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

Tài liệu tham khảo

https://en.wikipedia.org/wiki/Association_rule_learning

https://goldinlocks.github.io/Market-Basket-Analysis-in-Python/

https://www.activestate.com/blog/cohort-analysis-with-python/

https://towardsdatascience.com/an-rfm-customer-segmentation-with-python-cf7be647733d

https://laptrinhx.com/a-gentle-introduction-to-customer-segmentation-with-rfm-scores-4172308924/

Source code

1. **Dataset**
2. **Source code**
3. **Report**

1. 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.

2. Cách tiếp cận vấn đề

Thinking flow

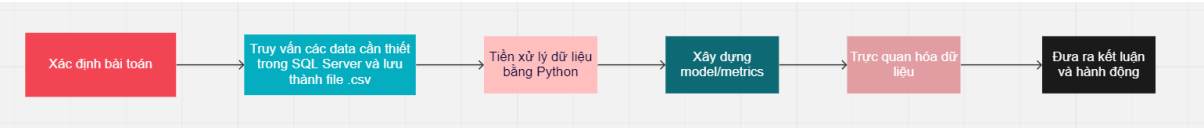
Xây dựng metrics retention rate để xem xét sự trung thành của khách hàng đối với hệ thống. Nếu tỷ lệ khách hàng rời bỏ lớn, thực hiện Root Cause Analysis để tìm hiểu nguyên nhân và đưa ra hướng khắc phục.

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.



3. Giới thiệu tổng quan về nguồn dữ liệu

Tổng quan:

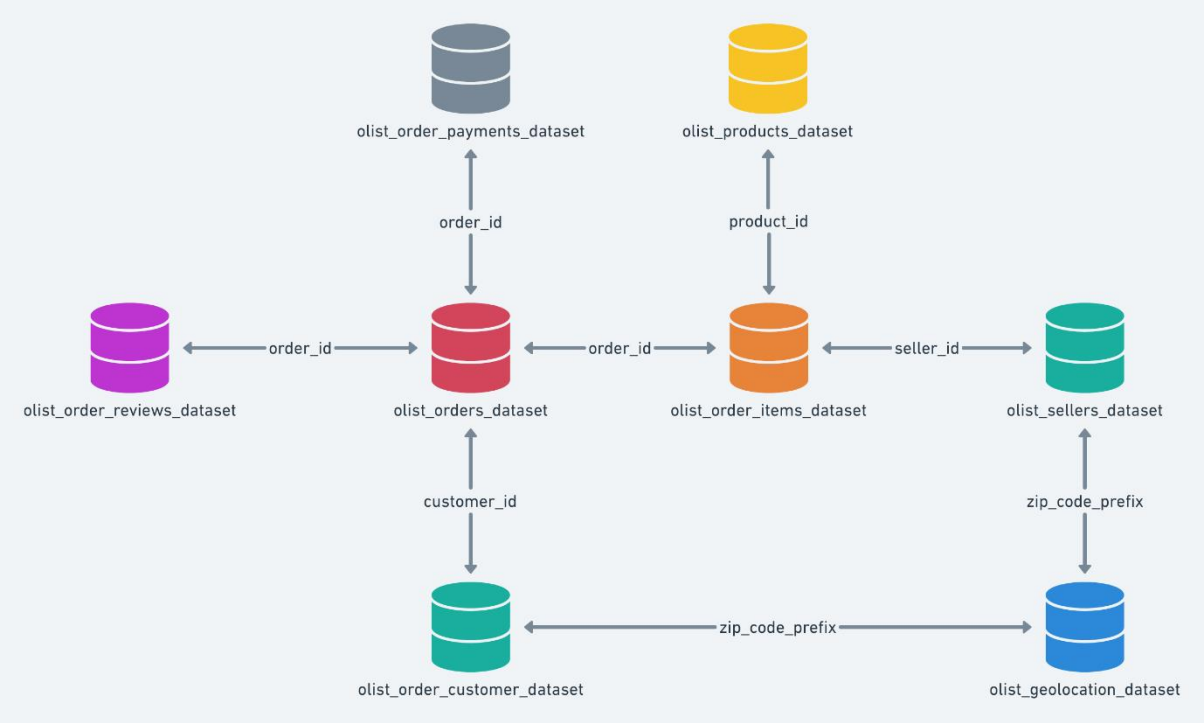
Bộ dữ liệu liên quan đến thương mại điện tử (e-commerce) của cửa hàng Olist (Brazil) với hơn 100000 đơn hàng được ghi nhận từ năm 2016-2018 và một số thông tin liên quan khác về khách hàng, sản phẩm, thanh toán,… được đính kèm.

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.

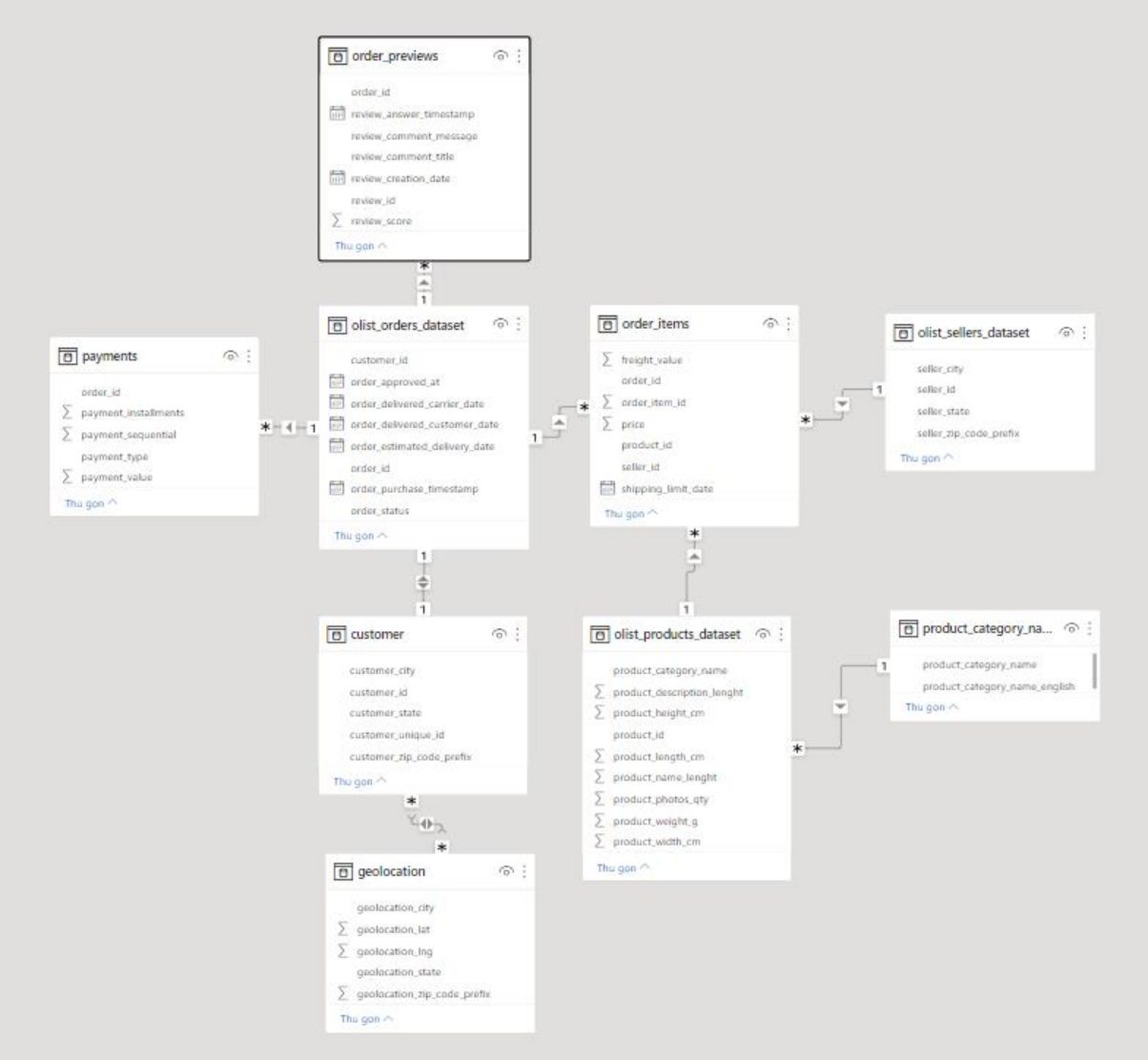
Nguồn dữ liệu: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

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.



