Customer Segmentation Project

Table of Content

Business Objective

Cửa hàng X chủ yếu bán các sản phẩm thiết yếu cho khách hàng như rau, củ, quả, thịt, cá, trứng, sữa, nước giải khát... Khách hàng của cửa hàng là khách hàng mua lẻ.

Chủ cửa hàng X mong muốn có thể bán được nhiều hàng hóa hơn cũng như giới thiệu sản phẩm đến đúng đối tượng khách hàng, chăm sóc và làm hài lòng khách hàng.

Method

Using RFM Analysis with Sklearn K-Means and GMM vs Pyspark Kmean and GMM

Data Requirement

01

Products_with_Categories.csv: toàn bộ sản phẩm của cửa hàng kèm theo giá và loại sản phẩm

02

Transactions.csv: thông tin các giao dịch của khách hàng gồm: Member_number, Date, productId, Items

Data preparation steps

Step 1

Xử lý cột ngày (Date)

- Chuyển Date từ chuỗi sang kiểu datetime.
- Tính ngày tham chiếu (max date) để tính Recency.

Step 2

Tạo biến giá trị giao dịch (amount)

- Công thức: amount = price * items.
- Dùng để tính Monetary (tổng chi tiêu).

Step 3

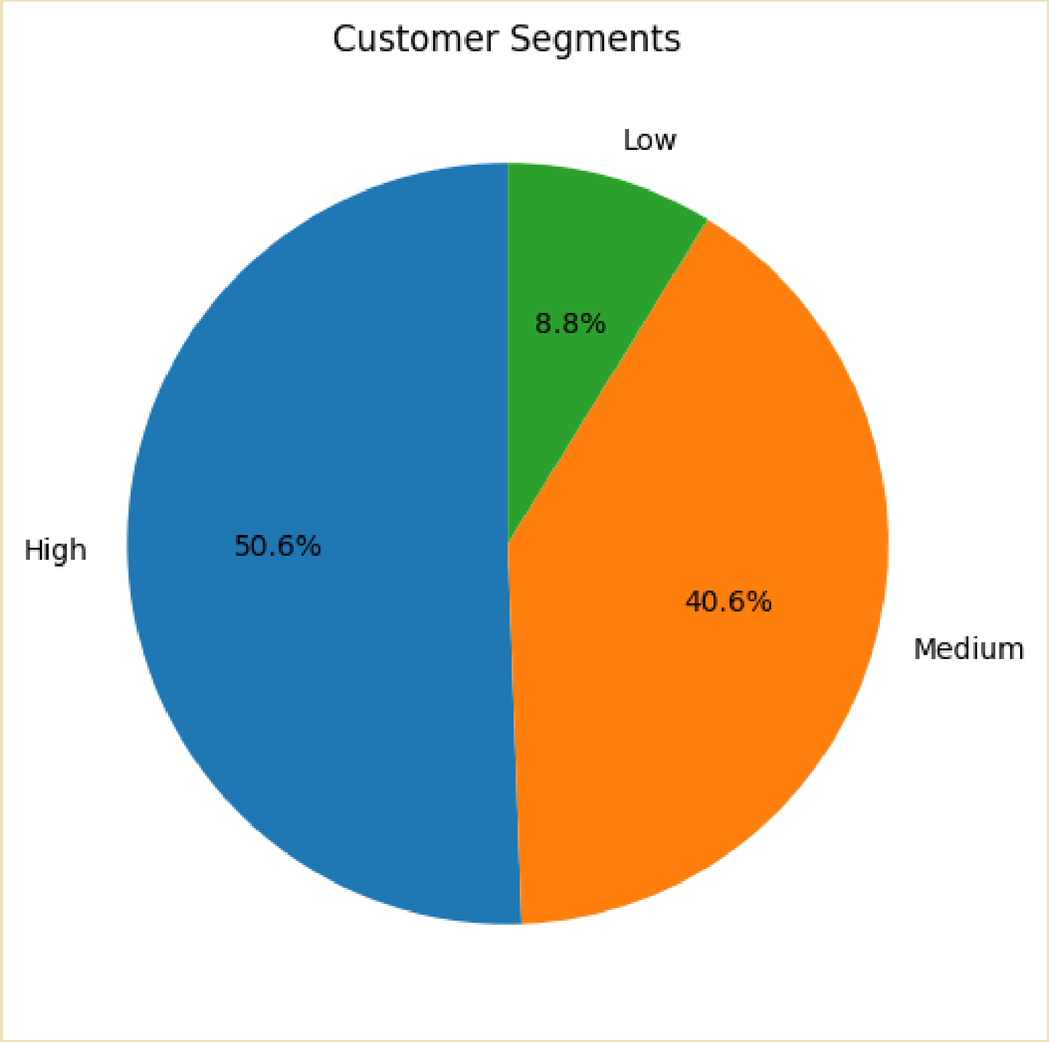
Tạo bảng RFM

- Recency: (ngày tham chiếu - ngày mua cuối cùng) theo khách hàng.
- Frequency: số lần mua hàng.
- Monetary: tổng amount.

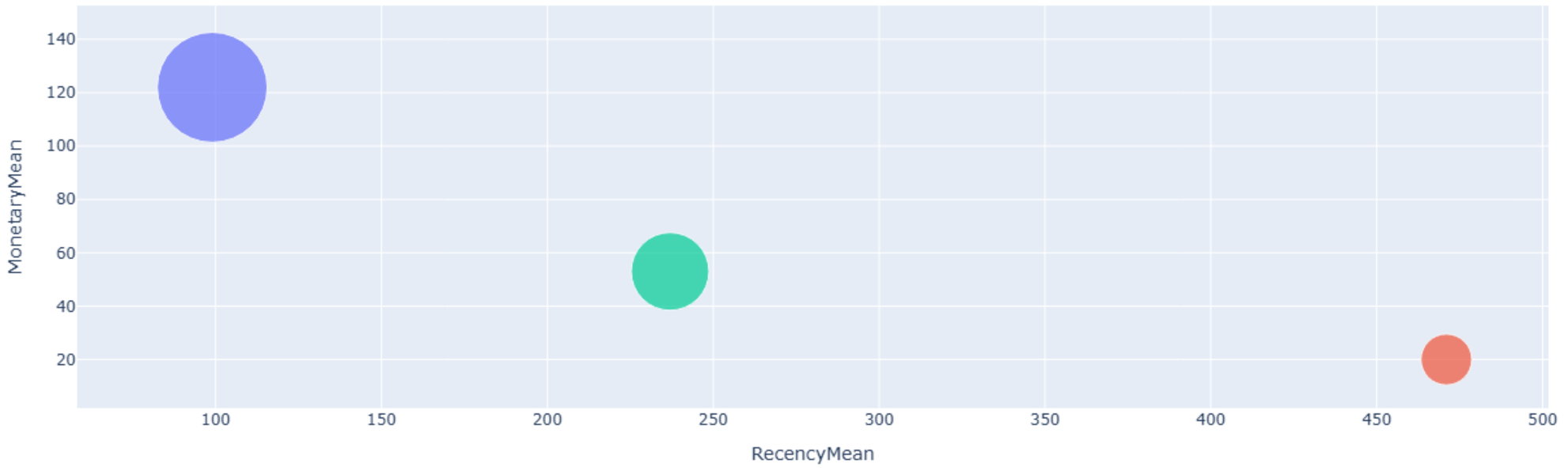
EDA result

	Member_number	Recency	Frequency	Monetary	R	F	M	RFM_Segment
1388	2433	3	31	375.31	4	4	4	444
1159	2193	91	27	361.45	3	4	4	344
772	1793	26	25	345.10	4	4	4	444
2225	3289	4	29	334.15	4	4	4	444
1692	2743	142	17	312.46	2	4	4	244

