

Customer Segmentation Using RFM Analysis - Documentation

1. Project Overview

This project focuses on performing customer segmentation based on transactional data from an online retail store. The segmentation is achieved through the use of RFM (Recency, Frequency, and Monetary) analysis, a technique commonly used in marketing analytics to assess customer value and behavior.

- Data Source: The dataset contains transactions from an online retail company between December 2009 and December 2011.
- Goal: The main objective is to segment customers based on how recently and frequently they purchased, and how much they spent, to provide valuable insights for targeted marketing strategies.

2. Methodology

The project is structured around several key steps:

1. Data Cleaning: Handling missing values, filtering for valid transactions, and preparing the dataset for analysis.
2. Feature Engineering: Creating relevant features such as total revenue per transaction and calculating the key RFM metrics.
3. RFM Analysis: Grouping customers based on Recency, Frequency, and Monetary values.
4. Customer Segmentation: Using the RFM scores to classify customers into meaningful segments for marketing actions.
5. Visualization: Plotting and interpreting the distribution of customer segments.

3. Data Cleaning and Preparation

The dataset required cleaning to ensure that the analysis was accurate. Key steps taken include:

- Removing Missing Values: Rows with missing 'Customer ID' were dropped since

they are crucial for identifying unique customers.

- Filtering Transactions: Transactions with negative quantities or prices (representing returns or errors) were filtered out.

Example of how this was done in code:

```
# Remove rows where 'Customer ID' is missing
```

```
cleaned_data = data.dropna(subset=['Customer ID'])
```

```
# Ensure there are no negative or zero values for Quantity and Price
```

```
cleaned_data = cleaned_data[(cleaned_data['Quantity'] > 0) & (cleaned_data['Price'] > 0)]
```

4. Feature Engineering

A new feature, `TotalRevenue`, was added to the dataset to capture the total transaction value for each purchase. This was calculated by multiplying the `Quantity` of items by the `Price` per item.

```
# Add 'TotalRevenue' column
```

```
cleaned_data['TotalRevenue'] = cleaned_data['Quantity'] * cleaned_data['Price']
```

5. RFM Analysis

RFM Analysis involves three key metrics:

- Recency: How recently a customer made their last purchase.
- Frequency: How often the customer makes purchases.
- Monetary: The total revenue generated by the customer.

The following code calculates these metrics by grouping the dataset by `Customer ID`:

```
# Group by Customer ID and calculate Recency, Frequency, and Monetary
```

```
rfm_table = cleaned_data.groupby('Customer ID').agg({
```

```
    'InvoiceDate': lambda x: (reference_date - x.max()).days, # Recency
```

```
    'Invoice': 'nunique', # Frequency
```

```
'TotalRevenue': 'sum' # Monetary
}).reset_index()
```

```
# Rename columns to 'Recency', 'Frequency', 'Monetary'
rfm_table.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']
```

6. Customer Segmentation

To classify customers, RFM quartiles were created for each metric, which allowed the segmentation of customers into different categories based on their behavior.

```
# Create quartiles for Recency, Frequency, and Monetary
rfm_table['R_Quartile'] = pd.qcut(rfm_table['Recency'], 4, labels=[4, 3, 2, 1])
rfm_table['F_Quartile'] = pd.qcut(rfm_table['Frequency'], 4, labels=[1, 2, 3, 4])
rfm_table['M_Quartile'] = pd.qcut(rfm_table['Monetary'], 4, labels=[1, 2, 3, 4])
```

```
# Combine R, F, and M quartiles into a single RFM score
rfm_table['RFM_Score'] = rfm_table['R_Quartile'].astype(str) +
rfm_table['F_Quartile'].astype(str) + rfm_table['M_Quartile'].astype(str)
```

7. Visualization

The distribution of RFM scores can be visualized to understand the spread of customers across different segments. This is useful for marketing teams to identify which groups need attention.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Distribution of RFM Score
plt.figure(figsize=(10,6))
sns.countplot(x='RFM_Score', data=rfm_table)
plt.title('Distribution of RFM Scores')
plt.show()
```

8. Conclusion

RFM analysis allows us to:

- Identify high-value customers who are frequent, recent, and high spenders.
- Discover at-risk customers who used to spend a lot but haven't made a recent purchase.
- Target potential loyalists who buy frequently but haven't yet spent large amounts.

This segmentation helps businesses create targeted marketing strategies, boost customer retention, and increase revenue.

9. Future Enhancements

- Automated Reporting: Building dashboards for dynamic RFM segmentation insights.
- Clustering: Using techniques like K-Means clustering to further refine customer segments.
- Predictive Analytics: Applying machine learning models to predict customer lifetime value based on RFM scores.