4. 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

	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
count	99441	99441	99441	99441	99441
unique	99441	96096	14994	4119	27
top	06b8999e2fba1a1fbc88172c00ba8bc7	8d50f5eadf50201ccdcedfb9e2ac8455	22790	sao paulo	SP
freq	1	17	142	15540	41746

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

	seller_id	seller_zip_code_prefix	seller_city	seller_state
count	3095	3095	3095	3095
unique	3095	2246	611	23
top	3442f8959a84dea7ee197c632cb2df15	14940	sao paulo	SP
freq	1	49	694	1849

Order_items

Tên cột	Mô tả	Kiểu dữ liệu
order_id	Id đơn hàng	Categorical
order_item_id	Id cho từng sản phẩm 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

	price	freight_value
count	112650.000000	112650.000000
mean	120.653739	19.990320
std	183.633928	15.806405
min	0.850000	0.000000
25%	39.900000	13.080000
50%	74.990000	16.260000
75%	134.900000	21.150000
max	6735.000000	409.680000

	order_id	order_item_id	product_id	seller_id	shipping_limit_date
count	112650	112650	112650	112650	112650
unique	98666	98666	32951	3095	93318
top	8272b63d03f5f79c56e9e4120aec44ef	8272b63d03f5f79c56e9e4120aec44ef	aca2eb7d00ea1a7b8ebd4e68314663af	6560211a19b47992c3666cc44a7e94c0	2017-07-21 18:25:23
freq	21	21	527	2033	21

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

	payment_value
count	103886.000000
mean	154.100380
std	217.494064
min	0.000000
25%	56.790000
50%	100.000000
75%	171.837500
max	13664.080000

	order_id	payment_sequential	payment_type	payment_installments
count	103886	103886	103886	103886
unique	99440	29	5	24
top	fa65dad1b0e818e3ccc5cb0e39231352	1	credit_card	1
freq	29	99360	76795	52546

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_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
count	99441	99441	99441	99441	99281	97658	96476	99441
unique	99441	99441	8	98875	90733	81018	95664	459
top	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2018-04-11 10:48:14	2018-02-27 04:31:10	2018-05-09 15:48:00	2018-05-08 23:38:46	2017-12-20 00:00:00
freq	1	1	96478	3	9	47	3	522

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

	review_score
count	99224.000000
mean	4.086421
std	1.347579
min	1.000000
25%	4.000000
50%	5.000000
75%	5.000000
max	5.000000

	review_id	order_id	review_comment_title	review_comment_message	review_creation_date	review_answer_timestamp
count	99224	99224	11568	40977	99224	99224
unique	98410	98673	4527	36159	636	98248
top	7b606b0d57b078384f0b58eac1d41d78	c88b1d1b157a9999ce368f218a407141	Recomendo	Muito bom	2017-12-19 00:00:00	2017-06-15 23:21:05
freq	3	3	423	230	463	4

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

	product_name_lenght	product_description_lenght	product_photos_qty	product_weight_g	product_length_cm	product_height_cm	product_width_cm
count	32341.000000	32341.000000	32341.000000	32949.000000	32949.000000	32949.000000	32949.000000
mean	48.476949	771.495285	2.188986	2276.472488	30.815078	16.937661	23.196728
std	10.245741	635.115225	1.736766	4282.038731	16.914458	13.637554	12.079047
min	5.000000	4.000000	1.000000	0.000000	7.000000	2.000000	6.000000
25%	42.000000	339.000000	1.000000	300.000000	18.000000	8.000000	15.000000
50%	51.000000	595.000000	1.000000	700.000000	25.000000	13.000000	20.000000
75%	57.000000	972.000000	3.000000	1900.000000	38.000000	21.000000	30.000000
max	76.000000	3992.000000	20.000000	40425.000000	105.000000	105.000000	118.000000

	product_id	product_category_name
count	32951	32341
unique	32951	73
top	1e9e8ef04dbcff4541ed26657ea517e5	cama_mesa_banho
freq	1	3029

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

	geolocation_lat	geolocation_lng
count	1.000163e+06	1.000163e+06
mean	-2.117615e+01	-4.639054e+01
std	5.715866e+00	4.269748e+00
min	-3.660537e+01	-1.014668e+02
25%	-2.360355e+01	-4.857317e+01
50%	-2.291938e+01	-4.663788e+01
75%	-1.997962e+01	-4.376771e+01
max	4.506593e+01	1.211054e+02

	geolocation_zip_code_prefix	geolocation_city	geolocation_state
count	1000163	1000163	1000163
unique	19015	8011	27
top	24220	sao paulo	SP
freq	1146	135800	404268

5. 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
```

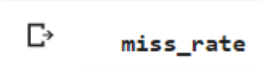
Giải thích:

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:

	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

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).