Customer Segments Pie chart



Scatter

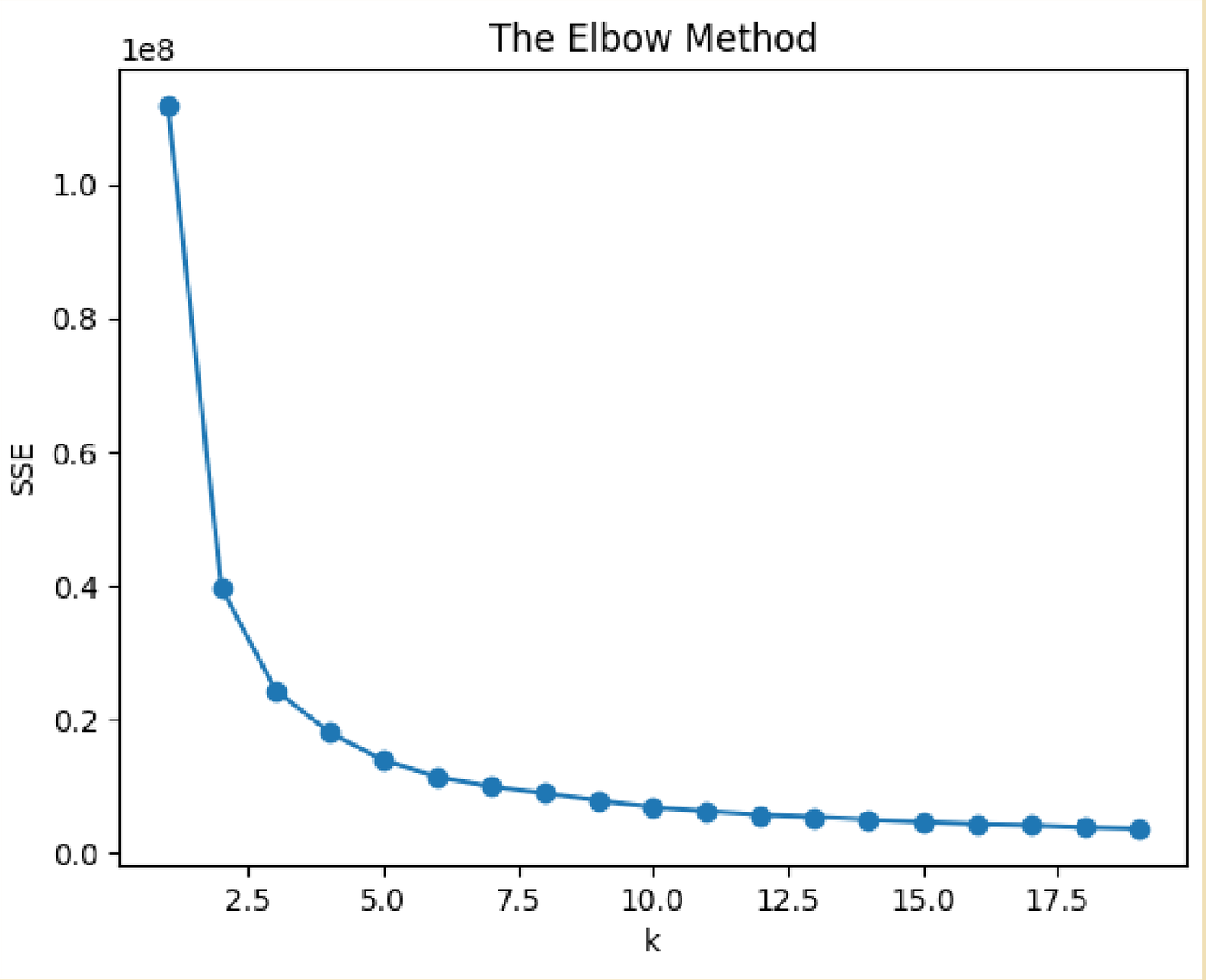


- Nhóm High tỷ lệ quay lại cao, khoảng cách ngắn, độ chi tiêu lớn hơn 2 nhóm còn lại.
- Nhóm Medium tỷ lệ quay lại trung bình, nhưng độ chi tiêu so với nhóm Low không lớn hơn nhiều.

