RFM And K-means for Customer Segmentation

Capstone Project - The Battle of Neighborhoods (Week 2) - Report

Qiao, Yipeng
7-8-2021

Table of Contents

1 Introduction

1.1 Background

I may have a convenience store after several years, then I will have many customers. I can use RFM (Recency, Frequency, Monetary Value) analysis to find out the customer structure.

What is RFM Analysis? RFM (Recency, frequency, monetary value) is a marketing analysis tool used to identify a company's or an organization's best customers by using certain measures. The RFM model is based on three quantitative factors:

- **Recency**: How recently a customer has made a purchase.
- **Frequency**: How often a customer makes a purchase.
- **Monetary Value**: How much money a customer spends on purchases.

After I have a RFM data, I will split the numbers into segments, then I can use K-means to find out how many customer classes I have, and what to do with them.

1.2 Problem

- Who are the best customers?
- Who are your loyal customers?
- Which customer is losing interest?
- Which customer have lost risk?
- Who are the lost customers?
- Does the store still run well?
- What should I do if it not well?

1.3 Interest

Such as store, bank, traffic, etc. RFM is really an easy and wide useful tool, everyone have customers can use it to understand the problems.

2 Data

The data have "InvoiceNo", "StockCode", "Description", "Quantity", "InvoiceDate", "UnitPrice", "CustomerID", "Country" in it, we need CustomerID to collect "Frequency". Quantity and UnitPrice to calculate the total amount which needed by "Monetary". InvoiceDate should be transformed into datatime for calculating "Recency". You can find more detail in 2.2 Data Explore ,2.3 Data clean and 2.4 Data processing to prepare part.

2.1 Data source

I found some data about sale from kaggle. You can dowload the data from kaggle. If you cannot access kaggle, you can download it from here.

2.2 Data explore

Show data info, we can find column name and data type, CustomerID should be integer, InvoiceDate should be datetime, they need convert. Other columns not used in my case, can be dropped.

```
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

Dataset columns

Describe data, Quantity and UnitPrice should not be less than 1, and should not be negative, we will fix it later.

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

Data describe result

Find if there is something missing, we can see "CustomerID" and "Description" have missing data, correct it later too.

