**VEF Data Analytics**

**07/2022**

**DỰ ÁN CUỐI KHÓA**

**Chủ đề: E-commerce**

**Dataset: Brazilian E-Commerce Public Dataset by Olist**

**Người thực hiện: Trần Mạnh Tường**

**Contact: tuong9245@gmail.com**

**Mục lục nội dung**

[1. Tổng quan về mục tiêu dự án 4](#_Toc112269308)

[2. Cách tiếp cận vấn đề 4](#_Toc112269309)

[Thinking flow 4](#_Toc112269310)

[Working flow 4](#_Toc112269311)

[3. Giới thiệu tổng quan về nguồn dữ liệu 4](#_Toc112269312)

[Tổng quan: 4](#_Toc112269313)

[Data Schema 5](#_Toc112269314)

[ERD 5](#_Toc112269315)

[4. Giới thiệu về dữ liệu 6](#_Toc112269316)

[Customer 6](#_Toc112269317)

[Sellers 7](#_Toc112269318)

[Order\_items 7](#_Toc112269319)

[Payments 8](#_Toc112269320)

[Orders 8](#_Toc112269321)

[Order\_reviews 9](#_Toc112269322)

[Products 9](#_Toc112269323)

[Geolocation 10](#_Toc112269324)

[5. Customer Segmentation 11](#_Toc112269325)

[Ý tưởng 11](#_Toc112269326)

[Truy xuất dữ liệu 11](#_Toc112269327)

[Query: 11](#_Toc112269328)

[Giải thích: 11](#_Toc112269329)

[Phân tích dữ liệu 11](#_Toc112269330)

[Tiền xử lý dữ liệu: 11](#_Toc112269331)

[RFM Score 12](#_Toc112269332)

[EDA 13](#_Toc112269333)

[Phân cụm khách hàng: 15](#_Toc112269334)

[Kết luận 17](#_Toc112269335)

[6. Retention Cohort 17](#_Toc112269336)

[Ý tưởng 17](#_Toc112269337)

[Truy xuất dữ liệu 17](#_Toc112269338)

[Phân tích dữ liệu 17](#_Toc112269339)

[Tiền xử lý dữ liệu 17](#_Toc112269340)

[Phân tích Cohort (Cohort Analysis) 18](#_Toc112269341)

[Kết luận 20](#_Toc112269342)

[7. Root Cause Analysis 20](#_Toc112269343)

[Ý tưởng 20](#_Toc112269344)

[Truy xuất dữ liệu 21](#_Toc112269345)

[Shipment 21](#_Toc112269346)

[Rating 21](#_Toc112269347)

[Payment 22](#_Toc112269348)

[Phân tích dữ liệu 22](#_Toc112269349)

[Shipment 22](#_Toc112269350)

[Rating 24](#_Toc112269351)

[Payment 26](#_Toc112269352)

[Kết luận 27](#_Toc112269353)

[8. Market Basket Analysis 27](#_Toc112269354)

[Ý tưởng 27](#_Toc112269355)

[Luật kết hợp 27](#_Toc112269356)

[Truy xuất dữ liệu 28](#_Toc112269357)

[Query: 28](#_Toc112269358)

[Giải thích: 28](#_Toc112269359)

[Phân tích dữ liệu 29](#_Toc112269360)

[Tiền xử lý dữ liệu 29](#_Toc112269361)

[EDA 29](#_Toc112269362)

[Luật kết hợp theo danh mục hàng 30](#_Toc112269363)

[Luật kết hợp giữa các sản phẩm 31](#_Toc112269364)

[Kết luận 31](#_Toc112269365)

[9. Demand prediction model 32](#_Toc112269366)

[Ý tưởng 32](#_Toc112269367)

[Phân tích dữ liệu 32](#_Toc112269368)

[Tiền xử lý dữ liệu 32](#_Toc112269369)

[Xây dựng mô hình 34](#_Toc112269370)

[Mô hình dựa trên SelectKBest 34](#_Toc112269371)

[Kết luận & hướng phát triển 34](#_Toc112269372)

[10. Tổng kết 35](#_Toc112269373)

[Tại sao có các bài toán? 35](#_Toc112269374)

[Kết luận 35](#_Toc112269375)

**Tài liệu tham khảo**

[**https://en.wikipedia.org/wiki/Association\_rule\_learning**](https://en.wikipedia.org/wiki/Association_rule_learning)

[**https://goldinlocks.github.io/Market-Basket-Analysis-in-Python/**](https://goldinlocks.github.io/Market-Basket-Analysis-in-Python/)

[**https://www.activestate.com/blog/cohort-analysis-with-python/**](https://www.activestate.com/blog/cohort-analysis-with-python/)

[**https://towardsdatascience.com/an-rfm-customer-segmentation-with-python-cf7be647733d**](https://towardsdatascience.com/an-rfm-customer-segmentation-with-python-cf7be647733d)

[**https://laptrinhx.com/a-gentle-introduction-to-customer-segmentation-with-rfm-scores-4172308924/**](https://laptrinhx.com/a-gentle-introduction-to-customer-segmentation-with-rfm-scores-4172308924/)

**Source code**

1. [**Dataset**](https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce)
2. [**Source code**](https://drive.google.com/drive/folders/12ippOgE3umta3J9tAoxobRoOVBR78HQB?usp=sharing)
3. [**Report**](https://drive.google.com/file/d/1592mYPd6yJ6e0DlAxdJOnVtpViDgY5Or/view?usp=sharing)

# Tổng quan về mục tiêu dự án

Từ bộ dữ liệu về giao dịch của sàn thương mại điện tử trong quá khứ, tìm ra cách để cải thiện hệ thống, từ đó tăng số lượng giao dịch của khách hàng, giữ chân khách hàng ở lại hệ thống.

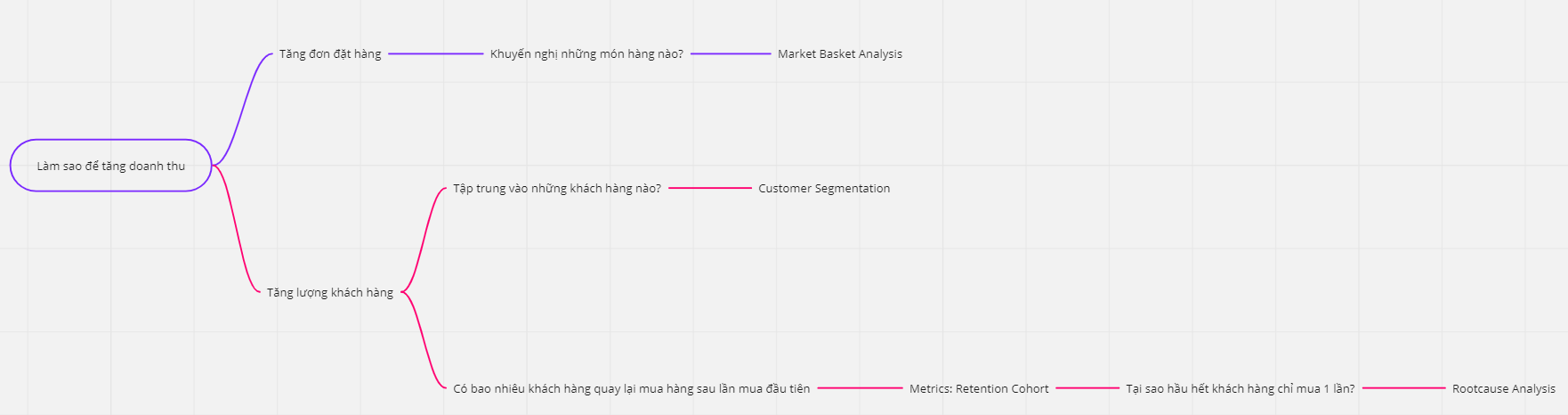
# Cách tiếp cận vấn đề

## Thinking flow

Xây dựng metrics retention rate để xem xét sự trung thành của khách hàng đối với hệ thống. Nếu tỷ lệ khách hàng rời bỏ lớn, thực hiện Root Cause Analysis để tìm hiểu nguyên nhân và đưa ra hướng khắc phục.

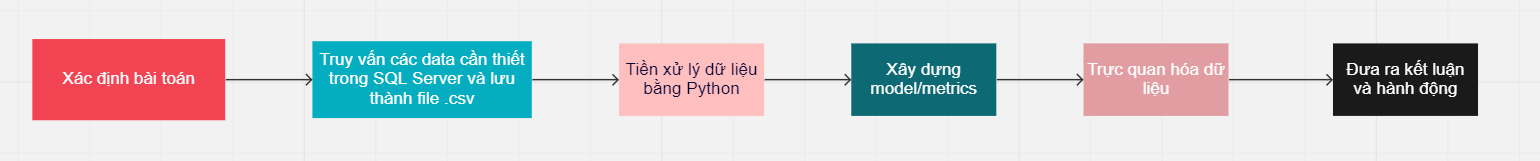
Đâu là nhóm khách hàng nên tập trung vào? Vì có một số nhóm khách hàng chỉ thực hiện giao dịch khi có khuyến mãi, nên ta cần loại nhóm này ra khỏi nhóm khách hàng tiềm năng. Thực hiện phân loại khách hàng (Customer segmentation) để tìm ra các nhóm khách hàng này.

Khi khách hàng tìm kiếm một sản phẩm, nên khuyến nghị sản phẩm nào để khách hàng có xác suất mua cao nhất? Sau khi khách hàng mua một món hàng, nên khuyến nghị món nào tiếp theo? (Ví dụ như khi mua máy tính thì khách hàng sẽ thường có nhu cầu mua các phụ kiện đi kèm như chuột, bàn phím,…). Thực hiện phân tích giỏ hàng (Market basket analysis) để tìm ra các sản phẩm có quan hệ với nhau.



## Working flow

Được thực hiện theo 6 bước trong việc phân tích dữ liệu: Ask, Prepare, Process, Analyze, Share, Act.



# Giới thiệu tổng quan về nguồn dữ liệu

## Tổng quan:

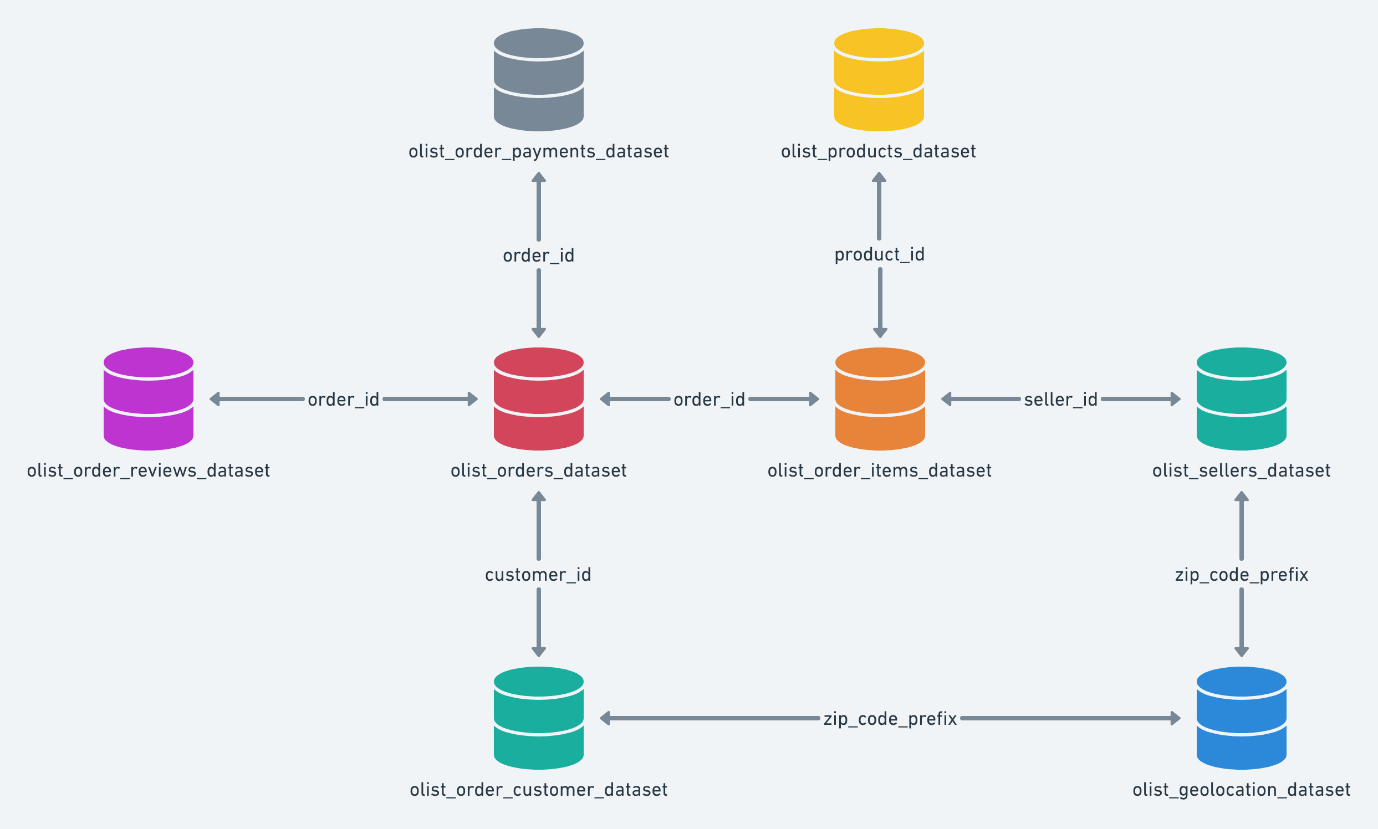
Bộ dữ liệu liên quan đến thương mại điện tử (e-commerce) của cửa hàng Olist (Brazil) với hơn 100000 đơn hàng được ghi nhận từ năm 2016-2018 và một số thông tin liên quan khác về khách hàng, sản phẩm, thanh toán,… được đính kèm.

Dữ liệu thu thập được bao gồm 8 file .csv (comma separated values) sẽ được lưu vào trong cơ sở dữ liệu (database) để truy xuất và thực hiện các nhiệm vụ khác.

Ở dự án này, cơ sở dữ liệu được sử dụng là Microsoft SQL Server, một cơ sở dữ liệu miễn phí do Microsoft phát hành.

