

PHÂN TÍCH RFM VÀ COHORT TRONG BỘ DỮ LIỆU RETAIL STORE CỦA OLIST

Objective:

Mục tiêu của tôi trong dự án này là phân tích hành vi mua sắm của khách hàng dựa trên bộ dữ liệu của Olist - tập đoàn siêu thị lớn nhất Brazil thông qua việc ứng dụng phương pháp phân tích RFM và phân tích Cohort nhằm xác định phân khúc khách hàng giá trị, và đưa ra các khuyến nghị nhằm tăng tỉ lệ giữ chân khách hàng

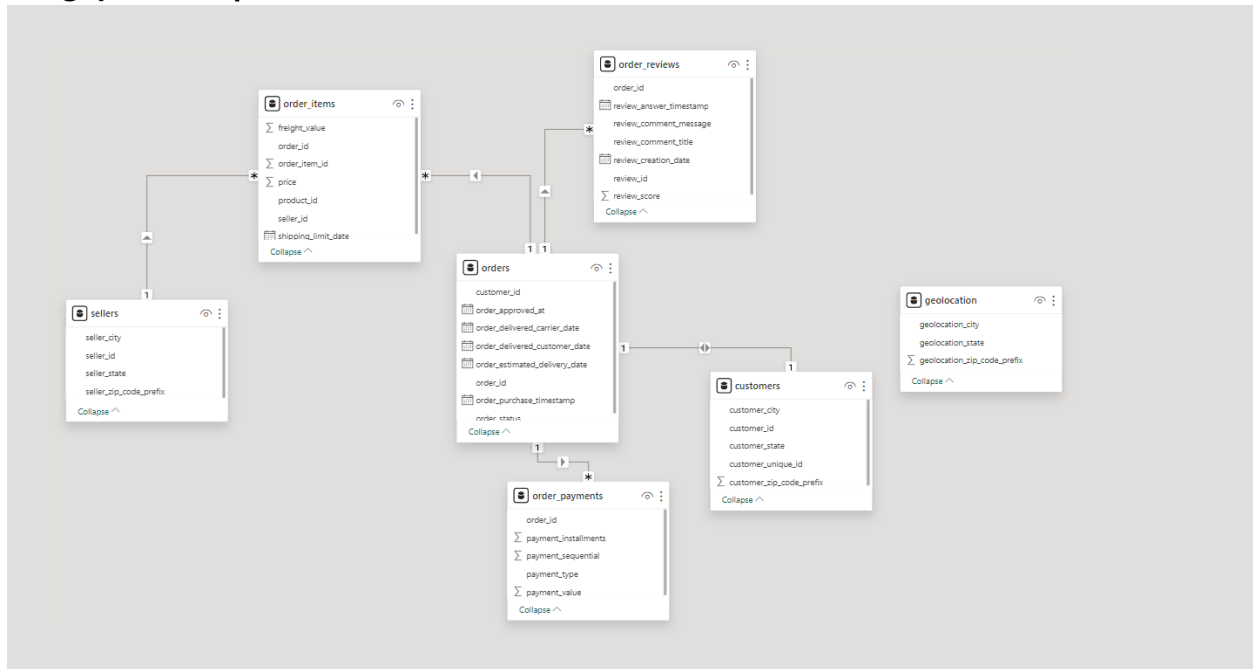
Demonstrated Skills:

- **Data Cleaning and Preparation:** Tôi sẽ dùng python để kiểm tra và làm sạch dữ liệu bao gồm xử lý các dữ liệu bị thiếu, tìm các số liệu bất thường
- **RFM Analysis:** Tôi sẽ tính điểm RFM (Recency, Frequency, Monetary) để phân loại khách hàng thông qua DAX trong PowerBi
- **Cohort Analysis:** Tôi sẽ thực hiện phân tích Cohort để tính tỉ lệ giữ chân khách hàng thông qua SQL
- **Data Visualization:** Tôi sẽ sử dụng PowerBi để trực quan hóa các số liệu qua các biểu đồ cột, biểu đồ đường,....
- **Strategic Insights and Recommendations:** Dựa trên phân tích RFM và Cohort, tôi sẽ đưa ra các insight và khuyến nghị phù hợp

Tools:

- **Google Colab:** Sử dụng Python thông qua google Colab để kiểm tra và làm sạch dữ liệu
- **Microsoft Sql Server:** Dùng để import dữ liệu và tính bảng phân tích Cohort
- **PowerBi:** Dùng để trực quan hóa dữ liệu, tính RFM

Tổng quan dữ liệu



Đây là bảng relationship biểu diễn mối quan hệ giữa các bảng trong dataset của OLIST, tổng quan dữ liệu bao gồm các dữ liệu về đơn hàng, khách hàng, sản phẩm, người bán và thanh toán

Task 1: Kiểm tra dữ liệu

Trước hết, tôi sẽ kiểm tra loại dữ liệu của các dataframe xem liệu chúng đã hợp lệ chưa

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   customer_id                          99441 non-null  object
1   customer_unique_id                   99441 non-null  object
2   customer_zip_code_prefix             99441 non-null  int64
3   customer_city                        99441 non-null  object
4   customer_state                       99441 non-null  object
dtypes: int64(1), object(4)
memory usage: 3.8+ MB
```

Bảng 1. Customers

