DATA ANALYTICS FINAL PROJECT

Dataset: Brazilian E-Commerce Public Dataset by Olist

By: Trần Mạnh Tường

TABLE OF CONTENTS

TABLE OF CONTENTS

1/ Tổng quan về mục tiêu dự án

Làm sao để tăng doanh thu?

2

Cách tiếp cận vấn đề

1	2	3	4	5	6
THINKING FLOW	Tăng doanh thu	Tăng đơn đặt hàng	Khuyến nghị những món hàng nào?	Market Basket Analysis		
		Tăng đơn đặt hàng	Tập trung vào những khách hàng nào?	Customer Segmentation		
			Có bao nhiêu khách quay lại sau lần mua đầu?	Retention Cohort	Tại sao khách chỉ mua 1 lần?	Rootcause Analysis

2

Cách tiếp cận vấn đề

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.

2

Cách tiếp cận vấn đề

WORKING FLOW

Ask

Xác định bài toán

Prepare

Truy vấn các data cần thiết trong SQL Server và lưu thành file .csv

Process

Tiền xử lý dữ liệu bằng Python

Analyze

Xây dựng model/metrics

Share

Trực quan hóa dữ liệu

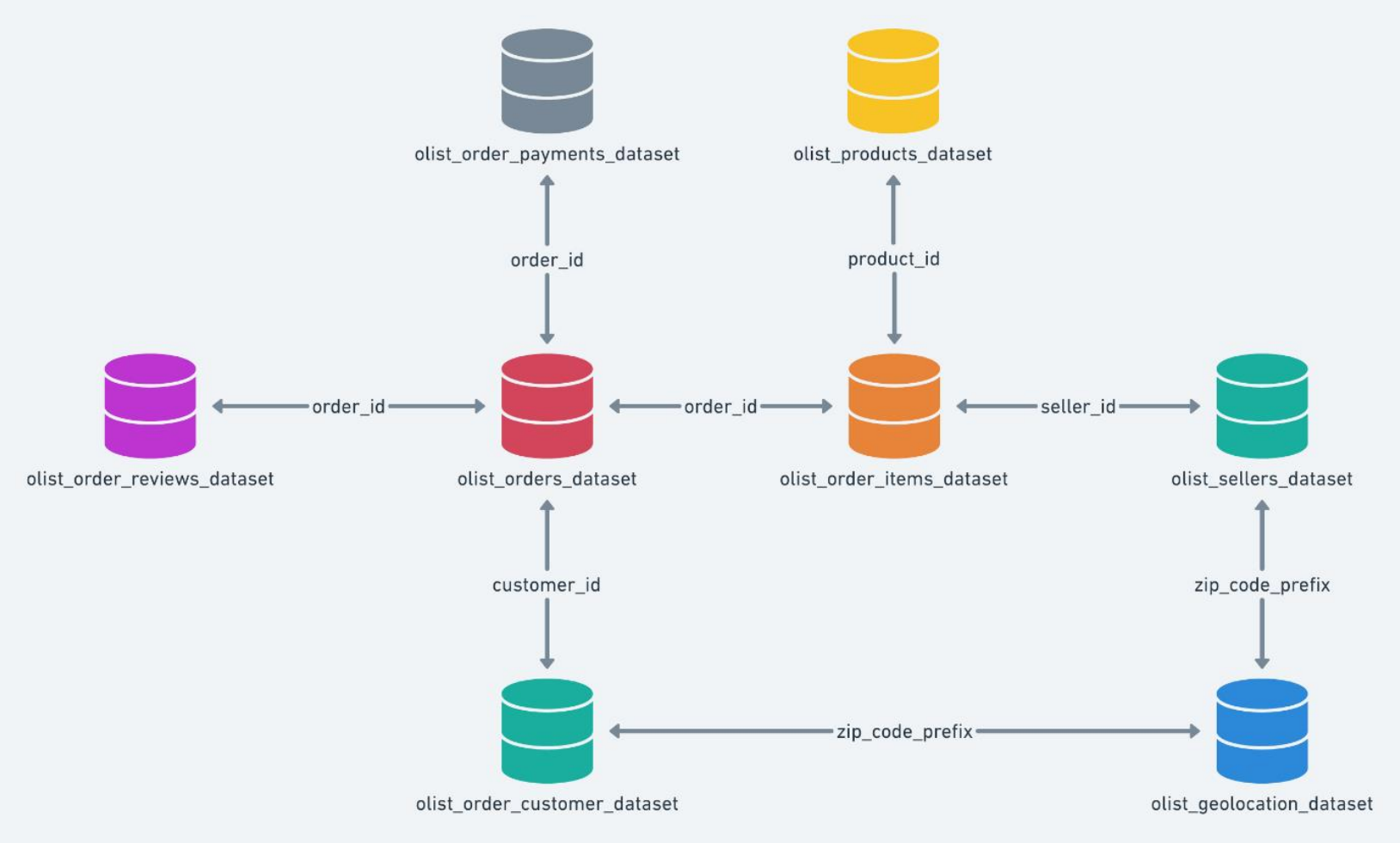
Act

Đưa ra kết luận và hành động

3

Tổng quan về nguồn dữ liệu

Mô hình Data warehouse



Các table

- Customers
- Sellers
- Geolocation
- Products
- Order_items
- Orders
- Order_reviews
- Payments

4

Giới thiệu về dữ liệu

Market Basket Analysis

Customer Segmentation

Retenton Cohort

```
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)

4

Giới thiệu về dữ liệu

Market Basket Analysis

Customer Segmentation

Retenton Cohort

```
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.

4

Giới thiệu về dữ liệu

Market Basket Analysis

Customer Segmentation

Retenton Cohort

Dùng lại data của Customer Segmentation

Rootcause Analysis

Rating

```
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.

4

Giới thiệu về dữ liệu

Market Basket Analysis

Customer Segmentation

Retenton Cohort

Dùng lại data của Customer Segmentation

Rootcause Analysis

Payment

