با سلام

با توجه به اطلاعات موجود در فایل اکسل برای تحلیل و بررسی دقیق رفتار مشتری ها تصمیم گرفتم مدل RFM را برای این فایل پیاده سازی کنم.

چرا مدل RFM؟

مدل RFM که مخفف Recency، Frequency و Monetary است، یک روش تحلیلی برای ارزیابی و دسته بندی مشتریان بر اساس زمان آخرین خرید، تعداد دفعات خرید و کل مبلغ خرید های انجام شده است. اهداف مارکتینگی این مدل شامل بهینه سازی کمپین های بازاریابی، افزایش وفاداری مشتریان، شناسایی و حفظ مشتریان با ارزش و ارائه پیشنهادات و تخفیف های هدفمند است. این اطلاعات به کسب کارها امکان می دهد تا استراتژی های بازاریابی خود را متناسب با نیازها و رفتارهای مشتریان تنظیم کنند.

(در نظر داشته باشید که در این فایل سعی شده است که فقط نتایج بدست آماده توضیح داده شود و خود کد با استفاده از کامنت کامل توضیح داده شده است.)

در ابتدا ستونی با نام final_price با استفاده از اطلاعات موجود محاسبه شد. این ستون میزان کل مبلغی را که مشتری در یک row پرداخت کرده بود به صورت مجزا محاسبه میکرد که برای مدل RFM مورد نیاز بود.

منطق محاسبه این بود که تعداد آیتم ها در هزینه ضرب شود و در نهایت از مجموع تخفیف ها کسر شود. در این منطق هزینه ارسال دیده نشده است تا هزینه های عملیاتی وارد اطلاعات نشود و اطلاعات برای اهداف مارکتینگی به راحتی مورد استفاده قرار گیرد.

```
In [14]: # final_price columns items * their price - total discount
df ["final_price"] = (df["items"] * df["price"] ) - (df["discount"] + df["voucher_discount"] )
```

	order_number	created_at	user_id	main_category	total_shipping_fee	final_shipping_fee	items	price	discount	voucher_discount	city	final_price
0	100000	2023-12-31 22:53:00	1000	A	29900	0	1	900000	558000	30000	A1	312000
1	100001	2023-12-16 19:25:00	1001	A	39000	0	1	1680000	0	84000	A2	1596000
2	100002	2023-12-03 01:06:00	1002	B	29900	0	1	2950000	0	0	A3	2950000
3	100003	2023-10-17 13:09:00	1003	B	28900	0	1	16900000	1352000	777400	A4	14770600
4	100004	2023-11-30 04:45:00	1004	B	39000	39000	1	330000	0	0	A2	330000

بعد از تمیز کردن اطلاعات در ستون تاریخ، باید یک مقدار و ملاک برای مقایسه ی تاریخ برآورد می شد، این مقدار و ملاک به این جهت اهمیت داشت که بتوانیم میزان Recency را به درستی محاسبه کنیم. در یک دیتای آپدیت باید today و یا تاریخ همان روز را ببینیم اما در این دیتا با توجه به قدیمی بودن اطلاعات یک روز پس از MAX تاریخ دیتای موجود در نظر گرفته شد.

```
In [13]: # Analysis as of: 2024-02-01 (max order date in the dataset: 2024-01-31)
today = datetime.strptime('2024-02-01', '%Y-%m-%d')
```

در این زمان score های ما با توجه به دیتا قابل محاسبه بودند.

پس از محاسبه ی score های مورد نیاز مدل RFM بر اساس دیتاهای موجود جدول نهایی مقدار و تعداد عددی هر کدام از موارد R / F / M به شرح زیر بدست آمد:

Name: count, dtype: int64

In [48]:
```
df_rfm
```

Out[48]:

	user_id	frequency	monetary	recency	r_score	f_score	m_score	rfm_sum
0	1000	3	8364699	31	4	2	4	10
1	1001	1	1596000	46	4	1	3	8
2	1002	1	2950000	59	3	1	4	8
3	1003	1	14770600	106	1	1	5	7
4	1004	3	2810850	62	3	2	3	8
...	...	...	...	...	...	...	...	...
375869	376869	1	139000	68	3	1	1	5
375870	376870	1	1950000	47	4	1	3	8
375871	376871	1	414000	22	5	1	2	8
375872	376872	1	4990000	54	4	1	4	9
375873	376873	1	270750	72	2	1	1	4

