PYTHON DATA Preparation & Visualization

Lesson 4: Case Study - RFM Customer Segmentation

Lecturer: **Dr. Nguyen Tuan Long**

Email: ntlong@neu.edu.vn

Mobile: 0982 746 235

CONTENTS

1. **The Business Problem:**
   - Why do we need to segment customers?
   - What is customer segmentation and why is it one of the most critical marketing strategies?
2. **Introduction to the RFM Model**
   - What are Recency, Frequency, and Monetary?
   - Decoding the three core aspects of customer behavior.
3. **End-to-End Hands-on Practice**
   - Loading and cleaning the data.
   - Calculating R, F, and M metrics.
   - Scoring and segmenting customers.
   - Analyzing segments and proposing specific business actions.

The Business Problem: Not All Customers Are Created Equal

- **The Issue:** A retail company has thousands of customers. Applying the same marketing strategy to everyone is inefficient and costly. The 80/20 rule often applies: a small fraction of customers typically generates the majority of revenue.
  - Sending discount emails to "VIP" customers might reduce profit margins, as they are willing to buy at full price.
  - Neglecting new customers might cause them to never return, wasting the initial customer acquisition cost.
  - Forgetting about old customers can lead them to switch to competitors, increasing the churn rate.

- **The Goal:** We need a method to **classify customers** into groups with similar behaviors, which allows us to devise tailored engagement and marketing strategies. This helps optimize marketing spend, increase customer loyalty, and maximize revenue.

The Solution: RFM Analysis

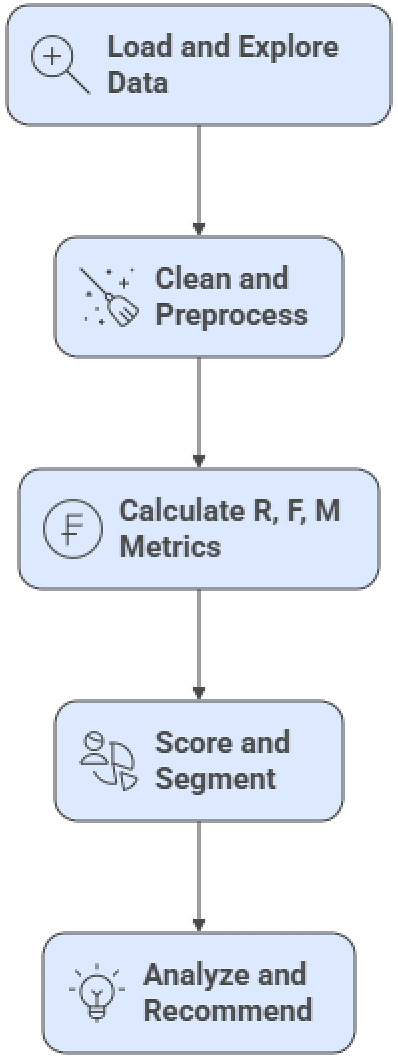
- **What is RFM?** It is a segmentation model based on customers' **purchasing behavior**, answering three critical questions:
    - **Recency:** How recently did a customer make a purchase? -> *More recent customers are more likely to respond.* A customer who shopped last week is more valuable than one who hasn't bought anything in a year.
    - **Frequency:** How often do they purchase? -> *More frequent customers are more loyal.* A customer who buys 10 times a year shows higher engagement than a one-time buyer.
    - **Monetary Value:** How much money have they spent in total? -> *Customers who spend more are the most valuable.* A customer who has spent $1,000 is more valuable than one who has only spent $10.
- **The Power of RFM:** It is simple, intuitive, easy to calculate, and highly effective at identifying valuable customer segments. RFM doesn't require complex algorithms but provides extremely useful insights for business decision-making.