=> Có thể nâng cao độ chi tiêu của nhóm Medium này lên

Using Sklearn K-Means

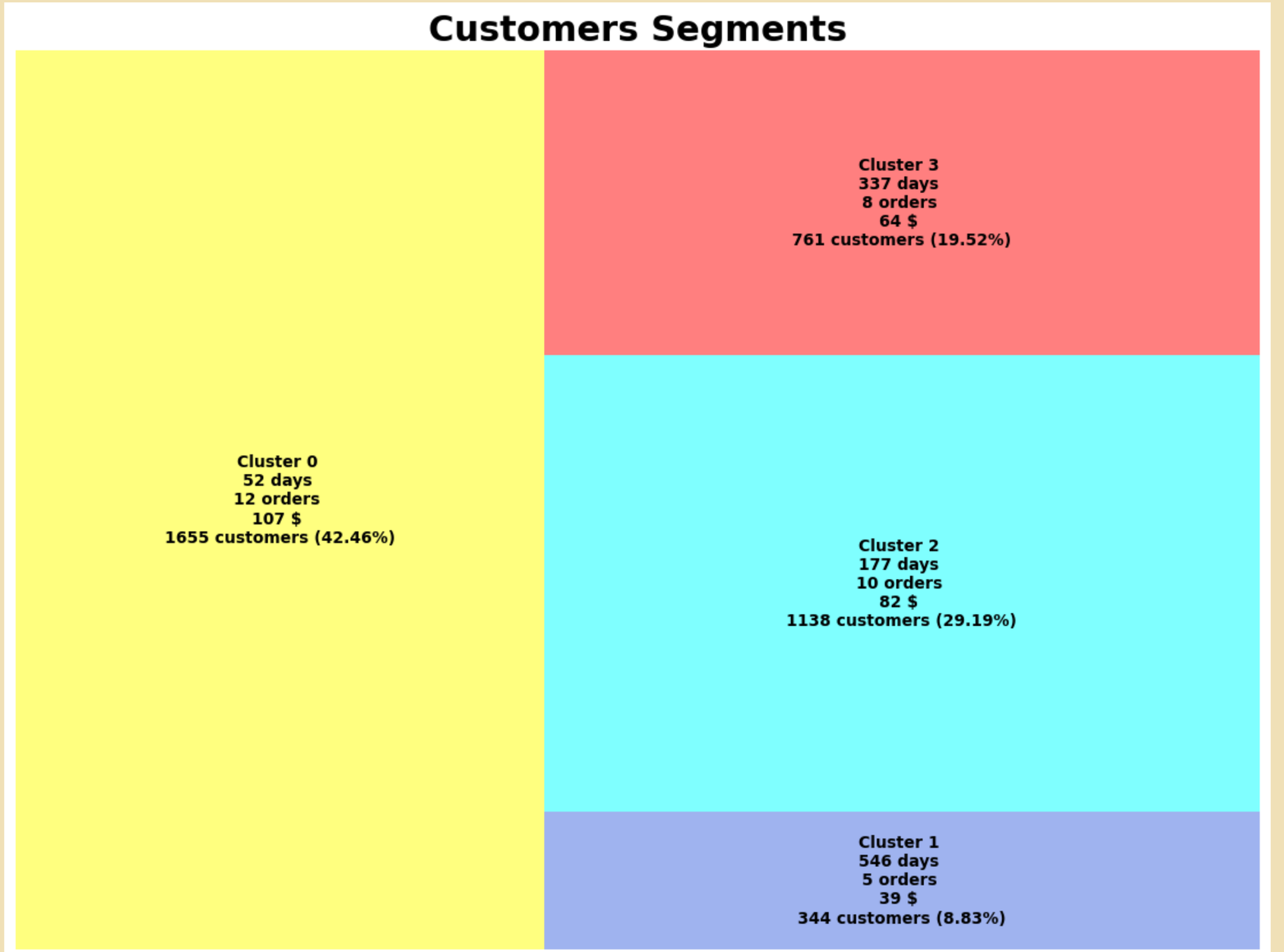
Chọn hệ Số K thích hợp



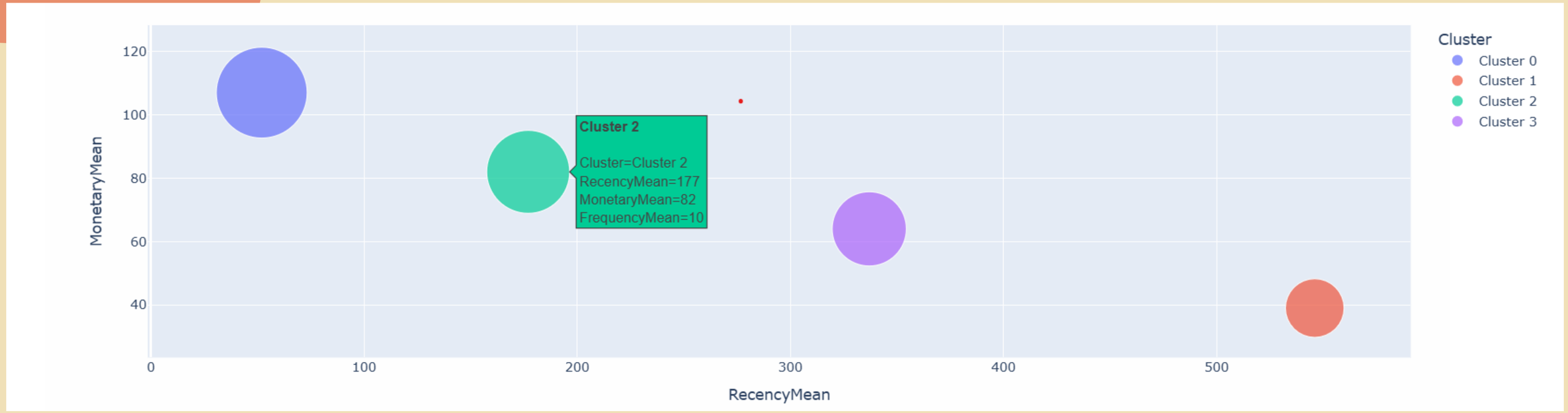
Using Sklearn K-Means

	Cluster	RecencyMean	FrequencyMean	MonetaryMean	Count	Percent
0	Cluster 0	52.0	12.0	107.0	1655	42.46
1	Cluster 1	546.0	5.0	39.0	344	8.83
2	Cluster 2	177.0	10.0	82.0	1138	29.19
3	Cluster 3	337.0	8.0	64.0	761	19.52

Sklearn K-Mean Tree Map



Sklearn K-Means



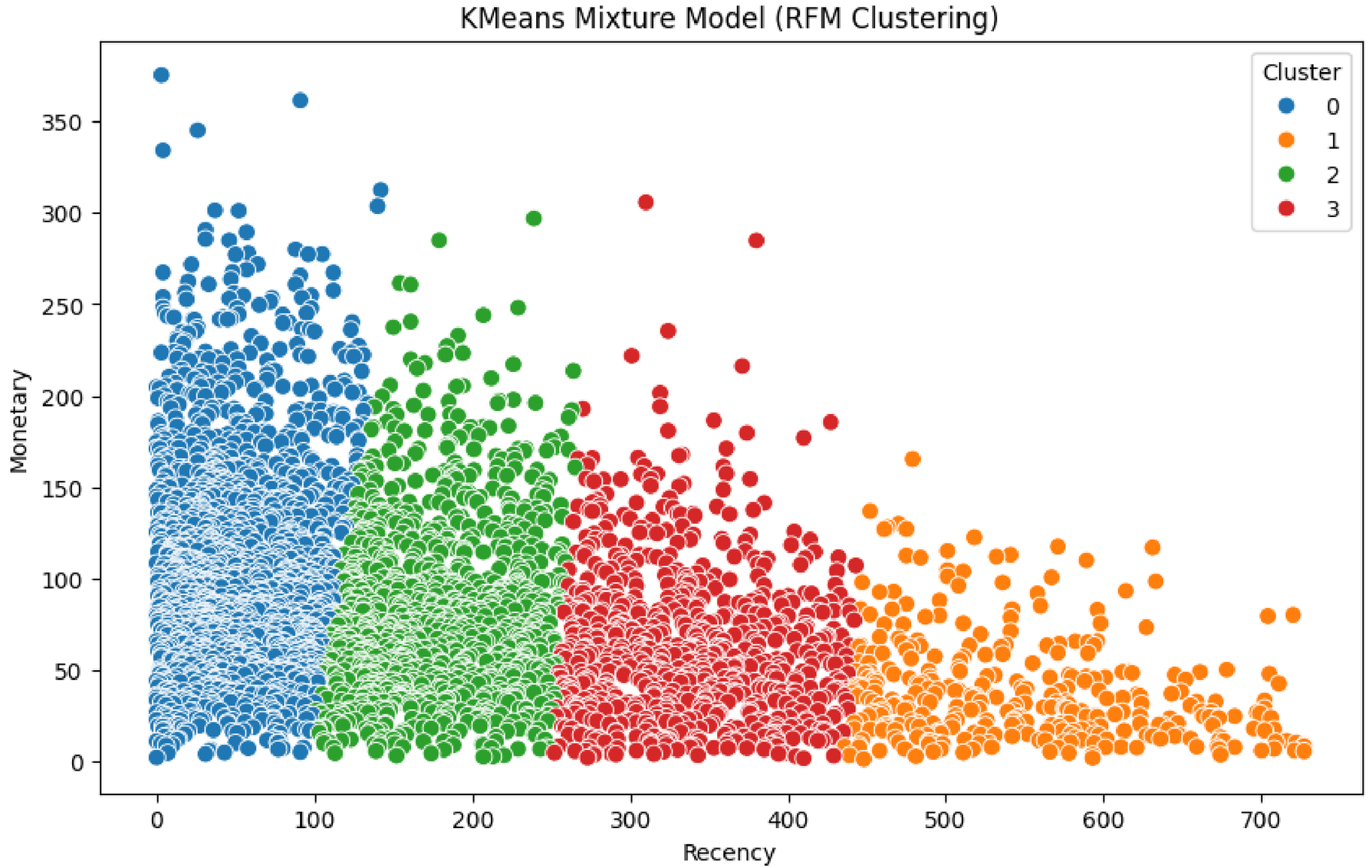
Nhóm Cluster 0 là nhóm khách trung thành, VIP