```
count    93358.000000
mean         1.033420
std          0.209097
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         15.000000
Name: frequency, dtype: float64
```

Kết quả thu được là

	customer_id	frequency	first_order_date	last_order_date	monetary	recency	RecencyScore	MonetaryScore	morethan_1
0	0000366f3b9a7992bf8c76cfdf3221e2	1	2018-05-10	2018-05-10	14190	111	4	2	False
1	0000b849f77a49e4a4ce2b2a4ca5be3f	1	2018-05-07	2018-05-07	2719	114	4	4	False
2	0000f46a3911fa3c0805444483337064	1	2017-03-10	2017-03-10	8622	537	1	3	False
3	0000f6ccb0745a6a4b88665a16c9f078	1	2017-10-12	2017-10-12	4362	321	2	4	False
4	0004aac84e0df4da2b147fca70cf8255	1	2017-11-14	2017-11-14	19689	288	2	1	False

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

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

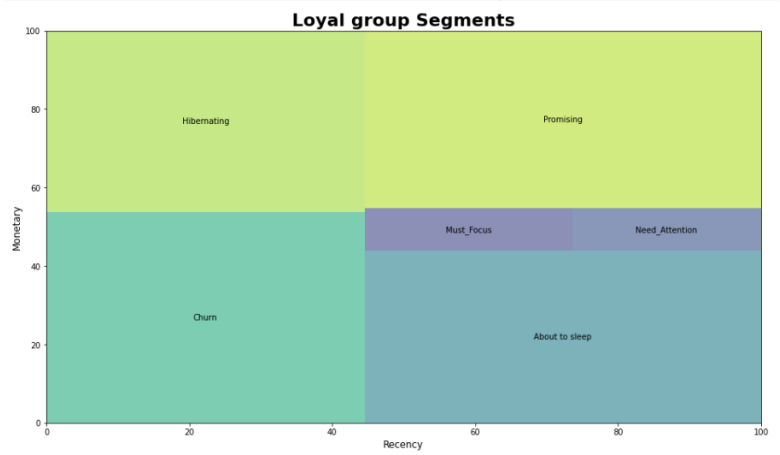
- Recency = 4 và Monetary = 3-4 : 'Must_Focus' (nên tập trung vào)
- Recency = 4 và Monetary = 1-2:'Promising' (tiềm năng)
- Recency = 3 và Monetary = 3-4:'Need_Attention' (cần hành động)
- Recency = 3 và Monetary = 1-2: 'About to sleep' (Sắp ngừng hoạt động)
- Recency = 2 và Monetary = 1-4: 'Hibernating' (đã ngừng hoạt động một thời gian)
- Recency = 1 và Monetary = 1-4:'Churn' (đã rời bỏ)

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

Trực quan hóa RM theo loyal và onetime

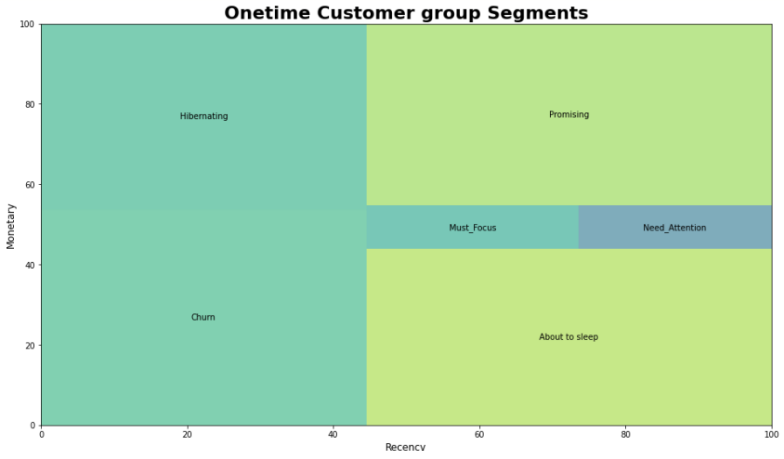
- Loyal

RM_group	recency mean	frequency mean	monetary mean	monetary 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



- Onetime

RM_group	recency mean	frequency mean	monetary mean	monetary count
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

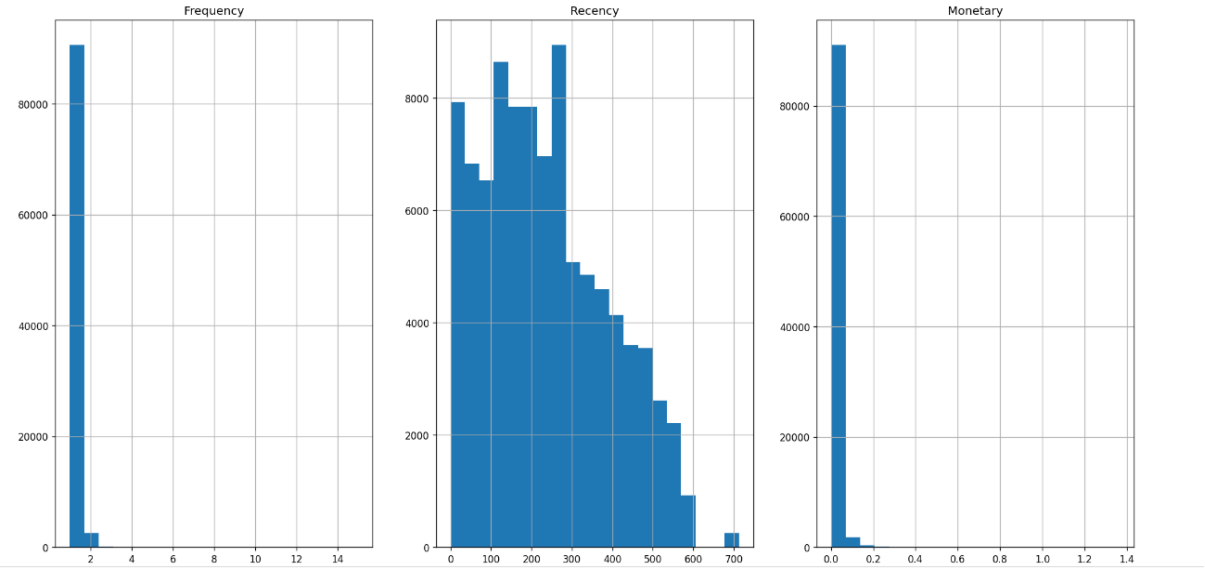


EDA

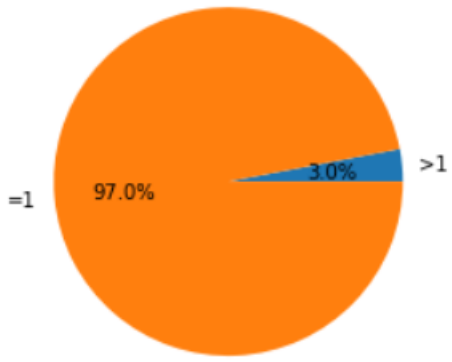
Thực hiện thống kê mô tả (Descriptive Analysis) trên các cột RFM.

	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

Tìm hiểu về phân phối của RFM.

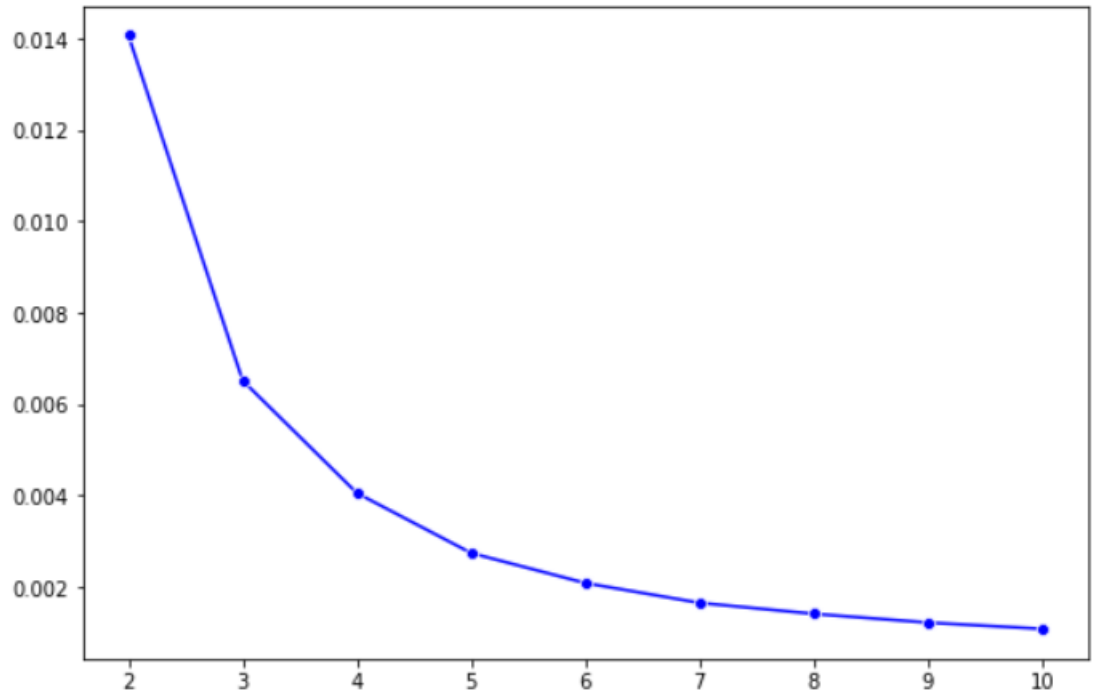


Dựa trên phân phối trên, ta dễ thấy rằng khách hàng thường chỉ mua một lần và không quay lại.



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).



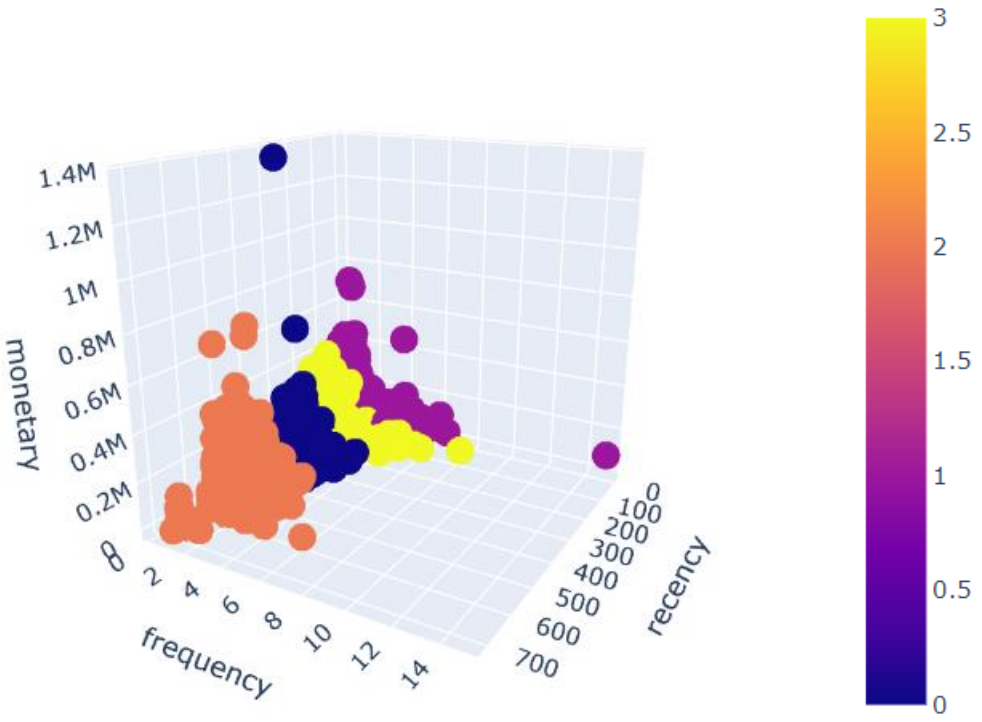
Số cụm tối ưu ở bài toán này (được tính theo wssd) có thể là 4 hoặc 5, ở bài toán này tôi sẽ chọn k = 4.

Phân cụm khách hàng:

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

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

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.

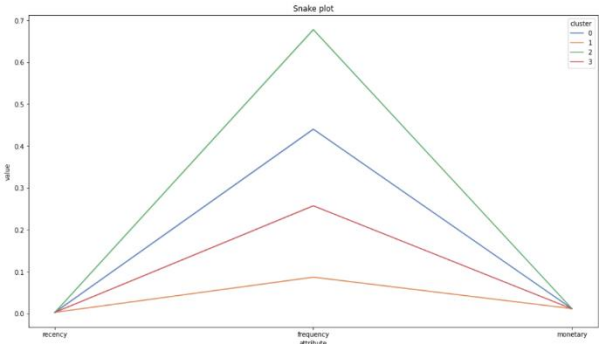


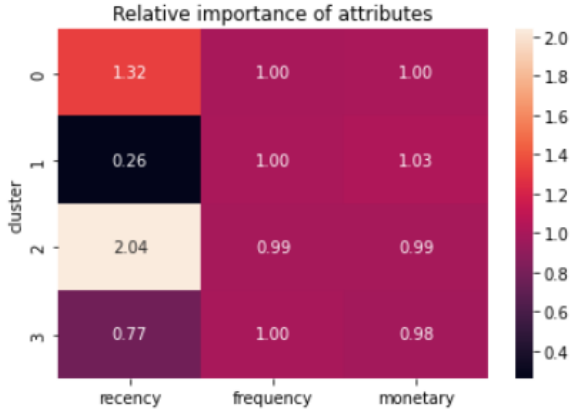
Ta đã có các cụm dữ liệu nhưng chưa biết tương quan giữa các chiều của nó, nên tiếp theo ta sẽ thực hiện phân tích hậu định để tìm ra tương quan giữa các chiều dữ liệu theo cụm.

Thống kê mô tả theo các cụm.

	cluster	recency				frequency				monetary			
	count	min	median	mean	max	min	median	mean	max	min	median	mean	max
cluster													
0	24301	249	307.0	313.831365	398	1	1.0	1.031768	4	1007	10687.0	16594.832229	1366408
1	25411	0	62.0	61.911731	123	1	1.0	1.038527	15	959	11082.0	16967.747983	727488
2	16553	399	475.0	483.372078	713	1	1.0	1.024890	6	1228	10501.0	16343.071528	757163
3	27093	123	183.0	183.428930	248	1	1.0	1.035323	9	1336	10800.0	16130.072343	417526

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.

6. 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.

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.

