UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA ANALYSIS WITH R/PYTHON COURSE

TOPIC: CUSTOMER SEGMENTATION WITH RFM AND K – MEANS CLUSTERING

Lecturer: Nguyen Phat Dat, MA
Group: 08

Ho Chi Minh City, June 8, 2022

MEMBERS OF GROUP

Order	Name	Student ID	Contribution
1	Phạm Thành Đạt	K194111601	20%
2	Huỳnh Nhật Hào	K194111603	20%
3	Trần Đức Duy	K194050694	20%
4	Lương Trường Phước	K194111624	20%
5	Trương Thành Sang	K184060799	20%

ACKNOWLEDGMENTS

In the process of realizing the objective of the report 'Customer segmentation with RFM and K-means', the group received enthusiastic help and support from classmates, lecturers and teaching assistants to complete the final report. This project is completed based on references, learning experiences, books and related reports.

The group would like to express their sincere thanks to Mr. Nguyen Phat Dat and Mr. Tran Le Tan Thinh - is a lecturer and teaching assistant who spent time and effort supporting the group in answering questions and asking questions during the process of making the report. Although we have tried best to complete the report, errors cannot be required. The team wishes to receive suggestions from teachers and teaching assistants to improve the report better.

Group 8

COMMITMENT

The report was prepared and developed by all team members, all guided through relevant lectures by lecturers and tutors. In the process of making the report, the team consulted and cited relevant sources. All of the above is true and has been committed by all team members.

Ho Chi Minh city, June 8 2022

Group 8

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ACRONYMS

Order	Acronyms	Explain
1	GMV	Gross merchandise value
2	R	Recency
3	F	Frequency
4	M	Monetary
5	RFM	Is a method used for analyzing customer value
6	EDA	Exploratory data analysis

CHAPTER 1. PROJECT OVERVIEW AND INTRODUCTION

1.1. Project overview

1.1.1. Reasons

Data mining is a powerful technique to help companies exploit behavior and trends in their customer data. Then, to promote customer relationships, it is one of the well known tools provided for Customer Relationship Management (CRM). However, there are some limitations for data mining tools, such as neural networks with long training times and genetic algorithms which are computationally complex. This research combines the quantitative value of RFM attributes and the K-means algorithm, to indicate the rules and meaning of the data file so that solutions can be given in the most effective way.

1.1.2. Objectives

The purposes related to this study are as follows: (1) separate the continuous attributes to improve the rough set algorithm; (2) grouping customer value as output (customer loyalty) divided into 3 classes, 5 and 7 based on subjective point of view, then see which class is the best in terms of accuracy rate; (3) find out customer characteristics to enhance CRM and have the best marketing strategy.

1.1.3. Objects and scopes

Objects: The study was conducted through an empirical method on a dataset with 52761 transactions of service group stores, clustering 5 customer segments with the characteristics of each cluster being tested for quality demonstrating the effectiveness and applicability of the study.

Scopes:

Time scopes: 1/1/2021- 31/3/2022

Space scopes: Transactions of service company.

1.2. Research methods and procedures

Research Methods:

Step 1: From the input data, the dataset is surveyed and Dataprocessing. Calculate the Recency, Frequency, Monetary values and finally complete the RFM data model.

Step 2: From the data exploration in step 1, problems and characteristics related to the values in the RFM model are also discovered. Therefore, when clustering by K-means method, accuracy will be guaranteed. Choose the right model and analyze customer groups, make a decision to choose customer groups based on the analysis results from the hybrid method.

Step 3: Comparison of customer segments by RFM and K-means methods. Then find the more optimal method.

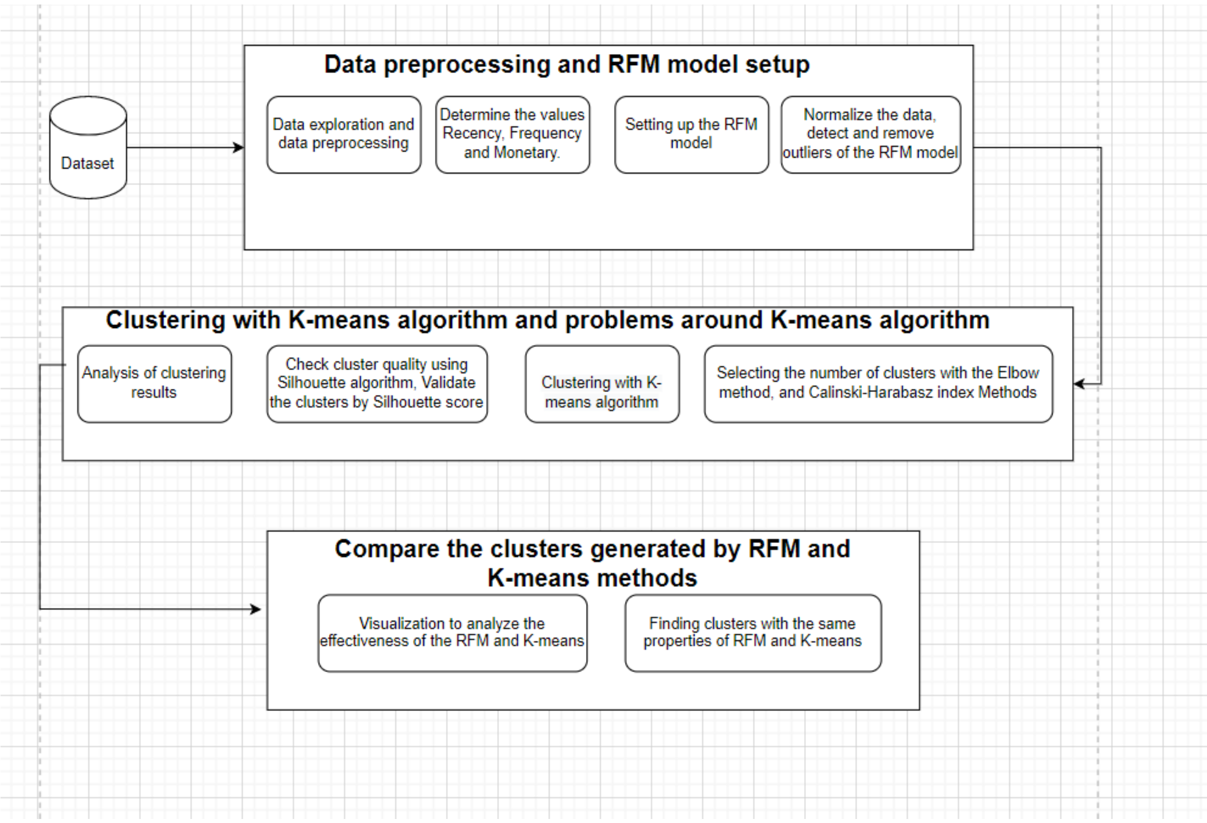


Figure 1 Research methods, procedures and experiments

1.3. Introduction

Due to the complexity and diversity of business activities, information about customers is essential and important for competitive advantage and is often of interest. Especially, the development of information technology in today's competitive and rapidly changing environment promotes transaction and payment activities of customers. Based on this relationship, information plays a central role in the opportunities and challenges in the day-to-day business operations of companies. That is useful information to help companies enhance their competitiveness. So, how to enhance the competitive strength in the market for the company as well as understand customers according to different segments?

Meeting and understanding customer requirements is one of the important factors. So, a great CRM will easily meet the needs of customers as well as enhance the power with customers for the company.

Therefore, the effective use of IT to support the CRM process is the shortest path to a successful CRM. Although understanding customer situations is somewhat varied, all companies provide products and services to customers to meet customer needs.

In recent years, data mining has not only become popular in search but also in commercialization. Data mining can help organizations uncover meaningful trends, patterns, and correlations in customers, their products, or their data, to drive improved customer relationships and subsequently reduce business risk.

From data analysis, more objective and multidimensional decisions of managers as well as accurate customer segment analysis contribute to the success of the company's CRM.

To solve the above problem, we rely on attribute RFM (recent hits-frequency-currency) and K-means method to group customer values. In general, the aim of this study is to create classification rules for achieving an excellent CRM for the company and the customer.

Figure 2 Build Your Customer Personal

- When we segment standard customers based on data, there are many benefits such as:
- Better match the product or service to the needs of the target customer segment. Create appropriate creative marketing communication ideas and messages.
- Improve the product or service offered to the target customer. Businesses can use market segmentation to make decisions to orient appropriate business and marketing activities to avoid wasting resources and costs when targeting incorrectly.
- Allows businesses to retain more customers by offering better products to existing customers.
- Enable your business to grow by selling to customers who have already purchased.
- Enhance business profitability by targeting customers who tend to have more disposable income by increasing order value.

1.4. Literature review

Customer value analysis is a type of analytical method to discover customer characteristics and deeper analysis of specific customers to abstract useful knowledge from big data.

Kaymart (2001) showed that the RFM model is one of the targeted methods of identifying well-known customer profiles with customer value. Its advantage is to extract features of customers using fewer criteria (three dimensions) as clusters to reduce customer and customer analytics model complexity.

Schijns and Schroder (1996) from the point of view of consumer behavior as well, RFM Model is a long familiar method to measure the strength of customer relationship.

Joseph Pine, Peppers & Rogers (2009) maintenance costs are much less expensive than acquisition costs. The importance of developing a good outcome with existing and new customer relationships. Instead of attracting new customers, they want to do the best they can, do more marketing to customers to retain customers and build lasting customer relationships.

He and Li (2016) suggested a three-dimensional approach to improving the customer lifetime (CLV), the satisfaction of the customer and customer behavior. The authors have concluded that the consumers are different from one another and so are their needs. Segmentation assists in finding their demand and expectations and providing a good service.

Jiang and Tuzhilin (2009) presents a direct clustering approach that clusters the customers not based on computed statistics, but by combining transactional data of several customers. The authors also showed that it is NP-hard to find an optimal segmentation solution. So, Tuzhilin came up with different sub-optimal clustering methods. The authors then experimentally examined the customer segments obtained by direct grouping, and it is observed to be better than the statistical approach.

Therefore, businesses intend through the use of RFM analytics to mine the database to know about the customers who spend the most money and create the greatest value for the business.

CHAPTER 2. APPROACH

2.1. RFM analysis

Based on a customer's past transactional behavior. In order to divide customers and find out the characteristics of customer groups, help the company have effective customer management and approach strategies. RFM includes 3 main indicators:

Recency (R): Last transaction time.

Frequency (F): Frequency of customer purchases.

Monetary value (M): Total amount spent by the customer.

After the RFM method is established, the above indicators of each customer will be ranked in hierarchical order, with a scale usually divided from 1 to 5.

Benefits of RFM analysis:

Increase customer retention rate.

Speed up response from customers.

Increase revenue from customers.

In our report, we divide the RFM model into 5 customer segments: Stars, Loyal, Potential loyal, Hold and improve, and Risky. This makes the assessment easier and more convenient. From there, it helps businesses or decision makers to have a better overview of each of their customers.

2.2. K-means Clustering

K-means is a simple clustering algorithm of unsupervised learning (unlabeled data) and is used to solve clustering problems. The idea of the k-means clustering algorithm is to divide a dataset into different clusters. where the number of clusters is given k. Clustering work is established based on the principle: Data points in the same cluster must have the same certain properties. That is, between points in the same cluster must be related to each other.

The k-means clustering algorithm is a method used in analyzing the cluster properties of data. It is especially used in data mining and statistics. It partitions the data into k different clusters. This algorithm helps us to determine which group our data really belongs to. The purpose is how to divide the data into different clusters so that the data in the same cluster has properties. same substance.

The idea of the k-means algorithm:

Initialize K data points in the dataset and temporarily treat it as the center of our data clusters.

For each data point in the dataset, its cluster center will be identified as 1 of the K nearest cluster centers.

After all the data points have their centers, recalculate the position of the cluster centers to ensure that the center of the cluster is in the center of the cluster.

Steps 2 and 3 will be repeated until the position of the cluster center does not change or the center of all data points does not change.

2.3. K-mean ++

The K-means algorithm has the disadvantage that it is sensitive to initializing centroids or mean points. Meaning, more than one cluster can end up with a single centroid. Similarly, more than one hub can be initialized to the same cluster leading to poor clustering.

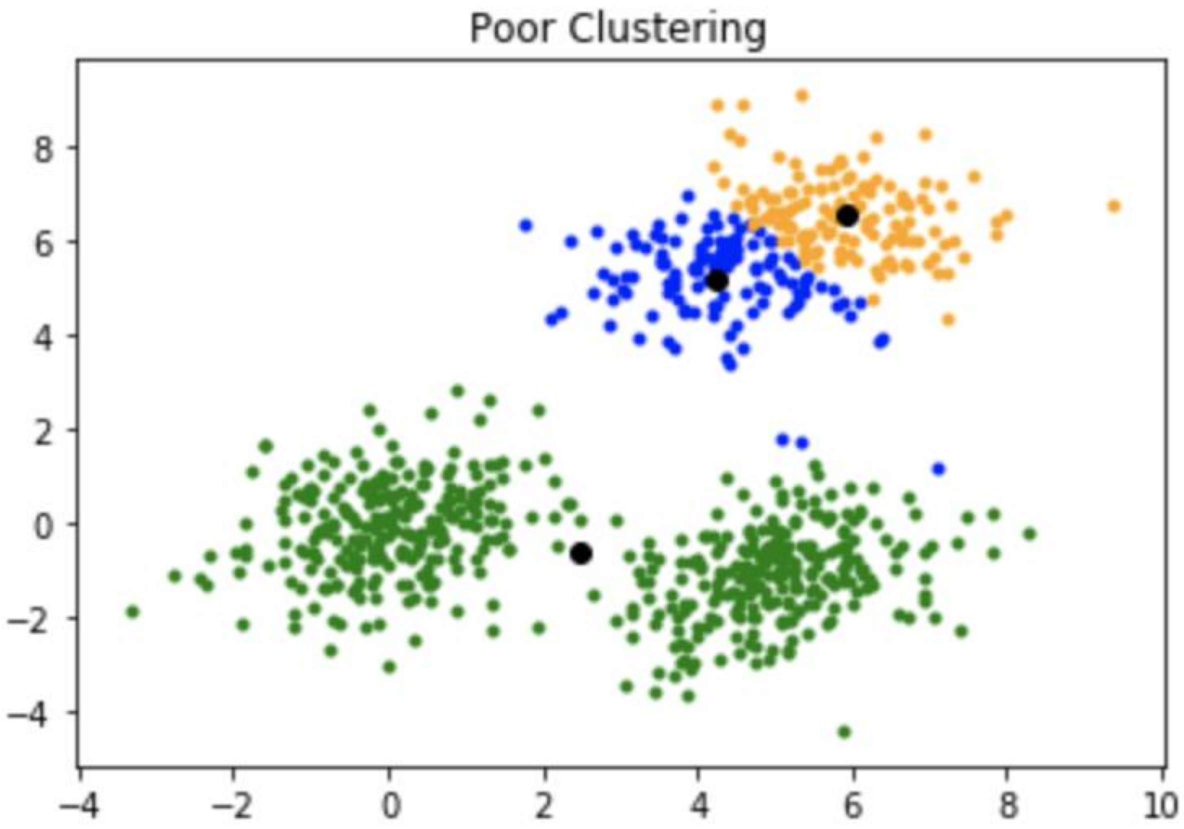


Figure 3 Poor clustering

To overcome the above disadvantage, we use K-mean ++. This algorithm ensures smarter centroid initialization and improves the quality of clustering. Apart from the initialization, the rest of the algorithm is the same as the standard

K-means algorithm. In other words, K-means++ is the standard K-means algorithm combined with smarter centroid initialization.

