ESADE MIBA

Presentation By Raissa Angnged

Online Retail Customer Analysis

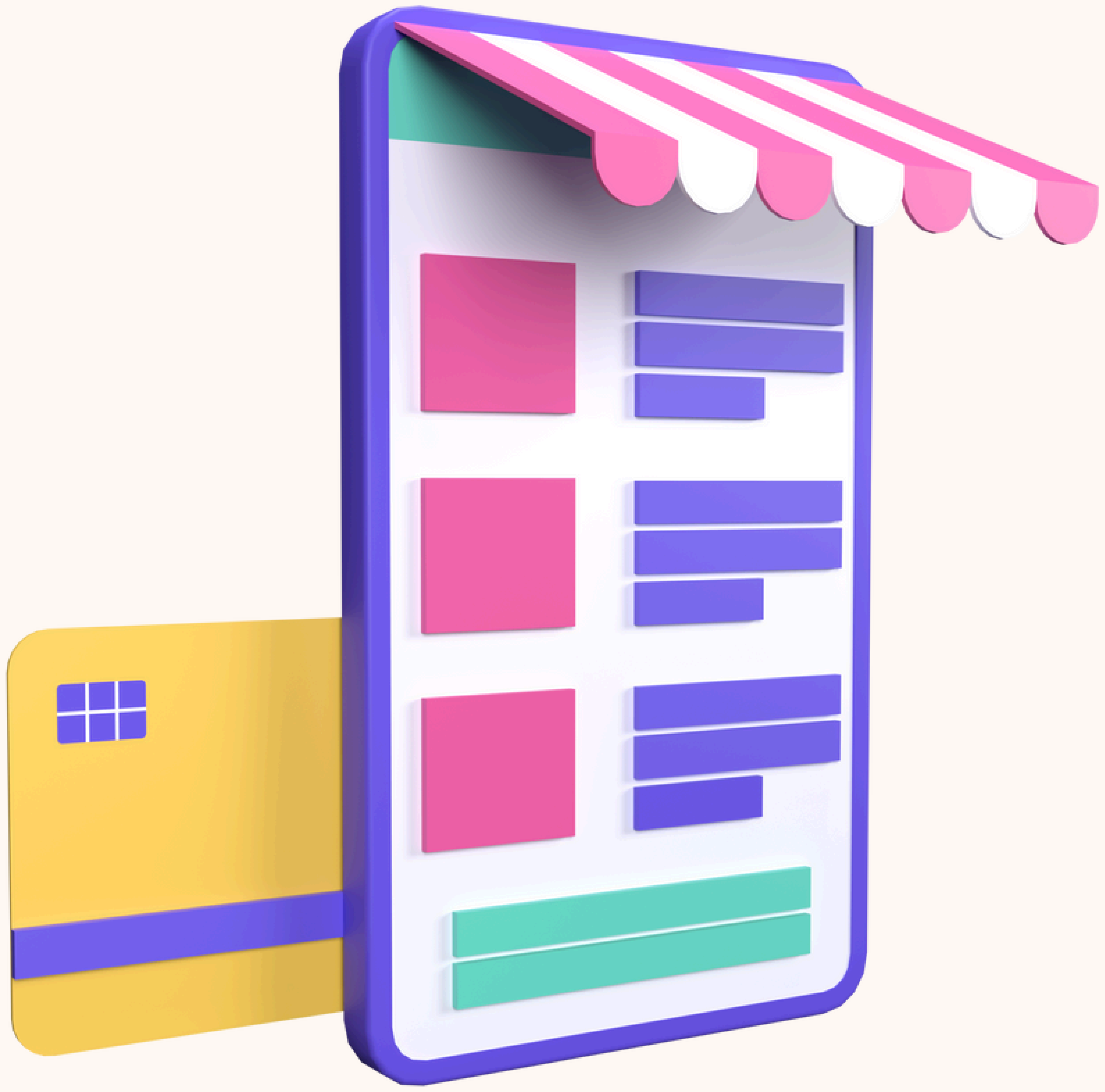
Unsupervised Learning for Clustering Customer Segments

Introduction & Business Context

- This analysis is for an **Online Retail company to optimize marketing and customer retention** by segmenting customers based on purchasing behavior

