Pattern Recognition for Customer Segmentation and High-Value Customer Identification Using Historical Transaction Data

YUNG-CHUN LAN

0. Introduction

0.1 Motivation

In retail business, not all customers are equal. A small group of customers buys very often or in large quantities, while most customers only make a few small purchases. For a company, it is important to identify and understand these customers so that marketing resources can be allocated more effectively among existing customers.

The Online Retail II dataset from UCI gives us a real set of transaction records from a UK-based online retailer. With this data, we move from raw invoices to customer-level behavior, and use pattern recognition methods to find different types of customers automatically, instead of doing it by hand.

In short, our goal is:

- To explore behavioral patterns among customers with historical transactions.
- To see how well classical pattern recognition (PR) methods can help us find and correctly treat different customers.

0.2 Project Overview

This project builds a pattern recognition pipeline on the Online Retail II dataset:

1. Data cleaning, aggregation, visualization
2. Data standardization, dimensionality reduction (PCA)
3. Unsupervised learning method (K-means)
4. Supervised learning method (compare LDA, KNN and Naïve Gaussian)

Overall, this project will show how can we use PR methods to turn messy transaction records into clear customer groups.

1. Data Cleaning, Aggregation, Visualization

1.1 Raw Data Overview

The dataset used in this project contains 541,910 rows and 8 columns, recording online retail transactions from 2010 to 2011. The data comes from the UCI Machine Learning Repository (Online Retail II dataset), which provides real transaction records from a UK-based online retailer. The fields include:

Invoice	Invoice number (those starting with "C" indicate returns)
StockCode	Product code
Description	Product description
Quantity	Quantity purchased
InvoiceDate	Transaction date and time
Price	Unit price
CustomerID	Customer identifier
Country	Customer's country

Invoice	Stockcode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom

Figure 1.1: The first 3 rows of raw data

1.2 Data Cleaning

A quick check of the dataset shows that the data is generally clean. The only issue is returned items, which appear as transactions with negative quantities. After removing these return records, the remaining dataset contains 397,925 rows and 8 columns and is ready for further processing and analysis.

1.3 Data Aggregation

The dataset is structured at the transaction-level, where each row represents information for a single invoice line. However, since our project focuses on customer behavior, we need to transform the data into a customer-level format.

To do this, we grouped the data by CustomerID and created the following six features:

Feature	Description	Effect
Recency	Number of days since the last purchase	Customer activity level
Frequency	Number of purchases	High-frequency vs. low-frequency customer
Monetary	Total spending	Measuring customer value
AvgUnitPrice	Average unit price	Assessing spending level
UniqueItems	Number of product types purchased	Identifying diverse vs. specialized customer groups
AvgQuantPerOrder	Average quantity per order	Assessing bulk vs. non-bulk purchasing behavior

This transformation produces a customer-level dataset with 4,339 rows and 6 columns, where each row corresponds to a unique customer and each column represents one of the derived behavioral features.

1.4 Data Visualization

First, we observe the statistical characteristics of each feature:

	Recency	Frequency	Monetary	AvgUnitPrice	UniqueItems	AvgQuantPerOrder
count	4339.00	4339.00	4339.00	4339.00	4339.00	4339.00
mean	92.52	4.27	2053.80	4.47	61.49	47.94
std	100.01	7.71	8988.25	34.21	85.36	1218.16
min	1.00	1.00	0.00	0.00	1.00	1.00
25%	18.00	1.00	307.25	2.20	16.00	6.00
50%	51.00	2.00	674.45	2.92	35.00	10.00
75%	142.00	5.00	1661.64	3.83	77.00	14.67
max	374.00	210.00	280206.02	2033.10	1787.00	74215.00

Next, we visualize the data of each feature and examine their distributions:

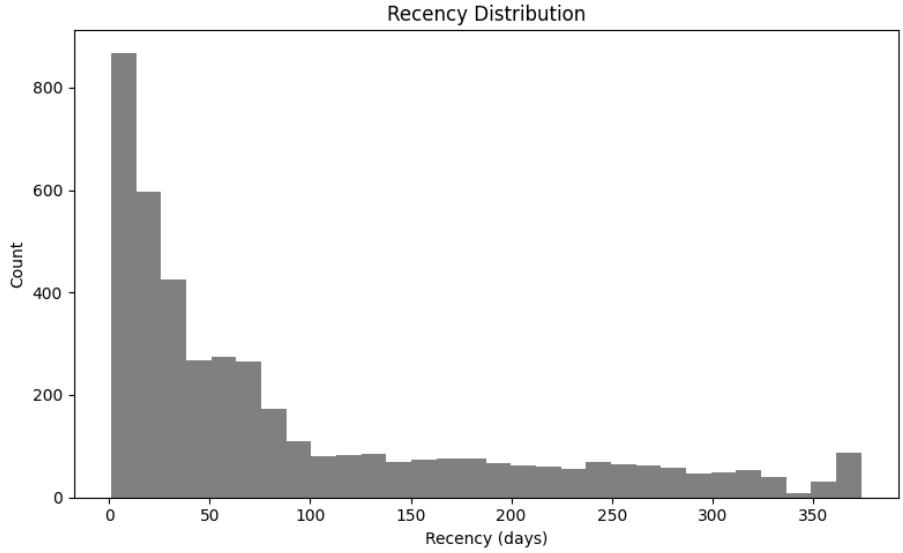


Figure 1.2: Recency shows a right-skewed distribution. Most customers made purchases within the last 50 days, while a smaller portion has not purchased for a long period (>200 days). This can serve as an important indicator of customer activity levels in the subsequent clustering analysis.

