Olist E-Commerce Customer Segmentation

Meera Sanjeevirao

2025-07-12

Contents

1. Introduction and Data Preparation

This report details the customer segmentation for the Olist e-commerce platform. Olist is a Brazilian technology company that functions as an e-commerce aggregator, providing small and medium-sized enterprises with a centralized platform to sell on major online marketplaces.

K-Means clustering is utilized in R to identify distinct customer groups based on their RFM (Recency, Frequency, Monetary) behavior.

The foundation of this analysis is on the processed `rfm_data.csv` file, which was generated by the `calculate_rfm.sql` script. This script transformed raw transactional data by joining customer, order, and payment tables to compute the core RFM metrics for each unique customer:

- Recency (R): The number of days between the customer's most recent order and the latest transaction date in the dataset.
- Frequency (F): The total count of completed orders for each customer.
- Monetary (M): The total sum of all payments made by each customer.

Once the aggregated RFM data is loaded into R, further preparation is needed. Because RFM distributions are often skewed, we apply a log transformation. The data is then scaled to give each metric equal importance during the clustering process, as the K-Means algorithm is sensitive to variations in feature scales.

```
# Load the dataset created from the SQL query
rfm_data <- read.csv("../data/processed/rfm_data.csv")

# Log-transform and scale the data for clustering
rfm_prepared <- rfm_data %>%
  # Add 1 to avoid log(0) issues for customers with 0 recency.
```

```r
  mutate(
    log_recency = log(recency + 1),
    log_frequency = log(frequency + 1),
    log_monetary = log(monetary + 1)
  )

rfm_scaled <- scale(rfm_prepared[, c("log_recency", "log_frequency", "log_monetary")])
```

