Data Science Project

Customer Value Maximization Pipeline with Segmentation

Hannan Baig

Executive Summary: Customer Segmentation and Value Maximization

Objective and Core Problem

This project was initiated to move beyond a costly "one-size-fits-all" marketing strategy, which leads to significant budget waste and high customer churn risk. The core problem addressed is the lack of specific customer profiles, preventing the targeted allocation of resources. The goal was to transform raw transactional data into actionable, high-value customer segments that directly inform sales strategy, retention efforts, and marketing expenditure.

Solution and Key Findings

A complete data science pipeline was created, starting with feature engineering to develop the RFM (Recency, Frequency, Monetary Value) model for 500 customers. To ensure robustness, three unsupervised learning algorithms were tested: K-Means Clustering, Hierarchical Clustering, and DBSCAN. The 3-cluster K-Means model was selected as the final segmentation strategy, offering the optimal balance of mathematical stability and business interpretability.

The analysis revealed three distinct and critical customer segments:

KMeans Cluster	Count	Mean Monetary Value	Segment Label	Business Action
0	235 (47%)	$9,028.65	Loyal High Spenders	Focus on retention, cross-selling high-margin items.
2	111 (22%)	$5,141.97	Recent Mid-Tier Buyers	Upselling and migration to Cluster 0.
1	154 (31%)	$2,903.19	At-Risk Low Spenders	Targeted discounts and win-back campaigns.

Table 1: Customer Segmentation based on KMeans Clustering

Business Impact and Recommendations

Financial and Strategic Impact

1. **Marketing Efficiency:** The analysis identifies that 47% of the customer base (Loyal Spenders) should not be targeted with costly blanket discounts. This allows for budget reallocation toward the high-risk, high-return At-Risk segment.

2. **Risk Mitigation:** Hierarchical Clustering revealed a large group of high-value customers drifting away (Recency > 80 days), signaling an immediate need for reactivation efforts.

Actionable Recommendations

- **Loyal High Spenders:** Implement a VIP-only program offering early access, premium service, and non-monetary rewards. Avoid discount-based incentives.

- **Recent Mid-Tier Buyers:** Prioritize upselling and cross-selling using recommendation engines to increase AOV and elevate them toward the Loyal Spenders segment.

- **At-Risk Low Spenders:** Immediately deploy targeted win-back campaigns featuring personalized high-value discounts and compelling offers.

Contents

1 Data Generation **5**

2 Statistical Overview **6**
- 2.1 Quantity Analysis . . . 6
- 2.2 Unit Price (Cost per Item) . . . 6
- 2.3 Total Price Analysis . . . 7
- 2.4 Overall Business Implication . . . 7

3 Ensuring Data Integrity **8**
- 3.1 Duplicate Check . . . 8
- 3.2 Invalid Transaction Check . . . 8

4 Outlier Detection through Interquartile Range (IQR) **9**
- 4.1 Quantity Analysis . . . 9
- 4.2 Unit Price Analysis . . . 10
- 4.3 Total Price Analysis . . . 10
- 4.4 Interpreting the Impact of the Outliers . . . 10

5 RFM Calculation and Their Correlation **11**
- 5.1 Summary Statistics . . . 11

6 Correlation – Finding the Hidden Connections **13**

7 RFM – Exploratory Data Analysis and Feature Scaling **15**
- 7.1 Histograms and Distribution Analysis . . . 15
- 7.2 Critical Analysis of Skewness . . . 16
- 7.3 Solution: Log Transformation . . . 16
- 7.4 Standardization of Log-Transformed Variables . . . 17
- 7.5 Interpretation of Scaled RFM Values . . . 18
- 7.6 Validation of Standard Scaling . . . 18
- 7.7 Correlation Heatmap Findings . . . 20

8 Clustering **21**
- 8.1 K-Means Clustering . . . 21
  - 8.1.1 Step 1: Determining the Optimal Number of Clusters (K) . . . 22
  - 8.1.2 Step 2: Applying the K-Means Model . . . 24
  - 8.1.3 Step 3: Analyzing the Customer Segments . . . 24
  - 8.1.4 Conclusion . . . 25

9 Clustering **26**
9.1 Hierarchical Clustering 26
9.2 Hierarchical Clustering Algorithm Concept 26
9.3 Customer Tree – The Dendrogram Visualization 27
9.4 Understanding the Dendrogram 27
9.5 Determining the Number of Clusters 28

10 Applying the Hierarchical Model (k = 4) **29**

11 Critical Business Impact & Strategy **29**

12 Visualizing the 4 Clusters **31**
12.1 Frequency vs Monetary Value 31
12.2 Recency vs Monetary Value 31

13 Clustering **32**
13.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise) . . 32

14 Comparing All Three Clustering Algorithms **37**
14.1 Contingency Table Analysis 37
14.2 Deep Dive into the 49 DBSCAN Noise Customers 38
14.3 Key Insight 38
14.4 Top 10 Noise Customers by Monetary Value 39
14.5 Business Impact 39

15 Final Model Selection and Segmentation **40**

16 Conclusion and Business Recommendations **40**
16.1 What I Achieved 40
16.2 Why I Chose K-Means 41
16.3 Real-World Impact 41

17 Future Work and Next Steps **41**

1 Data Generation

To kick things off, I needed a robust dataset to test my segmentation strategy. I decided to generate a realistic simulation.

I set up a scenario covering a two-year period (from early 2022 to the end of 2023) representing a store with 500 unique customers.

By effectively using `pandas`, `numpy`, for-loops, and `np.random`, I created a large dataset where each row represents an item a customer bought on a specific date. Each record includes:

- CustomerID
- TransactionDate
- ProductID
- Quantity
- UnitPrice
- TotalPrice

I didn't want random numbers—I wanted the data to tell a story. So I programmed the simulation to mimic a real checkout experience:

- **Customer Personalities**: Each customer was assigned a behavioral profile. Some were high-spenders, while others were frugal. This ensured real patterns would emerge during analysis.
- **Checkout Experience**: I simulated 5,000 distinct visits to the store. For each visit, the system generated a "receipt" where the customer bought between 1 and 6 items.
- **Pricing Logic**: Prices were tied to the customer's profile. High spenders were more likely to purchase expensive goods.

	A	B	C	D	E
		Recency_Log_Scaled	Frequency_Log_Scaled	MonetaryValue_Log_Scaled	
min		-2.730787258	-4.912603213	-2.923025596	
max		2.139743759	1.874456785	1.734055378	
mean		-0.00000000002000	0.00000000004600	-0.00000000002000	
median		0.191628924	0.139742667	0.15686468	
std		1.001001503	1.001001502	1.001001503	

2 Statistical Overview

It is risky to trust a dataset without understanding its boundaries, so I ran a statistical summary to understand the typical purchasing behavior.

2.1 Quantity Analysis

Statistic	Value	Interpretation
Mean	1.998	On average, customers buy about 2 units at a time.
Min	1.0	The smallest quantity purchased is 1.
Max	3.0	The largest quantity purchased is 3.
Std Dev	0.819	Quantities are tightly clustered around the mean.

2.2 Unit Price (Cost per Item)

Statistic	Value	Interpretation
Mean	\$103.73	Average product price.
Min	\$1.22	Cheapest product sold.
Max	\$393.87	Most expensive product sold.
Median	\$76.97	Half of items cost less than this.
Std Dev	\$85.98	Indicates a wide price range.

Insight: The mean (\$103.73) is higher than the median (\$76.97), indicating a right-skewed distribution. A few expensive items are pulling up the average. The standard deviation (\$85.98) is very large relative to the mean, showing extremely diverse product pricing.

The most expensive item (\$393.87) confirms that "high spender" customers from the simulation are behaving as intended.

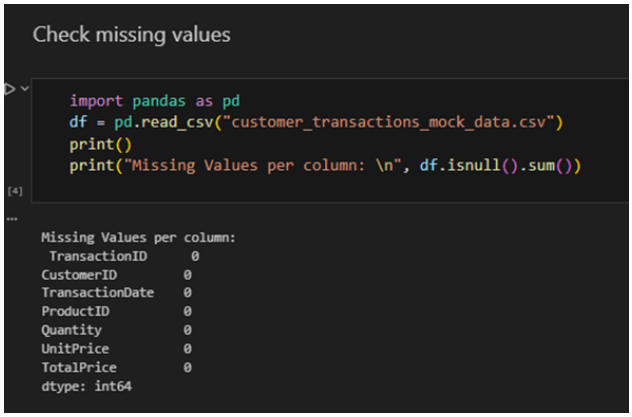
Statistic	Value	Interpretation
Mean	$207.33	Average revenue per line item.
Min	$1.54	Smallest revenue from a single line item.
Max	$1181.61	Largest revenue from a single line item.
Std Dev	$203.74	Nearly equal to the mean, indicating high volatility.