```
miss_rate
```

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).

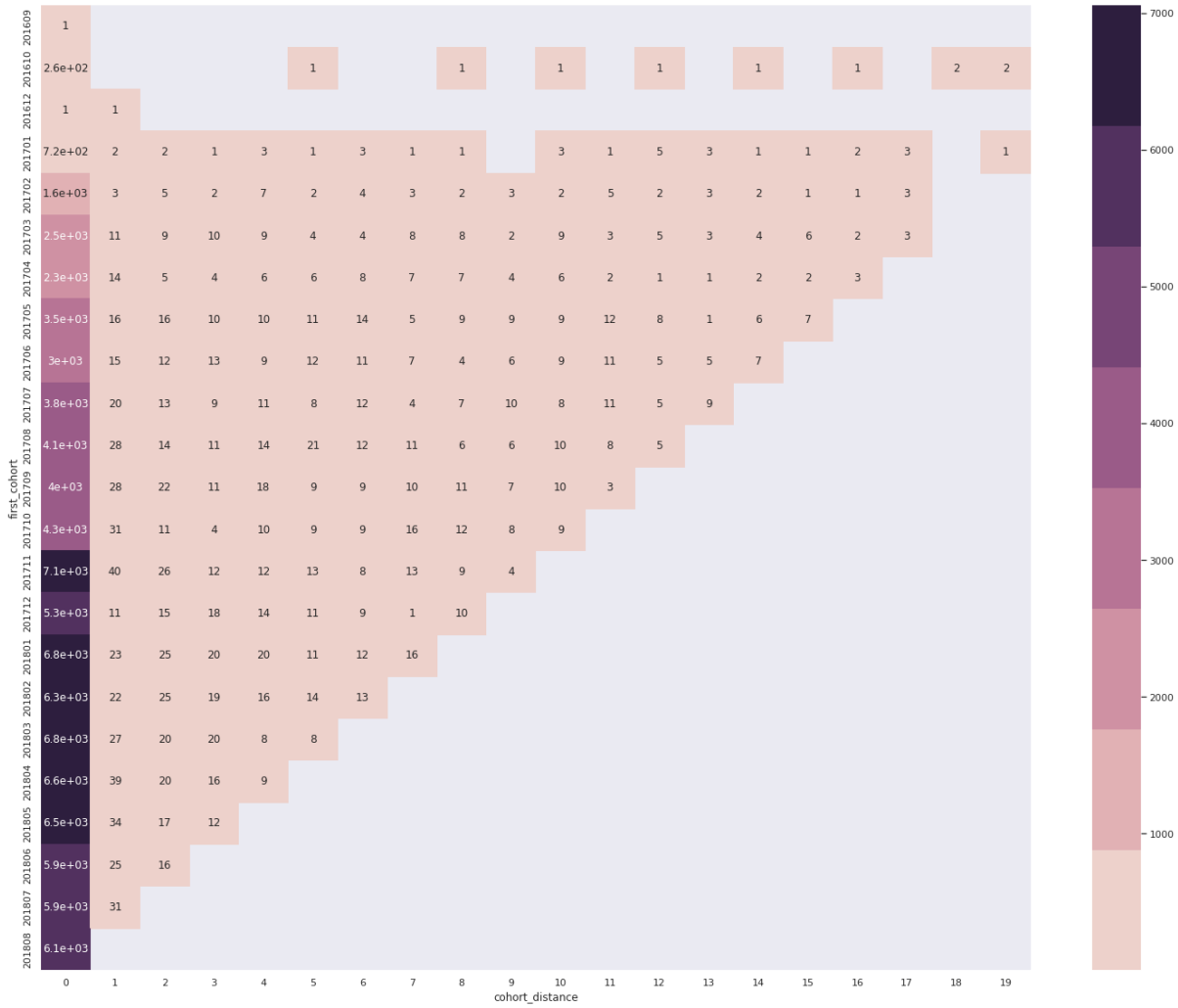
	order_id	customer_id	purchase_timestamp	value	datetime	month	year	cohort
0	00010242fe8c5a6d1ba2dd792cb16214	871766c5855e863f6eccc05f988b23cb	2017-09-13 08:59:02	7219	13/09/2017	09	2017	201709
1	00018f77f2f0320c557190d7a144bdd3	eb28e67c4c0b83846050ddfb8a35d051	2017-04-26 10:53:06	25983	26/04/2017	04	2017	201704
2	000229ec398224ef6ca0657da4fc703e	3818d81c6709e39d06b2738a8d3a2474	2018-01-14 14:33:31	21687	14/01/2018	01	2018	201801
3	00024acbcdf0a6daa1e931b038114c75	af861d436cfc08b2c2ddefd0ba074622	2018-08-08 10:00:35	2578	08/08/2018	08	2018	201808
4	00042b26cf59d7ce69dfabb4e55b4fd9	64b576fb70d441e8f1b2d7d446e483c5	2017-02-04 13:57:51	21804	04/02/2017	02	2017	201702

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.

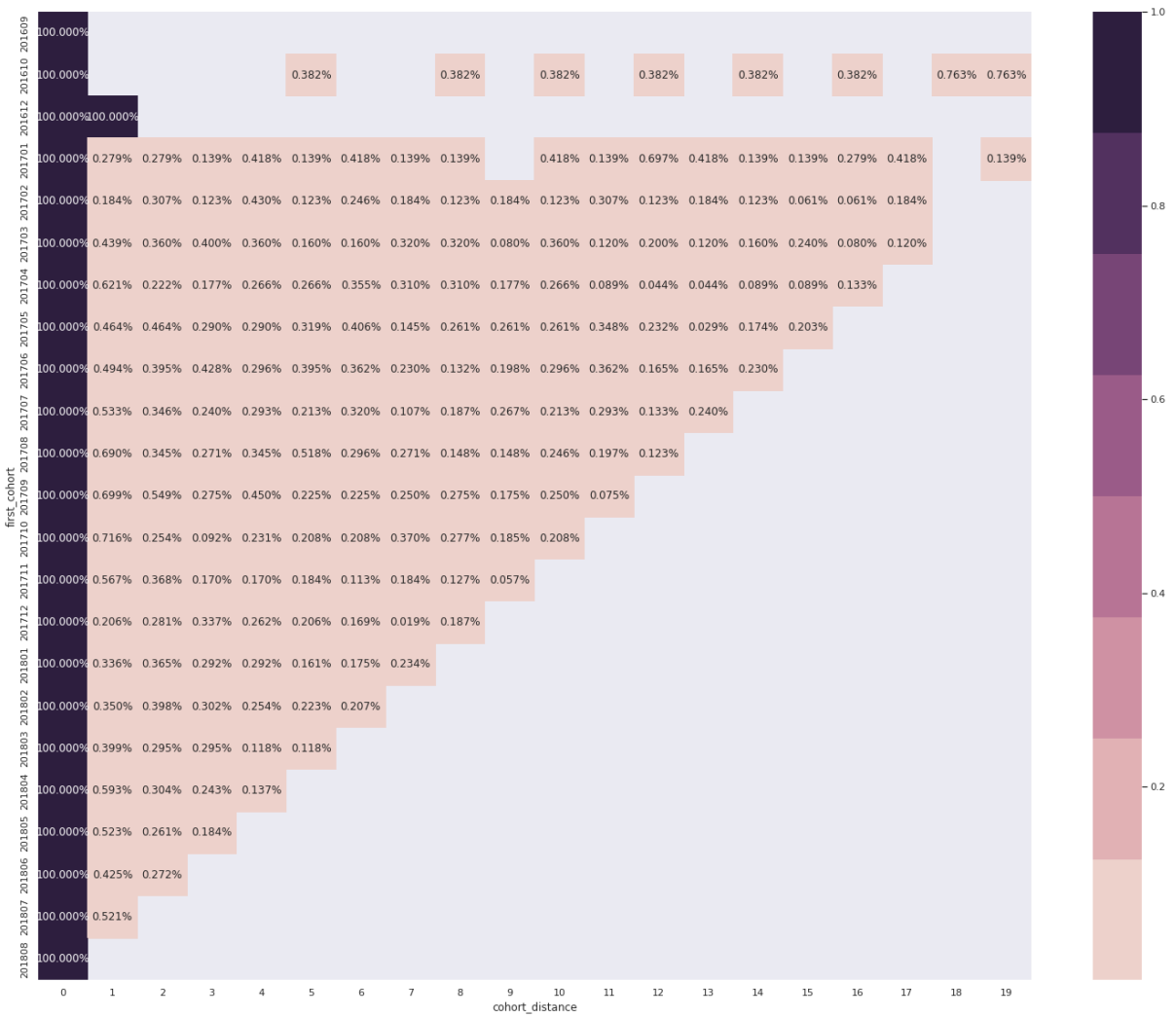
	order_id	customer_id	purchase_timestamp	value	datetime	month	year	cohort	first_cohort	cohort_distance