```

↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 27912 entries, 0 to 27911
Data columns (total 3 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   geolocation_zip_code_prefix          27912 non-null  int64
1   geolocation_city                     27912 non-null  object
2   geolocation_state                    27912 non-null  object
dtypes: int64(1), object(2)
memory usage: 654.3+ KB

```

Bảng 2. Geolocation

```

↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   order_id                             112650 non-null  object
1   order_item_id                        112650 non-null  int64
2   product_id                           112650 non-null  object
3   seller_id                            112650 non-null  object
4   shipping_limit_date                  112650 non-null  datetime64[ns]
5   price                               112650 non-null  int64
6   freight_value                        112650 non-null  int64
dtypes: datetime64[ns](1), int64(3), object(3)
memory usage: 6.0+ MB

```

Bảng 3. order_items

```

↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   review_id                             99224 non-null  object
1   order_id                             99224 non-null  object
2   review_score                          99224 non-null  int64
3   review_comment_title                  11565 non-null  object
4   review_comment_message                40947 non-null  object
5   review_creation_date                  99224 non-null  datetime64[ns]
6   review_answer_timestamp                99224 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(4)
memory usage: 5.3+ MB

```

Bảng 4. order_reviews

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   product_id                            32951 non-null  object
1   product_category_name                 32341 non-null  object
2   product_name_lenght                  32341 non-null  float64
3   product_description_lenght           32341 non-null  float64
4   product_photos_qty                   32341 non-null  float64
5   product_weight_g                     32949 non-null  float64
6   product_length_cm                    32949 non-null  float64
7   product_height_cm                    32949 non-null  float64
8   product_width_cm                     32949 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.3+ MB
```

Bảng 5. products

Vậy là sau khi kiểm tra các bảng thì dtype của chúng đều hợp lệ, hầu hết các bảng đều không có dữ liệu null, tuy nhiên thì ở bảng order_reviews, mục review_comment_message và title bị trống, khả năng cao là do người dùng đánh giá mà không để lại comment, mục products cũng có sản phẩm bị thiếu

Task 2: Xử lý dữ liệu bị thiếu

Đầu tiên là tôi sẽ thay thế các giá trị null trong bảng order_reviews bằng “No comment”, thể hiện rằng đánh giá không để lại comment

```
order_reviews[['review_comment_title', 'review_comment_message']] = order_reviews[['review_comment_title', 'review_comment_message']].fillna('No comment')
order_reviews.head()
```

	review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_date	review_answer_timestamp
0	7ea98afadbe948b63e8007a5c3ba824c	723f4709e0dc0b2fc2c8d68792c2ba6	5	No comment	No comment	2018-05-08	2018-05-10 20:10:26
1	3279a7d666c135d654d1c1a395b34d20	b7d87e06832b932da4cd4e7dee2974f	5	No comment	No comment	2018-05-08	2018-05-10 16:05:27
2	0300525ef63e9c77a5a798fd312e13f1	08fc1aa23fef732befd3559729544a6a	5	No comment	No comment	2018-05-08	2018-05-10 12:59:24
3	936cc8ba40587f4c477d6c538e924012	a58e64aaf179aee88af27f932d4da7b2	5	No comment	No comment	2018-05-08	2018-05-11 01:02:40
4	396a511f928dcbb73efeeab1586eab0	4e839b1ce670c701bd83a9d9aeb58969	5	No comment	No comment	2018-05-08	2018-05-10 18:40:41

Tiếp theo tôi sẽ xóa các hàng mà cột product_category_name bị thiếu trong bảng products vì các tên sản phẩm đã được encode nên ta không thể phân tích bằng tên sản phẩm mà thiếu product_category_name được

```
products = products.dropna(subset=['product_category_name'])
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32341 entries, 0 to 32340
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   product_id                            32341 non-null  object
1   product_category_name                 32341 non-null  object
2   product_name_lenght                  32341 non-null  float64
3   product_description_lenght           32341 non-null  float64
4   product_photos_qty                   32341 non-null  float64
5   product_weight_g                     32340 non-null  float64
6   product_length_cm                    32340 non-null  float64
7   product_height_cm                    32340 non-null  float64
8   product_width_cm                     32340 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.5+ MB
```

Sau khi xóa các dòng chứa dữ liệu null ở cột product_category_names thì còn sót lại duy nhất 1 giá trị null ở cột product_weight_g và product_length_cm

```

null_values_df = products[products[(['product_weight_g', 'product_length_cm']).isnull().any(axis=1)]
null_values_df

```

	product_id	product_category_name	product_name_lenght	product_description_lenght	product_photos_qty	product_weight_g	product_length_cm	product_height_cm
24798	09ff539a621711667c43eba6a3bd846d	bebes	60.0	865.0	3.0	NaN	NaN	NaN