*Nguồn dữ liệu: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce*

## Data Schema



## ERD

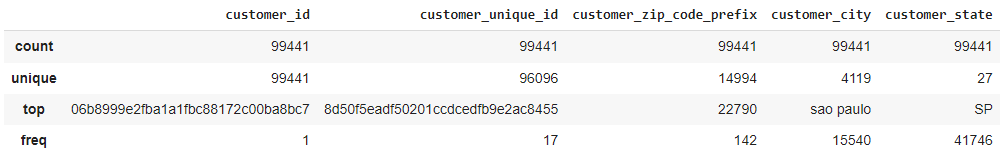
Để thuận tiện hơn cho việc truy vấn một số bảng đã được đổi tên bằng cách bỏ đi phần olist\_ ở đầu và \_dataset ở đuôi.



# Giới thiệu về dữ liệu

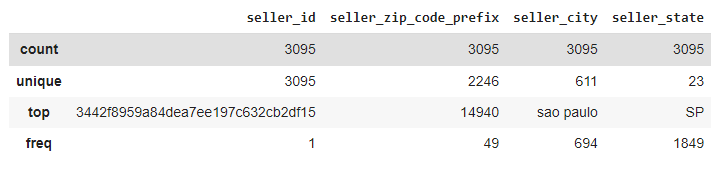
## Customer

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| customer\_id | Id của khách hàng khi thực hiện giao dịch | Categorical |
| customer\_unique\_id | Id riêng của từng khách hàng | Categorical |
| customer\_zip\_code\_prefix | Zip code của khách hàng | Categorical |
| customer\_city | Tên thành phố nơi giao dịch được thực hiện | Categorical |
| customer\_state | Mã bang nơi giao dịch được thực hiện | Categorical |



## Sellers

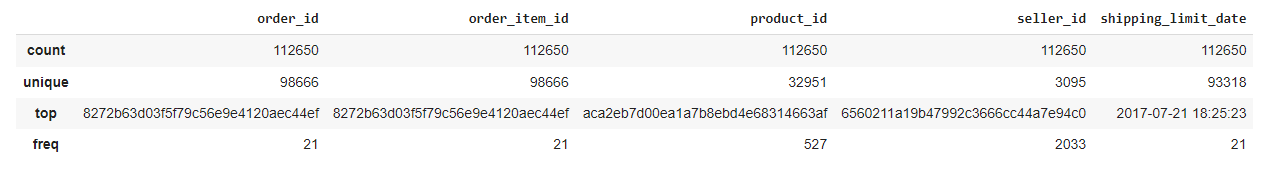
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| seller\_id | Id của người bán | Categorical |
| seller \_zip\_code\_prefix | Zip code của người bán | Categorical |
| seller\_city | Tên thành phố của người bán | Categorical |
| seller \_state | Mã thành phố của người bán | Categorical |



## Order\_items

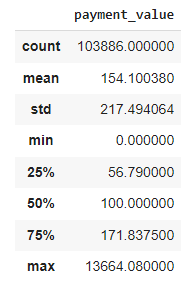
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| order\_id | Id đơn hàng | Categorical |
| order\_item\_id | Id cho từng sản phầm trong đơn hàng | Categorical |
| product\_id | Id sản phẩm | Categorical |
| seller\_id | Id của người bán | Categorical |
| shipping\_limit\_date | Hạn giao hàng | Categorical |
| price | Giá | Numerical |
| Freight\_value | Phí vận chuyển | Numerical |

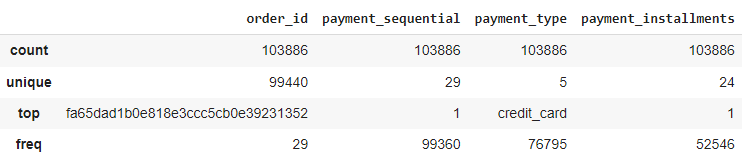




## Payments

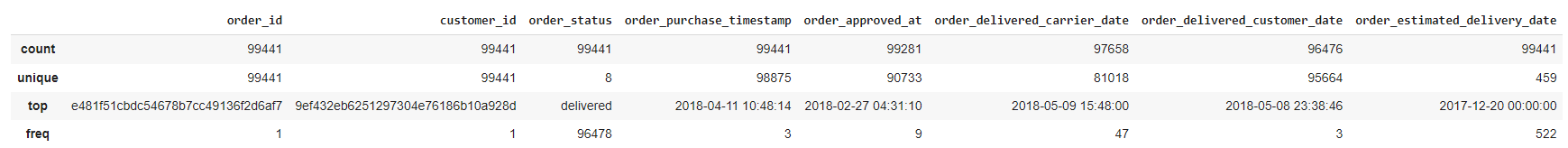
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| order\_id | Id đơn hàng | Categorical |
| payment\_sequential | ? | Categorical |
| payment\_type | Phương thức thanh toán | Categorical |
| payment\_installments | ? | Categorical |
| payment\_value | Giá trị thanh toán | Numerical |





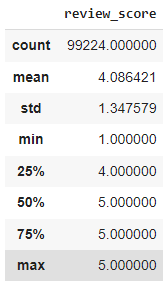
## Orders

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| order\_id | Id đơn hàng | Categorical |
| customer\_id | Id của khách hàng khi thực hiện giao dịch | Categorical |
| order\_status | Trạng thái đơn hàng | Categorical |
| order\_purchase\_timestamp | Thời gian mua | Categorical |
| order\_approved\_at | Ngày xác nhận đơn đặt hàng | Categorical |
| order\_delivered\_carrier\_date | Ngày đơn hàng được giao cho người vận chuyển | Categorical |
| order\_delivered\_customer\_date | Ngày đơn hàng được giao đến tay khách hàng | Categorical |
| order\_estimated\_delivery\_date | Ngày giao hàng dự kiến | Categorical |



## Order\_reviews

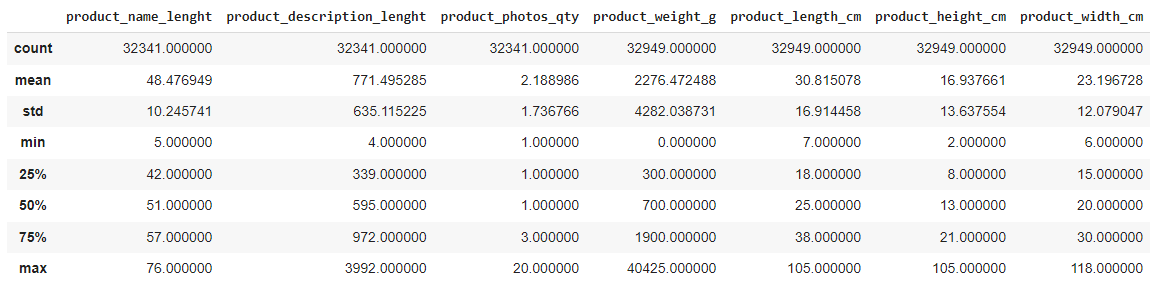
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| order\_id | Id đơn hàng | Categorical |
| review\_id | Id của đánh giá | Categorical |
| review\_score | Điểm đánh giá của khách hàng từ 1-5 | Numerical |
| review\_comment\_title | Tiêu đề của đánh giá | Categorical |
| review\_comment\_message | Nhận xét của khách hàng | Categorical |
| review\_creation\_date | Ngày đánh giá | Categorical |
| review\_answer\_timestamp | Ngày đánh giá được trả lời | Categorical |

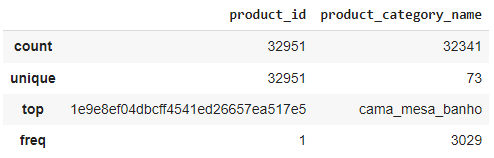




## Products

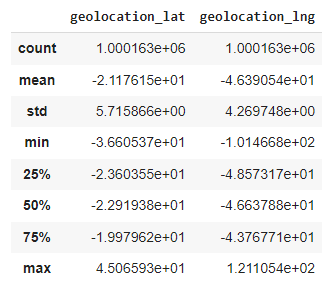
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| product\_id | Id sản phẩm | Categorical |
| product\_category\_name | Phân loại sản phẩm | Categorical |
| product\_name\_lenght | Độ dài tên sản phẩm | Numerical |
| product\_description\_lenght | Độ dài mô tả sản phẩm | Numerical |
| product\_photos\_qty | Số hình ảnh của sản phẩm trên trang bán | Numerical |
| product\_weight\_g | Khối lượng sản phẩm (gram) | Numerical |
| product\_length\_cm | Chiều dài sản phẩm (cm) | Numerical |
| product\_height\_cm | Chiều cao sản phẩm (cm) | Numerical |
| product\_width\_cm | Chiều rộng sản phẩm (cm) | Numerical |

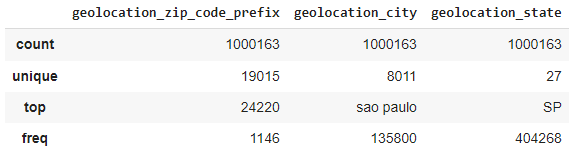




## Geolocation

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| geolocation\_zip\_code\_prefix | Zip code của vị trí | Categorical |
| geolocation\_lat | Vĩ độ | Numerical |
| geolocation\_lng | Kinh độ | Numerical |
| geolocation\_city | Tên thành phố | Categorical |
| geolocation\_state | Mã thành phố | Categorical |





# Customer Segmentation

## Ý tưởng

Dùng RFM (Recency, Frequency, Monetary) để phân cụm khách hàng, từ đó tìm ra nhóm khách hàng nên tập trung vào.

## Truy xuất dữ liệu

### Query:

with raw as

(

select t2.customer\_id, t2.order\_id, (t1.price+t1.freight\_value) as 'total\_value', t2.order\_purchase\_timestamp

from order\_items t1

inner join orders t2 on t1.order\_id = t2.order\_id

where t2.order\_status = 'delivered'

)

select c.customer\_unique\_id, r.order\_id, r.total\_value, r.order\_purchase\_timestamp

from raw r inner join customer c on r.customer\_id = c.customer\_id

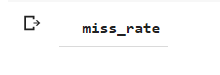
### Giải thích:

Tạo bảng tạm raw từ order\_items bao gồm tính tổng giá trị đơn hàng ‘total\_value’(price + freight\_value) join với orders để tìm ra các đơn hàng có trạng thái là ‘delivered’. Cuối cùng lấy raw join với customer để lấy ra customer\_unique\_id.

## Phân tích dữ liệu

### Tiền xử lý dữ liệu:

Đầu tiên là tìm và giải quyết những giá trị bị khuyết (missing value), vì sử dụng inner join để truy vấn nên hạn chế được phần lớn giá trị null. Kết quả trả về của bộ dữ liệu này là không có missing data.



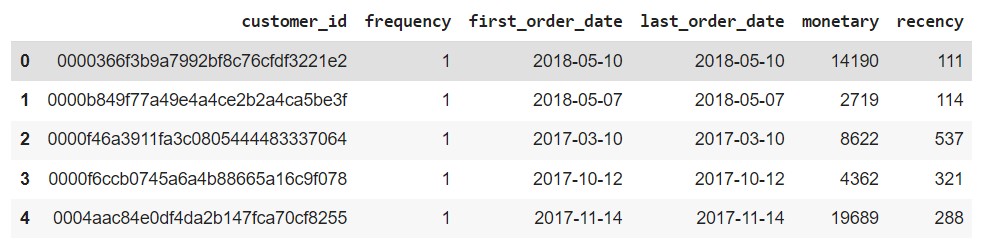
Sau đó, chuyển kiểu dữ liệu về format mong muốn và tính RFM. Với:

+ R: recency = ngày được chọn (ngày phân tích/ ngày hôm nay) – ngày mua hàng gần nhất của khách hàng. Trong bài này, tôi sẽ tính bằng ngày mua hàng cuối cùng (lớn nhất) được ghi lại trong bộ dữ liệu.

+ F: frequency = đếm số lần mua hàng của khách hàng.

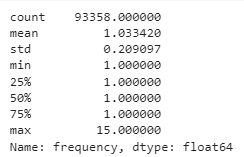
+M: monetary = tổng giá trị tất cả đơn hàng của khách hàng.

Dữ liệu thu được là:

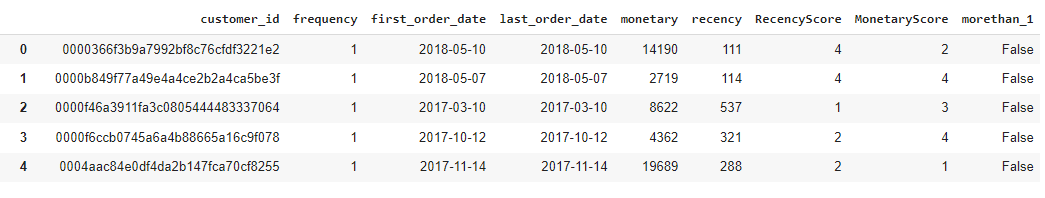


### RFM Score

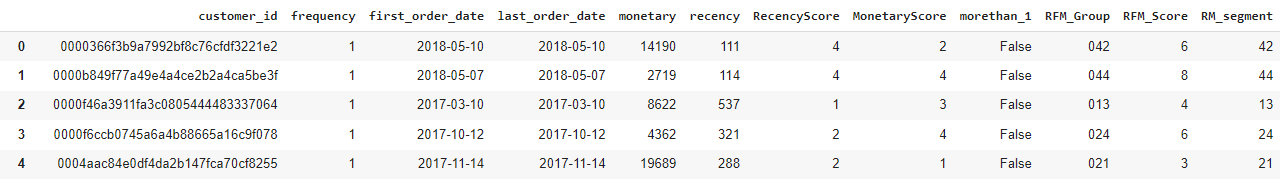
Chia các chỉ số R (Recency), M (Monetary) theo 4 nhãn. Riêng chỉ số F (Frequency) vì có tới 75% giá trị đều là 1 nên chỉ thực hiện chia theo 2 nhãn (1 với khách hàng mua trên 1 lần, 0 với khách hàng mua chỉ 1 lần).