375874 rows × 8 columns

In [116]:
```
df_rfm["r_score"].value_counts()
```

Out[116]:
```
r_score
3    81887
5    77301
1    74289
4    73720
2    68677
Name: count, dtype: int64
```

In [46]:
```
df_rfm["f_score"].value_counts()
```

Out[46]:
```
f_score
1    280142
2     37494
4     37464
3     20774
Name: count, dtype: int64
```

In [47]:
```
df_rfm["m_score"].value_counts()
```

Out[47]:
```
m_score
3    75176
4    75175
1    75175
2    75175
5    75173
Name: count, dtype: int64
```

با استفاده از ستون rfm-sum که مجموع 3 ستون تعریف شده بود می توانستیم اطلاعات را سگمنت کنیم، (دلیل این تعریف این بود که در صورتی که می خواستیم اطلاعات را بر اساس خود سه متغیر سگمنت کنیم تعداد مدل ها خیلی زیاد می شود و درک و تحلیل کردن رفتار مشتری برای ما بسیار دشوار بود. در این قسمت متوجه شدم که f-score یا همان Frequency ما فقط به 4 قسمت تقسیم می شود. با توجه به نزدیک بودن تاریخ های سفارش برخی از مشتری ها (این اتفاق ممکن است یک زمان مشخص مثل کمپین رخ دهد) پس دسته بندی مشتری را در این قسمت تا ماکزیموم 4 در نظر میگیریم.