```
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 *

4

Giới thiệu về dữ liệu

Retenton Cohort — Dùng lại data của Customer Segmentation

Rootcause Analysis

Shipment

```
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

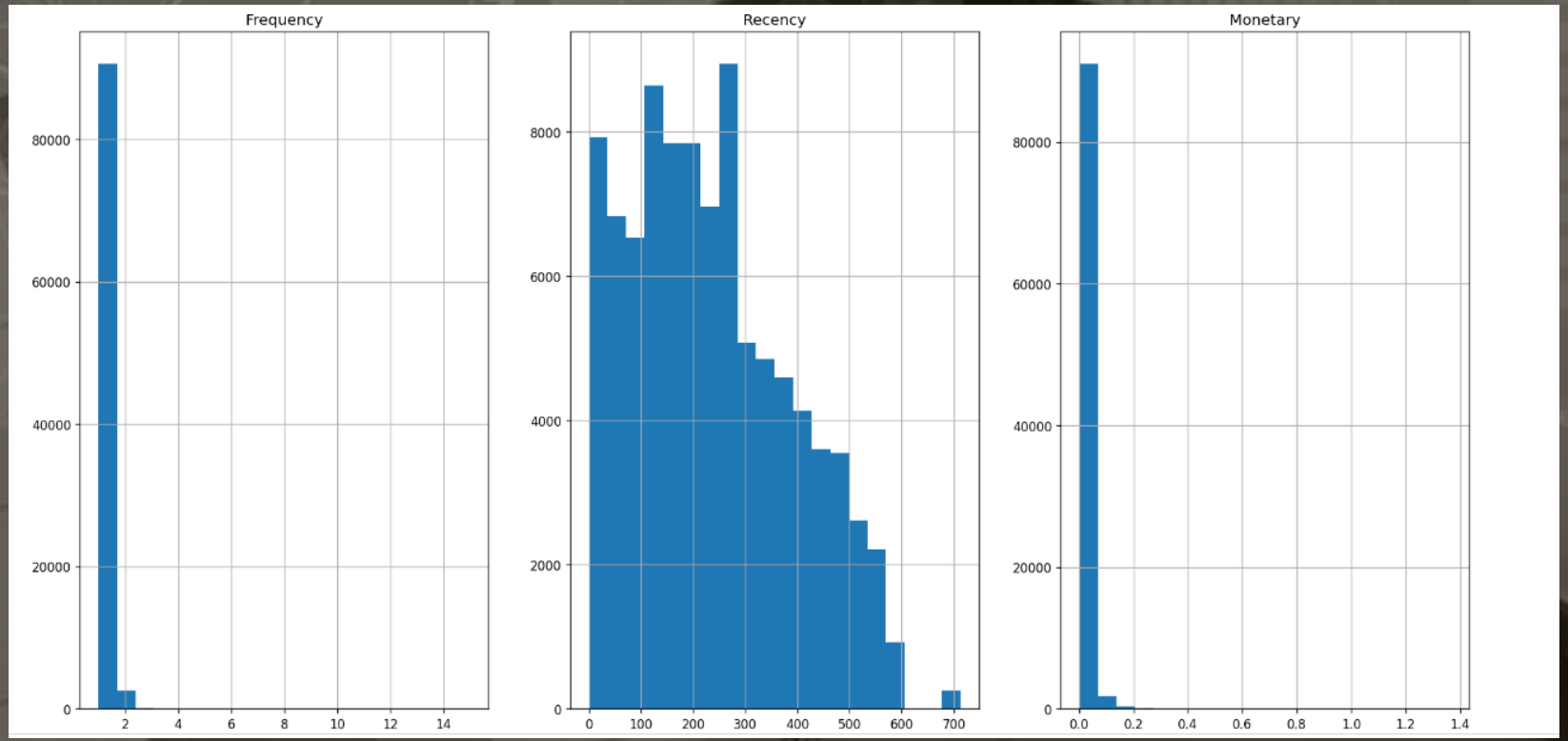
Từ bảng orders dùng hàm datediff với interval là day để tinh ra số ngày chênh lệnh giữa các columns daytime trong bảng.

5

Phân tích khám phá

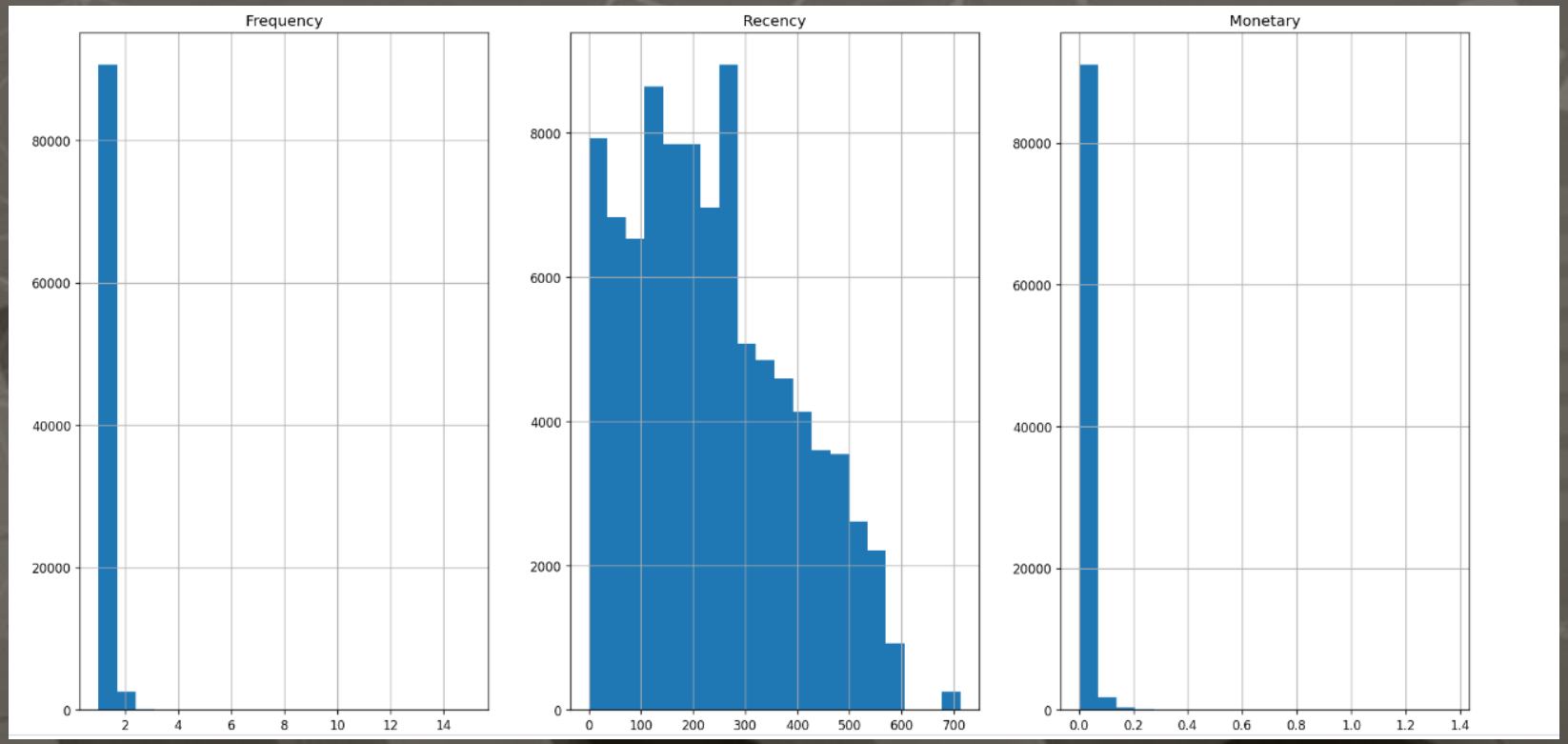
Customer Segmentation using RFM

	frequency	monetary	recency
count	93358.000000	9.335800e+04	93358.000000
mean	1.033420	1.651682e+04	237.478877
std	0.209097	2.262921e+04	152.595054
min	1.000000	9.590000e+02	0.000000
25%	1.000000	6.301000e+03	114.000000
50%	1.000000	1.077800e+04	218.000000
75%	1.000000	1.825100e+04	346.000000
max	15.000000	1.366408e+06	713.000000



Customer Segmentation using RFM

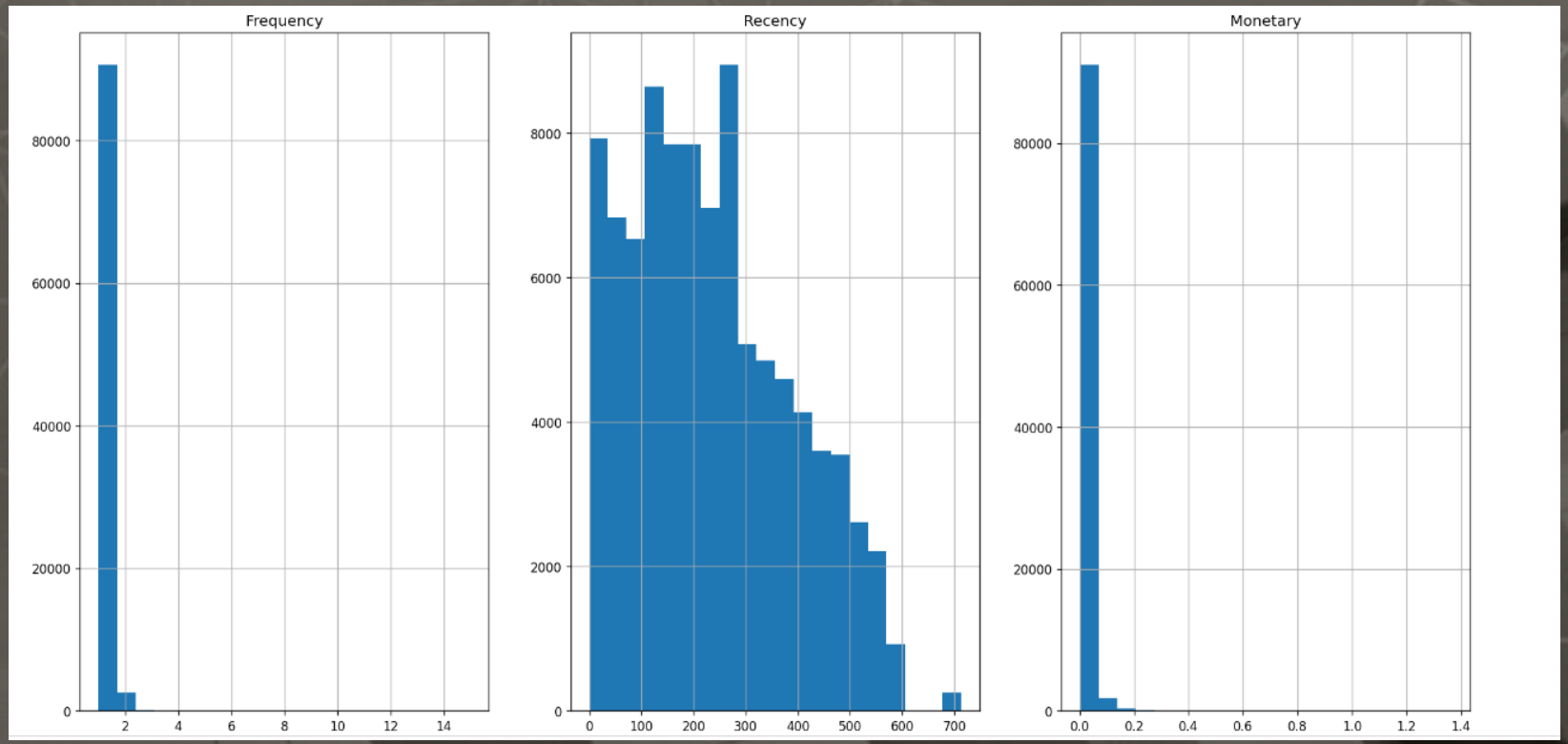
	frequency	monetary	recency
count	93358.000000	9.335800e+04	93358.000000
mean	1.033420	1.651682e+04	237.478877
std	0.209097	2.262921e+04	152.595054
min	1.000000	9.590000e+02	0.000000
25%	1.000000	6.301000e+03	114.000000
50%	1.000000	1.077800e+04	218.000000
75%	1.000000	1.825100e+04	346.000000
max	15.000000	1.366408e+06	713.000000

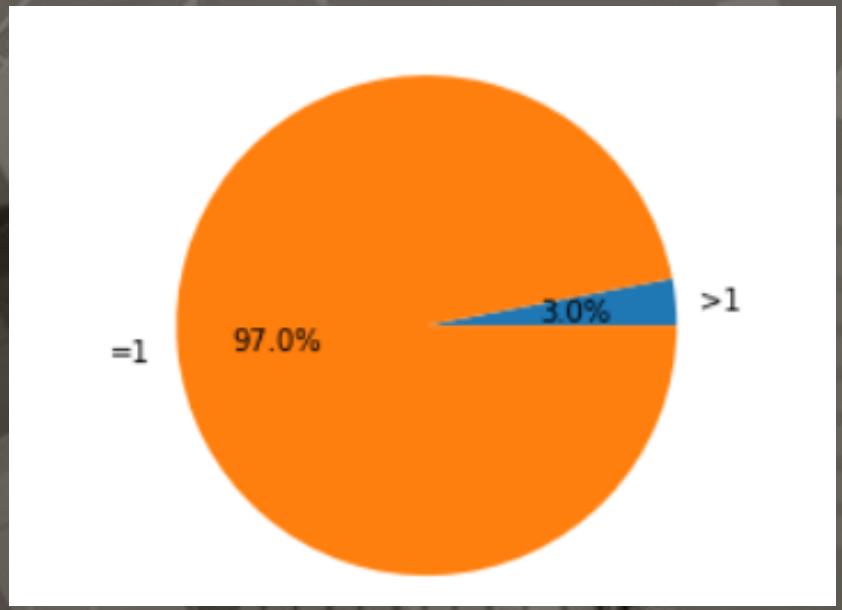


Phân phối của frequency là đáng chú ý nhất khi phần lớp chỉ xoay quanh giá trị 1

=> Hầu hết khách chỉ mua một lần

Customer Segmentation using RFM

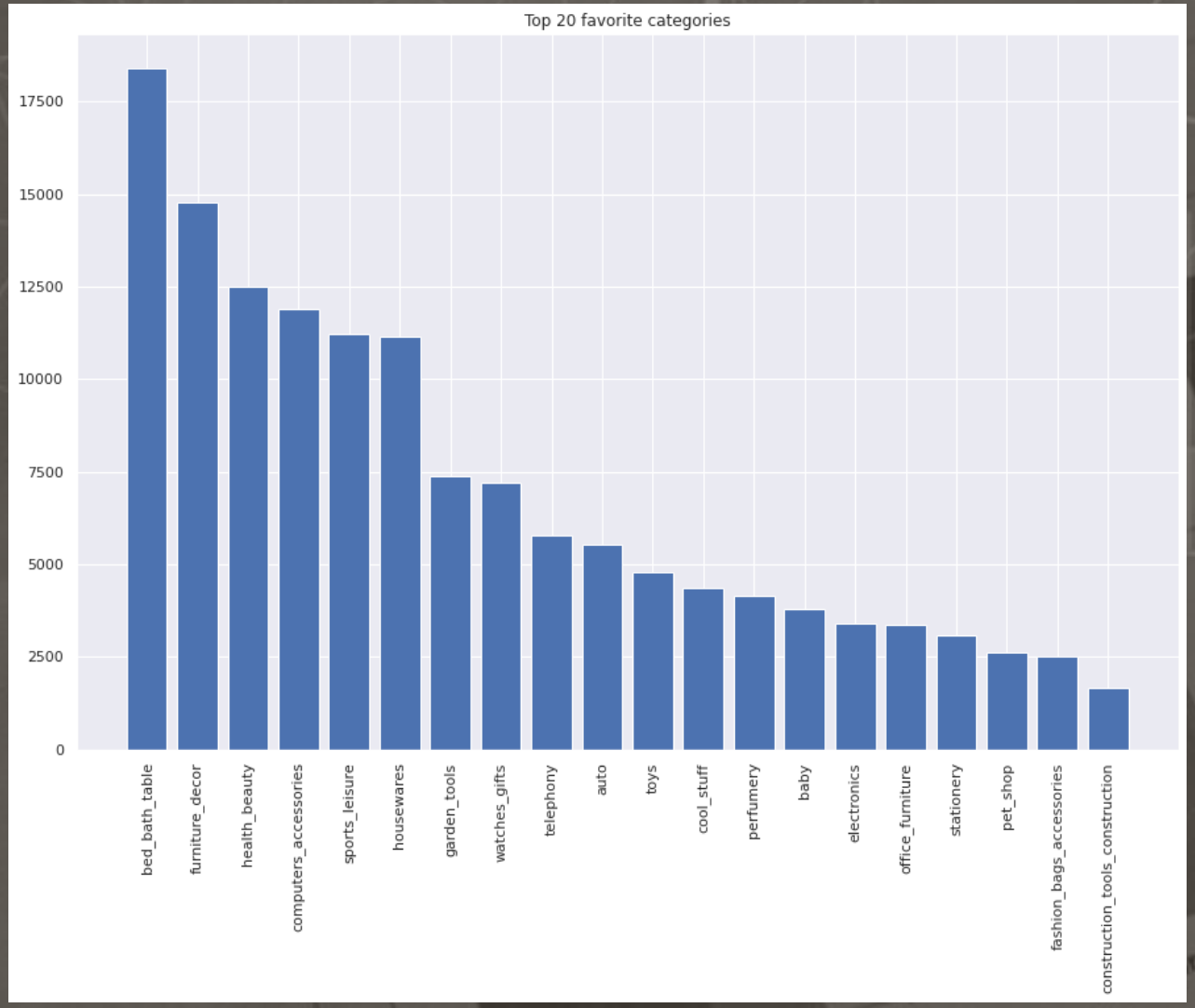


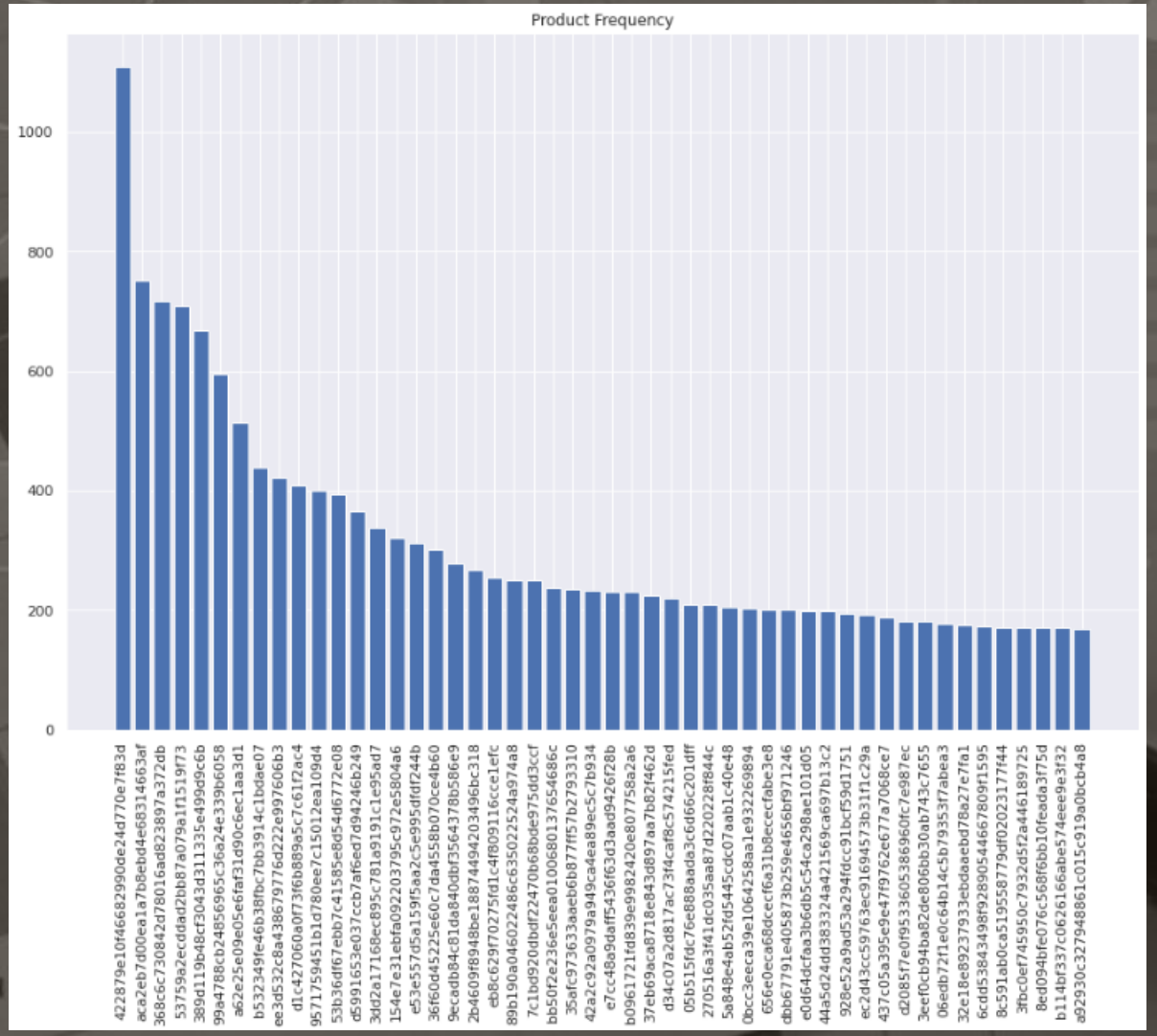


Tìm hiểu sâu hơn có thể thấy 97% khách hàng mua 1 lần và chỉ có 3% mua nhiều hơn 1 lần.

Market Basket Analysis

Items frequency





6

Giải quyết các bài toán