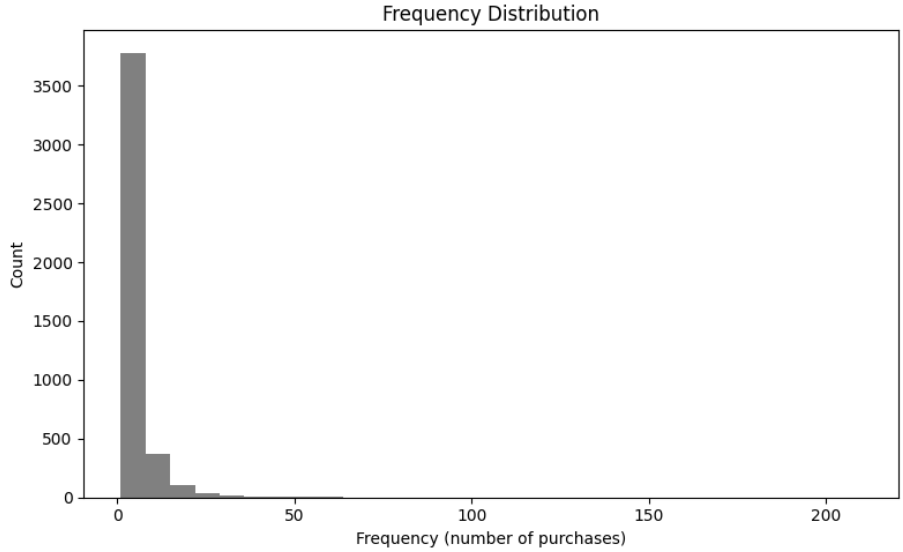


Figure 1.3: Frequency exhibits an extremely right-skewed distribution. Most customers made 1–3 purchases, with a median of 2, while the maximum reaches 210, indicating that loyal customers create a significant skew.

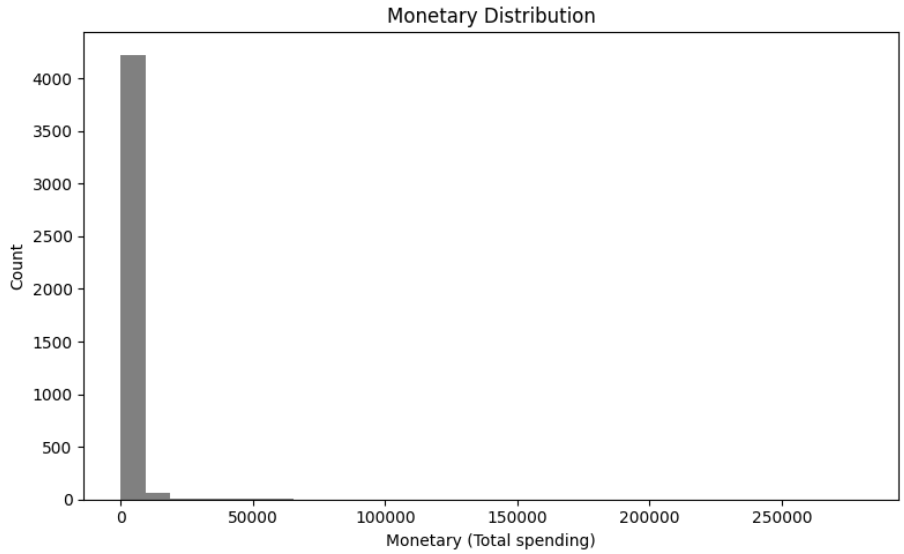


Figure 1.4: Monetary is extremely right-skewed, with a maximum value reaching 280,206 while the median is only 674. The mean value is severely inflated by enterprise clients (or corporate customers).

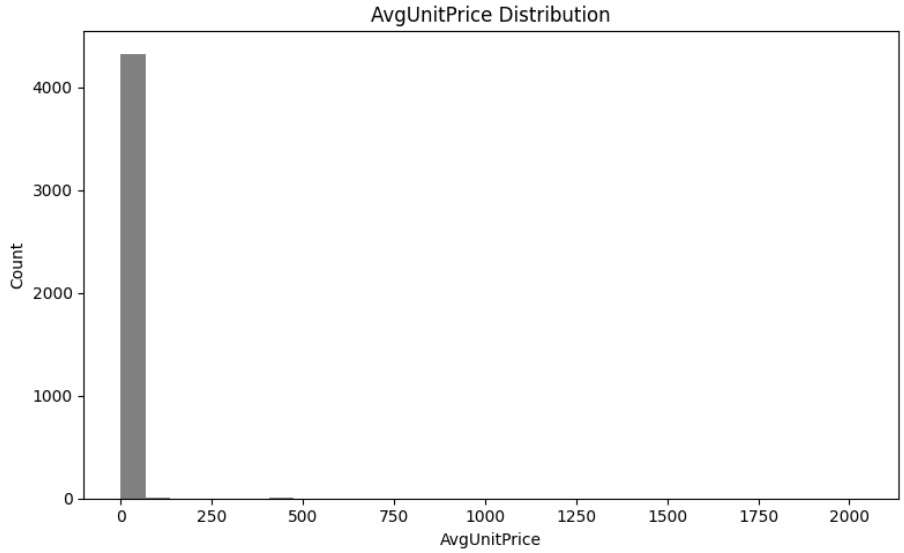


Figure 1.5: AvgUnitPrice exhibits a right-skewed distribution. The average unit price for most customers is clustered around £3, but the presence of a few high-priced items (with a maximum value of £2,033) stretches the tail of the distribution.

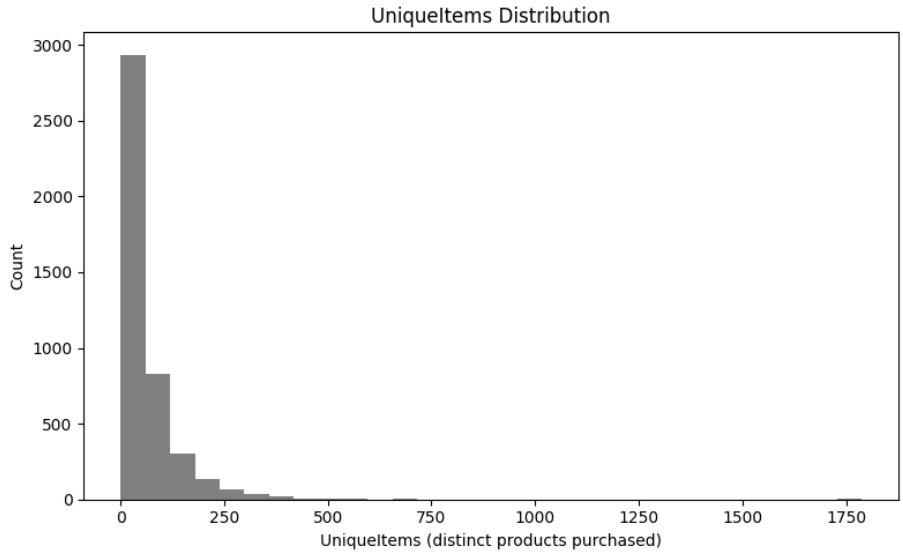


Figure 1.6: UniqueItems exhibits a long-tailed, right-skewed distribution. This suggests that most customers purchase only a small variety of items, while a minority of B2B or retailer clients purchase a large volume of different products, with the maximum reaching 1,787 types.