Chỉ còn lại một quan sát duy nhất chứa dữ liệu null và nó nằm ở hàng bebes, ta sẽ lấy trung bình siêu thị bebes và replace vào cột bị thiếu

```

# Chỉ chọn các cột số
numeric_columns = ['product_weight_g', 'product_length_cm', 'product_height_cm', 'product_width_cm']

# Tính giá trị trung bình cho các cột số trong category "bebes"
mean_values = products[products['product_category_name'] == 'bebes'][numeric_columns].mean()

# Áp dụng giá trị trung bình cho các giá trị null
for column in numeric_columns:
    products.loc[(products['product_category_name'] == 'bebes') & (products[column].isnull()), column] = mean_values[column]
products.info()

```

```

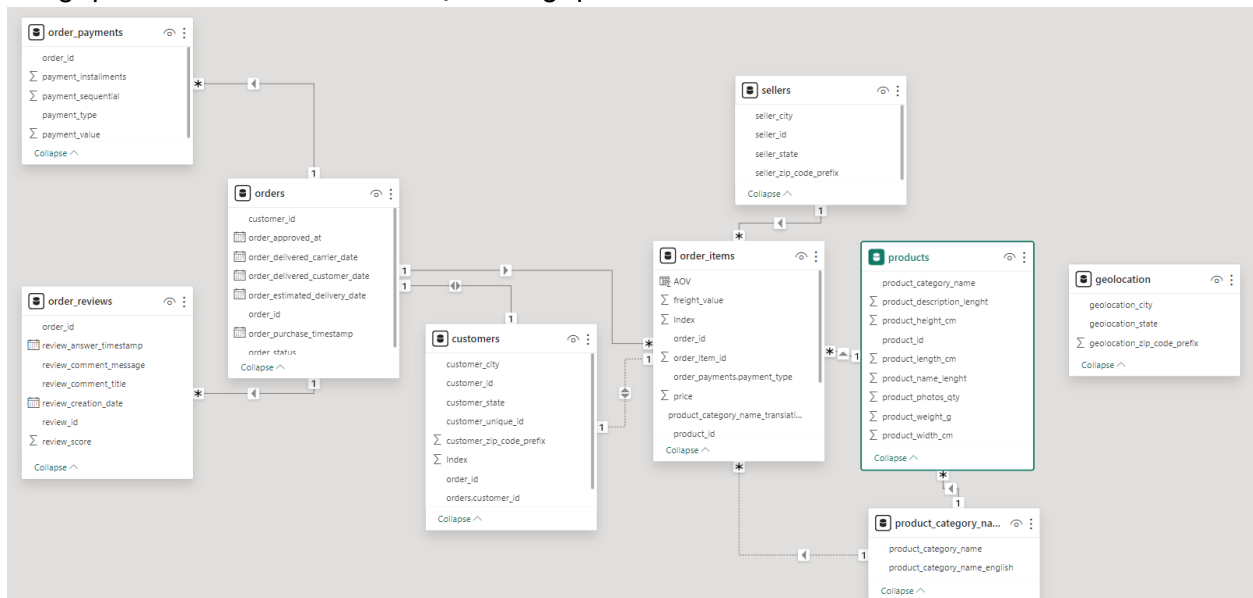
<class 'pandas.core.frame.DataFrame'>
Index: 32341 entries, 0 to 32340
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   product_id                            32341 non-null  object
1   product_category_name                 32341 non-null  object
2   product_name_lenght                  32341 non-null  float64
3   product_description_lenght           32341 non-null  float64
4   product_photos_qty                   32341 non-null  float64
5   product_weight_g                     32341 non-null  float64
6   product_length_cm                    32341 non-null  float64
7   product_height_cm                    32341 non-null  float64
8   product_width_cm                     32341 non-null  float64
dtypes: float64(7), object(2)
memory usage: 3.5+ MB

```

Sau khi thay thế giá trị null thì giờ ta đã có một dataset hoàn chỉnh, tiếp theo tôi sẽ thực hiện join các bảng lại với nhau trong PowerBi. Vì các bảng đều có cột chung nên việc join các bảng sẽ khá đơn giản và PowerBi có thể làm việc đó nhanh và gọn, đồng thời chúng cũng sẽ thuận tiện để giúp tôi trực quan hóa dữ liệu.

Task 3: Join các bảng lại với nhau bằng PowerBi

Sau khi join các bảng lại với nhau ta có relationship table giữa các bảng dưới đây, việc join các bảng lại với nhau sẽ giúp chúng ta dễ tham chiếu thông tin hơn, đồng thời sẽ thuận tiện trong quá trình visualize các dữ liệu tương quan với nhau



Task 4: Tính RFM bằng PowerBi

Đầu tiên ta sẽ tạo bảng RFM dựa trên các dữ liệu đã có trong bảng. Cột R- Recency sẽ tương ứng với lần mua gần đây nhất, ta dùng hàm datediff để tính số ngày từ lần cuối mua đến

hiện tại. Cột F-Frequency sẽ là tần suất giao dịch của khách hàng, ta sẽ đếm số lần mua hàng của khách bằng hàm DISTINCTCOUNT. Cột M sẽ là tổng số tiền khách hàng đã chi, ta sẽ dùng hàm sum để tính chúng

```
1 RFM = SUMMARIZE(customers, customers[customer_unique_id],
2 "R",
3 DATEDIFF(max(orders[order_purchase_timestamp]), TODAY(), DAY),
4 "F",
5 DISTINCTCOUNT(orders[order_id]),
6 "M",
7 SUM((order_payments[payment_value])))
```