The idea of the k-means algorithm:

Randomly select the first centroid from the data points.

For each data point, calculate its distance from the nearest, previously selected centers.

Select the next centroid from the data points such that the probability of choosing a point as the centroid is proportional to its distance from the nearest, previously selected centroid.

Repeat steps 2 and 3 until k centroids are sampled

2.4. Elbow method

To determine the number of clusters in the K-Means algorithm. With the question that is the best number of clusters divided by each set of data. The Elbow method is a way to help us choose the appropriate number of clusters based on the visualized graph by looking at the decline of the deformed function and selecting the elbow point.

The point where the elbow is that the rate of decline in the deformation function will change the most. That is, since this position, the number of clusters also does not help the function deformed significantly. If the algorithm is divided by the number of clusters at this position, the cluster property will be achieved in the most general manner without having the taste phenomenon.

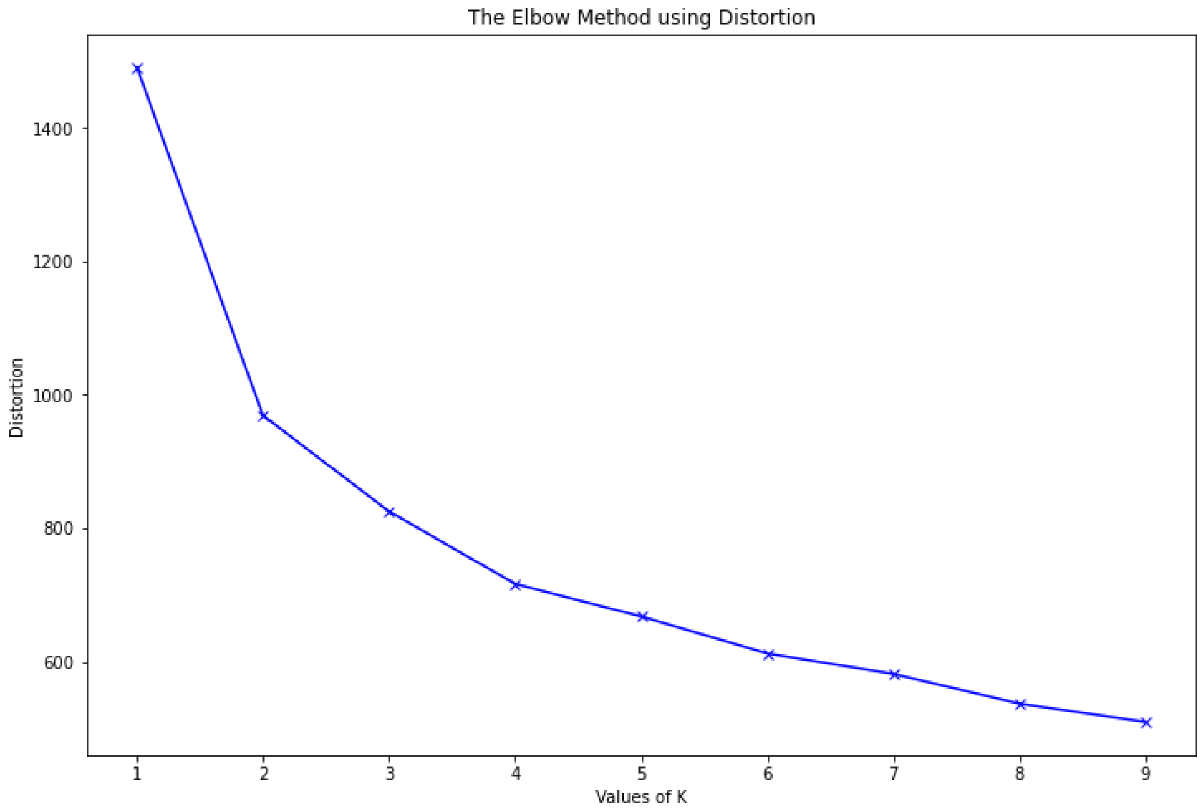


Figure 4 Use Elbow to select K

However, there are some cases that we will not easily detect the location of Elbow, especially for the data sets that the rules of clusters are not really easily detected. But in general, the Elbow method is still the best method used in finding the number of clusters to be divided.

2.5. Calinski-Harabasz Index

The CH Index (also known as Variance ratio criterion) is a measure of how similar an object is to its own cluster compared to other clusters. Here cohesion is estimated based on the distances from the data points in a cluster to its cluster centroid and separation is based on the distance of the cluster centroids to the global centroid.

A high CH means better clustering because the observations in each cluster are closer together (more dense), while the clusters themselves are further apart (well separated).

To better understand the formula of CH, we will go through the calculations step by step.

Step 1: Calculate inter-cluster dispersion

The first step is to calculate the inter-cluster dispersion or the between group sum of squares (BGSS).

The inter-cluster dispersion in CH measures the weighted sum of squared distances between the centroids of a cluster and the centroid of the whole dataset.

The between group sum of squares is calculated as:

$$BGSS = \sum_{k=1}^{K} n_k \times \|C_k - C\|^2$$

where:

nk : the number of observations in cluster k

Ck: the centroid of cluster k

C : the centroid of the dataset (barycenter)

K : the number of clusters

Step 2: Calculate intra-cluster dispersion

The second step is to calculate the intra-cluster dispersion or the within group sum of squares (WGSS).

The intra-cluster dispersion in CH measures the sum of squared distances between each observation and the centroid of the same cluster.

For each cluster k we will compute the WGSSk as:

$$WGSS = \sum_{i=1}^{n_k} \|X_{ik} - C_k\|^2$$

where:

nk: the number of observations in cluster k

Xik: the i-th observation of cluster k

Ck: the centroid of cluster k

And then sum all individual within group sums of squares:

$$WGSS = \sum_{k=1}^{K} WGSS_k$$

where:

WGSSk : the within group sum of squares of cluster k

K : the number of clusters

Step 3: Calculate Calinski-Harabasz Index

The Calinski-Harabasz index is defined as the sum of inter-cluster dispersion and the sum of intra-cluster dispersion for all clusters.

The Calinski-Harabasz index is calculated as:

$$CH = \frac{\frac{BGSS}{K-1}}{\frac{WGSS}{K-1}} = \frac{BGSS}{WGSS} \times \frac{N-K}{K-1}$$

where:

BGSS : between-group sum of squares (between-group dispersion)

WGSS : within-group sum of squares (within-group dispersion)

N : total number of observations

K : total number of clusters

Advantages

The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster.

The score is fast to compute.

Disadvantage

The Calinski-Harabasz index is generally higher for convex clusters than other concepts of clusters

2.6. Silhouette Index

2.6.1. Silhouette

Silhouette measures the distance of a data point in the cluster to the center point of the cluster, and the distance of that point itself to the center point of the nearest cluster. Helps us to know which data points fit inside the cluster (good) or far from the center of the cluster (not good) to evaluate more effectively in the clustering process.

Formula for Silhouette

Silhouette = (p-q)max(p,q)

Where

p: is the average distance from that point to points in the nearest cluster of which the data point is not part.

q: is the average distance from that point to all points in its own cluster.

2.6.2. Silhouette point

The score has a range of [-1, 1] as follows ;

Score +1 : Score close to +1 indicates that the sample is far from the neighboring cluster.

Score 0: A score of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters.

Score -1: A negative score indicates that the samples were assigned to the wrong clusters.

We calculate scores in Silhouette to evaluate the overall results of our analysis. The higher the mean, the better, and vice versa but only between -1 and 1, with 1 being the best result.

CHAPTER 3. EDA

3.1. Loading Packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import scipy as stats
from  yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
import squarify
```

Figure 5 Code loading packages

These are the libraries that we will use in this project. Most of the libraries are built-in when you use Anaconda. Yellowbrick and Squarify packages must be installed additionally.

3.2. Reading the data

```
path = 'D:\Ky6_2022\PHANTICHDULIEU\Final_project\DataTransaction_Jan2021_to_Mar2022.xlsx'
df = pd.read_excel(path)
✓ 6.6s
```

Figure 6 Code import data

The transaction data set from January 2021 to March 2022 is read in xlsx format into a data frame.

3.2.1. Checking the data

```
df.shape
✓ 0.4s
(52760, 7)
```

Figure 7 Code check size dataframe

The dataset consists of 52,760 rows and 7 columns.

```
df.dtypes
✓ 0.4s

DATE                      datetime64[ns]
Order_id                           int64
NEWVERTICAL_Merchant              object
MerchantID                         int64
User_id                            int64
GMV                                int64
Service Group                     object
dtype: object
```

Figure 8 Code check data type

The data types of the columns in the data set are formatted in accordance with the column names and are convenient in the processing.

```
df.head()
✓ 0.7s
```

	DATE	Order_id	NEWVERTICAL_Merchant	MerchantID	User_id	GMV	Service Group
0	2021-01-01	8733622706	Marketplace	37	61386143	100000	marketplace
1	2021-01-01	8726857991	Supermarket	9	48453125	5000	supermarket
2	2021-01-01	8737326894	Supermarket	9	49921027	106600	supermarket
3	2021-01-01	8732579078	supermarket	9	46022523	270000	supermarket
4	2021-01-01	8725567343	CVS	8	44014594	68000	cvs

Figure 9 Code show data

These are the first 5 rows of the dataset

3.2.2. Finding unique values

```
df[['DATE','Order_id','User_id','GMV','Service Group','NEWVERTICAL_Merchant']].nunique()
✓ 0.6s

DATE                      454
Order_id                49175
User_id                  6479
GMV                      9698
Service Group               6
NEWVERTICAL_Merchant       12
dtype: int64
```

Figure 10 Code check unique

3.3. Data Preprocessing

3.3.1. Null Value Check

```
df.isnull().sum()
✓ 0.4s

DATE                    0
Order_id                0
NEWVERTICAL_Merchant    0
MerchantID              0
User_id                 0
GMV                     0
Service Group           0
dtype: int64
```

Figure 11 Code check null values

No null values in all columns.

- Check the data set's last transaction date:

```
df.DATE.max()
✓ 0.4s

Timestamp('2022-03-31 00:00:00')
```

Figure 12 Code check date latest

3.3.2. Correlation Check

Figure 13 Heatmap show correlation of all variables

The variables are not correlated with each other.

3.3.3. Dropping negative values

```
(df['GMV']<0).sum()
✓ 0.5s

0
```

Figure 14 Code negative GMV

There is no negative value of GMV variable

3.3.4. *Removing duplicates*

```
df.duplicated().sum()
✓ 0.1s

3585

df.drop_duplicates(inplace=True)
✓ 0.6s

df.shape
✓ 0.3s

(49175, 7)
```

Figure 15 Code remove duplicates

There are 3,585 duplicate rows. We delete them, reducing the size of the data to 49,175 rows, 7 columns.

3.3.5. *Select features*

```
df = df[['DATE','Order_id','User_id','GMV']]
✓ 0.4s
```

Figure 16 Code select features

We selected relevant, useful variables for this analysis project.

3.4. Describe

```
df.describe(datetime_is_numeric=True)
```
✓ 0.6s

	DATE	Order_id	User_id	GMV
count	49175	4.917500e+04	4.917500e+04	4.917500e+04
mean	2021-11-15 20:10:48.036603648	1.823368e+10	4.114886e+07	1.435014e+05
min	2021-01-01 00:00:00	8.725567e+09	1.081010e+05	1.000000e+03
25%	2021-09-11 00:00:00	1.647234e+10	3.628736e+07	1.500000e+04
50%	2021-12-07 00:00:00	1.894498e+10	4.407502e+07	4.000000e+04
75%	2022-01-31 00:00:00	2.038546e+10	5.068920e+07	1.234000e+05
max	2022-03-31 00:00:00	2.255931e+10	6.149055e+07	2.000000e+07
std	NaN	3.017890e+09	1.395196e+07	3.806613e+05

Figure 17 Code describe data

The average value of each invoice is VND 143,501. There is a significant difference between the cheapest invoice (VND 1,000) and the most expensive bill (20,000,000 VND). The GMV variable can have many exceptions.

- Sample variance:

```
print(df.var())
```
✓ 0.8s

```
Order_id      9.107663e+18
User_id       1.946571e+14
GMV           1.449031e+11
dtype: float64
```

Figure 18 Code and result sample variance

Variables with large sample variance. The values of the variables tend to be strongly and discretely dispersed.

- Skewness

