Data Science Methodology

LECTURE 03: DATA SCIENCE CUSTOMER SEGMENTATION USING RFM MODEL

1

INTRODUCTION

- **Different customers have different needs.**
- **With the increase in customer numbers and their variations, it becomes not easy to understand the requirement of each customer.**
- **Identifying potential customers' needs can improve the marketing campaign, which ultimately increases the sales.**
- **Segmentation can play a better role in grouping those customers into various segments.**

2

TOPICS OF TODAY

- **What is Customer Segmentation?**
- **Need of Customer Segmentation**
- **Types of Segmentation**
- **Customer Segmentation using RFM Analysis**
- **Identify Potential Customer Segments using RFM in Python**
- **Conclusion**

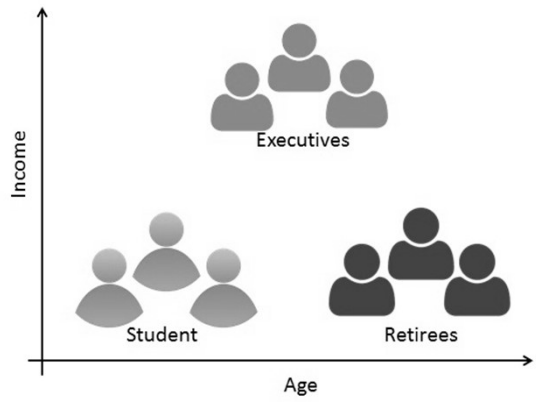
3

WHAT IS CUSTOMER SEGMENTATION?

- ***Customer segmentation is a method of dividing customers into groups or clusters on the basis of common characteristics.***
- **We can segment customers into the B2C model using various customer's demographic characteristics such as occupation, gender, age, location, and marital status.**
- **Psychographic characteristics such as social class, lifestyle and personality characteristics and behavioral characteristics such as spending, consumption habits, product/service usage, and previously purchased products.**

4

EXAMPLE: AGE/INCOME SEGMENTATION



5

NEED OF CUSTOMER SEGMENTATION

- **It helps in identifying the most potential customers.**
- **It helps managers to easily communicate with a targeted groups of the audience.**
- **It improves the quality of service, and customer loyalty, via a better understanding of segments.**
- **It helps managers to design special offers for targeted customers, to encourage them to buy more products.**
- **It also helps in identifying new products that customers could be interested in.**

6

TYPES OF SEGMENTATION

7

CUSTOMER SEGMENTATION USING RFM ANALYSIS

- **RFM (Recency, Frequency, Monetary) analysis is a behavior-based approach grouping customers into segments.**
- **It groups the customers on the basis of their previous purchase transactions.**
- **How recently, how often, and how much did a customer buy.**
- **RFM filters customers into various groups for the purpose of better service.**
- **There may be a segment of customers who are the big spenders but what if they purchased only once! How recently they purchased? Do they often purchase our product?**

8

RFM MODEL

- **Recency (R): Who have purchased recently? Number of days since last purchase (least recency)**
- **Frequency (F): Who has purchased frequently? It means the total number of purchases. (high frequency)**
- **Monetary Value(M): Who has high purchase amount? It means the total money customer spent (high monetary value)**

9

RFM MODEL

- **Here, Each of the three variables(Recency, Frequency, and Monetary) can be divided into four equal quartiles (Q1-Q4), which creates 64 (4x4x4) different customer segments.**

10

STEPS OF RFM(RECENCY, FREQUENCY, MONETARY):

- **Calculate the Recency, Frequency, Monetary values for each customer.**
- **Add segment bin values to RFM table using quartile.**
- **Sort the customer RFM score in ascending order.**

11

EXAMPLE

1.Calculate the Recency, Frequency, Monetary values for each customer.

CustomerID	monetary	frequency	recency
12346.0	325	77183.60	1
12747.0	2	4196.01	103
12748.0	0	33719.73	4596
12749.0	3	4090.88	199
12820.0	3	942.34	59

12

EXAMPLE

2. Add segment bin values to RFM table using quartile.

CustomerID	monetary	frequency	recency	r_quartile	f_quartile	m_quartile
12346.0	325	77183.60	1	1	1	1
12747.0	2	4196.01	103	4	1	4
12748.0	0	33719.73	4596	4	1	4
12749.0	3	4090.88	199	4	1	4
12820.0	3	942.34	59	3	2	4