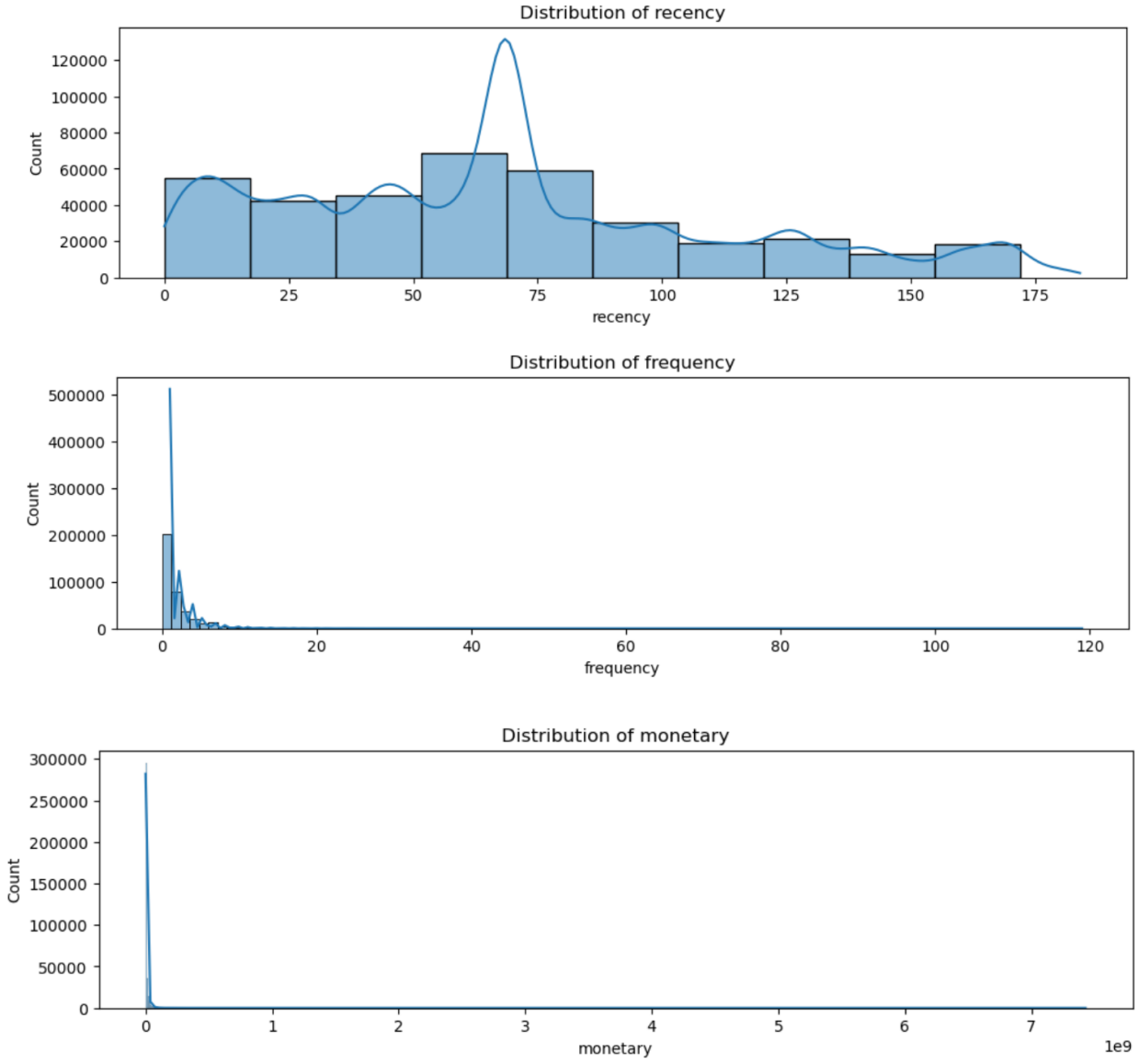
پس در نتیجه توانستیم به دسته بندی و تقسیم بندی های زیر برسیم.

```
# Adjusting the RFM segment labels based on a 4-point scale for Frequency
# Champions: Best customers who bought most recently, most often, and are heavy spenders
df_rfm = assign_label(df_rfm, (5,5), (4,4), 'champions')

# Loyal Customers: Customers who buy on a regular basis. Responsive to promotions.
df_rfm = assign_label(df_rfm, (3,4), (4,4), 'loyal customers')

# Potential Loyalist: Recent customers with average frequency.
df_rfm = assign_label(df_rfm, (4,5), (2,3), 'potential loyalist')

# New Customers: Customers who have a high overall RFM score but are not frequent shoppers.
df_rfm = assign_label(df_rfm, (5,5), (1,1), 'new customers')

# Promising: Recent shoppers, but spent a small amount.
df_rfm = assign_label(df_rfm, (4,4), (1,1), 'promising')

# Needing Attention: Above average recency, frequency, and monetary values. May not have shopped recently.
df_rfm = assign_label(df_rfm, (2,3), (2,3), 'needing attention')

# About to Sleep: Below average recency, frequency, and monetary values. Will lose them if not reactivated.
df_rfm = assign_label(df_rfm, (2,3), (1,2), 'about to sleep')

# At Risk: Shopped long ago, bought few, and spent little.
df_rfm = assign_label(df_rfm, (1,2), (2,3), 'at risk')

# Can't Lose Them: Made big purchases, and often, but haven’t returned for a long time.
df_rfm = assign_label(df_rfm, (1,2), (4,4), 'cant lose them')

# lost: Last purchase was long back, low spenders, and low number of orders.
df_rfm = assign_label(df_rfm, (1,2), (1,2), 'lost')
```

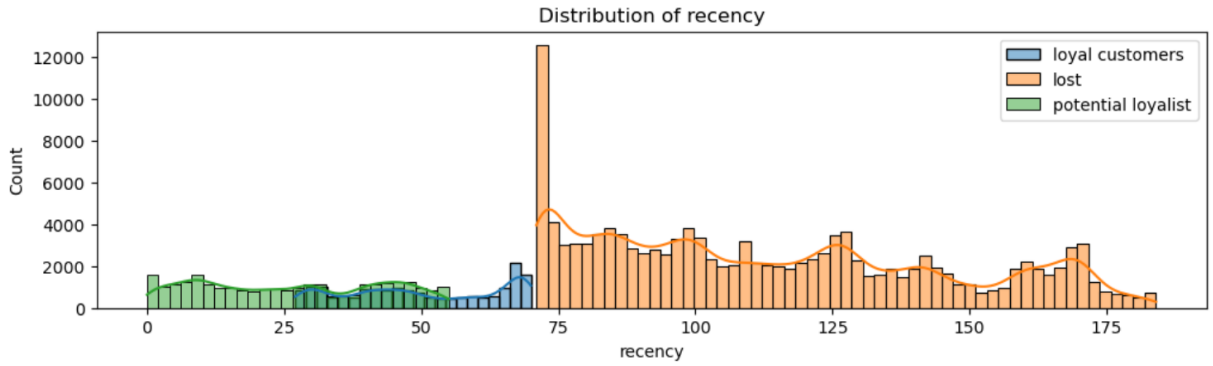
تعریف هر کدام از دسته بندی صورت گرفته در قسمت دیسکریپشن کد وارد شده است اما برای اینکه منطق موارد را توضیح دهیم می توانیم یکی از موارد را با هم بررسی کنیم.