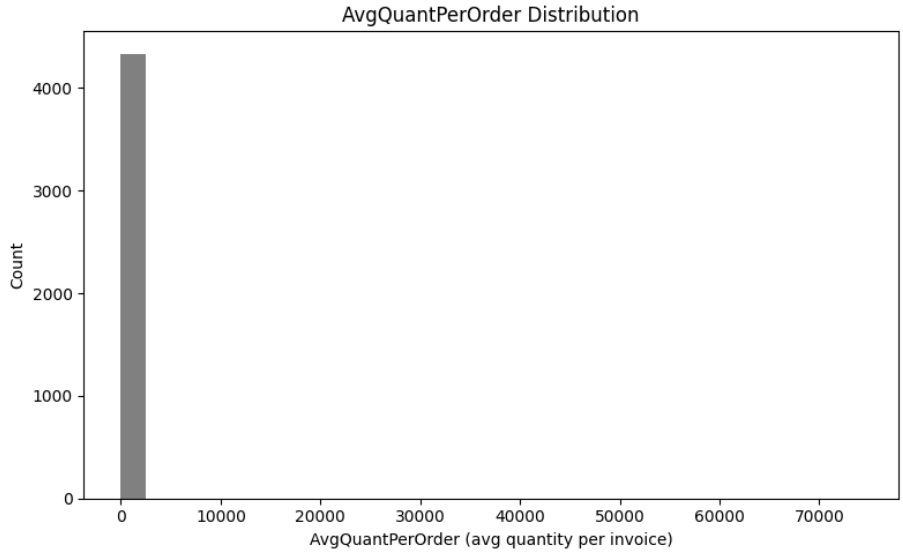


Figure 1.7: AvgQuantPerOrder is extremely right-skewed, with a maximum value of 74,215, which is significantly higher than the median of 10. This indicates that a small number of enterprise clients are making bulk purchases in a single order, thereby significantly influencing the feature distribution.

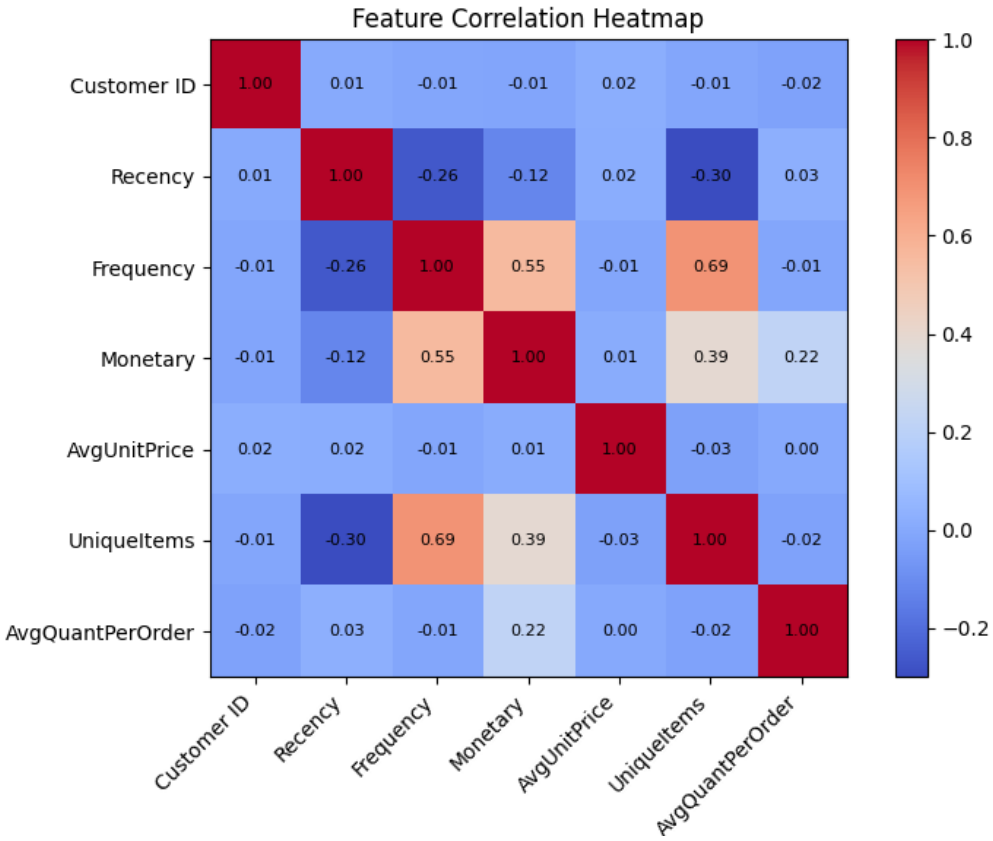


Figure 1.8: Feature correlation heatmap

1.8 Summary

We completed data cleaning, feature aggregation, and exploratory data analysis. All features exhibit right-skewed distributions, and understanding how to handle outliers and differences in feature scales will form the foundation for subsequent dimensionality reduction and clustering.

2. Data Standardization and Dimensionality Reduction

2.1 Standardization

Because the features vary widely in scale and cannot be directly compared, and because PCA is sensitive to variables with large variances, we applied Z-score standardization before performing PCA. This transforms each feature to have a mean of 0 and a standard deviation of 1:

$$z_i = \frac{x_i - \mu}{\sigma}, where$$

$$x_i: original\ value\ of\ the\ feature,$$

$$\mu: mean\ of\ the\ feature,$$

$$\sigma: standard\ deviation\ of\ the\ feature$$

Customer ID	Recency _scaled	Frequency _scaled	Monetary _scaled	AvgUnitPrice _scaled	UniqueItems _scaled	AvgQuantPerOrder _scaled
12346	2.33	-0.42	8.36	-0.10	-0.71	60.89
12347	-0.91	0.35	0.25	-0.05	0.49	-0.03
12348	-0.18	-0.04	-0.03	0.04	-0.46	0.02

Figure 2.1: The first 3 rows of standardized data

2.2 PCA

After standardizing the data, we applied PCA to reduce the dimensionality of the customer features. PCA decomposes the covariance matrix and selects the eigenvectors corresponding to the largest eigenvalues, which capture the greatest amount of variance. Using these principal components allows us to represent the data in fewer dimensions while retaining most of the important information. This reduced representation not only simplifies visualization but also provides a cleaner and more meaningful feature space for the following clustering analysis.