```
CustomerID     135080
Description      1454
InvoiceNo           0
StockCode           0
Quantity            0
InvoiceDate         0
UnitPrice           0
Country             0
dtype: int64
```

Missing data in dataset

We also find 5227 duplicated data in dataset. Need to fix.

2.3 Data clean

- We don't need "InvoiceNo","StockCode","Country","Description", drop them from dataset.
- There is 135080 rows null data in CustomerID, the row should be deleted from the dataset.
- Find 5227 duplicated data and delete the rows from the dataset.
- Quantity and UnitPrice should not be less than 1 and should not be negative number.
- Format InvoiceDate, from string to datatime.

2.4 Data processing

2.4.1 Prepare RFM data

Find min and max date in InvoiceDate, use the next day as collectTime. Use collectTime and InvoiceDate to calculate the Recency (How recently a customer has made a purchase), named R.

Group by CustomerID and count rows number, it is Frequency (How often a customer makes a purchase), named F.

We use Quantity* UnitPrice as total amount, Group by CustomerID and sum total amount as Monetary Value (How much money a customer spends on purchases), named M.

Join R, F, M table by CustomerID and we got RFM data like below.

CustomerID	R	F	M
12346	325	1	77183.60
12347	2	182	4310.00
12348	75	31	1797.24
12349	18	73	1757.55
12350	310	17	334.40

RFM table (first 5 rows)

3 Methodology

3.1 Data segmentation

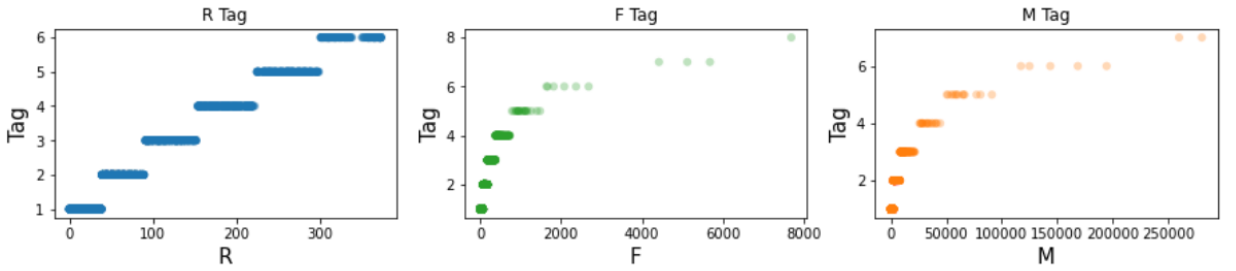
3.1.1 1D Data Segmentation

For 1D data, we use Percentile (20%, 40%, 60%, 80%) or Jenks Natural Breaks to split the data, then chose a better group