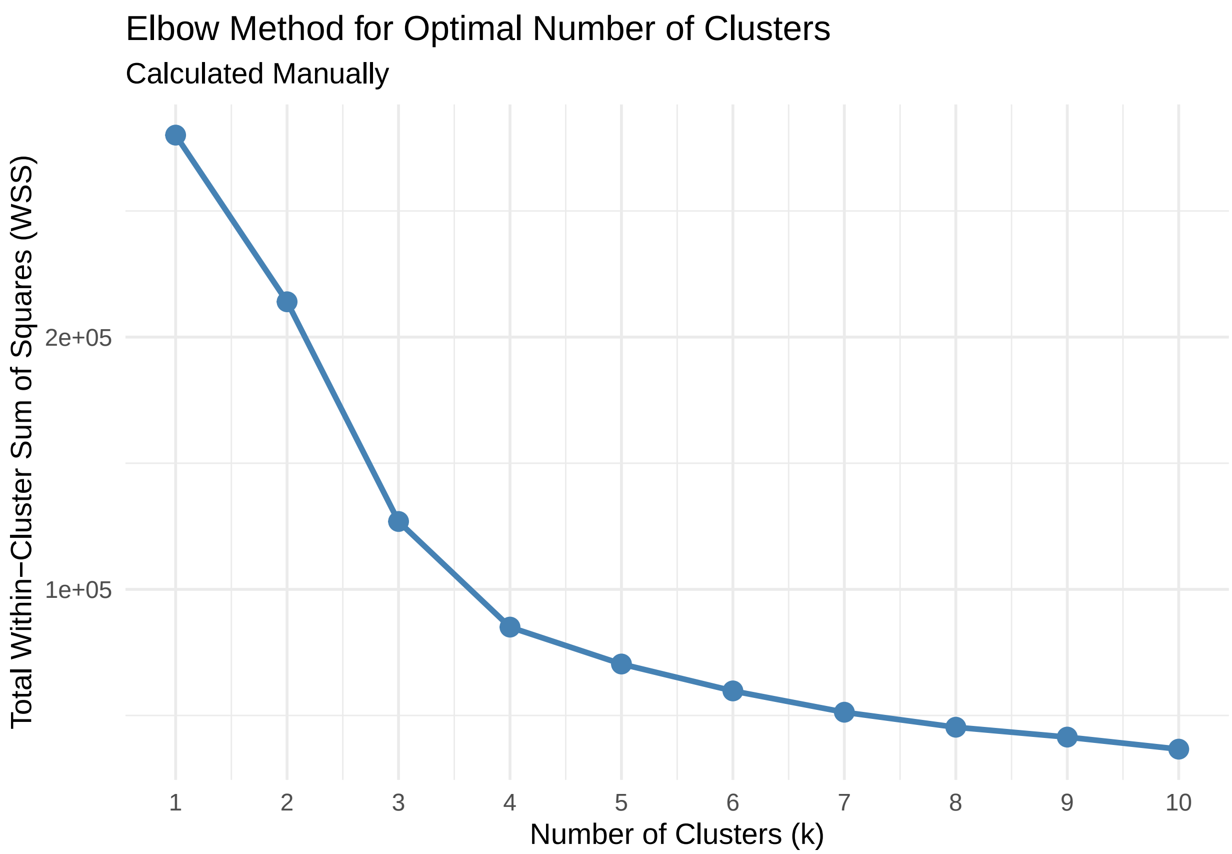
2. K-Means Clustering

Determining Optimal Clusters

To find the optimal number of segments, we use the Elbow Method. This technique calculates the Total Within-Cluster Sum of Squares (WSS) for different cluster counts (k). We are looking for the "elbow" on the plot—the point where the WSS begins to decrease at a much slower rate. This point indicates the most appropriate number of clusters.

The plot below shows a distinct elbow at k=4, making 4 clusters a reasonable choice for this analysis.

```r
# Calculate the Within-Cluster Sum of Squares (WSS) manually
# Completed for a range of k values to create our own Elbow Plot.

# Function to compute total WSS for a given k
calculate_wss <- function(k, data) {
  kmeans(data, k, nstart = 25)$tot.withinss
}

# Set the range of clusters to test (e.g., 1 to 10)
k_values <- 1:10

# Calculate WSS for each k value
# This loop is memory efficient as it processes one k at a time
wss_values <- sapply(k_values, calculate_wss, data = rfm_scaled)

# Create a data frame for plotting
elbow_data <- data.frame(
  clusters = k_values,
  wss = wss_values
)

# Plot the elbow curve using ggplot2
ggplot(elbow_data, aes(x = clusters, y = wss)) +
  geom_line(linewidth = 1, color = "steelblue") +
  geom_point(size = 3, color = "steelblue") +
  labs(
    title = "Elbow Method for Optimal Number of Clusters",
    subtitle = "Calculated Manually",
    x = "Number of Clusters (k)",
    y = "Total Within-Cluster Sum of Squares (WSS)"
  ) +
  scale_x_continuous(breaks = k_values) +
  theme_minimal()
```



Running the Algorithm

We now run the K-Means algorithm with 4 centers and add the cluster assignments back to our data for analysis.

```r
set.seed(123) # for reproducibility
kmeans_result <- kmeans(rfm_scaled, centers = 4, nstart = 25)
rfm_prepared$cluster <- as.factor(kmeans_result$cluster)
```

3. Cluster Analysis & Interpretation

To understand what each cluster represents, we calculate the average RFM values for each group and assign a descriptive persona.

```r
# Calculate the mean RFM values for each cluster
cluster_summary <- rfm_prepared %>%
  group_by(cluster) %>%
  summarise(
    avg_recency = mean(recency),
    avg_frequency = mean(frequency),
    avg_monetary = mean(monetary),
    customer_count = n()
  ) %>%
  arrange(desc(avg_monetary))
```

```r
# Assign meaningful persona names based on the characteristics
cluster_summary <- cluster_summary %>%
  mutate(persona = case_when(
    # Catches multi-purchase customers who are now less recent.
    avg_frequency > 1.5 & avg_monetary > 200   ~ "Loyal Customers",

    # Catches high-value, single-purchase customers who are now inactive.
    avg_recency > 250 & avg_monetary > 200   ~ "Hibernating High Spenders",

    # Catches recent, lower-value, single-purchase customers.
    avg_recency < 150                         ~ "New Customers",

    # Catches all other single-purchase, non-recent, low-value customers.
    TRUE                                      ~ "At-Risk Low Spenders"
  ))

# Display the summary table
kable(cluster_summary, caption = "Customer Segment Characteristics")
```

Table 1: Customer Segment Characteristics

cluster	avg_recency	avg_frequency	avg_monetary	customer_count	persona
1	332.9702	1.000000	318.43528	27955	Hibernating High Spenders
2	268.4495	2.113888	308.58879	2801	Loyal Customers
3	114.7623	1.000000	122.69970	24669	New Customers
4	364.2912	1.000000	69.31792	37932	At-Risk Low Spenders

4. Visualizing the Segments

Visualizations help illustrate the distinct characteristics of our segments.