Customer Segmentation using RFM

Ý tưởng

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

Tiền xử lý dữ liệu

Tìm và xử lý mising value

Format dữ liệu

Nhóm dữ liệu và tính RFM

Customer Segmentation using RFM

Ý tưởng

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

Tiền xử lý dữ liệu

Tìm và xử lý mising value ✗

Format dữ liệu ✓

Nhóm dữ liệu và tính RFM ✓

	customer_id	frequency	first_order_date	last_order_date	monetary	recency
0	0000366f3b9a7992bf8c76cfdf3221e2	1	2018-05-10	2018-05-10	14190	111
1	0000b849f77a49e4a4ce2b2a4ca5be3f	1	2018-05-07	2018-05-07	2719	114
2	0000f46a3911fa3c0805444483337064	1	2017-03-10	2017-03-10	8622	537
3	0000f6ccb0745a6a4b88665a16c9f078	1	2017-10-12	2017-10-12	4362	321
4	0004aac84e0df4da2b147fca70cf8255	1	2017-11-14	2017-11-14	19689	288

Customer Segmentation using RFM

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

	customer_id	frequency	first_order_date	last_order_date	monetary	recency	RecencyScore	MonetaryScore	morethan_1	RFM_Group	RFM_Score	RM_segment
0	0000366f3b9a7992bf8c76cfdf3221e2	1	2018-05-10	2018-05-10	14190	111	4	2	False	042	6	42
1	0000b849f77a49e4a4ce2b2a4ca5be3f	1	2018-05-07	2018-05-07	2719	114	4	4	False	044	8	44
2	0000f46a3911fa3c0805444483337064	1	2017-03-10	2017-03-10	8622	537	1	3	False	013	4	13
3	0000f6ccb0745a6a4b88665a16c9f078	1	2017-10-12	2017-10-12	4362	321	2	4	False	024	6	24
4	0004aac84e0df4da2b147fca70cf8255	1	2017-11-14	2017-11-14	19689	288	2	1	False	021	3	21

Customer Segmentation using RFM

RFM Score

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)

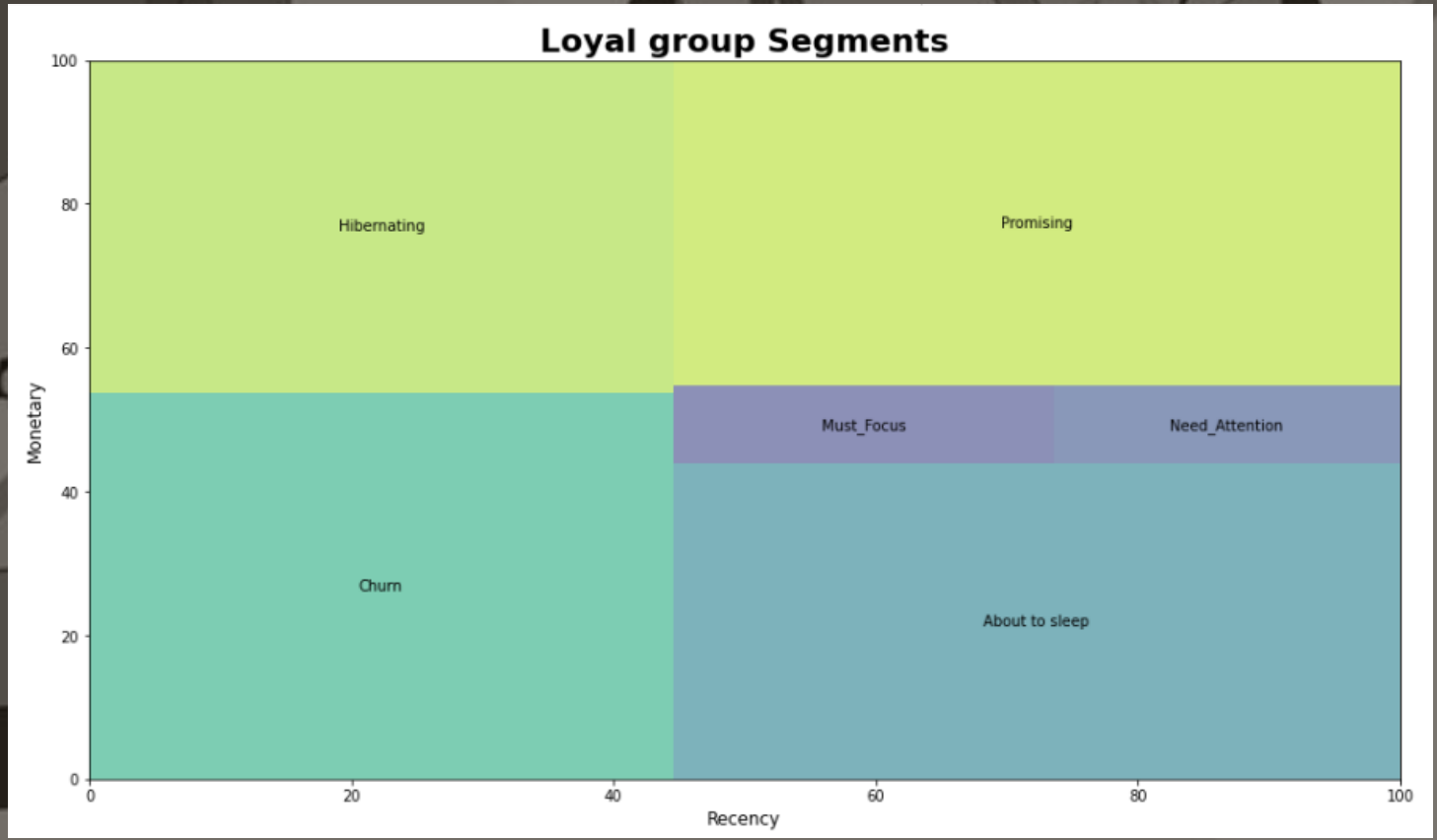
Customer Segmentation using RFM

RFM Score