0	00010242fe8c5a6d1ba2dd792cb16214	871766c5855e863f6eccc05f988b23cb	2017-09-13 08:59:02	7219	13/09/2017	09	2017	201709	201709	0
1	00018f77f2f0320c557190d7a144bdd3	eb28e67c4c0b83846050ddfb8a35d051	2017-04-26 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

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

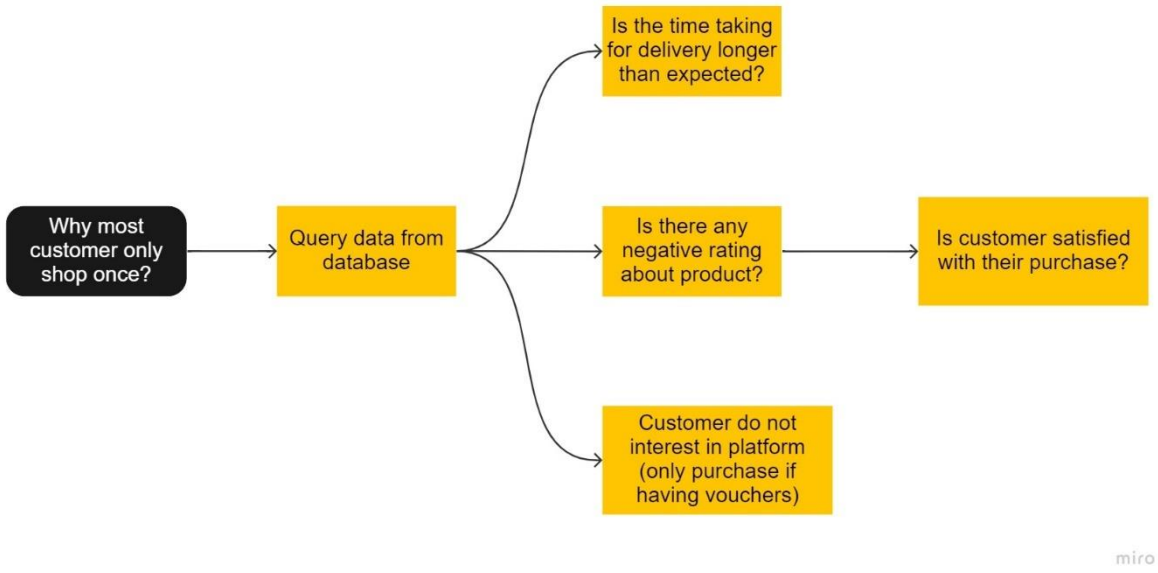
Khách mua hàng rất nhiều nhưng chủ yếu chỉ mua một lần, khách mua một lần chiếm đến 97% tổng lượng khách. Lượng khách quay lại có xu hướng giảm dần theo thời gian. Ở phần tiếp theo, tôi sẽ tìm hiểu nguyên nhân tại sao dẫn đến việc khách chỉ mua một lần và không quay lại.

7. 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

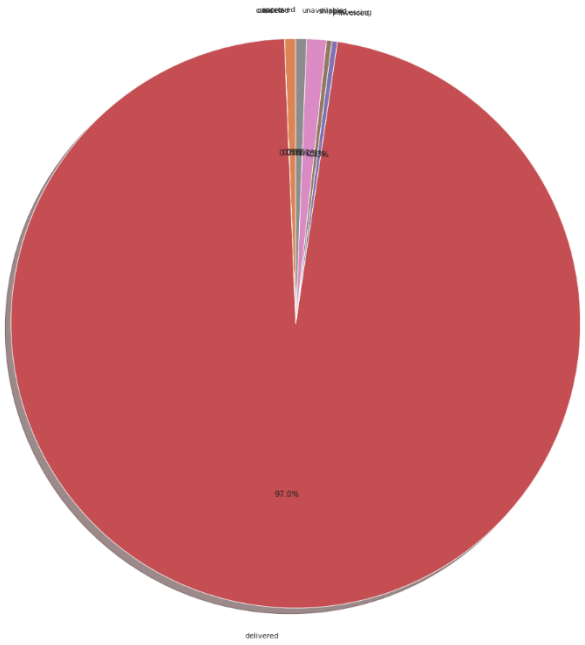
Phần lớn các phương pháp được sử dụng là EDA và kiểm định giả thuyết để đưa ra kết luận.

Shipment

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

	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

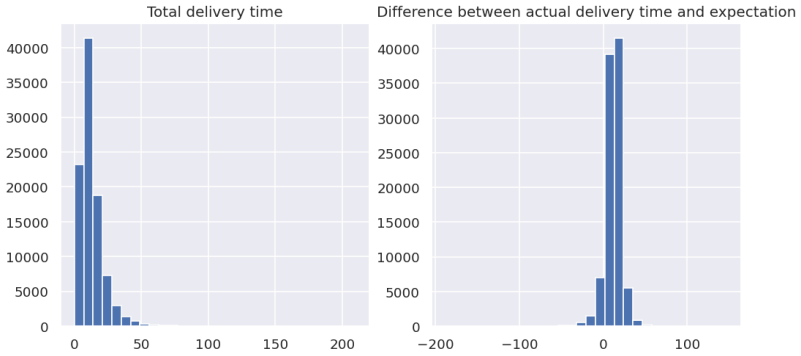
Xem xét tỉ lệ giao hàng thành công. Có tới 97% đơn hàng đã được giao thành công