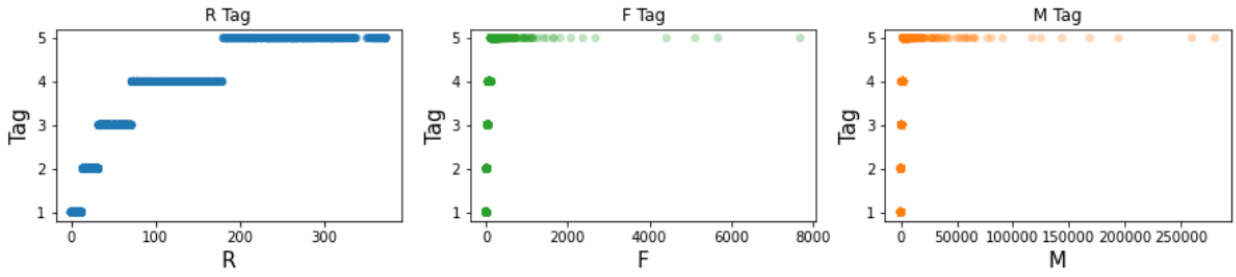
The Jenks Natural Breaks Classification (or Optimization) system is a data classification method designed to optimize the arrangement of a set of values into "natural" classes. A Natural class is the most optimal class range found "naturally" in a data set. A class range is composed of items with similar characteristics that form a "natural" group within a data set.

This classification method seeks to minimize the average deviation from the class mean while maximizing the deviation from the means of the other groups. The method reduces the variance within classes and maximizes the variance between classes. It is also known as the goodness of variance fit (GVF), which equals the subtraction of SDCM (sum of squared deviations for class means) from SDAM (sum of squared deviations for array mean).

Below are results for Jenks Natural Breaks and use Percentile (20%, 40%, 60%, 80%) split:



Split by Jenks Natural Breaks



Split by Percentile (20%, 40%, 60%, 80%)

After compare "Jenks Natural Breaks" and "Split by Percentile", it showed "R" split by percentile is ok, but "F" and "M" is not fit. We should use "Jenks Natural Breaks" for the segmentation.

Below is RFM segments.

type	1	2	3	4	5	6	7	8
R	[min,3528.0]	(3528.0,3579.0]	(3579.0,3642.0]	(3642.0,3712.0]	(3712.0,3789.0]	(3789.0,max]		
F	[min,68.0]	(68.0,182.0]	(182.0,378.0]	(378.0,740.0]	(740.0,1477.0]	(1477.0,2677.0]	(2677.0,5670.0]	(5670.0,max]