13

EXAMPLE

- **3. Concatenate all scores in single column (RFM_Score).**

CustomerID	monetary	frequency	recency	r_quartile	f_quartile	m_quartile	RFM_Score
12346.0	325	77183.60	1	1	1	1	111
12747.0	2	4196.01	103	4	1	4	414
12748.0	0	33719.73	4596	4	1	4	414
12749.0	3	4090.88	199	4	1	4	414
12820.0	3	942.34	59	3	2	4	324

14

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- `#import modules`
  ```
  import pandas as pd # for dataframes
  import matplotlib.pyplot as plt # for plotting graphs
  import seaborn as sns # for plotting graphs
  import datetime as dt
  ```
- **Loading Dataset**
- Let's first load the required HR dataset using the pandas read CSV function.

```
data = pd.read_excel("Online_Retail.xlsx")

df=pd.DataFrame(data)
print(df.head()) #first 5 rows
print(df.tail()) #last 5 rows
```

https://www.kaggle.com/code/sarahm/customer-segmentation-using-rfm-analysis

15

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

```
data = pd.read_excel("Online_Retail.xlsx")
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   lower        1816 non-null    object
 3   Description  540455 non-null  object
 4   Quantity     541909 non-null  int64
 5   InvoiceDate  541909 non-null  datetime64[ns]
 6   UnitPrice    541909 non-null  float64
 7   CustomerID   406829 non-null  float64
 8   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 37.2+ MB
```

16

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- data = pd.read_excel("Online_Retail.xlsx")
- data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   lower        1816 non-null    object
 3   Description  540455 non-null  object
 4   Quantity     541909 non-null  int64
 5   InvoiceDate  541909 non-null  datetime64[ns]
 6   UnitPrice    541909 non-null  float64
 7   CustomerID   406829 non-null  float64
 8   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 37.2+ MB
```

17

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

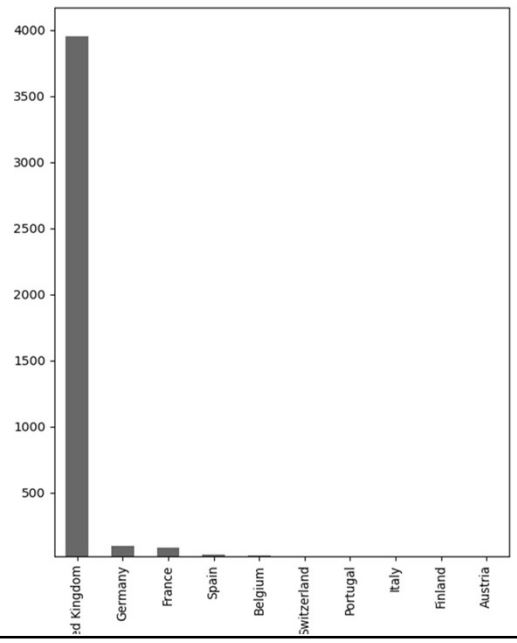
- Removing Duplicates
- Sometimes you get a messy dataset.
- You may want to remove duplicates, which will skew your analysis.
- In python, pandas offer function drop_duplicates(), which drops the repeated or duplicate records.

```
filtered_data=data[['Country','CustomerID']].drop_duplicates()
```

18

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Data Insights**
- `#Top ten country's customer`
  `filtered_data.Country.value_counts( )[:10].plot(kind='bar')`

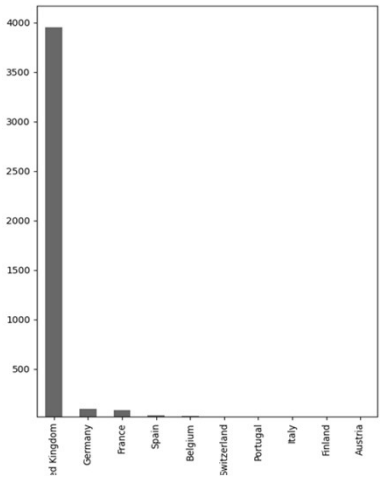


19

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Data Insights**
- **In the shown graph, you can observe most of the customers are from the "United Kingdom". So, we can filter data for United Kingdom customers.**

```
uk_data=data[data.Country=='United Kingdom’]
```



20

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

Data Insights

```
uk_data=data[data.Country=='United Kingdom']
uk_data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 361878 entries, 0 to 541893
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    361878 non-null  object
 1   StockCode    361878 non-null  object
 2   lower        1297 non-null    object
 3   Description  361878 non-null  object
 4   Quantity     361878 non-null  int64
```

21

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