Kết quả thu được là

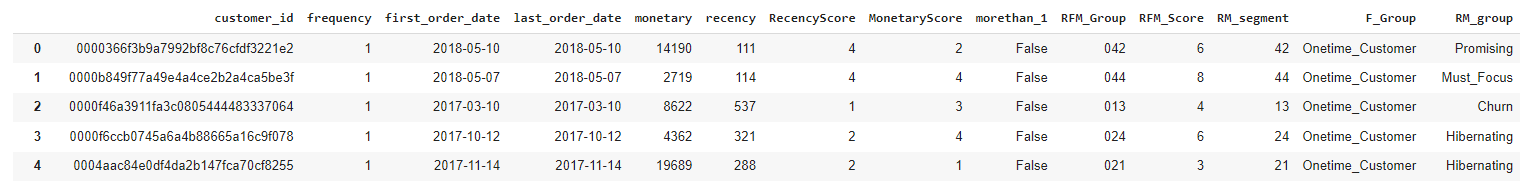


Dựa trên các score đã tính được, tính RFM score bằng cách nối các score lại với nhau.



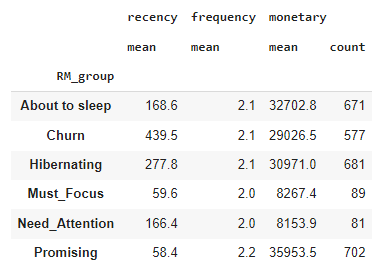
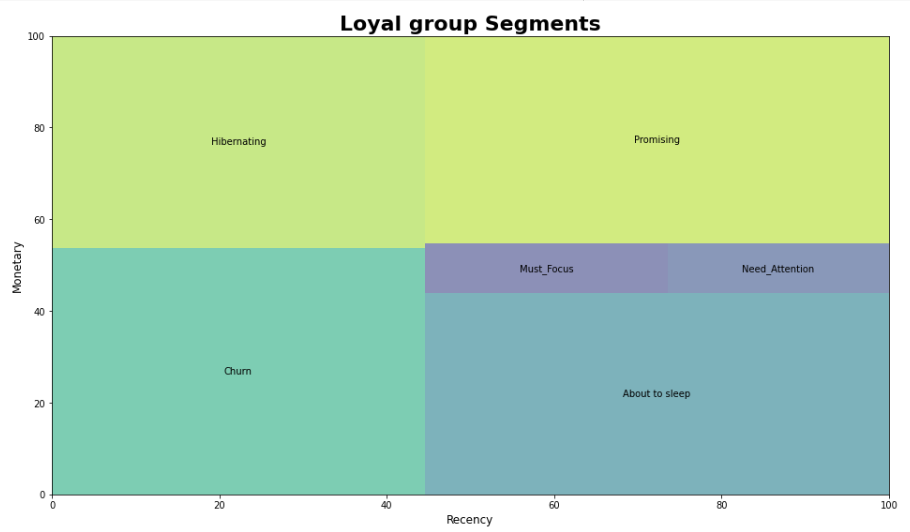
Gắn nhãn khách hàng dựa trên:

* F :khách hàng trung thành (loyal) hay (khách mua một lần) – onetime customer
* RM score: được gắn theo 6 nhãn
  + Recency = 4 và Monetary = 3-4:'Must\_Focus' (nên tập trung vào)
  + Recency = 4 và Monetary = 1-2:'Promising' (tiềm năng)
  + Recency = 3 và Monetary = 3-4:'Need\_Attention' (cần hành động)
  + Recency = 3 và Monetary = 1-2: 'About to sleep' (Sắp ngừng hoạt động)
  + Recency = 2 và Monetary = 1-4: 'Hibernating' (đã ngừng hoạt động một thời gian)
  + Recency = 1 và Monetary = 1-4:'Churn' (đã rời bỏ)

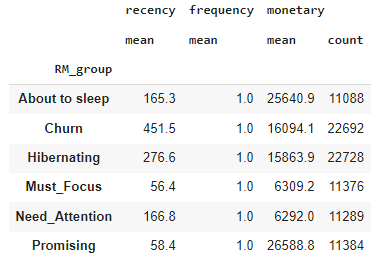
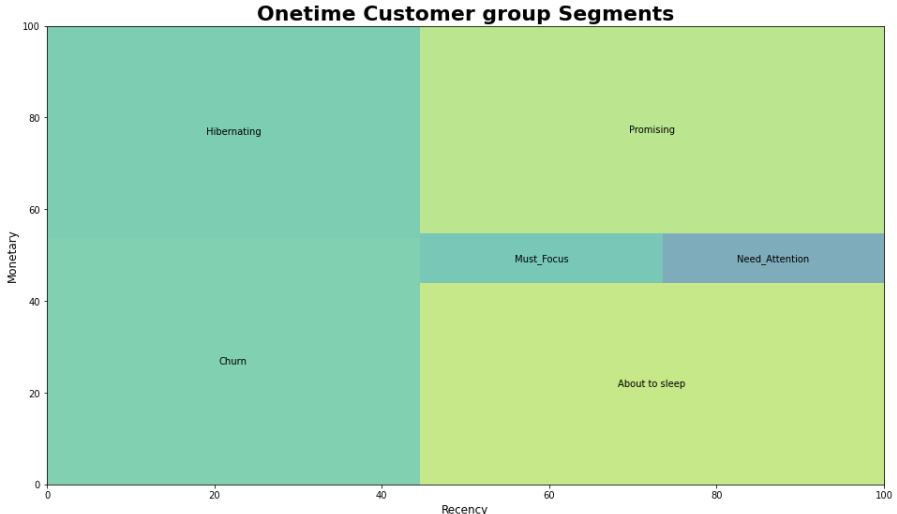


Trực quan hóa RM theo loyal và onetime

* Loyal

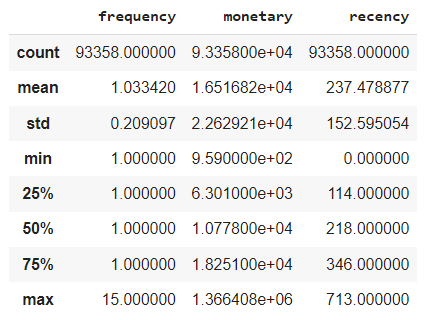
 

* Onetime

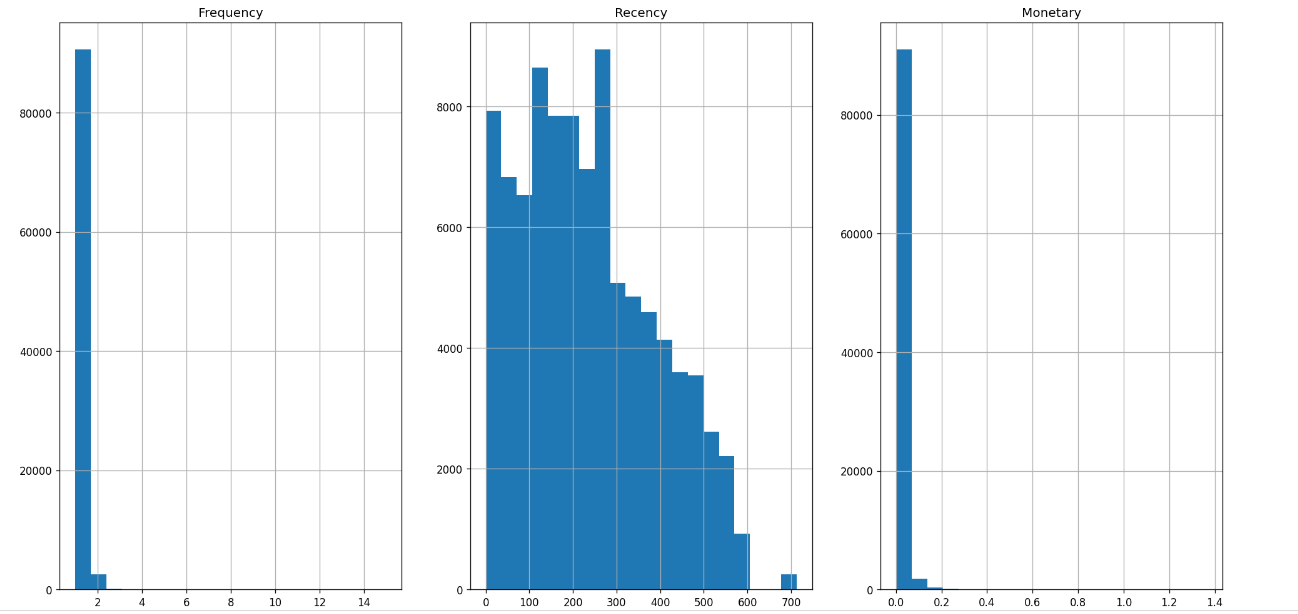
 

### EDA

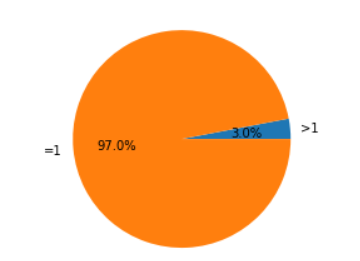
Thực hiện thống kê mô tả (Descriptive Analysis) trên các cột RFM.



Tìm hiểu về phân phối của RFM.

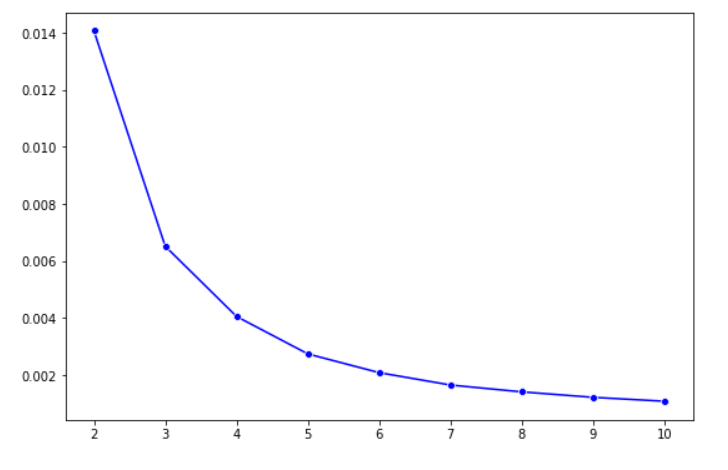


Dựa trên phân phối trên, ta dễ thấy rằng khách hàng thường chỉ mua một lần và không quay lại.



Có tới 97% khách hàng rời bỏ hệ thống chỉ sau một lần mua hàng. Để tìm hiểu kĩ hơn về nguyên nhân của vấn đề này, tôi sẽ thực hiện ở những phần sau.

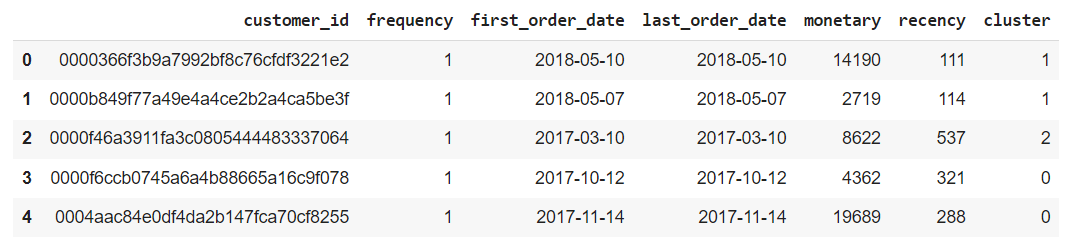
Cuối cùng, trước khi phân cụm, ta cần tìm số cụm tối ưu cho bài toán phân cụm (k-optimal).



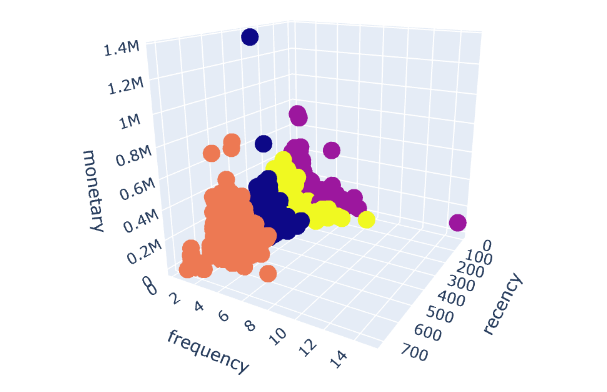
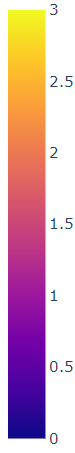
Số cụm tối ưu ở bài toán này (được tính theo wssd) có thể là 4 hoặc 5, ở bài toán này tôi sẽ chọn k = 4.

### Phân cụm khách hàng:

Thực hiện phân cụm với số cụm bằng 4 ta thu được

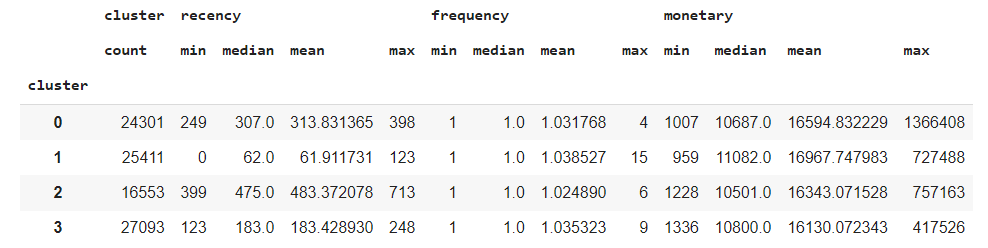


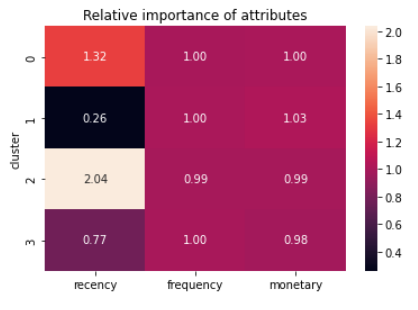
Để dễ giao tiếp với các bên liên quan hơn, ta cần trực quan hóa dữ liệu theo các cụm đã tìm được. Ở đây, tôi sử dụng thư viện plotly.express để tạo mô hình 3D theo 3 chiều dữ liệu RFM.