Tiếp theo, chúng ta sẽ tạo cột R_Score để gán điểm cho khách hàng dựa trên giá trị Recency. Đầu tiên, chúng ta tính các phần trăm vị trí (80%, 60%, 40%, 20%) của giá trị Recency trong tập dữ liệu. Dựa vào các ngưỡng này, mỗi khách hàng sẽ được gán một điểm từ 1 đến 5: điểm 1 cho khách hàng ít mua gần đây nhất (cao hơn hoặc bằng 80%) và điểm 5 cho khách hàng mua gần đây nhất (dưới 20%)

```
1 R_Score =
2 var percent_80 = PERCENTILE.EXC(RFM[R], 0.8)
3 var percent_60 = PERCENTILE.EXC(RFM[R], 0.6)
4 var percent_40 = PERCENTILE.EXC(RFM[R], 0.4)
5 var percent_20 = PERCENTILE.EXC(RFM[R], 0.2)
6 return
7 SWITCH(TRUE(),
8     RFM[R] >= percent_80 , 1,
9     RFM[R] >= percent_60 , 2,
10    RFM[R] >= percent_40 , 3,
11    RFM[R] >= percent_20 , 4,
12    5)
```

Tiếp tục với cột M_Score chúng ta sẽ gán điểm cho khách hàng dựa trên giá trị Monetary. Đầu tiên, chúng ta tính các phần trăm vị trí 80%, 60%, 40%, và 20% của giá trị Monetary trong tập dữ liệu. Dựa vào các ngưỡng này, mỗi khách hàng sẽ được gán một điểm từ 1 đến 5: điểm 5 cho khách hàng có giá trị mua hàng cao nhất (trên 80%) và điểm 1 cho khách hàng có giá trị mua hàng thấp nhất (dưới 20%)

```

1 M_Score =
2 var percent_80 = PERCENTILE.EXC(RFM[M], 0.8)
3 var percent_60 = PERCENTILE.EXC(RFM[M], 0.6)
4 var percent_40 = PERCENTILE.EXC(RFM[M], 0.4)
5 var percent_20 = PERCENTILE.EXC(RFM[M], 0.2)
6 return
7 SWITCH(TRUE(),
8     RFM[M] >= percent_80 , 5,
9     RFM[M] >= percent_60 , 4,
10    RFM[M] >= percent_40 , 3,
11    RFM[M] >= percent_20 , 2,
12    1)

```

Tương tự với cột F_Score, ta cũng sẽ đặt ra các ngưỡng, mỗi khách hàng sẽ được gán một điểm từ 1 đến 5: điểm 5 cho khách hàng có tần suất mua hàng cao nhất (trên 80%) và điểm 1 cho khách hàng có tần suất mua hàng thấp nhất (dưới 20%)

```

1 F_Score =
2 var percent_80 = PERCENTILE.EXC(RFM[F], 0.8)
3 var percent_60 = PERCENTILE.EXC(RFM[F], 0.6)
4 var percent_40 = PERCENTILE.EXC(RFM[F], 0.4)
5 var percent_20 = PERCENTILE.EXC(RFM[F], 0.2)
6
7 return
8 SWITCH(TRUE(),
9     RFM[F] >= percent_80 , 5,
10    RFM[F] >= percent_60 , 4,
11    RFM[F] >= percent_40 , 3,
12    RFM[F] >= percent_20 , 2,
13    1)

```

Task 5: Tạo bảng Cohort

Ta sẽ tiến hành tạo bảng Cohort bằng SQL để tính lượng khách hàng ở lại và rời đi. Ở đây tôi sẽ tính lượng khách hàng rời đi và ở lại hàng tháng theo từng năm và tôi sẽ bỏ qua năm 2016 vì dữ liệu của năm 2016 chỉ có 2 tháng, các bước cụ thể như sau:

Đầu tiên tôi sẽ tạo 1 CTE chứa thông tin khách hàng và ngày đầu tiên mua hàng của họ trong năm bằng cách chọn ngày mua hàng nhỏ nhất sau khi đã join dữ liệu khách hàng với dữ liệu order.

```

with raw1 as (
    select customer_unique_id, min(order_purchase_timestamp) as first_order_date
    from dbo.orders o
    join dbo.customers c on o.customer_id = c.customer_id
    where year(order_purchase_timestamp) = 2017
    group by customer_unique_id
),

```

Tiếp theo tôi sẽ tạo thêm 1 CTE chứa ID khách hàng và tất cả các ngày mua hàng của họ trong năm.

```

),
raw2 as (
    SELECT c.customer_unique_id, o.order_purchase_timestamp
    FROM dbo.orders o
    JOIN dbo.customers c ON o.customer_id = c.customer_id
    where year(order_purchase_timestamp) = 2017
)