برای مثال : new customers مشتریانی هستند که نسبت به تاریخ فعلی(ماکزیموم date در دیتا ست) در نزدیکترین زمان ممکن ثبت سفارش داشتند اما در طول بازه ی زمانی دیتا تعداد سفارش های آن ها بسیار نا چیز یا حتی صفر بوده است.

در ابتدا برای بررسی نوع توزیع 3 عامل بدست آمده، یعنی recency, frequency, monetary آن ها را به صورت ویژوال نمایش دادیم.

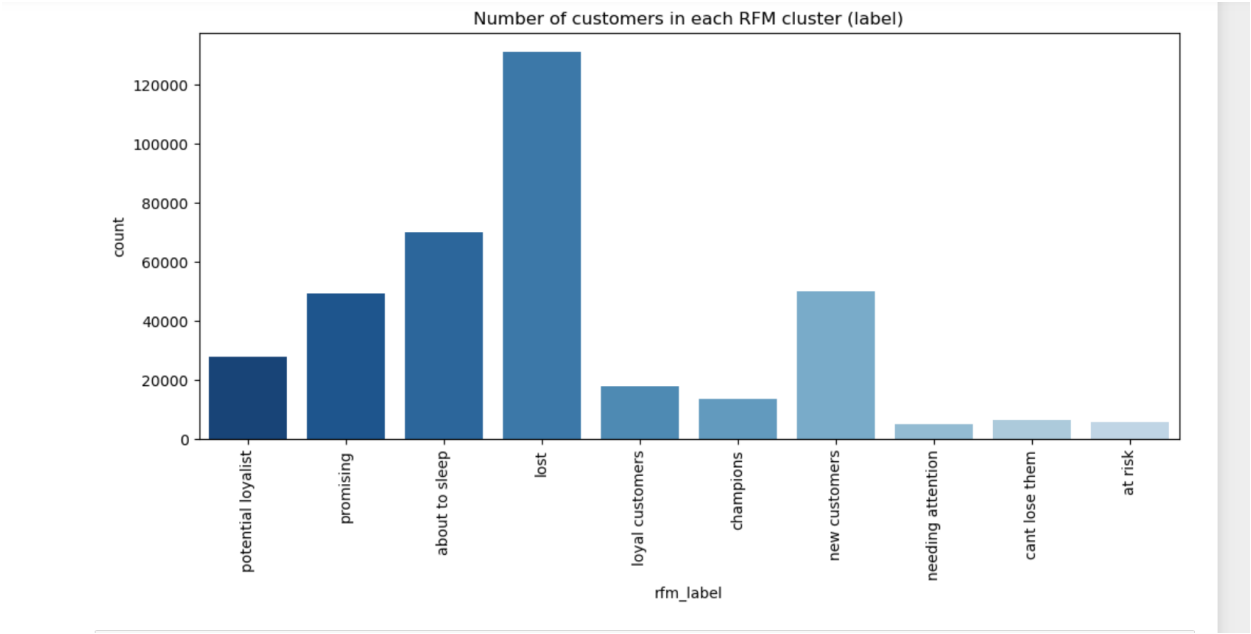


با توجه به توزیعی که قابل مشاهده است متوجه می شویم که یکی از دسته بندی های تعریف شده ای که داریم احتمالا از مابقی دسته بندی ها بزرگ تر خواهد بود و با توجه به سوال های مطرح شده در مورد رفتار مشتریان وفادار و مشتریانی که churn شده اند دیتای زیر بررسی شد.