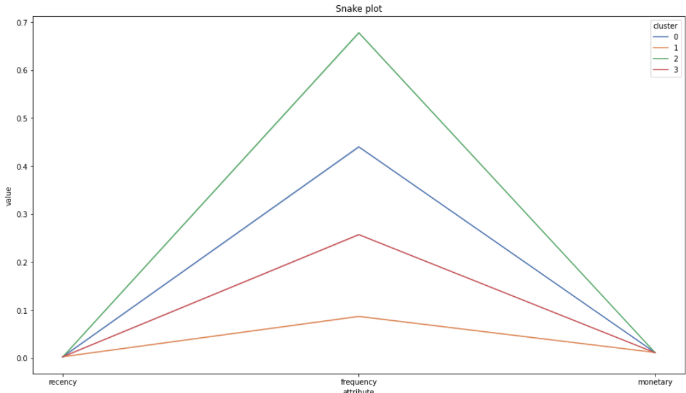
 

Ta đã có các cụm dữ liệu nhưng chưa biết tương quan giữa các chiều của nó, nên tiếp theo ta sẽ thực hiện phân tích hậu định để tìm ra tương quan giữa các chiều dữ liệu theo cụm.

Thống kê mô tả theo các cụm.



Tương quan giữa các chiều thuộc tính theo cụm

Bốn cụm có cả monetary và recency gần như giống nhau.

frequency:

+ 0: trung bình

+ 1: cực thấp

+ 2: cao (số lần mua hàng)

+ 3: tương đối thấp

Tất cả các cụm đều có tầm quan trọng tương đối với frequency và monetary gần như giống nhau:

recency:

+ 0: trung bình

+ 1: cực thấp

+ 2: cao

+ 3: tương đối thấp

## Kết luận

Tóm lại, khách hàng thuộc nhóm 2 dường như mua hàng nhiều lần nhất so với các nhóm khác nhưng lần mua cuối cùng của họ đã quá lâu khiến tôi khẳng định rằng họ không còn quan tâm đến nền tảng của chúng ta nữa. Cụm 0 chỉ có một chút khác biệt so với cụm 2, vì vậy chúng tôi sẽ không tập trung vào cụm này quá. Chúng ta không thể quyết định dựa trên tần suất vì dữ liệu cho thấy hầu hết khách hàng chỉ mua hàng một lần. Loại bỏ yếu tố tần suất thì thu hút cụm 1,3 dường như hợp lý nhất vì họ vừa mới mua hàng gần đây.

# Retention Cohort

## Ý tưởng

Sau khi thực hiện phân cụm, dữ liệu cho thấy số lần mua hàng của khách hàng (frequency) thưởng chỉ là 1 lần, các giá trị lớn hơn 1 đều được gán là giá trị ngoại biên (outliners). Nên ta tiến hành xây dựng metric Retention Cohort để xem tỷ lệ quay lại của khách hàng sau lần mua hàng đầu tiên cũng như tỷ lệ rời bỏ (Churn Rate) của khách hàng.

## Truy xuất dữ liệu

Dữ liệu để xây dựng metric Retention Cohort giống với dữ liệu dùng để phân cụm khách hàng (Customer Segmentation)

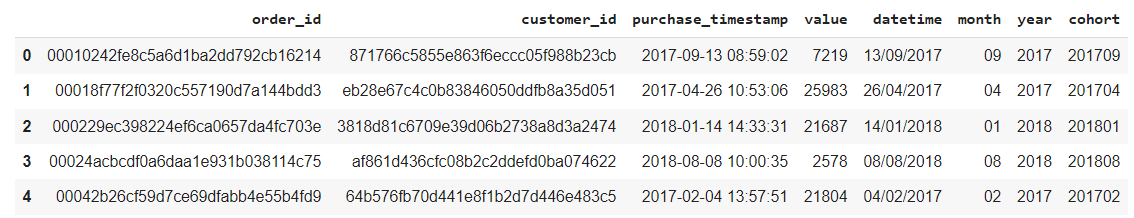
## Phân tích dữ liệu

### Tiền xử lý dữ liệu

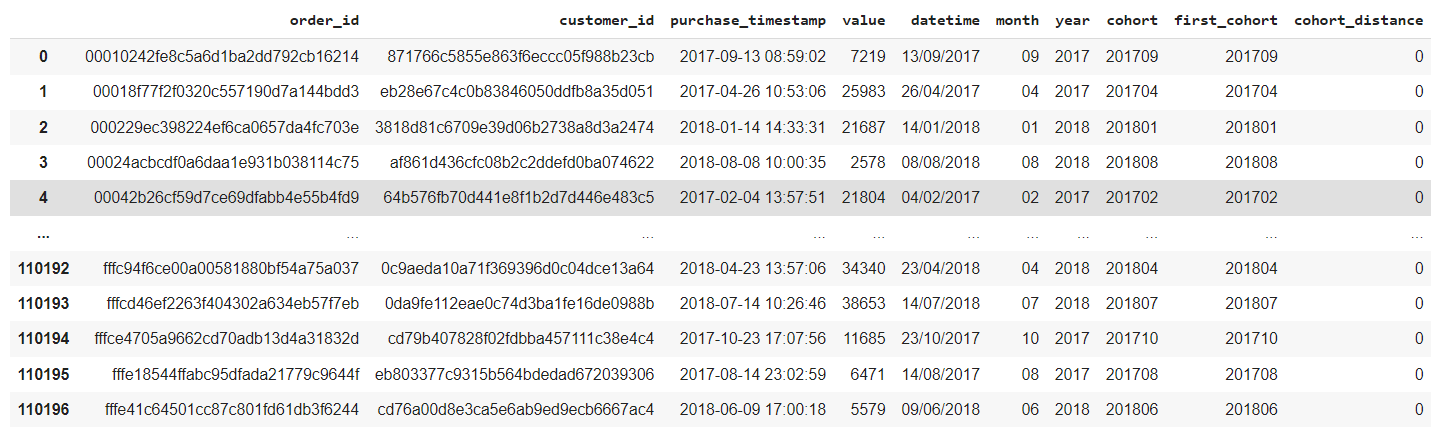
Bước đầu tiên trong tiền xử lý dữ liệu là kiểm tra và xử lý các giá trị bị khuyết. Trong bộ dữ liệu của bài này, kết quả trả về là không có dữ liệu nào bị khuyết.



Tiếp theo, ta cần chuyển các cột dữ liệu ngày về đúng format và chiết tách tháng, năm và tính cohort = năm\*100 + tháng (việc tính toán này chỉ để thuận tiện cho việc sắp xếp các tháng, năm khi thực hiện pivot table và tính ra cohort distance chứ không mang ý nghĩa về mặt toán học).

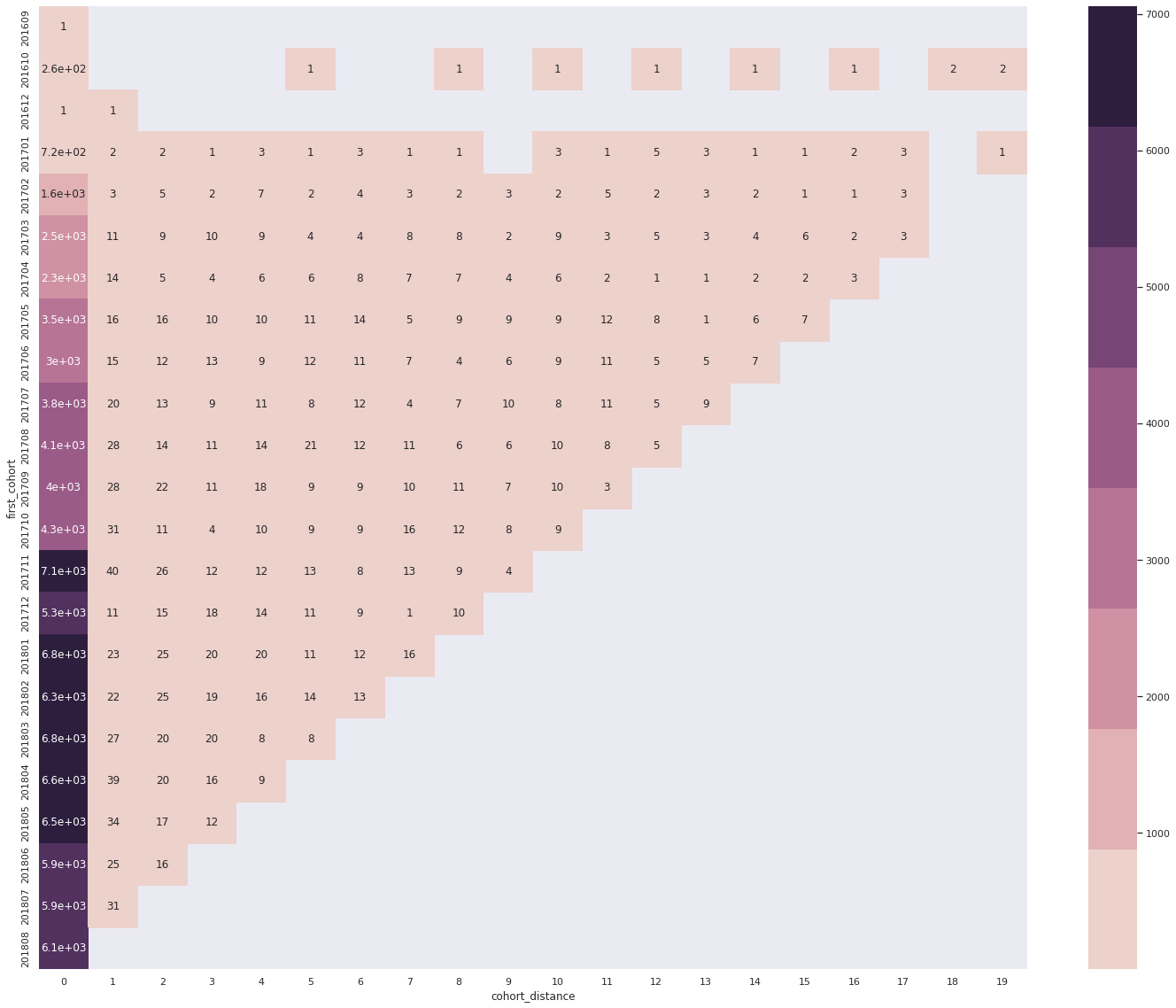


Tìm ra ngày mua hàng đầu tiên của khách hàng (first\_purchase) bằng min cohort sau đó join lại với data gốc để tìm cohort distance. Cohort distance = cohort - first\_purchase.

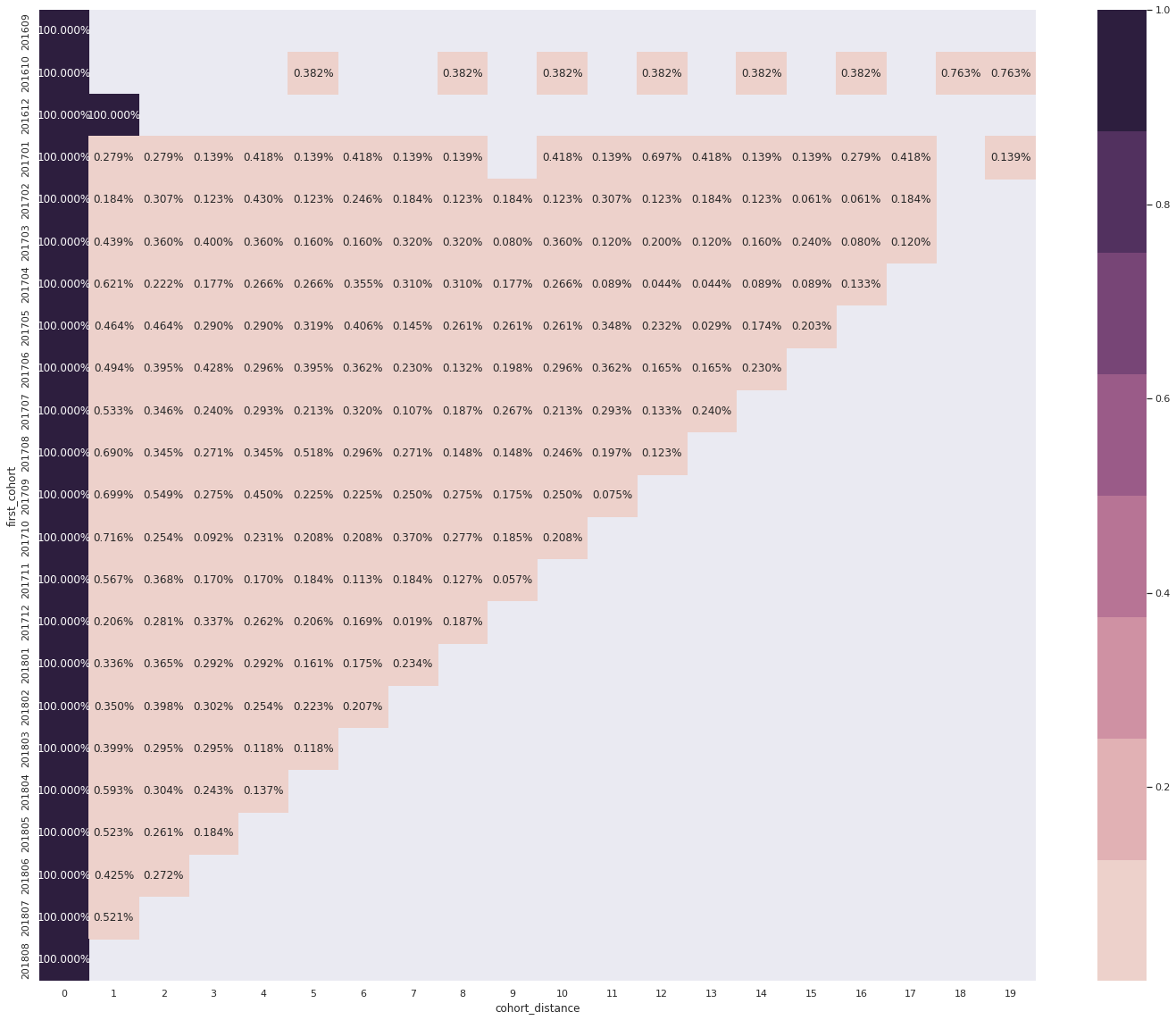


### Phân tích Cohort (Cohort Analysis)

Sau khi có cohort distance dùng pivot table và heatmap để trực quan hóa.



Chia các cột cohort distance khác 0 cho cột đầu tiên (cohort distance bằng 0) để ra được retention rate.