Roadmap

5 steps:

1. **Load and Explore Data:** Understand the OnlineRetail.csv dataset. This is the foundational step to identify potential issues.
2. **Clean and Preprocess:** Handle missing values and incorrect formats. This is the most critical step to ensure the accuracy of the analysis.
3. **Calculate R, F, M Metrics:** Apply groupby and aggregation skills to transform transaction data into customer behavior data.
4. **Score and Segment:** Use quantiles to divide customers into score groups. This is the "standardization" step that allows for customer comparison.
5. **Analyze and Recommend:** Derive insights from the segments and propose specific business actions. This is the final step to turn analysis into value.

RFM Analysis Process



Step 1 - Load and Explore Data

Objective: Get familiar with the online retail dataset (link) and identify necessary cleaning steps.

Analysis:

- CustomerID has many missing values -> Needs to be handled. We cannot analyze the behavior of anonymous customers.
- InvoiceDate is an object type -> Needs to be converted to datetime to perform time-based calculations.
- Quantity and UnitPrice have negative values -> Needs investigation (likely returned orders). We need to remove them so they don't affect the total spending calculation.

Step 2 - Clean and Preprocess Data

Objective: Prepare a "clean" dataset for analysis.

```
# Drop rows with missing CustomerID
df.dropna(subset=['CustomerID'], inplace=True)

# Remove returned orders (Quantity < 0)
df = df[df['Quantity'] > 0]

# Convert data types
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df['CustomerID'] = df['CustomerID'].astype(int).astype(str)

# Create TotalPrice column
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']
```

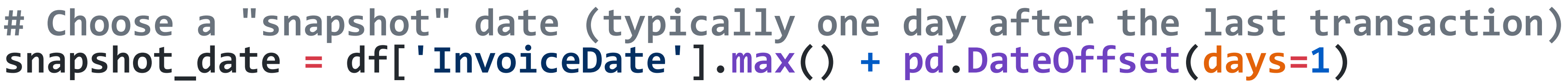
Explanation: dropna(subset=['CustomerID']) is a critical decision. We accept the loss of some transaction data to be able to perform analysis at the customer level. inplace=True modifies the DataFrame directly without needing reassignment.

Step 3 - Calculate R, F, M Metrics

Objective: Apply learned skills to calculate the three core metrics.

Logic:

- **Recency:** Get the last purchase date for each customer and calculate the difference to a "snapshot" date. The snapshot date is usually chosen as one day after the last transaction in the data to ensure all Recency values are positive.



```python
# Choose a "snapshot" date (typically one day after the last transaction)
snapshot_date = df['InvoiceDate'].max() + pd.DateOffset(days=1)
```

- **Frequency:** Count the number of unique invoices for each customer. Using nunique() on the InvoiceNo column instead of count() prevents counting the same order multiple times if it contains multiple products.
- **Monetary:** Calculate the sum of TotalPrice for each customer.

Calculating R, F, M

```
# Calculate R, F, M
rfm = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda date: (snapshot_date - date.max()).days,
    'InvoiceNo': 'nunique',
    'TotalPrice': 'sum'
})

# Rename columns for clarity
rfm.rename(columns={'InvoiceDate': 'Recency',
                    'InvoiceNo': 'Frequency',
                    'TotalPrice': 'Monetary'}, inplace=True)

rfm.head()
```

Explanation: This is a powerful application of the .agg() method. We can apply different functions to different columns within the same groupby command, making the code very concise and efficient.

Step 4 - Scoring and Segmentation

- **Objective:** Convert the R, F, and M values into scores from 1 to 5 for easy comparison and combination.
- **Method:** Using **Quantiles**. We will divide customers into 5 equal-sized groups for each metric. This ensures that each score group has a relatively similar number of customers.
- **Recency:** Score 5 for the most recent group, score 1 for the least recent.
- **Frequency & Monetary:** Score 5 for the highest frequency/spending group, score 1 for the lowest.

```
# Create score labels
r_labels = range(5, 0, -1) # 5 is best
f_labels = range(1, 6) # 5 is best
m_labels = range(1, 6) # 5 is best

# Assign scores
rfm['R_score'] = pd.qcut(rfm['Recency'], q=5,
labels=r_labels)
rfm['F_score'] =
pd.qcut(rfm['Frequency'].rank(method='first'), q=5,
labels=f_labels)
rfm['M_score'] = pd.qcut(rfm['Monetary'], q=5,
labels=m_labels)
```

Note: rank(method='first') is used for Frequency to handle cases where many customers have the same frequency. Without this step, qcut might fail if there are too many duplicate values at the quantile boundaries.