M	[min,2392.83]	(2392.83,8347.20]	(8347.20,21429.39]	(21429.39,44534.3]	(44534.3,91062.38]	(91062.38,194390.79]	(194390.79,max]	

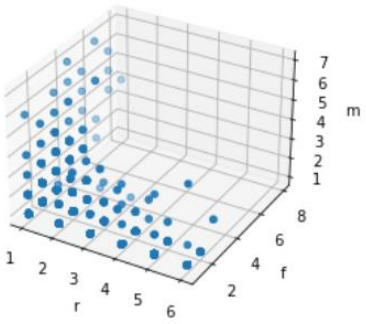
RFM segments

3.2 Classify customer level with RFM

3.2.1 Create taged RFM data

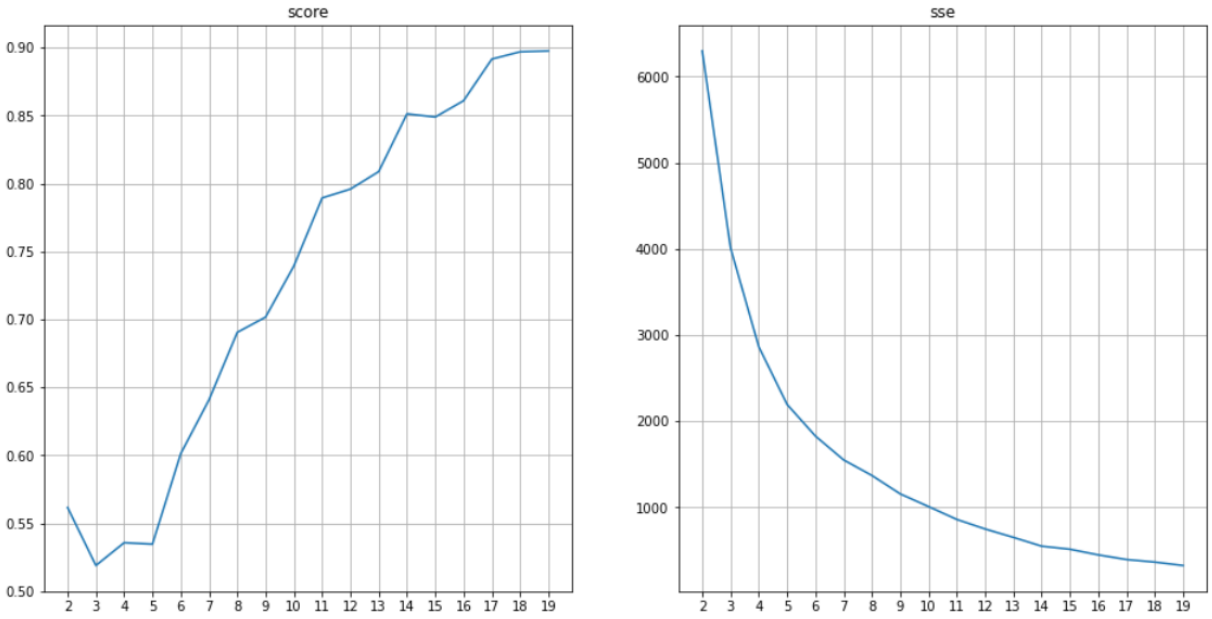
We will classify customer level, use tagged RFM data as below:

CustomerID	rTag	fTag	mTag
12346	6	1	5
12347	1	2	2
12348	2	1	1
12349	1	2	1
12350	6	1	1



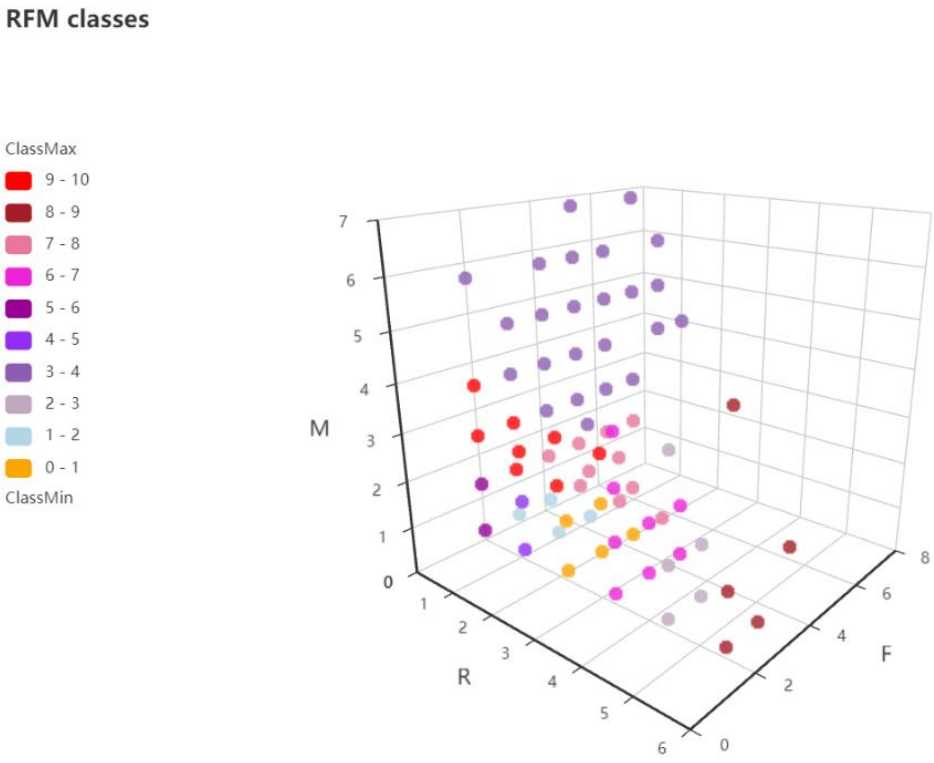
Tagged RFM (first 5 rows)

We run 20 times K-means and compare silhouette score and sse, the best K is 10.



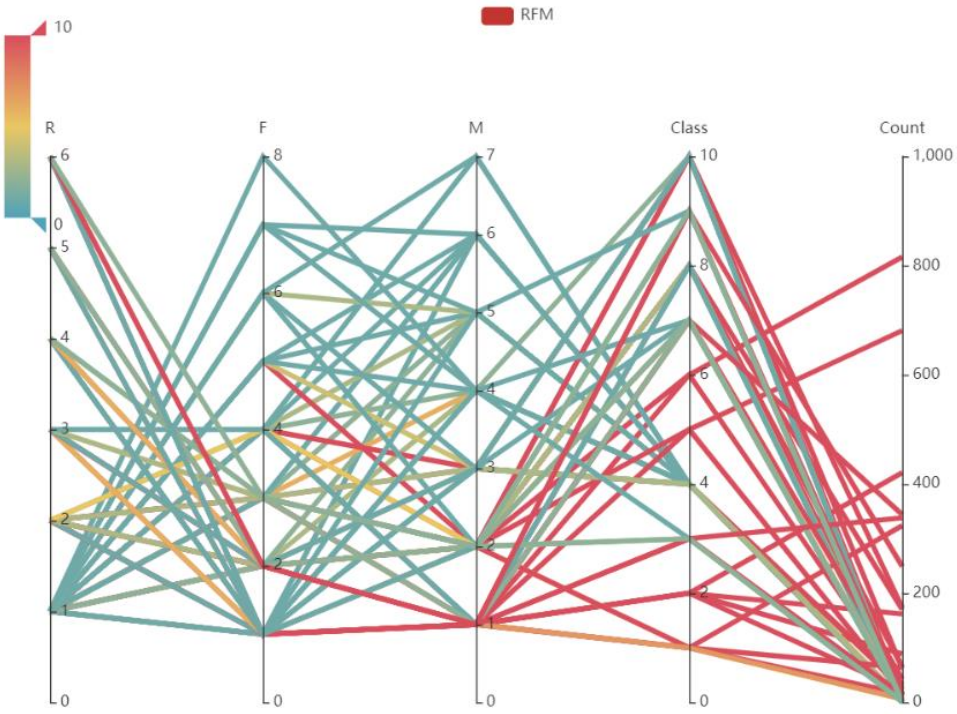
silhouette score and sse

At last we have a cluster mode. The distribution can be shown as below.



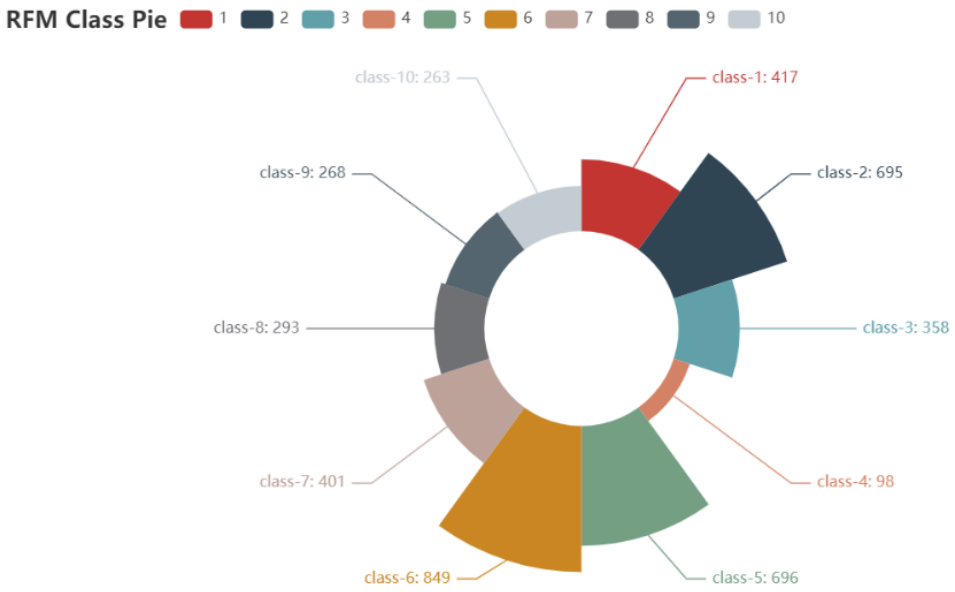
Scatter chart – classify by K-means

Also, we can use a parallel to show it with count



Parallel chart – classify by K-means, with count

And we can show the proportion



Pie chart – classify by K-means, with proportion

4 Result and Discussion

We have done the RFM segmentation and have the table below.

type	1	2	3	4	5	6	7	8
R	[min,3528.0]	(3528.0,3579.0]	(3579.0,3642.0]	(3642.0,3712.0]	(3712.0,3789.0]	(3789.0,max]		
F	[min,68.0]	(68.0,182.0]	(182.0,378.0]	(378.0,740.0]	(740.0,1477.0]	(1477.0,2677.0]	(2677.0,5670.0]	(5670.0,max]
M	[min,2392.83]	(2392.83,8347.20]	(8347.20,21429.39]	(21429.39,44534.3]	(44534.3,91062.38]	(91062.38,194390.79]	(194390.79,max]	

RFM segments

We created K-means model. K is 10, score is 0.749867, sse is 1002.771267.

Cluster centers is below:

R	F	M
1.000000	1.000000	1.038869