Trực quan hóa RM theo loyal và onetime

Loyal customer

RM_group	recency mean	frequency mean	monetary mean	count
About to sleep	168.6	2.1	32702.8	671
Churn	439.5	2.1	29026.5	577
Hibernating	277.8	2.1	30971.0	681
Must_Focus	59.6	2.0	8267.4	89
Need_Attention	166.4	2.0	8153.9	81
Promising	58.4	2.2	35953.5	702



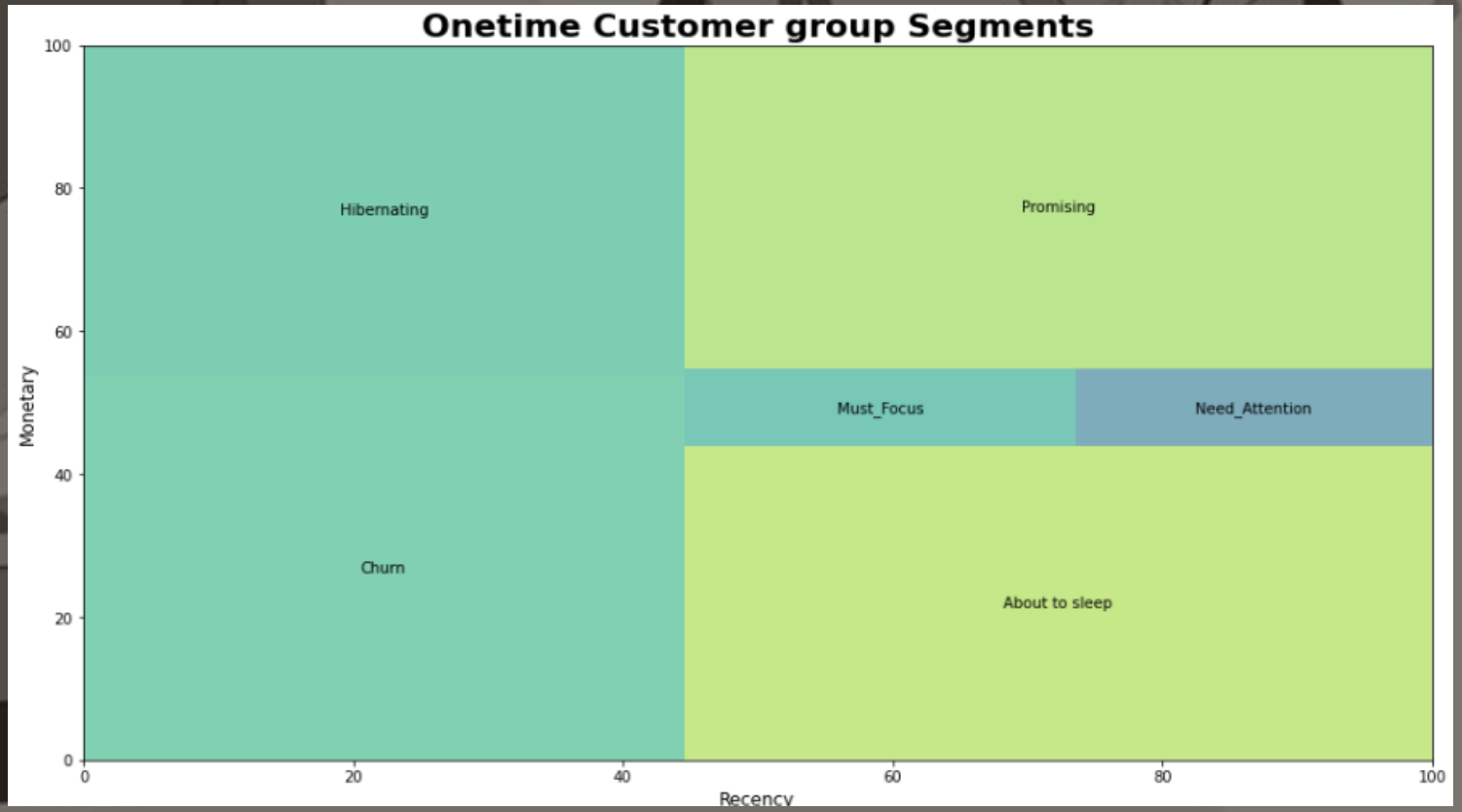
Customer Segmentation using RFM

RFM Score

Trực quan hóa RM theo loyal và onetime

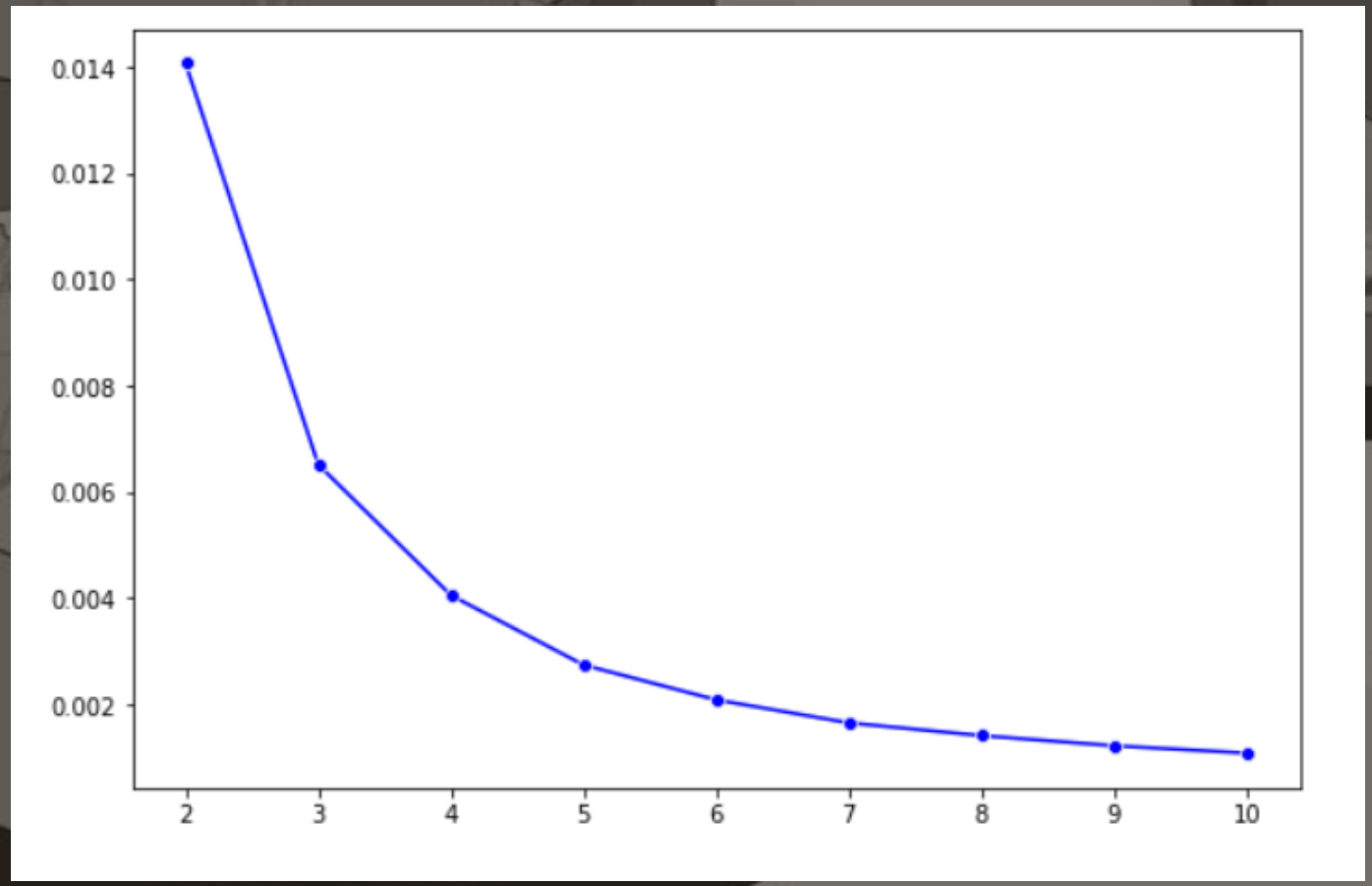
Onetime customer

	recency	frequency	monetary	
	mean	mean	mean	count
RM_group				
About to sleep	165.3	1.0	25640.9	11088
Churn	451.5	1.0	16094.1	22692
Hibernating	276.6	1.0	15863.9	22728
Must_Focus	56.4	1.0	6309.2	11376
Need_Attention	166.8	1.0	6292.0	11289
Promising	58.4	1.0	26588.8	11384



Customer Segmentation using RFM

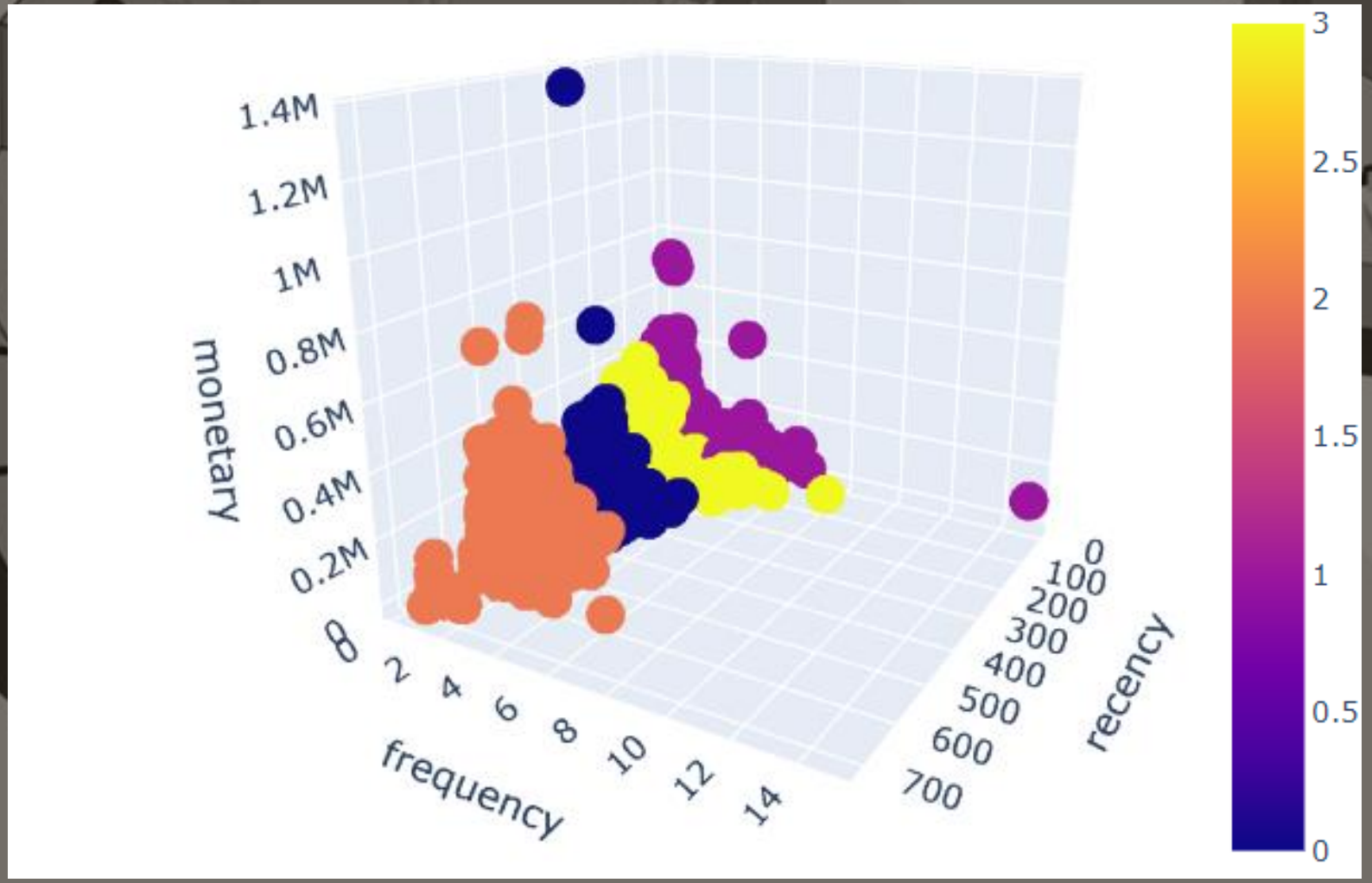
Tìm số cụm tối ưu (k-optimal)



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.

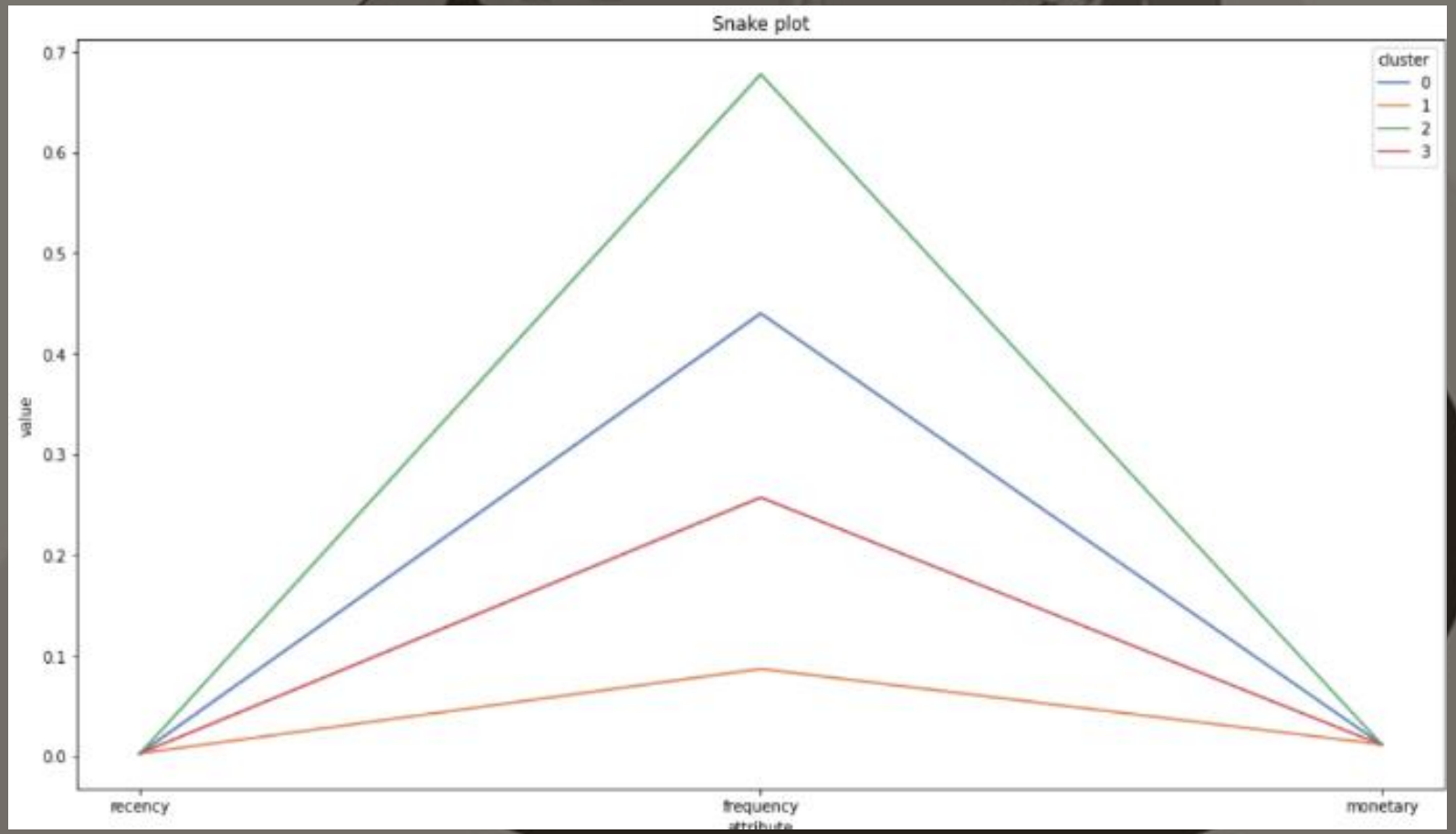
Customer Segmentation using RFM

Phân cụm khách hàng



Customer Segmentation using RFM

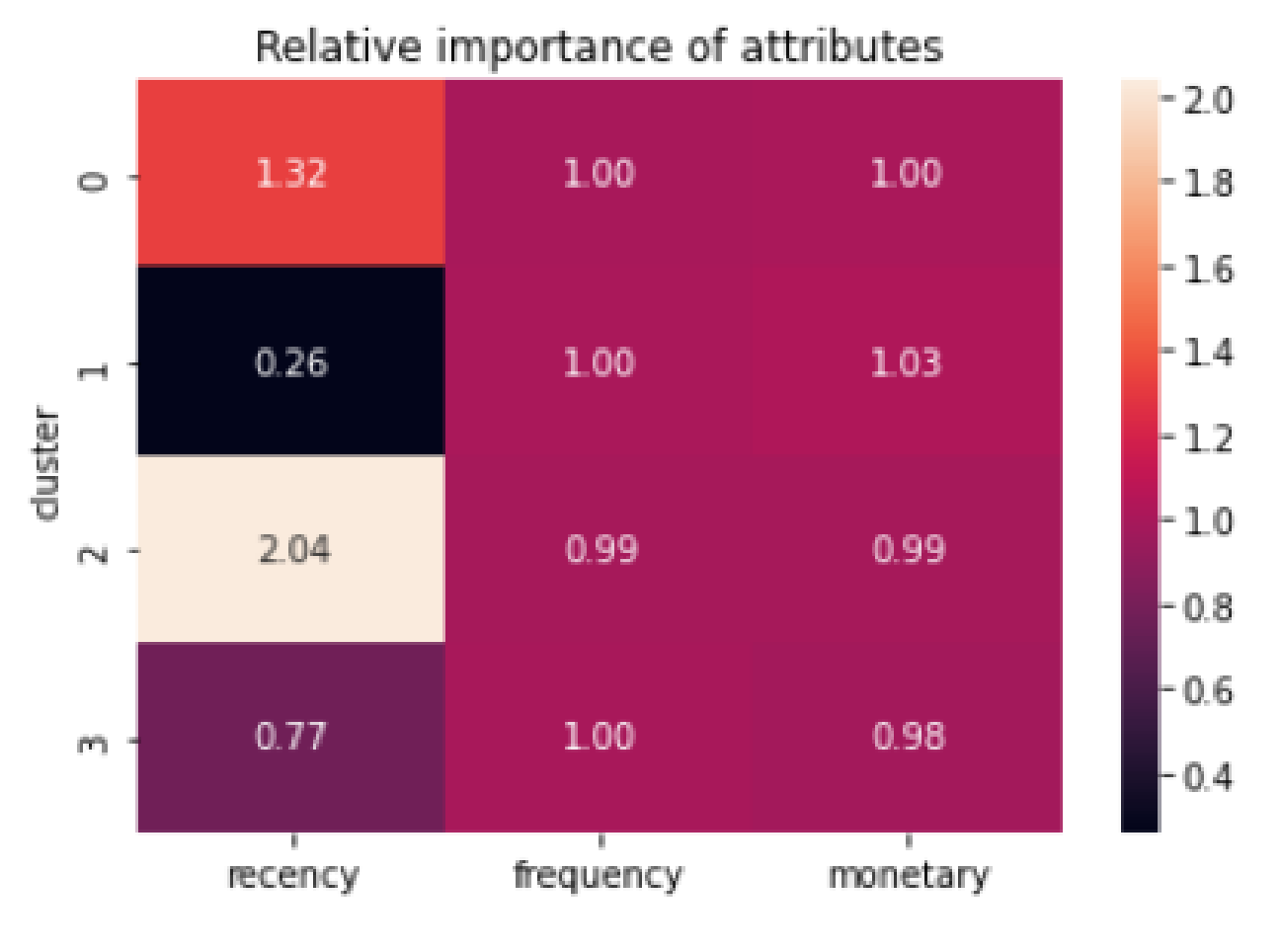
Phân cụm khách hàng



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

Customer Segmentation using RFM

Phân cụm khách hàng



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

Customer Segmentation using RFM

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) thường 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.

Tiền xử lý dữ liệu

Tìm và xử lý mising value 

Format dữ liệu, trích xuất tháng và năm 

Tính các cột dữ liệu mới 

First cohort = min purchase date

Cohort = Năm*100 + Tháng

Cohort distance = Cohort – First cohort