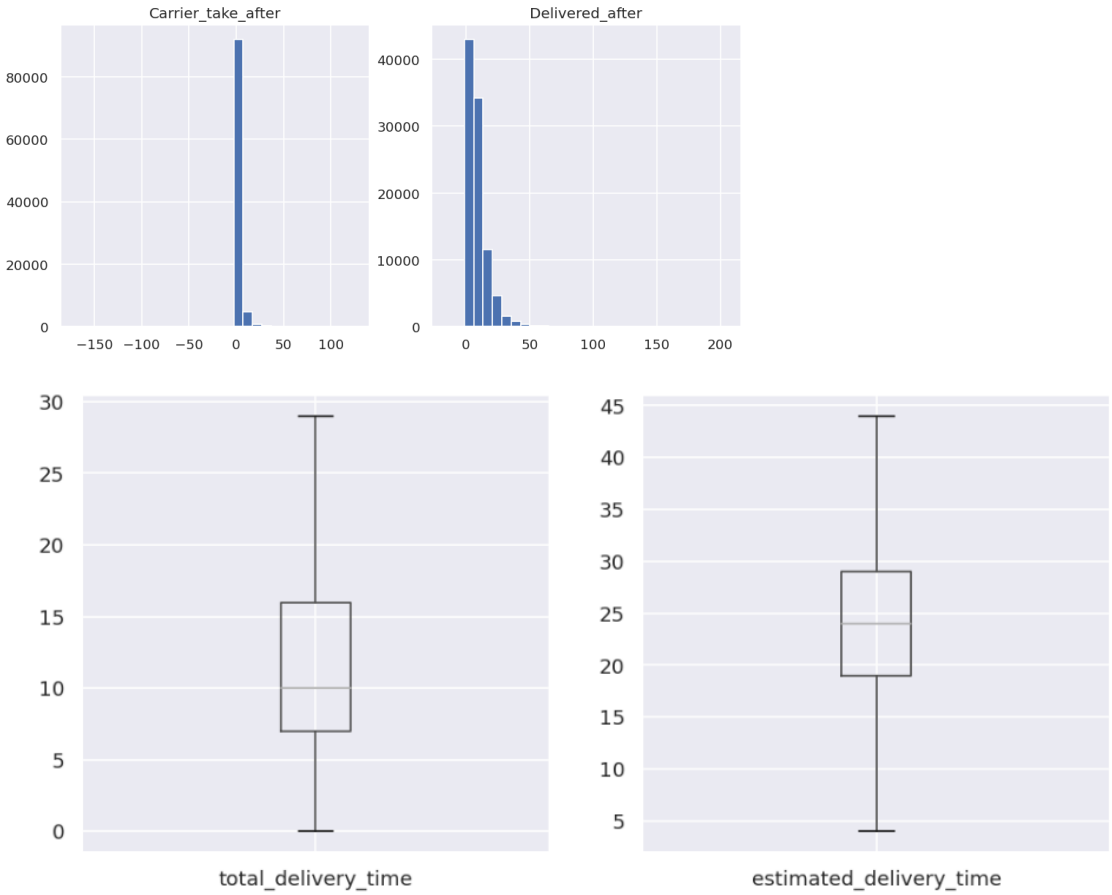


Thực hiện thống kê mô tả ở các cột dữ liệu dạng số.

	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? 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

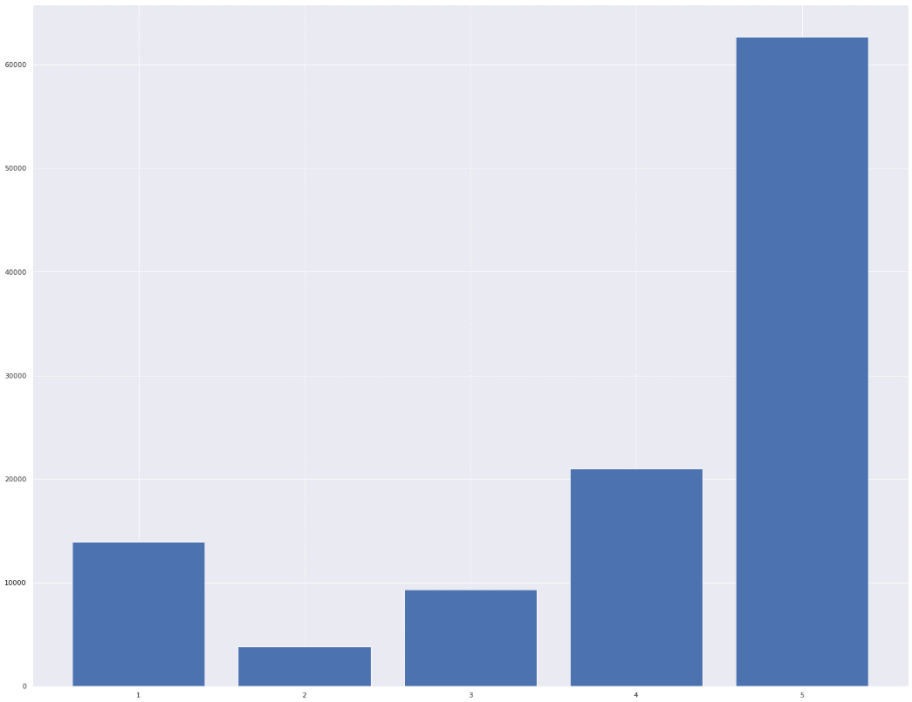
```
Ttest_indResult(statistic=-284.582619728147, pvalue=0.0)
```

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.

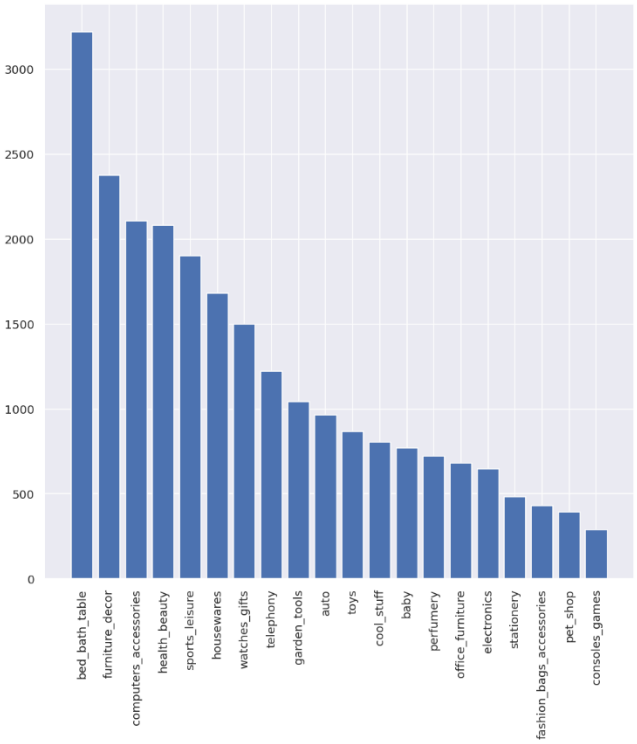
Rating

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.

	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



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.

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

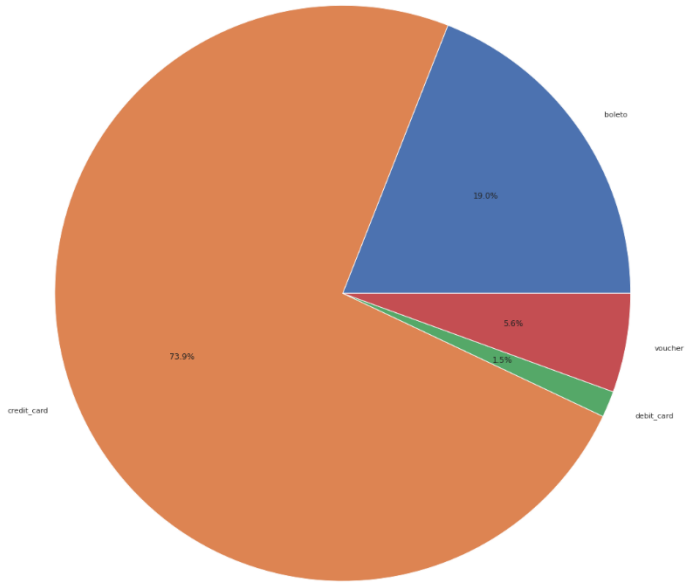
Với mức ý nghĩa là 95%, α = 0.05: p-value < α nên ta bác bỏ H0. Ta có thể kết luận rằng tuy lượng đánh giá cao nhiều, nhưng với mục tiêu làm hài lòng hơn 80% đơn hàng thì hệ thống

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.

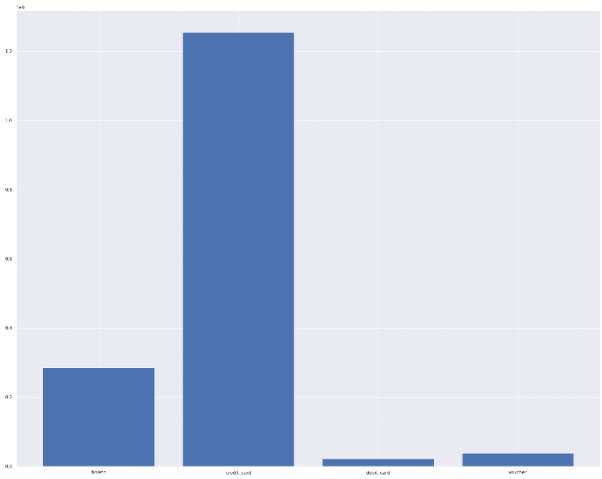
Payment

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