5.428115	1.055911	1.023962
1.081081	3.260442	1.825553
2.000000	1.000000	1.021552
4.000000	1.132832	1.032581
2.280702	2.108187	1.251462
1.000000	2.000000	1.000000
1.000000	4.242424	3.863636
3.000000	1.000000	1.018182
1.019802	1.970297	2.163366

classify cluster center

We can see rfm data and which class they belong, collect the classes as 6 level "new, normal, good, attention, risk, lost". Fnally we have 6 class 10 subclass and 73 detail class.

subclass	class
1	new
2	lost
3	normal
4	attention
5	risk
6	new
7	normal
8	lost
9	good
10	good

class subclass mapping

SeqNO	R Tag	F Tag	M Tag	Subclass	Count
1	1	1	1	1	816
2	1	1	2	1	33
3	5	1	1	2	338
4	5	1	2	2	3
5	5	1	4	2	1
6	5	2	1	2	14
7	5	2	2	2	2
8	6	1	1	2	249
9	6	1	2	2	1
10	6	1	5	2	1
11	6	2	1	2	15
12	6	3	2	2	2
13	1	3	1	3	89
14	1	3	2	3	17
15	1	3	3	3	27
16	1	4	1	3	11
17	1	4	2	3	65

SeqNO	R Tag	F Tag	M Tag	Subclass	Count
18	1	5	2	3	12
19	2	3	2	3	24
10	2	3	3	3	3
21	2	4	1	3	1
22	2	4	2	3	5
23	2	1	1	4	68
24	2	1	2	4	15
25	4	1	1	5	34
26	4	1	2	5	2
27	4	1	3	5	1
28	4	1	4	5	1
29	4	2	1	5	43
30	4	2	2	5	6
31	4	3	1	5	2
32	2	2	1	6	16
33	2	2	2	6	63
34	2	3	1	6	23
35	3	2	1	6	64
36	3	2	2	6	16
37	3	2	3	6	1
38	3	3	1	6	7
39	3	3	2	6	3
40	3	4	1	6	1
41	4	3	2	6	2
42	1	2	1	7	42
43	1	1	6	8	1
44	1	2	5	8	2

SeqNO	R Tag	F Tag	M Tag	Subclass	Count
45	1	3	4	8	6
46	1	3	5	8	2
47	1	3	6	8	1
48	1	4	3	8	30
49	1	4	4	8	2
50	1	4	5	8	3
51	1	4	6	8	1
52	1	4	7	8	1
53	1	5	3	8	4
54	1	5	4	8	1
55	1	5	5	8	1
56	1	5	6	8	1
57	1	6	2	8	1
58	1	6	3	8	1
59	1	6	5	8	3
60	1	6	7	8	1
61	1	7	4	8	1
62	1	7	5	8	1
63	1	7	6	8	1
64	1	8	4	8	1
65	3	1	1	9	32
66	3	1	2	9	6
67	1	1	3	10	3