## Kết luận

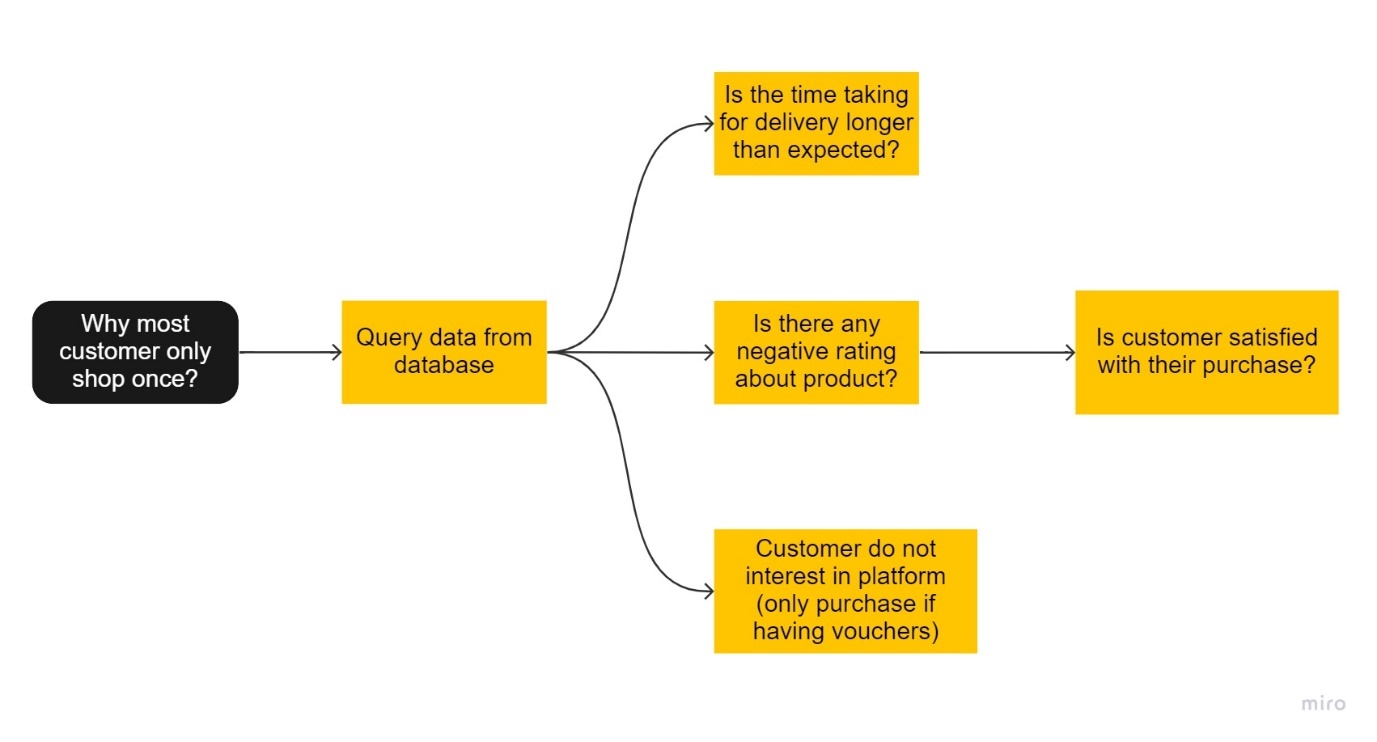
Khách mua hàng rất nhiều nhưng chủ yếu chỉ mua một lần, khách mua một lần chiếm đến 97% tổng lượng khách. Lượng khách quay lại có xu hướng giảm dần theo thời gian. Ở phần tiếp theo, tôi sẽ tìm hiểu nguyên nhân tại sao dẫn đến việc khách chỉ mua một lần và không quay lại.

# Root Cause Analysis

## Ý tưởng

Thực hiện phân tích chuyên sâu các yếu tố liên quan để tìm ra nguyên nhân tại sao phần lớn khách thưởng chỉ mua hàng một lần và không quay lại. Các yếu tố tôi quan tâm và có thể đo lường được ở bộ dữ liệu này bao gồm: Giao hàng (Shipment), Đánh giá (Rating), Giao dịch (Payment)

Các bước suy luận và thực hiện bao gồm:



## Truy xuất dữ liệu

### Shipment

#### Query:

select o.order\_id, o.order\_status, o.order\_purchase\_timestamp,

datediff(day,o.order\_purchase\_timestamp,o.order\_approved\_at) as 'aproved\_after',

datediff(day, o.order\_approved\_at, o.order\_delivered\_carrier\_date) as 'carrier\_take\_after',

datediff(day, o.order\_delivered\_carrier\_date, o.order\_delivered\_customer\_date) as 'delivered\_after',

datediff(day, o.order\_purchase\_timestamp, o.order\_delivered\_customer\_date) as 'total\_delivery\_time',

datediff(day, o.order\_purchase\_timestamp, o.order\_estimated\_delivery\_date) as 'estimated\_delivery\_time'

from orders o

#### Giải thích:

Từ bảng orders dùng hàm datediff với interval là day để tinh ra số ngày chênh lệnh giữa các columns daytime trong bảng.

### Rating

#### Query:

select p.review\_id, pd.product\_id, t.product\_category\_name\_english, p.review\_comment\_title, p.review\_comment\_message, p.review\_score

from order\_previews p

left join order\_items i on p.order\_id = i.order\_id

left join products pd on i.product\_id = pd.product\_id

inner join product\_category\_name\_translation t on pd.product\_category\_name = t.product\_category\_name

#### Giải thích:

Dùng order\_previews để lấy ra review\_id, title, message, score. Join với order\_items để lấy ra product\_id sau đó join với product\_category\_name\_translation để tìm ra category.

### Payment

#### Query:

select p.order\_id , p.payment\_type ,p.payment\_value

from payments p

#### Giải thích:

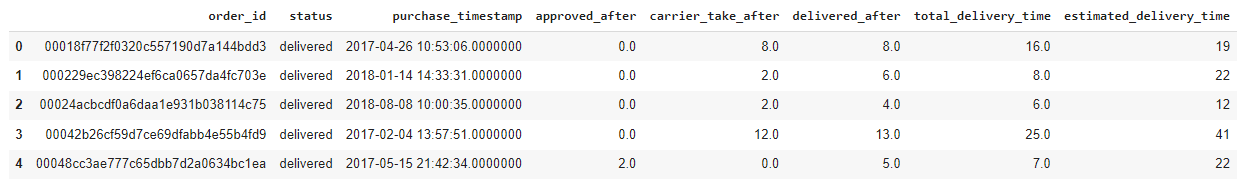
Vì không cần dùng hết tất cả các thuộc tính trong bảng payments nên không sử dụng select \*

## Phân tích dữ liệu

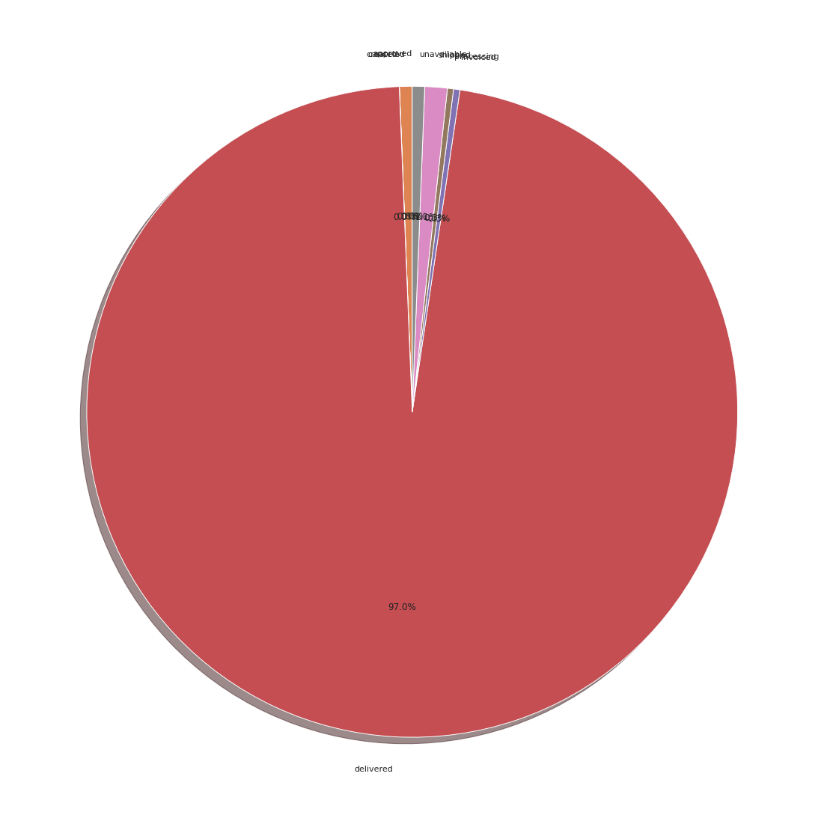
Phần lớn các phương pháp được sử dụng là EDA và kiểm định giả thuyết để đưa ra kết luận.

### Shipment

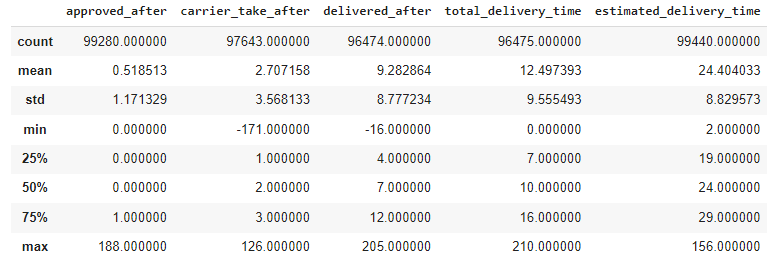
Các cột dữ liệu mới phần lớn đã được tính bằng SQL thông qua hàm datediff khi truy xuất dữ liệu.



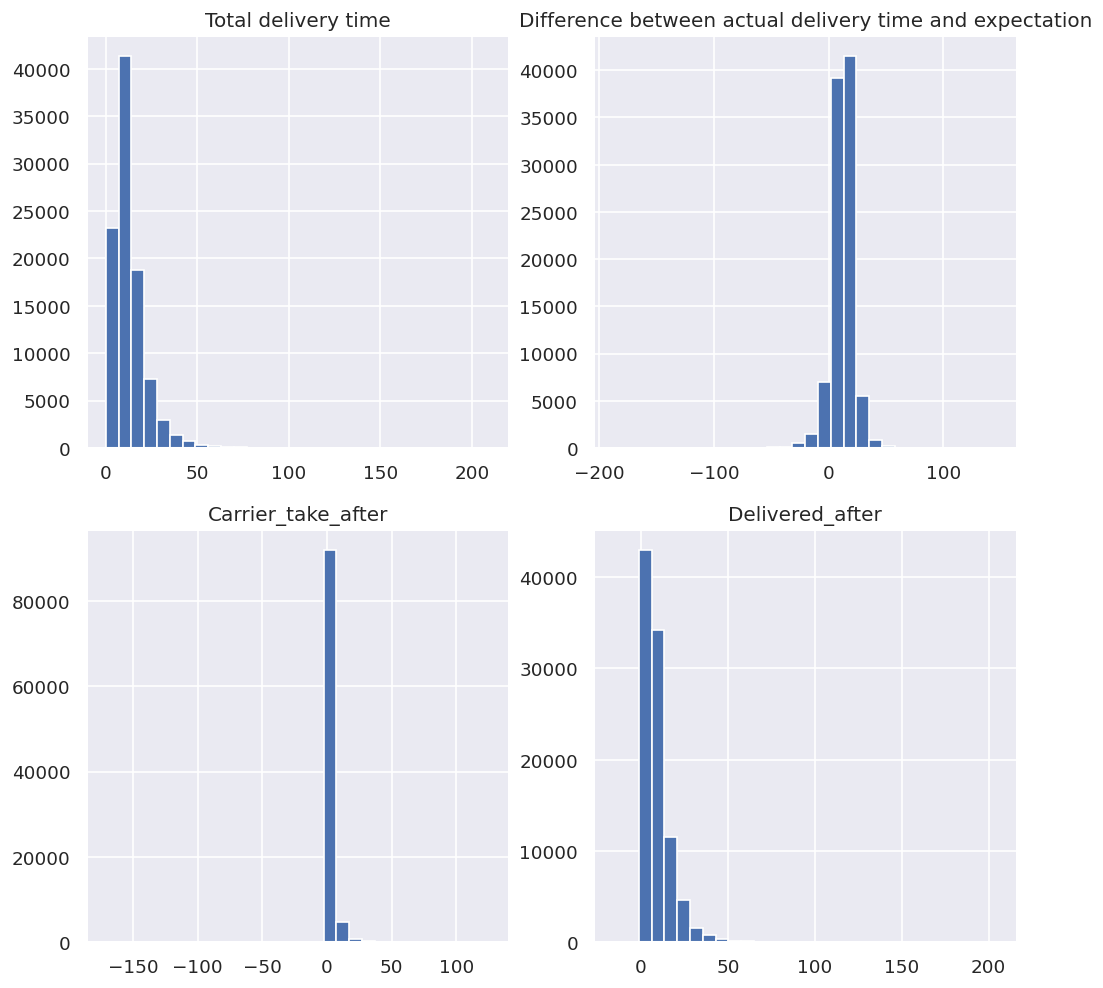
Xem xét tỉ lệ giao hàng thành công. Có tới 97% đơn hàng đã được giao thành công