```
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.

- 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.

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.

8. 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.

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

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

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

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

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

- 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.

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

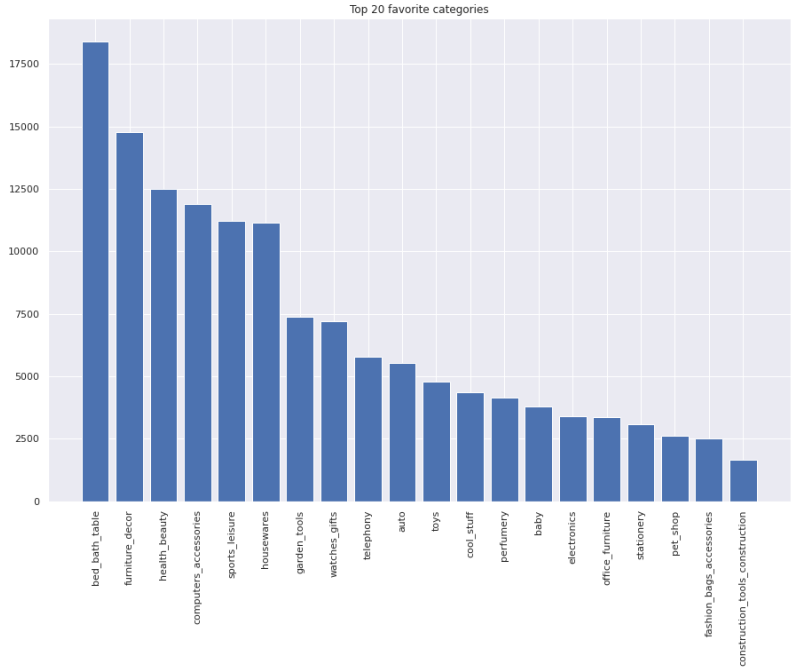
Tiền xử lý dữ liệu

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.

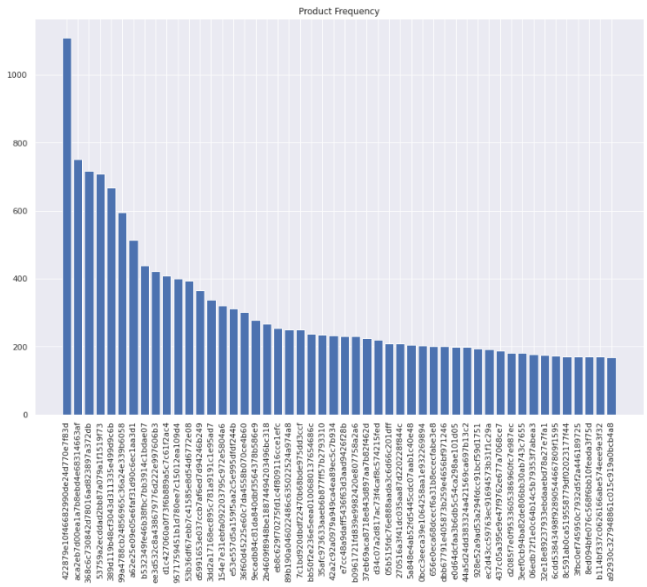
	miss_rate
category	1.394843

EDA

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



Tìm ra các sản phẩm được mua nhiều nhất.



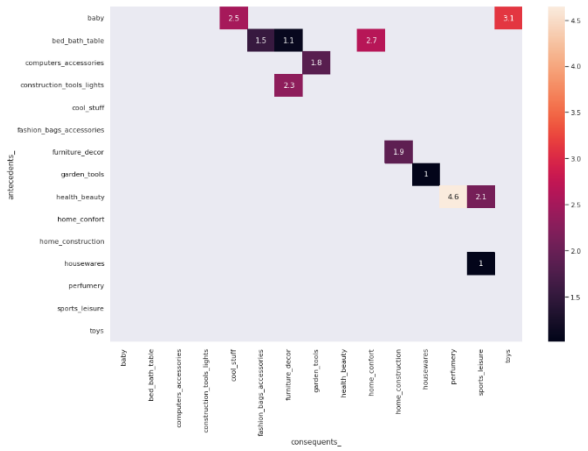
Luật kết hợp theo danh mục hàng

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.