Nhóm Cluster 2 là nhóm khách có tiềm năng, điểm quay lại khá thường xuyên

Nhóm Cluster 3 là nhóm khách lâu không quay lại sắp rời bỏ

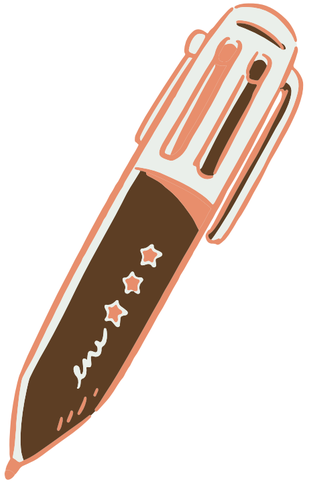
Nhóm Cluster 1 là nhóm khách sắp mất, tiêu ít, khó quay lại

Sklearn K-Mean

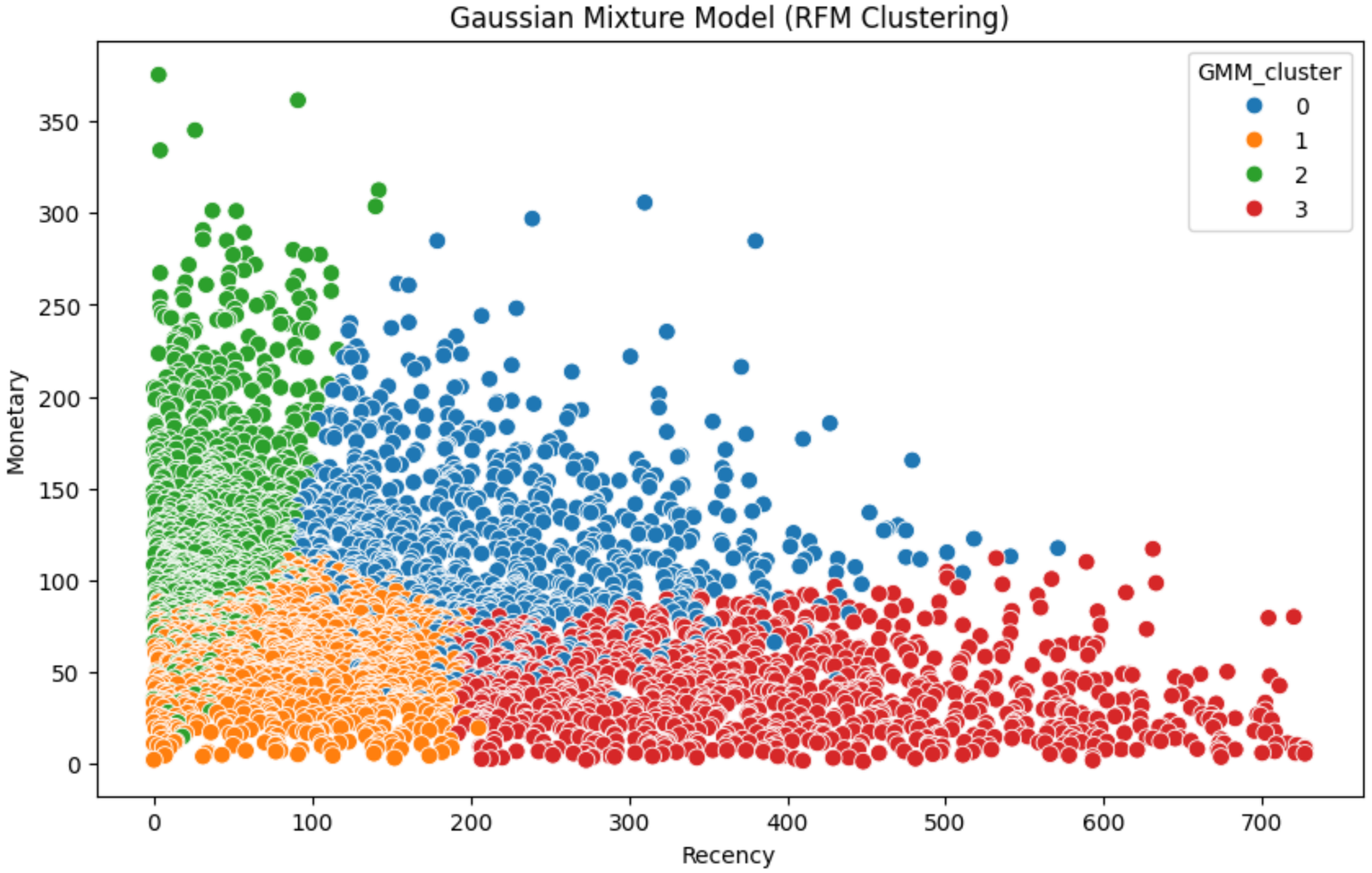


Sklearn GMM

GMM_cluster	Recency	Frequency	Monetary
0	218.56	13.09	122.43
1	93.61	7.89	55.46
2	40.37	14.77	139.16
3	386.55	5.51	41.17