We computed the six principal components and their corresponding explained variance ratios, summarized in the figure below:

	Recency _scaled	Frequency _scaled	Monetary _scaled	AvgUnitPrice _scaled	UniqueItems _scaled	AvgQuantPerOrder _scaled	Explained_Var ratio
PC1	**-0.308**	**0.595**	**0.484**	-0.017	**0.559**	0.066	0.37
PC2	0.362	-0.052	0.379	0.154	-0.165	**0.820**	0.19
PC3	-0.009	0.033	-0.005	**0.984**	0.014	-0.173	0.17
PC4	0.851	0.222	0.192	-0.073	0.113	-0.414	0.14
PC5	-0.223	-0.110	0.701	-0.047	-0.574	-0.339	0.09
PC6	-0.015	0.762	-0.307	-0.005	-0.563	0.084	0.04

Figure 2.2: Principal components and each explained variance ratio

- **PC1 (37.3%): Customer Value and Activity**

  PC1 shows strong positive loadings on Frequency, Monetary, and UniqueItems, and a negative loading on Recency. It reflects overall customer value and engagement: higher PC1 scores correspond to customers who purchase more frequently, spend more, and buy a wider variety of products.PC2 (18.6%): Bulk or Wholesale Purchasing Behavior

- **PC2 (18.6%): Bulk or Wholesale Purchasing Behavior**

  PC2 is primarily driven by AvgQuantPerOrder, indicating customers who buy large quantities in a single order. Higher PC2 scores suggest wholesale or corporate purchasing behavior.

- **PC3 (16.6%): Unit Price Preference**

  PC3 is dominated by AvgUnitPrice and captures customers' preference for higher or lower priced items.

- **PC4 (14.2%): Recency-Driven Behavior**

  PC4 is heavily influenced by Recency, representing how recently customers made their last purchase.

- **PC5 (8.8%): High-Price but Low-Diversity Purchases**

  PC5 reflects customers who tend to buy expensive items but with relatively low product variety.

- **PC6 (4.5%): Frequent Purchases of the Same Item**

  PC6 captures customers who repeatedly purchase similar or identical items.

Overall, the first three principal components together explain 72.55% of the total variance. Their loading patterns clearly correspond to three key dimensions of customer behavior—value and activity, bulk purchasing tendency, and unit-price preference. Therefore, we retain the first three components as the reduced representation for subsequent clustering and behavioral pattern analysis.

Customer ID	PC1	PC2	PC3
12346	6.682	54.062	-10.741
12347	0.882	-0.362	-0.022
12348	-0.239	0.029	0.028

Figure 2.3: The first 3 rows of the PCA-transformed dataset (PC1–PC3).

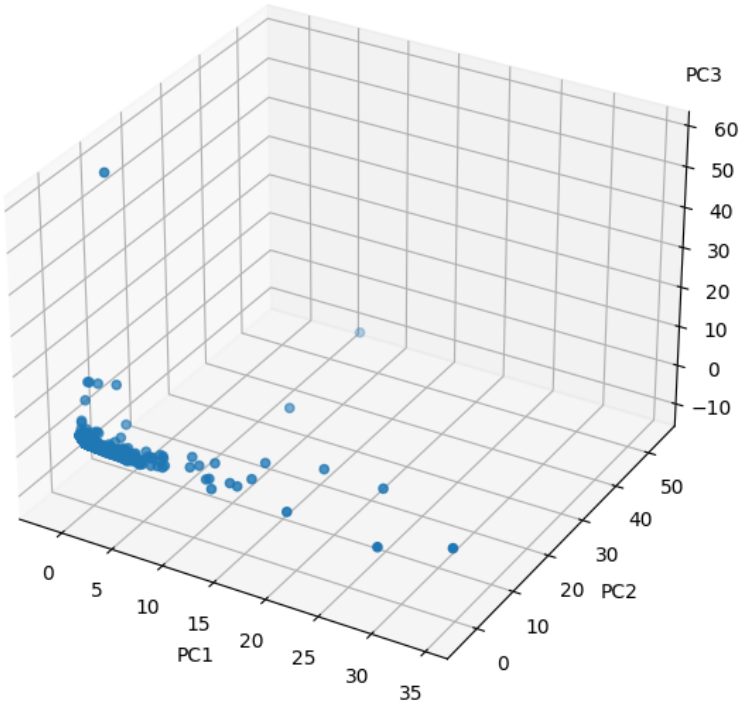


Figure 2.4: 3D PCA visualization

The 3D PCA plot (PC1–PC3) shows that customer behaviors lie on three major dimensions: PC1 represents customer value and activity (high Frequency, Monetary, and UniqueItems), PC2 captures bulk purchasing behavior (high AvgQuantPerOrder), and PC3 reflects unit-price preference.

Most customers cluster near the origin, suggesting typical retail purchasing patterns. A smaller group extends along PC1, representing high-value customers, while another group stretches along PC2, corresponding to wholesale or bulk-purchasing clients. This structure indicates a natural segmentation within the customer base and provides a strong foundation for clustering.

3. Unsupervised Learning Method

After completing data standardization and PCA, this section focuses on applying unsupervised learning. We perform K-means clustering on two versions of the dataset:

(1) the standardized features after PCA transformation.

(2) the standardized features without PCA.

Our goal is to compare how these two data representations affect the performance and behavior of the K-means algorithm, and to generate cluster labels for subsequent supervised learning.

3.1 K-means Clustering & Silhouette Score

K-means is an unsupervised method that groups data points into k clusters based on similarity. Each data point $x_i \in \mathbb{R}^d$ is assigned a label $c_i \in \{1, \dots, k\}$, and each cluster has a center $\mu_1, \dots, \mu_k \in \mathbb{R}^d$. The goal is to find the cluster labels and centers that minimize the total distance between each point and its assigned cluster center:

$$\min J = \sum_{i=1}^{n} ||x_i - \mu_{c_i}||^2$$

The algorithm repeats two simple steps:

- Step 1 (Assign points):

$$c_i = \underset{j \in \{1, \dots, k\}}{argmin} ||x_i - u_j||^2$$

- Step 2 (Update centers):

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \ , C_k = \{x_i : c_i = k\}$$

The process continues until the cluster centers stop changing or the improvement becomes very small. In practice, this means the clustering is stable:

$$\forall\, k,\ \ \mu_k^{(t+1)} = \mu_k^{(t)} \quad or \quad J^{(t+1)} - J^{(t)} < \varepsilon$$

This simple iterative process makes K-means easy to use and fast for many real-world datasets.