```
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    361878 non-null  object
 1   StockCode    361878 non-null  object
 2   lower        1297 non-null    object
 3   Description  361878 non-null  object
 4   Quantity     361878 non-null  int64
 5   InvoiceDate  361878 non-null  datetime64[ns]
 6   UnitPrice    361878 non-null  float64
 7   CustomerID   361878 non-null  float64
 8   Country      361878 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 27.6+ MB
```

22

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Data Insights**
- **The `nunique()` function in pandas is convenient in getting summary statistics about the uniqueness of the columns. This function returns the count, of unique values in each column of the data.**

```
print(data.nunique())
```

```
InvoiceNo      25900
StockCode       4070
lower            953
Description     4223
Quantity         722
InvoiceDate    23260
UnitPrice       1630
CustomerID      4372
Country           38
dtype: int64
```

23

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Data Insights**
- **The `describe()` function in pandas is convenient in getting various summary statistics. This function returns the count, mean, standard deviation, minimum and maximum values and the quantiles of the data.**

```
print(uk_data.describe())
```

```
       Quantity      UnitPrice     CustomerID
count  361878.000000  361878.000000  361878.000000
mean       11.077029       3.256007   15547.871368
std       263.129266      70.654731    1594.402590
min    -80995.000000       0.000000   12346.000000
25%         2.000000       1.250000   14194.000000
50%         4.000000       1.950000   15514.000000
75%        12.000000       3.750000   16931.000000
max     80995.000000   38970.000000   18287.000000
```

24

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Data Insights**
- **We observed some of the customers have ordered in a negative quantity, which is not possible. So, we need to filter Quantity greater than zero.**

```
uk_data = uk_data[(uk_data['Quantity']>0)]
uk_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 354345 entries, 0 to 541893 Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    354345 non-null  object
 1   StockCode    354345 non-null  object
 2   lower        1285 non-null    object
 3   Description  354345 non-null  object
 4   Quantity     354345 non-null  int64
 5   InvoiceDate  354345 non-null  datetime64[ns]
```

25

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Filter required Columns**

We can filter the necessary columns for RFM analysis. We only need here five columns `CustomerID`, `InvoiceDate`, `InvoiceNo`, `Quantity`, and `UnitPrice`.

`CustomerId` will uniquely define your customers,

`InvoiceDate` help us calculate recency of purchase,

`InvoiceNo` helps us to count the number of time transaction performed(frequency).

`Quantity` purchased in each transaction, and

`UnitPrice` of each unit purchased by the customer will help us calculate the total purchased amount.

26

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Filter required Columns**

```
uk_data=uk_data[['CustomerID','InvoiceDate','InvoiceNo','Quantity',
  'UnitPrice']]
uk_data['TotalPrice'] = uk_data['Quantity'] * uk_data['UnitPrice']
uk_data['InvoiceDate'].min(),uk_data['InvoiceDate'].max()
(Timestamp('2010-12-01 08:26:00'), Timestamp('2011-12-09
  12:49:00'))
PRESENT = dt.datetime(2011,12,10)
uk_data['InvoiceDate'] = pd.to_datetime(uk_data['InvoiceDate'])
print(uk_data.head())
```

27

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Filter required Columns**

```
uk_data=uk_data[['CustomerID','InvoiceDate','InvoiceNo','Quantity','UnitPrice']]
uk_data['TotalPrice'] = uk_data['Quantity'] * uk_data['UnitPrice']
uk_data['InvoiceDate'].min(),uk_data['InvoiceDate'].max()
(Timestamp('2010-12-01 08:26:00'), Timestamp('2011-12-09 12:49:00'))
PRESENT = dt.datetime(2011,12,10)
uk_data['InvoiceDate'] = pd.to_datetime(uk_data['InvoiceDate'])
print(uk_data.head())

dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 27.0+ MB
   CustomerID         InvoiceDate InvoiceNo  Quantity  UnitPrice  TotalPrice
0     17850.0 2010-12-01 08:26:00    536365         6       2.55       15.30
1     17850.0 2010-12-01 08:26:00    536365         6       3.39       20.34
2     17850.0 2010-12-01 08:26:00    536365         8       2.75       22.00
3     17850.0 2010-12-01 08:26:00    536365         6       3.39       20.34
4     17850.0 2010-12-01 08:26:00    536365         6       3.39       20.34
```

28

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **RFM Analysis**
- Here, we are going to perform following operations:
  - *For Recency*, Calculate the number of days between present date and date of last purchase each customer.
  - *For Frequency*, Calculate the number of orders for each customer.
  - *For Monetary*, Calculate sum of purchase price for each customer.