Sklearn GMM

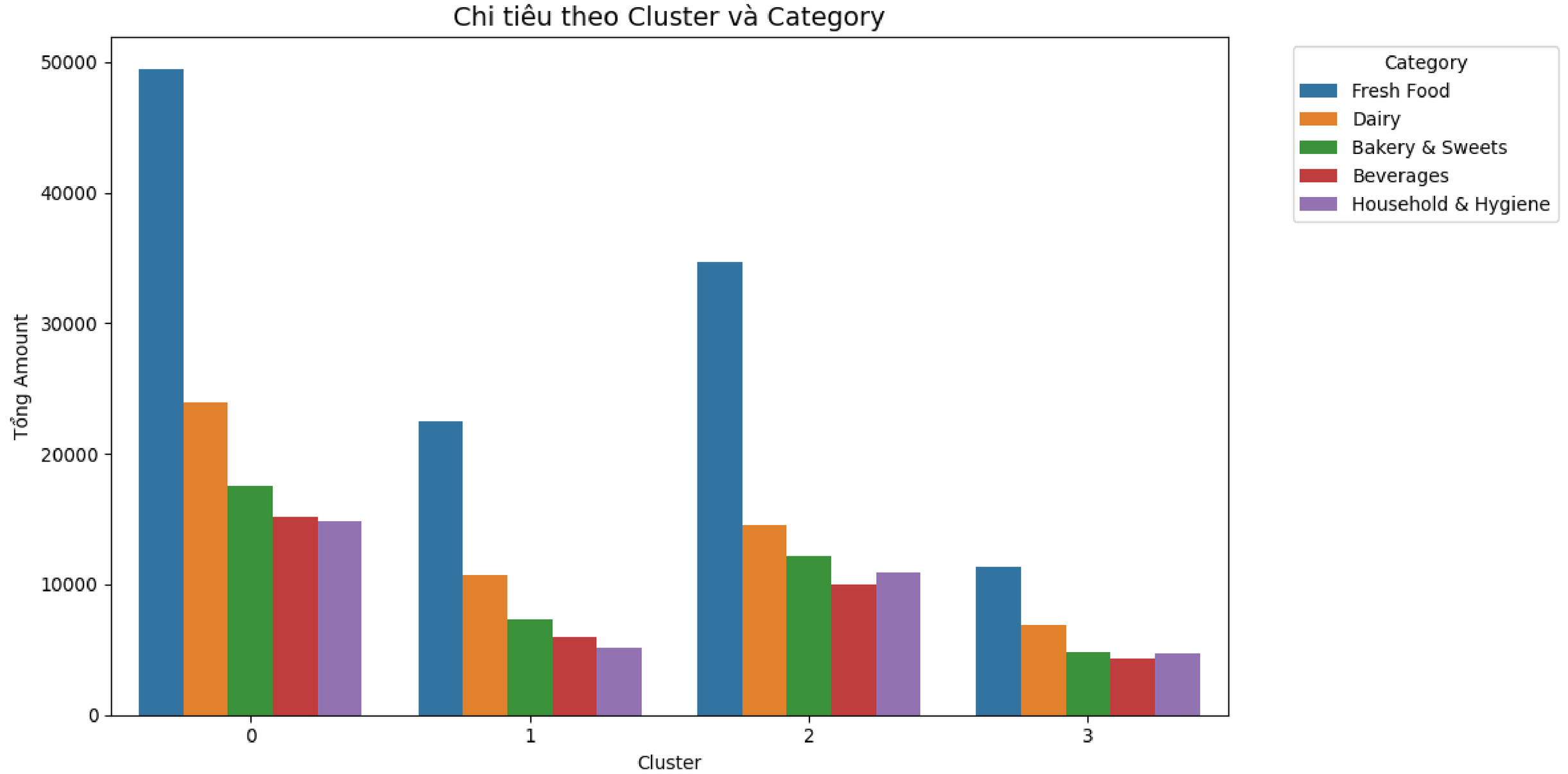

Gaussian Mixture Model (RFM Clustering)
GMM_cluster
0
1
2
3
Monetary
Recency
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600
700

Trong Sklearn giữa GMM và Kmeans thì Kmeans phân cụm tốt hơn nên chọn Kmean

Top Category by Cluster

	Cluster	Category	amount
0	0	Fresh Food	49443.80
1	0	Dairy	23946.98
2	0	Bakery & Sweets	17587.26
3	0	Beverages	15169.30
4	0	Household & Hygiene	14858.61
5	1	Fresh Food	22508.80
6	1	Dairy	10759.32
7	1	Bakery & Sweets	7307.44
8	1	Beverages	5955.50
9	1	Household & Hygiene	5133.90
10	2	Fresh Food	34704.40
11	2	Dairy	14558.26
12	2	Bakery & Sweets	12208.55
13	2	Household & Hygiene	10914.92
14	2	Beverages	10001.90
15	3	Fresh Food	11376.35
16	3	Dairy	6880.88
17	3	Bakery & Sweets	4825.33
18	3	Household & Hygiene	4743.31
19	3	Beverages	4329.60