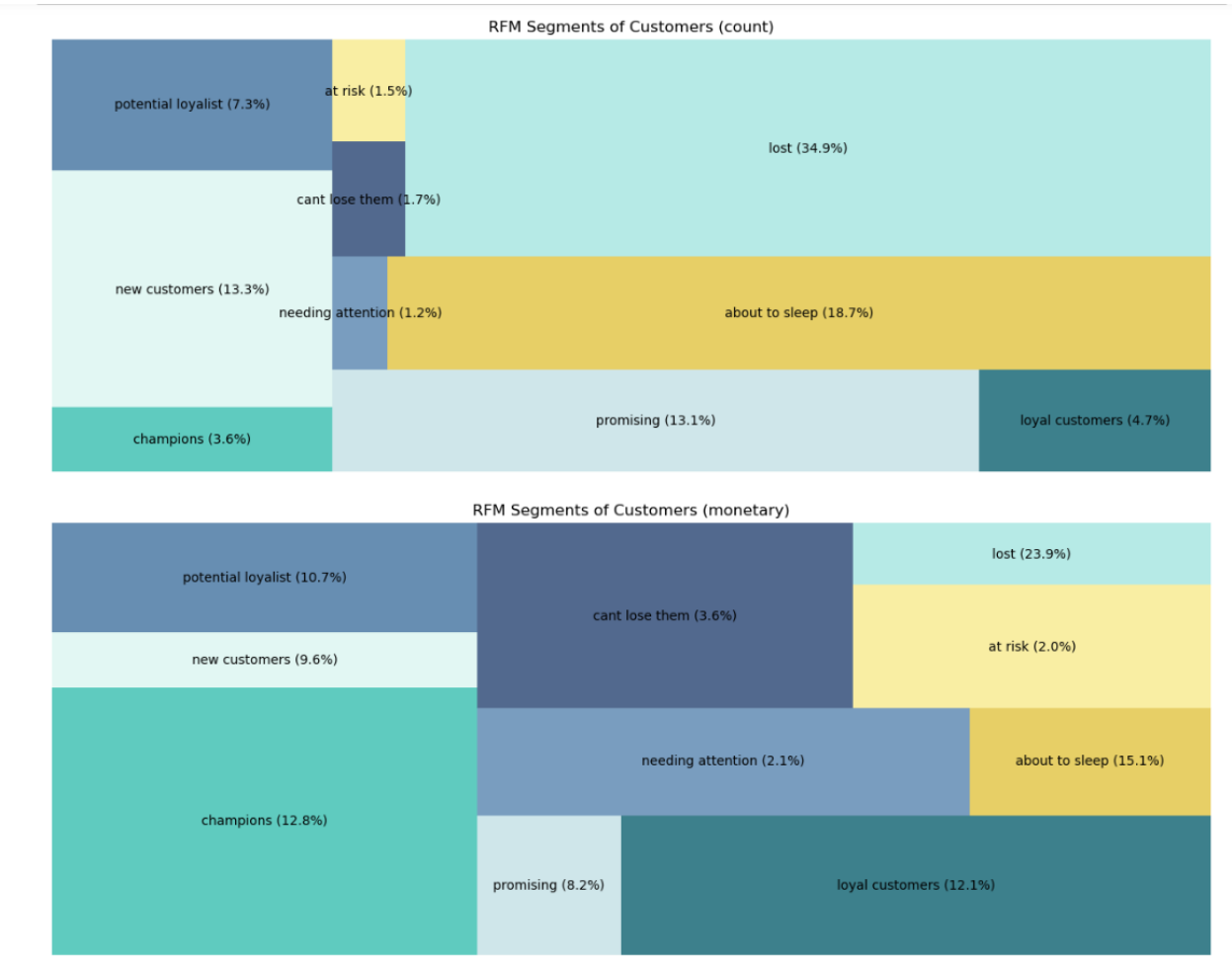
با توجه به حدسی که زده بودیم متوجه شدیم که تعداد مشتریانی که در دسته بندی lost قرار گرفته اند در عامل recency نسبت به تعداد مشتریان وفادار تعداد قابل توجهی هستند.

برای مشاهده ی بهتر این مقایسه تعداد مشتریان را در هر یک از دسته بندی های RFM مشاهده کردیم.

