UNIVERSITY OF ECONOMICS AND LAW

Faculty of Information Systems

Report

RFM ANALYSIS FOR SALES SYSTEM OF NON-STORE ONLINE RETAIL AT UK

Instructor: PhD Ho Trung Thanh

Implementation: Mai Nguyen An Binh

Student ID: K184060777

TP. HCM City, May, 2021

TABLE OF CONTENTS

CHAPTER 1: IDENTIFICATION OF DATA MINING PROBLEMS

1.1 Data overview

1.1.1. Data Set Information

This Online Retail II data set contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique all-occasion gift-ware. Many customers of the company are wholesalers.

Link Dataset: https://archive.ics.uci.edu/ml/datasets/Online+Retail+II#

Link Download: https://archive.ics.uci.edu/ml/machine-learning-databases/00502/

The reason for choosing the topic: the topic is attractive, close and easy to run RFM with this data.

1.1.2. Attribute Information

The data consists of 8 columns and 541911 rows of data.

Name	DataType	Meaning
Invoice	Nominal	A 6-digit integral number uniquely assigned to each transaction. If this code starts with the letter 'c', it indicates a cancellation
StockCode	Nominal	A 5-digit integral number uniquely assigned to each distinct product
Description	Nominal	Product (item) name
Quantity	Numeric	The quantities of each product (item) per transaction

InvoiceDate	Numeric	The day and time when a transaction was generated
Price	Numeric	Product price per unit in sterling (£)
Customer ID	Nominal	A 5-digit integral number uniquely assigned to each customer
Country	Nominal	The name of the country where a customer resides
Amout	Numeric	The amount of each per transaction

1.1.3. Problem statement

Based on the attributes already available in the data set to analyze customer value through RFM. After having the input are 3 factors: Recency, Frequency, Monetary . We will use K-Mean clustering, an unsupervised learning algorithm to group customers with the same VIP level into a group.

1.1.4. Data Mining Tools

- Data Mining: Jupyter notebook
- Platform: Anaconda Nigarator (anaconda3)

1.1.5. Attached Library

CHAPTER 2: THEORETICAL BASIS OF RFM ANALYSIS

2.1. What is RFM Analysis?

- RFM is a method used for analyzing customer value. It is commonly used in database marketing and direct marketing and has received particular attention in retail and professional services industries.
- For example, it can be used to gauge how recently a customer has purchased a product or service (recent visit), how many times a customer has purchased since a certain date (frequency) and how much the customer spent in a given day a period of time (currency).
  - Recency (R) – recent purchase (or possibly visit, remember you)
  - Frequency (F) – frequency of customer purchases (how long or how many products)
  - Monetary Value (M) – Money, value, how much customers spend.

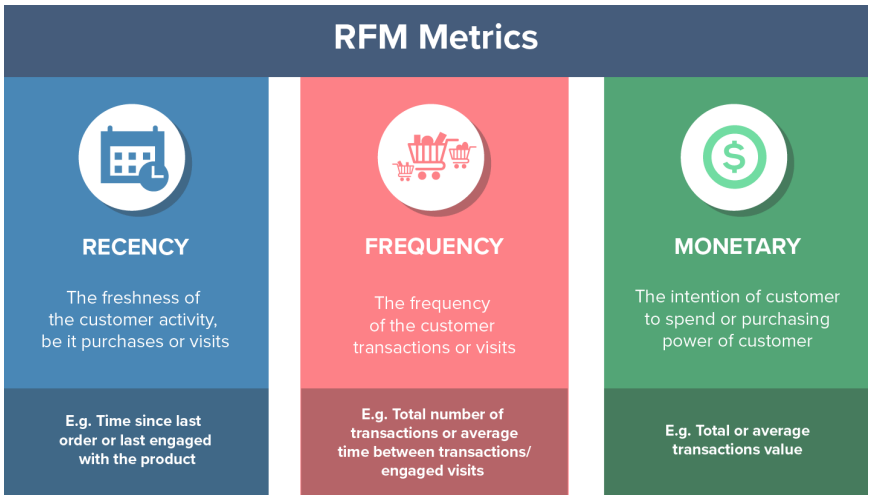


Figure 2.1: RFM metrics

- While there are countless ways to perform segmentation, RFM analysis is popular for three reasons:
  - Use objective scales – provide a superior and concise description of the customer.
  - Simple – administrators can use it effectively without the need for sophisticated software or data analysts.