RFM Segment Scatter Plot

This plot visualizes the distinct characteristics of our four segments. The x-axis (Recency) is on a log scale to better distribute the customers.

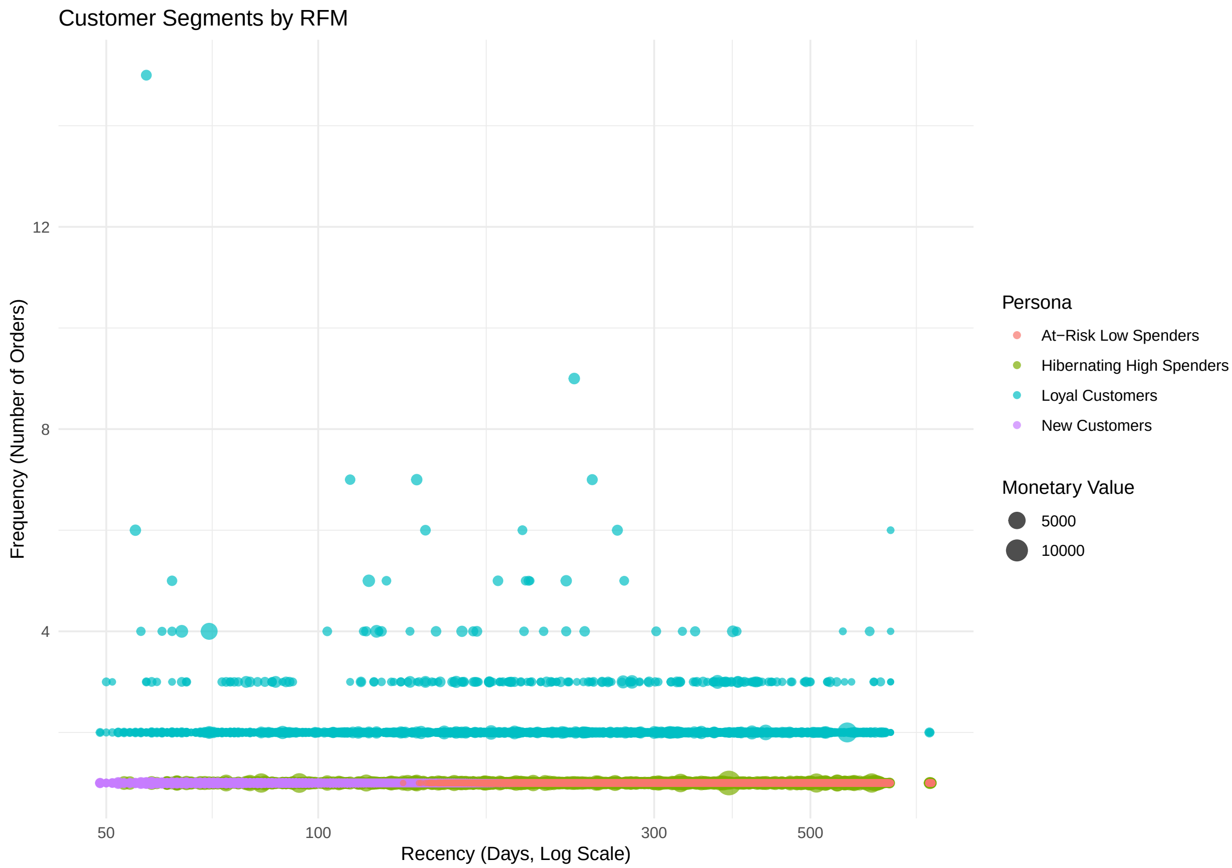
We can see a clear separation:

Loyal Customers and **New Customers** are on the right side of the plot, indicating low recency (they've purchased recently). They are differentiated by their frequency and spending. **Hibernating High Spenders** and **At-Risk Low Spenders** are on the left (high recency).