```
rfm= uk_data.groupby('CustomerID').agg({'InvoiceDate': lambda date:
(PRESENT - date.max()).days, 'InvoiceNo': lambda num: len(num),
'TotalPrice': lambda price: price.sum()})
```

29

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **RFM Analysis**

```
print(rfm.head())
```

CustomerID	InvoiceDate	InvoiceNo	TotalPrice
12346.0	325	1	77183.60
12747.0	2	103	4196.01
12748.0	0	4596	33719.73
12749.0	3	199	4090.88
12820.0	3	59	942.34

30

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **RFM Analysis**

```
# Change the name of columns
rfm.columns =['recency','frequency', 'monetary']
rfm['recency'] = rfm['recency'].astype(int)

print(rfm.head())
```

```
             recency  frequency monetary
CustomerID
12346.0          325          1    77183
12747.0            2        103     4196
12748.0            0       4596    33719
12749.0            3        199     4090
12820.0            3         59      942
```

31

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Computing Quartile of RFM values**

- **Customers with the lowest recency, highest frequency and monetary amounts considered as top customers.**
- `qcut()` **is Quartile-based discretization function. qcut bins the data based on sample quartiles.**
- **For example, 1000 values for 4 quartiles would produce a categorical object indicating quartile membership for each customer.**

32

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **Computing Quartiles of RFM values**

```
• rfm['r_quartile'] = pd.qcut(rfm['recency'], 4, ['1','2','3','4'])
• rfm['f_quartile'] = pd.qcut(rfm['frequency'], 4, ['4','3','2','1'])
• rfm['m_quartile'] = pd.qcut(rfm['monetary'], 4, ['4','3','2','1’])
• print(rfm.head())
```

CustomerID	recency	frequency	monetary	r_quartile	f_quartile	m_quartile
12346.0	325	1	77183	4	4	1
12747.0	2	103	4196	1	1	1
12748.0	0	4596	33719	1	1	1
12749.0	3	199	4090	1	1	1
12820.0	3	59	942	1	2	2

33

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **RFM result interpretations**
- concatenate all three quartiles(r_quartile,f_quartile,m_quartile) in a single column, this rank will help you to segment the customers well group.

```
• rfm['RFM_Score'] = rfm.r_quartile.astype(str)+
  rfm.f_quartile.astype(str) + rfm.m_quartile.astype(str)
  print(rfm.head())
```

CustomerID	recency	frequency	monetary	r_quartile	f_quartile	m_quartile	RFM_Score
12346.0	325	1	77183	4	4	1	441
12747.0	2	103	4196	1	1	1	111
12748.0	0	4596	33719	1	1	1	111
12749.0	3	199	4090	1	1	1	111
12820.0	3	59	942	1	2	2	122

34

IDENTIFY POTENTIAL CUSTOMER SEGMENTS USING RFM IN PYTHON

- **RFM result interpretations**
- We need to see the highest priority customers (RFM_Score=111). To do that,
- `# Filter out Top/Best customers`

```
print(rfm[rfm['RFM_Score']=='111'].sort_values('monetary',ascending
=False).head())
print(rfm.columns)
```

```
CustomerID recency  frequency   monetary r_quartile f_quartile m_quartile   RFM_Score
18102            0        431  259657.30          1          1          1         111
17450            8        337  194550.79          1          1          1         111
17511            2        963   91062.38          1          1          1         111
16684            4        277   66653.56          1          1          1         111
14096            4       5111   65164.79          1          1          1         111
Index(['recency', 'frequency', 'monetary', 'r_quartile', 'f_quartile', 'm_quartile',
'RFM_Score'],dtype='object')
```

35

RFM MODEL CONCLUSION

- We have learned customer segmentation, Need of Customer Segmentation, and Types of Segmentation.
- We have focused on Customer segmenation via RFM model.
- We have learned RFM implementation via Python. Also, we have seen some basic concepts of pandas module such as handling duplicates, groupby, and qcut() for bins based on sample quartiles.

36

ASSIGNMENT 1

- **Find another data set, with at least 50,000 records**
- **What else does the Lambda function for aggregation in the Pandas module provide?**
- **Give examples of its use on the downloaded data set you found**
- **Perform RFM model on it**

37