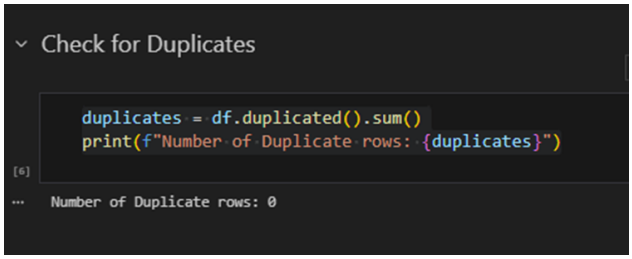
2.3 Total Price Analysis

Insight: The mean Total Price is roughly double the mean Unit Price, which makes sense because the average Quantity is 2. The extremely large standard deviation indicates that a few big purchases dramatically influence overall revenue.

2.4 Overall Business Implication

The store carries a wide range of product prices ($1.22 to $393.87). Customers generally buy small quantities (1–3 units per product), leading to an average line-item revenue of around $207.





3 Ensuring Data Integrity

Before feeding the data into any segmentation model, I needed to verify that the dataset contained no structural issues.

3.1 Duplicate Check

Double-counting transactions inflates revenue and corrupts segmentation results.

No duplicate rows found.

3.2 Invalid Transaction Check

Real-world datasets sometimes contain:

- Negative quantities (refunds)
- Zero prices (errors or promotions)

The system returned:

Empty DataFrame

This confirms that every transaction represents a valid sale with positive quantity and price.

```
Q1 Value of Quantity is:  1.0  and the Q3 value is:  3.0
IQR of column Quantity is: 2.0
Lower Bound of Quantity is: -2.0
Upper Bound of Quantity is:  6.0
Quantity: 0 outliers detected

Q1 Value of UnitPrice is:  34.010000000000005  and the Q3 value is:  154.18
IQR of column UnitPrice is: 120.17
Lower Bound of UnitPrice is: -146.245
Upper Bound of UnitPrice is:  334.435
UnitPrice: 205 outliers detected

Q1 Value of TotalPrice is:  58.74  and the Q3 value is:  285.34000000000003
IQR of column TotalPrice is: 226.60000000000002
Lower Bound of TotalPrice is: -281.16
Upper Bound of TotalPrice is:  625.24
TotalPrice: 839 outliers detected
```

4 Outlier Detection through Interquartile Range (IQR)

This analysis helps identify product sales that are unusually small or unusually large compared to the bulk of the store's transactions.

- **Q1 (First Quartile / 25th Percentile):** The value where 25% of the data points fall below it.
- **Q3 (Third Quartile / 75th Percentile):** The value where 75% of the data points fall below it (or 25% fall above it).
- **IQR (Interquartile Range):** The range that contains the middle 50% of the data.

$$IQR = Q3 - Q1$$

To find an outlier or any unusual data point, a common practice in Data Science is to define boundaries around the IQR. These boundaries are calculated as:

$$LowerBound = Q1 - 1.5 \times IQR$$

$$UpperBound = Q3 + 1.5 \times IQR$$

Values outside these limits are considered potential outliers.

4.1 Quantity Analysis

Since everyone bought between 1 and 3 items, there were zero outliers here.

4.2 Unit Price Analysis

The store sold 205 products with a unit price greater than $334.44. Since the maximum price was $393.87, these represent the very high-end luxury items.

These items are rare compared to the general stock and confirm the right-skewness previously discussed.

4.3 Total Price Analysis

A total of 839 line items generated revenue greater than $625.24. These high-value sales come from either:

- single expensive products, or
- multiple units of moderately expensive products.

Although only a small fraction of the 15,143 total records, these transactions contribute disproportionately to total revenue.

4.4 Interpreting the Impact of the Outliers

Many analyses remove outliers because they distort averages. However, in customer segmentation, these outliers are often the VIP customers.

These 839 high-value transactions likely represent the "big spenders" intentionally included in the simulation. Removing them would erase the store's most valuable customers from analysis.

```
    CustomerID  Recency  Frequency  MonetaryValue
0     CUST1000        9         41        6855.79
1     CUST1001       49         10        2856.43
2     CUST1002        7         25        2175.73
3     CUST1003       22         33        1085.57
4     CUST1004      134         27        6887.81
..         ...      ...        ...            ...
495   CUST1495       69         31       11320.65
496   CUST1496      128         28        2917.11
497   CUST1497       91         30        5462.50
498   CUST1498       80         27        7750.99
499   CUST1499      150         18        7521.02

[500 rows x 4 columns]
          Recency   Frequency  MonetaryValue
count  500.000000  500.000000      500.0000
mean    71.310000   30.286000     6279.1636
std     65.035075   10.224543     4430.0027
min      1.000000    4.000000      377.0000
25%     22.000000   23.000000     2579.4425
50%     54.000000   30.000000     5292.9750
75%    105.250000   37.000000     9319.3500
max    500.000000   57.000000    20453.5000
```

5 RFM Calculation and Their Correlation

Analyzing individual receipts does not reveal customer loyalty. Instead, the analysis shifts focus to customer-level behavior using the RFM model:

- **R – Recency:** How recently the customer made a purchase (lower = more recent).
- **F – Frequency:** How often the customer buys.
- **M – Monetary Value:** How much the customer spends.

A snapshot date of January 1, 2024 was used to compute Recency.

5.1 Summary Statistics

Recency (R)

- The average customer last visited 71 days ago.
- 25% of customers visited within the last 22 days.
- The maximum recency is 500 days, indicating a highly inactive *"churned"* customer.

Frequency (F)

- The average customer visited 30 times over two years (around 15 visits per year).
- The middle 50% of customers visited between 23 and 37 times.

Monetary (M)

- The average customer spent $6,279 over two years.
- The highest-spending customer spent more than $20,453.
- Mean ($6,279) is higher than the median ($5,293), reflecting high-spending customers pulling the average upward.

This process reduces 15,000+ line items into a concise dataset of 500 customers, saved as `RFM_data.csv`.

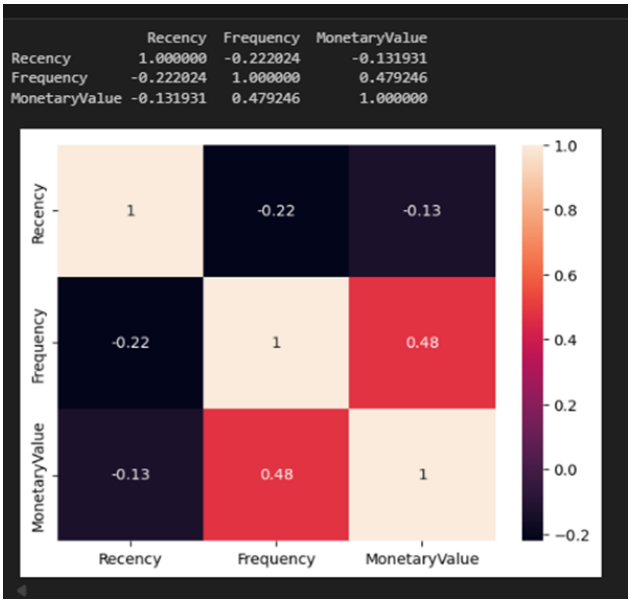


Figure 1: Enter Caption

6 Correlation – Finding the Hidden Connections

Correlations range from -1.0 (strong negative) to +1.0 (strong positive). The analysis revealed the following relationships:

Recency vs. Frequency (Correlation: -0.22)

- A mild negative correlation.
- Customers who shop frequently tend to have lower Recency values (i.e., they visited recently).
- Indicates healthy customer activity.

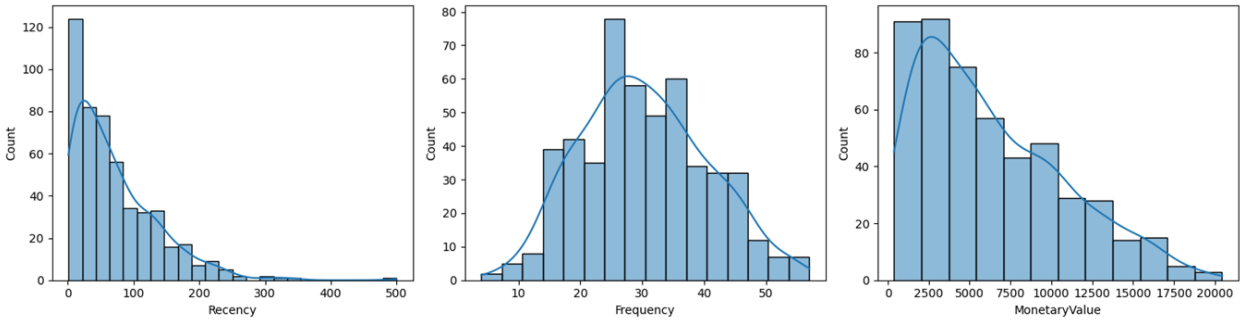
Frequency vs. Monetary Value (Correlation: +0.48)

- A strong positive correlation.
- Customers who visit more often tend to spend more.
- This is the most profitable customer segment, ideal for targeted marketing investment.