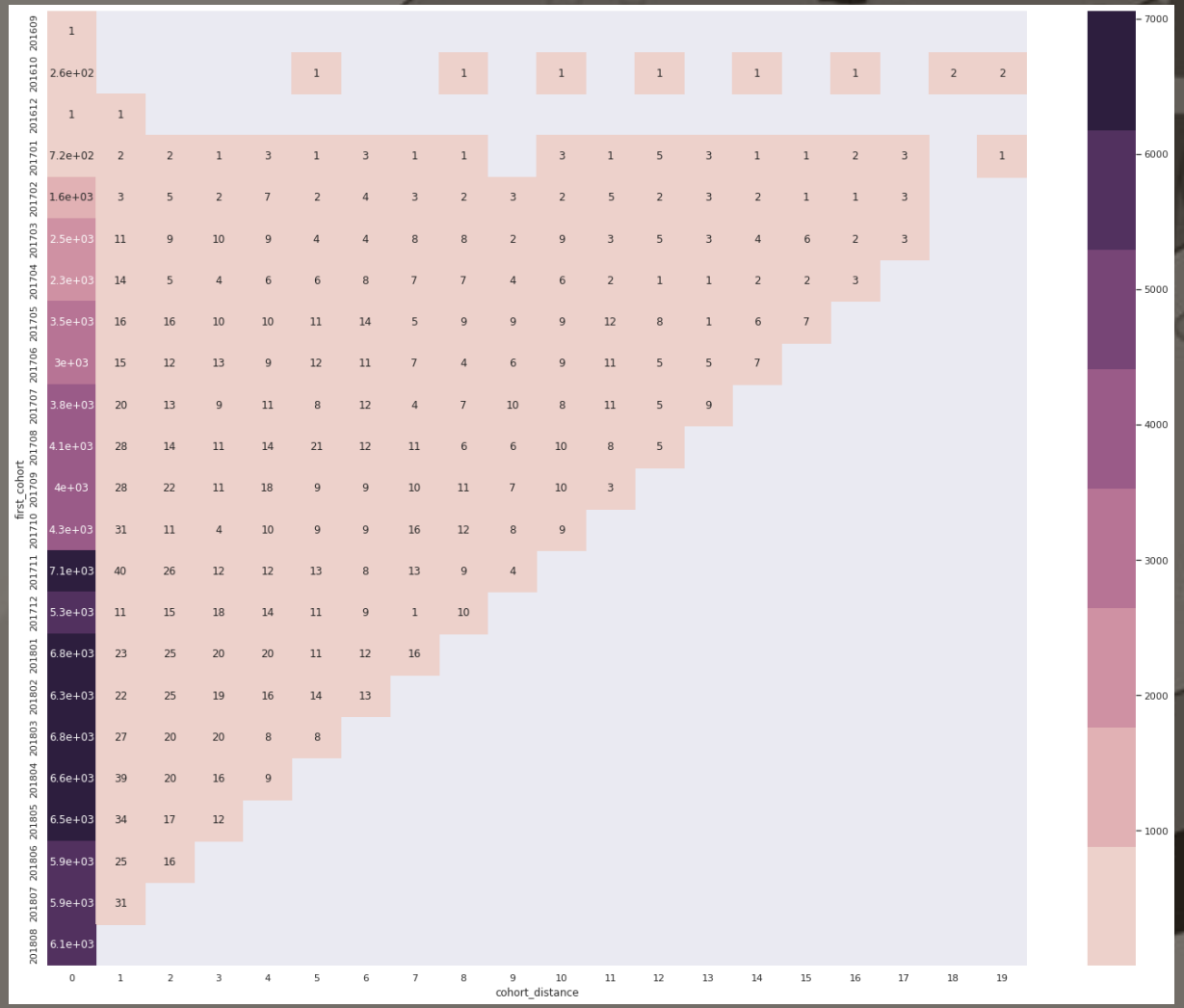
Retention Cohort

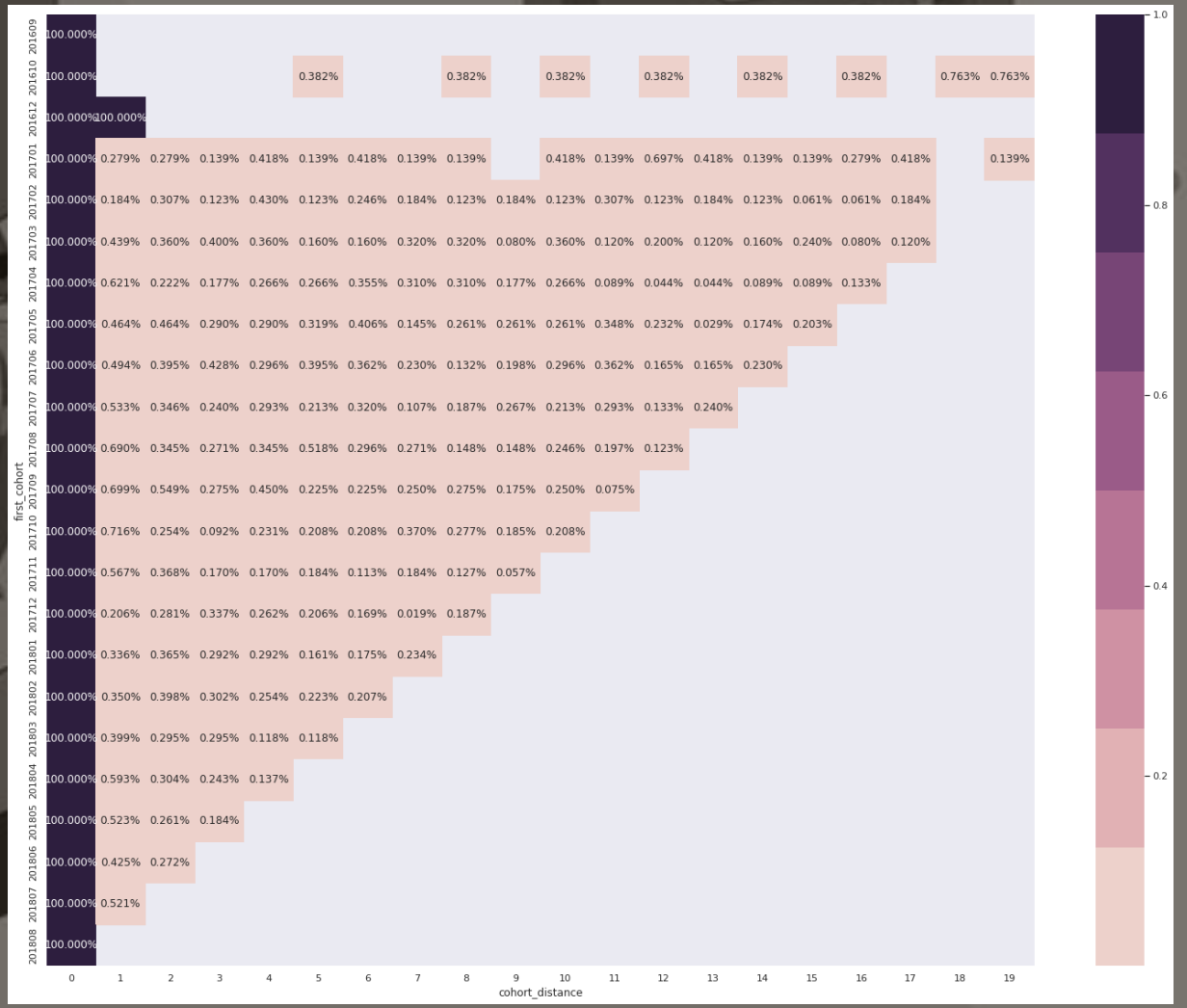
Tiền xử lý dữ liệu

	order_id	customer_id	purchase_timestamp	value	datetime	month	year	cohort	first_cohort	cohort_distance
0	00010242fe8c5a6d1ba2dd792cb16214	871766c5855e863f6eccc05f988b23cb	2017-09-13 08:59:02	7219	13/09/2017	09	2017	201709	201709	0
1	00018f77f2f0320c557190d7a144bdd3	eb28e67c4c0b83846050ddfb8a35d051	2017-04-26 10:53:06	25983	26/04/2017	04	2017	201704	201704	0
2	000229ec398224ef6ca0657da4fc703e	3818d81c6709e39d06b2738a8d3a2474	2018-01-14 14:33:31	21687	14/01/2018	01	2018	201801	201801	0
3	00024acbcdf0a6daa1e931b038114c75	af861d436cfc08b2c2ddefd0ba074622	2018-08-08 10:00:35	2578	08/08/2018	08	2018	201808	201808	0
4	00042b26cf59d7ce69dfabb4e55b4fd9	64b576fb70d441e8f1b2d7d446e483c5	2017-02-04 13:57:51	21804	04/02/2017	02	2017	201702	201702	0
...	...	...	...	...	...	...	...	...	...	...
110192	fffc94f6ce00a00581880bf54a75a037	0c9aeda10a71f369396d0c04dce13a64	2018-04-23 13:57:06	34340	23/04/2018	04	2018	201804	201804	0
110193	fffcd46ef2263f404302a634eb57f7eb	0da9fe112eae0c74d3ba1fe16de0988b	2018-07-14 10:26:46	38653	14/07/2018	07	2018	201807	201807	0
110194	fffce4705a9662cd70adb13d4a31832d	cd79b407828f02fdbba457111c38e4c4	2017-10-23 17:07:56	11685	23/10/2017	10	2017	201710	201710	0
110195	fffe18544ffabc95dfada21779c9644f	eb803377c9315b564bdedad672039306	2017-08-14 23:02:59	6471	14/08/2017	08	2017	201708	201708	0
110196	fffe41c64501cc87c801fd61db3f6244	cd76a00d8e3ca5e6ab9ed9ecb6667ac4	2018-06-09 17:00:18	5579	09/06/2018	06	2018	201806	201806	0

Retention Cohort

Cohort Analysis





Retention Cohort

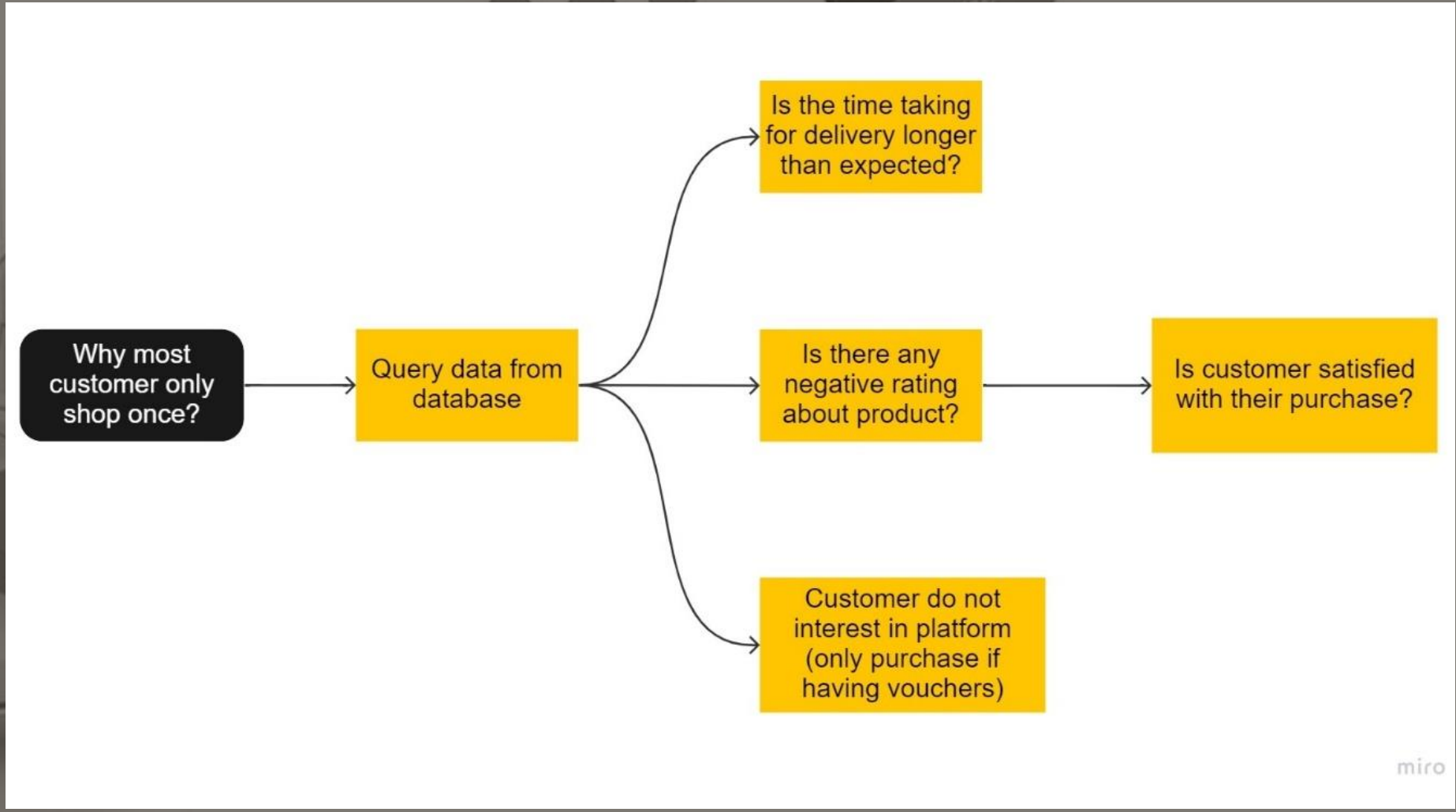
Kết luận