- The goal of this project is to apply **unsupervised learning techniques to identify distinct customer groups** and implement more effective marketing strategies

- **Machine learning enables data-driven segmentation**, enhancing marketing effectiveness and overall business profitability



Overview of ML Lifecycle

1. Define Goals

- To leverage machine learning for customer segmentation, enabling improved and personalized strategies

2. Prepare Data

- Explore the data
- Clean the data
- Analyze correlations
- Feature engineering
- and transformation

3. Create Model

- Test clustering algorithms (KMeans, Hierarchal, and DBSCAN)
- Choose optimal number of clusters

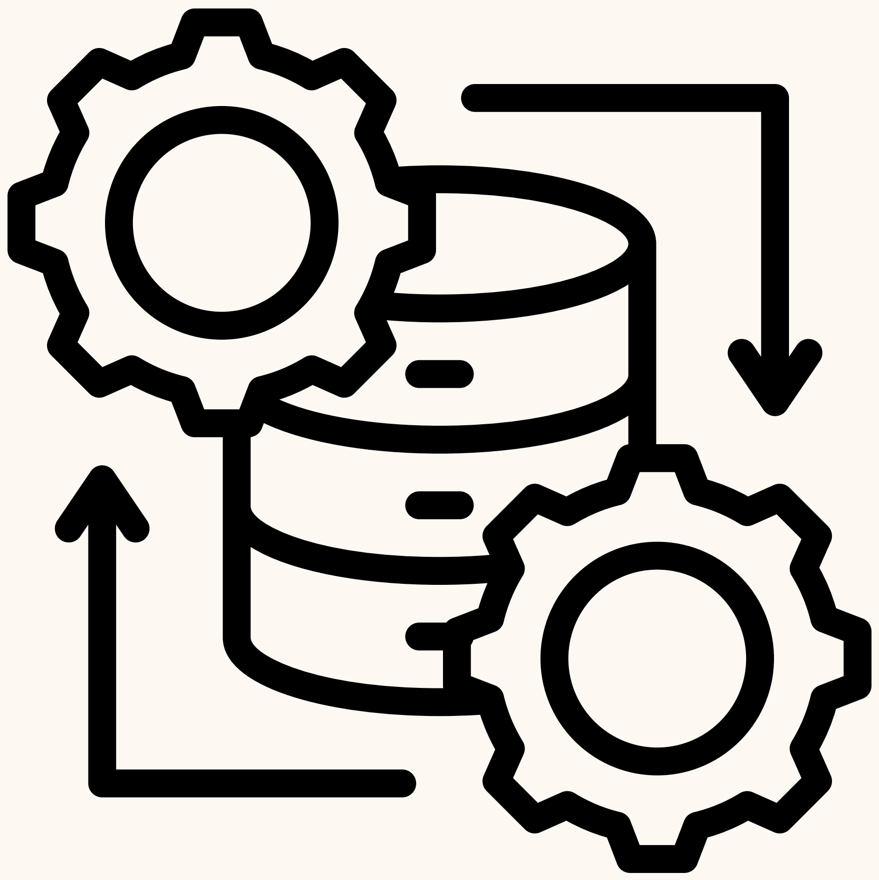
4. Interpret Model

- Understand the resulting customer segments
- Interpret insights from the cluster characteristisc

5. Implement Model

- Apply business knowledge in improving marketing strategies given the new insights