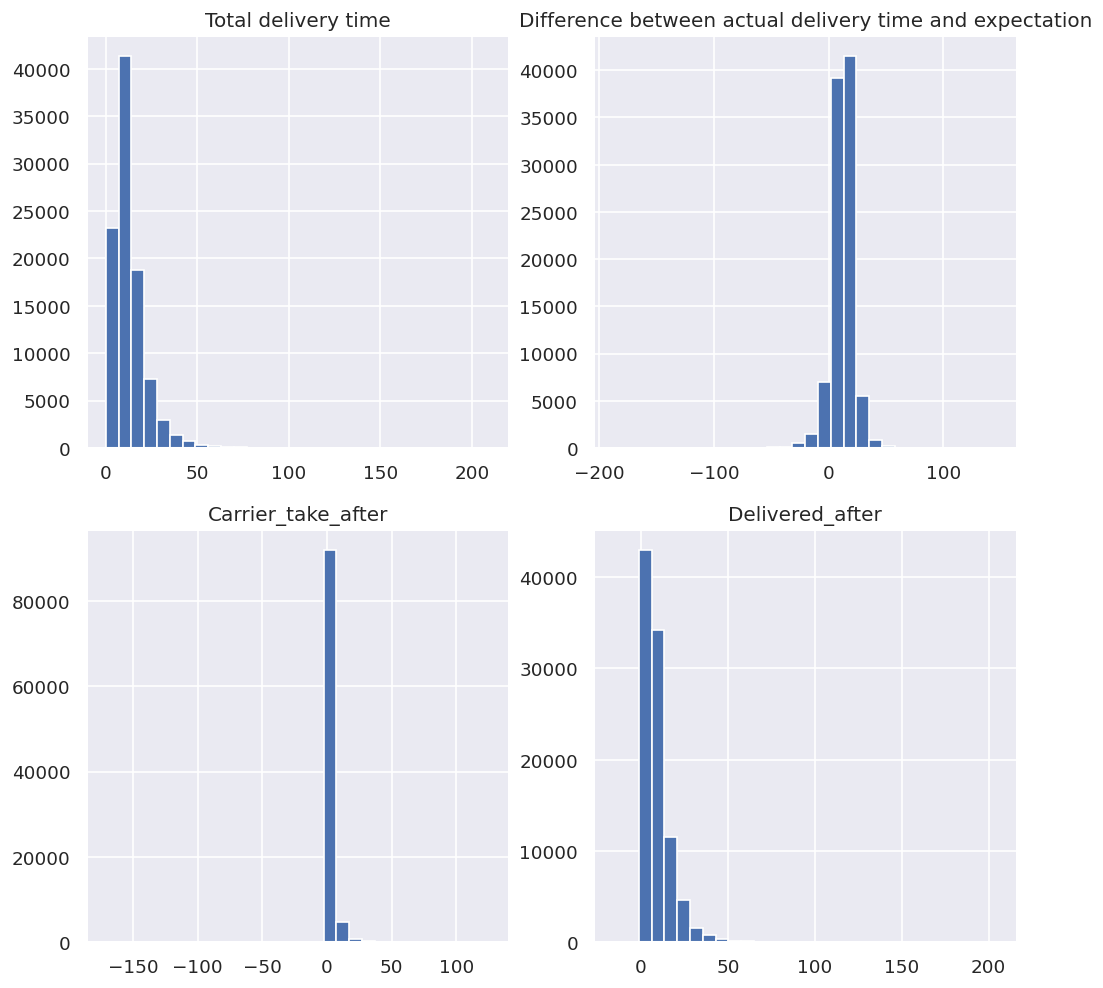


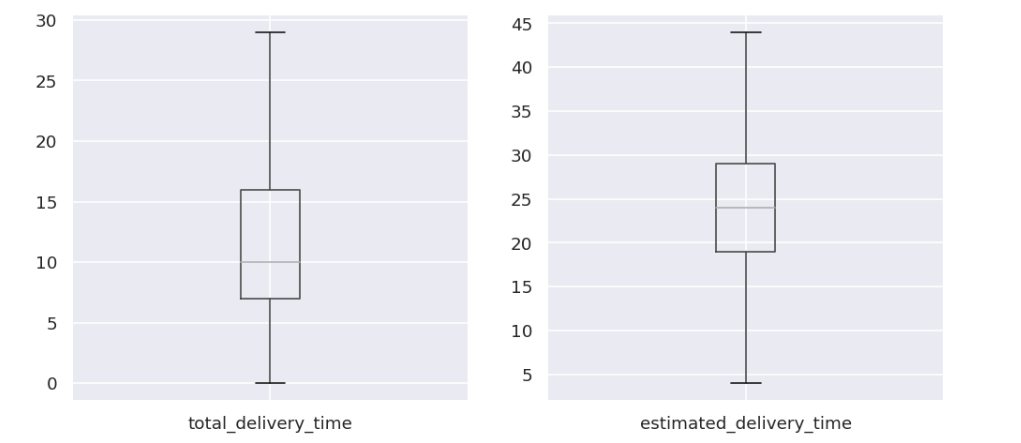
Thực hiện thống kê mô tả ở các cột dữ liệu dạng số.



Kết quả cho thấy, trung bình tổng ngày giao hàng nhỏ hơn số ngày ước tính. Điều này có thực sự đúng? Trước khi trả lời câu hỏi trên, ta hãy xem phân phối của tổng ngày giao hàng (total\_delivery\_time), số ngày cần để người bán gửi hàng cho đơn vị vận chuyển (carrier\_take\_after), số ngày đơn vi vận chuyển giao hàng đến khách hàng (delivered\_after), chênh lệch giữa ngày giao hàng thực (diff). Diff nhỏ hơn 0 nghĩa là thời gian giao hàng thực tế lâu hơn dự kiến.







Biểu đồ hộp cho thấy thời gian giao hàng thực tế lớn hơn dự kiến. Thực hiện kiểm định xem liệu thời gian giao hàng thực tế có thực sự lâu hơn dự kiến không?

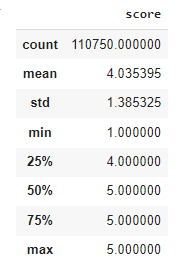
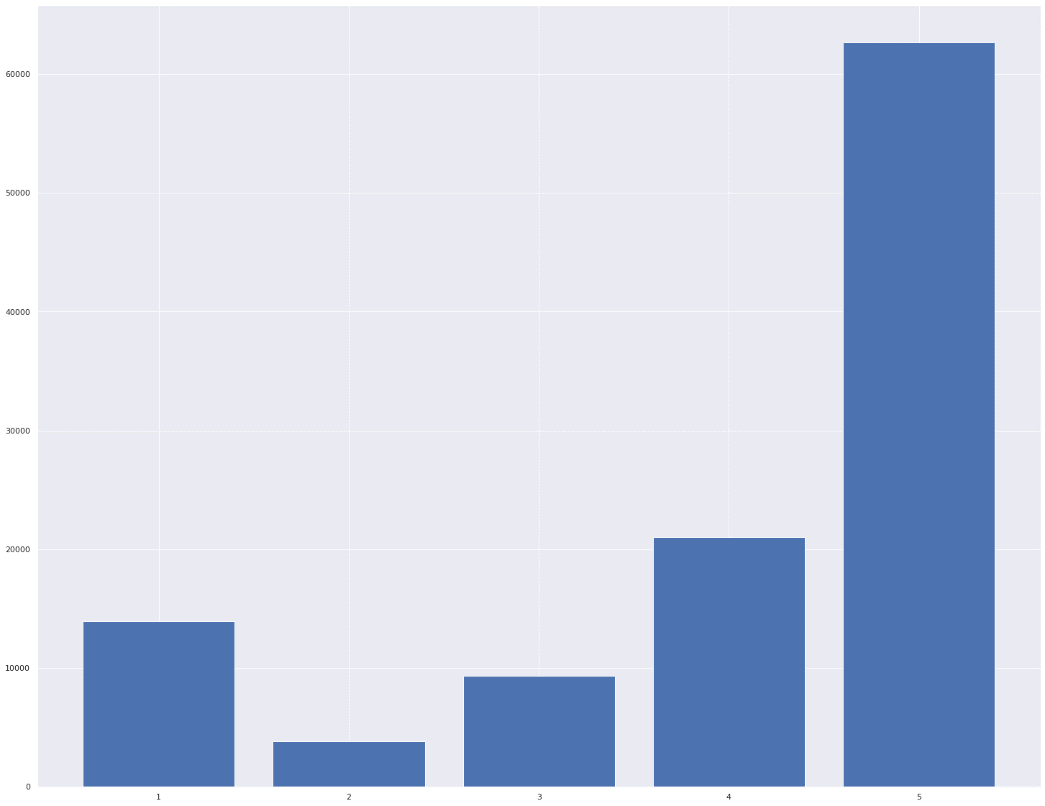
* Giả thuyết không - H0: thực tế <= dự kiến
* Giả thuyết đối – Ha: thực tế > dự kiến



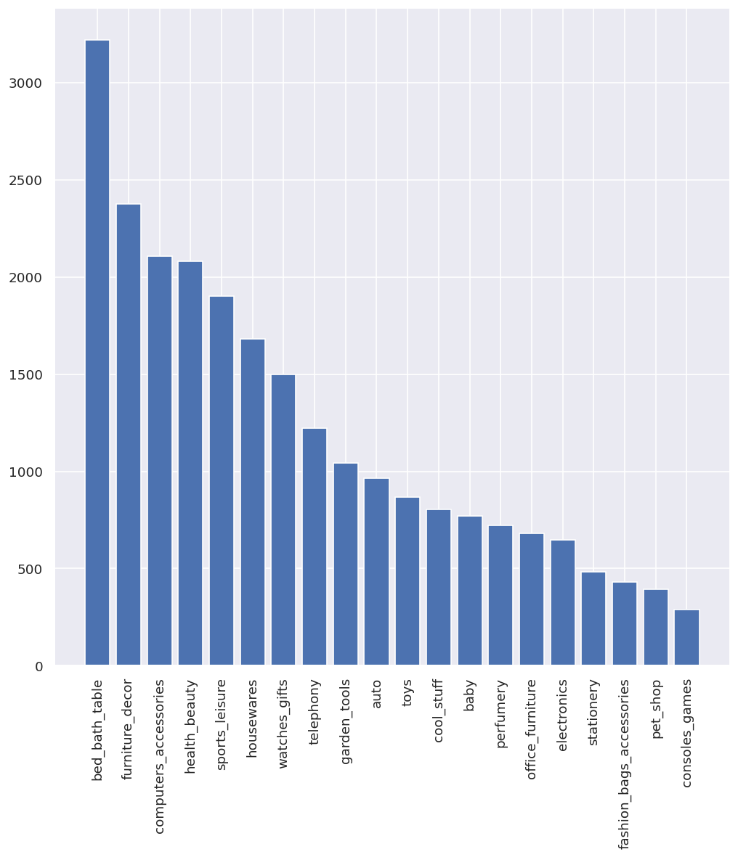
Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta bác bỏ H0. Ta có thể kết luận rằng thời gian giao hàng thực tế lớn hơn dự kiến. Đây là một trong những nguyên nhân dẫn đến việc khách rời bỏ hệ thống.

### Rating

Thực hiện thống kê mô tả dựa trên điểm đánh giá. Điểm trung bình nhận được khá cao, hơn 4 và có một nửa đánh giá là 5 điểm.



Tìm ra những sản phẩm bị đánh giá thấp.



Thực hiện kiểm định chi-square xem liệu hệ thống đã làm hài lòng nhiều hơn 80% đơn hàng chưa.

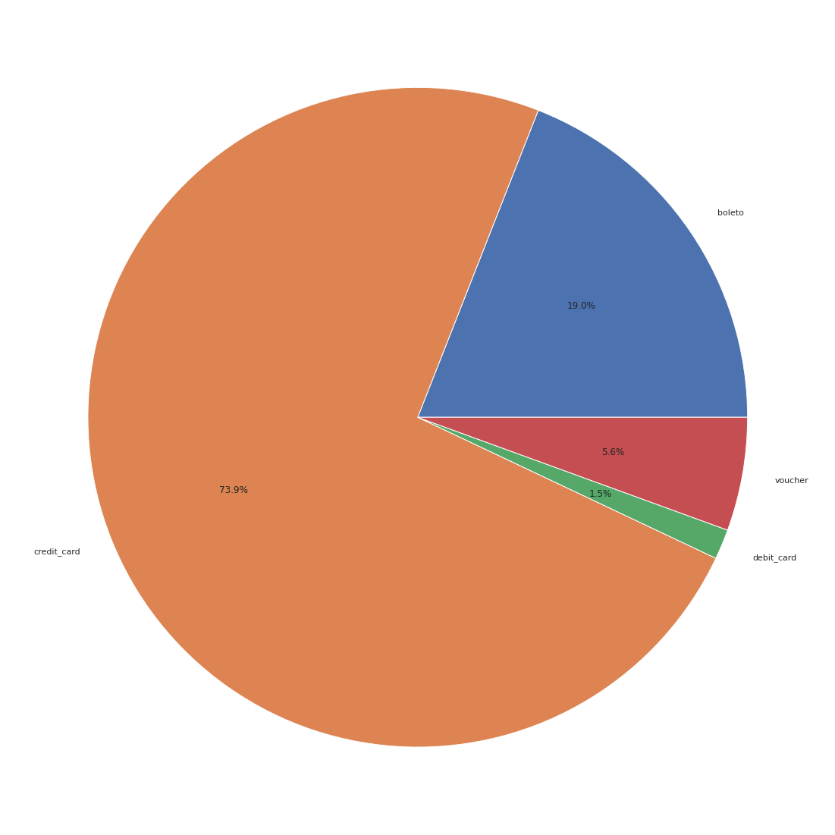


Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta bác bỏ H0. Ta có thể kết luận rằng tuy lượng đánh giá cao nhiều, nhưng với mục tiêu làm hài lòng hơn 80% đơn hàng thì hệ thống đã không đạt được. Các mặt hàng cần được điểm định lại là bed\_bath\_table, home\_decor,… vì nhận nhiều đánh giá không tốt.

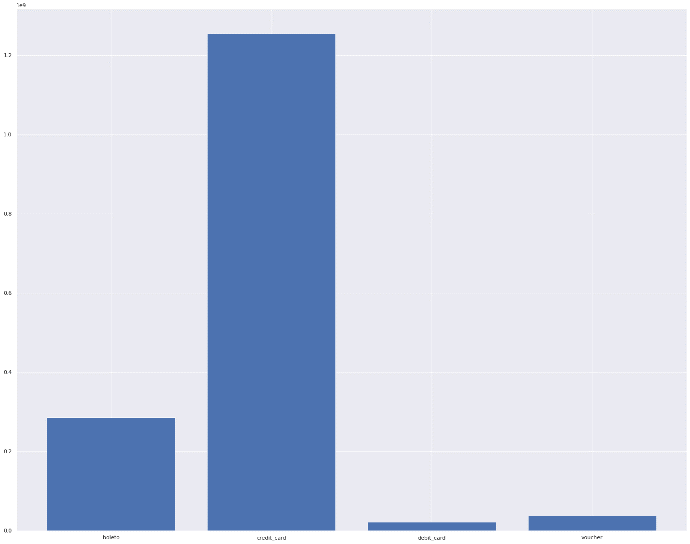
### Payment

Đặt ra giả thuyết là những khách hàng không hứng thú với hệ thống thường chỉ mua hàng khi được tặng các voucher khuyến mãi. Dùng dữ liệu payment để kiểm định xem liệu khách đến mua hàng có thực sự quan tâm đến ứng dụng hay không?