When choosing the best value of k, we use the Silhouette Score because it shows how well each point fits inside its own cluster compared to how close it is to other clusters. For each data point x_i, the score is defined as:

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

where

$a(i)$: the average distance from point i to other points in the same cluster.

$b(i)$: the average distance from point i to the closest different cluster.

We can interpret the score simply:

- $a(i)$ small and $b(i)$ large: $s(i) \rightarrow 1$, good clustering
- $a(i)$ close to $b(i)$: $s(i) \rightarrow 0$, point may be on a boundary
- $a(i)$ large and $b(i)$ small $\rightarrow$ $s(i) < 0$, likely misclustered

The overall Silhouette Score is the average over all $s(i)$:

$$S = \frac{1}{n}\sum_{i=1}^{n} s(i)$$

This gives us a simple way to compare different values of *k* and choose the one that creates the clearest and most meaningful clusters.

3.2 Clustering with the standardized features after PCA

First, we compute the Silhouette Scores for k values ranging from 2 to 10 using the PCA-transformed data for comparison.

k	2	3	4	5	6	7	8	9	10
score	0.932	0.924	0.925	0.563	0.510	0.514	0.518	0.520	0.520

Figure 3.1: The Silhouette Score of k=2-10 (after PCA)

The average Silhouette Scores show that k = 2, 3, and 4 all produce unusually high values in the PCA-transformed space. Starting from k = 5, the Silhouette Score drops sharply (from around 0.92 to 0.56).

In this case, considering the scale of a small-to-medium retail business, a smaller number of customer segments is more practical and easier to interpret. Therefore, balancing the Silhouette Score results with business interpretability, we select k = 3 as the number of clusters.

Next, we examine the number of data points in each cluster for k = 3:

Cluster ID	1	2	3
Quantity	4320	1	18

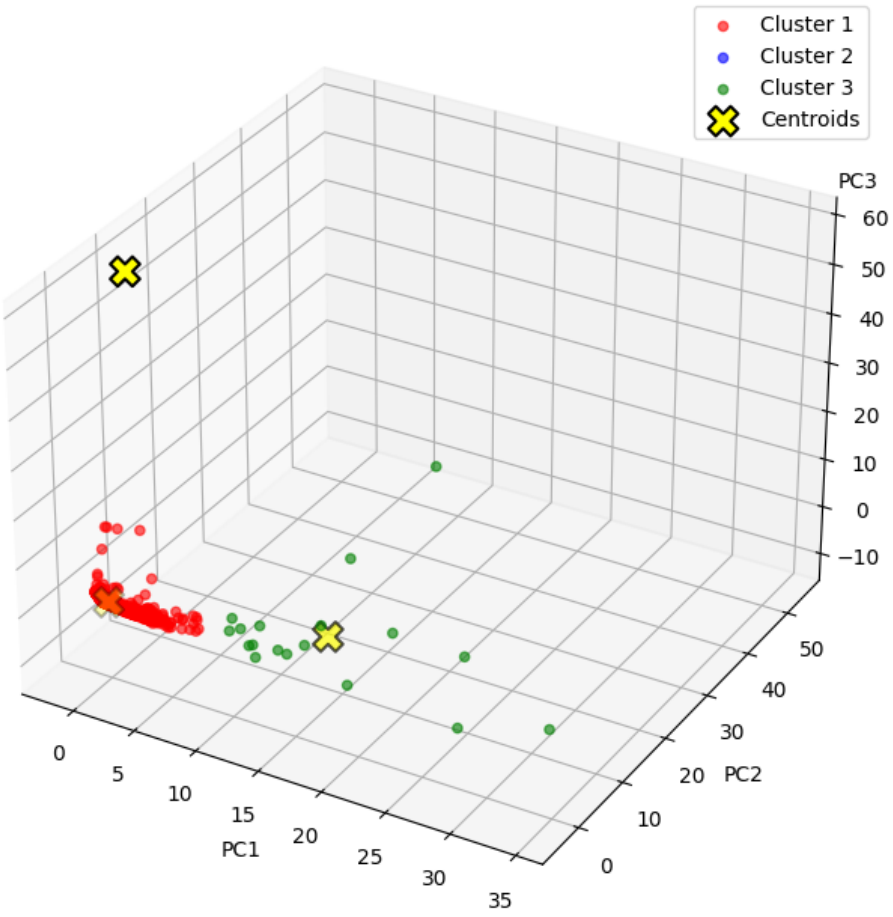


Figure 3.2 Number of data and visualization when k=3 (after PCA)

Although K-means on the PCA-reduced data achieves a high Silhouette Score, the actual clustering results are highly unbalanced (4320 / 1 / 18). This suggests that PCA compressed important customer-behavior information, causing the variation within the middle group to collapse and making it difficult for K-means to identify meaningful customer segments. As a result, the clusters do not align with real business patterns. Therefore, this approach is not suitable as the final customer segmentation model.

3.3 Clustering with the standardized features without PCA

First, we compute the Silhouette Scores for *k* values from 2 to 10 using the six-dimensional standardized data for comparison:

k	2	3	4	5	6	7	8	9	10
score	0.446	0.470	0.466	0.466	0.466	0.510	0.470	0.502	0.399

Figure 3.3: The Silhouette Score of k=2-10 (6-D standardized data without PCA)

Based on the Silhouette Scores from k = 2 to 10, k = 3 (0.470) is among the higher values and is consistent with the scores of nearby cluster counts. Although k = 6 (0.510) and k = 9 (0.502) are slightly higher, using too many clusters tends to over-segment the data, making the results harder to interpret and less meaningful in a business context. In contrast, k = 3 provides a good balance between clustering quality and interpretability, forming clearer and more practical customer groups. Therefore, we choose k = 3 as the number of clusters.

Next, we examine the number of data points in each cluster for k = 3:

Cluster ID	1	2	3
Quantity	3180	1140	18

Figure 3.4 Number of data when k=3 (6-D standardized data without PCA)

After applying K-means with k = 3 on the six-dimensional standardized features, the three clusters show a reasonable distribution: Cluster 1 contains 3,181 customers, Cluster 2 contains 1,140 customers, and Cluster 3 forms a small group of 18 customers. Compared to the extremely unbalanced PCA-based clustering results (4321 / 1 / 18), the six-dimensional feature clustering better reflects actual customer behavior and provides clearer business meaning.