```
df.skew()
✓ 0.9s
```

```
Order_id     -0.831500
User_id      -1.175935
GMV          10.915268
dtype: float64
```

Figure 19 Code and result skew coefficient

GMV has a positive skewed distribution. The other two variables have a negatively skewed distribution.

- Kurtosis

```
df.kurtosis()
✓ 0.5s
```

```
Order_id       0.200620
User_id        0.947497
GMV          255.202516
dtype: float64
```

Figure 20 Code and result kurtosis coefficient

The GMV variable has a leptokurtic distribution, while the other two variables have a platykurtic distribution.

3.5. Plots

3.5.1. Box plots

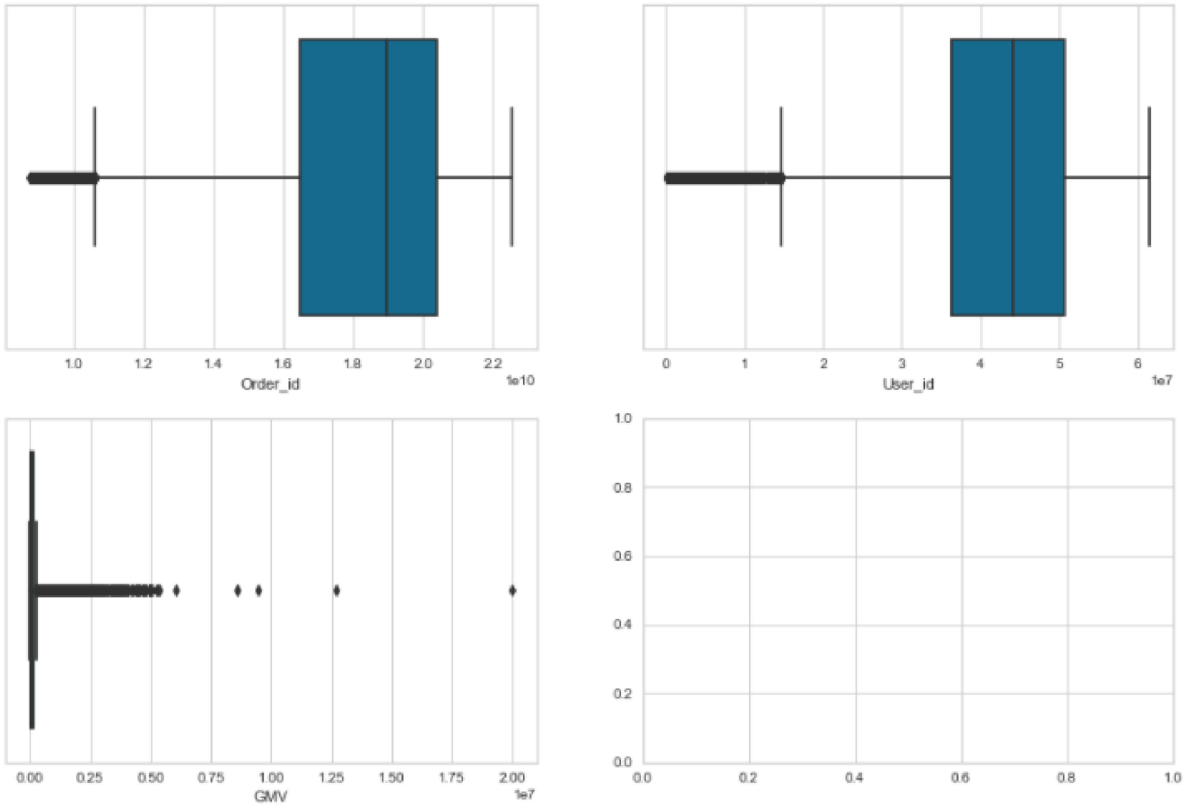


Figure 21 Box plots of all variables

All three variables have many exceptions, especially the GMV variable. Order_id and User_id have a left skewed distribution, GMV has a right skewed distribution.

3.5.2. Scatter Plots

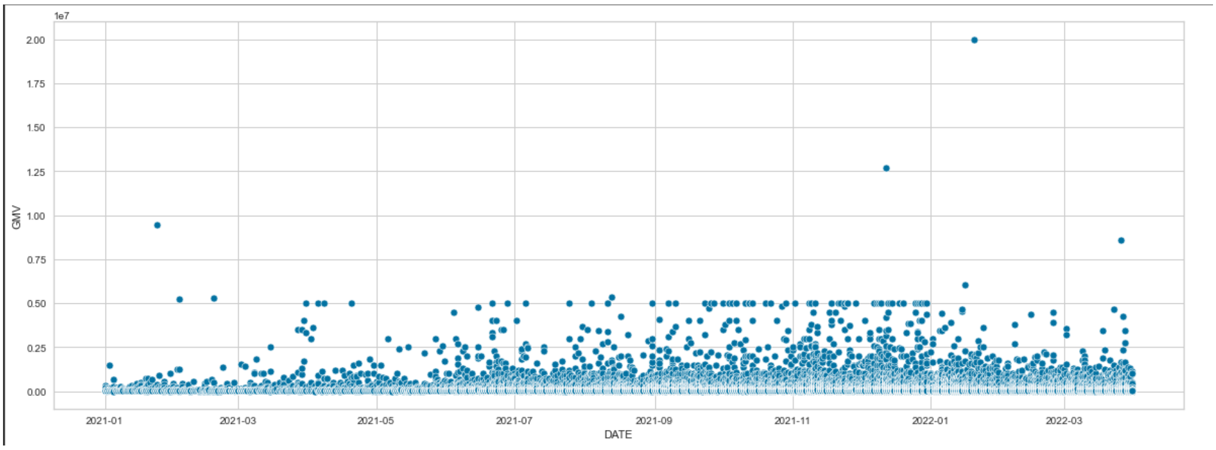


Figure 22 Scatter plots between DATE and GMV variables

Most of the order values are under VND 5,000,000. They tend to stay the same as the DATE variable increases.

3.5.3. *Histogram*

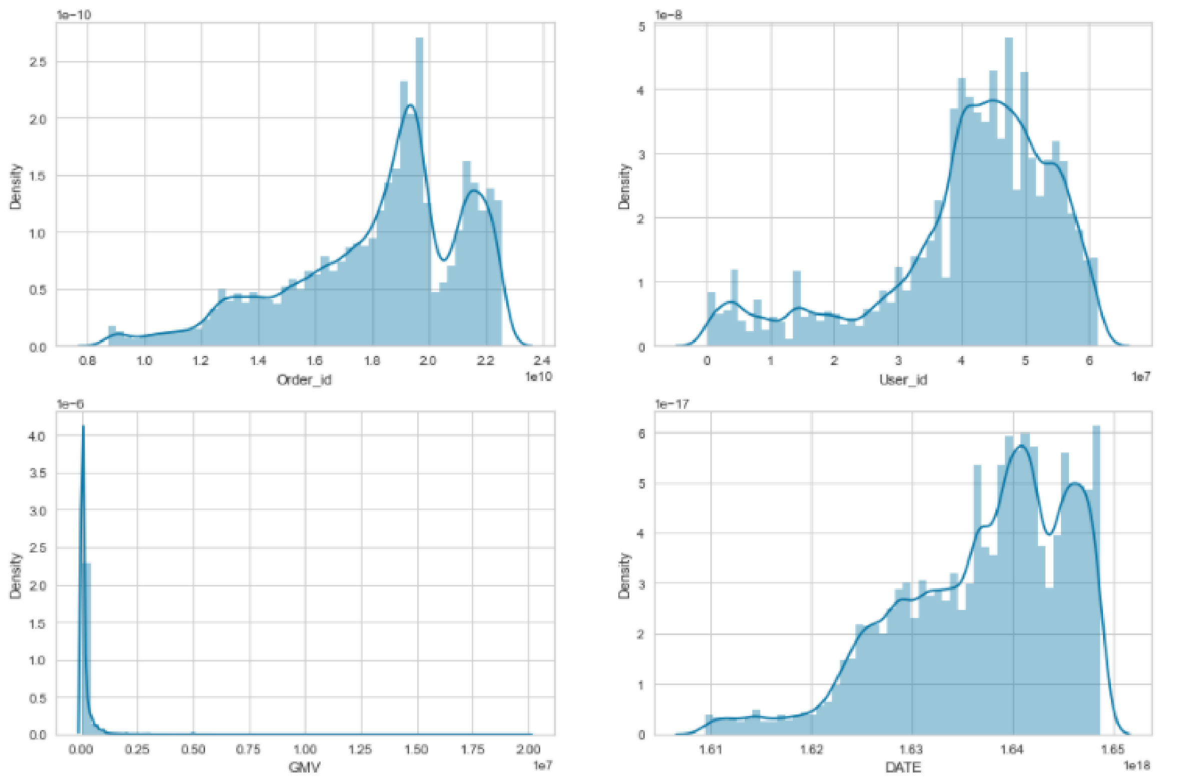


Figure 23 Histogram of Order_id, User_id , GMV, DATE variables

All variables have no symmetric distribution. The GMV variable has the highest frequency at values below 1,000,000.

CHAPTER 4. DATA ANALYSIS

4.1. Create table RFM

- Create Pivot table with column “RecentOrderDate”

```
pin_date = dt.datetime(2022, 4,30)
```
✓ 0.4s

```
df1 = pd.pivot_table(data = df,
                     index = ['User_id'],
                     values = ['DATE'],
                     aggfunc = {'DATE':max}
                     )

df1.columns = ['RecentOrderDate']
```
✓ 0.6s

Figure 24 Code create pivot table

We set the timeline for this project on April 34, 2022, right after the data collection time.

- Create a column “Recency”

```
customer = pd.DataFrame({'User_id': df['User_id'].unique()})
df2 = pd.merge(customer, df1.reset_index(), on = ['User_id'])
df2['Recency'] = df2['RecentOrderDate'].apply(lambda x: (pin_date - x).days)
```
✓ 0.3s

Figure 25 Code create Recency variable

```
dfFrequency = df.groupby('User_id').Order_id.nunique().to_frame()
dfFrequency.columns = ['Frequency']
```
✓ 0.1s

```
df2 = pd.merge(df2, dfFrequency.reset_index(), on = 'User_id')
```
✓ 0.8s

Figure 26 Code create Frequency variable

- Create a column “Monetary”

```
dfMonetary = df.groupby('User_id').GMV.sum().to_frame()
dfMonetary.columns = ['Monetary']
df2 = pd.merge(df2, dfMonetary.reset_index(), on = 'User_id')
```
✓ 0.5s

Figure 27 Code create Monetary variable

- Dataframe result:

```
df2 = df2[['User_id','Recency','Frequency','Monetary']]
df2.head()
```
✓ 0.4s

	User_id	Recency	Frequency	Monetary
0	61386143	66	6	339100
1	48453125	30	27	3180097
2	49921027	336	2	119600
3	46022523	48	63	10805283
4	44014594	438	4	150000

Figure 28 Code show set up data for RFM

We removed the RecentOrderDate variable, keeping the variables necessary for the analysis.

4.2. Standardization

- Code standardize data by Z-score method:

```
df2['Re_zs'] = stats.zscore(df2['Recency'])
df2['Fe_zs'] = stats.zscore(df2['Frequency'])
df2['Mo_zs'] = stats.zscore(df2['Monetary'])
df2.set_index('User_id', inplace=True)
```

Figure 29 Code standardize data

Our variables have significantly different ranges. Recency has a range of hundreds, Frequency has a range of tens, and Monetary has a range of hundreds of thousands. So we standardize the data, to make it easier to compare their impact. This will facilitate our further analysis.

- Result after standardization:

User_id	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs
61386143	66	6	339100	-0.713776	-0.090166	-0.154905
48453125	30	27	3180097	-1.048483	1.100775	0.431826
49921027	336	2	119600	1.796531	-0.317012	-0.200237
46022523	48	63	10805283	-0.881129	3.142389	2.006602
44014594	438	4	150000	2.744869	-0.203589	-0.193959

Figure 30 Data after standardization

After data normalization, the variables will have values mostly between -3 and 3.

4.3. Remove outliers

Outliers will affect customer segmentation using RFM. In addition, the K-means method is also very sensitive to them. Therefore, we use the Z-score method to eliminate outliers and create data set homogeneity when using two methods RFM and K-means.

- Outlier detection:

```
#Remove outlier
outlier=df2[((df2[['Re_zs','Fe_zs','Mo_zs']] < -3) | (df2[['Re_zs','Fe_zs','Mo_zs']] >3)).any(axis=1)]
outlier.head()
```

✓ 0.1s

User_id	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs
46022523	48	63	10805283	-0.881129	3.142389	2.006602
59720332	484	1	10000	3.172551	-0.373723	-0.222872
11368352	468	2	8000	3.023792	-0.317012	-0.223285
7367023	473	3	637000	3.070279	-0.260300	-0.093382
61488523	480	4	280000	3.135361	-0.203589	-0.167111

Figure 31 Code detecting outliers

Values greater than 3 and less than -3 are considered outliers and will be discarded.

- Remove outlier:

```
df2 = df2[((df2[['Re_zs','Fe_zs','Mo_zs']] >= -3) & (df2[['Re_zs','Fe_zs','Mo_zs']] <= 3)).all(axis=1)]
df2.head()
```

✓ 0.1s

User_id	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs
61386143	66	6	339100	-0.713776	-0.090166	-0.154905
48453125	30	27	3180097	-1.048483	1.100775	0.431826
49921027	336	2	119600	1.796531	-0.317012	-0.200237
44014594	438	4	150000	2.744869	-0.203589	-0.193959
31058664	96	9	1229224	-0.434853	0.079969	0.028926

Figure 32 Code removing outliers

- Shape of dataframe after removing outliers:

```
df2.shape
✓ 0.4s

(6317, 6)
```

Figure 33 Code show size of data after removing outliers

After removing outliers, the size of the data set remains 6317 rows, 6 columns.

4.4. Customer segmentation using RFM

4.4.1. Calculate score, rank

- We need to assign a score from 1 to 5 to recency, frequency and monetary value individually for each customer:

```
df2["Recency_score"] = pd.qcut(df2['Recency'], 5, labels=[5,4,3, 2, 1])
df2["Frequency_score"] = pd.qcut(df2['Frequency'].rank(method="first"), 5, labels=[1, 2, 3,4,5])
df2["Monetary_score"] = pd.qcut(df2['Monetary'], 5, labels=[1, 2, 3,4,5])
df2['rfm'] = df2['Recency_score'].astype(str) + df2['Frequency_score'].astype(str) + df2['Monetary_score'].astype(str)
df2['Score']=(df2['Recency_score'].astype(int)+df2['Frequency_score'].astype(int)+df2['Monetary_score'].astype(int))

✓ 0.1s
```

Figure 34 Code calculating values of RFM

- We convert columns into rfm scores between 1 to 5.
- '5' being the highest and '1' being the least.
- The higher the monetary value, the higher the score is 5.
- Smaller value of recency indicates recent purchases, so it takes the higher value of 5.
- Frequency is the same as monetary, higher the frequency, higher the score.
- ‘rfm’ variable includes the ratings of Recency, Frequency, Monetary combined.
- ‘Score’ is the sum of the ratings of Recency, Frequency, Monetary.
- Result:

	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs	Recency_score	Frequency_score	Monetary_score	rfm	Score
User_id											
61386143	66	6	339100	-0.713776	-0.090166	-0.154905	4	4	4	444	12
48453125	30	27	3180097	-1.048483	1.100775	0.431826	5	5	5	555	15
49921027	336	2	119600	1.796531	-0.317012	-0.200237	1	2	3	123	6
44014594	438	4	150000	2.744869	-0.203589	-0.193959	1	3	3	133	7
31058664	96	9	1229224	-0.434853	0.079969	0.028926	3	4	5	345	12

Figure 35 Table RFM

4.4.2. Calculate level:

```
df2['Score'].plot.hist(bins=12)
plt.show()
```

✓ 0.4s

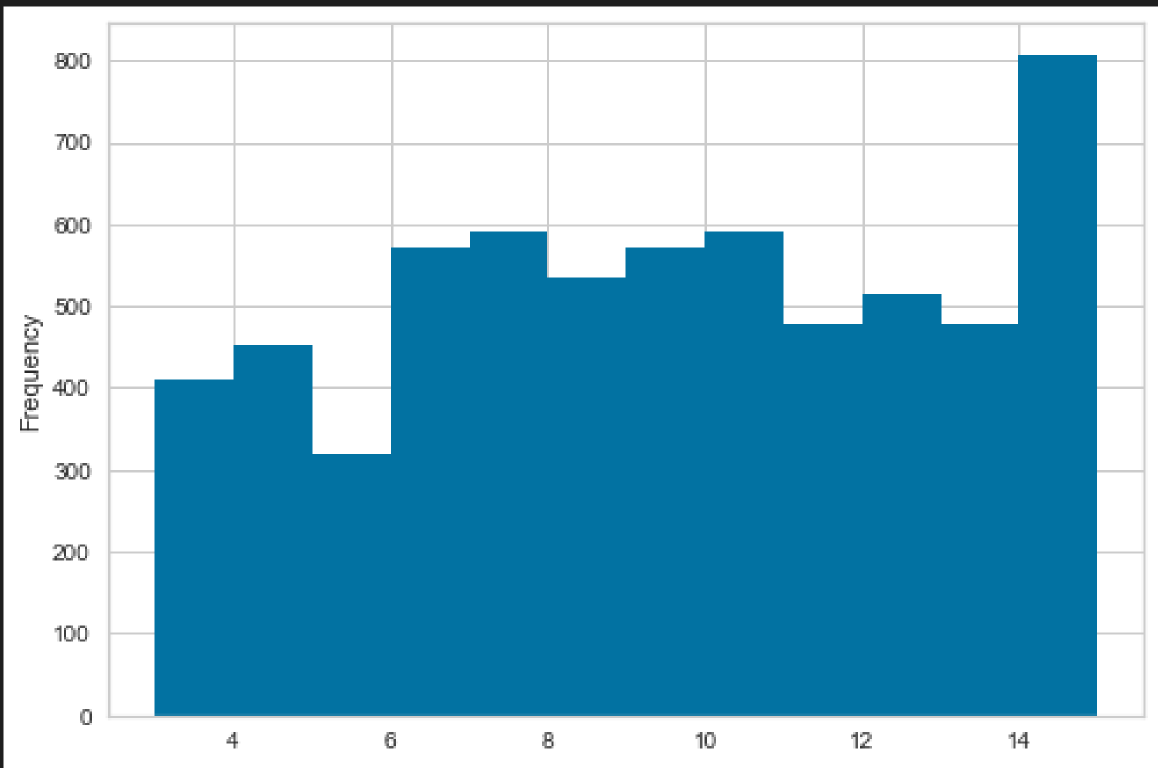


Figure 36 Histogram of Score variable

```python
def rfm_level(score):
    if score <= 6 :
        return 'Bronze'
    elif ((score >6) and (score <= 10)):
        return 'Sliver'
    else:
        return 'Gold'
```
✓ 0.5s

```python
df2['level'] = df2['Score'].apply(lambda score : rfm_level(score))
df2.head()
```
✓ 0.1s

Figure 37 Code calculate level

From the Histogram depicting the distribution of the Score variable. We divide Score into 3 levels. Customers with scores less than or equal to 6 are labeled Bronze, from 6 to less than or equal to 10 are labeled Silver, and 10 or more are labeled Gold.

Here are the results:

User_id	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs	Recency_score	Frequency_score	Monetary_score	rfm	Score	level
61386143	66	6	339100	-0.713776	-0.090166	-0.154905	4	4	4	444	12	Gold
48453125	30	27	3180097	-1.048483	1.100775	0.431826	5	5	5	555	15	Gold
49921027	336	2	119600	1.796531	-0.317012	-0.200237	1	2	3	123	6	Bronze
44014594	438	4	150000	2.744869	-0.203589	-0.193959	1	3	3	133	7	Sliver
31058664	96	9	1229224	-0.434853	0.079969	0.028926	3	4	5	345	12	Gold

Figure 38 Table RFM with level variable

4.4.3. Divide RFM customer segmentation:

Convert column 'rfm' to int type:

```python
df2.rfm = df2.rfm.astype(int)
```
✓ 0.7s

Figure 39 Code convert column 'rfm' to int type

The transactional data we are analyzing is retail industry data. So we decided to divide customers into 5 segments as Stars, Loyal, Potential loyal, Hold and improve, and Risky according to a recent PhD thesis on RFM by Umit Uysal

(2019). Accordingly, the segments don't overlap or leave any gaps, so are more practical to implement.

```
def label_rfm_segments(rfm):

    if (rfm >= 111) & (rfm <= 155):
        return 'Risky'

    elif (rfm >= 211) & (rfm <= 255):
        return 'Hold and improve'

    elif (rfm >= 311) & (rfm <= 353):
        return 'Potential loyal'

    elif ((rfm >= 354) & (rfm <= 454)) or ((rfm >= 511) & (rfm <= 535)) or (rfm == 541):
        return 'Loyal'

    elif (rfm == 455) or (rfm >= 542) & (rfm <= 555):
        return 'Star'

    else:
        return 'Other'
✓ 0.5s
```

Figure 40 Code function calculate customer segment

User_id	Recency	Frequency	Monetary	Re_zs	Fe_zs	Mo_zs	Recency_score	Frequency_score	Monetary_score	rfm	Score	level	rfm_segment_name
61386143	66	6	339100	-0.713776	-0.090166	-0.154905	4	4	4	444	12	Gold	Loyal
48453125	30	27	3180097	-1.048483	1.100775	0.431826	5	5	5	555	15	Gold	Star
49921027	336	2	119600	1.796531	-0.317012	-0.200237	1	2	3	123	6	Bronze	Risky
44014594	438	4	150000	2.744869	-0.203589	-0.193959	1	3	3	133	7	Sliver	Risky
31058664	96	9	1229224	-0.434853	0.079969	0.028926	3	4	5	345	12	Gold	Potential loyal

Figure 41 Table data with RFM segment

4.4.4. Visualizing and analyzing each segment of RFM

Visualizing against each of the factors:

- Scatterplot:

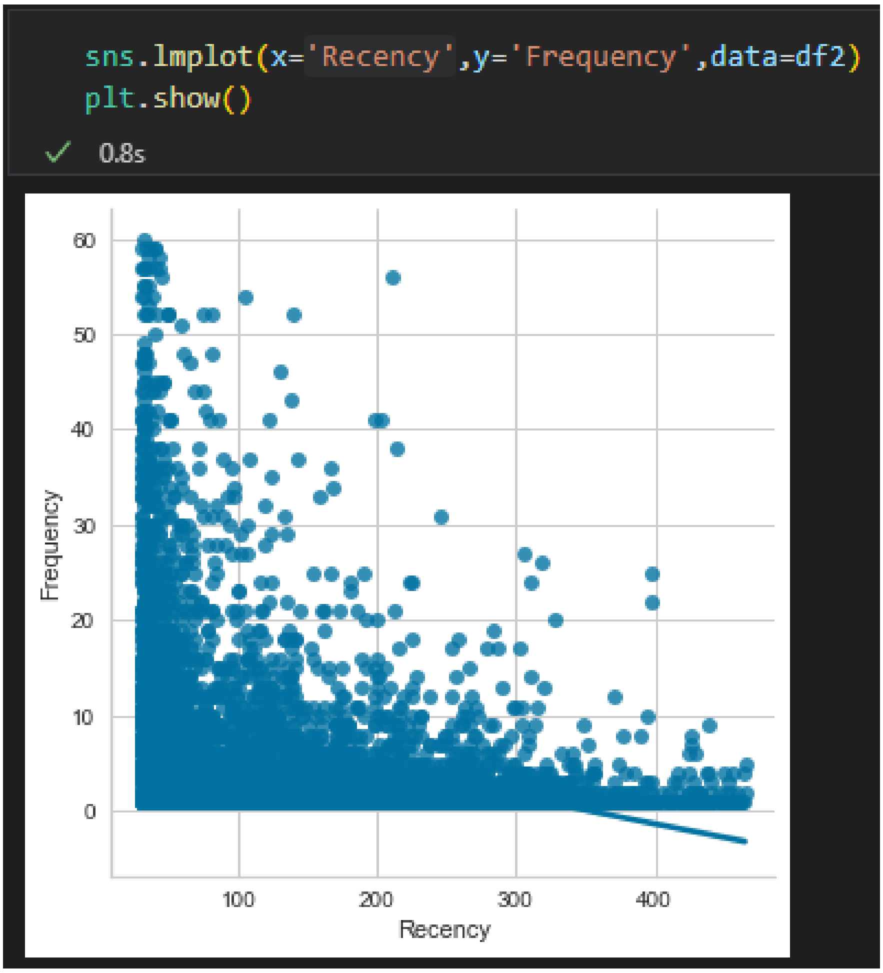


Figure 42 Scatter plot between Recency and Frequency

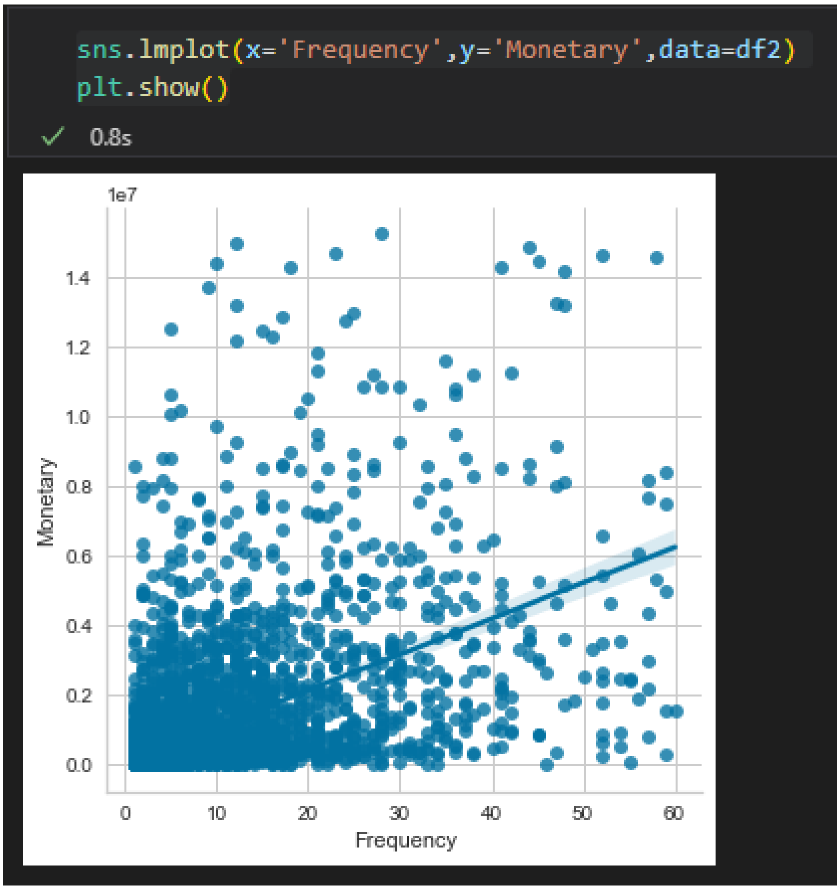


Figure 43 Scatter plot between Recency and Monetary

From the graph, we see that Frequency and Recency have an inverse linear relationship. The lower the number of recent purchases by customers, the higher the frequency of purchases.

- Frequency and Monetary have a positive linear relationship. The more frequent customers buy, the higher the total bill value for them.