Data Prepatation

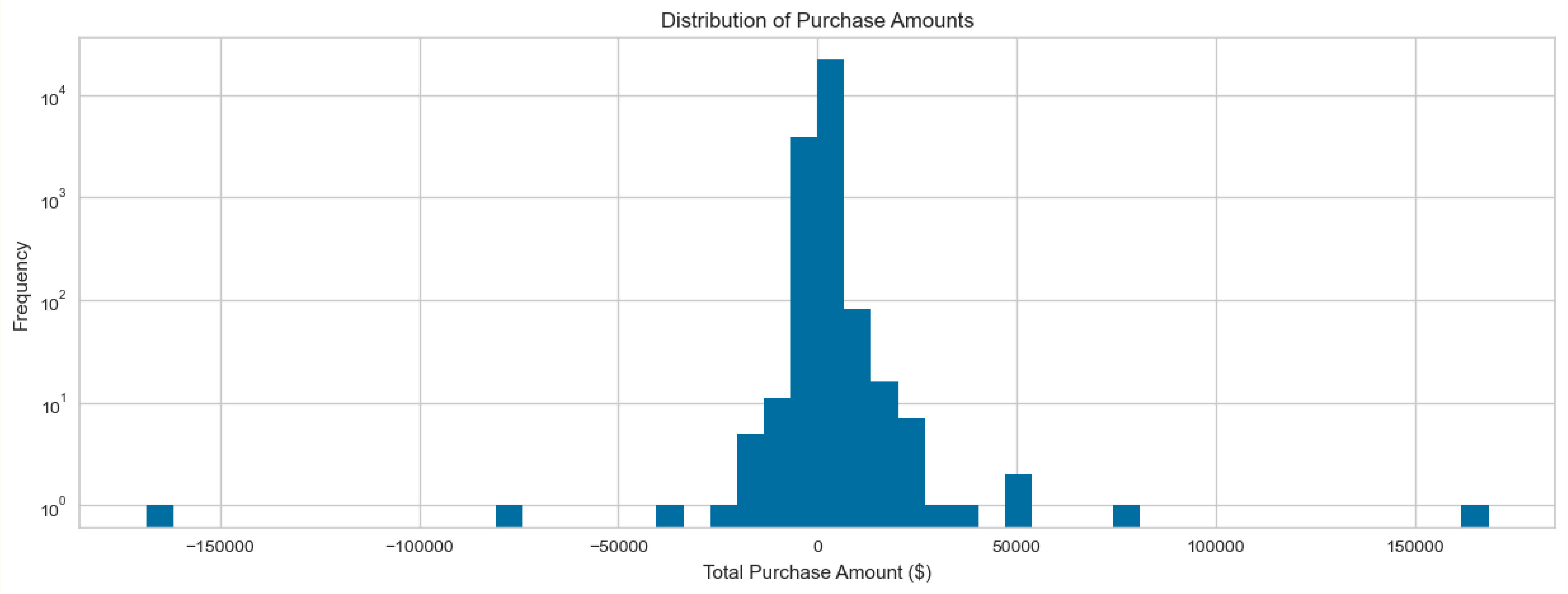


Overview of Dataset

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

- The dataset has information on details of an invoice, the product purchased, and the customer’s details
- Each row represents a purchase invoice
- There is a total of 541,909 rows and 8 columns

Data Exploration and Cleaning



- Looked into the distribution of each invoice's purchase amount
- Noticed there is a significant amount of negative quantities and price, likely from returns or cancellations
- Removed the transactions with negative quantities and price

Feature Engineering

- Created a dataframe for the customer features to be used in the clustering
- Feature dashboard includes newly created features, such as:
  - Total price = quantity * unit price
  - Invoice year, month, day, hour
  - Purchase frequency rate
  - Average items / spend per order
- Aggregated the customer IDs and assigned a single country per customer - based on most frequent country