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)



Root Cause Analysis

Shipment

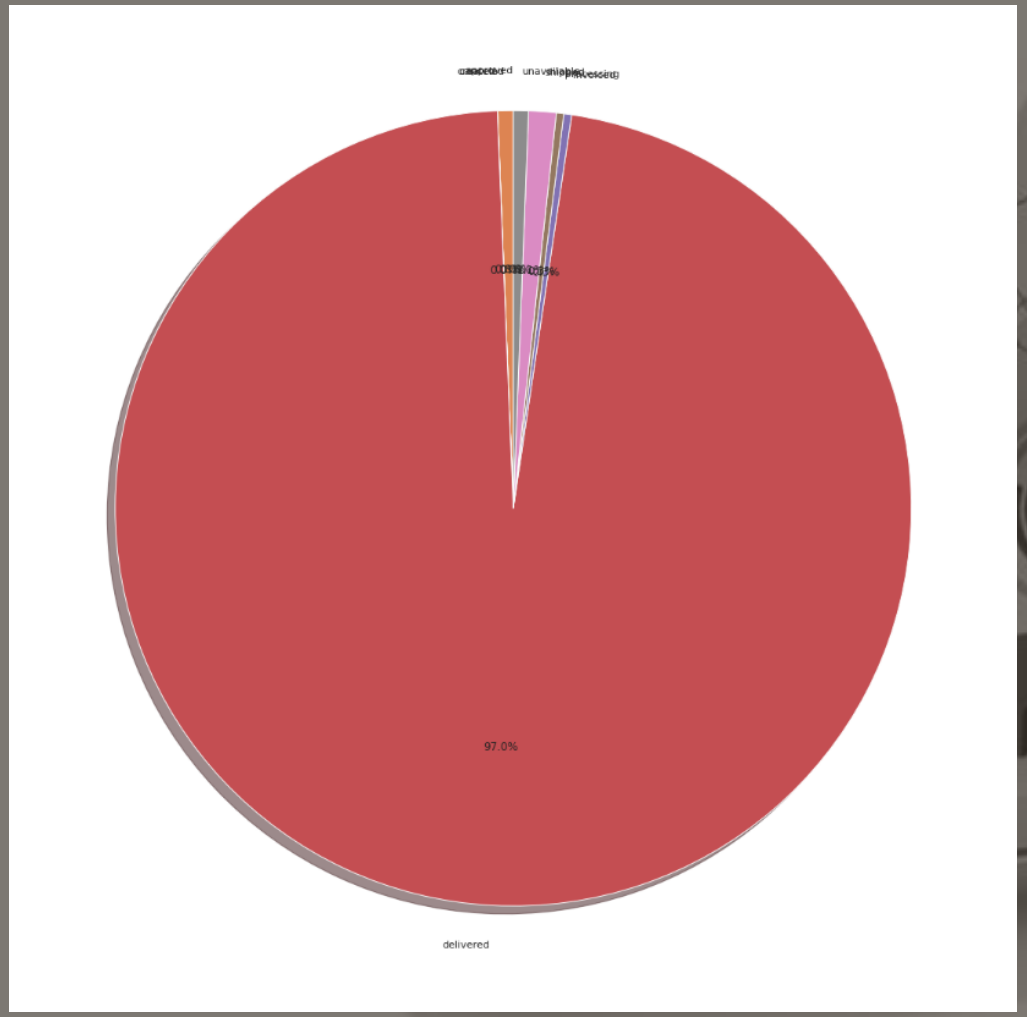
	order_id	status	purchase_timestamp	approved_after	carrier_take_after	delivered_after	total_delivery_time	estimated_delivery_time
0	00018f77f2f0320c557190d7a144bdd3	delivered	2017-04-26 10:53:06.0000000	0.0	8.0	8.0	16.0	19
1	000229ec398224ef6ca0657da4fc703e	delivered	2018-01-14 14:33:31.0000000	0.0	2.0	6.0	8.0	22
2	00024acbcdf0a6daa1e931b038114c75	delivered	2018-08-08 10:00:35.0000000	0.0	2.0	4.0	6.0	12
3	00042b26cf59d7ce69dfabb4e55b4fd9	delivered	2017-02-04 13:57:51.0000000	0.0	12.0	13.0	25.0	41
4	00048cc3ae777c65dbb7d2a0634bc1ea	delivered	2017-05-15 21:42:34.0000000	2.0	0.0	5.0	7.0	22

	approved_after	carrier_take_after	delivered_after	total_delivery_time	estimated_delivery_time
count	99280.000000	97643.000000	96474.000000	96475.000000	99440.000000
mean	0.518513	2.707158	9.282864	12.497393	24.404033
std	1.171329	3.568133	8.777234	9.555493	8.829573
min	0.000000	-171.000000	-16.000000	0.000000	2.000000
25%	0.000000	1.000000	4.000000	7.000000	19.000000
50%	0.000000	2.000000	7.000000	10.000000	24.000000
75%	1.000000	3.000000	12.000000	16.000000	29.000000
max	188.000000	126.000000	205.000000	210.000000	156.000000

Kết quả cho thấy, trung bình tổng ngày giao hàng nhỏ hơn số ngày ước tính. Điều này có thực sự đúng?

Root Cause Analysis

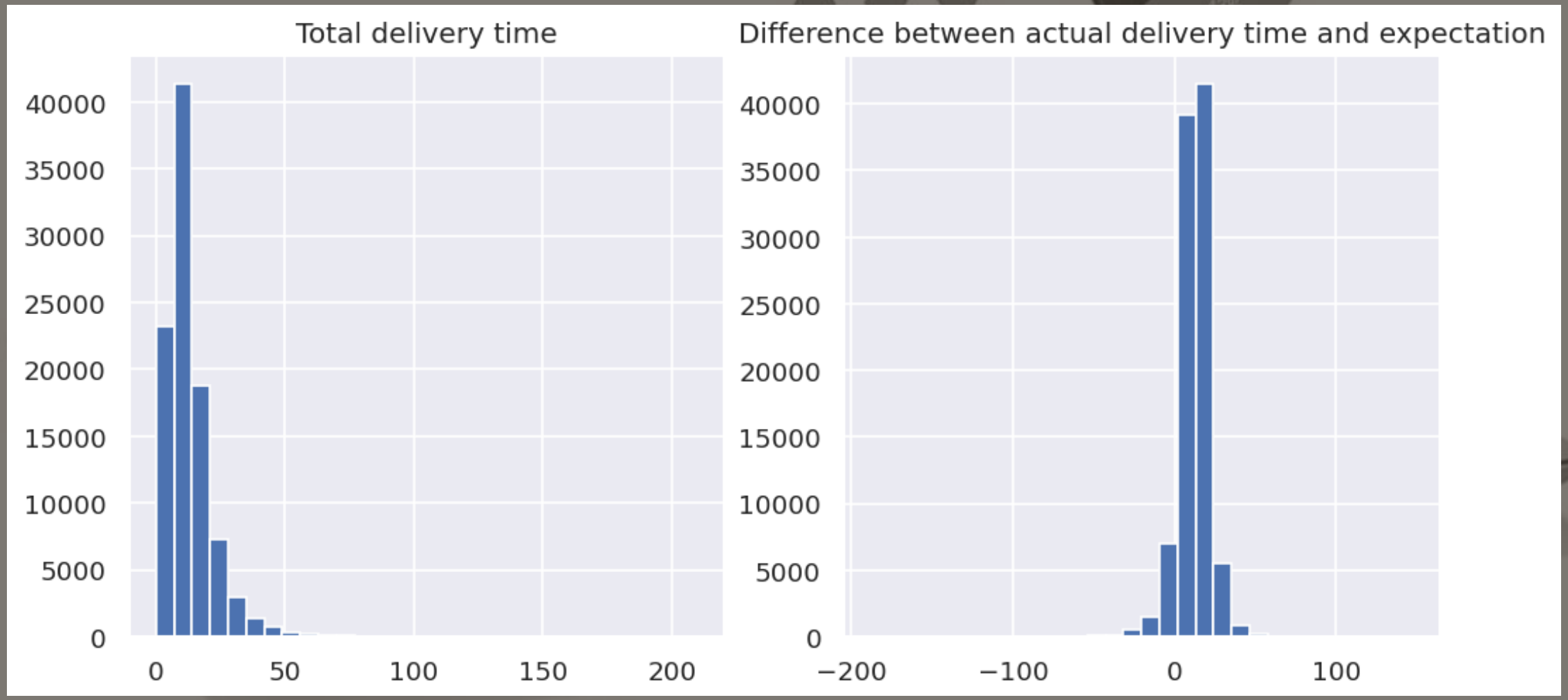
Shipment



Có tới 97% đơn hàng đã được giao thành công

Root Cause Analysis

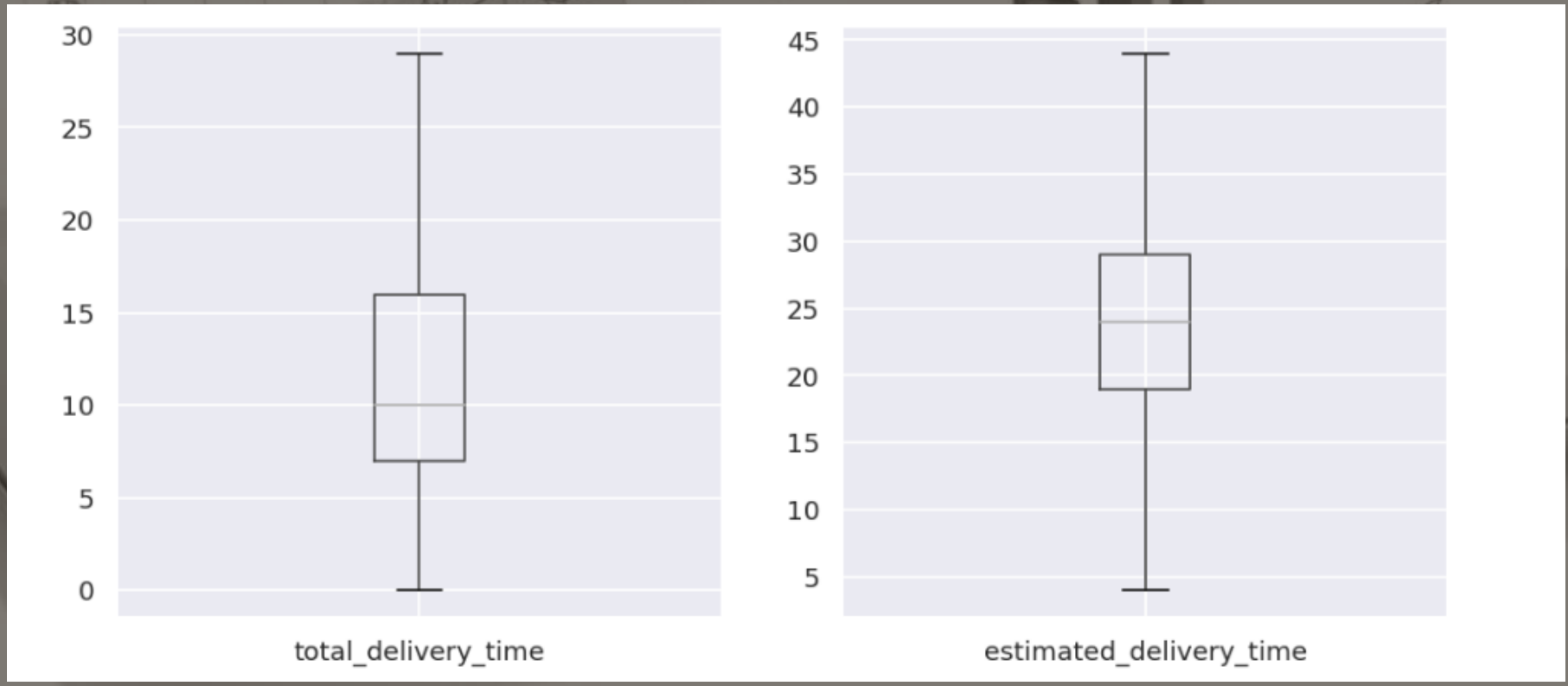
Shipment



Root Cause Analysis

Shipment

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



Root Cause Analysis