```
sns.lmplot(x='Recency',y='Monetary',data=df2)
plt.show()
```
✓ 0.6s

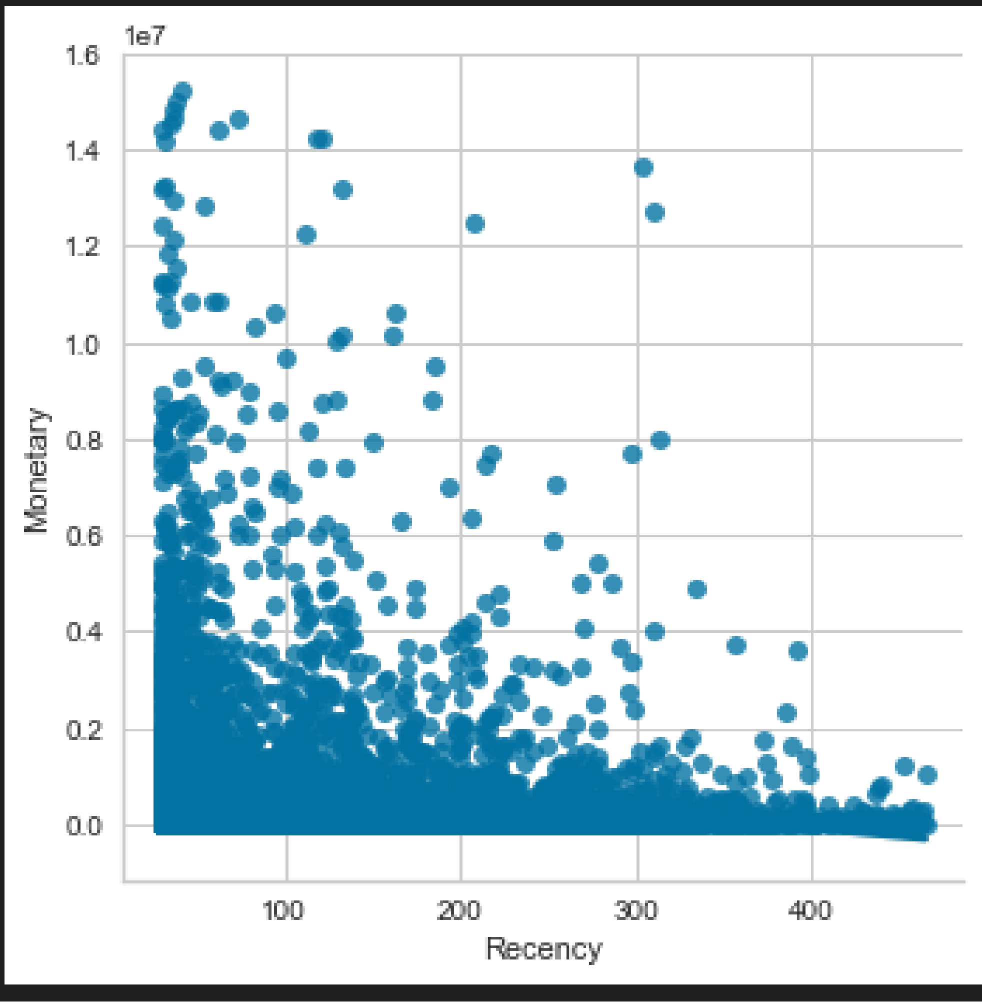


Figure 44 Scatter plot between Recency and Monetary

Recency and Monetary variables have an unclear linear relationship. But we can still find that the lower the number of last purchases a customer has, the higher their bill will be.

- Heatmap:

We use heatmap to clearly analyze the relationship between the three variables Recency, Frequency and Monetary.

```
cross_table1 = pd.crosstab(index=df2['Monetary_score'], columns=df2['Frequency_score'])
cross_table2 = pd.crosstab(index=df2['Monetary_score'], columns=df2['Recency_score'])
cross_table3 = pd.crosstab(index=df2['Frequency_score'], columns=df2['Recency_score'])
plt.figure(figsize=(20,30))
plt.subplot(311)
ax1 = sns.heatmap(cross_table1, cmap='viridis', annot=True, fmt=".0f")
ax1.invert_yaxis()
ax1.set_ylabel('Monetary')
ax1.set_xlabel('Frequency')
ax1.set_title('Monetary vs Frequency')
plt.subplot(312)
ax2 = sns.heatmap(cross_table2, cmap='viridis', annot=True, fmt=".0f")
ax2.invert_yaxis()
ax2.set_ylabel('Monetary')
ax2.set_xlabel('Recency')
ax2.set_title('Monetary vs Recency')
plt.subplot(313)
ax3 = sns.heatmap(cross_table3, cmap='viridis', annot=True, fmt=".0f")
ax3.invert_yaxis()
ax3.set_ylabel('Frequency')
ax3.set_xlabel('Recency')
ax3.set_title('Recency vs Frequency')
plt.show()
```

Figure 45 Code show heatmap of Recency, Frequency, Monetary

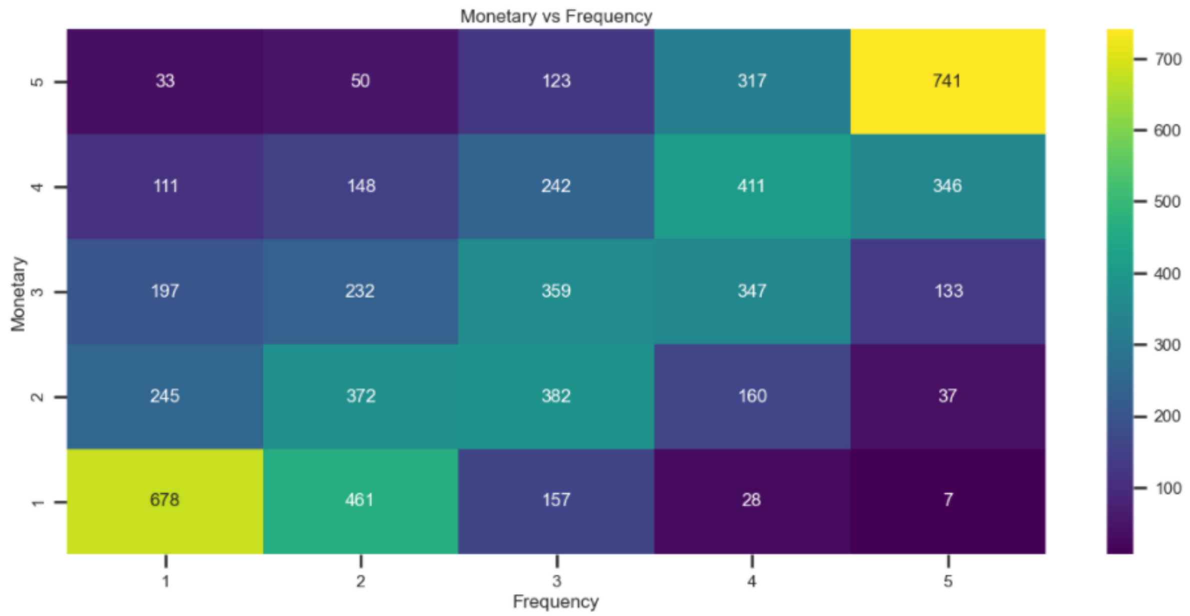


Figure 46 Heatmap between Frequency and Monetary

- From the chart, the most obvious thing is that the customer group with the highest purchase frequency and order value and the customer group with the lowest purchase frequency and order value account for the majority of the total number of customers.

- The number of customers with the lowest purchase frequency but with high order value and vice versa accounts for a very small number of total customers.
- The light colored cells tend to move diagonally up to the right, corresponding to the increasing Frequency and Monetary points. From that, it can be concluded that the two variables Frequency and Monetary have a strong positive linear correlation.

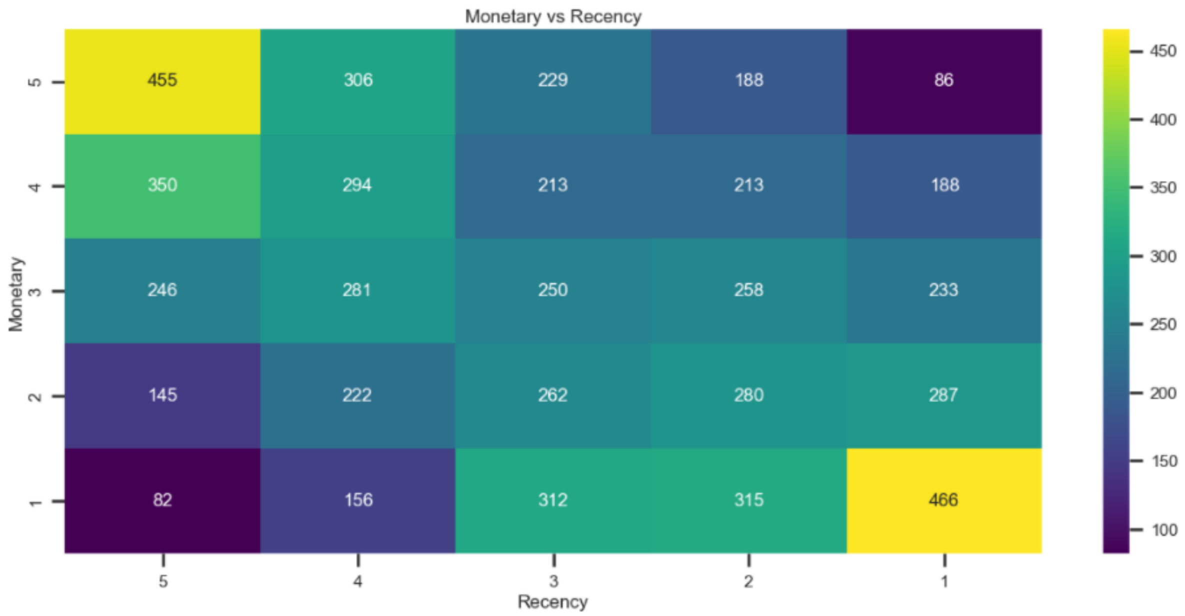


Figure 47 Heatmap between Recency and Monetary

- From the chart, it is easy to see that the two variables Recency and Monetary with the highest and lowest ranks account for the majority of total customers.
- The number of customers with the highest Rencency, the lowest Monetary and vice versa accounts for a very small number of total customers.
- The light colored cells tend to move diagonally up to the left corresponding to the increasing Frequency and Monetary points, but it is not clear. From that, it can be concluded that the two variables Frequency and Monetary have a positive and weak linear correlation.

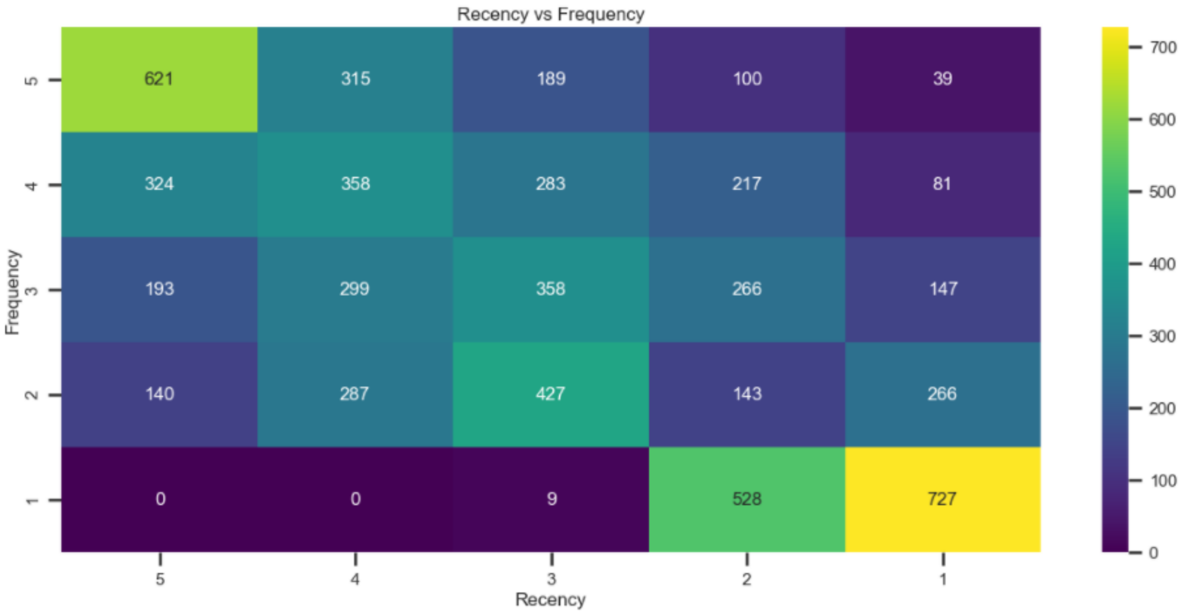


Figure 48 Heatmap between Recency and Frequency

- From the chart, it is easy to see that the two variables Recency and Frequency have the highest and lowest ranks, accounting for most of the total number of customers.
- The number of customers with the highest Rencency rank, the lowest Frequency and vice versa account for a very small number of total customers.
- The light colored cells tend to move diagonally up to the left corresponding to the increasing Frequency and Monetary points. From that, it can be concluded that the two variables Frequency and Recency have a strong positive linear correlation.

Visualizing customer segments of RFM:

- Barplot about number customer of each segment:

Figure 49 Code show countplot

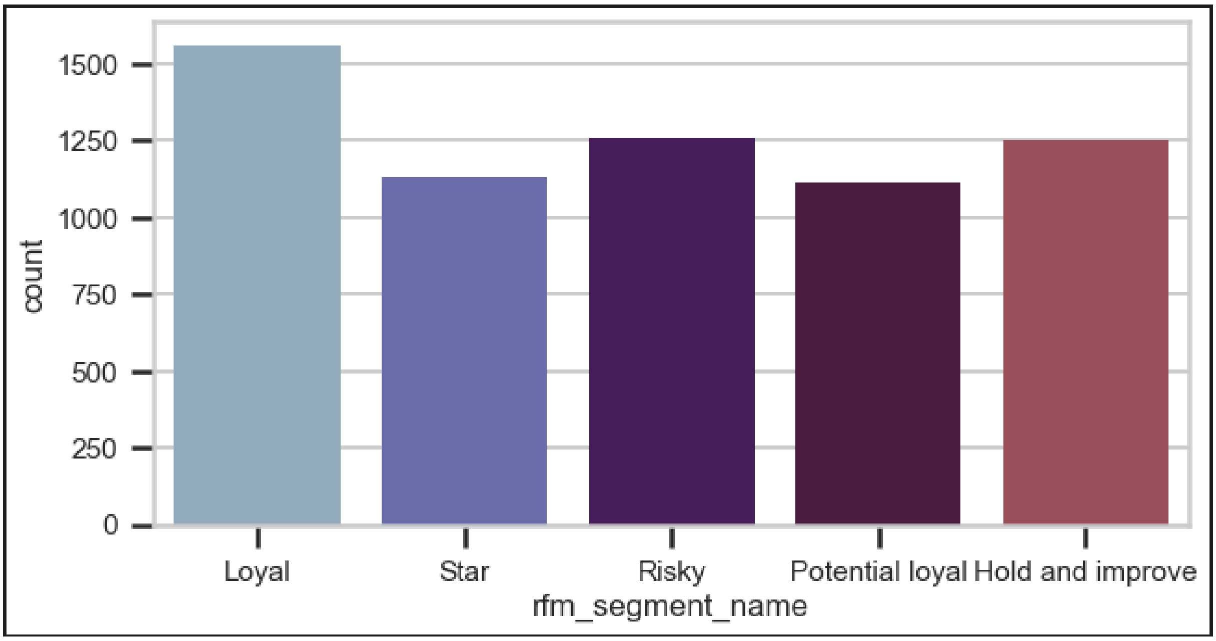


Figure 50 Barplot about number customer of each segment

The number of customers in the Loyal segment accounts for the largest number. Next is Risky and Hold and improve. Customers in the Star and Potential loyal segments account for the smallest number. This shows that businesses still have a lot to do to improve customers in the Risky segment.

- Summary of customer segments:

```
df2.groupby('rfm_segment_name').agg({
    'Recency' : ['mean', 'min','max'],
    'Frequency' : ['mean', 'min','max'],
    'Monetary' : ['mean','min','max','count']
})
```

✓ 0.1s

	Recency			Frequency			Monetary			
	mean	min	max	mean	min	max	mean	min	max	count
rfm_segment_name										
Hold and improve	188.549442	141	243	3.411483	1	56	4.595315e+05	1000	12500000	1254
Loyal	61.579116	30	140	5.256246	1	54	5.700453e+05	1000	14256000	1561
Potential loyal	114.839173	83	140	3.194969	1	32	3.662626e+05	1000	10182621	1113
Risky	313.379365	244	465	2.200000	1	31	2.580220e+05	1000	13676486	1260
Star	39.257750	30	82	16.976971	4	60	2.035796e+06	12000	15231000	1129

Figure 51 Code and result summary of customer segments

- Star: Recency, Frequency, Monetary indexes are all at the best level. 1129 customers have bought 2035796 units by shopping thrice every 40 days. Businesses need to focus on loyalty programs and new product introductions. These customers have proven to have a higher willingness to pay, so don't use discount pricing to generate incremental sales. Instead, focus on value added offers through product recommendations based on previous purchases.

- Loyal: Recency, Frequency, Monetary indexes are all at a good level. We can say that 1561 customers have bought 370045 units by shopping thrice every 62 days. Loyalty programs are effective for these repeat visitors. Advocacy programs and reviews are also common strategies. Lastly, consider rewarding these customers with Free Shipping or other benefits.
- Potential loyal: Recency, Frequency, Monetary indexes are all at an average level. 1113 customers have bought 366262 units by shopping thrice every 115 days. Business focus on increasing monetization through product recommendations based on past purchases and incentives tied to spending thresholds.
- Hold and improve: Recency, Frequency, Monetary indexes are all at a low level. 1254 customers have bought 459531 units by shopping thrice every 189 days. Business needs to have clear strategies in place for first time buyers such as triggered welcome emails that will pay dividends.
- Ricky: Recency, Frequency, Monetary indexes are all at a poor level. 1260 customers have bought 258022 units by shopping thrice every 313 days. Business needs to review policies on price, product quality, and employee attitudes to adjust.
- Score scales of each segment:

```python
df2.reset_index().groupby('rfm_segment_name').agg(
    customers=('User_id','count'),
    min_rfm=('rfm', 'min'),
    max_rfm=('rfm', 'max'),
).reset_index().sort_values(by='min_rfm')
```

✓ 0.1s

	rfm_segment_name	customers	min_rfm	max_rfm
3	Risky	1260	111	155
0	Hold and improve	1254	211	255
2	Potential loyal	1113	311	353
1	Loyal	1561	354	541
4	Star	1129	455	555

Figure 52 Code and result range of all segments

- Treemap:

```python
#treemap
df_treemap = df2.reset_index().groupby('rfm_segment_name').agg(
    customers=('User_id', 'count')
).reset_index()
fig, ax = plt.subplots(1, figsize = (10,10))

squarify.plot(sizes=df_treemap['customers'],
              label=df_treemap['rfm_segment_name'],
              alpha=.8,
              color=['tab:red', 'tab:purple', 'tab:brown', 'tab:pink', 'tab:gray']
             )
plt.axis('off')
plt.show()
```

✓ 0.3s

Figure 53 Code show treemap

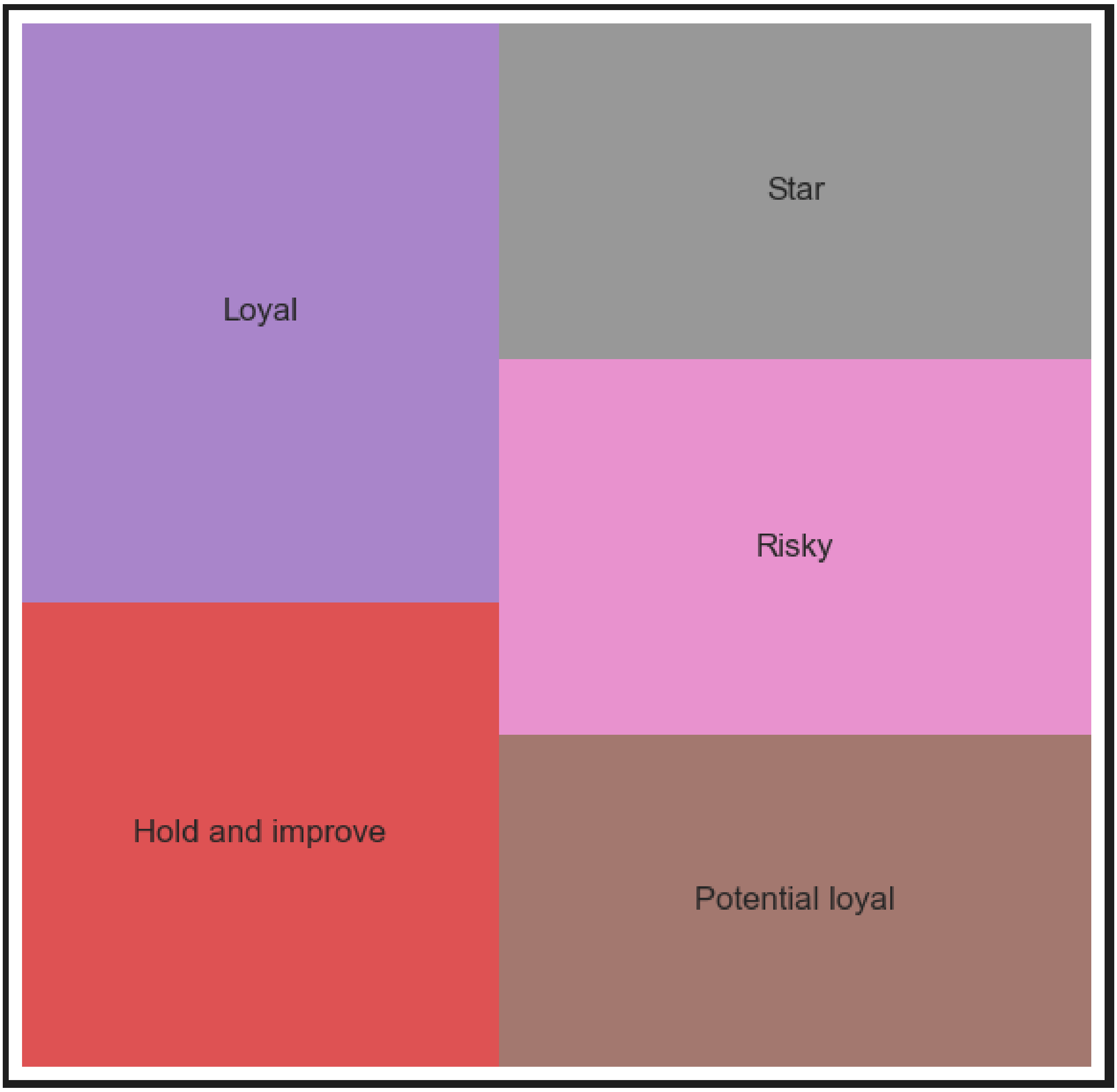


Figure 54 Treemap show 5 customer segments by RFM

The treemap gives us a really clear view of the named RFM segments and the relative volumes of customers present in each one. For this particular data set, where the tail is long and there are likely to be loads of truly lapsed customers in the “Risky” and “Hold and improve” segments, it could be beneficial to use an alternative binning approach so that the lapsed customers are dumped into a “churned” segment and can be excluded from costly marketing activity.

4.5. Customer Segmentation with K-means

4.5.1. Select K using Elbow and Calinski-Harabasz index Methods

- Select main columns:

```
df3 = df2[['Re_zs','Fe_zs','Mo_zs']]
df3.head()
```

✓ 0.1s

	Re_zs	Fe_zs	Mo_zs
User_id			
61386143	-0.713776	-0.090166	-0.154905
48453125	-1.048483	1.100775	0.431826
49921027	1.796531	-0.317012	-0.200237
44014594	2.744869	-0.203589	-0.193959
31058664	-0.434853	0.079969	0.028926

Figure 55 Code select main columns

- Using Elbow to select K:

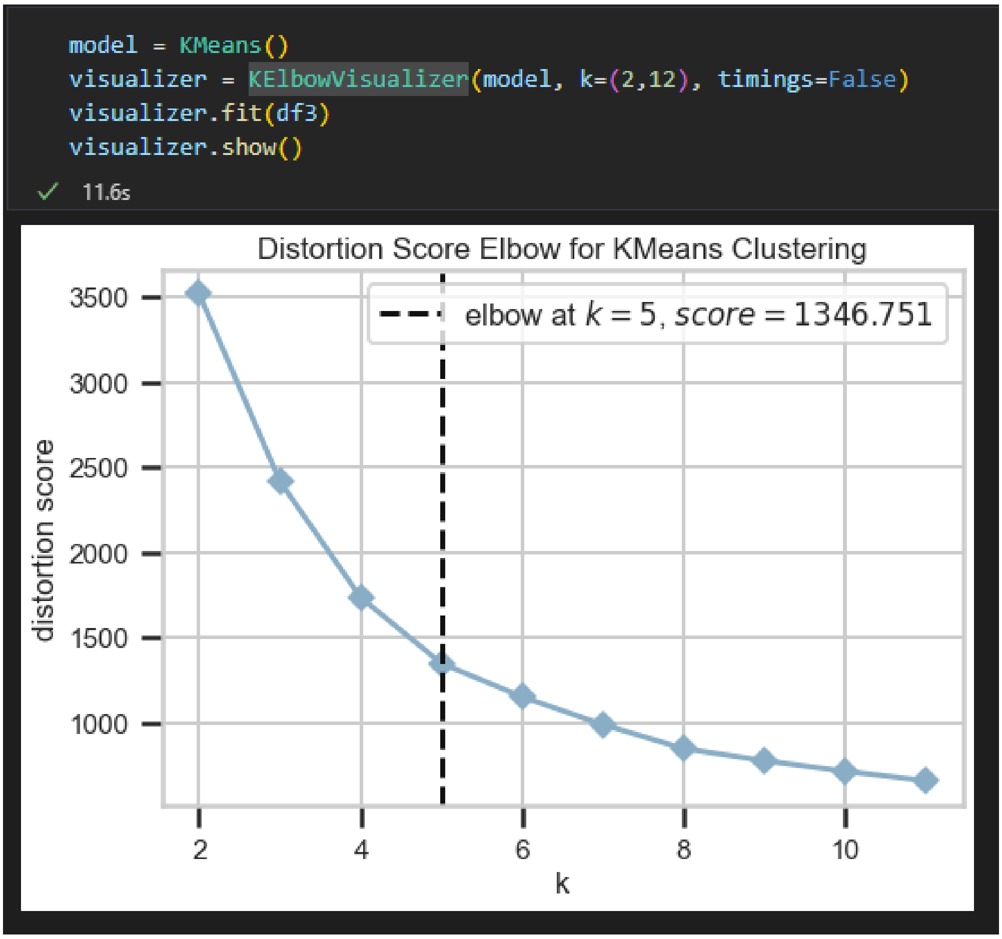


Figure 56 Line plot show SSE Score

With the SSE curve like the elbow, we have the elbow bend with K = 5 which will be the appropriate number of clusters. Explaining this, when increasing the number of clusters, the value of the SSE curve also increases almost evenly, that is, the difference between points in the cluster is almost unchanged. In other words, the SSE curve tends to gradually decrease in slope after the "elbow" point, and this position on the SEE curve is considered as the optimal point for the input parameter in the K-means clustering method.

- Using Calinski Harabasz Score to select K:

To ensure that the number of analyzed customer groups is 5 from the Elbow method, which is the best, the study measures the Calinski Harabasz Score on the number of clusters K=5 to obtain the following results:

```
visualizer = KElbowVisualizer(
model, k=(3,12), metric='calinski_harabasz', timings=False, locate_elbow=True)
visualizer.fit(df3)
visualizer.show()
```

✓ 9.3s

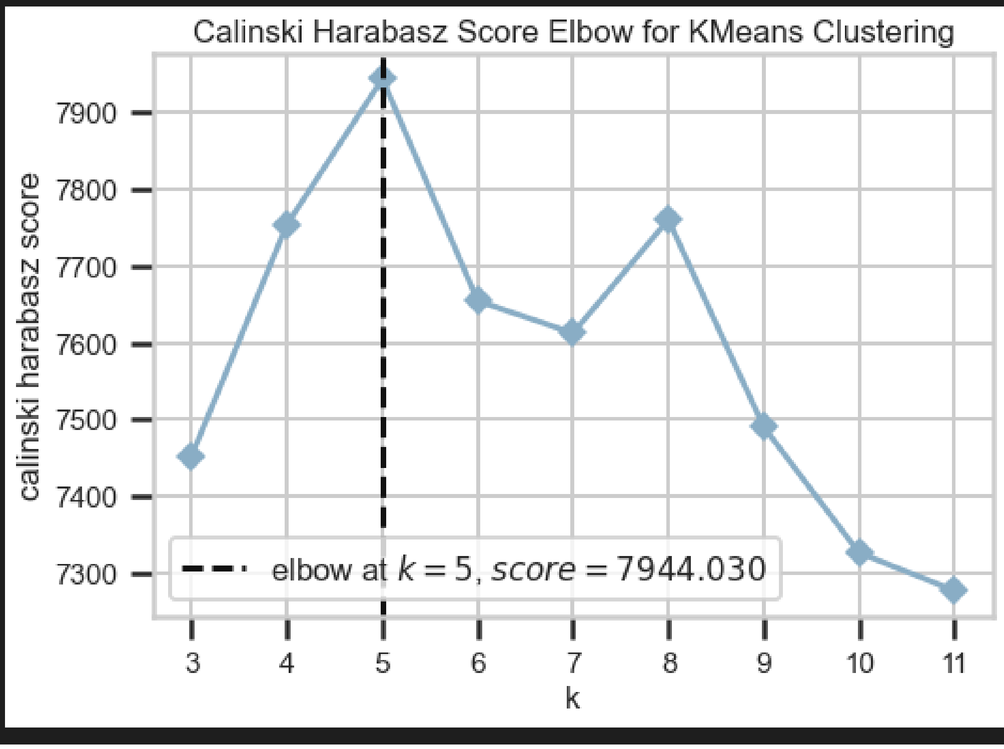


Figure 57 Line plot show Calinski Harabasz Score

With the obtained mean score of 7944 and the highest for all cluster numbers between 3 and 11. This explains that, with a cluster number of 5, the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters is the best.

4.5.2. Applying K-means

```
model = KMeans(n_clusters = 5, init = "k-means++",
max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = model.fit_predict(df3)
labels3 = model.labels_
centroids3 = model.cluster_centers_
```

✓ 0.9s

Figure 58 Code use K-means clustering

```
print(y_kmeans)
print(" Our cluster centers are as follows")
print(centroids3)
```

```
[0 3 4 ... 0 0 0]
 Our cluster centers are as follows
[[-0.83229391 -0.09539448 -0.10727875]
 [ 1.043932   -0.28694395 -0.15288658]
 [-0.04276801 -0.23210819 -0.12938576]
 [-0.84166064  1.28124341  0.59067421]
 [ 2.23103533 -0.3271202  -0.19048709]]
```

Figure 59 Code show cluster centers

- Number customer of each cluster by Recency, Frequency, Monetary

```python
clusters = df3.copy()
clusters['cluster']=model.fit_predict(df3)
clusters.groupby(['cluster']).count()
```

✓ 0.9s

cluster	Re_zs	Fe_zs	Mo_zs
0	2331	2331	2331
1	1359	1359	1359
2	1733	1733	1733
3	461	461	461
4	433	433	433

Figure 60 Summary number customer of each cluster by Recency, Frequency, Monetary

We have 5 groups after running K-means, numbered from 0 to 4.

4.5.3. Visualizing customer segments of K-means

- Barplot about number customer of each cluster:

```
sns.countplot(y_kmeans)
✓ 0.2s
c:\Users\truon\anaconda3\lib\site-packages\seaborn\_decorators.py:36
valid positional argument will be `data`, and passing other argument
  warnings.warn(

<AxesSubplot:ylabel='count'>
```

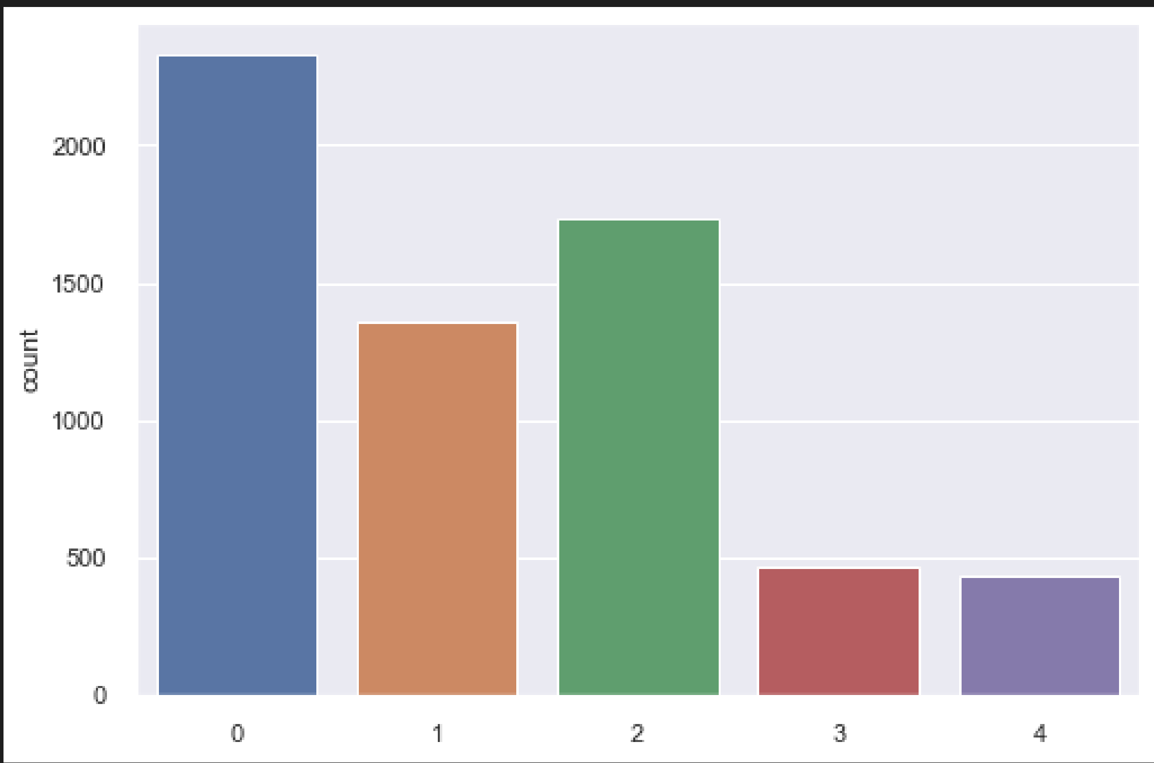


Figure 61 Barplot about number customer of each cluster

Cluster 0 has the most customers, followed by cluster 2, cluster 1, cluster 3 and cluster 4 is the cluster with the least number of customers.

- Scatterplot 3D of the clusters:

```
x=df3.values
fig = plt.figure(figsize = (15,15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x[y_kmeans == 0,0],x[y_kmeans == 0,1],x[y_kmeans == 0,2], s = 40 , color = 'blue', label = "cluster 0")
ax.scatter(x[y_kmeans == 1,0],x[y_kmeans == 1,1],x[y_kmeans == 1,2], s = 40 , color = 'orange', label = "cluster 1")
ax.scatter(x[y_kmeans == 2,0],x[y_kmeans == 2,1],x[y_kmeans == 2,2], s = 40 , color = 'green', label = "cluster 2")
ax.scatter(x[y_kmeans == 3,0],x[y_kmeans == 3,1],x[y_kmeans == 3,2], s = 40 , color = 'red', label = "cluster 3")
ax.scatter(x[y_kmeans == 4,0],x[y_kmeans == 4,1],x[y_kmeans == 4,2], s = 40 , color = 'purple', label = "cluster 4")
ax.set_xlabel('Recency-->')
ax.set_ylabel('Frequency-->')
ax.set_zlabel('Monetary-->')
ax.legend()
plt.show()
```

Figure 62 Code show scatter plot 3D with 5 clusters

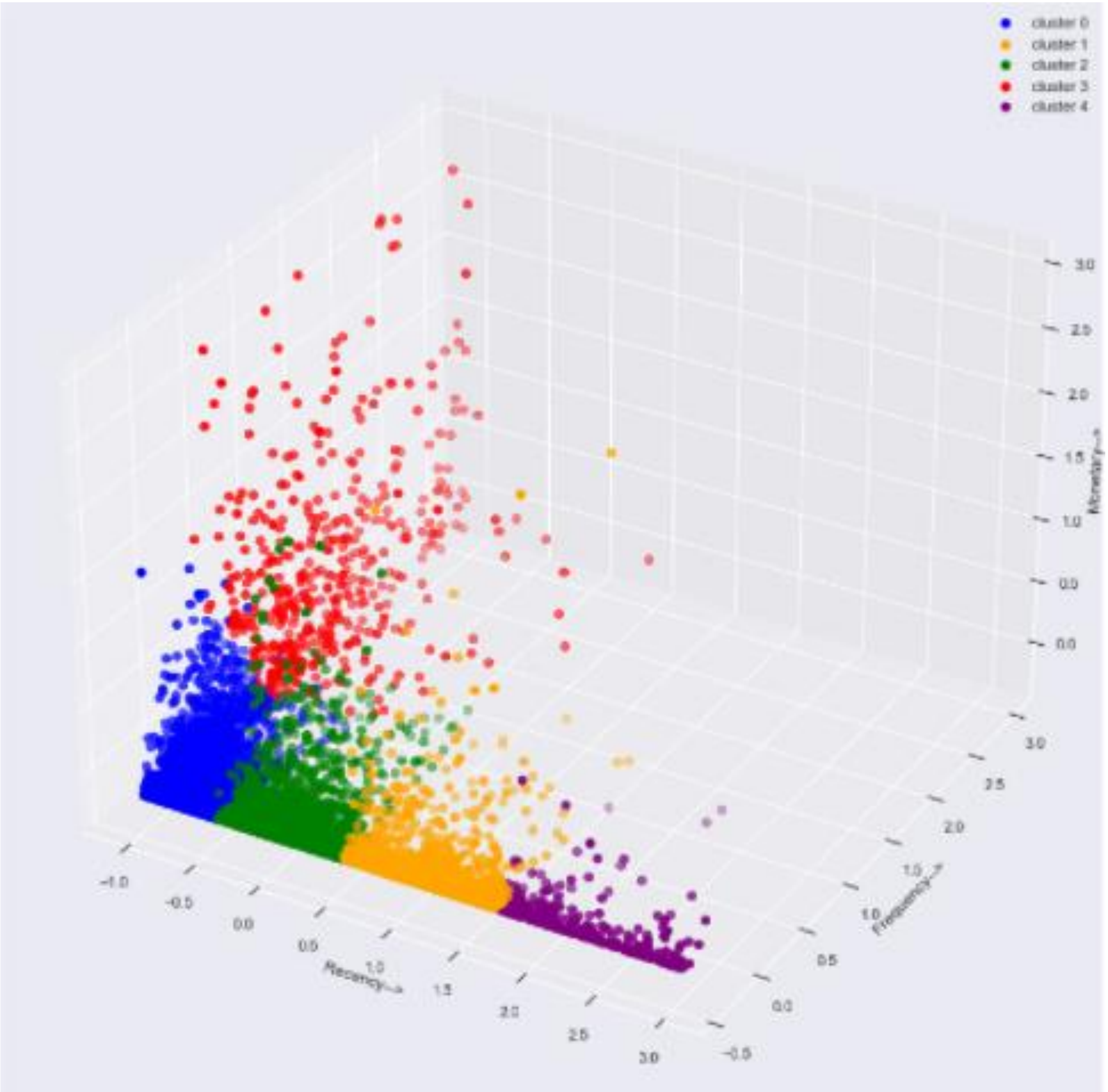


Figure 63 Scatter plot 3D show 5 clusters

The clustering results are visualized on a 3D scatter plot, with the densities of clusters 0.1 and 4 being the most stable, followed by cluster 2 which is less stable with a few points located quite far from the center. cluster. Cluster 3 has the lowest stability with points located quite discretely from each other.

4.5.4. Validation by Silhouette score:

```
#validation
sil_score = silhouette_score(df3, labels3, metric='euclidean')
print('Silhouette Score: %.3f' % sil_score)
from yellowbrick.cluster import SilhouetteVisualizer
model = KMeans(5)
visualizer = SilhouetteVisualizer(model)
visualizer.fit(df3)
visualizer.poof()
```

Figure 64 Code show Silhouette score