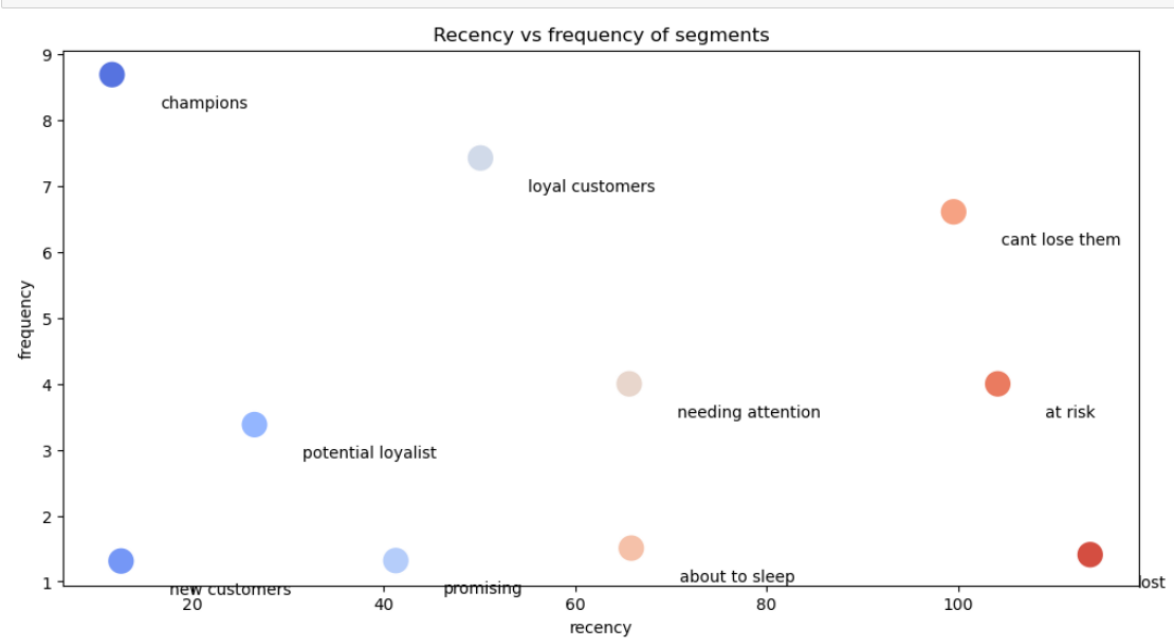


با توجه به این مقایسه متوجه شدیم که بیشترین تاثیر را در دسته بندی موجود عامل Recency دارد و احتمالا خرید به صورت کلی در بازه ی نزدیک رو به کاهش بوده است.

درصد قرار گیری مشتری ها در دسته بندی در نمودار زیر قابل مشاهده است.



با توجه به اینکه Recency تاثیر زیادی روی سگمنتیشن موجود داشته است با استفاده از نمودار زیر سعی کردم با اضافه کردن یک عامل دیگر صحیح بودن سگمنتشن را بررسی کنم که عامل دیگری که اضافه شد Frequency بود.



با توجه به نمودار به طور مثال زمانی که می گوییم مشتریانی وجود دارند که از دست رفته اند "Lost"، در نمودار در مقایسه با هر معیار بررسی شده در کمترین قسمت قرار دارند.

بعد از سگمنت کردن مشتری ها بر اساس الگوی خرید به بررسی مشتری های وفادار و از دست رفته پرداختم: (البته طبق الگوی ما در نظر گرفتن فقط یک دیتا مثل تعداد خرید عاملی برای سگمنتیشن نیست اما فقط برای اینکه دیدی داشته باشیم مورد استفاده قرار می گیرد.)

به طور مثال متوجه شدم که مشتری پس از 5 بار خرید می تواند جزو مشتری های وفادار باشد و مشتریان وفادار به صورت میانگین 7 خرید داشتند.

```
In [129]: top=least_item[least_item["rfm_label"]=='loyal customers']["order_number"].max()
          but=least_item[least_item["rfm_label"]=='loyal customers']["order_number"].min()
          avg=least_item[least_item["rfm_label"]=='loyal customers']["order_number"].mean()

In [130]: top
Out[130]: 50

In [131]: but
Out[131]: 5

In [132]: avg
Out[132]: 7.427709144243245
```

و متوجه شدم زمانی که 1 ماه از آخرین زمان سفارش مشتری گذشته باشد در دسته بندی churn شده قرار می گیرد.

```
In [151]: top2=last_item[last_item["rfm_label"]=='lost']["created_at"].max()
          but2=last_item[last_item["rfm_label"]=='lost']["created_at"].min()
          avg2=last_item[last_item["rfm_label"]=='lost']["created_at"].mean()

In [152]: top2
Out[152]: 3

In [153]: but2
Out[153]: 1

In [154]: avg2
Out[154]: 1.4076860341923338
```

در مورد الگوی رفتاری مشتریان وفادار به صورت میانگین در طول زمان دیتا 7 خرید دارند (Average Purchase Frequency) در صورتی که این عدد برای کل مشتریان 2 خرید بوده و از طرفی همین معیار برای مشتریان از دست رفته نیز 2 خرید بوده است.