Recency vs. Monetary Value (Correlation: -0.13)

- A very weak negative correlation.

- Recency is not a strong predictor of spending.
- A customer may not visit frequently yet still spend large amounts whenever they do.



7 RFM – Exploratory Data Analysis and Feature Scaling

7.1 Histograms and Distribution Analysis

Metric	Observed Distribution	Critical Interpretation and Implication
Recency	Highly Right-Skewed (Concentrated near 0)	High Engagement & Churn Risk: Most customers purchased recently, showing strong engagement. The long tail indicates a segment of lapsed customers needing re-engagement.
Frequency	Close to Normal (Bell-Shaped)	Stable Customer Base: Most customers show moderate and consistent purchasing habits, forming the core of the business.
Monetary Value	Highly Right-Skewed	Pareto Principle: A small group of VIP customers contributes most of the revenue. Majority have low spend, offering upselling opportunities.

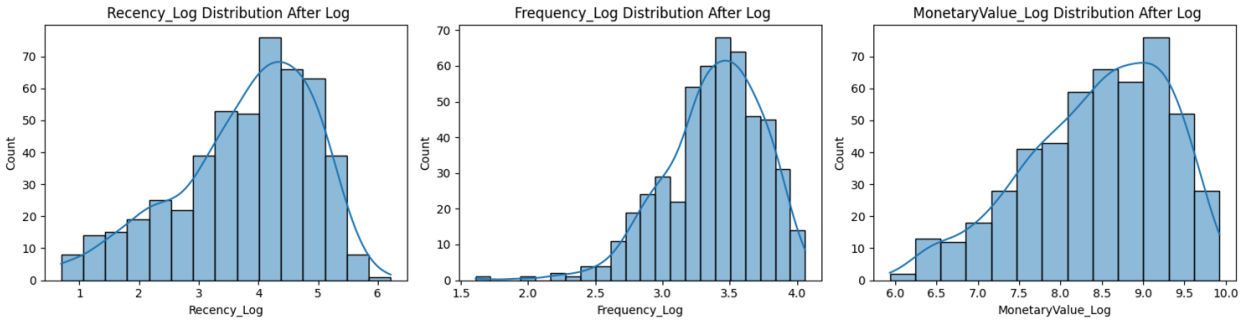
Table 2: Distribution Analysis and Implications for Customer Behavior

The RFM patterns indicate:

- Most customers are recently active — a positive sign.
- Frequency is moderate and stable, suggesting predictable buying cycles.
- Spending is concentrated among a few high-value customers.

Overall Business Interpretation:

- The customer base is healthy, but:
  - Purchase frequency can still be improved.
  - High spenders must be retained.
  - Lapsed customers should be re-engaged.



7.2 Critical Analysis of Skewness

RFM Metric	Skewness Observation	Implication for Analysis
Recency	Highly Right-Skewed	Mean heavily inflated due to outliers (customers with very long recency).
Frequency	Mildly Right-Skewed	Most balanced metric; close to a normal distribution.
Monetary Value	Highly Right-Skewed	Influenced by VIP customers with extremely high spending.

Business Impact of Skewness:

- Clustering models may overemphasize separating VIP customers.
- Subtle variations in Recency and Frequency become overshadowed.
- Outliers distort K-Means cluster centers.

7.3 Solution: Log Transformation

The logarithmic transformation reduces skewness by compressing extreme values.

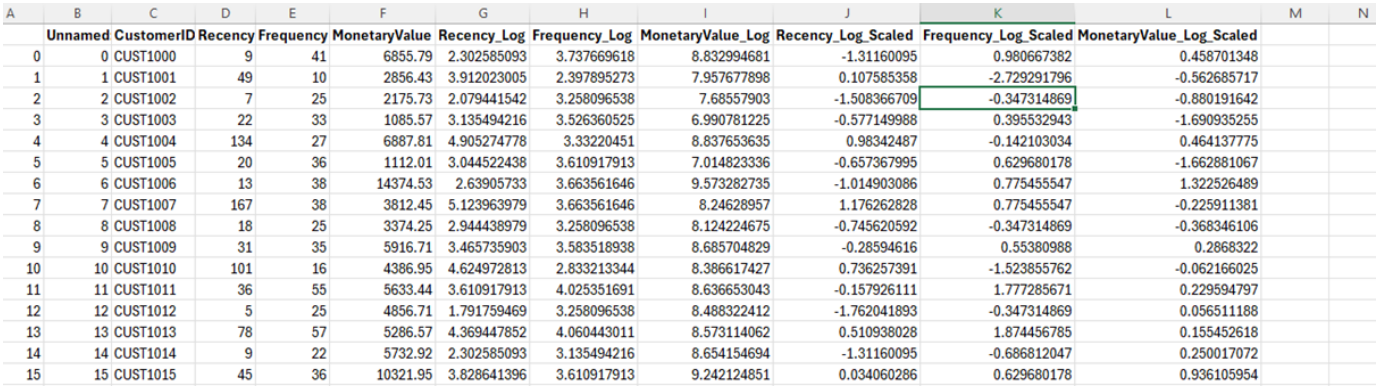
1. **Compression of Large Values:** Log grows slowly, reducing outlier impact.
2. **Expansion of Small Values:** Smaller values become more distinguishable.
3. **Result:** Distributions become more symmetric and closer to normal.

New columns created:

$$Recency\_Log, \quad Frequency\_Log, \quad MonetaryValue\_Log$$

RFM Metric	Observation After Log Transform	Interpretation & Modeling Impact	Assessment
Recency_Log	Near Normal, centered at 4–5	Long tail compressed; distribution balanced	Success
Frequency_Log	Strong Left Skew	Overcorrected; original distribution was near-normal	Failure
MonetaryValue_Log	Approximates Normal, centered 8.5–9.5	Outliers reduced; more balanced for modeling	Success

Table 3: Log-Transformed RFM Metrics and Their Implications



	Unnamed	CustomerID	Recency	Frequency	MonetaryValue	Recency_Log	Frequency_Log	MonetaryValue_Log	Recency_Log_Scaled	Frequency_Log_Scaled	MonetaryValue_Log_Scaled
0	0	CUST1000	9	41	6855.79	2.302585093	3.737669618	8.832994681	-1.31160095	0.980667382	0.458701348
1	1	CUST1001	49	10	2856.43	3.912023005	2.397895273	7.957677898	0.107585358	-2.729291796	-0.562685717
2	2	CUST1002	7	25	2175.73	2.079441542	3.258096538	7.68557903	-1.508366709	-0.347314869	-0.880191642
3	3	CUST1003	22	33	1085.57	3.135494216	3.526360525	6.990781225	-0.577149988	0.395532943	-1.690935255
4	4	CUST1004	134	27	6887.81	4.905274778	3.33220451	8.837653635	0.98342487	-0.142103034	0.464137775
5	5	CUST1005	20	36	1112.01	3.044522438	3.610917913	7.014823336	-0.657367995	0.629680178	-1.662881067
6	6	CUST1006	13	38	14374.53	2.63905733	3.663561646	9.573282735	-1.014903086	0.775455547	1.322526489
7	7	CUST1007	167	38	3812.45	5.123963979	3.663561646	8.24628957	1.176262828	0.775455547	-0.225911381
8	8	CUST1008	18	25	3374.25	2.944438979	3.258096538	8.124224675	-0.745620592	-0.347314869	-0.368346106
9	9	CUST1009	31	35	5916.71	3.465735903	3.583518938	8.685704829	-0.28594616	0.55380988	0.2868322
10	10	CUST1010	101	16	4386.95	4.624972813	2.833213344	8.386617427	0.736257391	-1.523855762	-0.062166025
11	11	CUST1011	36	55	5633.44	3.610917913	4.025351691	8.636653043	-0.157926111	1.777285671	0.229594797
12	12	CUST1012	5	25	4856.71	1.791759469	3.258096538	8.488322412	-1.762041893	-0.347314869	0.056511188
13	13	CUST1013	78	57	5286.57	4.369447852	4.060443011	8.573114062	0.510938028	1.874456785	0.155452618
14	14	CUST1014	9	22	5732.92	2.302585093	3.135494216	8.654154694	-1.31160095	-0.686812047	0.250017072
15	15	CUST1015	45	36	10321.95	3.828641396	3.610917913	9.242124851	0.034060286	0.629680178	0.936105954

Figure 2: Enter Caption

7.4 Standardization of Log-Transformed Variables

Although log transformation fixes skewness, the scales still differ.

Need for Scaling

- Frequency_Log ranges 1.5–4.0
- MonetaryValue_Log ranges 6.0–10.0

K-Means would assign higher weight to metrics with larger numeric ranges.

