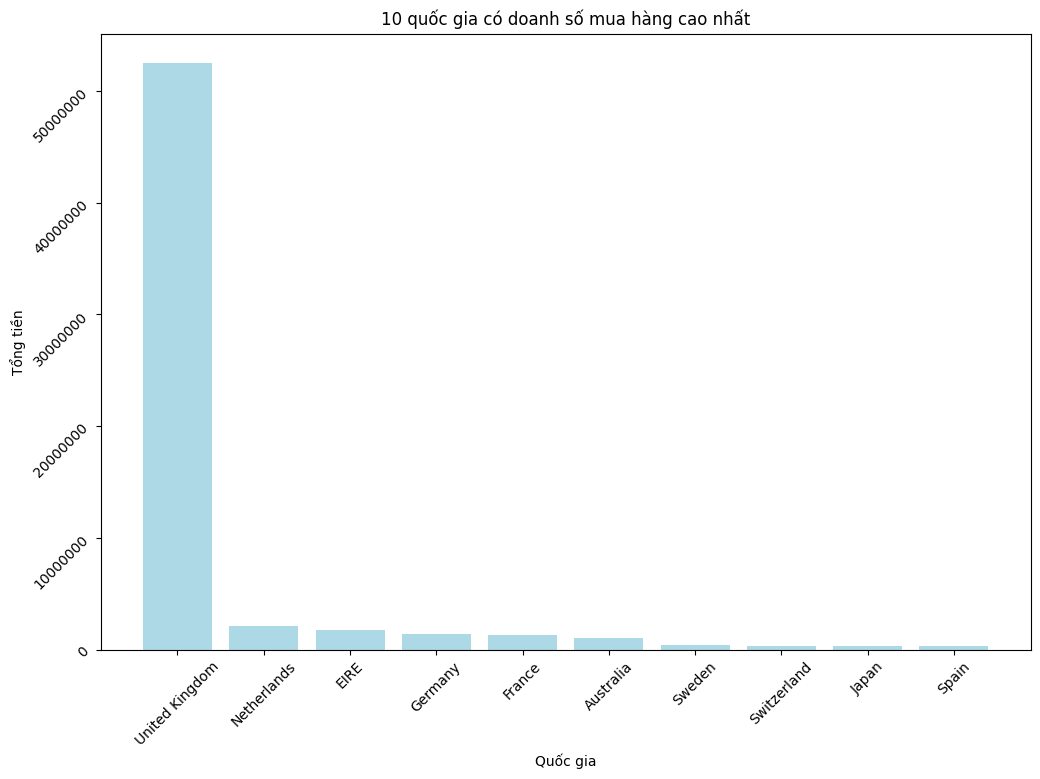


## 4.4) Giá của 10 sản phẩm có doanh thu cao nhất:

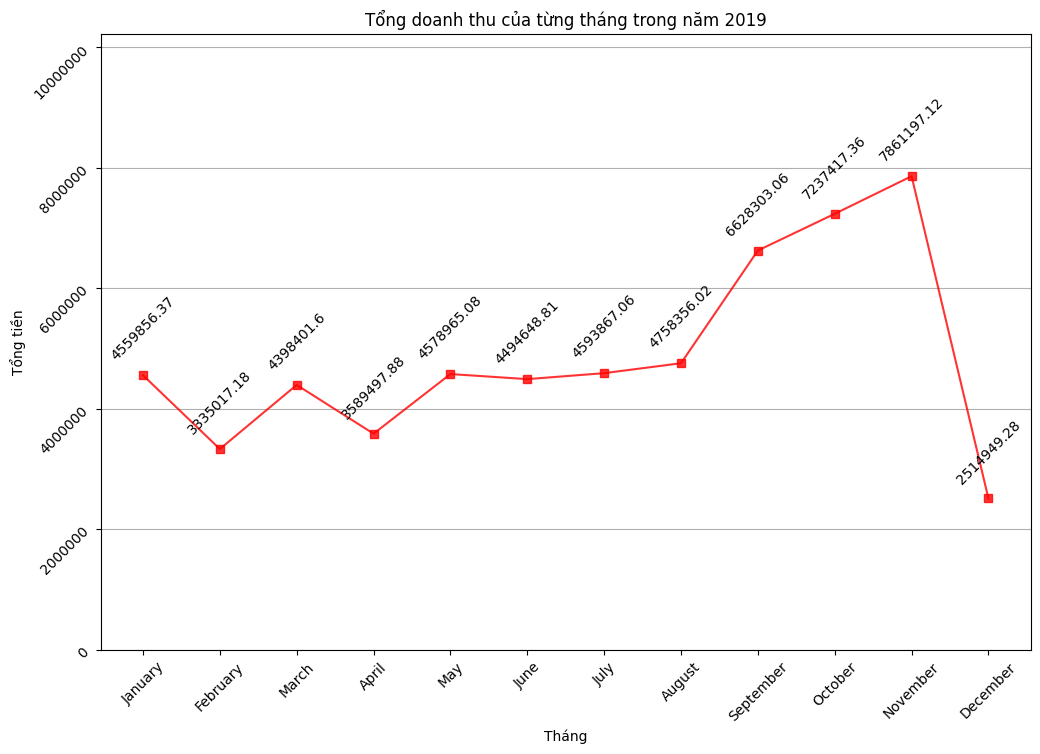


## 4.5) Tổng số lượng bán ra của 10 sản phẩm có doanh thu cao nhất:

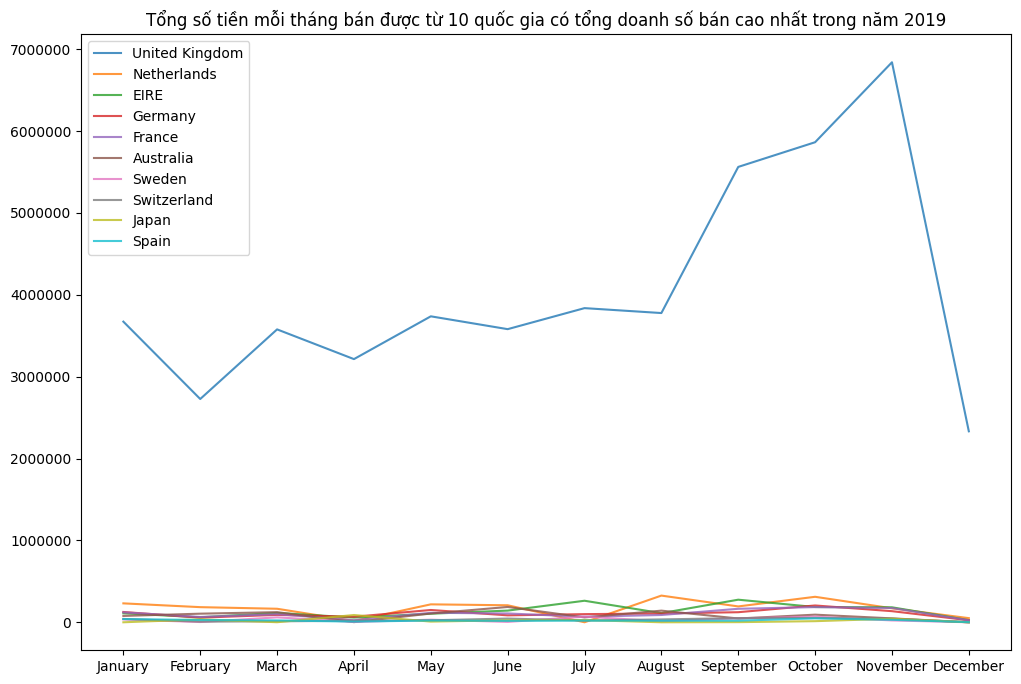
## 4.6) 10 quốc gia có doanh số mua hàng cao nhất:



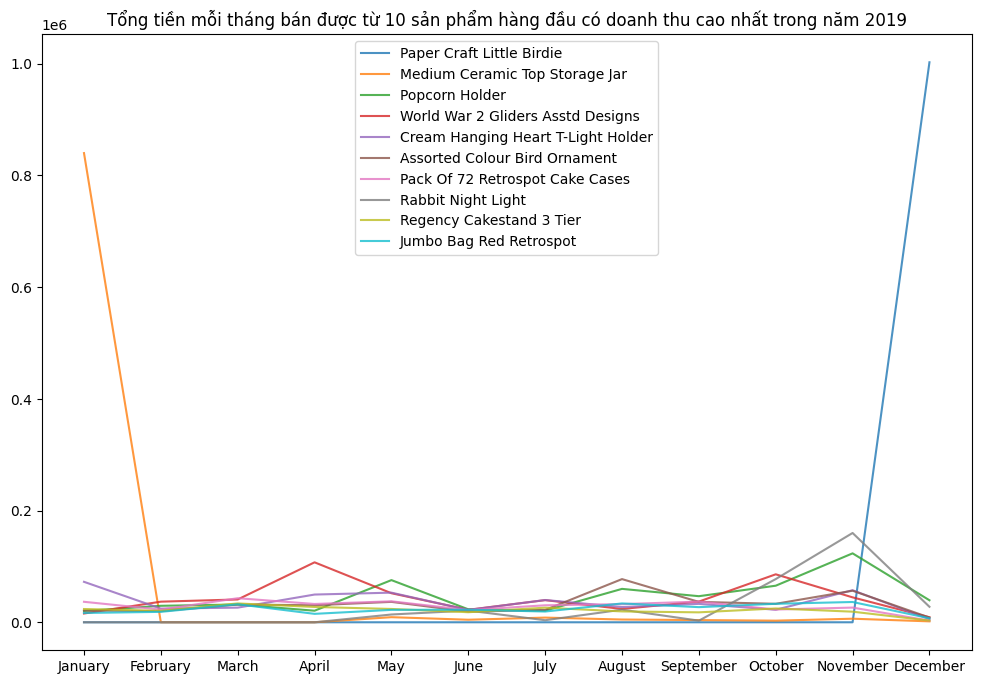
## 4.7) Tổng doanh thu từng tháng trong năm 2019:



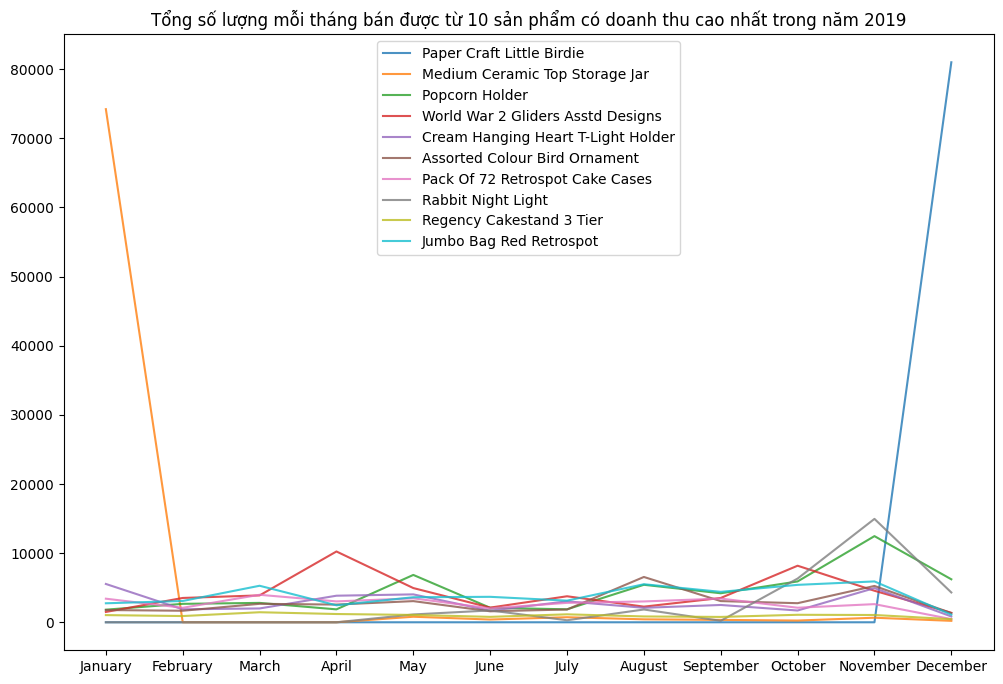
## 4.8) Tổng tiền mỗi tháng bán được từ 10 sản phẩm hàng đầu có doanh thu cao nhất trong năm 2019:



## 4.9) Tổng tiền mỗi tháng bán được từ 10 sản phẩm hàng đầu có doanh thu cao nhất trong năm 2019



## 4.10) Tổng số lượng mỗi tháng bán được từ 10 sản phẩm có doanh thu cao nhất trong năm 2019:



## 4.11) Trả lời câu hỏi đặt ra:

1. Xu hướng bán hàng thay đổi qua từng tháng như thế nào?

Dựa vào biểu đồ 'Tổng doanh thu từng tháng trong năm 2019', doanh số của doanh nghiệp này tăng tương đối hàng tháng kể từ tháng 1, nhưng vào tháng 12 năm 2019, doanh số giảm đáng kể. Có thể giải thích sự giảm này dựa vào biểu đồ 'Tổng số tiền mỗi tháng bán được từ 10 quốc gia có tổng doanh số bán cao nhất trong năm 2019', có thể thấy doanh số bán đến Anh vào tháng 12 năm 2019 cũng đã bị giảm, thực ra không phải do doanh nghiệp không bán được hàng, mà là do tập dữ liệu nhóm sử dụng ở đây chỉ cập nhật đến ngày 9/12/2019 cho nên không tổng quát được tháng 12.