Tỷ lệ thanh toán theo hình thức credit\_card là chủ yếu, voucher chỉ chiếm 5.6% xếp hạng 3 /4 hình thức thanh toán phổ biến trên nền tảng.



Tổng giá trị thanh toán ghi nhận được theo hình thức tỷ lệ thuận với số giao dịch theo hình thức.



Cuối cùng, ta kiểm định xem giá trị của từng đơn hàng được thanh toán bằng voucher có lớn hơn so với các hình thức còn lại không?

* Giả thuyết không – H0: voucher >= boleto
* Giả thuyết đối – Ha: voucher < boleto



Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta không đủ cơ sở bác bỏ H0. Giá trị đơn hàng thanh toán bằng voucher có thể lớn hơn boleto.

* Giả thuyết không – H0: voucher >=  credit\_card
* Giả thuyết đối – Ha: voucher < credit\_card



Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta không đủ cơ sở bác bỏ H0. Giá trị đơn hàng thanh toán bằng voucher có thể lớn hơn credit\_card.

* Giả thuyết không – H0: voucher >=  debit\_card
* Giả thuyết đối – Ha: voucher < debit\_card



Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta không đủ cơ sở bác bỏ H0. Giá trị đơn hàng thanh toán bằng voucher có thể lớn hơn dedit\_card.

## Kết luận

Chúng ta có thể tuyên bố rằng thời gian giao hàng trễ là nguyên nhân chính dẫn đến hầu hết khách hàng chỉ mua một lần. Bên cạnh đó, nếu mục tiêu của chúng ta là hơn 80% khách hàng hài lòng với việc mua hàng của họ, thì dữ liệu cho thấy mục tiêu đó gần như không đạt. Cuối cùng, dữ liệu cho thấy có thể tồn tại những nhóm khách hàng không quan tâm đến nền tảng của chúng ta vì tỷ lệ phần trăm sử dụng phiếu thưởng không quá nhiều nhưng giả thuyết cho thấy giá trị của một phiếu thưởng dùng để mua hàng cao hơn các phương thức thanh toán khác.

# Market Basket Analysis

## Ý tưởng

Một trong những ý tưởng tôi đưa ra để thu hút khách hàng là khuyến nghị những món hàng thường được mua chung với nhau. Ví dụ, người mua quần áo cho trẻ em thì thường mua thêm sữa,…Ở bài toán này, trước hết tôi sẽ dùng luật kết hợp (Association Rule) để tìm ra quy luật giữa các loại hàng. Sau đó chọn ra các cặp loại hàng tốt nhất để tìm ra quy luật giữa các sản phẩm của mặt hàng đó.

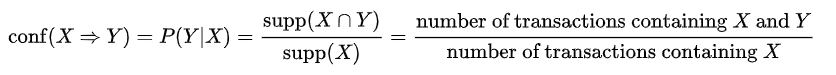
## Luật kết hợp

Luật kết hợp là mối quan hệ giữa các tập thuộc tính trong cơ sở dữ liệu. Luật kết hợp là phương tiện hữu ích để khám phá các mối liên kết trong dữ liệu.  
Một luật kết hợp là một mệnh đề kéo theo có dạng X -> Y, trong đó X, Y ⊆ I, thỏa mãn điều kiện X giao Y = rỗng. Các tập hợp X và Y được gọi là các tập hợp thuộc tính (itemset). Tập X gọi là nguyên nhân, tập Y gọi là hệ quả. Có 2 độ đo quan trọng đối với luật kết hợp: Độ hỗ trợ (support) và độ tin cậy (confidence), được định nghĩa như phần dưới đây.

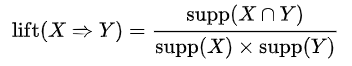
* Độ hỗ trợ: Là tần suất tập hợp thuộc tính (itemset) xuất hiện trong tập dữ liệu.



* Độ tin cậy: Độ tin cậy là tỷ lệ phần trăm của tất cả các giao dịch thỏa mãn X cũng đáp ứng Y.



Một trong những kết quả trả về của luật kết hợp mà tôi quan tâm ở bài toán này là Lift.



* Nếu Lift > 1, điều đó cho chúng tôi biết mức độ mà hai lần xuất hiện đó phụ thuộc vào nhau và làm cho các quy tắc đó có khả năng hữu ích để dự đoán hậu quả trong các tập dữ liệu trong tương lai.
* Nếu Lift < 1, điều đó cho chúng tôi biết các vật phẩm được thay thế cho nhau. Điều này có nghĩa là sự hiện diện của một mặt hàng có ảnh hưởng tiêu cực đến sự hiện diện của mặt hàng khác và ngược lại.

## Truy xuất dữ liệu

### Query:

select o.order\_id, o.product\_id,o.order\_item\_id, p1.product\_category\_name\_english

from order\_items o

left join products p on o.product\_id = p.product\_id

left join product\_category\_name\_translation p1 on p.product\_category\_name = p1.product\_category\_name

### Giải thích:

Dùng bảng order\_items để join với products lấy ra product\_id và category, join tiếp products với product\_category\_name\_translation để lấy ra tên tiếng Anh của các category (product\_category\_name\_english)

## Phân tích dữ liệu

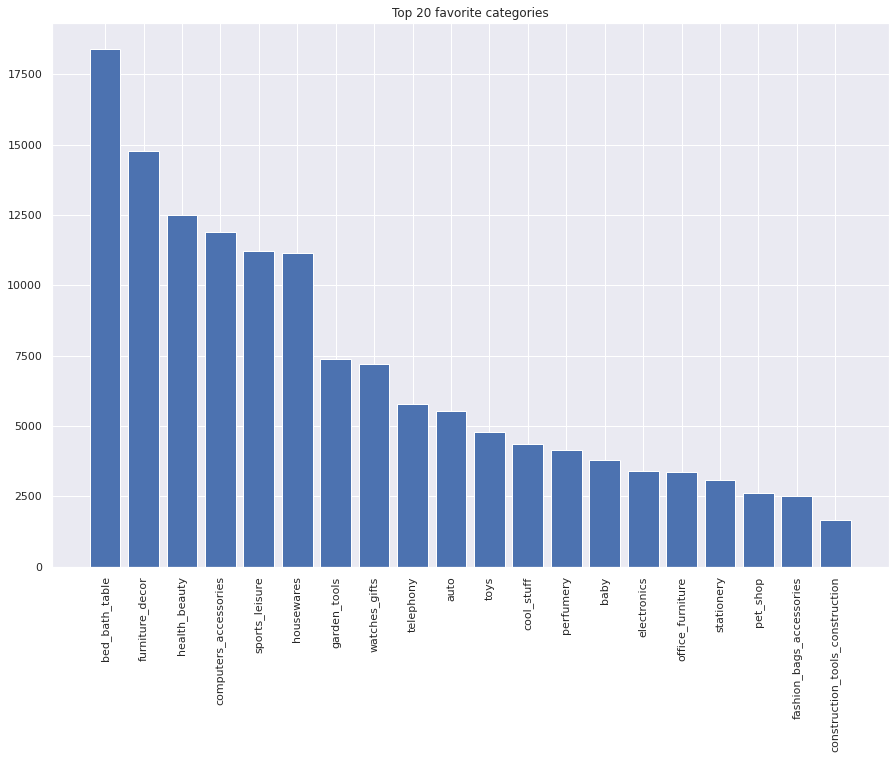
### Tiền xử lý dữ liệu

Đầu tiên ta tìm và xử lý các dữ liệu bị khuyết. Ở bài toán này có cột category bị khuyết dữ liệu nhưng không nhiều, khoảng 1.4% nằm trong ngưỡng cho phép của tôi (khoảng 5%) nên ta không xóa cột dữ liệu này. Tôi cũng sẽ không xóa dòng có dữ liệu bị khuyết mà sẽ thay vào đó giá trị mode.

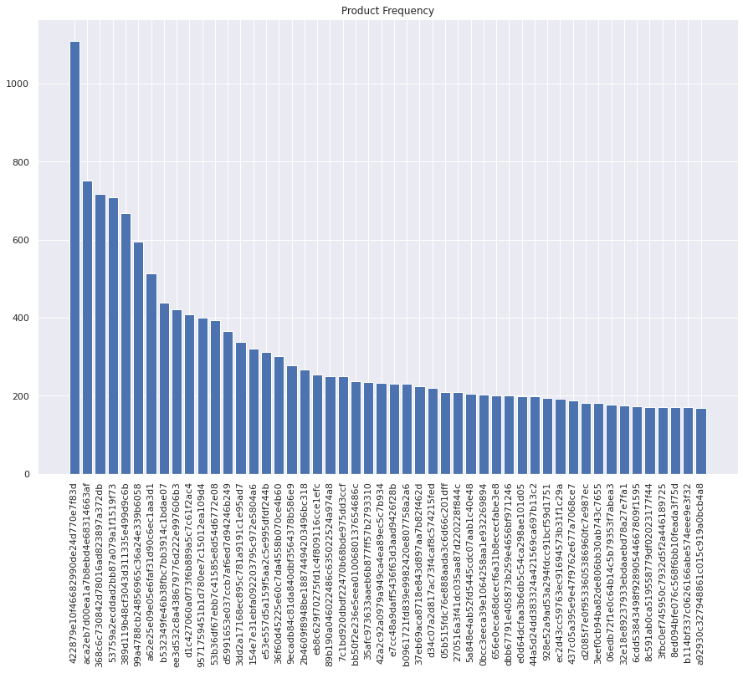


### EDA

Tìm ra tần suất xuất hiện của các loại hàng trong giỏ hàng.

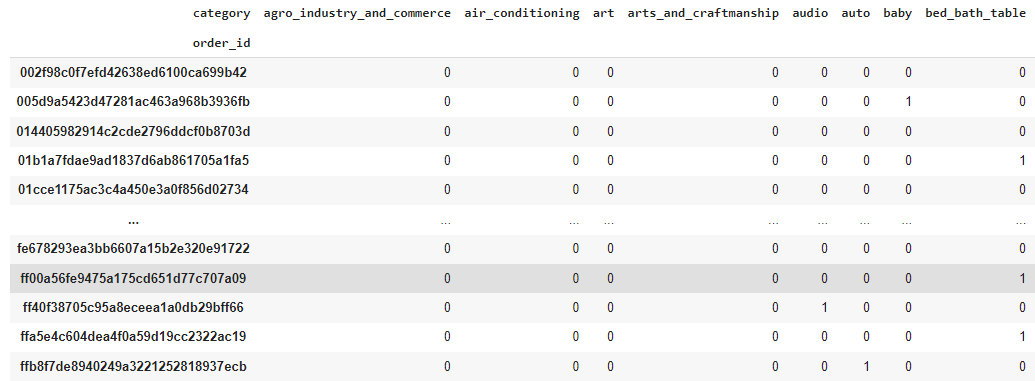


Tìm ra các sản phẩm được mua nhiều nhất.



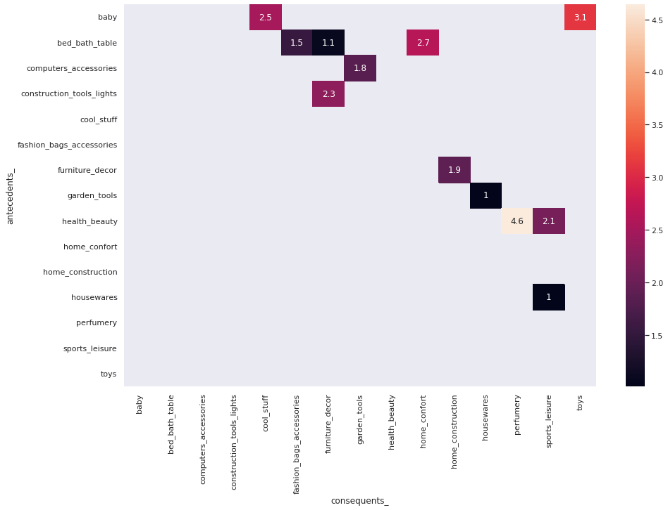
### Luật kết hợp theo danh mục hàng

Tạo ra giỏ hàng bằng cách nhóm lại theo order\_id và category. Kết quả thu được là một ma trận có 71 cột (71 category) nhưng vì quá nhiều nên tôi sẽ chỉ trình bày 1 vài cột.

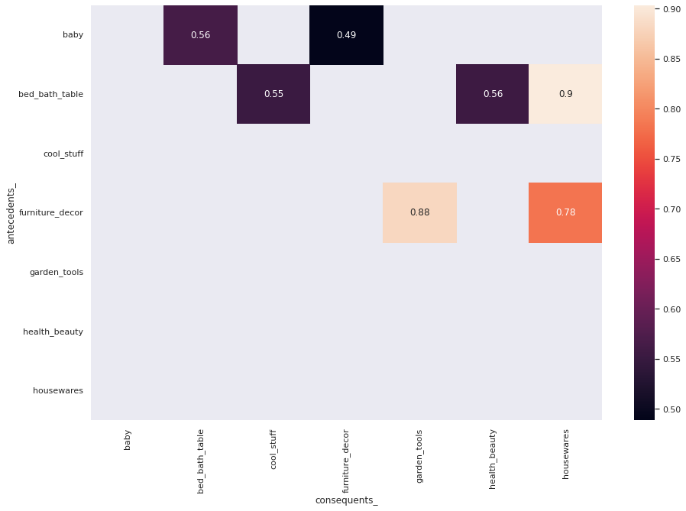


Dùng thuật toán apriori để tìm các luật kết hợp sau đó dùng heatmap để visualize các quy luật theo lift

Lift > 1: Mặt hàng bổ sung => nên xuất hiện cùng nhau.



Lift < 1: Mặt hàng thay thế => không nên xuất hiện cùng nhau.