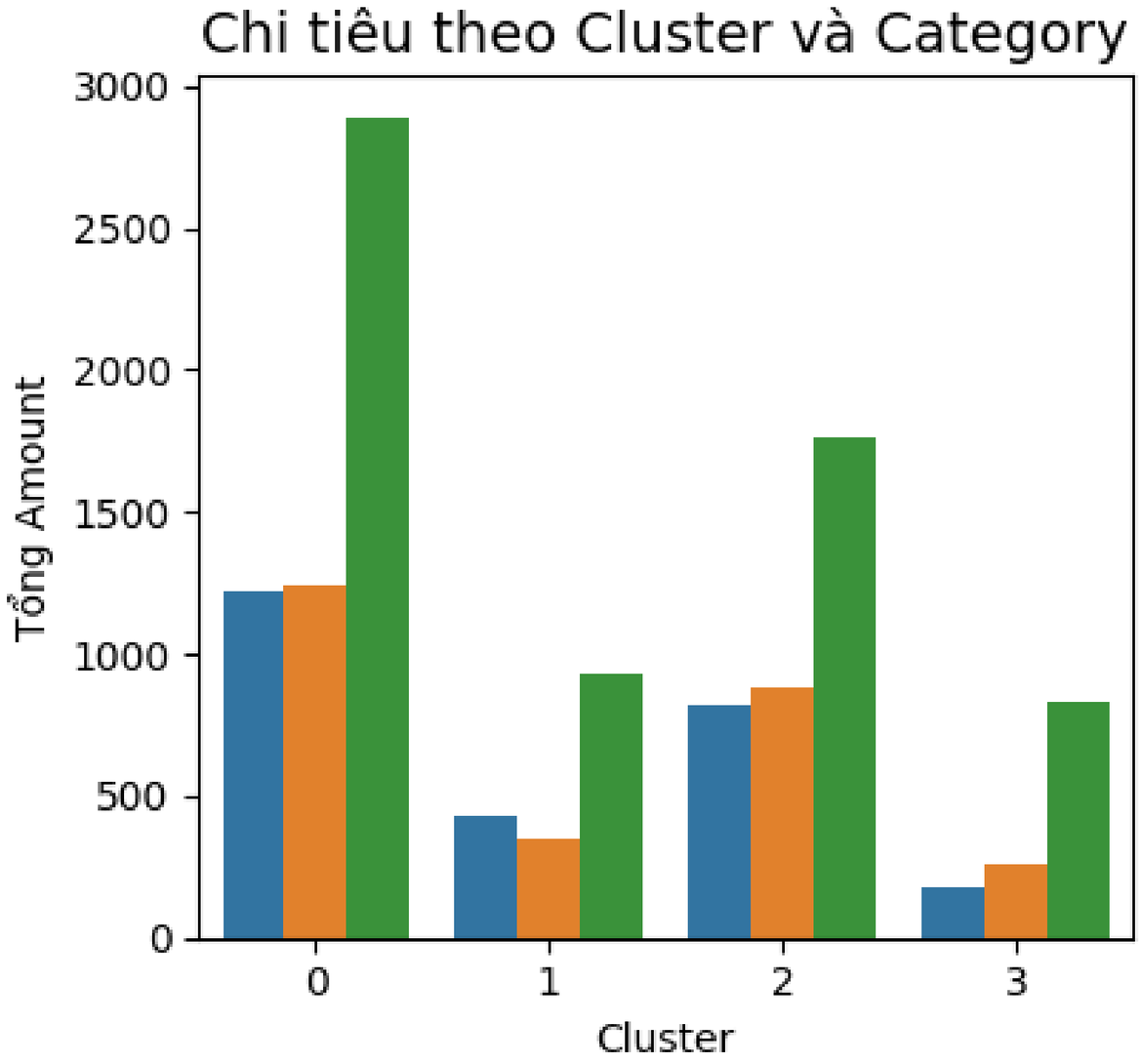
Top Category by Cluster



Top Low Category by Cluster

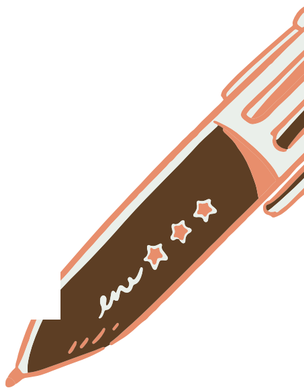
	index	Cluster	Category	amount
0	7	0	Personal Care	1218.20
1	9	0	Snacks	1239.50
2	8	0	Pet Care	2889.60
3	20	1	Snacks	350.90
4	18	1	Personal Care	428.70
5	19	1	Pet Care	934.20
6	29	2	Personal Care	818.70
7	31	2	Snacks	881.50
8	30	2	Pet Care	1767.60
9	40	3	Personal Care	175.85
10	42	3	Snacks	256.50
11	41	3	Pet Care	834.40

Top Low Category by Cluster

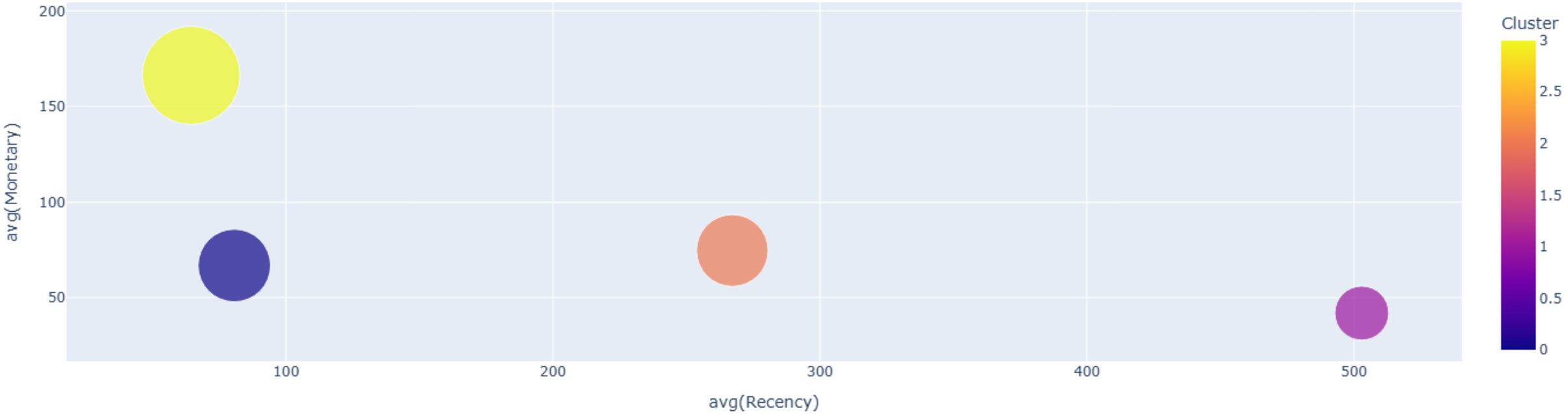


Using Pyspark K-Means

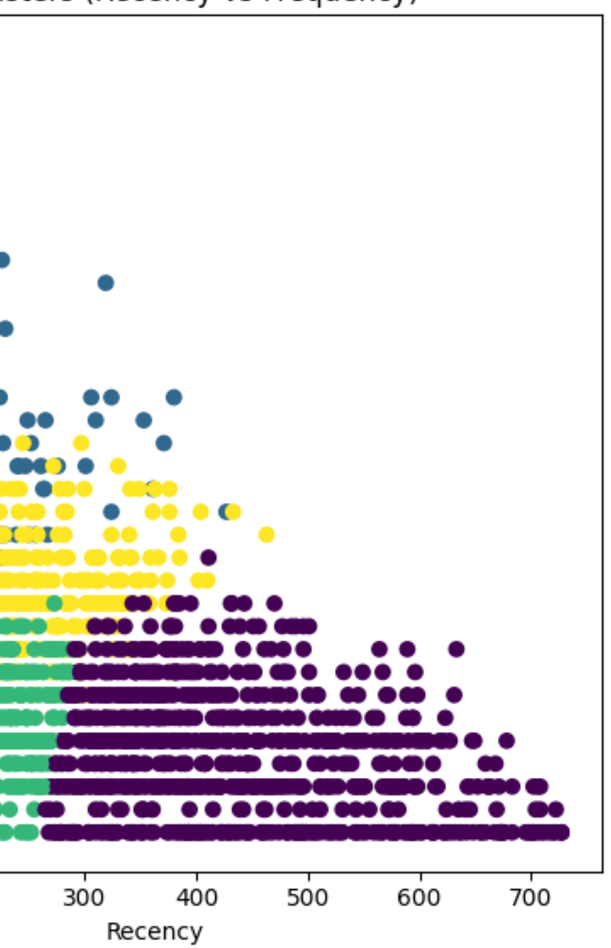
```
+-------+-----------------+-----------------+-----------------+--------+
|Cluster|      avg(Recency)|     avg(Monetary)|    avg(Frequency)|count(1)|
+-------+-----------------+-----------------+-----------------+--------+
|      1| 88.51097804391217|187.94173652694616|19.157684630738522|     501|
|      3|124.70099160945843|106.68883295194517|12.469870327993897|    1311|
|      2|123.68032786885246| 49.30309016393434| 6.667213114754098|    1220|
|      0|430.65011547344113|43.859110854503484| 5.409930715935335|     866|
+-------+-----------------+-----------------+-----------------+--------+
```