Combining Scores and Creating Segments

Objective: Create a combined RFM score and assign segment labels.

```
# Create a combined RFM score string
rfm['RFM_Score'] = rfm['R_score'].astype(str) +
rfm['F_score'].astype(str) + rfm['M_score'].astype(str)

# Define segments based on R and F scores
segment_map = {
    r'[1-2][1-2]': 'Hibernating',
    r'[1-2][3-4]': 'At-Risk',
    r'[1-2]5': 'Cannot Lose Them',
    r'3[1-2]': 'About to Sleep',
    r'33': 'Need Attention',
    r'[3-4][4-5]': 'Loyal Customers',
    r'41': 'Promising',
    r'51': 'New Customers',
    r'[4-5][2-3]': 'Potential Loyalists',
    r'5[4-5]': 'Champions'
}

# Assign segment labels
rfm['Segment'] = rfm['R_score'].astype(str) +
rfm['F_score'].astype(str)
rfm['Segment'] = rfm['Segment'].replace(segment_map,
regex=True)
rfm.head()
```

Explanation: We use Regular Expressions (regex) to assign labels to customer groups based on their R and F scores. This is a common and effective approach that focuses on the two most important factors: Recency and Frequency.

Step 5 - Analysis and Recommendations

Objective: Derive insights from the newly created segments.

```
# Analyze the characteristics of each segment
segment_analysis =
rfm.groupby('Segment').agg({
    'Recency': 'mean',
    'Frequency': 'mean',
    'Monetary': ['mean', 'count']
}).round(1)

segment_analysis
```

```
# Plot the number of customers in each segment
segment_analysis['Monetary']['count'].plot(kind='
barh', figsize=(10, 6));
```

```
# Plot Mean of Monetary in each segment
segment_analysis['Monetary']['mean'].plot(kind='barh', figsize=(10, 6));
```

Analysis: The segment_analysis table clearly shows the average characteristics of each group. For example, the 'Champions' segment has the lowest Recency (most recent purchases) and the highest Frequency and Monetary values. Conversely, the 'Hibernating' segment has a very high Recency (has not purchased in a long time) and low Frequency/Monetary values.

Proposing Business Actions

Based on the analysis, we can recommend:

- **Champions (555, 554...):** Your best customers.
  - *Action:* Grant exclusive perks, invite them to a loyalty program, ask for product reviews. Avoid sending discounts as they are willing to buy at full price.
- **Loyal Customers (344, 455...):** Faithful customers.
  - *Action:* Send thank-you emails, introduce new products, upsell/cross-sell. They are the ideal group for testing new products.
- **At-Risk (134, 244...):** Past frequent customers who haven't returned in a while.
  - *Action:* Send "We miss you!" campaigns with special offers to win them back. Act quickly before they are lost completely.
- **Hibernating (111, 122...):** "Sleeping" customers, almost lost.
  - *Action:* May not require significant investment. Send low-cost marketing campaigns to see if they can be "reawakened."

Summary

Knowledge Recap:

- The end-to-end RFM analysis process.
- Applying Pandas skills (cleaning, groupby, agg, qcut) to a real-world problem.
- How to turn analytical results into actionable business recommendations.

Practice

Based on the created RFM DataFrame:

1. Calculate the percentage of customers in each segment.
2. Conduct a deeper analysis of the 'Champions' segment: What products do they typically buy? (Requires merging back with the original DataFrame).
3. Experiment with a different number of score groups (e.g., 4 instead of 5) and see how the segment distribution changes.

Q&A