1. Những sản phẩm được mua thường xuyên nhất là gì?

Dựa vào biểu đồ '10 sản phẩm có doanh số cao nhất', các sản phẩm được mua nhiều nhất và mang lại lợi nhuận cao là Paper Craft Little Birdie, Medium Ceramic Top Storage Jar, Popcorn Holder, World War 2 Gliders Asstd Designs, Cream Hanging Heart T-Light Holder, Assorted Colour Bird Ornament, Pack Of 72 Retrospot Cake Cases, Rabbit Night Light, Regency Cakestand 3 Tier, Jumbo Bag Red Retrospot.

Ta có thể thấy mặt hàng trang trí được bán khá chạy, gồm các tấm thiệp, hay là những đồ decor trong phòng, cho nên phân loại này cần được mở rộng hơn, đa dạng hơn với nhiều mẫu mã, màu sắc. Ngoài ra, phân loại sản phẩm bán chạy thứ 2 gồm những phân loại hộp đựng thức ăn, hay là dụng cụ nhà bếp, có thể dựa vào biểu đổ 'Tổng số lượng bán ra của 10 sản phẩm có doanh số cao nhất' mà thấy rằng những mặt hàng này được bán với số lượng rất lớn, mang lại doanh thu khá cao cho doanh nghiệp, ta sẽ cần phân tích sâu hơn trên phân loại này cũng như là mở rộng quy mô bán lẻ.

Dựa vào biểu đồ 'Tổng tiền mỗi tháng bán được từ 10 sản phẩm hàng đầu có doanh thu cao nhất trong năm 2019', ta có thể thấy 2 sản phẩm đứng đầu có doanh số bán khá lạ

Ở đây, sản phẩm đứng đầu doanh số là Paper Craft Little Birdie, nhưng sau vài bước kiểm tra, thì nhóm nhận ra chỉ có 1 khách hàng mua với 1 đơn hàng với 80995 sản phẩm, lúc này, ta nên loại sản phẩm này ra khỏi top 10 sản phẩm bởi vì không còn tính tổng quát.

Tương tự với sản phẩm đứng thứ 2, Medium Ceramic Top Storage Jar.

1. Mỗi giao dịch khách hàng thường mua bao nhiêu sản phẩm?

Dựa vào biểu đồ '10 giao dịch có tổng thanh toán cao nhất', khách hàng dường như mua số lượng sản phẩm khác nhau dựa trên nhu cầu của họ, nhưng ở đây chúng ta có thể thấy sự khác biệt có thể rất đáng kể. Sau một vài bước kiểm tra thì hầu như nhưng giao dịch này đều được thực hiện vào năm 2019, với số lượng sản phẩm khá cao, hầu như mỗi đơn hàng đều gần 300 mặt hàng khác nhau.

1. Đâu là phân khúc khách hàng có lợi nhuận cao nhất?

Nhóm khách hàng mang lại lợi nhuận cao nhất là nhóm khách hàng mua sản phẩm được bao gồm trong biểu đồ 'Giá của 10 sản phẩm có doanh thu cao nhất', biểu đồ 'Tổng số lượng bán ra của 10 sản phẩm có doanh số nhất' và biểu đồ '10 sản phẩm có doanh thu cao nhất', cùng với nhóm khách hàng đến từ quốc gia được bao gồm trong biểu đồ '10 quốc gia có doanh số bán cao nhất'.Có thể thấy Vương Quốc Anh là thị trường khá màu mỡ, nên mở rộng thêm những siêu thị bán lẻ ở quốc gia này.

1. Dựa trên những câu hỏi trên, có thể đề xuất chiến lược nào cho doanh nghiệp để thu được nhiều lợi nhuận hơn?

Dựa trên những phát hiện, có một số chiến lược để doanh nghiệp đạt được lợi nhuận cao hơn.

Thứ nhất, dựa trên biểu đồ 'Tổng số lượng bán ra của 10 sản phẩm có doanh số nhất' cùng với '10 sản phẩm có doanh số bán cao nhất', doanh nghiệp nên mở rộng 2 loại sản phẩm là trang trí, và đồ dùng nhà bếp, những phân loại mang lại doanh số cao nhất trong khoảng thời gian phân tích.

Thứ hai, tăng doanh số bán hàng, cũng như là cải thiện chất lượng cho các sản phẩm được bao gồm trong biểu đồ '10 sản phẩm có doanh số bán cao nhất' bằng các phương pháp tiếp thị đa dạng để thu hút người mua.

Thứ ba, tăng doanh số bán hàng tại các quốc gia được bao gồm trong biểu đồ '10 quốc gia có doanh số bán cao nhất' bằng các phương pháp tiếp thị đa dạng, quảng cáo các sản phẩm tại các quốc gia không được bao gồm để thu hút và tăng sự quan tâm của người mua từ những quốc gia này.