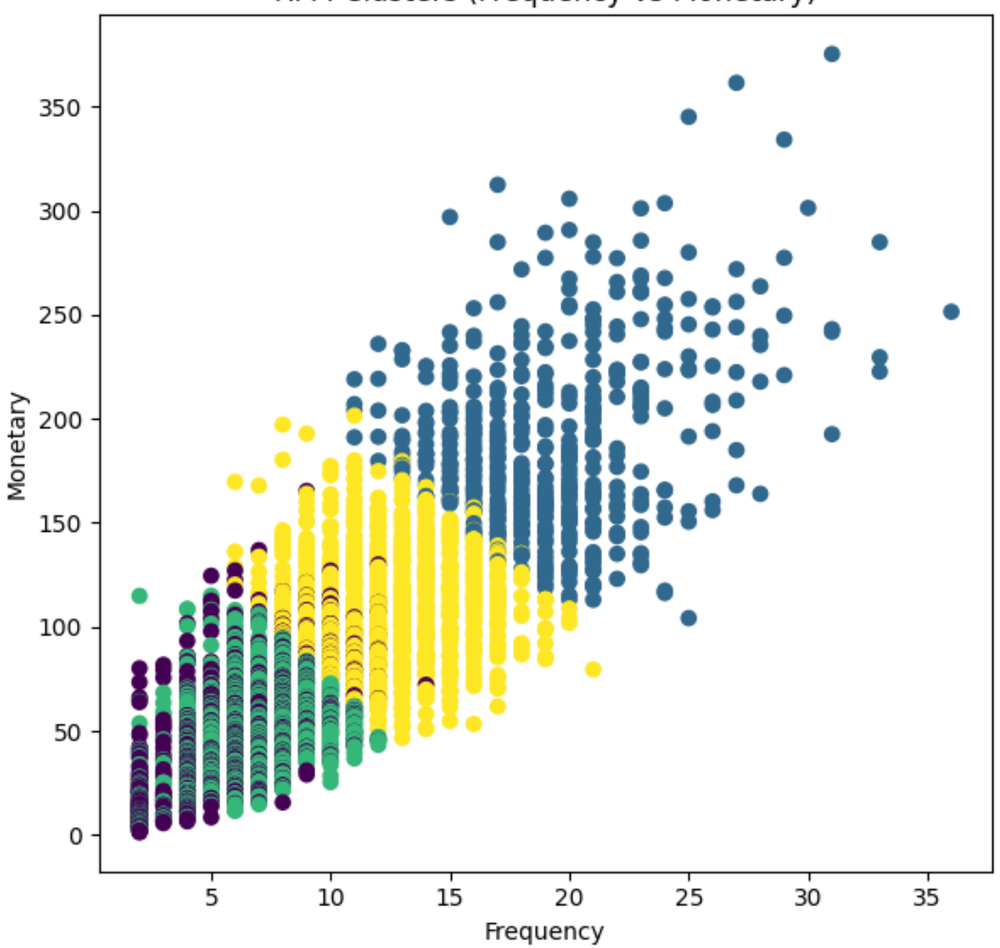


Using K-Means + RFM result Pyspark



Using Pyspark K-Means



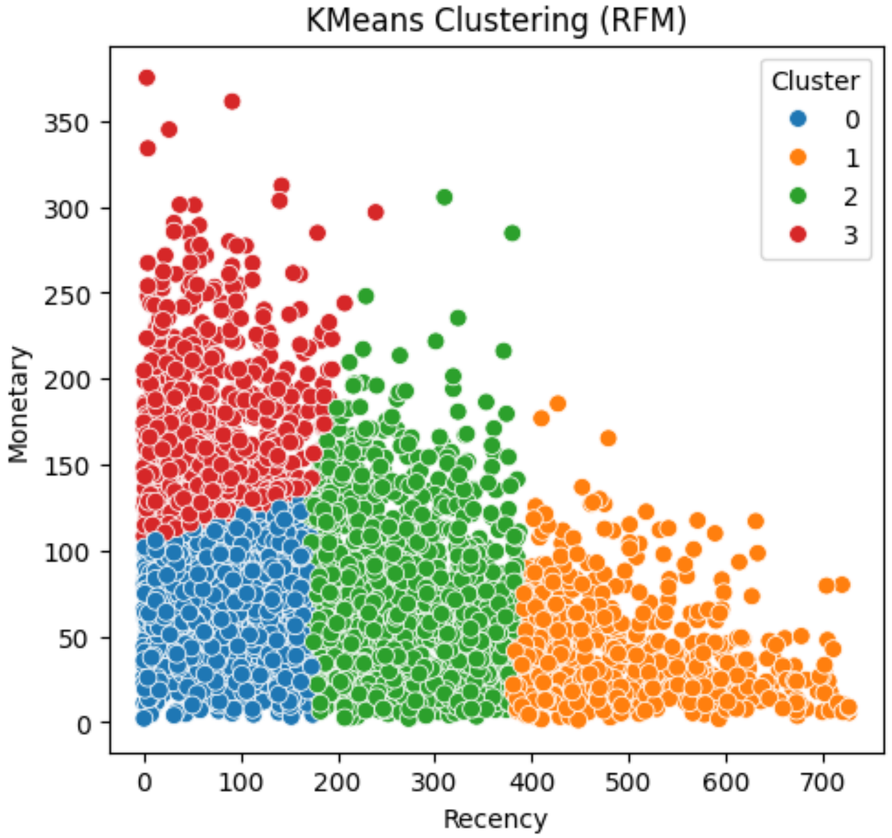


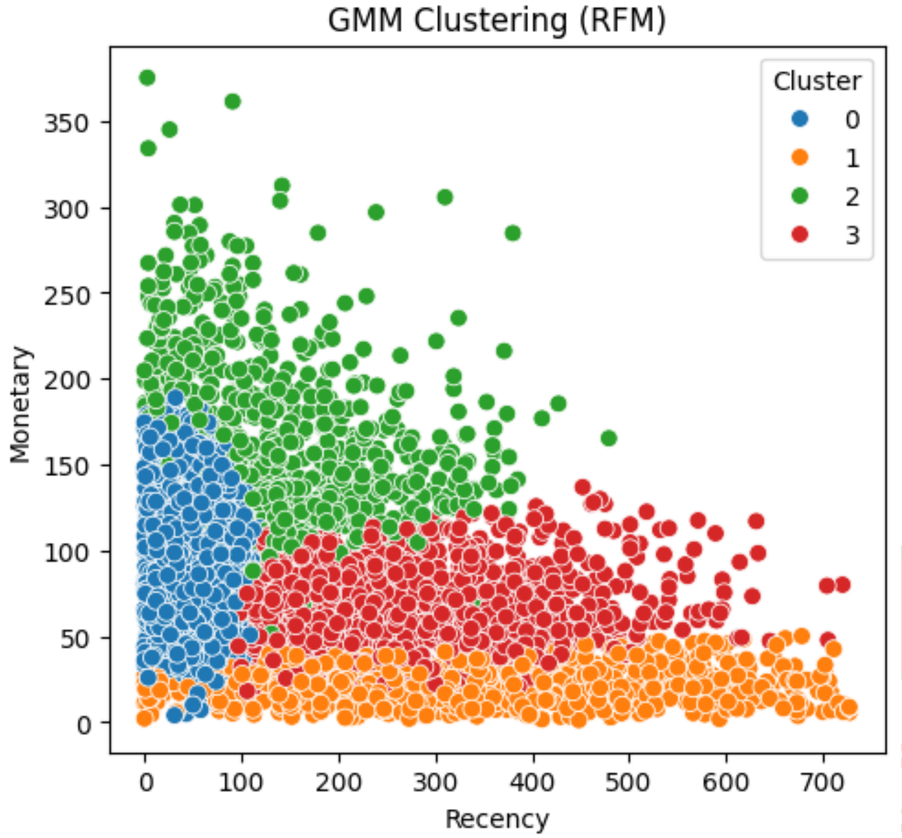
Using Pyspark GMM

Recency	Frequency	Monetary	features	scaledFeatures	probability	gmm_cluster
35	13	53.8	[35.0,13.0,53.8]	[-0.9549047485593...	[0.70090045578193...	0
242	12	100.0	[242.0,12.0,100.0]	[0.33955163459685...	[2.21787794675946...	2
122	8	70.3	[122.0,8.0,70.3]	[-0.4108578628850...	[2.25156899279448...	3
323	8	60.65	[323.0,8.0,60.65]	[0.84607804539713...	[1.22274262713200...	1
28	21	204.96	[28.0,21.0,204.96]	[-0.9986786359125...	[0.88812720931359...	0
706	4	24.0	[706.0,4.0,24.0]	[3.24113502486014...	[7.88554125937265...	1
199	15	182.94	[199.0,15.0,182.94]	[0.07065489799918...	[1.01926199275532...	2
88	12	155.9	[88.0,12.0,155.9]	[-0.6234738871715...	[0.43778390436978...	2
86	9	48.75	[86.0,9.0,48.75]	[-0.6359807121295...	[0.02249030509850...	3
152	12	89.36	[152.0,12.0,89.36]	[-0.2232554885145...	[1.87536419333754...	3
21	13	89.8	[21.0,13.0,89.8]	[-1.0424525232656...	[0.63180488977712...	0
41	11	75.86	[41.0,11.0,75.86]	[-0.9173842736853...	[0.43724370452456...	3
89	19	186.75	[89.0,19.0,186.75]	[-0.6172204746925...	[0.38772926149021...	2
88	10	75.1	[88.0,10.0,75.1]	[-0.6234738871715...	[0.02452561569252...	3
240	7	71.4	[240.0,7.0,71.4]	[0.32704480963882...	[2.09646315491551...	1
86	11	104.7	[86.0,11.0,104.7]	[-0.6359807121295...	[0.08840687628840...	3
91	11	75.49	[91.0,11.0,75.49]	[-0.6047136497345...	[0.01973189727802...	3
221	8	94.6	[221.0,8.0,94.6]	[0.20822997253752...	[2.96985951634305...	2