Shipment

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

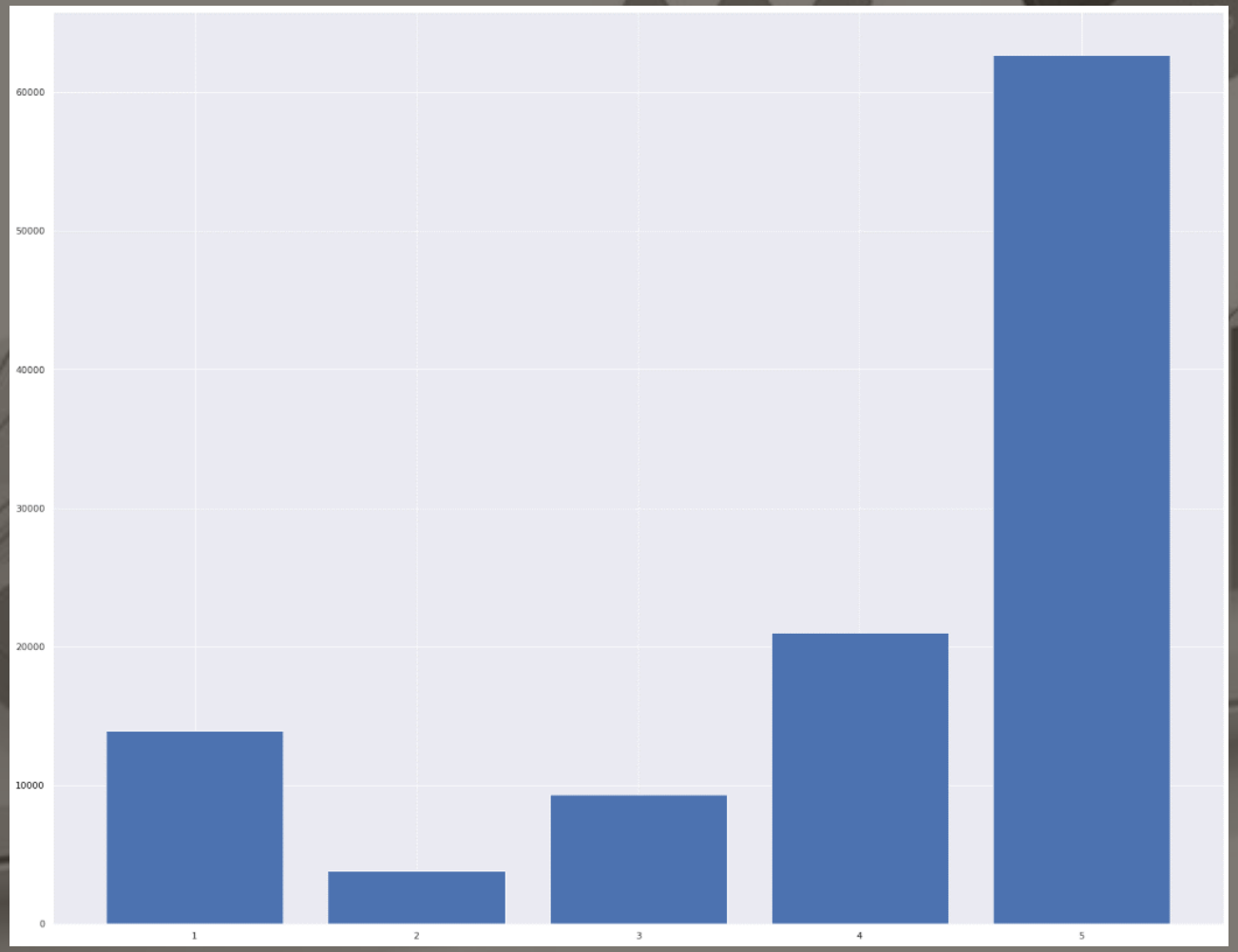
```
Ttest_indResult(statistic=-284.582619728147, pvalue=0.0)
```

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.

Root Cause Analysis

Rating

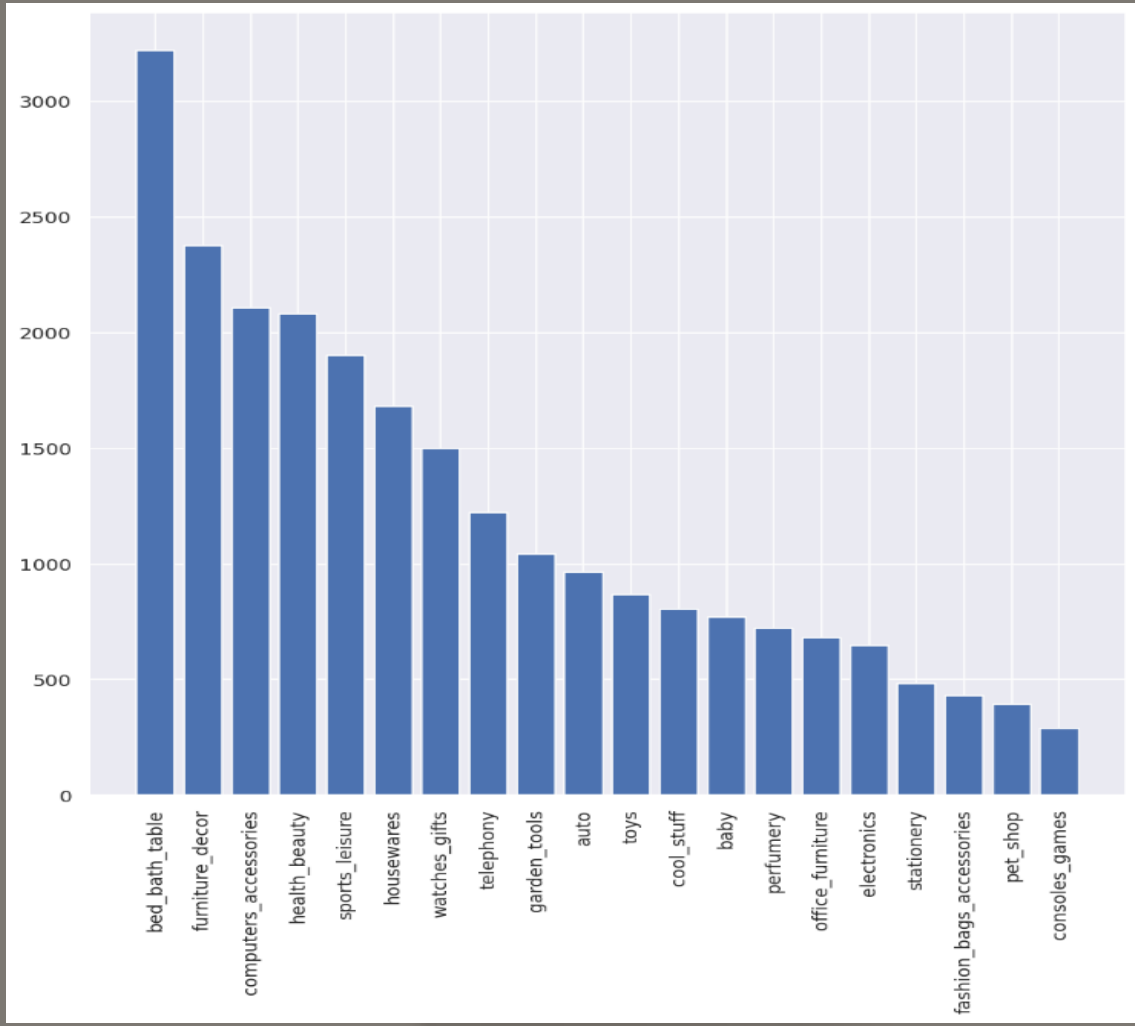
	score
count	110750.000000
mean	4.035395
std	1.385325
min	1.000000
25%	4.000000
50%	5.000000
75%	5.000000
max	5.000000



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.

Root Cause Analysis

Rating



Trong các sản phẩm bị đánh giá thấp, Bed_bath_table, furniture_decor, computer_accessories và health_beauty là top 5 sản phẩm bị đanh giá thấp nhất. Mà trong phần EDA ta đã thấy được tần suất xuất hiện của các danh mục này trong giỏ hàng là khá nhiều. Nên việc bị đanh quá thấp có khả năng ảnh hưởng đến trải nghiệm của phần lớn khách hàng và việc quay lại mua hàng của họ.

Root Cause Analysis

Rating

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

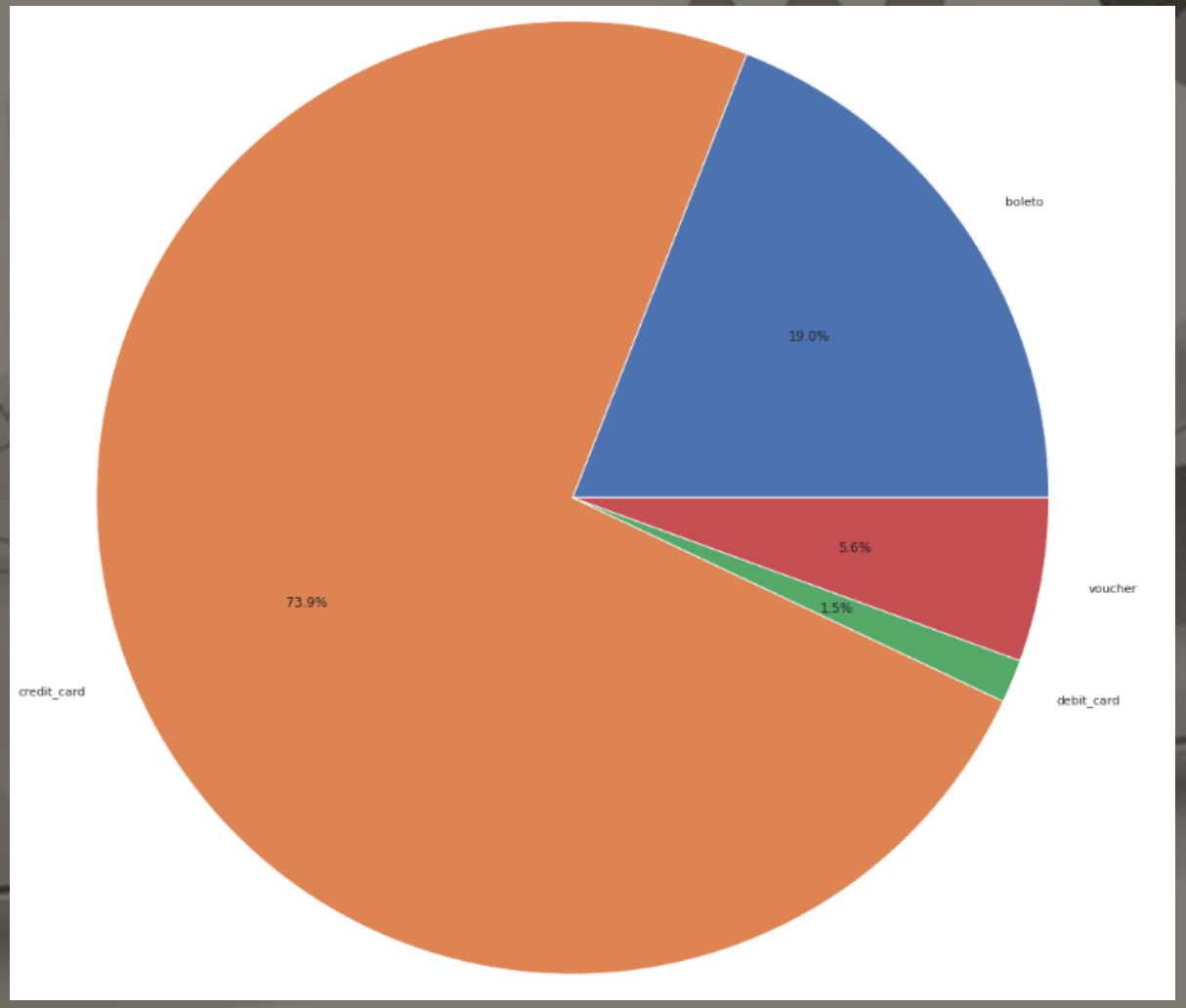
```
Power_divergenceResult(statistic=1364.3842550790068, pvalue=1.1531276790402767e-298)
```

Với mức ý nghĩa là 95%, $\alpha = 0.05$: 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.

Root Cause Analysis

Payment

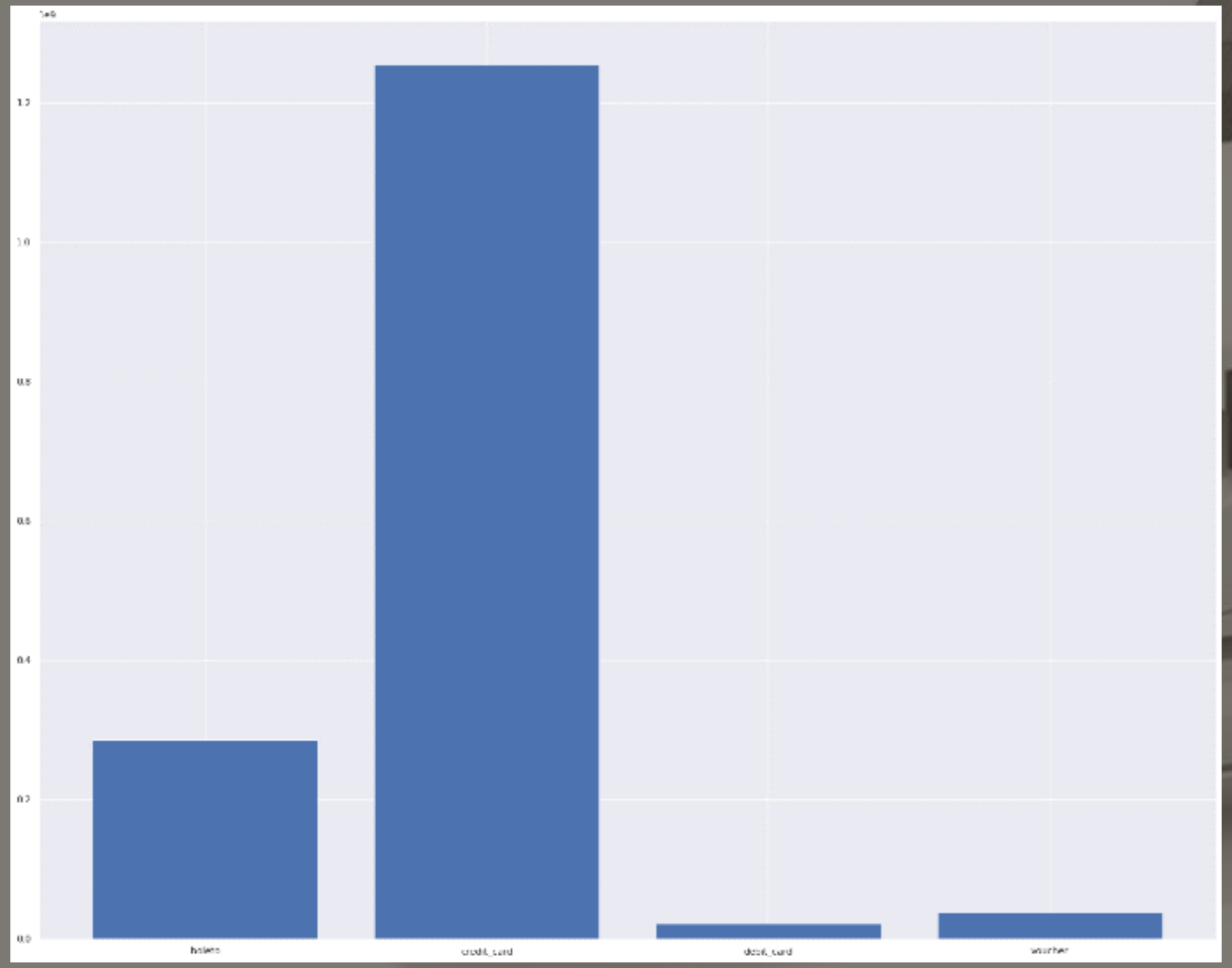
Đặt ra giả thuyết là những khách hàng không hứng thú với hệ thống thường chỉ mua hàng khi được tặng các voucher khuyến mãi. Dùng dữ liệu payment để kiểm định xem liệu khách đến mua hàng có thực sự quan tâm đến ứng dụng hay không?



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.

Root Cause Analysis

Payment



Giả thuyết không – H0: voucher >= boleto

Giả thuyết đối – Ha: voucher < boleto