68	1	1	4	10	2
69	1	2	2	10	1
70	1	2	3	10	1
71	1	2	4	10	3

SeqNO	R Tag	F Tag	M Tag	Subclass	Count
72	2	1	3	10	1
73	2	2	3	10	3

subclass detailclass mapping

Let 's join the two table. Extract class table and sort by class, subclass, detail class.

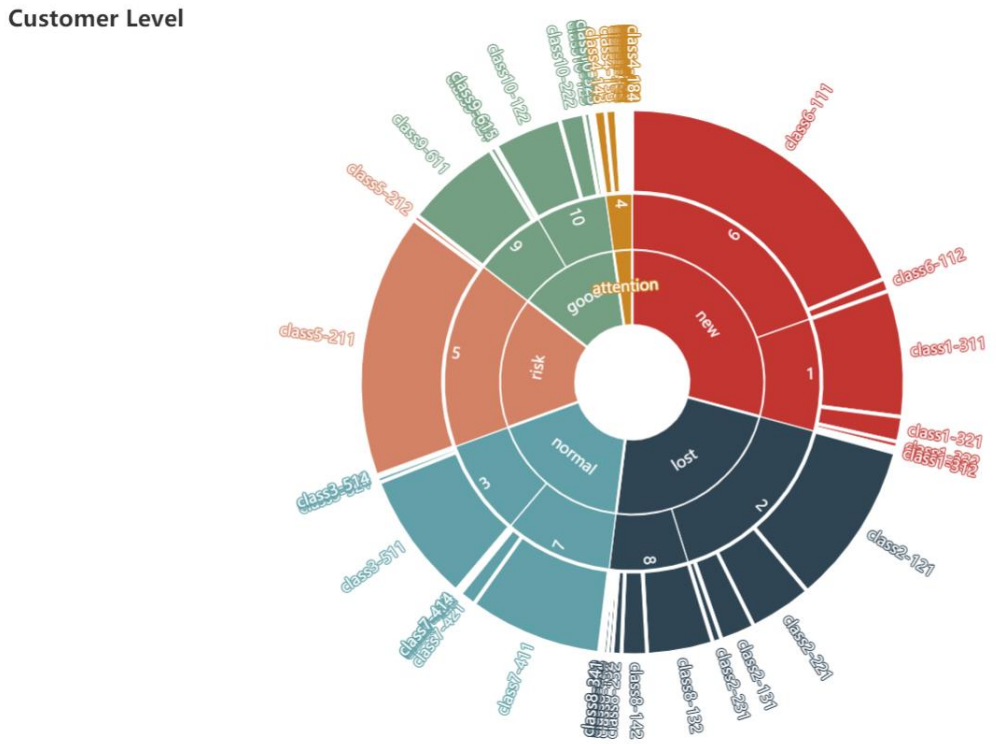
SeqNO	Class	Subclass	Detail Class	Count
1	attention	4	class4-211	681
2	attention	4	class4-212	15
3	good	9	class9-311	324
4	good	9	class9-312	6
5	good	10	class10-113	3
6	good	10	class10-114	2
7	good	10	class10-122	174
8	good	10	class10-123	16
9	good	10	class10-124	3
10	good	10	class10-213	1
11	good	10	class10-223	3
12	lost	2	class2-511	338
13	lost	2	class2-512	3
14	lost	2	class2-514	1
15	lost	2	class2-521	14
16	lost	2	class2-522	2
17	lost	2	class2-611	249
18	lost	2	class2-612	1
19	lost	2	class2-615	1
20	lost	2	class2-621	15
21	lost	2	class2-632	2
22	lost	8	class8-116	1
23	lost	8	class8-125	2

SeqNO	Class	Subclass	Detail Class	Count
24	lost	8	class8-134	6
25	lost	8	class8-135	2
26	lost	8	class8-136	1
27	lost	8	class8-143	30
28	lost	8	class8-144	2
29	lost	8	class8-145	3
30	lost	8	class8-146	1
31	lost	8	class8-147	1
32	lost	8	class8-153	4
33	lost	8	class8-154	1
34	lost	8	class8-155	1
35	lost	8	class8-156	1
36	lost	8	class8-162	1
37	lost	8	class8-163	1
38	lost	8	class8-165	3
39	lost	8	class8-167	1
40	lost	8	class8-174	1
41	lost	8	class8-175	1
42	lost	8	class8-176	1
43	lost	8	class8-184	1
44	new	1	class1-111	816
45	new	1	class1-112	33
46	new	6	class6-221	162
47	new	6	class6-222	63
48	new	6	class6-231	23
49	new	6	class6-321	64
50	new	6	class6-322	16