- Intuitive – the output of this segmentation method is easy to understand and interpret.

2.2. Benefit of RFM

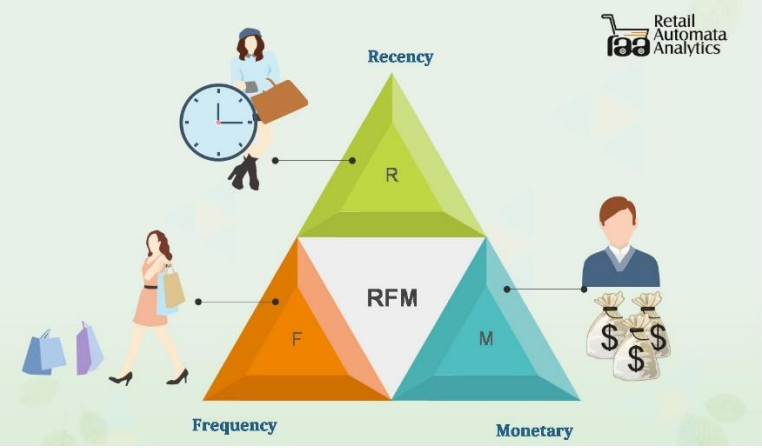
RFM analysis helps marketers find answers to the following questions:

- Who are companies' best customers?
- Which of companies' customers could contribute to your churn rate?
- Who has the potential to become valuable customers?
- Which of companies' customers can be retained?
- Which of companies' customers are most likely to respond to engagement campaigns?

2.3. Performing RFM analysis

Step by Step (3 main steps)

- The first step in building an RFM model is to assign Recency, Frequency and Monetary values to each customer.
- The second step is to divide the customer list into tiered groups for each of the three dimensions (R, Fand M).
- The third step is to select and label groups of customers to whom specific types of communications will be sent, based on the RFM segments.



CHAPTER 3: DATA PREPROCESSOR

3.1. Remove meaningless attributes during mining data

Understanding the necessary factors for RFM, we perform the removal of unnecessary columns as follows.

```
# import the pandas
import pandas as pd
```

```
dataset = pd.read_excel("D:\Studying\DA\DA-K18406C-K184060777-MaiNguyenAnBinh-RFM\online_retail_II.xlsx")
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541910 entries, 0 to 541909
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      541910 non-null  object
 1   StockCode    541910 non-null  object
 2   Description  540456 non-null  object
 3   Quantity     541910 non-null  int64
 4   InvoiceDate  541910 non-null  datetime64[ns]
 5   Price        541910 non-null  float64
 6   Customer ID  406830 non-null  float64
 7   Country      541910 non-null  object
 8   Amount       541910 non-null  float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 37.2+ MB
```

3.1.1. Delete the Stock code column

Stock code column stores information about integral number uniquely assigned to each distinct product. It is no nesscessary for analysising RFM.

```
#Delete the Stock code column
dataset = dataset.drop(['StockCode'],axis=1)
```

3.1.2. Delete the Description column

Description column stores information about name of product. It is no nesscessary for analysising RFM.

```
#Delete the Description column
dataset = dataset.drop(['Description'],axis=1)
```

3.1.3. Delete the Price column

Price column stores information about price for per product. It is no nesscessary for analysising RFM.

```
#Delete the Price column
dataset = dataset.drop(['Price'],axis=1)
```

3.1.4. Delete the Quanity column

Quanity column stores information about quanity for per transaction. It is no nesscessary for analysising RFM.

```
#Delete the Quanity column
dataset = dataset.drop(['Quanity'],axis=1)
```

3.1.5. Delete the Country column

Country column stores information about where to sell products . It is no nesscessary for analysising RFM.

```
#Delete the Country column
dataset = dataset.drop(['Country'],axis=1)
```

3.1.6. Remaining data

After doing the task “Remove meaningless attributes during mining data”. The remaining data is 4 columns as shown below.