Next, we examine the statistical characteristics of the six features across the three clusters and visualize them using boxplots:

Recency_scaled					
cluster	mean	med.	std.	min.	max.
1	0.527	0.635	0.339	0.915	0.875
2	1.482	1.445	0.705	0.455	2.815
3	0.681	0.895	0.759	0.915	2.335

Frequency_scaled					
cluster	mean	med.	std.	min.	max.
1	0.077	0.165	0.749	0.425	10.608
2	0.352	0.425	0.183	0.425	3.858
3	8.675	6.909	7.679	0.425	26.702

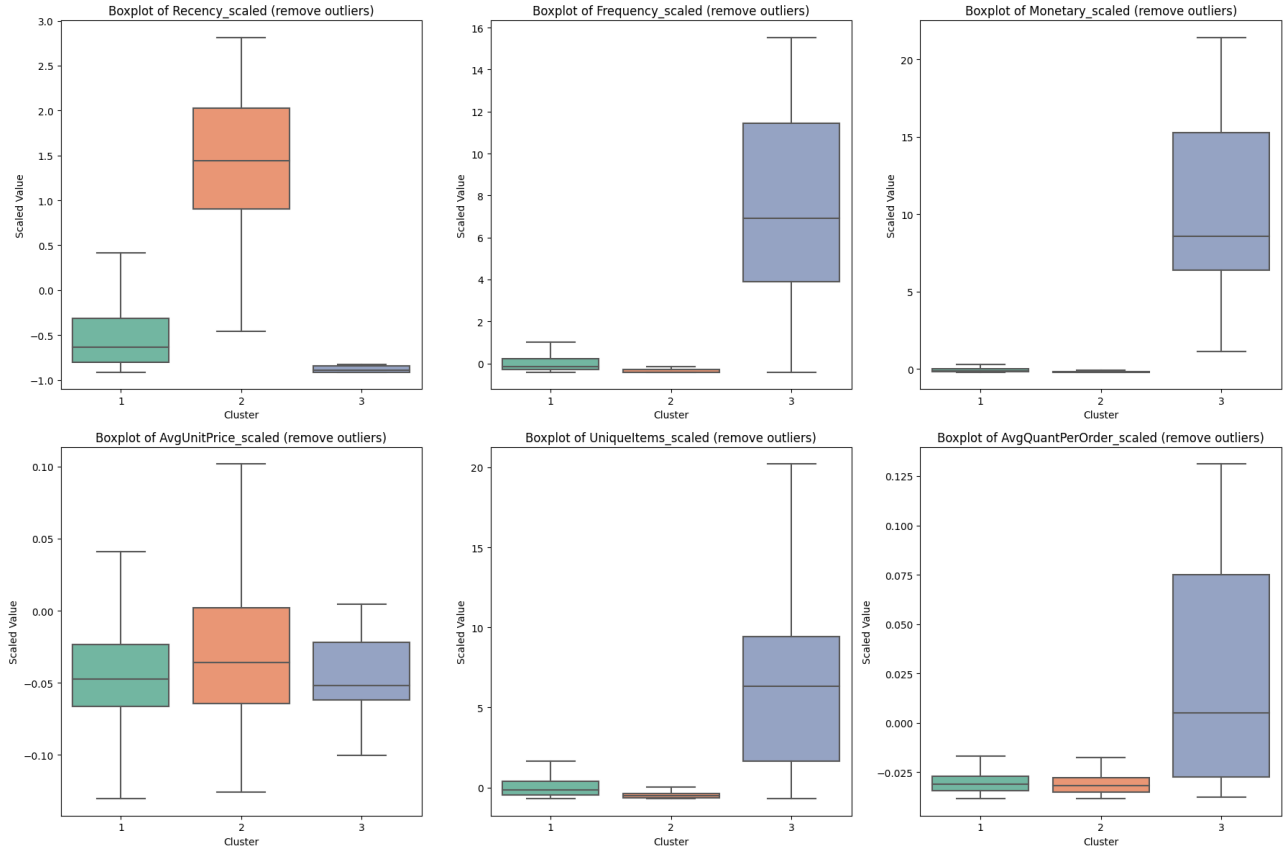
Monetary_scaled					
cluster	mean	med.	std.	min.	max.
1	0.006	0.123	0.430	0.229	7.188
2	0.167	0.194	0.208	0.228	4.727
3	11.680	8.573	8.453	1.124	30.950

AvgUnitPrice_scaled					
cluster	mean	med.	std.	min.	max.
1	0.034	0.047	0.117	0.131	3.086
2	0.095	0.036	1.939	0.126	59.311
3	0.009	0.052	0.231	0.100	0.923

UniqueItems_scaled					
cluster	mean	med.	std.	min.	max.
1	0.119	0.158	0.866	0.709	7.692
2	0.443	0.521	0.273	0.709	1.131
3	7.027	6.327	6.483	0.709	20.216

AvgQuantPerOrder_scaled					
cluster	mean	med.	std.	min.	max.
1	0.021	0.031	0.191	0.039	10.256
2	0.015	0.032	0.147	0.039	3.491
3	4.625	0.005	14.975	0.037	60.892