```
Silhouette Score: 0.452
```

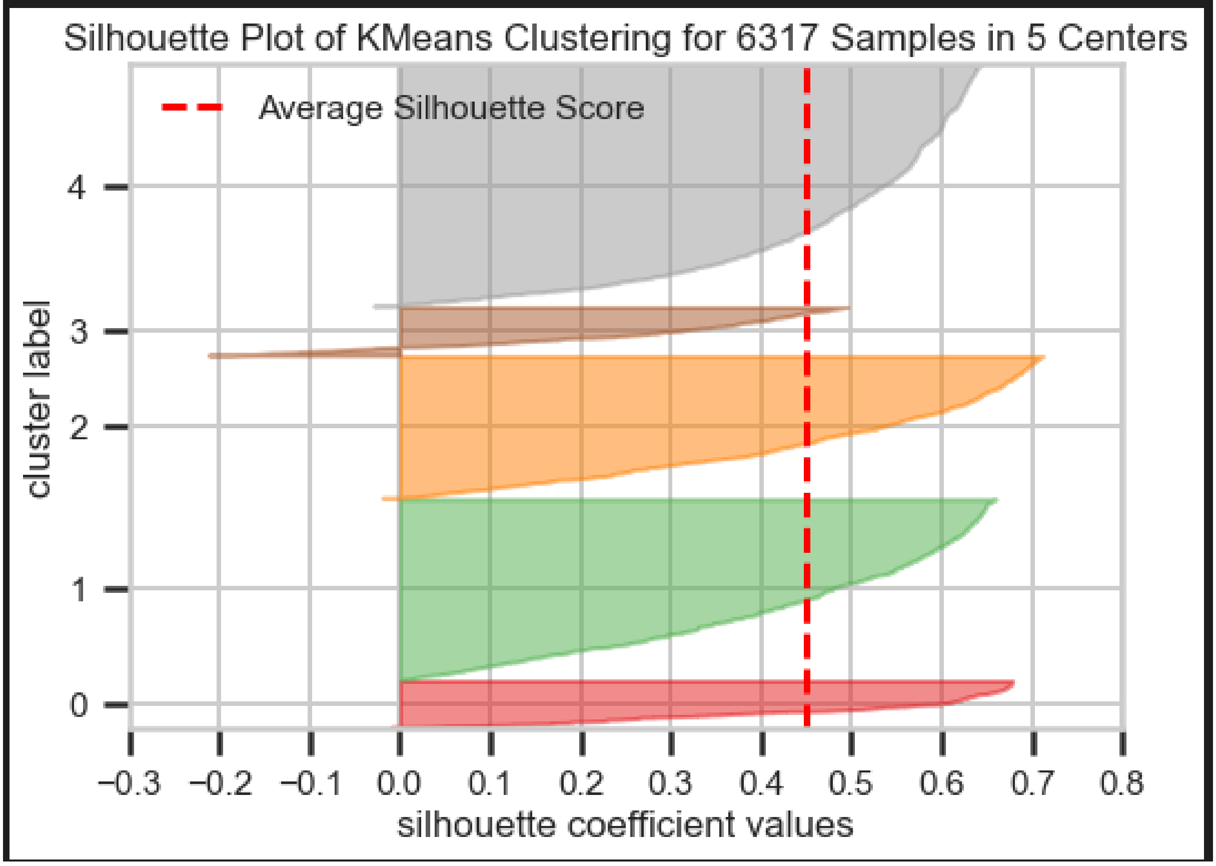


Figure 65 Silhouette plot of K-means clustering for K=5

With 5 clusters, the Silhouette Score is 0.452, close to 1. This explains that, with the number of clusters of 5, the distance from the objects in the cluster to the cluster center has been optimized and no cluster eccentricity occurs overlap between clusters.

4.5.5. Analysis clusters created K-means

Summary all of clusters

```
df2['cluster']= clusters['cluster']
df2.groupby('cluster').agg({
    'Recency' : ['mean','min','max'],
    'Frequency' : ['mean','min','max'],
    'Monetary' : ['mean','min','max','count']})
```

	Recency			Frequency			Monetary			
	mean	min	max	mean	min	max	mean	min	max	count
cluster										
0	53.233376	30	107	5.907336	1	22	5.698448e+05	1000	8572500	2331
1	254.918322	196	320	2.526858	1	31	3.478350e+05	1000	13676486	1359
2	138.081362	92	200	3.502020	1	25	4.629538e+05	1000	10600000	1733
3	52.245119	30	214	30.182213	10	60	3.949253e+06	27000	15231000	461
4	382.588915	319	465	1.826790	1	25	1.693870e+05	1000	4900000	433

Figure 66 Code show summary of 5 clusters

- From the summary statistics we can conclude that our most active, most recent, best segment is segment 3. But segment 3 only accounts for 7.3% of total customers. From that, it can be inferred that this is a star customer group.
- The second active, recent, profitable segment is the 0 segment. Segment 0 has a large number of customers, accounting for more than 1/3 of the total number of customers. From that, it can be inferred that this is a loyal customer group.
- Segment 2 is the 3rd best segment with the Recency, Frequency, Monetary values ranking 3rd. Segment 2 has the number of customers accounting for nearly 1/3 of the total number of customers. From that, this is the segment of potential loyal customers.

- Segment 1 is the 3rd best segment with the Recency, Frequency, Monetary values ranking 3rd. Segment 1 has the number of customers accounting for over 1/5 of the total number of customers. From that, this is the segment of hold and improve customers.
- Segment 4 can be defined as totally lapsed with on average one purchase made over a year ago. The average frequency of purchases is less than 2 times a year. Customers in this segment also account for a very small number. This is the segment of risky customers.

```
df2.groupby(['cluster','rfm_segment_name']).size()
```

✓ 0.5s

```
cluster  rfm_segment_name
0        Loyal                1425
         Potential loyal       164
         Star                  742
1        Hold and improve      532
         Risky                 827
2        Hold and improve      710
         Loyal                  79
         Potential loyal       944
3        Hold and improve       12
         Loyal                  57
         Potential loyal         5
         Star                  387
4        Risky                 433
```

Figure 67 Summary clusters and RFM segments

From the above results, we see:

- Cluster 4 and Risky segment have the same number of customers.
- Cluster 3 includes 3 parts: Hold and improve, Loyal, Potential loyal. In which Loyal accounts for the majority.

- Cluster 2 includes 3 parts: Hold and improve, Loyal, Potential loyal. In which, Hold and improve, Potential loyal account for the most.
- Cluster 1 includes 2 segments: Hold and improve, Risky. Both are not much different from each other.
- Cluster 0 includes 3 segments: Loyal, Potential loyal, Star. In which Loyal accounted for the majority.

Using line plots to compare RFM with K-means:

```
#compare
df4 = df2[['Re_zs', 'Fe_zs', 'Mo_zs','rfm_segment_name', 'cluster']].reset_index()
rfm_melted = pd.melt(frame= df4, id_vars= ['User_id','rfm_segment_name', 'cluster'],
var_name = 'metrics', value_name = 'value')
rfm_melted.head()
```

✓ 0.7s

	User_id	rfm_segment_name	cluster	metrics	value
0	61386143	Loyal	0	Re_zs	-0.713776
1	48453125	Star	3	Re_zs	-1.048483
2	49921027	Risky	4	Re_zs	1.796531
3	44014594	Risky	4	Re_zs	2.744869
4	31058664	Potential loyal	0	Re_zs	-0.434853

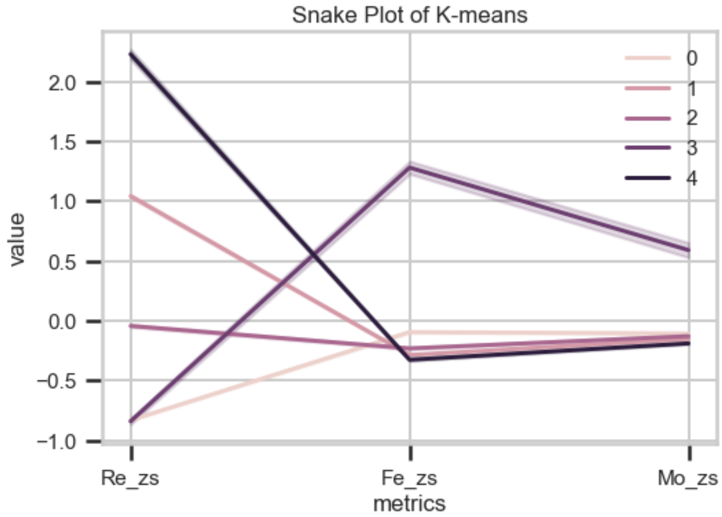
Figure 68 Code show values R,F,M of cluster and rfm_segment_name

```
sns.lineplot(x = 'metrics', y = 'value', hue = 'rfm_segment_name', data = rfm_melted)
plt.title('Snake Plot of RFM')
plt.legend(loc = 'upper right')
plt.show()
```

Figure 69 Code show line plot of RFM

```
sns.lineplot(x = 'metrics', y = 'value', hue = 'cluster', data = rfm_melted)
plt.title('Snake Plot of K-means')
plt.legend(loc = 'upper right')
plt.show()
```

Figure 70 Code show line plot K-means



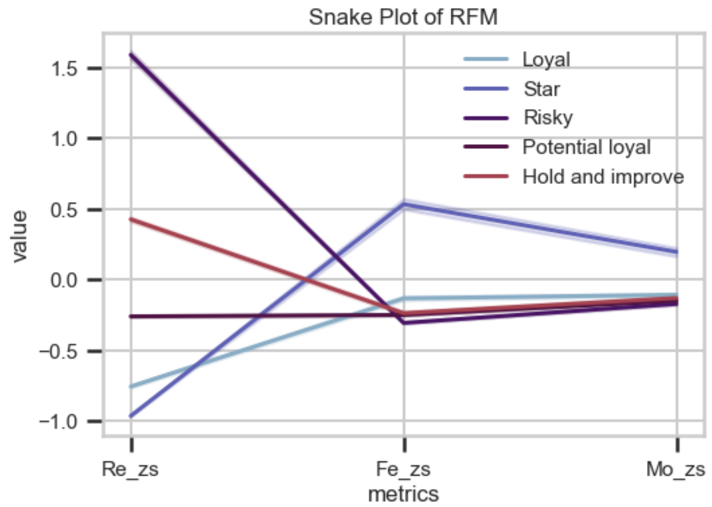


Figure 71 Line plots K-means and RFM

- Cluster 3 with Star: Recency does not have much difference, Frequency of cluster 3 is higher than Star, so is Monetary.
- Cluster 0 with Loyal: Almost the same in 3 factors
- Cluster 2 with Potential Loyal: Frequency and Monetary are not much different, Recency of cluster 2 is higher than Potential Loyal
- Cluster 1 with Hold and improve: Frequency and Monetary do not have much difference, Recency of cluster 1 is higher than Hold and improve
- Cluster 4 with Risky: Frequency and Monetary do not have much difference, Recency of cluster 4 is higher than Risky.

- From the above results, we see that the difference between the row segmentation of RFM and K-means methods is not so obvious, which is mainly affected by the Recency variable.
- Besides, we easily realize, clustering by K-means tends to specialize customers in a good segment (Star, Loyal) and risk segment (Risky) than RFM method.

CHAPTER 6. CONCLUSION

We made two kinds of segmentation, RFM quantiles and K-Means clustering methods. Customers have been categorized into 5 buckets based on Recency, Frequency and Monetary value of their purchases. Targeted strategy to be applied for each customer segment.

With the application of methods and algorithms such as Silhouette, Calinski Harabasz Score, Z-Score, Verification Rules help ensure reliability and accuracy of data analysis results.

With the result, we figured out ‘best’ customers, the most profitable group. This also tells business on which customer group we should focus on and to whom to give special offers or promotions among the customers. Businesses can select the best communication channel for each segment and improve new marketing strategies.

Through customer segmentation, we can develop business strategies and business organization in accordance with the target of serving each segment. At the same time, tracking the customer structure change on each segment over time also helps to assess the development level of the company's customer base? The company needs to roll out adjustments and strategies on how to develop in the direction of increasing the proportion of VIP customers, keeping customers coming back to shop more often and improving order value per purchase.

This data set is the transaction information of an online store, collected from January 2021 to March 2022. They have many similarities with online retail stores in our country. Therefore, this project is highly practical and will be studied further in the future.

CHAPTER 7. REFERENCES

[1] Ching-Hsue Cheng, You-Shyang Chen. (2009). Classifying the segmentation of customer value via RFM model and RS theory. *Taiwan: Expert Systems with Applications 36.*

[2] Kim-Giao Tran, Van-Ho Nguyen, Thanh Ho. (2009). Customer segmentation analysis and customer lifetime value prediction using Pareto/NBD and Gamma-Gamma model. *Vietnam National University, Ho Chi Minh City, Vietnam.*

[3] Rendra Gustriansyah, Nazori Suhandi, Fery Antony. (2020). Clustering optimization in RFM analysis based on k-means. *Faculty of Computer Science, Universitas Indo Global Mandiri, Indonesia.*

[4] Dessi G. (2020). RFM Customer Segmentation with K Means.

[5] UYSAL, Ümit Cengiz. (2009). Rfm-based Customer Analytics in Public Procurement Sector.

[6] Ho, T., & Nguyễn, S. (2021). An interdisciplinary research between analyzing customer segmentation in marketing and machine learning method. *Science & Technology Development Journal - Economics - Law and Management*, *6*(1), 2005-2015.

[7] U. Kaymak, "Fuzzy target selection using RFM variables," Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569), 2001, pp. 1038-1043 vol.2, doi: 10.1109/NAFIPS.2001.944748.

[8] Segment selection by relationship strength (Schijns and Schroder (1996))

[9] Do you want to keep your customers forever?(Joseph Pine, Peppers & Rogers(2009))

[10] X. He and C. Li, "The Research and Application of Customer Segmentation on E-Commerce Websites," 2016 6th International Conference on Digital Home (ICDH), 2016, pp. 203-208, doi: 10.1109/ICDH.2016.050.

[11] T. Jiang and A. Tuzhilin, "Improving Personalization Solutions through Optimal Segmentation of Customer Bases," in IEEE Transactions

on Knowledge and Data Engineering, vol. 21, no. 3, pp. 305-320, March 2009, doi: 10.1109/TKDE.2008.163.
[12] David Xuân. (2020). *Tự học ML* | Thuật toán K-mean ++. https://cafedev.vn/tu-hoc-ml-thuat-toan-k-mean/. [Accessed 2/8/2022].
[13] Misha PyShark. (2022). Calinski-Harabasz Index for K-Means Clustering Evaluation using Python. https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/. [Accessed 3/8/2022].
[14] Admin. (2020). Bài 9: Unsupervised Learning: Clustering phần 2 - Lập trình AI bằng Python. https://vncoder.vn/bai-hoc/unsupervised-learning-clustering-phan-2-405. [Accessed 3/8/2022].
[15] Johann Siemens, Unsplash. (2021). How to visualize RFM data using treemaps. https://practicaldatascience.co.uk/data-science/how-to-visualise-rfm-data-using-treemaps?fbclid=IwAR1by6lgu0AEU_gKPMStB0LbgnWBBwTBX8P0nfn-YSh-wWlvsg30iF-udss. [Accessed 2/8/2022].