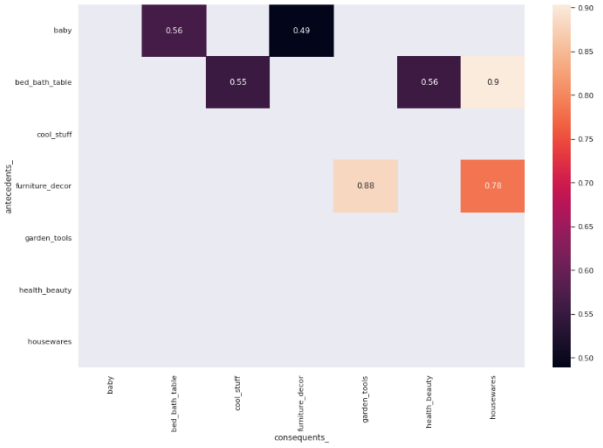
category	agro_industry_and_commerce	air_conditioning	art	arts_and_craftmanship	audio	auto	baby	bed_bath_table
order_id								
002f98c0f7efd42638ed6100ca699b42	0	0	0	0	0	0	0	0
005d9a5423d47281ac463a968b3936fb	0	0	0	0	0	0	1	0
014405982914c2cde2796ddcf0b8703d	0	0	0	0	0	0	0	0
01b1a7fdae9ad1837d6ab861705a1fa5	0	0	0	0	0	0	0	1
01cce1175ac3c4a450e3a0f856d02734	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
fe678293ea3bb6607a15b2e320e91722	0	0	0	0	0	0	0	0
ff00a56fe9475a175cd651d77c707a09	0	0	0	0	0	0	0	1
ff40f38705c95a8eceea1a0db29bff66	0	0	0	0	1	0	0	0
ffa5e4c604dea4f0a59d19cc2322ac19	0	0	0	0	0	0	0	1
ffb8f7de8940249a3221252818937ecb	0	0	0	0	0	1	0	0

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}

	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

{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

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.

9. 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

antecedents_	bed_bath_table	category
fashion_bags_accessories	1.541502	fashion_bags_accessories
furniture_decor	1.092521	furniture_decor
home_confort	2.651383	home_confort
baby	0.563560	baby
cool_stuff	0.552180	cool_stuff
health_beauty	0.556653	health_beauty
housewares	0.903639	housewares

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

	category	month_year	no_items_sold	lift	lift_items
105	baby	01/2017	57	0.563560	32.122912
106	baby	01/2018	242	0.563560	136.381487
107	baby	02/2017	37	0.563560	20.851715
108	baby	02/2018	186	0.563560	104.822134
109	baby	03/2017	45	0.563560	25.360194

Thực hiện pivot table với index là month_year, columns là category, values là lift_items với aggfunc là sum (tổng lift_items)

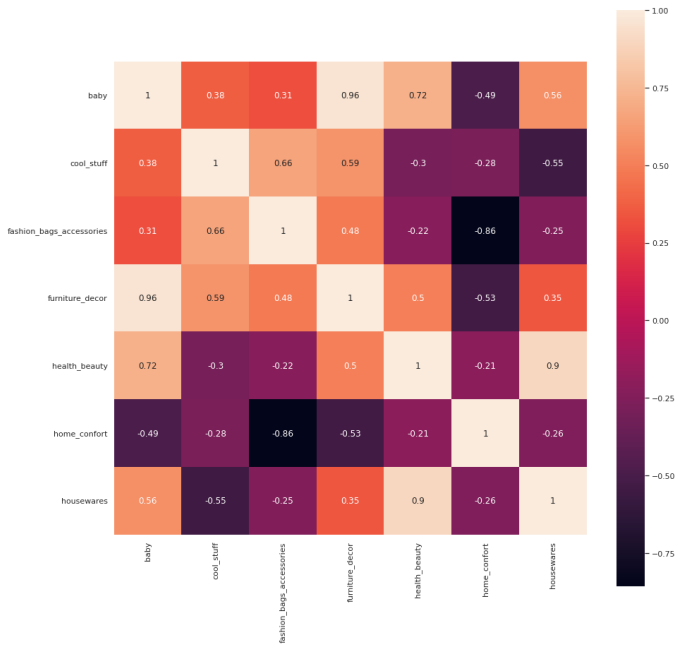
Tìm y (output) cho bài toán học có giám sát (supervised learning). Ở case này y là số lượng sản phẩm thuộc danh mục bed_bath_table được bán ra mỗi tháng.

	category	month_year	no_items_sold
126	bed_bath_table	01/2017	68
127	bed_bath_table	01/2018	1356
128	bed_bath_table	02/2017	266
129	bed_bath_table	02/2018	952
130	bed_bath_table	03/2017	417

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

	month_year	baby	cool_stuff	fashion_bags_accessories	furniture_decor	health_beauty	home_confort	housewares	date	category	no_items_sold
0	01/2017	32.122912	28.713350	61.660079	300.443225	48.985507	NaN	34.338285	01/2017	bed_bath_table	68.0
1	01/2018	136.381487	176.145360	215.810277	926.457654	396.893939	53.027668	385.853891	01/2018	bed_bath_table	1356.0
2	02/2017	20.851715	39.204767	67.826087	356.161787	100.197628	7.954150	94.882104	02/2017	bed_bath_table	266.0
3	02/2018	104.822134	106.570704	175.731225	621.644346	424.726614	60.981818	454.530462	02/2018	bed_bath_table	952.0
4	03/2017	25.360194	69.574656	114.071146	462.136306	133.040184	39.770751	225.006133	03/2017	bed_bath_table	417.0
5	03/2018	131.309448	154.610347	234.308300	954.863195	410.810277	58.330435	489.772387	03/2018	bed_bath_table	1138.0

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

	month_year	baby	cool_stuff	fashion_bags_accessories	health_beauty	date	category	no_items_sold
1	01/2018	136.381487	176.145360	215.810277	396.893939	01/2018	bed_bath_table	1356.0
2	02/2017	20.851715	39.204767	67.826087	100.197628	02/2017	bed_bath_table	266.0
3	02/2018	104.822134	106.570704	175.731225	424.726614	02/2018	bed_bath_table	952.0
4	03/2017	25.360194	69.574656	114.071146	133.040184	03/2017	bed_bath_table	417.0
5	03/2018	131.309448	154.610347	234.308300	410.810277	03/2018	bed_bath_table	1138.0

Tiếp theo sử dụng MinMaxScaler để scale các biến x

	month_year	baby	cool_stuff	fashion_bags_accessories	health_beauty	date	category	no_items_sold
1	01/2018	0.706897	0.995984	0.545455	0.641396	01/2018	bed_bath_table	1356.0
2	02/2017	0.000000	0.000000	0.000000	0.000000	02/2017	bed_bath_table	266.0
3	02/2018	0.513793	0.489960	0.397727	0.701564	02/2018	bed_bath_table	952.0
4	03/2017	0.027586	0.220884	0.170455	0.070999	03/2017	bed_bath_table	417.0
5	03/2018	0.675862	0.839357	0.613636	0.671480	03/2018	bed_bath_table	1138.0

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

	baby	cool_stuff	fashion_bags_accessories	health_beauty	no_items_sold	month
1	0.706897	0.995984	0.545455	0.641396	1356.0	1
2	0.000000	0.000000	0.000000	0.000000	266.0	2
3	0.513793	0.489960	0.397727	0.701564	952.0	2
4	0.027586	0.220884	0.170455	0.070999	417.0	3
5	0.675862	0.839357	0.613636	0.671480	1138.0	3

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

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

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.

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

R^2 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.