```

Bước tiếp theo chúng ta sẽ join 2 bảng vừa tạo lại với nhau để tìm ra khoảng cách của ngày mua hàng đầu tiên với những ngày mua hàng tiếp theo bằng hàm datediff

```

final_table as (
    select
        a.customer_unique_id,
        MONTH(a.first_order_date) as cohort_month,
        MONTH(b.order_purchase_timestamp) as order_month,
        DATEDIFF(MONTH, a.first_order_date, b.order_purchase_timestamp) as month_diff
    from raw1 as a
    left join raw2 as b on a.customer_unique_id = b.customer_unique_id
)

```

Cuối cùng tôi sẽ tổng hợp lại số khách hàng trong mỗi cohort và sẽ dùng hàm count distinct để trả về số lượng còn lại sau mỗi tháng sau đó sẽ nhóm và sắp xếp lại theo cohort_month để ta có thể phân tích xu hướng của khách hàng thuận tiện hơn

```

select
    cohort_month,
    count(distinct customer_unique_id) as number_of_user,
    count(distinct case when month_diff = 0 then customer_unique_id end) as month_0,
    count(distinct case when month_diff = 1 then customer_unique_id end) as month_1,
    count(distinct case when month_diff = 2 then customer_unique_id end) as month_2,
    count(distinct case when month_diff = 3 then customer_unique_id end) as month_3,
    count(distinct case when month_diff = 4 then customer_unique_id end) as month_4,
    count(distinct case when month_diff = 5 then customer_unique_id end) as month_5,
    count(distinct case when month_diff = 6 then customer_unique_id end) as month_6,
    count(distinct case when month_diff = 7 then customer_unique_id end) as month_7,
    count(distinct case when month_diff = 8 then customer_unique_id end) as month_8,
    count(distinct case when month_diff = 9 then customer_unique_id end) as month_9,
    count(distinct case when month_diff = 10 then customer_unique_id end) as month_10,
    count(distinct case when month_diff = 11 then customer_unique_id end) as month_11
from final_table
group by cohort_month
order by cohort_month

```

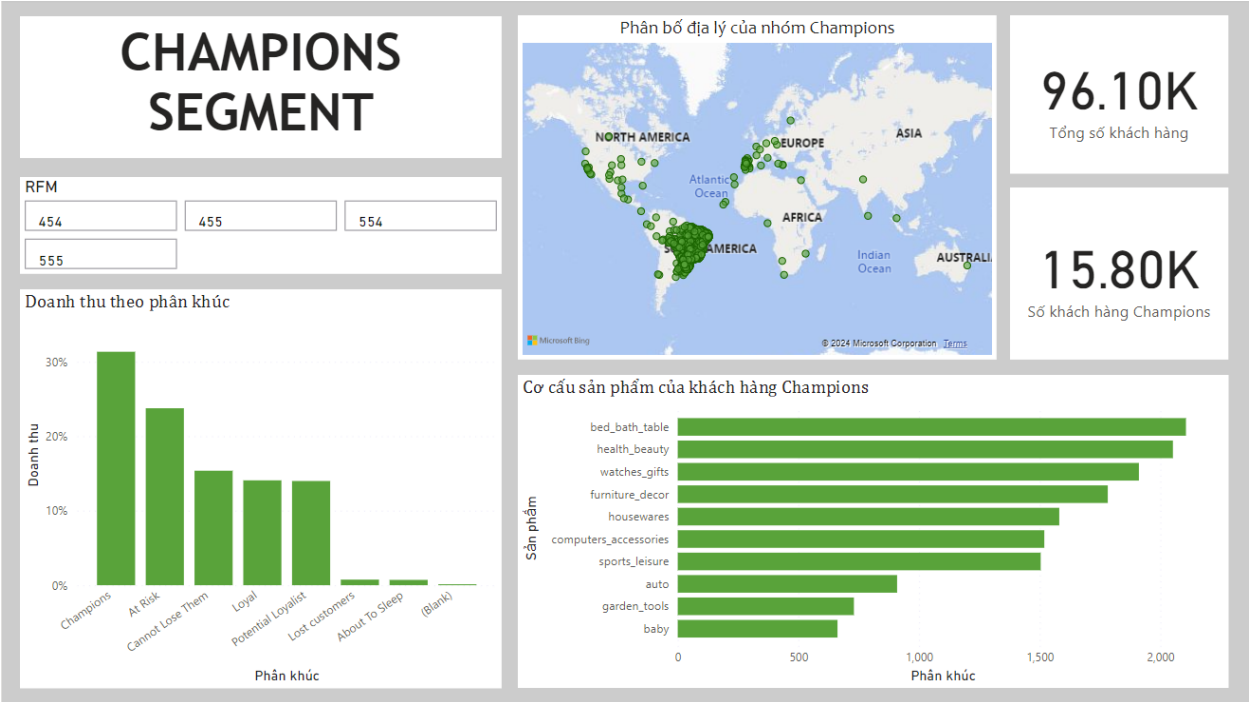
Ta sẽ thực hiện tương tự để tính số lượng khách hàng còn lại trong năm 2018. Sau khi có được bảng cohort, tôi sẽ sao chép chúng vào file excel để tổng hợp, chi tiết các bạn có thể vào github của tôi để xem.