The key difference between them is their historical spending, which is represented by the size of the points.

```r
# Add the persona names back to the main data frame for plotting
rfm_prepared <- rfm_prepared %>%
  left_join(cluster_summary %>% select(cluster, persona), by = "cluster")

# Create the scatter plot
ggplot(rfm_prepared, aes(x = recency, y = frequency, color = persona, size = monetary)) +
  geom_point(alpha = 0.7) +
```

```r
  scale_x_continuous(trans = 'log10') +
  labs(
    title = "Customer Segments by RFM",
    x = "Recency (Days, Log Scale)",
    y = "Frequency (Number of Orders)",
    color = "Persona",
    size = "Monetary Value"
  ) +
  theme_minimal()
```




```r
# Export final, processed data into CSV file for Tableau use
# Create the 'output' directory if it doesn't already exist
if (!dir.exists("output")) {
  dir.create("output")
}

# Now, safely write the CSV file into that directory
write.csv(
  rfm_prepared,
  "output/olist_tableau_data.csv",
  row.names = FALSE
)
```

5. Conclusion & Recommendations

The K-Means analysis successfully segmented Olist customers into four distinct and actionable groups. By replacing generic labels with personas that reflect the true nature of the clusters, we can develop more precise marketing strategies:

Loyal Customers: These are repeat buyers with high frequency and good monetary value. They are the backbone of the business, but haven't purchased as recently as new customers.

- Action: Target with loyalty programs, exclusive access to new products, and personalized "thank you" offers to reward their business and keep them engaged.

Hibernating High Spenders: This group represents customers who have spent a significant amount in the past but have not purchased in a long time. They are a high-value, high-risk segment with huge potential if won back.

- Action: Launch a targeted win-back campaign. Use personalized emails reminding them of past purchases and offer a compelling discount to encourage their return.

New Customers: This segment consists of recent, first-time buyers who have spent a modest amount. The primary goal is to nurture them into becoming loyal customers.

- Action: Encourage a second purchase through a welcome series, targeted follow-up promotions, and product recommendations based on their initial order.

At-Risk Low Spenders: This is the largest group, containing customers who made a single, low-value purchase a long time ago and have not returned.

- Action: This group has a low probability of converting. It is not cost-effective to spend significant marketing resources here. A low-touch, automated email campaign is sufficient.

This data-driven segmentation provides a strong foundation for developing targeted marketing strategies to improve customer retention and maximize lifetime value.

6. Sources & Further Reading

This section provides resources for readers interested in a deeper understanding of the analytical techniques used in this report.

RFM Analysis RFM is a classic marketing analysis technique used to identify a company's best customers by measuring their transactional behavior.

- **RFM Analysis: A Data-Driven Approach to Customer Segmentation** - A practical guide from HubSpot on how to calculate and apply RFM for customer segmentation.
- **What is RFM Analysis?** - An in-depth article from CleverTap explaining the components of RFM and how to use scoring to create actionable segments.

Cluster Analysis Cluster analysis is a broad set of techniques for finding natural groupings or "clusters" within a dataset without any prior labels.

- **Lesson 14: Cluster Analysis** - Penn State statistical overview of cluster analysis methods and their application.

K-Means Clustering K-Means is a popular and efficient algorithm that partitions data into a pre-specified number (k) of clusters. It aims to make the data points within a cluster as similar as possible while making the clusters themselves as distinct as possible.

- **What is k-means clustering?** - An IBM explanation of how the K-Means algorithm works, including its iterative process of assigning data points and updating cluster centers.
- **K-Means Clustering | Definition, Algorithm & How-To Guide** - A detailed walkthrough of the K-Means algorithm steps with visual examples.

Elbow Method The Elbow Method is a heuristic used to help determine the optimal number of clusters for the K-Means algorithm. It works by plotting the within-cluster sum of squares (WCSS) for a range of cluster counts and identifying the "elbow" of the curve.

- **Elbow Method in K-Means Clustering** - A guide that defines the Elbow Method, explains the WCSS metric, and discusses its application and drawbacks.