```
Ttest_indResult(statistic=-36.92209549728959, pvalue=1.0)
```

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.

Root Cause Analysis

Payment

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

```
Ttest_indResult(statistic=-56.80306498567032, pvalue=1.0)
```

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

```
Ttest_indResult(statistic=-11.885860999458496, pvalue=1.0)
```

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.

Root Cause Analysis

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.

Market Basket Analysis

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.

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

$$support = P(A \cap B) = \frac{(\text{number of transactions containing } A \text{ and } B)}{(\text{total number of transactions})}$$

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

$$\text{conf}(X \Rightarrow Y) = P(Y|X) = \frac{\text{supp}(X \cap Y)}{\text{supp}(X)} = \frac{\text{number of transactions containing } X \text{ and } Y}{\text{number of transactions containing } X}$$

Market Basket Analysis

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.

Lift

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cap Y)}{\mathrm{supp}(X) \times \mathrm{supp}(Y)}$$

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.

Market Basket Analysis

Tiền xử lý dữ liệu

Tìm và xử lý mising value

Gom cụm dữ liệu

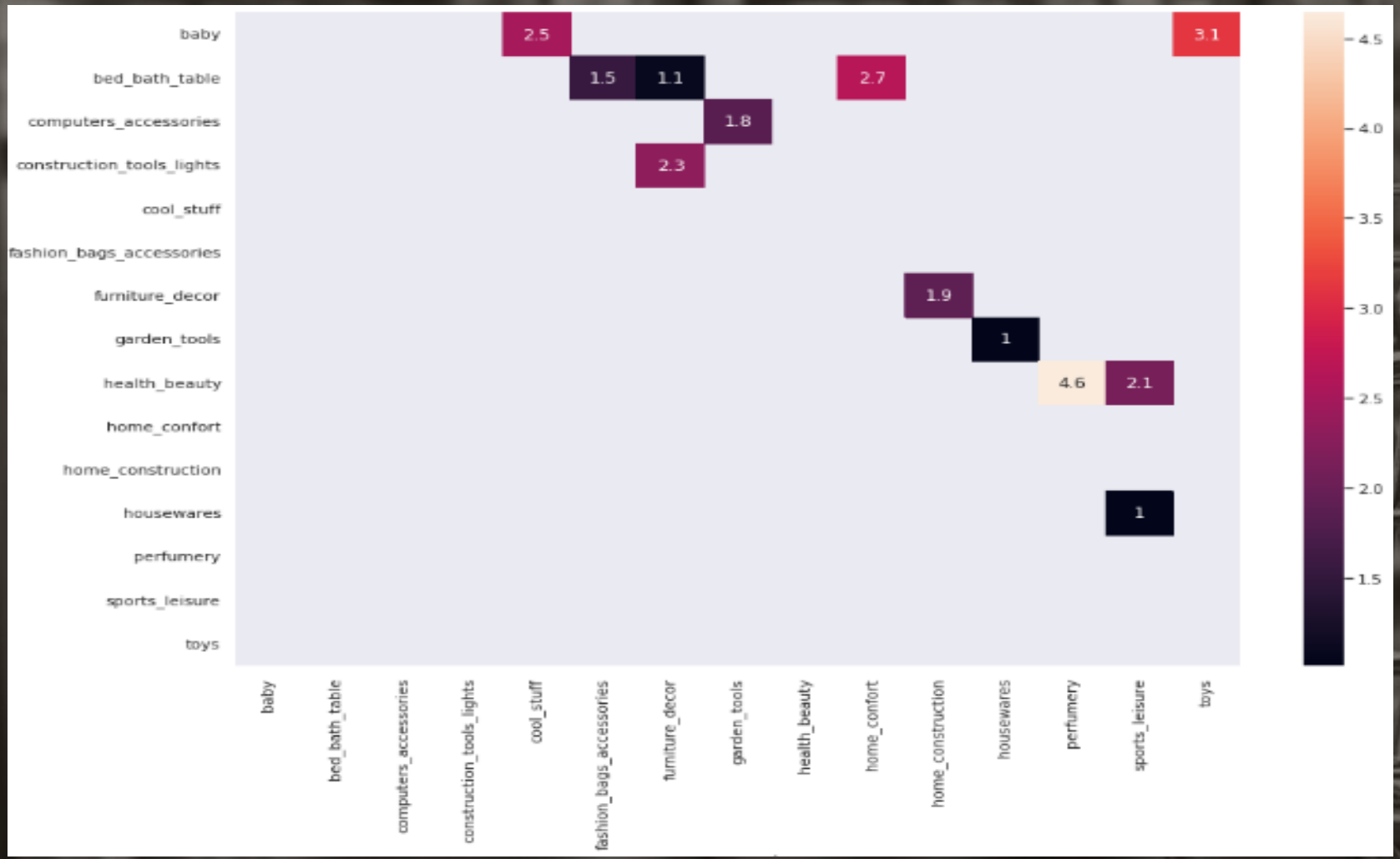
	miss_rate
category	1.394843

Cột category bị khuyết dữ liệu nhưng không nhiều, khoảng 1.4% nằm trong ngưỡng cho phép (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.

category	agro_industry_and_commerce	air_conditioning	art	arts_and_craftmanship
order_id				
002f98c0f7efd42638ed6100ca699b42	0	0	0	0
005d9a5423d47281ac463a968b3936fb	0	0	0	0
014405982914c2cde2796ddcf0b8703d	0	0	0	0
01b1a7fdae9ad1837d6ab861705a1fa5	0	0	0	0
01cce1175ac3c4a450e3a0f856d02734	0	0	0	0
...	...	...	...	...
fe678293ea3bb6607a15b2e320e91722	0	0	0	0
ff00a56fe9475a175cd651d77c707a09	0	0	0	0
ff40f38705c95a8eceea1a0db29bff66	0	0	0	0
ffa5e4c604dea4f0a59d19cc2322ac19	0	0	0	0
ffb8f7de8940249a3221252818937ecb	0	0	0	0

Market Basket Analysis

Luật kết hợp theo danh mục hàng hóa

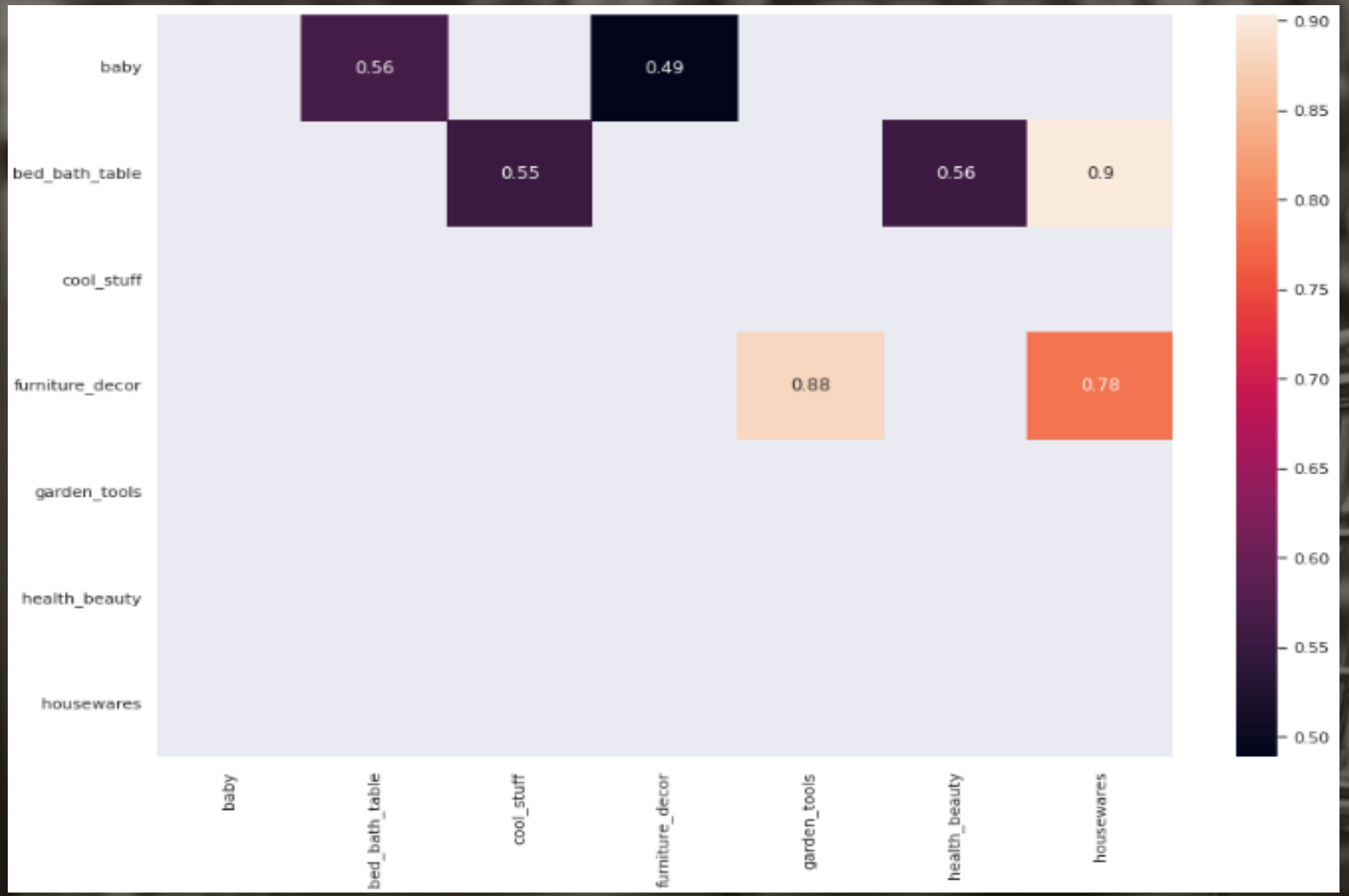


Lift > 1: Mặt hàng bổ sung

=> nên xuất hiện cùng nhau.

Market Basket Analysis

Luật kết hợp theo danh mục hàng hóa



Lift < 1: Mặt hàng thay thế

=> không nên xuất hiện cùng nhau.

Market Basket Analysis

Luật kết hợp theo sản phẩm

{health_beauty = > perfumery}

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
13	(189e539d996a9b8ba4bba1a140a024a7)	(a669398f595527fc03acc1ebda6b3cce)	0.007937	0.007937	0.007937	1.000000	126.0
14	(a669398f595527fc03acc1ebda6b3cce)	(189e539d996a9b8ba4bba1a140a024a7)	0.007937	0.007937	0.007937	1.000000	126.0
17	(d8c0c707d3724304033f593878cbf1e6)	(2ae501b303a5a8e6f75c8c36f366b2d5)	0.007937	0.007937	0.007937	1.000000	126.0
18	(2ae501b303a5a8e6f75c8c36f366b2d5)	(d8c0c707d3724304033f593878cbf1e6)	0.007937	0.007937	0.007937	1.000000	126.0
22	(3e7d7087ff8bbc0e9568b56ba3504a34)	(8ee57a1f636eb2e009706bbdb0818ecc)	0.007937	0.007937	0.007937	1.000000	126.0
23	(8ee57a1f636eb2e009706bbdb0818ecc)	(3e7d7087ff8bbc0e9568b56ba3504a34)	0.007937	0.007937	0.007937	1.000000	126.0
35	(e2f1ccf86759df28dd1e9f2e0e3242d4)	(eb9b44e05684527fbfdfd0ff5cb86250)	0.007937	0.007937	0.007937	1.000000	126.0
36	(eb9b44e05684527fbfdfd0ff5cb86250)	(e2f1ccf86759df28dd1e9f2e0e3242d4)	0.007937	0.007937	0.007937	1.000000	126.0
25	(521527593ca1726b992318e034dd5690)	(a25583531530c0913ea4dee2c5c73685)	0.007937	0.011905	0.007937	1.000000	84.0
26	(a25583531530c0913ea4dee2c5c73685)	(521527593ca1726b992318e034dd5690)	0.011905	0.007937	0.007937	0.666667	84.0

Market Basket Analysis

Luật kết hợp theo sản phẩm

{bed_bath_table = > home_confront}

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(fb783e3e545937820b57fe539b2c5a6c)	(0fa81e7123fd0ebe03adbbe99d912827)	0.005908	0.013294	0.005908	1.000000	75.222222
4	(84f456958365164420cfc80fbe4c7fab)	(64fb265487de2238627ce43fe8a67efc)	0.010340	0.008863	0.005908	0.571429	64.476190
5	(64fb265487de2238627ce43fe8a67efc)	(84f456958365164420cfc80fbe4c7fab)	0.008863	0.010340	0.005908	0.666667	64.476190
9	(f4d705aa95ccca448e5b0deb6e5290ba)	(c211ff3068fcd2f8898192976d8b3a32)	0.010340	0.010340	0.005908	0.571429	55.265306
10	(c211ff3068fcd2f8898192976d8b3a32)	(f4d705aa95ccca448e5b0deb6e5290ba)	0.010340	0.010340	0.005908	0.571429	55.265306
6	(ad0a798e7941f3a5a2fb8139cb62ad78)	(946344697156947d846d27fe0d503033)	0.013294	0.014771	0.008863	0.666667	45.133333
7	(946344697156947d846d27fe0d503033)	(ad0a798e7941f3a5a2fb8139cb62ad78)	0.014771	0.013294	0.008863	0.600000	45.133333
3	(4d0ec1e9b95fb62f9a1fbe21808bf3b1)	(9ad75bd7267e5c724cb42c71ac56ca72)	0.013294	0.019202	0.008863	0.666667	34.717949
1	(35afc973633aaeb6b877ff57b2793310)	(99a4788cb24856965c36a24e339b6058)	0.053176	0.070901	0.042836	0.805556	11.361690
2	(99a4788cb24856965c36a24e339b6058)	(35afc973633aaeb6b877ff57b2793310)	0.070901	0.053176	0.042836	0.604167	11.361690

Market Basket Analysis

Luật kết hợp theo sản phẩm

{bed_bath_table = > home_confront}

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(fb783e3e545937820b57fe539b2c5a6c)	(0fa81e7123fd0ebe03adbbe99d912827)	0.005908	0.013294	0.005908	1.000000	75.222222
4	(84f456958365164420cfc80fbe4c7fab)	(64fb265487de2238627ce43fe8a67efc)	0.010340	0.008863	0.005908	0.571429	64.476190
5	(64fb265487de2238627ce43fe8a67efc)	(84f456958365164420cfc80fbe4c7fab)	0.008863	0.010340	0.005908	0.666667	64.476190
9	(f4d705aa95ccca448e5b0deb6e5290ba)	(c211ff3068fcd2f8898192976d8b3a32)	0.010340	0.010340	0.005908	0.571429	55.265306
10	(c211ff3068fcd2f8898192976d8b3a32)	(f4d705aa95ccca448e5b0deb6e5290ba)	0.010340	0.010340	0.005908	0.571429	55.265306
6	(ad0a798e7941f3a5a2fb8139cb62ad78)	(946344697156947d846d27fe0d503033)	0.013294	0.014771	0.008863	0.666667	45.133333
7	(946344697156947d846d27fe0d503033)	(ad0a798e7941f3a5a2fb8139cb62ad78)	0.014771	0.013294	0.008863	0.600000	45.133333
3	(4d0ec1e9b95fb62f9a1fbe21808bf3b1)	(9ad75bd7267e5c724cb42c71ac56ca72)	0.013294	0.019202	0.008863	0.666667	34.717949
1	(35afc973633aaeb6b877ff57b2793310)	(99a4788cb24856965c36a24e339b6058)	0.053176	0.070901	0.042836	0.805556	11.361690
2	(99a4788cb24856965c36a24e339b6058)	(35afc973633aaeb6b877ff57b2793310)	0.070901	0.053176	0.042836	0.604167	11.361690

Market Basket Analysis

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.

ADDITION

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

Tiền xử lý dữ liệu

- Chuẩn bị dữ liệu ✓
- Tìm và xử lý mising value ✓
- Xóa cột dữ liệu có tương quan cao ✓
- Scale dữ liệu bằng MinMaxScaler ✓

ADDITION

Demand Prediction Model

Mô hình ban đầu

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

```
MSE       = 19896.55
R-squared = 0.8021
```

Mô hình 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.

```
R-squared = 0.8932
MSE       = 10734.82
```

ADDITION

Demand Prediction Model

Kết luận & Hướng phát triển

- R^2 ở mô hình mới lớn hơn mô hình cũ, ta có thể kết luận mô hình mới tốt hơn mô hình cũ.
- 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.

7

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.

Một số tài liệu liên quan

Data soucre

Project Documentation

Source code

Query (SQL)

Raw data

(Data after quering)

Cohort Analysis using SQL