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541910 entries, 0 to 541909
Data columns (total 4 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      541910 non-null  object
 1   InvoiceDate  541910 non-null  datetime64[ns]
 2   Customer ID  406830 non-null  float64
 3   Amount       541910 non-null  float64
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 16.5+ MB
```

3.2. Remove null or unknown attributes

3.2.1. Remove null attributes

NULL is the term used to represent a missing value. It does not equalize during the running of the algorithm. On the other hand, it also dilutes our data.

Find NULL value in data

```
# Find NULL value
total = df.isnull().sum().sort_values(ascending=False)
percent_1=df.isnull().sum()/df.isnull().count()*100
percent_2 = (round(percent_1,1)).sort_values(ascending=False)
missing_data=pd.concat([total,percent_2],axis=1,keys=['Total','%'])
missing_data.head(5)
```

Result of “Find NULL value”

	Total	%
Amount	3	0.0
Unnamed: 0	0	0.0
CustomerID	0	0.0
OrderDate	0	0.0
OrderID	0	0.0

Data has 3 Null values, this is a small number, so we skip this step ““Find NULL value”.

3.2.2. Remove unknown attributes

The data value of Unknown has no meaning in the process of finding the relationship between factors in RFM, so we delete those Unknown attributes, string data is very important and cannot be replaced as numeric by value. average or most common value. There are two ways to handle the unknow string type: remove rows from the dataframe or see what percentage of the remaining values are and then replace unknow with that ratio. Here, we choose to delete the line containing the Unknow value.

3.5. Data preprocessing results

After doing the task “Remove null or unknown attributes”, data have 406830 rows. This ensures non-discrete running of the algorithm and gives the most efficient RFM model.

```
df = pd.read_csv("D:\Studying\DA\dataCustomerRFM.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303295 entries, 0 to 303294
Data columns (total 5 columns):
 #   Column      Non-Null Count   Dtype  
---  ------      --------------   -----  
 0   Unnamed: 0  303295 non-null  int64  
 1   CustomerID  303295 non-null  object 
 2   OrderDate   303295 non-null  object 
 3   OrderID     303295 non-null  object 
 4   Amount      303292 non-null  float64
dtypes: float64(1), int64(1), object(3)
memory usage: 11.6+ MB
```

CHAPTER 4: RFM ANALYSIS

4.1. Calculate the values of Recency, Frequency and Monetary

4.1.1. Calculate the values of Recency

```
from datetime import datetime
# 1. Calculate Recency
# 1.1. Find the most recent orderDate.

dfRecentOrder = pd.pivot_table(data = df,
                index = ['CustomerID'],
                values = ['OrderDate'],
                aggfunc = {'OrderDate':max}
               )

dfRecentOrder.columns = ['RecentOrderDate']
df = pd.merge(df, dfRecentOrder.reset_index(), on = ['CustomerID'])
df['RecentOrderDate'] = df['RecentOrderDate'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d %H:%M:%S'))
df['Recency'] = df['RecentOrderDate'].apply(lambda x: (datetime.now() - x).days)
```

Because the higher the number of days since the last purchase, the lower the customer's active level. Therefore, the smaller this value is, the higher the customer will rank. Therefore, we need to change the sign of Recency so that the value of the variable is covariant with the customer's rank.