Visualisation

Ở phần trực quan hóa dữ liệu, tôi sẽ sử dụng PowerBi để tạo Dashboard. Ở đây tôi sẽ tập trung phân tích vào phân khúc khách hàng dựa trên RFM

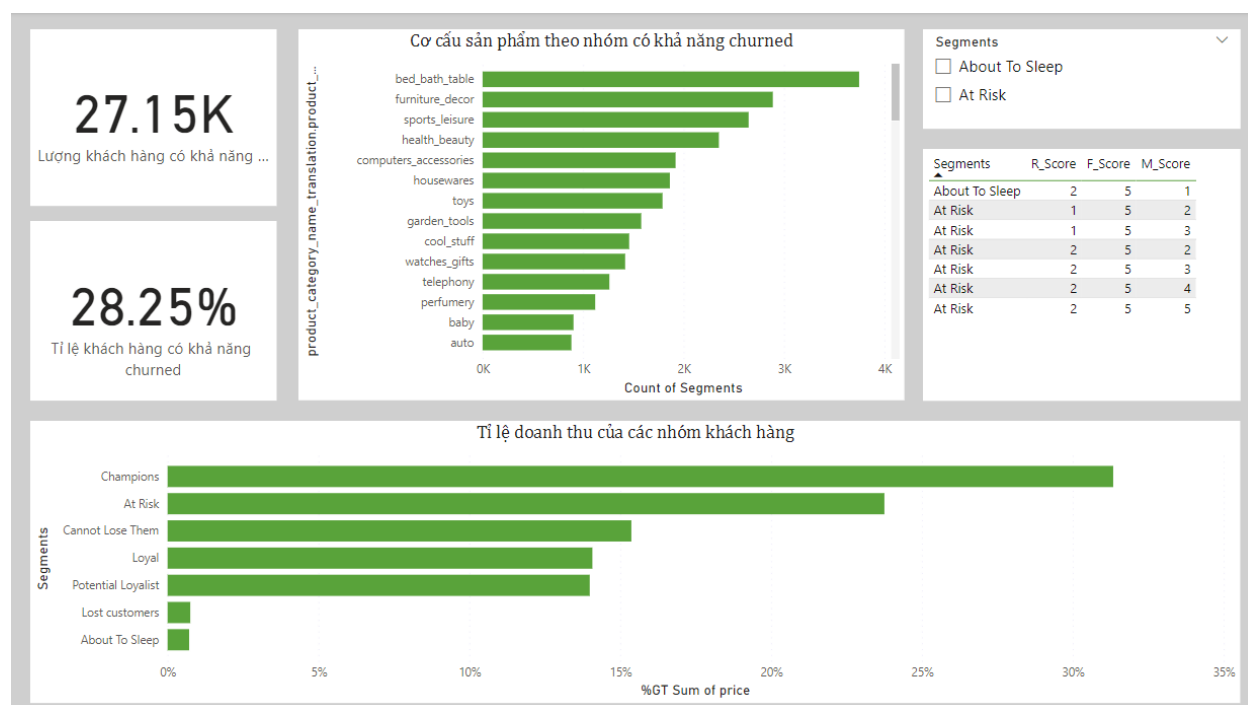
Dưới đây là bảng tổng quan phân tích phân khúc Champions - Phân khúc được coi là giá trị nhất của doanh nghiệp:

- Doanh thu của tệp khách hàng này chiếm tới 30% tổng doanh thu của siêu thị, khẳng định được giá trị của tệp khách này.
- So với quy mô khách hàng, tệp khách Champions có số lượng là 15 nghìn so với 96 nghìn tổng số khách hàng, chiếm hơn 15% tổng số khách. Đây là một con số khá là tích cực, cho thấy doanh nghiệp có một lượng khách trung thành ổn định
- Khách hàng sẽ tập trung chủ yếu ở Nam Mỹ và rải rác ở các vùng lân cận như Bắc Mỹ và một số quốc gia Châu Âu
- Danh mục sản phẩm được mua nhiều nhất của nhóm khách hàng này là đồ nội thất gia dụng (bed_bath_table, furniture_decor, housewares), tiếp đến là các danh mục đồ liên quan đến sức khỏe và làm đẹp, cuối cùng là đồng hồ. Sự kết hợp giữa nội thất, làm đẹp và trang sức gợi ý rằng phân khúc này tập trung vào lối sống chất lượng cao