Figure 3.5: Statistical summary of each cluster across features



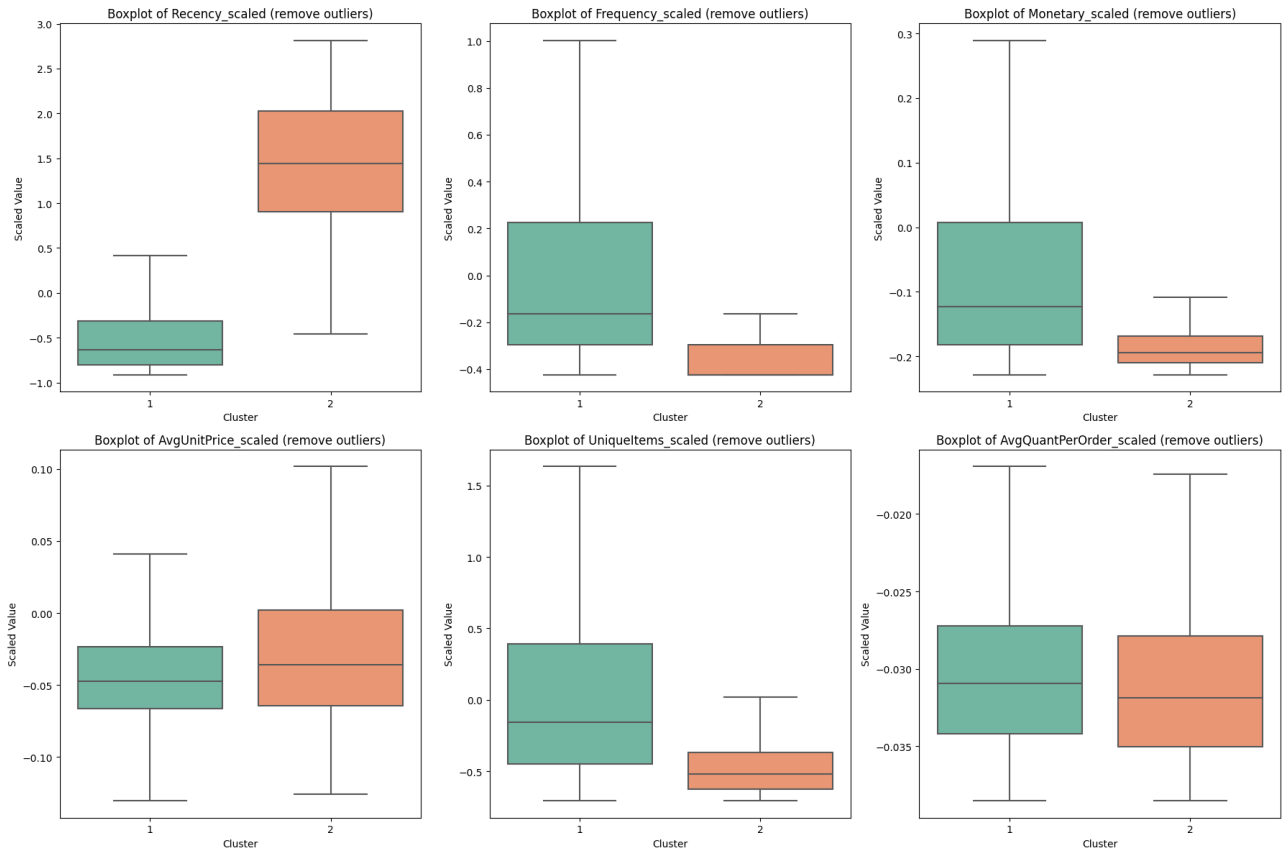


Figure 3.6: Boxplot of each cluster across features

Cluster 1	
Recency	Mostly above average, meaning these customers purchased recently.
Frequency	Purchase counts are around the average and slightly left-skewed.
Monetary	Spending is slightly below average and left-skewed.
AvgUnitPrice	Similar across all three clusters.
UniqueItems	Concentrated near the average.
AvgQuantPerOrder	Slightly below average and close to Cluster 2.

Cluster 2	
Recency	Highest among all clusters, have not purchased for a long time.
Frequency	Lowest among all clusters, showing infrequent transactions.
Monetary	Lowest among all clusters, with minimal total spending.
AvgUnitPrice	Similar across all three clusters.
UniqueItems	Lowest among all clusters, meaning they buy fewer types of products.
AvgQuantPerOrder	Slightly below average and close to Cluster 1.

Cluster 3	
Recency	Lowest among all clusters, meaning these customers purchased very recently.
Frequency	Highest among all clusters, indicating extremely frequent transactions.
Monetary	Highest among all clusters, with extremely large spending.
AvgUnitPrice	Similar across all three clusters.
UniqueItems	Highest among all clusters, showing purchases across a wide range of product types.
AvgQuantPerOrder	Highest among all clusters, reflecting large-quantity purchases per order.

Across the three customer groups, we can see clear differences in behavior:

- Cluster 1 is made up of regular shoppers who buy recently, spend around the average, and purchase a moderate range of items.
- Cluster 2 includes inactive, low-value customers who have not bought anything for a long time, spend very little, and only make a few small purchases.
- Cluster 3 consists of high-value bulk buyers who purchase very recently, buy very often, spend a lot, choose many different products, and place large orders.

These three groups can therefore be labeled as Regular Shoppers (Cluster 1), Inactive Low-Value Customers (Cluster 2), and High-Value Bulk Buyers (Cluster 3).

3.4 Clustering Quality Evaluation

Using the six standardized features, K-means with k = 3 gives a Silhouette Score of 0.470, showing that the clusters are fairly tight within each group and well separated from each other. The cluster sizes (3181, 1140, and 18) make sense based on customer behavior and also match real business patterns. The boxplots show clear differences across the three groups in spending, purchase frequency, product variety, and order quantity, meaning the clusters have clear boundaries. Overall, k = 3 provides a stable and easy-to-interpret clustering result for this dataset.

4. Supervised Learning Methods

This section covers supervised learning methods. Using the cluster labels generated from the earlier K-means analysis, we split the data into training and test sets, then built decision boundaries using LDA, KNN, and Naïve Gaussian models. Finally, we compared the prediction results of each model.

4.1 Proposed Methods

4.1.1 LDA

Under the assumption $p(\vec{x}|w_i) \sim N(\vec{\mu}_i, \Sigma)$, we derive the discriminant function:

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_i) + \ln P(\omega_i)$$
$$= \vec{w}_i^T \vec{x} + w_{i0}$$
$$where\ \vec{w}_i = \Sigma^{-1}\vec{\mu}_i,\ \ w_{i0} = -\frac{1}{2}\vec{\mu}_i^T \Sigma^{-1}\vec{\mu}_i + \ln P(\omega_i)$$

The decision boundary between class i and class j:

$$\vec{w}^T(\vec{x} - \vec{x}_0) = \vec{0},$$
$$where\ \vec{w} = \Sigma^{-1}(\vec{\mu}_i - \vec{\mu}_j), \vec{x}_0 = \frac{1}{2}(\vec{\mu}_i + \vec{\mu}_j) - \frac{\ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right]}{(\vec{\mu}_i - \vec{\mu}_j)^T \Sigma^{-1}(\vec{\mu}_i - \vec{\mu}_j)}(\vec{\mu}_i - \vec{\mu}_j)$$