```
# Change direction of recency
df['Recency'] = - df['Recency']
```

```
print(df['Recency'])
```

```
0         684
1         666
2         601
3         662
4         633
         ...
303290    612
303291    612
303292    610
303293    628
303294    589
Name: Recency, Length: 303295, dtype: int64
```

Next, we will calculate the frequency of customers buying over the entire study period.

4.1.2. Calculate the values of Frequency

```
# 2. Calculate Frequency
dfFrequency = df.groupby('CustomerID').OrderID.nunique().to_frame()
dfFrequency.columns = ['Frequency']
df = pd.merge(df, dfFrequency.reset_index(), on = 'CustomerID')
```

Result of “Calculate the values of Frequency”

4.1.3. Calculate the values of Monetary

```
# 3. Calculate Monetary
dfMonetary = df.groupby('CustomerID').Amount.sum().to_frame()
dfMonetary.columns = ['Monetary']
df = pd.merge(df, dfMonetary.reset_index(), on = 'CustomerID')
```

Result of “Calculate the values of Monetary”

4.2. Divide the customer and paste label groups

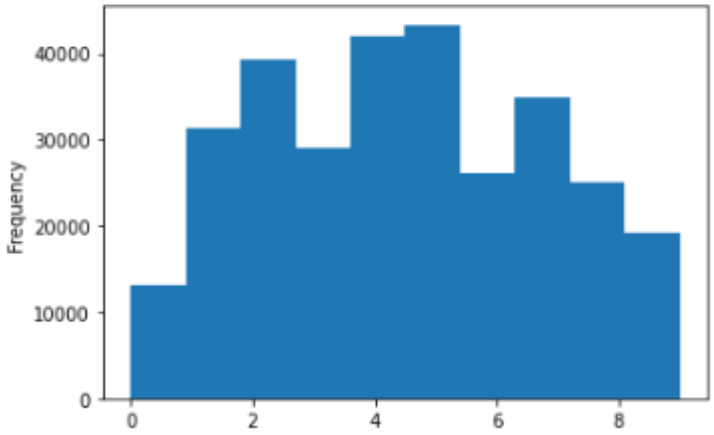
So we have finished calculating the rank for each customer. Let's see how the customer rank distribution is through histogram with the number of bins = 10.

```
orderFrequencies = df['Frequency'].rank(method='first')
df['rFrequency'] = pd.qcut(orderFrequencies, 10, labels = False)
df[['rRecency', 'rMonetary']] = df[['Recency', 'Monetary']].apply(lambda x: pd.qcut(x, 10, labels = False))
df['rank'] = (df['rFrequency'] + df['rRecency'] + df['rMonetary'])/3
df['FinalRank'] = df['rank'].apply(int)
```

So we have finished calculating the rank for each customer. Let's see how the customer rank distribution is through histogram with the number of bins = 10.