# PHẦN 4 – KHAI PHÁ DỮ LIỆU

## 4.1) Giới thiệu thuật toán:

### 4.1.1) Thuật toán gom cụm:

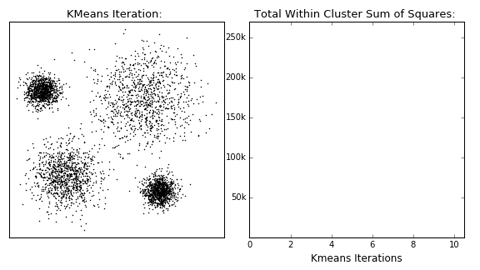
#### a) Kmeans:

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

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

1. Chọn **K** điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

**Miêu tả thuật toán chạy:**



#### b) DBSCAN:

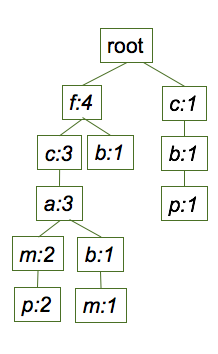
*Đã giới thiệu ở 3.2.3.a*

### 4.1.2) Thuật toán luật kết hợp:

#### a) Thuật toán FP-Growth:

* FP-Growth biểu diễn dữ liệu các giao dịch bằng một cấu trúc dữ liệu gọi là FP-Tree.
* FP-Growth sử dụng FP-Tree để xác định trực tiếp các tập hạng mục phổ biến (không sinh các tập hạng mục ứng viên từ các tập hạng mục ứng viên trước).
* Khi một FP-Tree đã được xây dựng, FP-Growth sử dụng cách tiếp cận chia để trị đệ quy để khai thác các tập phổ biến.
* Với mỗi giao dịch, FP-Tree xây dựng một đường đi (path) trong cây.
* Hai giao dịch có chứa cùng một số các mục, thì đường đi của chúng sẽ có phần (đoạn) chung.
* Càng nhiều các đường đi có phần tử chung, thì việc biểu diễn bằng FP-Tree sẽ càng gọn (compressed/compacted).

**Xây dựng FP-Tree**

* Ban đầu, FP-Tree chỉ chứa duy nhất nút gốc (được biểu diễn bởi ký hiệu null).
* Cơ sỡ dữ liệu các giao dịch được duyệt lần 1, để xác định độ hỗ trợ của mỗi mục.
* Các mục không thường xuyên (infrequent items) bị loại bỏ.
* Các mục thường xuyên (frequent items) được sắp xếp theo thứ tự giảm dần về độ hỗ trợ.
* Cơ dở dữ liệu các giao dịch được duyệt lần thứ 2, để xây dựng FP-Tree.

#### b) Thuật toán Apriori:

Apriori là thuật toán sử dụng khai thác các tập đối tượng trong dữ liệu có mối quan hệ liên quan với nhau. Apriori nhắm đến mục tiêu khai thác sự thường xuyên và mối liên quan kết hợp trên database.

Ba thành phần đã cho bao gồm thuật toán apriori.

* Support
* Confidence
* Lift

*Miêu tả thuật toán:*

Cho cơ sở dữ liệu gồm các giao dịch T là tập các giao dịch t1, t2, …, tn.

T = {t1, t2, …, tn}. T gọi là cơ sở dữ liệu giao dịch (Transaction Database)

Mỗi giao dịch ti bao gồm tập các đối tượng I (gọi là itemset)

I = {i1, i2, …, im}. Một itemset gồm k items gọi là k-itemset

Mục đích của luật kết hợp là tìm ra sự kết hợp (association) hay tương quan (correlation) giữa các items. Những luật kết hợp này có dạng X =>Y

Trong Basket Analysis, luật kết hợp X =>Y có thể hiểu rằng những người mua các mặt hàng trong tập X cũng thường mua các mặt hàng trong tập Y. (X và Y gọi là itemset).

Ví dụ, nếu X = {Apple, Banana} và Y = {Cherry, Durian} và ta có luật kết hợp X =>Y thì chúng ta có thể nói rằng những người mua Apple và Banana thì cũng thường mua Cherry và Durian.

*Tóm tắt thuật toán:*

1. Duyệt (Scan) toàn bộ transaction database để có được support S của 1-itemset, so sánh S với min\_sup, để có được 1-itemset (L1)

2. Sử dụng Lk-1 nối (join) Lk-1 để sinh ra candidate k-itemset. Loại bỏ các itemsets không phải là frequent itemsets thu được k-itemset

3. Scan transaction database để có được support của mỗi candidate k-itemset, so sánh S với min\_sup để thu được frequent k –itemset (Lk)

4. Lặp lại từ bước 2 cho đến khi Candidate set (C) trống (không tìm thấy frequent itemsets)

5. Với mỗi frequent itemset I, sinh tất cả các tập con s không rỗng của I

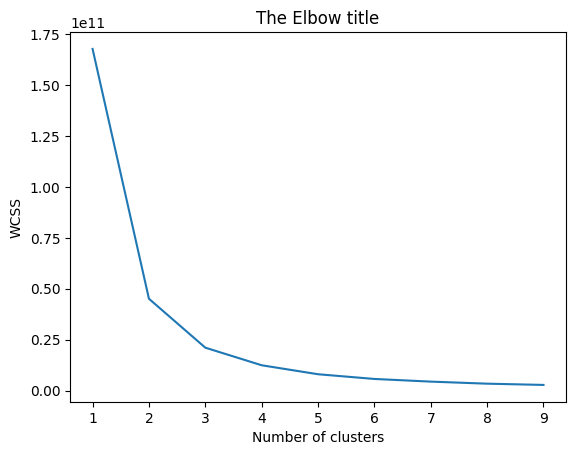
6. Với mỗi tập con s không rỗng của I, sinh ra các luật s => (I-s) nếu độ tin cậy (Confidence) của nó > =min\_conf

## 4.2) Thực hiện thuật toán:

### 4.2.1) Thuật toán gom cụm:

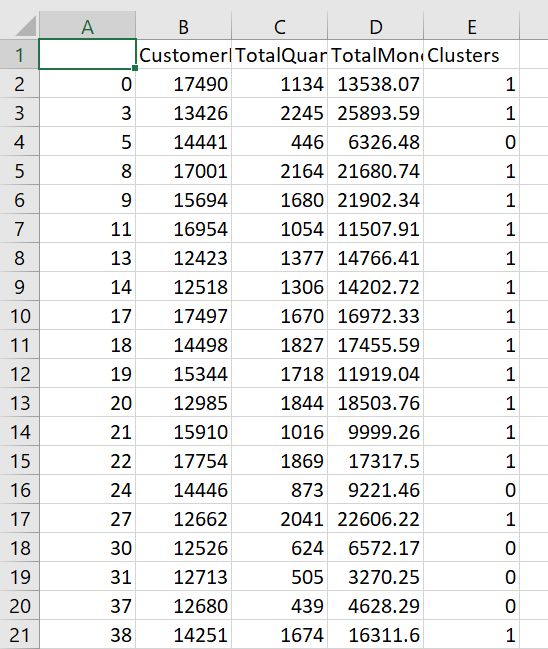
#### a) Kmeans:

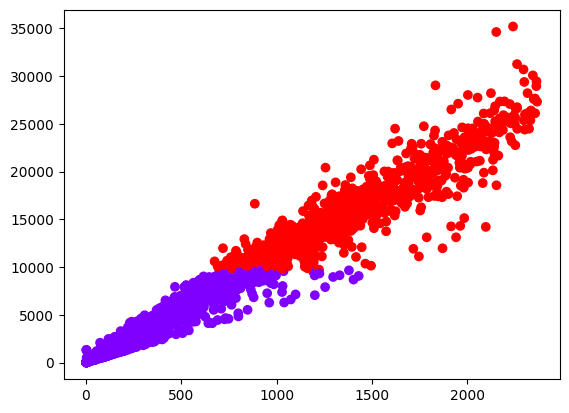
Thực hiện thuật toán gom cụm Kmeans đối với khách hàng dựa trên tổng tiền đã chi và tổng lượng hàng đã mua.