385	2	73.5	[385.0,2.0,73.5]	[1.23378961909610...	[3.15611527025950...	2
78	10	73.84	[78.0,10.0,73.84]	[-0.6860080119617...	[0.06231676177161...	3

only showing top 20 rows

Compare Pysark Kmean vs Pyspark GMM

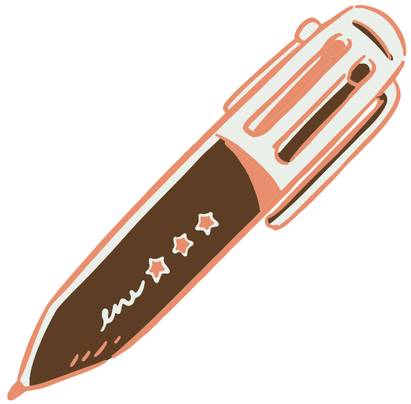




Compare Pyspark và SKlearn

Bảng so sánh Silhouette Score

Model	Framework	Silhouette Score
GMM	Sklearn	0.2385
KMeans	Sklearn	0.3315
GMM	PySpark	0.2080
KMeans	PySpark	0.5534



Compare Pyspark và SKlearn

KMeans > GMM về silhouette score

•Cả trong sklearn và PySpark, KMeans có score cao hơn GMM.

•Điều này cho thấy: các cluster do KMeans tạo ra **rõ ràng, tách biệt hơn** theo khoảng cách Euclidean so với cluster GMM.

Sklearn vs PySpark

•Silhouette score của PySpar thường cao hơn sklearn (đặc biệt là KMeans).

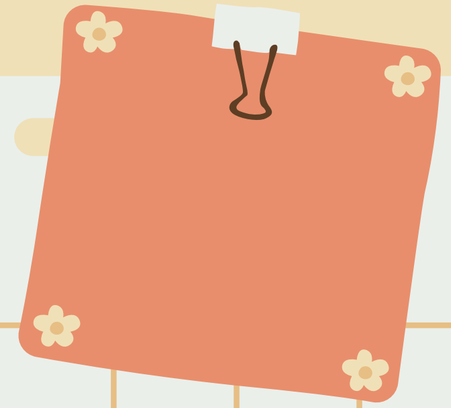
•Nguyên nhân có thể là cách tính khoảng cách (PySpark dùng squaredEuclidean) và cách scale dữ liệu.

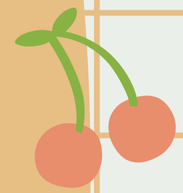
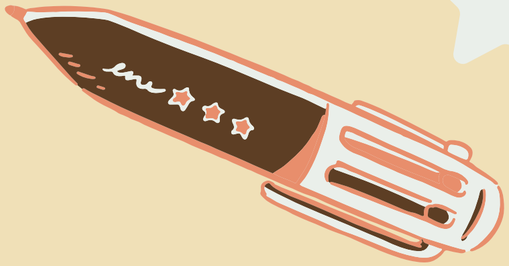
=>Chọn Kmeans Pyspark là tốt nhất

Comment

Nhìn chung cả 2 cách phân tích theo RFM và RFM + KMean + Pyspark đều cho thấy cửa hàng đang hoạt động tốt với sự phân bổ của nhóm khách hàng Trung thành OK + ổn áp, chiếm tỷ lệ lớn nhất. Nhóm khách Rủi ro cũng chỉ chiếm 10%-15%. Ngoài ra, nhóm có phân tích thêm từng nhóm Cluster thường mua những category nào. Tuy nhiên, cũng không thu lại được hiệu quả cao lắm.

=> Phần dataset này, nhóm đề xuất nếu có dataset về ngày mua + giờ mua sẽ phân tích được hiệu suất khách tập trung ở khung giờ nào.


Thank You