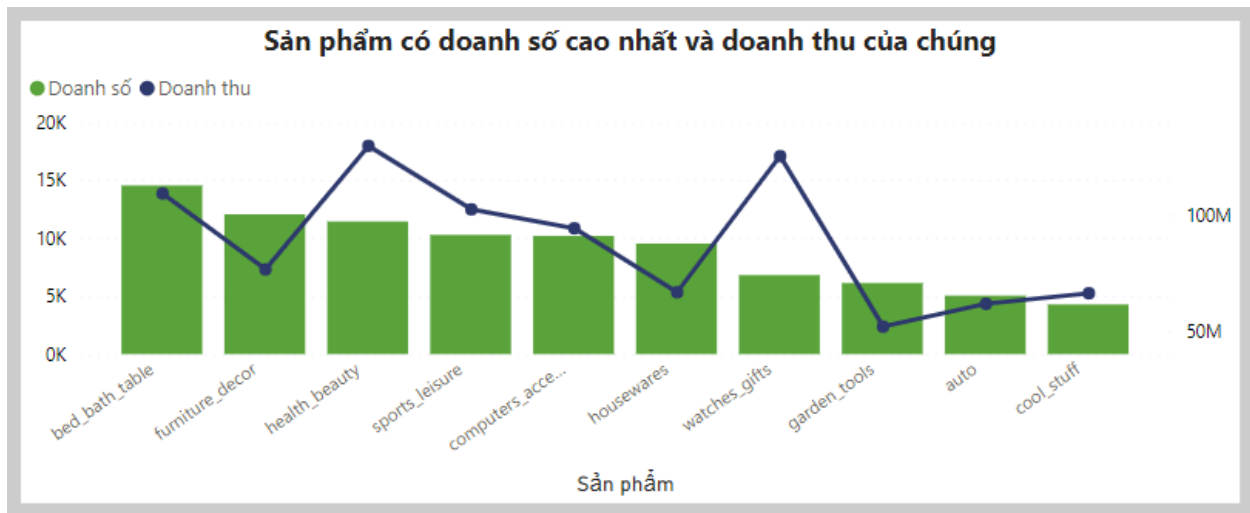


Tiếp theo ta sẽ phân tích nhóm khách hàng có khả năng rời bỏ cao nhất, đó là nhóm At Risk và About to sleep, đây là 2 nhóm có lần mua gần nhất xa nhất. Cụ thể ở dashboard dưới đây:

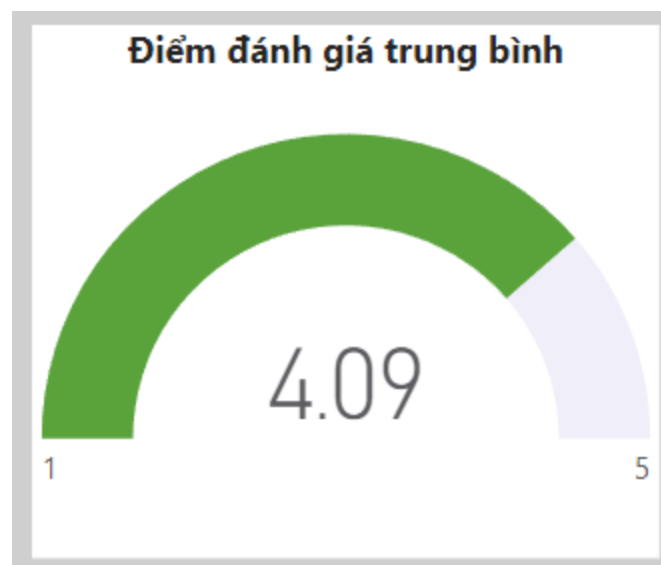
- Tổng số lượng của các tệp này là hơn 27 nghìn khách hàng, chiếm 28.25% tổng số khách hàng. Đây là một tỉ lệ khá đáng kể, cho thấy doanh nghiệp cần có chiến lược giữ chân khách hàng hiệu quả
- Cơ cấu sản phẩm của các nhóm khách hàng này chủ yếu là đồ gia dụng và đồ thể thao, trong đó cơ cấu của bed_bath_table chiếm đáng kể trong số này. Một trong những sản phẩm được mua ít nhất của nhóm này là baby, auto, perfumery; Những mặt hàng này cũng ko nằm trong top sản phẩm được mua nhiều nhất của tệp Champions, siêu thị nên phân phối lại danh mục hàng hóa nhằm tăng doanh thu và tỉ lệ giữ chân khách hàng
- Tuy là nhóm khách hàng có tỉ lệ rời bỏ cao, tệp khách At Risk lại có chiếm 20% tỉ lệ doanh thu của siêu thị, chỉ xếp sau tệp Champions, điều này chứng tỏ là lượng khách vãng lai của siêu thị là rất nhiều và họ cũng là một nguồn doanh thu ổn định tiềm năng. siêu thị nên đưa ra các chiến lược để giữ chân tệp khách này lại



Tiếp theo tôi sẽ đánh giá doanh thu của các sản phẩm có doanh số cao nhất. Đứng đầu trong doanh số cao nhất là bed_bath_table và furniture_decor, doanh thu của bed_bath_table khá cao nhưng của furniture_decor lại có phần hơi thấp, điều này chứng tỏ AOV (Average order value) của dòng sản phẩm này thấp. Ngược lại, sản phẩm health_beauty có doanh số xếp thứ 3 nhưng doanh thu lại nắm vị trí có doanh thu cao gần nhất sau watches_gift, một sản phẩm cũng có doanh số thấp nhưng doanh thu cao ngất ngưỡng, điều này chứng tỏ AOV của health_beauty và watches_gift là rất cao. Dòng sản phẩm Garden_tools, auto, cool_stuff lại vừa có doanh số thấp vừa doanh thu thấp, siêu thị nên cân nhắc phân phối lại sản phẩm