```
import matplotlib.pyplot as plt

df['rank'].plot.hist(bins = 10)
plt.show()
```



We see that the graph has a normal distribution shape. This shows that the company's customer base will mostly be in the average rank points, such as 4-6. With too high or too low rank points, the lower the number of concentrated customers.

Average rank of variables: A fairly simple way is to average the ranks of the variables to obtain the aggregate rank value for each customer. We can keep each aggregate rank as a group or combine multiple ranks into a group according to the range of values as below:

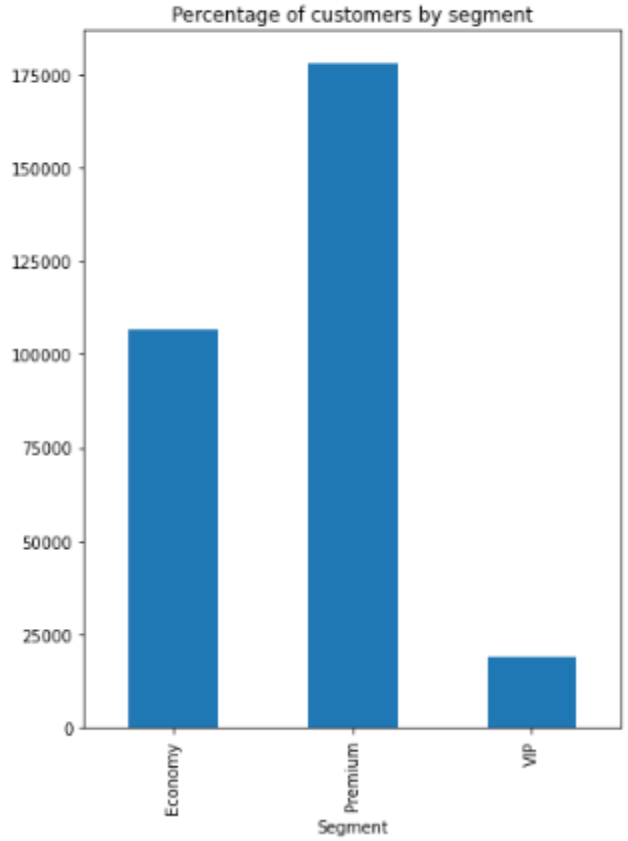
- VIP customers: rank from >= 7 to 10.
- Premium customers: rank from 4 from 7.
- Economy customers: rank from 0 to <= 4.

```
df['Segment'] = 'Economy'
df.loc[(df['rank'] < 7) & (df['rank'] >= 4), 'Segment'] = 'Premium'
df.loc[df['rank'] >= 7, 'Segment'] = 'VIP'
```

To see the structure of the company's customers, we draw a graph for the "Percentage of customers by segment" request.

```
df.groupby('Segment').CustomerID.count().plot.bar(figsize = (6, 8))
plt.title('Percentage of customers by segment')
```

```
Text(0.5, 1.0, 'Percentage of customers by segment')
```

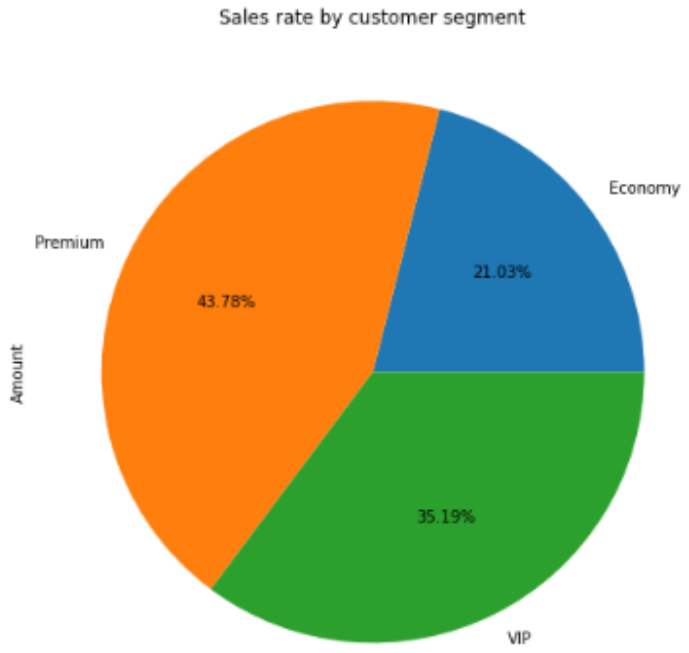


Through the chart, we can see that customers mainly fall into Pemium customers (over 175000 customers); Economy(over 100000 customers); VIP(less 25000 customers). This is an extremely accurate result because online business products are consumer goods. It meets the needs of every citizen.

The results show that the number of VIP customers of the company is very small. The company needs to have a policy to change products to catch up with market tastes or strengthen marketing to attract more consumers.