SeqNO	Class	Subclass	Detail Class	Count
51	new	6	class6-323	1
52	new	6	class6-331	7
53	new	6	class6-332	3
54	new	6	class6-341	1
55	new	6	class6-432	2
56	normal	3	class3-131	89
57	normal	3	class3-132	170
58	normal	3	class3-133	27
59	normal	3	class3-141	11
60	normal	3	class3-142	65
61	normal	3	class3-152	12
62	normal	3	class3-232	24
63	normal	3	class3-233	3
64	normal	3	class3-241	1
65	normal	3	class3-242	5
66	normal	7	class7-121	421
67	risk	5	class5-411	344
68	risk	5	class5-412	2
69	risk	5	class5-413	1
70	risk	5	class5-414	1
71	risk	5	class5-421	43
72	risk	5	class5-422	6
73	risk	5	class5-431	2

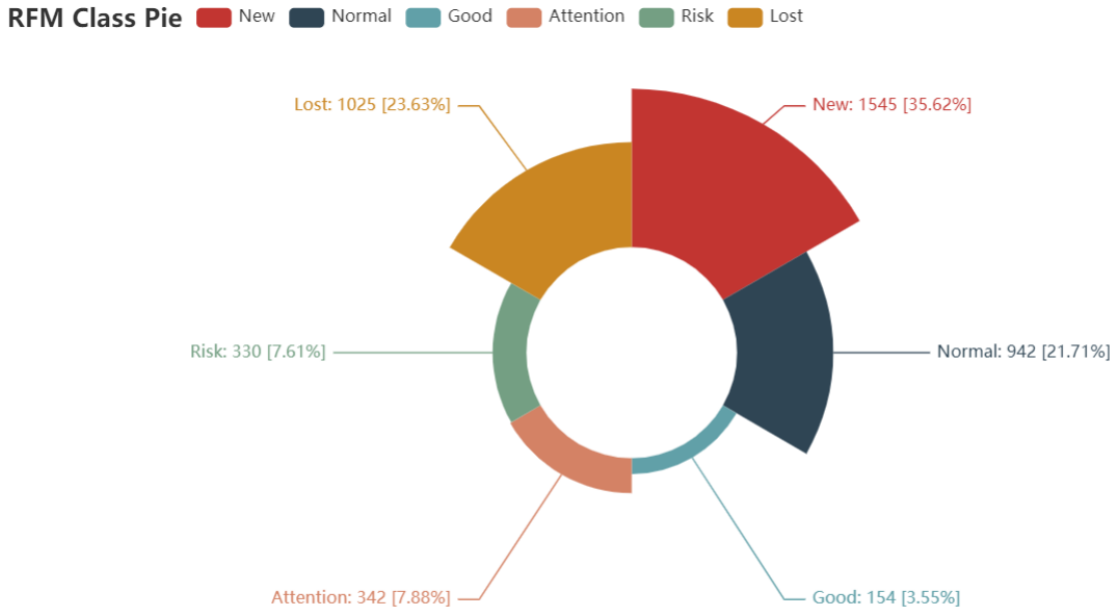
class table

It can be shown as below.



Sunburst chart – customer classes

We can see class proportion.



Pie chart – class proportion

5 Conclusion

At last we have 6 primary class and 10 sub class with 73 detail class.

- Lost customers are 23.63%.
- Risk customers are 7.61%.
- Attention customers are 7.88%.
- Normal customers are 21.71%.
- New customers are 33.62%.
- Good customers are 3.55%.

So, I find out:

- Lack of customer stickiness: total of level risk and attention is more then 15%, need to do preference analysis for them. Some promotion activities should be useful. Level new is 33.62%, member points and gifts may helpful.
- The goods structure is not fit consume preference: level good is much smaller than level normal.
- Level lost is about 1/4, need more promotion activities and advertising to bring them back.

6 References

pyecharts sklearn pandas

7 Appendices

You can download the data from kaggle