در ادامه متوجه شدم که پس از کتگوری A که به نحوی برای تمامی مشتریان جذاب بوده و در آن خرید اتفاق افتاده است برای مشتریان وفادار کتگوری C در رتبه ی دوم قرار دارد که همین مورد در مشتریان از دست رفته نیز قابل مشاهده است.

```
Average Purchase Frequency for lost: 2.12 orders per customer

Category Preferences for lost (Percentage of Total Orders):
main_category
A   25.31
C   21.05
B   20.31
H    9.65
G    7.99
E    7.46
D    4.74
F    3.49
Name: proportion, dtype: float64

City Distribution for lost (Percentage of Total Orders):
city
A2     56.72
A3      5.37
A16     3.82
A4      3.54
A17     3.36
       ...
A179    0.00
A148    0.00
A139    0.00
A147    0.00
A185    0.00
Name: proportion, Length: 183, dtype: float64
```

```
Average Purchase Frequency for Loyal Customers: 7.43 orders per customer

Category Preferences for Loyal Customers (Percentage of Total Orders):
main_category
A   23.87
C   21.52
B   16.62
H   10.87
G   10.57
E    7.64
D    4.94
F    3.96
Name: proportion, dtype: float64

City Distribution for Loyal Customers (Percentage of Total Orders):
city
A2     60.71
A3      4.38
A16     3.62
A4      3.16
A17     2.90
       ...
A144    0.00
A175    0.00
A137    0.00
A156    0.00
A121    0.00
Name: proportion, Length: 166, dtype: float64
```

```
Average Purchase Frequency: 2.25 orders per customer

Category Preferences (Percentage of Total Orders):
main_category
A   24.28
B   21.67
C   20.39
H   10.26
G    9.28
E    6.76
D    4.16
F    3.19
Name: proportion, dtype: float64

City Distribution (Percentage of Total Orders):
city
A2     56.47
A3      5.20
A16     3.70
A4      3.49
A17     3.43
       ...
A178    0.00
A148    0.00
A170    0.00
A184    0.00
A185    0.00
Name: proportion, Length: 185, dtype: float64
```

در تحلیلی دیگر می توان مواردی مانند میانگین درصد تخفیف، میانگین درصد وچر، میانگین تعداد آیتم ها در هر سفارش و میانگین پرداختی هزینه ی ارسال را در مورد این سگمنت ها بررسی کرد:

مشتریان از دست رفته :

```
Average Discount Rate for lost: 0.22
Average Voucher Discount Rate for lost: 0.04
Average Items per Order for lost: 1.73
Average Final Shipping Fee for lost: 289.10
```

مشتریان از وفادار :

```
Average Discount Rate for Loyal Customers: 0.26
Average Voucher Discount Rate for Loyal Customers: 0.04
Average Items per Order for Loyal Customers: 1.88
Average Final Shipping Fee for Loyal Customers: 2473.19
```

کل مشتریان:

```
Average Discount Rate for All Customers: 0.22
Average Voucher Discount Rate for All Customers: 0.04
Average Items per Order for All Customers: 1.77
Average Final Shipping Fee for All Customers: 3652.54
```

میانگین درصد تخفیف روی آیتم ها برای مشتری های وفادار نسبت به عموم مشتری ها و مشتری های از دست رفته بیشتر بوده است، این دیتا نشان می دهد که تخفیف روی آیتم ها به صورت مستقیم برای تبدیل مشتری ها به مشتری های وفادار تاثیر به سزایی دارد.

میانگین پرداختی هزینه ی ارسال در مشتری های وفادار از عموم مشتری ها کمتر بوده ولی در مقایسه با مشتری های از دست رفته عدد بالاتری بوده است از طرفی مشتری های از دست رفته به صورت میانگین هزینه ی خیلی کمی برای ارسال پرداخت کردند اما مجدد خرید نکردند به همین دلیل به این نتیجه میرسیم که تخفیف روی هزینه ی ارسال تاثیر آنچنانی روی تبدیل مشتری ها به مشتری های وفادار ندارد.