```
df.groupby('Segment').Amount.sum().plot.pie(autopct = '%.2f%%', figsize = (8, 8))
plt.title('Sales rate by customer segment')
```

```
Text(0.5, 1.0, 'Sales rate by customer segment')
```



According to the pareto principle 20% of customers will bring 80% of sales. Therefore, businesses need to identify the most important customers to take special care of. These customer groups are called VIP, Priority or premium customers, depending on the business has different calling. Dividing customers into different groups based on shopping needs will help businesses do business more efficiently, market to the right customers, and customers will be better served.

Here, economy customers are the customers who bring the biggest revenue to the company, accounting for nearly 80%. The company has yet to reach its goal of 20% of its VIP customers bringing in 80% of the profits.

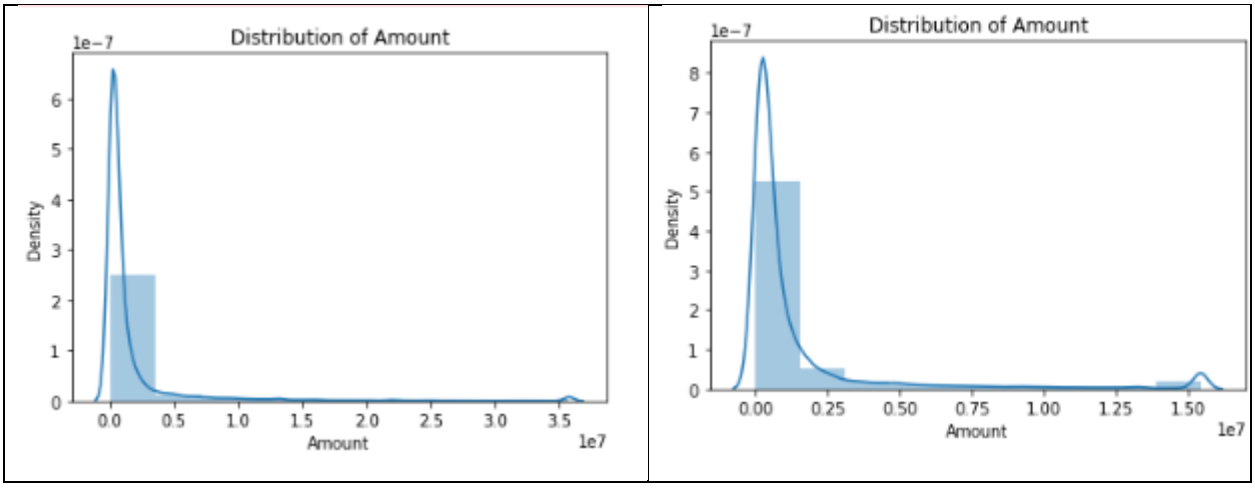
```
# Thống kê mô tả
print(df.describe())

# Vẽ biểu đồ phân phối các biến
import seaborn as sns
import matplotlib.pyplot as plt

def _plotNumeric(colname, n_bins = 10, hist = True, kde = True):
    sns.distplot(df[colname], hist = hist, kde = kde, bins = n_bins)
    plt.title('Distribution of {}'.format(colname))
    plt.show()

_plotNumeric('Amount', hist = True, kde = True, n_bins = 10)
import numpy as np
def _fillOutlier(colname):
    mu = np.mean(df[colname])
    sigma = np.std(df[colname])
    x_min = max(mu - 3*sigma, 0)
    x_max = mu + 3*sigma
    print('x_min: ', x_min)
    print('x_max: ', x_max)
    out_lower_id = df[df[colname] < x_min].index
    out_upper_id = df[df[colname] > x_max].index
    df[colname].loc[out_lower_id] = x_min
    df[colname].loc[out_upper_id] = x_max

_fillOutlier('Amount')
_plotNumeric('Amount', hist = True, kde = True, n_bins = 10)
```



The chart above shows the Amount variation (figure 1). Besides, we can see Amount details at different time intervals (figure 2).

REFERENCES

Slide “HTThanh - RFM analysis for Customer Segmentation - V2” of PhD Ho Trung Thanh