4.1.2 KNN

Given a training dataset:

$$\mathrm{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\}$$
$$where\ \vec{x}_i \in \mathbb{R}^d : d - dimensional\ feature\ vector,$$
$$y_i \in \{\omega_1, \dots, \omega_C\} : class\ label.$$

For a sample $\vec{x}$ to be classified, we compute its Euclidean distance to all training points $\vec{x}_i$. Select the k nearest neighbors, and determine the class by majority vote. Formally, if $N_k(x)$ denotes the set of the k nearest neighbors of $\vec{x}$, then:

$$\hat{\omega}(x) = \underset{\omega_c}{argmax} \sum_{x_j \in N_k(x)} I_{\{y_j = \omega_c\}}$$

4.1.3 Naïve Gaussian

Naïve Bayes assumes that all features are conditionally independent within the same class:

$$p(x|\omega_i) = \prod_{j=1}^{d} p(x_j|\omega_i)$$

Naïve Gaussian further assumes that each feature follows a one-dimensional normal distribution:

$$x_j|\omega_i \sim N(\mu_{ij}, \sigma_{ij}^2)$$

$$p(x_j|\omega_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(-\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2}\right)$$

Under above assumption, we derive the discriminant function:

$$g_i(x) = \sum_{j=1}^{d} \left[-\frac{1}{2}\ln\left(2\pi\sigma_{ij}^2\right) - \frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2}\right]$$

4.2 Concepts and Evaluation Basis

In this case, it is important to recognize that most decision-making costs will be invested in high-value customers. Every high-value customer can be critical to the company's success, so correctly identifying the High-Value Bulk Buyers (Cluster 3) is the top priority.

- If a high value customer is misclassified into another group the company may overlook an important opportunity to maintain engagement with that customer.
- If a low or medium value customer is misclassified as high-value, it may distort insights and lead to suboptimal resource planning.

Since there are only 18 high-value customers in our dataset, overall accuracy is not an appropriate metric. For example, a model could classify all C1 and C2 samples correctly but misclassify all C3 samples, and still achieve a high accuracy score. Such a model would be unacceptable. We need to place greater emphasis on the performance of Cluster 3, so we focus on the following as an evaluation of each model:

1. Confusion matrix: (a_{ij}: actual class i, predicted as class j)

Actual \ Pred	Pred C1	Pred C2	Pred C3
Actual C1	a11	a12	a13
Actual C2	a21	a22	a23
Actual C3	a31	a32	a33

(a) Recall: Among all samples that truly belong to class Ci, the proportion that is correctly classified.

$$Recall_i = \frac{aii}{ai1 + ai2 + ai3}$$

(b) Precision: Among all samples predicted as class Ci, the proportion that actually belongs to Ci.

$$Precision_i = \frac{aii}{a1i + a2i + a3i}$$

2. Macro-average F1 score

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C} F1_i \ , where\ F1_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}$$

3. Balanced accuracy score

$$Balanced\ Accuracy = \frac{1}{C}\sum_{i=1}^{C} Recall_i$$

4.3 Implementation and Results

We split the data within each cluster into 60% for training and 40% for testing. The results of the different models are presented below.

1. LDA

Confusion matrix:

$$\begin{bmatrix} 1269 & 4 & 0 \\ 42 & 414 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

Classification report:

	precision	recall	F1-score	support
C1	0.97	1.00	0.98	1273
C2	0.99	0.91	0.95	456
C3	1.00	1.00	1.00	7
Acc.				0.97
Macro Avg. F1.				0.98
Balanced Acc.				0.97

2. KNN (Set k=1, the highest Macro Avg. F1 and Balanced Acc.)

Confusion matrix:

$$\begin{bmatrix} 1268 & 5 & 0 \\ 6 & 450 & 0 \\ 2 & 0 & 5 \end{bmatrix}$$

Classification report:

	precision	recall	F1-score	support
C1	0.99	1.00	0.99	1273
C2	0.99	0.91	0.99	456
C3	1.00	0.71	0.83	7
Acc.				0.99
Macro Avg. F1.				0.94
Balanced Acc.				0.90

3. Naïve Gaussian

Confusion matrix:

$$\begin{bmatrix} 1204 & 45 & 24 \\ 10 & 446 & 0 \\ 1 & 0 & 6 \end{bmatrix}$$

Classification report:

	precision	recall	F1-score	support
C1	0.99	0.95	0.97	1273
C2	0.91	0.98	0.94	456
C3	0.2	0.86	0.32	7
Acc.				0.95
Macro Avg. F1.				0.74
Balanced Acc.				0.93

We can observe that among the three models:

- LDA performs the best, it makes no mistakes in identifying high-value customers, and it also delivers strong results for both C1 and C2.
- KNN misclassifies 2 C3 samples as C1.
- Naïve Gaussian performs the worst, it misclassifies 24 C1 samples as C3, and additionally misclassifies 1 C3 sample as C1.

Discussion

- **LDA**

  The three clusters exhibit strong directional differences that make them close to linearly separable. Cluster 3 differs dramatically from the other two groups across Recency, Frequency, Monetary, UniqueItems, and AvgQuantPerOrder. Likewise, C1 and C2 also show clear gaps in Recency, Frequency, Monetary, and UniqueItems. These differences create linear separability between clusters, allowing LDA to achieve excellent performance.

- **KNN**

  KNN relies purely on Euclidean distance. As a result, if a data point happens to be close to another class in certain dimensions, it can easily be misclassified.

- **Naïve Gaussian**

  Naïve Gaussian treats each feature independently, so like KNN, it is also sensitive to dimensions where samples appear close to other classes. Additionally, the assumption that all features are independent within a class is unrealistic for this dataset because many customer behaviors are inherently correlated. This mismatch between the model assumption and the actual data structure leads to its overall weakest performance.

4.4 Conclusion

In this project, we accomplished the following:

1. We cleaned the dataset and visualized the features, discovering that all of them exhibit varying degrees of right-skewed distributions.
2. We standardized the dataset and applied PCA for dimensionality reduction and found that three principal components capture 72.55% of the total variance.
3. Using K-means clustering, we generated labels and, based on the PCA insights, silhouette scores, cluster sizes, and business interpretation, determined that the six standardized features should be grouped into three clusters, which were then meaningfully labeled.
4. We compared LDA, KNN, and Naïve Gaussian, and found that LDA performs the best for this six-dimensional, right-skewed dataset where clusters show strong directional separation and the minority high-value group carries significant importance.