Action: Standard Scaling

Each column is transformed to:

$$z = \frac{x - \mu}{\sigma}$$

Final columns:

$$Recency\_Log\_Scaled, \quad Frequency\_Log\_Scaled, \quad MonetaryValue\_Log\_Scaled$$

Scaled Value	Interpretation	Customer Profile
Close to 0	Near-average customer behavior	Typical customer
Large Positive	Well above average	High-value customer (frequent or high spender)
Large Negative	Well below average	Low-value or inactive customer

	Recency_Log_Scaled	Frequency_Log_Scaled	MonetaryValue_Log_Scaled
min	-2.730787258	-4.912603213	-2.923025596
max	2.139743759	1.874456785	1.734055378
mean	-0.00000000002000	0.00000000004600	-0.00000000002000
median	0.191628924	0.139742667	0.15686468
std	1.001001503	1.001001502	1.001001503

7.5 Interpretation of Scaled RFM Values

Example Row Interpretation

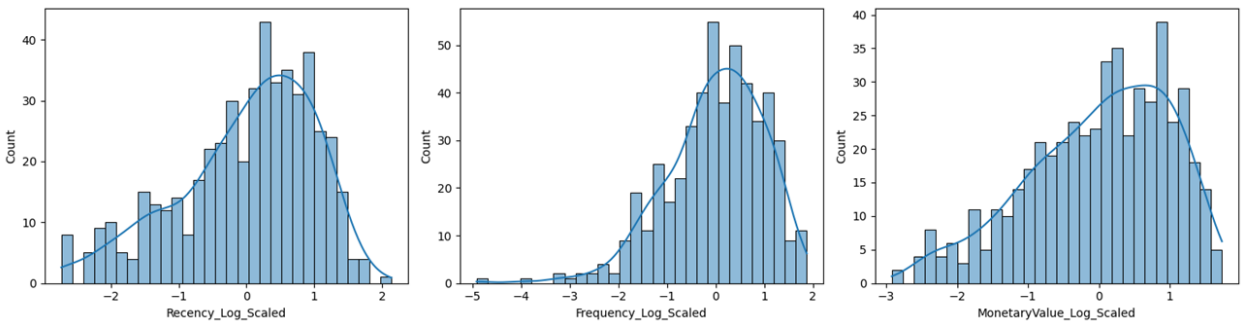
Row	*Recency_Log_Scaled*	*Frequency_Log_Scaled*	*MonetaryValue_Log_Scaled*
0	−1.31	+0.98	+0.46

- Recency: 1.31 SD more recent than average.
- Frequency: 0.98 SD more frequent than average.
- Monetary: 0.46 SD more spending than average.

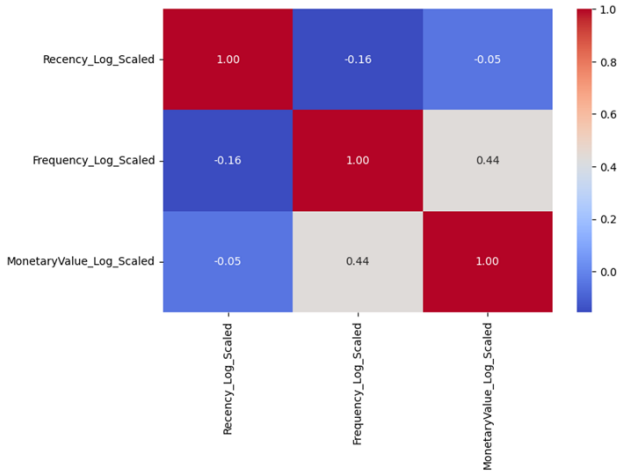
7.6 Validation of Standard Scaling

The output confirms:

- Means ≈ 0



- Standard deviations ≈ 1.00



7.7 Correlation Heatmap Findings

- Strong positive correlation between Frequency and Monetary Value (+0.44).
- Scaling preserved important behavioral relationships.

8 Clustering

8.1 K-Means Clustering

Having successfully cleaned, log-transformed, and scaled the data for all 500 customers, the dataset was ready for the K-Means clustering algorithm. K-Means is an unsupervised machine learning technique used to partition N data points into K non-overlapping clusters.

The motivation for using clustering is simple:

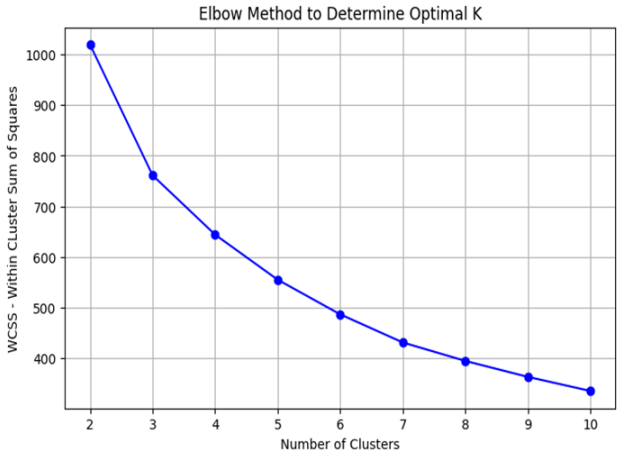
- Without segmentation, businesses treat all customers the same.
- K-Means helps discover groups such as "High-Value Loyal Customers", "Bargain Shoppers", or "Recently Active Low-Value Customers".

The crucial challenge is selecting the appropriate number of clusters K. Too few clusters (e.g., $K = 2$) oversimplifies the groups; too many (e.g., $K = 100$) overcomplicates the segmentation.

```
Data Shape: (500, 3)

Sample Data:
    Recency_Log_Scaled  Frequency_Log_Scaled  MonetaryValue_Log_Scaled
0            -1.311601              0.980667                  0.458701
1             0.107585             -2.729292                 -0.562686
2            -1.508367             -0.347315                 -0.880192
3            -0.577150              0.395533                 -1.690935
4             0.983425             -0.142103                  0.464138

The value of WCSS score for k = 2 is: [1019.1784344400421]
The value of WCSS score for k = 3 is: [1019.1784344400421, 761.1520576579288]
The value of WCSS score for k = 4 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479]
The value of WCSS score for k = 5 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171]
The value of WCSS score for k = 6 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171, 486.62734237346115]
The value of WCSS score for k = 7 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171, 486.62734237346115, 431.11859173868453]
The value of WCSS score for k = 8 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171, 486.62734237346115, 431.11859173868453, 394.662289566222]
The value of WCSS score for k = 9 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171, 486.62734237346115, 431.11859173868453, 394.662289566222, 363.1610301201067]
The value of WCSS score for k = 10 is: [1019.1784344400421, 761.1520576579288, 643.9882743673479, 555.0212503164171, 486.62734237346115, 431.11859173868453, 394.662289566222, 363.1610301201067, 335.3404255451963]
```



8.1.1 Step 1: Determining the Optimal Number of Clusters (K)

Two standard methods were used to determine the optimal value of K between 2 and 10.

A. The Elbow Method The Elbow Method evaluates cluster compactness via the Within-Cluster Sum of Squares (WCSS). Lower WCSS values indicate tighter clusters.

K	WCSS	Change from Previous
2	1019.18	–
3	761.15	-258.03
4	643.99	-117.16
5	555.02	-88.97
6	486.63	-68.39
7	431.12	-55.51
8	394.66	-36.46
9	363.16	-31.50
10	335.34	-27.82

The WCSS plot shows a sharp drop from $K = 2$ to $K = 3$, after which reductions slow

```
value of Silhouette Score for k = 2 is: [0.2978940646634863]
value of Silhouette Score for k = 3 is: [0.2978940646634863, 0.3125404872222474]
value of Silhouette Score for k = 4 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596]
value of Silhouette Score for k = 5 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803]
value of Silhouette Score for k = 6 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803, 0.25957077455702426]
value of Silhouette Score for k = 7 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803, 0.25957077455702426, 0.2682251517436623]
value of Silhouette Score for k = 8 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803, 0.25957077455702426, 0.2682251517436623, 0.2706806464303547]
value of Silhouette Score for k = 9 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803, 0.25957077455702426, 0.2682251517436623, 0.2706806464303547, 0.2744941928670349]
value of Silhouette Score for k = 10 is: [0.2978940646634863, 0.3125404872222474, 0.3036503561422596, 0.261004441625803, 0.25957077455702426, 0.2682251517436623, 0.2706806464303547, 0.2744941928670349, 0.2758070935063144
```

significantly, forming the "elbow". This indicates that $K = 3$ is the optimal number of clusters.

B. Silhouette Analysis The Silhouette Score assesses cluster quality by combining cohesion and separation. For each data point:

$$s = \frac{b - a}{\max(a, b)}$$

where:

- a = average distance to points in the same cluster (cohesion)
- b = minimum average distance to points in the nearest cluster (separation)

Scores closer to +1 indicate well-separated clusters.