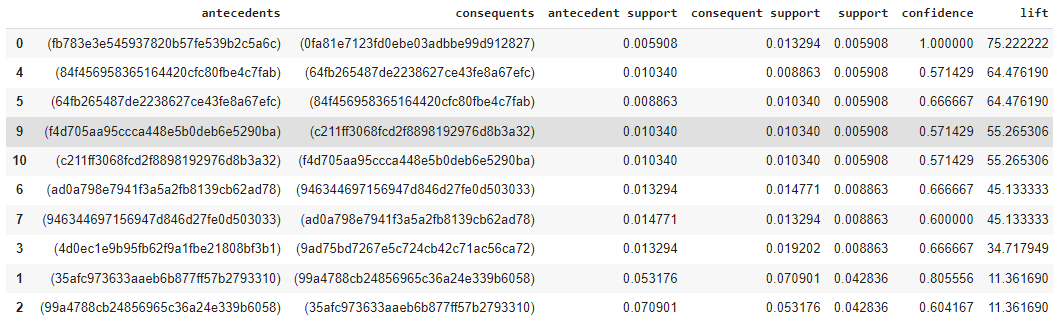
### Luật kết hợp giữa các sản phẩm

Thực hiện tương tự các bước trên để tìm ra luật giữa các sản phẩm, nhưng bổ sung điều kiện về loại hàng. Ở đây tôi chỉ tìm quy luật của sản phẩm thuộc 2 loại hàng có lift cao nhất là {health\_beauty = > perfumery} và {bed\_bath\_table = > home\_confront}.

{health\_beauty = > perfumery}



{bed\_bath\_table = > home\_confront}



## Kết luận

Theo quy luật về danh mục, chúng ta nên giới thiệu sản phẩm health\_beauty nếu khách hàng đã mua perfumery và ngược lại. Bên cạnh đó, có nhiều danh mục sẽ nâng cao xác suất khách hàng mua sản phẩm của A cùng với sản phẩm của B như bed\_bath\_table và home\_confront, ... Đồng thời chúng ta nên tránh baby và furniture\_decor, bed\_bath\_table và cool\_stuff,… xuất hiện đồng thời vì nó sẽ làm giảm xác suất mua hàng.

# Demand prediction model

## Ý tưởng

Như đã tìm được ở phần Market Basket Analysis, bed\_bath\_table là một trong những danh mục được quan tâm nhiều nhất. Vì vậy, tôi sẽ dự đoán nhu cầu mua các sản phẩm thuộc danh mục này bằng cách xây dựng mô hình hồi quy tuyến tính (linear regression) theo các biến:

* Lift \* số lượng các sản phẩm có liên quan đã được bán
* Tháng

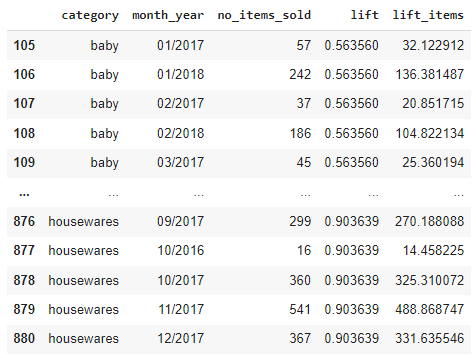
## Phân tích dữ liệu

### Tiền xử lý dữ liệu

Truy xuất lift của các sản phẩm có lift với bed\_bath\_table



Sử dụng lại kết quả từ phân tích trên để tính *Lift \* số lượng các sản phẩm có liên quan đã được bán.*

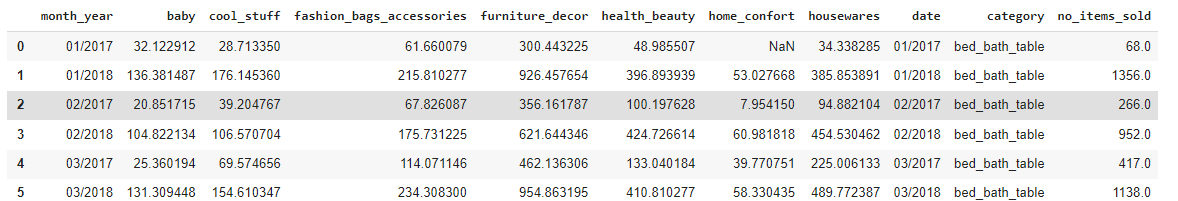


Thực hiện pivot table với index là month\_year, columns là category, values là lift\_items với aggfunc là sum (tổng lift\_items)

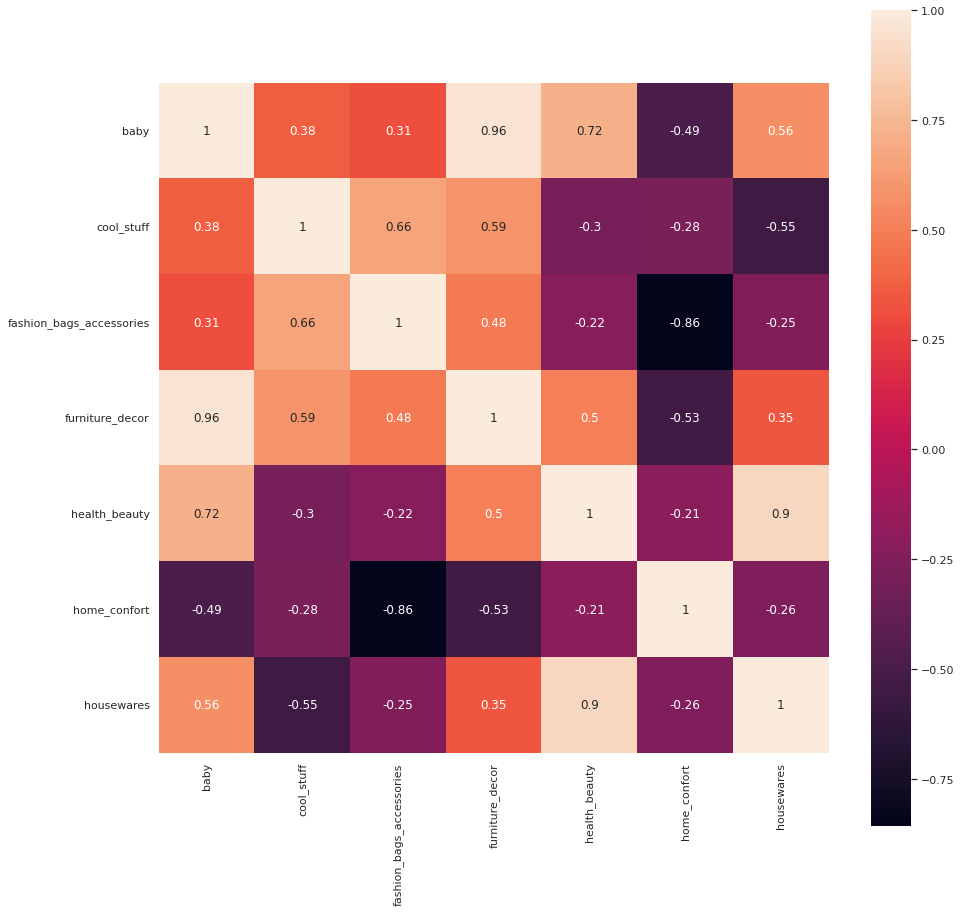
Tìm y (output) cho bài toán học có giám sát (supervised learning). Ở case này y là số lượng sản phẩm thuộc danh mục bed\_bath\_table được bán ra mỗi tháng.



Nối x và y lại để có một dataframe hoàn chỉnh.



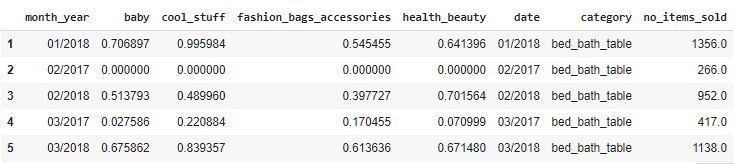
Sau đó xóa đi các dòng chứa dữ liệu bị khuyết và xóa đi những cột có tương quan cao với nhau. Độ tương quan tôi có thể chấp nhận là nhỏ hơn 80% (0.8).



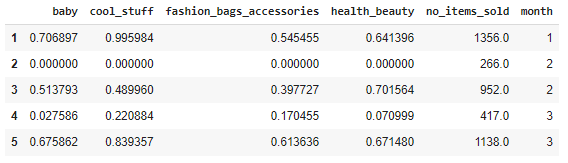
Các cột dữ liệu bị xóa là ['health\_beauty', 'housewares']. Kết quả thu được là



Tiếp theo sử dụng MinMaxScaler để scale các biến x



Cuối cùng tách tháng ra khỏi month\_year và xóa đi các cột không cần thiết. Dataframe dùng để train như sau



### Xây dựng mô hình

Chia tập train, test theo tỷ lệ 7:3, gọi mô hình hồi quy tuyến tính, fit dữ liệu tập train vào mô hình và dự báo dựa vào x\_test.

Thực hiện đánh giá mô hình dựa trên y\_pred (kết quả dự báo của x\_test) và y\_test. Kết quả thu được là



### Mô hình dựa trên SelectKBest

Dùng hàm selectkbest để tự động chọn ra 5 features tốt nhất cho mô hình hồi quy. Sau đó thực hiện xây dựng một mô hình mới và tiến hành đánh giá.



R2 ở mô hình mới lớn hơn mô hình cũ, ta có thể kết luận mô hình mới tốt hơn mô hình cũ.

## Kết luận & hướng phát triển

Việc xây dựng mô hình dự báo nhu cầu cho sản phẩm thuộc danh mục bed\_bath\_table sẽ giúp dự đoán được nguồn cung cần thiết để đáp ứng nhu cầu khách hàng tham gia hệ thống. Vì bed\_bath\_table là danh mục được quan tâm nhiều nhất nên trong bài này tôi chỉ thực hiện việc dự đoán xoay quanh danh mục này. Nhưng thực tế cần xây dựng mô hình dự báo cho tất cả các danh mục/ sản phẩm hiện có trên hệ thống. Để giải quyết bài toán này, tôi đưa ra hướng phát triển cho bài toán là xây dựng pipeline hoặc đơn giản hơn là một user-defined function để xây dựng được mô hình dự đoán cho tất cả các danh mục.

\*\*Lưu ý: Ở bài toán này vì tôi đã xác định các biến liên quan đến mô hình bao gồm lift \* các sản phẩm liên quan nên ta không thể xây dựng mô hình từ một dataframe lớn bao gồm toàn bộ danh mục vì sẽ để lại nhiều cột/ dòng dữ liệu bị khuyết.

# 10. Tổng kết

## Tại sao có các bài toán?

Đầu tiên, khi tôi thực hiện phân cụm khách hàng (Customer Segmentation) tôi đã chú ý đến việc phần lớn khách hàng chỉ mua hàng một lần (frequency = 1). Việc xây dựng metrics Retention Cohort là để xem xét chi tiết hơn giả thuyết trên. Metrics cho thấy tỷ lệ ở lại khá thấp (khoảng 0.5%) tương ứng với tỷ lệ rời bỏ hơn 99.5%, một tỷ lệ rời khá cao. Từ đó, tôi đã tiến hành phân tích chuyên sâu hơn bằng cách thực hiện Root Cause Analysis để tìm hiểu nguyên nhân. Có 3 khía cạnh mà tôi quan tâm đến, đó là: Giao hàng, Đánh giá, Sự hứng thú của khách hàng thể hiện qua hình thức thanh toán (Khách hàng không hứng thú với hệ thống sẽ chỉ mua hàng khi có voucher). Cuối cùng, tôi tìm được nguyên nhân bao gồm: thời gian giao hàng chậm hơn dự kiến, sản phẩm được quan tâm nhiều nhất là bed\_bath\_table lại là sản phẩm bị đánh giá thấp nhiều nhất,… (tôi sẽ nói kĩ hơn ở phần sau).

Để khuyến khích khách mua hàng, một trong những biện pháp tôi đưa ra là khuyến nghị các sản phẩm. Và để tìm ra các sản phẩm nên khuyến nghị, tôi thực hiện Market Basket Analysis bằng thuật toán apriori để tìm ra quy luật giữa các danh mục/ sản phẩm, đâu là món hàng nên xuất hiện cùng nhau và không nên. Cuối cùng, để đáp ứng nhu cầu khách hàng dựa trên những món hàng ta đã khuyến nghị thì nguồn cung cần là bao nhiêu? Việc xây dựng mô hình hồi quy tuyến tính Demand Prediction sẽ giúp giải quyết bài toán đó.

## Kết luận

Các cụm khách hàng nên tập trung vào là 1 và 3, dù họ chỉ mới mua hàng ít lần nhưng phần lớn những giao dịch vừa mới được thực hiện, điều này cho thấy họ chưa hoàn toàn rời bỏ hệ thống. Có thể tặng họ các voucher mua sắm, nhưng giảm giá trị của từng voucher lại, vì nếu không mua hàng lâu mà được tặng voucher giá trị bằng cả một đơn hàng thì sẽ giảm hứng thú của khách hàng đối với hệ thống (đợi đến khi có voucher mới mua hàng).

Các chiến lược để thu hút khách hàng quay lại: Tập trung vào nâng cao trải nghiệm và độ hài lòng của khách hàng thông qua việc thiện tốc độ giao hàng, vì nó là nguyên nhân chính khiến nhiều khách hàng chỉ mua hàng một lần trong quá khứ. Song song với việc đó, nên kiểm định lại chất lượng của hàng hóa được bán, đặc biệt là hàng hóa thuộc danh mục bed\_bath\_table, vì kết quả phân tích cho thấy, đây là danh mục hàng được mua nhiều nhất và cũng có quy luật với nhiều danh mục khác, nhưng lại là danh mục bị đánh giá thấp nhiều nhất. Việc khách hàng không mua sản phẩm thuộc danh mục này có thể ảnh hưởng rất lớn đến các danh mục khác cũng như việc mua hàng.

Các chiến lược để khuyến khích mua hàng: Xây dựng hệ thống khuyến nghị dựa trên những thông tin có được thông qua phân tích giỏ hàng (Market Basket Analysis) để gợi ý các món hàng phù hợp với nhu cầu khách hàng, tránh các món hàng không nên xuất hiện cùng nhau. Từ đó có thể tăng giá trị mỗi đơn hàng, hoặc tăng số lần mua hàng của khách hàng.