Để đánh giá khách quan thì tôi sẽ thêm một biểu đồ nhỏ là điểm review của khách hàng



Ta có thể thấy điểm đánh giá nằm ở mức khá cao là 4.09/5, ở ngưỡng trên trung bình. Điều này phản ánh chất lượng sản phẩm và dịch vụ là rất tốt, đồng thời giúp nhà hàng xây

cohort_month	number_of_user	month_0	month_1	month_2	month_3	month_4	month_5	month_6	month_7	month_8	month_9	month_10	month_11	
1	765	765	3	2	1	3	1	4	1	1	0	3	1	
2	1752	1752	4	5	2	7	2	4	3	3	4	2		
3	2636	2636	13	10	10	9	4	4	8	9	2			
4	2353	2353	14	5	4	8	6	8	7	7				
5	3596	3596	18	18	14	11	12	15	6					
6	3139	3139	15	11	13	8	12	12						
7	3895	3895	20	14	10	11	8							
8	4184	4184	29	14	11	15								
9	4131	4131	28	22	12									
10	4470	4470	31	11										
11	7305	7305	40											
12	5487	5487												
Total		43713	43713	215	112	77	72	45	47	25	20	6	5	1
cohort_month	number_of_user	month_0	month_1	month_2	month_3	month_4	month_5	month_6	month_7	month_8	month_9	month_10	month_11	
1	765	0	762	1	1	-2	2	-3	3	0	1	-3	2	
2	1752	0	1748	-1	3	-5	5	-2	1	0	-1	2		
3	2636	0	2623	3	0	1	5	0	-4	-1	7			
4	2353	0	2339	9	1	-4	2	-2	1	0				
5	3596	0	3578	0	4	3	-1	-3	9					
6	3139	0	3124	4	-2	5	-4	0						
7	3895	0	3875	14	10	11	8							
8	4184	0	4155	15	3	-4								
9	4131	0	4103	6	10									
10	4470	0	4439	20										
11	7305	0	7265											
12	5487	0												
Total		43713	0	38011	71	30	5	17	-10	10	-1	7	-1	2
cohort_month	number_of_user	month_0	month_1	month_2	month_3	month_4	month_5	month_6	month_7	month_8	month_9	month_10	month_11	
1	765	100%	0%	67%	50%	300%	33%	400%	25%	100%	0%	0%	33%	
2	1752	100%	0%	125%	40%	350%	29%	200%	75%	100%	133%	50%		
3	2636	100%	0%	77%	100%	90%	44%	100%	200%	113%	22%			
4	2353	100%	1%	36%	80%	200%	75%	133%	88%	100%				
5	3596	100%	1%	100%	78%	79%	109%	125%	40%					
6	3139	100%	0%	73%	118%	62%	150%	100%						
7	3895	100%	1%	70%	71%	110%	73%							
8	4184	100%	1%	48%	79%	136%								
9	4131	100%	1%	79%	55%									
10	4470	100%	1%	35%										
11	7305	100%	1%											
12	5487	100%												
cohort_month	number_of_user	month_0	month_1	month_2	month_3	month_4	month_5	month_6	month_7	month_8				

Đề xuất giải pháp cho siêu thị

Phân phối lại dòng sản phẩm:

- Cân nhắc bỏ bớt các sản phẩm mang lại ít doanh thu cũng như doanh số của siêu thị như dòng sản phẩm baby, auto, cool_stuff
- Tập trung vào những dòng sản phẩm tiềm năng như health_beauty, computer_accessory, watches_gif

Mở rộng dịch vụ và sản phẩm bổ sung:

- Có thể nhập thêm sản phẩm liên quan đến sửa chữa nội thất, hoặc nhập thêm một số loại nội thất theo mùa nhằm giữ chân khách
- Kết hợp dịch vụ bảo trì và nâng cấp nội thất
- Kết hợp dịch vụ tư vấn thiết kế nội thất

Tạo chương trình khuyến mãi với chu kỳ dài:

- Tạo chương trình khách hàng thân thiết với phần thưởng dài hạn. Chẳng hạn, tích điểm không chỉ dựa trên số lần hay giá trị mua hàng mà còn dựa trên tương tác khác như đánh giá sản phẩm, chia sẻ trên mạng xã hội, tham gia sự kiện
- Tạo ưu đãi dài hạn cho khách hàng trung thành như là giảm giá hoặc quà tặng sau mỗi chu kỳ vài năm hoặc khi họ có nhu cầu nâng cấp, bảo trì nội thất

Một số yếu tố khác có thể cân nhắc:

- Cung và cầu của thị trường: Cân nhắc thị hiếu và nhu cầu của người dùng, xem xét liệu phong cách nội thất còn phù hợp không
- Đối thủ cạnh tranh: Nghiên cứu các sản phẩm và chiến lược của đối thủ cạnh tranh
- Yếu tố vận chuyển: Vì đồ nội thất khá cồng kềnh, siêu thị có thể xem xét các yếu tố liên quan đến vận chuyển như là chi phí vận chuyển, thời gian giao hàng, độ thỏa dụng của khách hàng,.....