K	Silhouette Score	Interpretation
2	0.2979	Good separation
3	0.3125	**Highest Score (Optimal)**
4	0.3037	Second highest
5	0.2610	Significant drop
6	0.2596	Lowest
7	0.2682	Slight recovery
8	0.2707	Slight recovery
9	0.2745	Slight recovery
10	0.2758	Slight recovery

Although $K = 3$ is optimal, the overall magnitude ($s = 0.3125$) is modest, suggesting overlapping clusters. Still, both the Elbow Method and Silhouette Analysis converge on $K = 3$.

D	E	F	G	H	I	J	K	L	M	N
Customer	Recency	Frequency	MonetaryValue	Recency_Log	Frequency_Log	MonetaryValue_Log	Recency_Log_Scaled	Frequency_Log_Scaled	MonetaryValue_Log_Scaled	Cluster
CUST1000	9	41	6855.79	2.302585093	3.737669618	8.832994681	-1.31160095	0.980667382	0.458701348	2
CUST1001	49	10	2856.43	3.912023005	2.397895273	7.957677898	0.107585358	-2.729291796	-0.562685717	1
CUST1002	7	25	2175.73	2.079441542	3.258096538	7.68557903	-1.508366709	-0.347314869	-0.880191642	2
CUST1003	22	33	1085.57	3.135494216	3.526360525	6.990781225	-0.577149988	0.395532943	-1.690935255	2
CUST1004	134	27	6887.81	4.905274778	3.33220451	8.837653635	0.98342487	-0.142103034	0.464137775	0
CUST1005	20	36	1112.01	3.044522438	3.610917913	7.014823336	-0.657367995	0.629680178	-1.662881067	2
CUST1006	13	38	14374.53	2.63905733	3.663561646	9.573282735	-1.014903086	0.775455547	1.322526489	0
CUST1007	167	38	3812.45	5.123963979	3.663561646	8.24628957	1.176262828	0.775455547	-0.225911381	0
CUST1008	18	25	3374.25	2.944438979	3.258096538	8.124224675	-0.745620592	-0.347314869	-0.368346106	2
CUST1009	31	35	5916.71	3.465735903	3.583518938	8.685704829	-0.28594616	0.55380988	0.2868322	0
CUST1010	101	16	4386.95	4.624972813	2.833213344	8.386617427	0.736257391	-1.523855762	-0.062166025	1
CUST1011	36	55	5633.44	3.610917913	4.025351691	8.636653043	-0.157926111	1.777285671	0.229594797	0
CUST1012	5	25	4856.71	1.791759469	3.258096538	8.488322412	-1.762041893	-0.347314869	0.056511188	2
CUST1013	78	57	5286.57	4.369447852	4.060443011	8.573114062	0.510938028	1.874456785	0.155452618	0
CUST1014	9	22	5732.92	2.302585093	3.135494216	8.654154694	-1.31160095	-0.686812047	0.250017072	2
CUST1015	45	36	10321.95	3.828641396	3.610917913	9.242124851	0.034060286	0.629680178	0.936105954	0
CUST1016	81	21	1014.44	4.406719247	3.091042453	6.923077295	0.543803573	-0.809903093	-1.769937417	1
CUST1017	84	38	14237.48	4.442651256	3.663561646	9.563703438	0.57548806	0.775455547	1.311348627	0

8.1.2 Step 2: Applying the K-Means Model

Using $K = 3$, the K-Means model was trained on the scaled RFM data. Each of the 500 customers was assigned a label: 0, 1, or 2.

8.1.3 Step 3: Analyzing the Customer Segments

Cluster	Count	Avg. Recency	Avg. Frequency	Avg. Monetary	Profile
0	235 (47%)	77.29	36.35	$9,028.65	Loyal Spenders
1	154 (30.8%)	106.44	20.75	$2,903.19	At-Risk / Low Value
2	111 (22.2%)	9.93	30.68	$5,141.97	New / Active Customers

Table 4: Customer Segmentation Summary

Interpretation of Segments **1. Cluster 0: The Loyal Spenders**

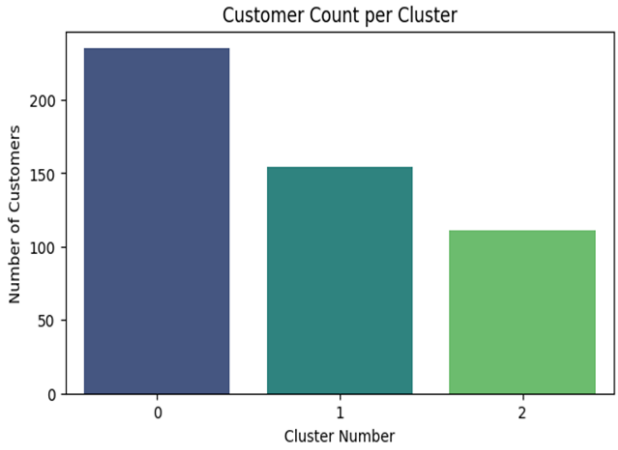
- Highest frequency (36 visits)
- Highest monetary value ($9,028)
- Strategy: Retain and upsell

2. Cluster 1: At-Risk / Low Value Customers

- Highest recency (106 days since last visit)
- Lowest spend
- Strategy: Win-back campaigns, strong discounts

3. Cluster 2: New / Active Customers

- Very low recency (9.93 days)



- Good frequency and spend
- Strategy: Nurture into high-value loyal customers

8.1.4 Conclusion

Despite moderate Silhouette Scores, the three clusters show meaningful behavioral differences. These distinctions justify unique and effective marketing strategies for each segment, demonstrating the practical success of the RFM-based K-Means segmentation model.

9 Clustering

9.1 Hierarchical Clustering

My focus then moved beyond the K-Means approach to explore **Hierarchical Clustering (HC)**. This method offers a different perspective on grouping the 500 customers by building a tree-like structure that shows how individual customers merge into larger and larger groups based on their scaled RFM scores.

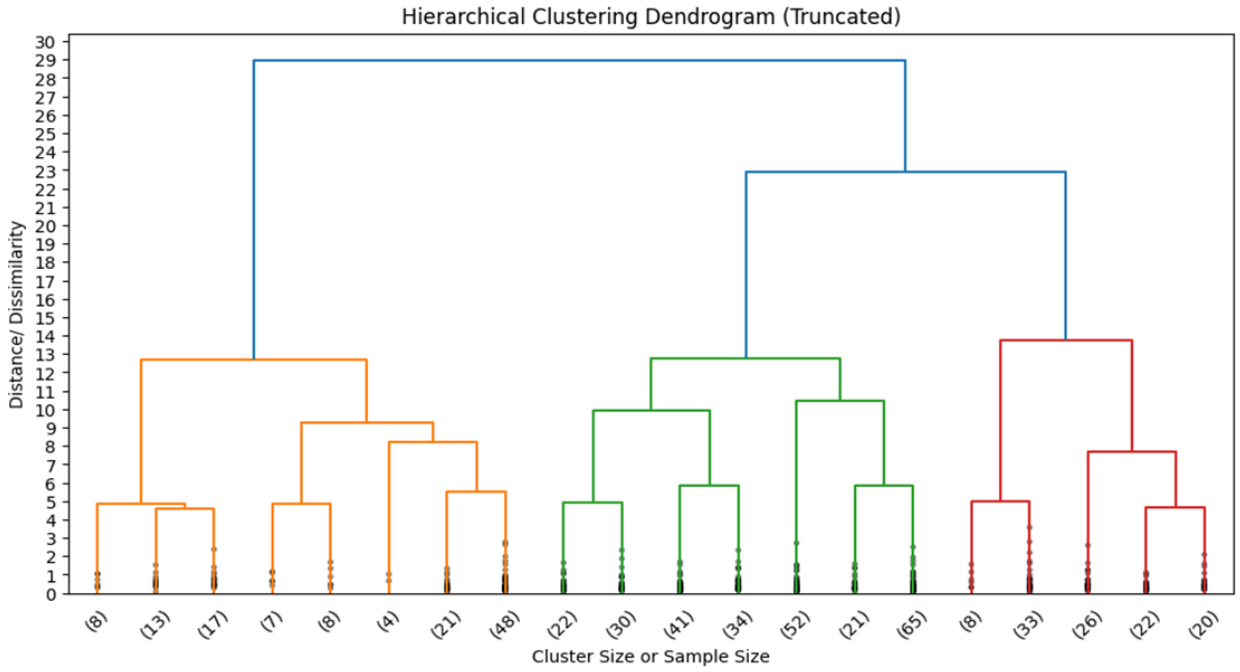
Problem: Identify natural groups (segments) of customers based on their behavior (R, F, M) so the company can tailor marketing strategies for each segment.