Elbow method để chọn k:

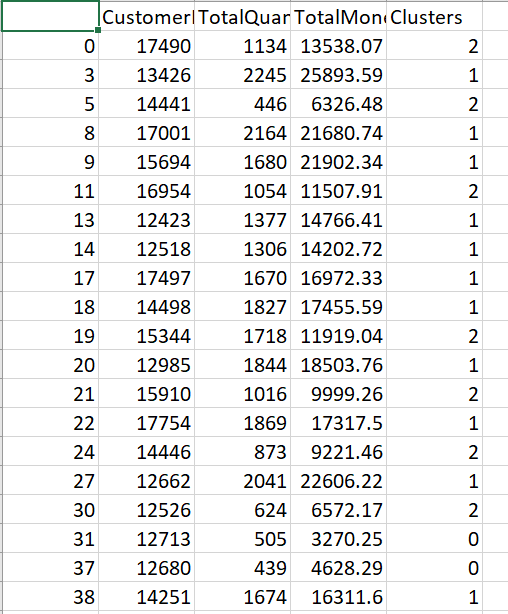
Nhận thấy rằng điểm gấp khúc là ở 2 và 3 nên nhóm sẽ chạy thuật toán Kmeans với k ở 2 và ở 3.

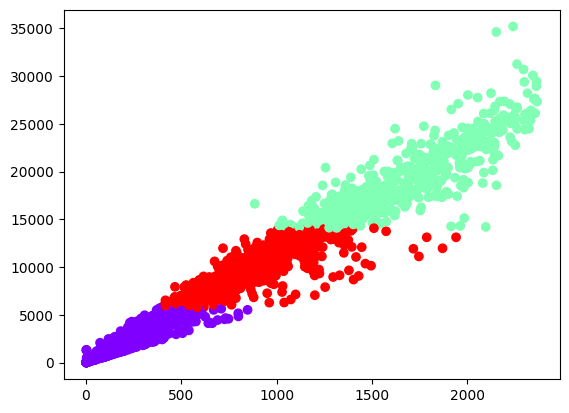
Kết quả thu được đối với k = 2:



Thể hiện trên biểu đồ:

Kết quả thu được với k = 3:



Thể hiện trên biểu đồ

**Nhận xét:**

*Đối với k = 2:*

* Khách hàng nhóm 1 sẽ mua số lượng từ 0 tới khoảng 1500, số tiền chi tiêu sẽ vào khoảng từ 0 tới 10000.
* Khách hàng nhóm 2 sẽ mua số lượng từ 600 tới khoảng 1500, số tiền chi tiêu sẽ vào khoảng 10000 tới 35000.

*Đối với k = 3:*

* Khách hàng nhóm 1 sẽ mua số lượng từ 0 tới khoảng 800, số tiền chi tiêu sẽ vào khoảng 6000.
* Khách hàng nhóm 2 sẽ mua số lượng từ 400 tới khoảng 1900, số tiền chi tiêu sẽ vào khoảng 6000 tới 14000.
* Khách hàng nhóm 3 sẽ mua số lượng từ 1000 tới khoảng 2500, số tiền chi tiêu sẽ vào khoảng 15000 tới 35000.

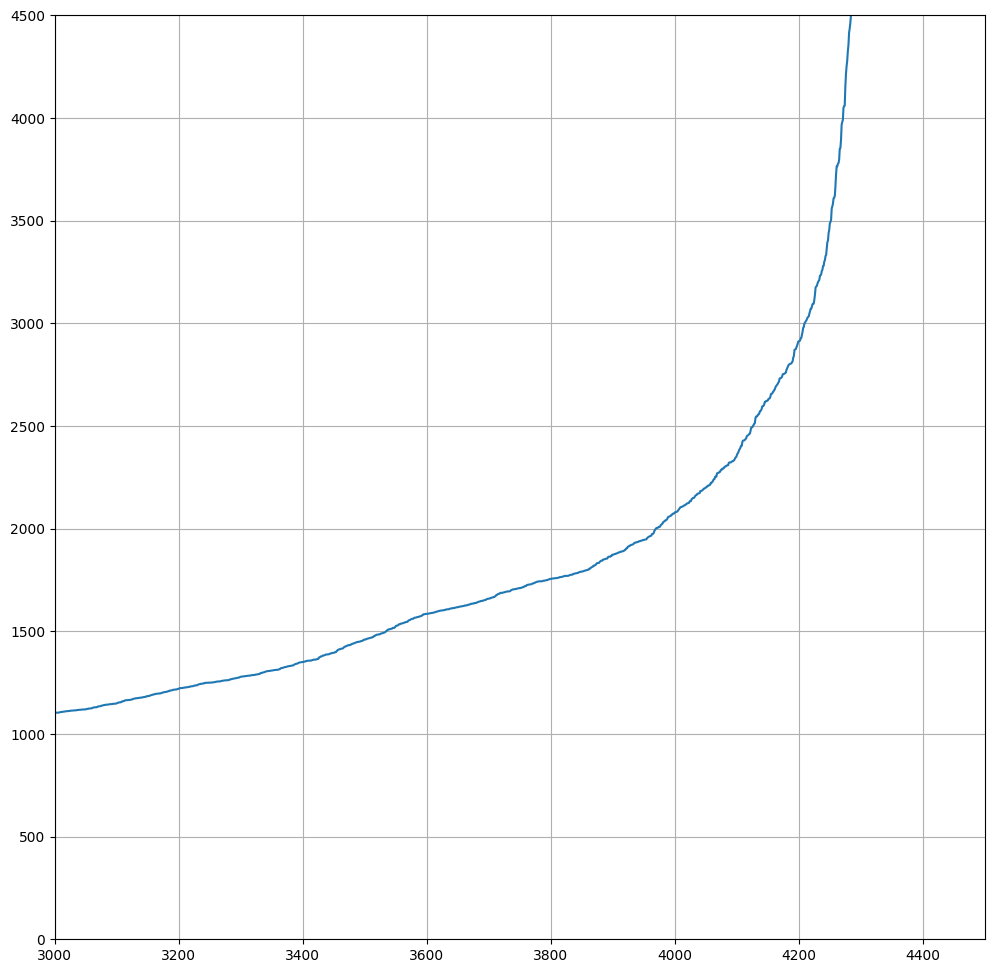
**Kết luận:**

Theo nhóm, chọn k = 3 sẽ tốt hơn vì:

* Có thể phân loại được nhóm khách hàng chi tiết hơn so với k = 2, từ đó có thể đưa ra những chương trình ưu đãi phù hợp hơn với tất cả các khách hàng.
* Do tham khảo ở COOPMART và CGV; Đối với COOPMART sẽ có các loại thẻ đồng, bạc, vàng, bạch kim; Đối với CGV cũng phân làm 3 nhóm khách hàng là Member, VIP, VVIP.

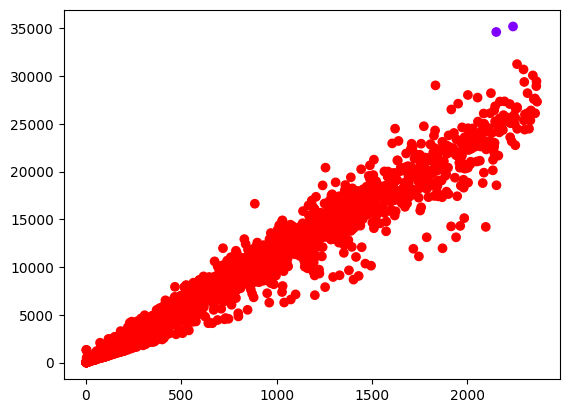
#### b) DBSCAN:

Elbow method để chọn eps cho DBSCAN:



Nhóm quyết định chọn eps = 3000, min\_sample = dimension + 1 = 4.

Kết quả thu được:



Nhận xét: Không phân biệt được rõ ràng loại khách hàng.

#### c) So sánh 2 thuật toán gom cụm:

Trong tập dữ liệu SalesTransaction.csv, thuật toán Kmeans sẽ hiệu quả hơn, vì khi dùng elbow method cho Kmeans, ta có thể xác định được có bao nhiêu cụm, và Kmeans thì dựa trên trung bình để xác định cụm

Trong khi DBSCAN có vẻ dễ thực hiện hơn nhưng lại dựa vào khoảng cách Euclide giữa các điểm để tìm core point, border và noise và dường như không có ý nghĩa đối với tập dữ liệu là các transaction.

#### d) Kết luận:

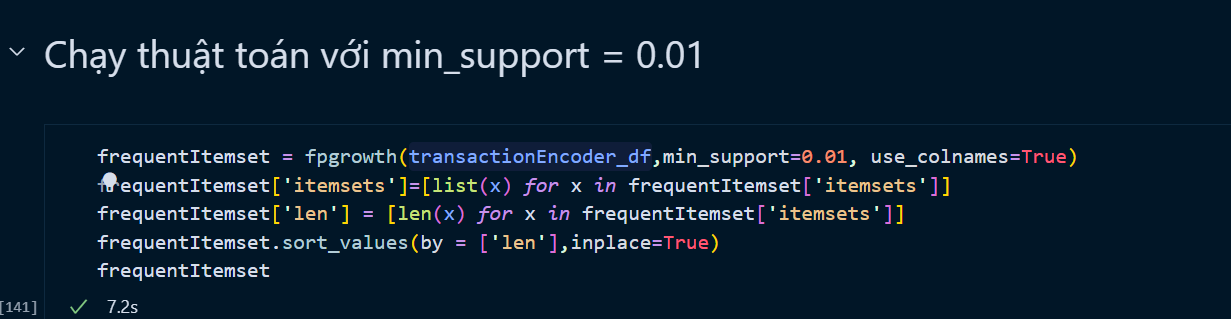
Dựa vào thuật toán Kmeans, doanh nghiệp có tập dữ liệu này có thể phân cụm khách hàng làm 3 nhóm, từ đó sẽ có những ưu đãi phù hợp cho từng đối tượng khách hàng, giúp giữ chân khách hàng và thu hút thêm nhiều khách hàng khác.

### 4.2.2) Thuật toán luật kết hợp:

#### a) Chuẩn bị:

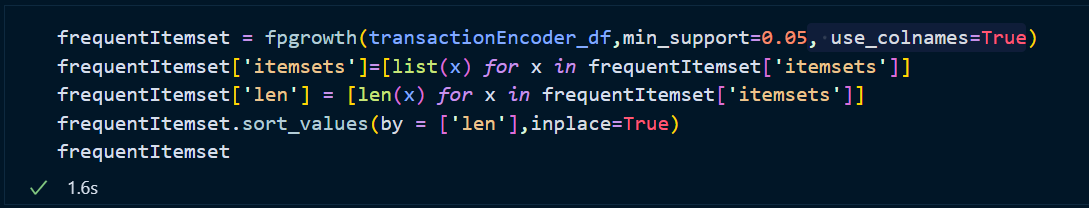
#### b) Thuật toán FP-Growth:

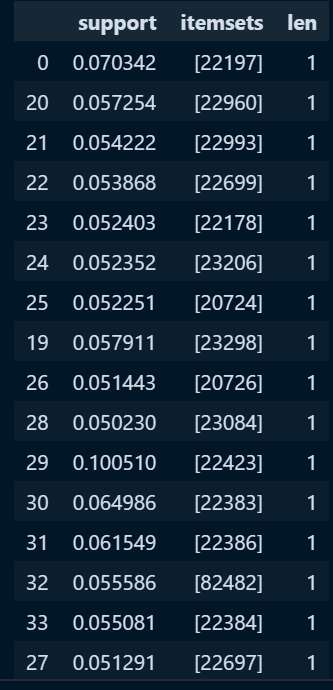
Chạy thuật toán với min\_support = 0.01:



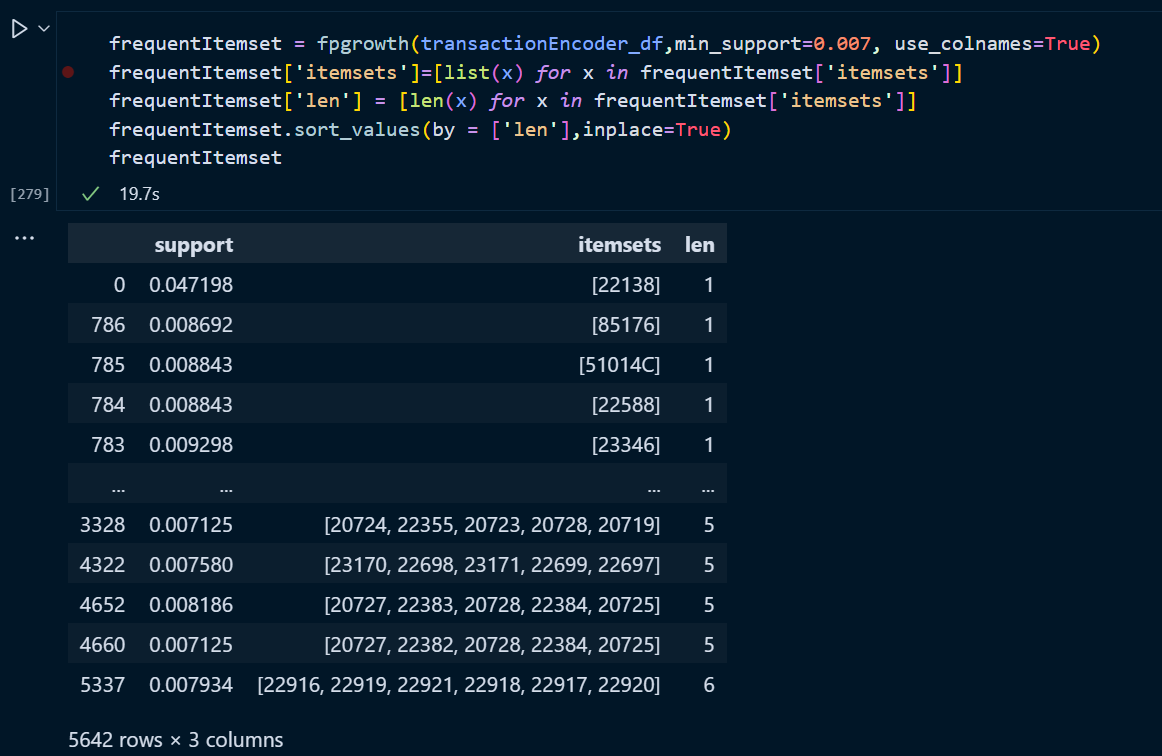


Chạy thuật toán với min\_support = 0.05:

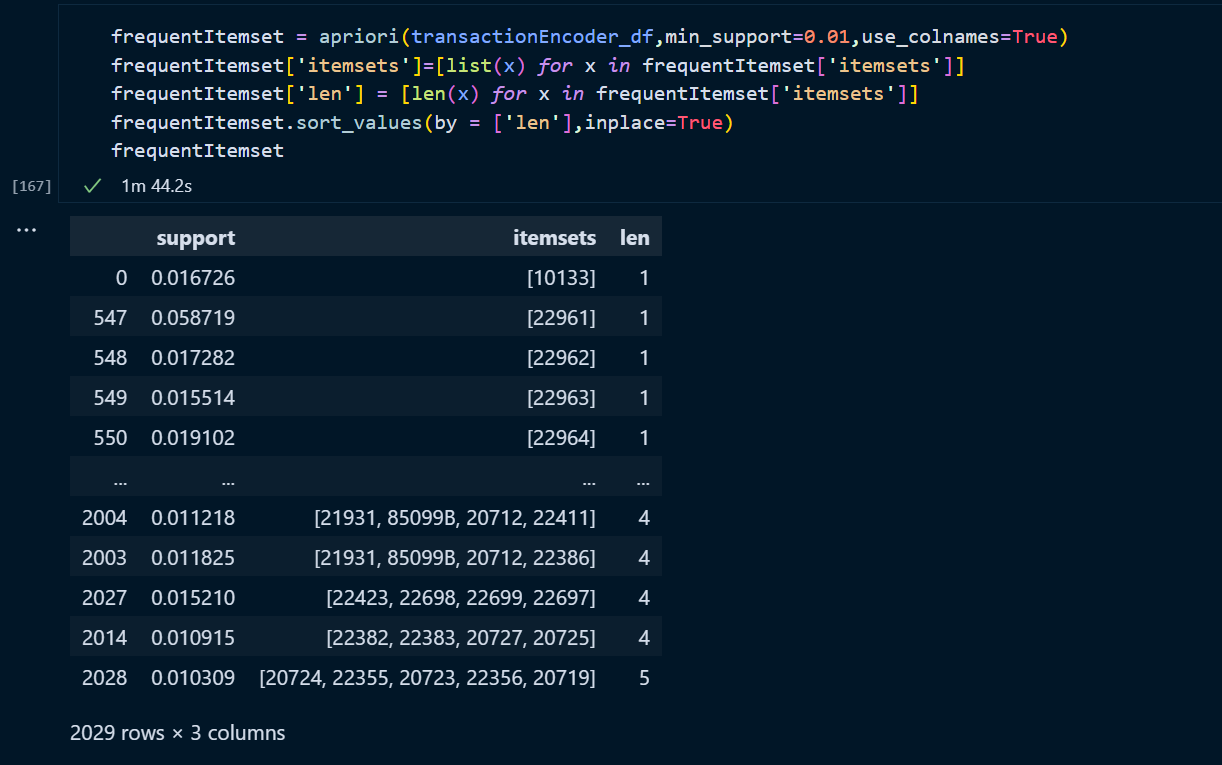




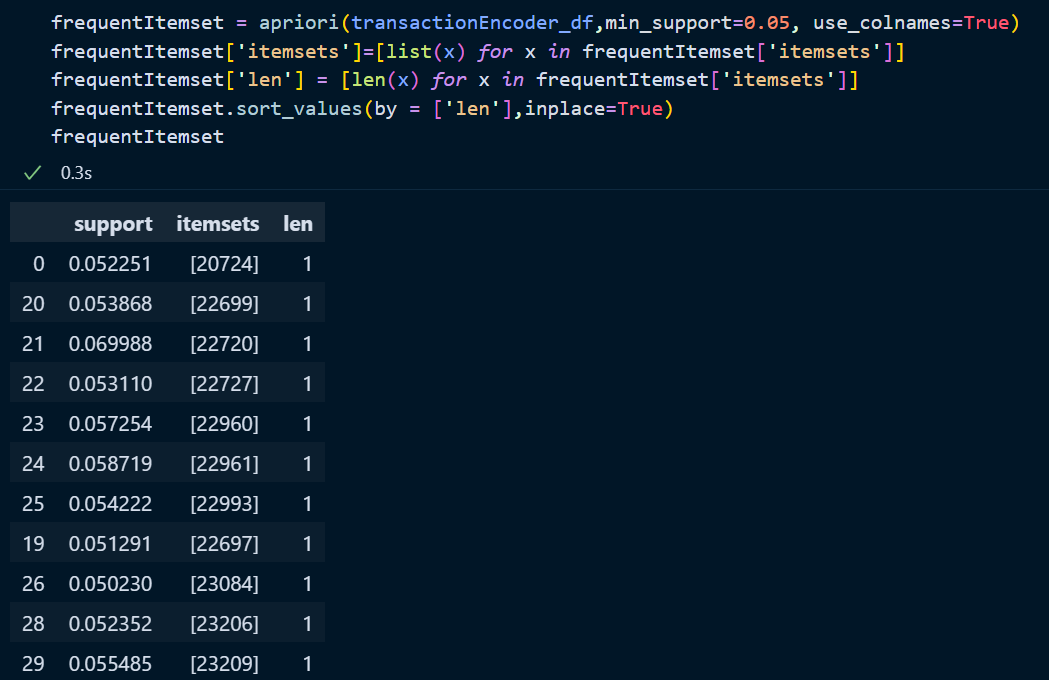
Chạy thuật toán với min\_support = 0.007:



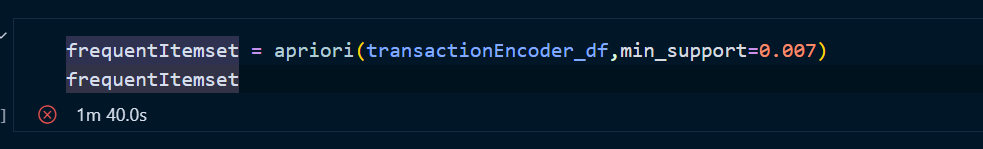
#### c) Thuật toán Apriori:

Chạy thuật toán với min\_support = 0.01:

Chạy thuật toán với min\_support = 0.05:



Chạy thuật toán với min\_support = 0.007



#### d) So sánh 2 thuật toán luật kết hợp:

Điểm giống nhau:

Cả 2 thuật toán đều chạy ra cùng 1 kết quả.

Điểm khác nhau:

|  |  |  |
| --- | --- | --- |
|  | FP-Growth | Apriori |
| 0.01 | Thời gian chạy: 7.2s | Thời gian chạy: 1m44s |
| 0.05 | Thời gian chạy: 1.6s | Thời gian chạy: 0.3s |
| 0.007 | Thời gian chạy: 19.7s | Không thể chạy |

So sánh khác min\_support:

Với min\_support 0.01:

|  |  |
| --- | --- |
| Số lượng phần tử trong một itemset | Số lượng itemset chứa từng đó phần tử: |
| 1 | 830 |
| 2 | 912 |
| 3 | 216 |
| 4 | 25 |
| 5 | 1 |

Với min\_support 0.007:

|  |  |
| --- | --- |
| Số lượng phần tử trong một itemset | Số lượng itemset chứa từng đó phần tử: |
| 1 | 1173 |
| 2 | 2898 |
| 3 | 216 |
| 4 | 314 |
| 5 | 36 |
| 6 | 1 |

Nhận xét:

* Đối với khác mức min\_support: Min\_support càng cao thì itemset được chọn sẽ càng ít và số lượng phần tử trong một itemset cũng sẽ ít. Còn nếu min\_support càng thấp thì itemset sẽ nhiều hơn và số lượng phần tử trong 1 itemset sẽ nhiều hơn.
* Hiệu quả trên các bộ dữ liệu: Apriori làm việc tốt với cơ sở dữ liệu lớn. FP-Growth làm việc tốt với các cơ sở dữ liệu nhỏ và các tập phổ biến sinh ra không quá dài (độ dài lớn nhất của một tập frequent itemset không quá lớn).
* Thuật toán Apriori thực hiện nhiều lần quét để có thể tạo ra các tập ứng viên candidate itemsets. Thuật toán FP Growth chỉ cần quét cơ sở dữ liệu đúng hai lần.

#### e) Kết luận:

Dựa vào thuật toán luật kết hợp, doanh nghiệp có tập dữ liệu này có thể thiết kế website mà các sản phẩm nào có cùng itemset sẽ được đề xuất chung với nhau, giúp tăng doanh thu cho doanh nghiệp.

# PHẦN 5 – KẾT QUẢ, THỰC NGHIỆM VÀ THẢO LUẬN

Nguồn:

<https://codelearn.io/sharing/jupyter-notebook-tutorial>

<http://uet.vnu.edu.vn/~thuyhq/Student_Thesis/K44_Do_Thi_Dieu_Ngoc_Thesis.pdf>

<https://vi.wikipedia.org/wiki/Visual_Studio_Code>

<https://aws.amazon.com/vi/what-is/python/>

dataset: <https://www.kaggle.com/datasets/gabrielramos87/an-online-shop-business?select=Sales+Transaction+v.4a.csv>

Outlier detection:

Phương pháp số học: <https://machinelearningcoban.com/tabml_book/ch_data_processing/process_outliers.html>

DBSCAN:

<https://phamdinhkhanh.github.io/deepai-book/ch_ml/DBSCAN.html>

FP-Growth:

<https://nhannguyen95.github.io/bai-tap-khai-pha-tap-pho-bien-bang-thuat-toan-fp-growth/>

Kmeans:

<https://machinelearningcoban.com/2017/01/01/kmeans/>

Apriori:

<http://bis.net.vn/forums/p/389/749.aspx>

So sánh apriori và Fpgrowth:

https://viblo.asia/p/khai-pha-du-lieu-va-lop-bai-toan-khai-thac-cac-tap-pho-bien-p2-m68Z0W06KkG