9.2 Hierarchical Clustering Algorithm Concept

Hierarchical Clustering builds a hierarchy of clusters, represented as a tree structure (the **dendrogram**).

1. **Start:** Every single customer is considered a separate cluster (500 customers = 500 clusters).
2. **Merge:** The algorithm finds the two closest clusters and merges them.
3. **Repeat:** Continue merging the closest remaining clusters until only one cluster remains.

Output (Linkage Matrix): The result, `lnkg`, records the merging process: which clusters were merged, at what distance, and how many points are in each new cluster.



9.3 Customer Tree – The Dendrogram Visualization

Vertical Axis (Distance / Dissimilarity)

This measures how far apart customers or groups are:

- **Low Height:** Very similar groups (e.g., two customers who both spent $50 and bought 3 times).
- **High Height:** Very different groups.

Horizontal Axis

Shows the clusters. The dendrogram was truncated to show only the last 20 major clusters.

9.4 Understanding the Dendrogram

Top Split (Height $\sim$ 29):

- Splits the entire customer base into two major groups.
- Very high distance, indicating fundamentally different RFM behavior.

Secondary Split (Height $\sim$ 23):

- Further divides one of the main groups into two subgroups.

9.5 Determining the Number of Clusters

A horizontal cutting line determines the number of clusters:

- Cut at Height 25 $\rightarrow$ 2 vertical lines $\Rightarrow \mathbf{k = 2}$
- Cut at Height 15 $\rightarrow$ 3 vertical lines $\Rightarrow \mathbf{k = 3}$
- Cut at Height 13 $\rightarrow$ 4 vertical lines $\Rightarrow \mathbf{k = 4}$

10 Applying the Hierarchical Model (k = 4)

Cluster	Count	Recency	Frequency	Monetary	Segment
4	68 (13.6%)	9.82 (Best)	35.16 (Best)	$10,137 (Highest)	VIPs
2	265 (53.0%)	82.42 (High)	34.68 (High)	$7,520 (High)	At-Risk Loyalists
3	41 (8.2%)	10.12 (Best)	30.93 (Good)	$2,073 (Lowest)	Promising / New
1	126 (25.2%)	101.04 (Worst)	18.21 (Worst)	$2,955 (Low)	Hibernating / Lost

11 Critical Business Impact & Strategy

Cluster 4: The Champions (VIPs)

Profile:

- Bought very recently (9–10 days ago)
- Highest frequency and monetary value

Strategy:

- No discounts needed
- Offer VIP treatment
- Encourage referrals

Cluster 3: Promising / Recent Low-Spenders

Profile:

- Very recent activity
- Buy often but spend little

Strategy:

- Upsell, cross-sell
- Bundles to increase AOV
- Volume-based incentives

Cluster 2: At-Risk Loyalists

Profile:

- High-value customers
- Haven't purchased in ∼82 days

Strategy:

- Immediate win-back campaigns
- Feedback survey
- Prevent shifting to Cluster 1

Cluster 1: Hibernating / Lost

Profile:

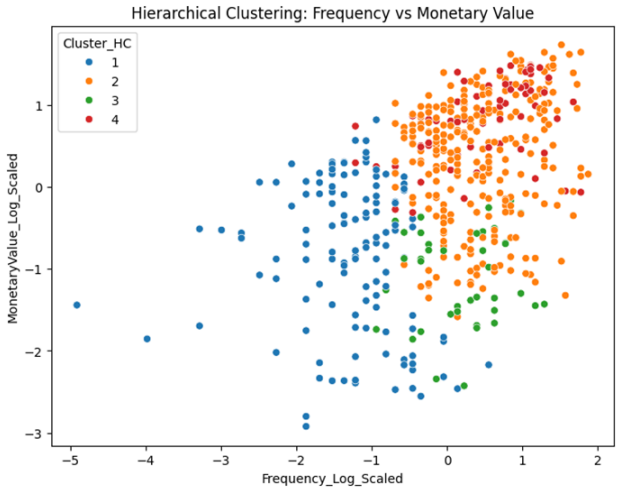
- Inactive for > 100 days
- Low spending and frequency

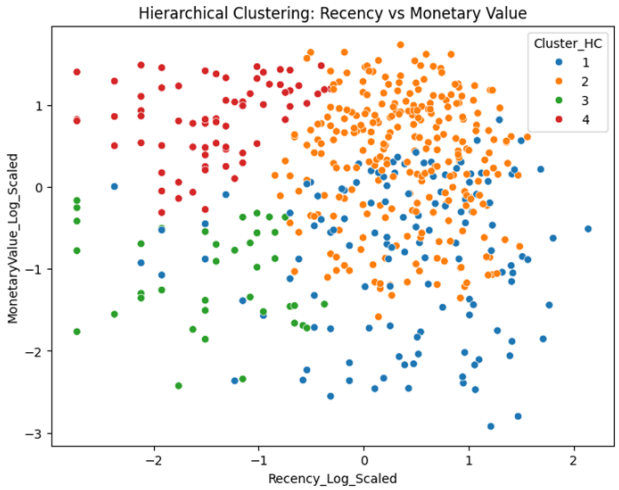
Strategy:

- Aggressive discounts
- Win-back campaigns

Key Insight: Choosing 4 clusters helps distinguish:

- High-value but active customers (Cluster 4)
- High-value but at-risk customers (Cluster 2)





12 Visualizing the 4 Clusters

12.1 Frequency vs Monetary Value

- Cluster 4 and 2: high-value segments (upper right)
- Cluster 1 and 3: low-value segments (lower left)

12.2 Recency vs Monetary Value

- Cluster 4 and 3: most recent customers
- Cluster 1 and 2: less recent, becoming dormant

The visualizations confirm that the Hierarchical Clustering method successfully created four distinct and actionable customer segments.

13 Clustering

13.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

My final step explores a third, fundamentally different clustering approach: **DBSCAN**. DBSCAN is an unsupervised density-based algorithm that groups together points that are closely packed while marking low-density points as *noise* or outliers.

Unlike K-Means or Hierarchical Clustering, DBSCAN does **not require specifying the number of clusters**. Instead, it relies on two hyperparameters:

- **Epsilon** (ε) — the maximum radius of a neighborhood.
- **MinPts** — the minimum number of points required to form a dense region.

Intuition Behind DBSCAN

Imagine a large park with people scattered around:

1. A small **radius** ε defines your local neighborhood.
2. A "crowd" exists only if a person has at least **MinPts** other people within that radius.

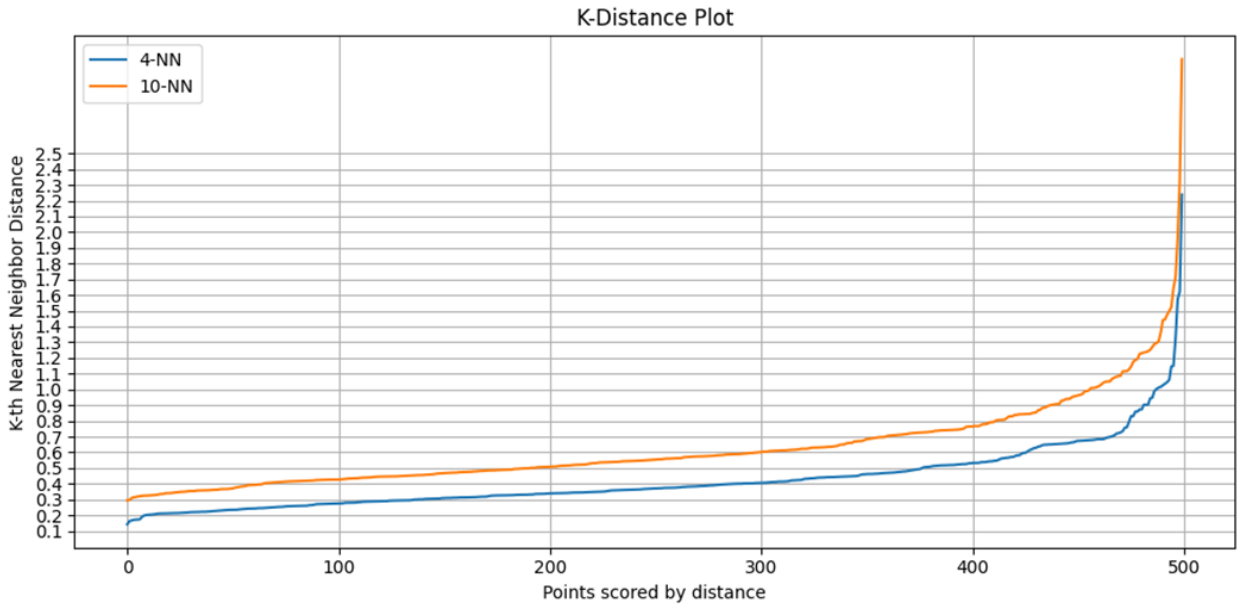
A cluster is formed when at least MinPts are packed inside a radius of distance epsilon.

Points without enough neighbors are labeled as **noise**.

The Challenge: Choosing ε

Choosing ε is crucial:

- If ε is too small $\rightarrow$ almost all points become noise.
- If ε is too large $\rightarrow$ all points merge into one giant cluster.



The K-Distance Plot

The **K-Distance Plot** (k-th nearest neighbor graph) helps determine a good ε.

Key observations from the plot:

- **Flat Region (0–450 on X-axis)**: Gradual slope, showing that ∼90% of customers lie between distances 0.2–0.6.
- **Vertical Spike (480–500)**: These points are clear outliers.
- **Elbow/Knee**: The bend occurs between **0.5 and 0.7**, suggesting this is the optimal ε range.

B	C	D	E	F	G	H	I	J	K
eps	min_samples	no_of_clusters	dbscan	noise	percentage_of_noise	silhouette	calinski_harabasz_score	davies_bouldin_score	
0.5	4	6	-15	71	14.2	0.096421607	13.10991598	0.713881946	
0.55	4	4	-26	50	10	0.084577497	13.35772716	0.751746625	
0.65	4	3	-20	31	6.2	0.233771904	9.390968293	0.736076553	
0.55	5	3	-54	69	13.8	0.226310792	16.34152573	0.708408021	
0.5	8	3	-140	165	33	0.144155388	17.04152375	0.818976141	
0.6	6	2	-37	49	9.8	0.300548664	36.57103037	0.758531234	
0.65	8	2	-54	62	12.4	0.278788957	23.36408161	0.729855436	
0.5	6	2	-113	117	23.4	0.170002884	7.094539405	0.845803307	
0.6	7	1	-74	74	14.8	0.300548664	36.57103037	0.758531234	
0.6	8	1	-83	83	16.6	0.300548664	36.57103037	0.758531234	
0.7	4	1	-22	22	4.4	0.278788957	23.36408161	0.729855436	
0.7	5	1	-32	32	6.4	0.278788957	23.36408161	0.729855436	
0.7	6	1	-34	34	6.8	0.278788957	23.36408161	0.729855436	
0.7	7	1	-40	40	8	0.278788957	23.36408161	0.729855436	
0.7	8	1	-43	43	8.6	0.278788957	23.36408161	0.729855436	
0.65	5	1	-42	42	8.4	0.233771904	9.390968293	0.736076553	
0.65	6	1	-46	46	9.2	0.233771904	9.390968293	0.736076553	
0.65	7	1	-50	50	10	0.233771904	9.390968293	0.736076553	
0.55	6	1	-82	82	16.4	0.226310792	16.34152573	0.708408021	

```
 Selected DBSCAN candidate:
eps                        0.600000
min_samples                6.000000
no_of_clusters             2.000000
dbscan                   -37.000000
noise                     49.000000
percentage_of_noise        9.800000
silhouette                 0.300549
calinski_harabasz_score   36.571030
davies_bouldin_score       0.758531
Name: 12, dtype: float64

The chosen epsilon value is: 0.6 and the chosen min samples value is: 6

Final DBSCAN fit: eps 0.6 min_samples 6
```

DBSCAN Hyperparameter Search

A grid search tested 25 combinations:

- ε: 0.5–0.7
- MinPts: 4–8

The best model (highest Silhouette Score with meaningful clusters):

- $\varepsilon = 0.60$
- MinPts = 6

This produced:

- **2 core clusters**
- **49 noise points**
- Silhouette Score: **0.30** (slightly lower than K-Means score of 0.3125)

Final DBSCAN Results

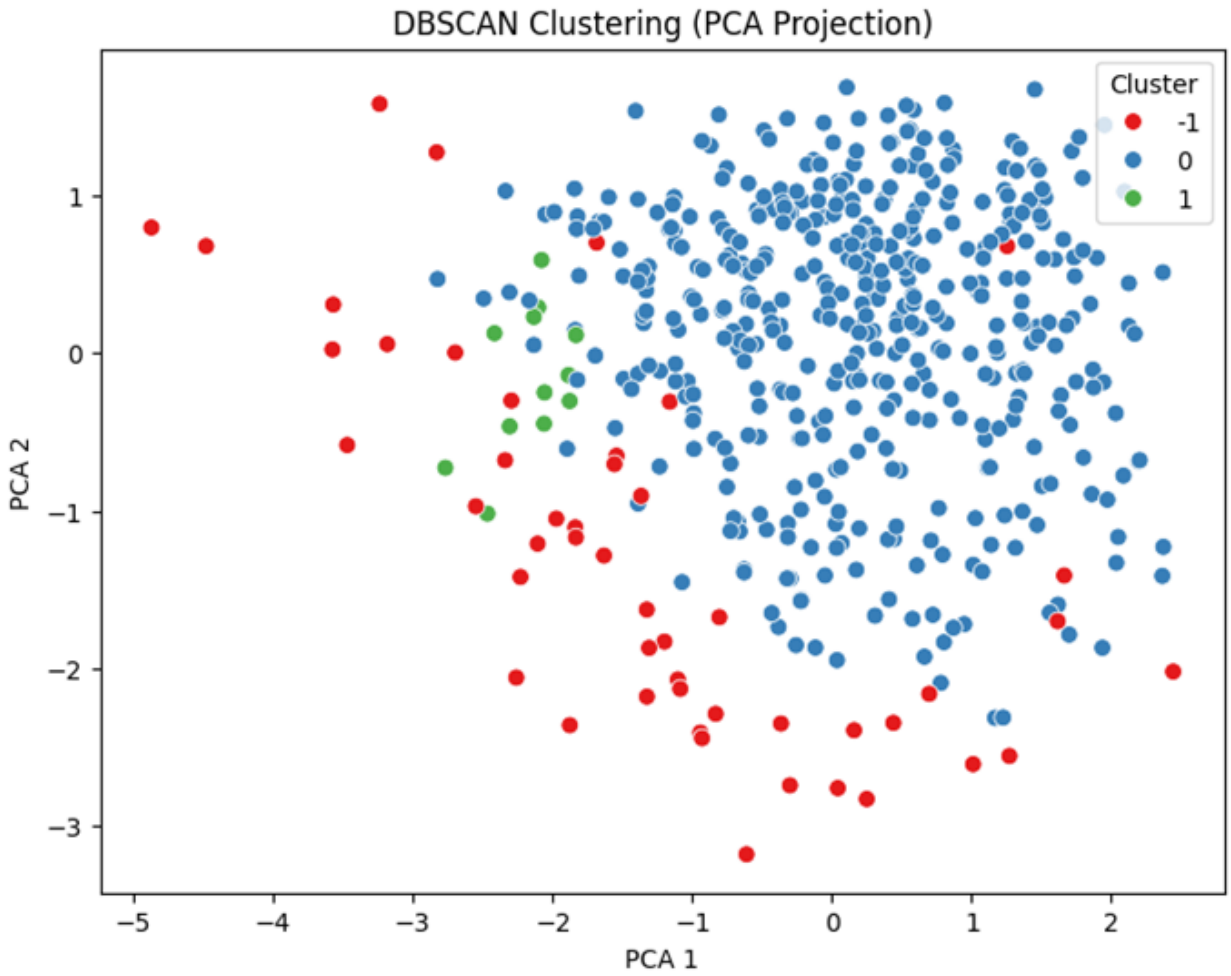
Cluster	Count	Recency	Frequency	Monetary Value
0 (Core)	439	71.46	31.37	$6,888.60
1 (Low Value)	12	104.08	21.67	$714.92
-1 (Noise)	49	61.96	22.69	$2,181.78

Business Interpretation

Cluster 0 (439 customers): High-value core customers. Maintain with standard loyalty programs.

Cluster 1 (12 customers): Very low-value and dormant. Minimal investment required.

Noise (-1, 49 customers): Outliers with unusual behavior. Investigate for hidden opportunities.



DBSCAN Visualization via PCA

The 3D RFM data was reduced to 2D using **Principal Component Analysis (PCA)**.

- **PCA 1**: Represents overall value/loyalty.
- **PCA 2**: Represents recency/activity.

Cluster interpretation:

- **Cluster 0 (Blue)**: A massive dense blob in the center — the core customer base.
- **Cluster 1 (Green)**: A small cluster at the far left — low-value, low-activity customers.
- **Noise (Red)**: Scattered heavily — the 49 outliers.

DBSCAN confirms a single dominant segment with unique outlier behaviors.

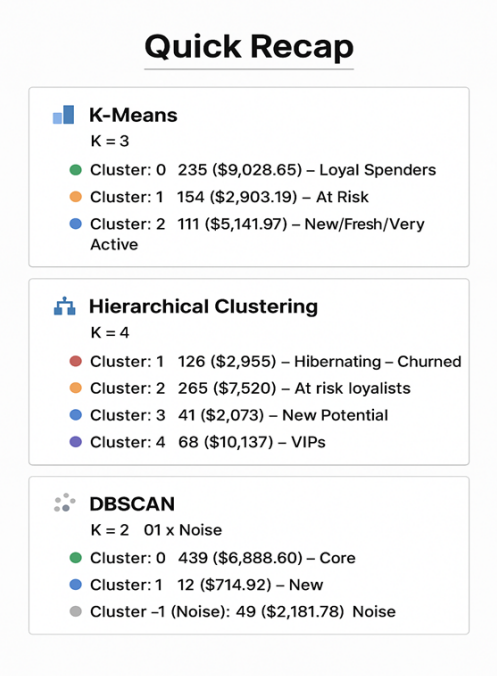


Figure 3: Enter Caption

14 Comparing All Three Clustering Algorithms

14.1 Contingency Table Analysis

Cluster labels from all 500 customers were merged using Pandas `merge()`. `crosstab()` was used to compare model overlap.

1. K-Means vs. Hierarchical Clustering

K-Means	HC 1	HC 2	HC 3
0 (Loyal High Spenders)	0	214	21
1 (At-Risk Low Spenders)	116	38	0
2 (New/Active)	10	13	88

Key Insights:

- 214 customers appear in both K-Means 0 and HC 2 (loyal core segment).
- 116 customers match between K-Means 1 and HC 1 (at-risk).
- 88 customers align between K-Means 2 and HC 3 (new/active).

This indicates high structural agreement between the two models.

2. K-Means vs. DBSCAN

K-Means	DBSCAN -1	Cluster 0	Cluster 1
0 (Loyal High Spenders)	1	234	0
1 (At-Risk Low Spenders)	24	118	12
2 (New/Active)	24	87	0

Interpretation:

- DBSCAN **confirms one massive central cluster**: 234 of the most loyal customers (K-Means 0) appear in DBSCAN Cluster 0.
- DBSCAN reveals **49 outliers**, indicating anomalies not captured by K-Means.
- An important point to note is that the overwhelming majority of all KMeans customers (234 from Cluster 0, 118 from Cluster 1, 87 from Cluster 2) are placed inside the single, giant DBSCAN Cluster 0.
- Another important to note is that nearly all of the DBSCAN Noise (48 out of 49) came from the less valuable/less engaged segments (KMeans 1 and 2). This means the customers that K-Means struggled most to define (the ones on boundaries) are the same ones DBSCAN flagged as statistical anomalies. These 48 customers are the most problematic or exceptional in the entire dataset

14.2 Deep Dive into the 49 DBSCAN Noise Customers

I isolated the 49 DBSCAN Noise Customers and saved them as `Customers_with_DBSCAN_Noise.csv`.

The summary statistics of this file reveal the following:

Metric	Mean (Noise)	75th Percentile	Max (Noise)
Recency (days)	61.96	49.0	500.0
Frequency	22.69	29.0	57.0
Monetary Value ($)	2181.78	2946.16	15387.91

14.3 Key Insight

The noise group represents a mixture of diverse customer behaviors.

- The mean Recency, Frequency, and Monetary Value are low-to-mid, indicating most noise customers are erratic low spenders.

- **Critically, the maximum Monetary Value is $15,387.91.**
- This confirms these 49 customers represent unique, one-off cases (e.g., "Big Spender / One-Timer") that do not conform to stable mass-market customer patterns.

14.4 Top 10 Noise Customers by Monetary Value

CustomerID	Recency	Frequency	Monetary Value ($)
CUST1262	1	42	15387.91
CUST1013	78	57	5286.57
CUST1025	24	11	4857.33
CUST1298	2	16	4643.24
CUST1386	4	51	4430.18
CUST1375	6	55	4375.72
CUST1271	1	39	4012.02
CUST1117	1	35	3722.32
CUST1282	1	22	3235.97

14.5 Business Impact

- **Noise does NOT mean low value.** The top noise customer (CUST1262) has the *highest Monetary Value in the entire dataset* ($15,387.91) and purchased just yesterday (Recency = 1). These customers are labeled as noise because their purchasing patterns are extreme and atypical.
- **Actionable Intelligence:** The 49 noise customers should be extracted for special attention.
  - High spenders (e.g., CUST1262) should move to a "VIP Analyst" list for personalized handling.
  - Customers with high Recency but low Monetary Value may represent data-entry errors or fraud risks.

  The noise cluster is a goldmine for exception-based strategies.

15 Final Model Selection and Segmentation

K-Means was selected for its balance of simplicity and effectiveness (Silhouette Score = 0.3125, the highest achieved).

Exploring Hierarchical Clustering was essential, as it uniquely revealed the *At-Risk Loyalist* segment, which became a key actionable insight.

DBSCAN was used primarily to detect noise/outliers, revealing 49 customers with erratic or non-standard behavior, including one exceptionally high-spending customer flagged for manual VIP review.

I selected the 3-cluster K-Means model as the final segmentation strategy and exported the results as `Final_Customer_Segments.csv`.

KMeans Cluster	Count	Mean Monetary Value ($)	Segment Label	Business Action
0	235 (47%)	9028.65	Loyal High Spenders	Focus on retention and cross-selling high-margin items.
2	111 (22%)	5141.97	Recent Mid-Tier Buyers	Focus on immediate upselling and migration to Cluster 0.
1	154 (31%)	2903.19	At-Risk Low Spenders	Focus on targeted discounts and win-back campaigns.

16 Conclusion and Business Recommendations

16.1 What I Achieved

The primary goal of this project was to move beyond viewing customers as a single, undifferentiated mass of transactions and instead identify them as distinct groups with unique behavioral patterns. The intention was not merely to run an algorithm, but to convert raw sales data into an actionable strategy that a marketing manager can effectively use.

To achieve this, I engineered features by transforming raw transaction logs into an RFM (Recency, Frequency, Monetary Value) model. This enabled me to score every customer based on their actual value to the business rather than relying on demographics alone.

To uncover natural customer groupings, I experimented with three unsupervised learning techniques:

1. **K-Means Clustering:** Provided the cleanest, most balanced segmentation, resulting in three distinct and interpretable customer groups.

2. **Hierarchical Clustering:** Produced four clusters, giving a more granular view by separating "VIPs" from "Core Loyalists." Although useful, the insights were similar to K-Means but more complex.

3. **DBSCAN:** A density-based approach that helped identify outliers. It indicated that while most customers behave similarly, around 50 customers formed "noise," likely one-off bulk buyers or anomalies.

16.2 Why I Chose K-Means

After comparing results from all models, I selected the 3-cluster K-Means model for final deployment due to its balance of accuracy and interpretability—critical for business use. The three resulting segments were practically actionable:

- **The Loyal Spenders (Cluster 0):** High-frequency, high-value customers (average spending: $9k).
  - *Business Action:* Do not discount them unnecessarily. Instead, offer VIP perks such as "Early Access" programs.
- **The New & Active (Cluster 2):** Recently engaged customers (average last visit: 10 days ago) with moderate spending (about $5k).
  - *Business Action:* Target with upselling campaigns or "buy again" nudges to convert them into Loyal Spenders.
- **The At-Risk (Cluster 1):** Customers who have not returned in over 3 months and exhibit low spending (around $2.9k).
  - *Business Action:* Prioritize these customers for win-back strategies such as personalized discounts (e.g., "We miss you – enjoy 20% off").

16.3 Real-World Impact

By segmenting customers effectively, the business can avoid costly, inefficient marketing decisions. Instead of offering unnecessary discounts to Loyal Spenders (who would purchase anyway), resources can be redirected toward At-Risk customers who genuinely require incentives. This enables optimized marketing spend and maximizes Customer Lifetime Value (CLV).

17 Future Work and Next Steps

Although this analysis offers a detailed snapshot of current customer dynamics, data science is inherently iterative. If extended into a production environment, the next strategic steps would include:

1. **Predict Churn Using Classification Models**
   Right now, I am describing what has happened. The next logical step is to predict what will happen. I would use the "At-Risk" labels generated here as the target variable

to train a Supervised Learning model (like Logistic Regression or Random Forest). This would allow me to flag a customer as "likely to churn" before they actually stop buying..

2. **Automate the Pipeline**
   Currently, this is a static report. In a real job, I would wrap this code into a Python script or an API that runs automatically every Sunday night. This would ensure the marketing team has fresh segments every Monday morning without manual intervention.

3. **A/B Test Marketing Strategies**
   To quantify the impact of segmentation, I would perform controlled experiments. For example, split the At-Risk cluster into two groups: offer a discount to Group A but not to Group B. The difference in conversion rates would measure campaign ROI.

4. **Market Basket Analysis**
   Now that I know who the customers are, I want to know what they buy together. I would perform an Association Rule Mining (Apriori Algorithm) analysis on the "Loyal Spenders" cluster to understand which products drive their loyalty, helping with inventory planning and bundle offers.