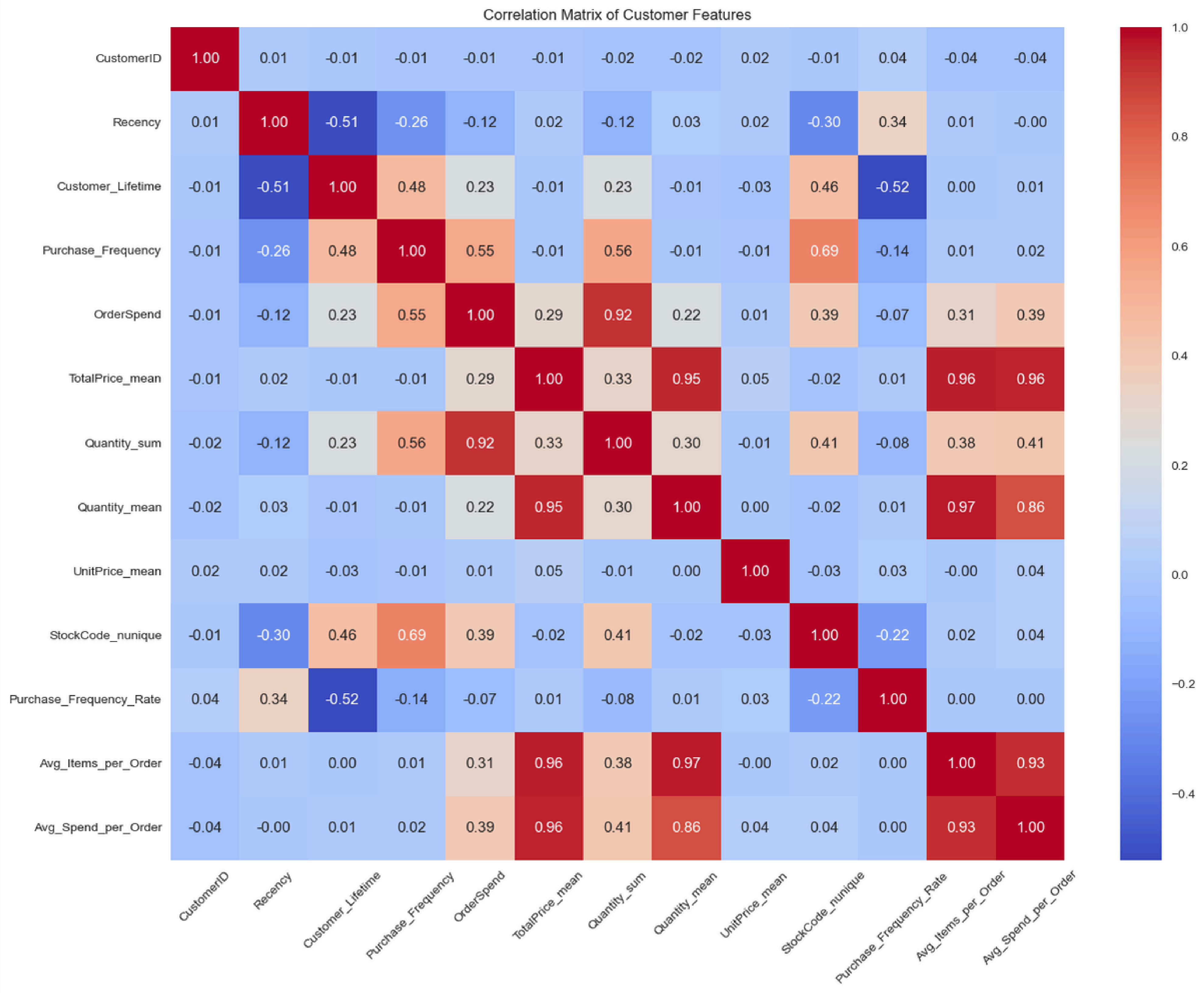
```python
df_clean['TotalPrice'] = df_clean['Quantity'] * df_clean['UnitPrice']
df_clean['Year'] = df_clean['InvoiceDate'].dt.year
df_clean['Month'] = df_clean['InvoiceDate'].dt.month
df_clean['Day'] = df_clean['InvoiceDate'].dt.day
df_clean['DayOfWeek'] = df_clean['InvoiceDate'].dt.dayofweek
df_clean['Hour'] = df_clean['InvoiceDate'].dt.hour

max_date = df_clean['InvoiceDate'].max()

#Create additional features
customer_features = df_clean.groupby('CustomerID').agg({
    'InvoiceDate': [lambda x: (max_date - x.max()).days,
                    lambda x: (x.max() - x.min()).days],
    'InvoiceNo': 'nunique',
    'TotalPrice': ['sum', 'mean', 'std'],
    'Quantity': ['sum', 'mean', 'std'],
    'UnitPrice': ['mean', 'std'],
    'StockCode': 'nunique'
})
```

```python
#Calculate more features
customer_features['Purchase_Frequency_Rate'] = customer_features['Purchase_Frequency'] / \
                                               customer_features['Customer_Lifetime'].clip(lower=1)
customer_features['Avg_Items_per_Order'] = customer_features['Quantity_sum'] / \
                                           customer_features['Purchase_Frequency'].clip(lower=1)
customer_features['Avg_Spend_per_Order'] = customer_features['OrderSpend'] / \
                                           customer_features['Purchase_Frequency'].clip(lower=1)

#Assign 1 country (most frequent) per customer
customer_features['Country'] = df_clean.groupby('CustomerID')['Country'].agg(lambda x: x.mode()[0])
```

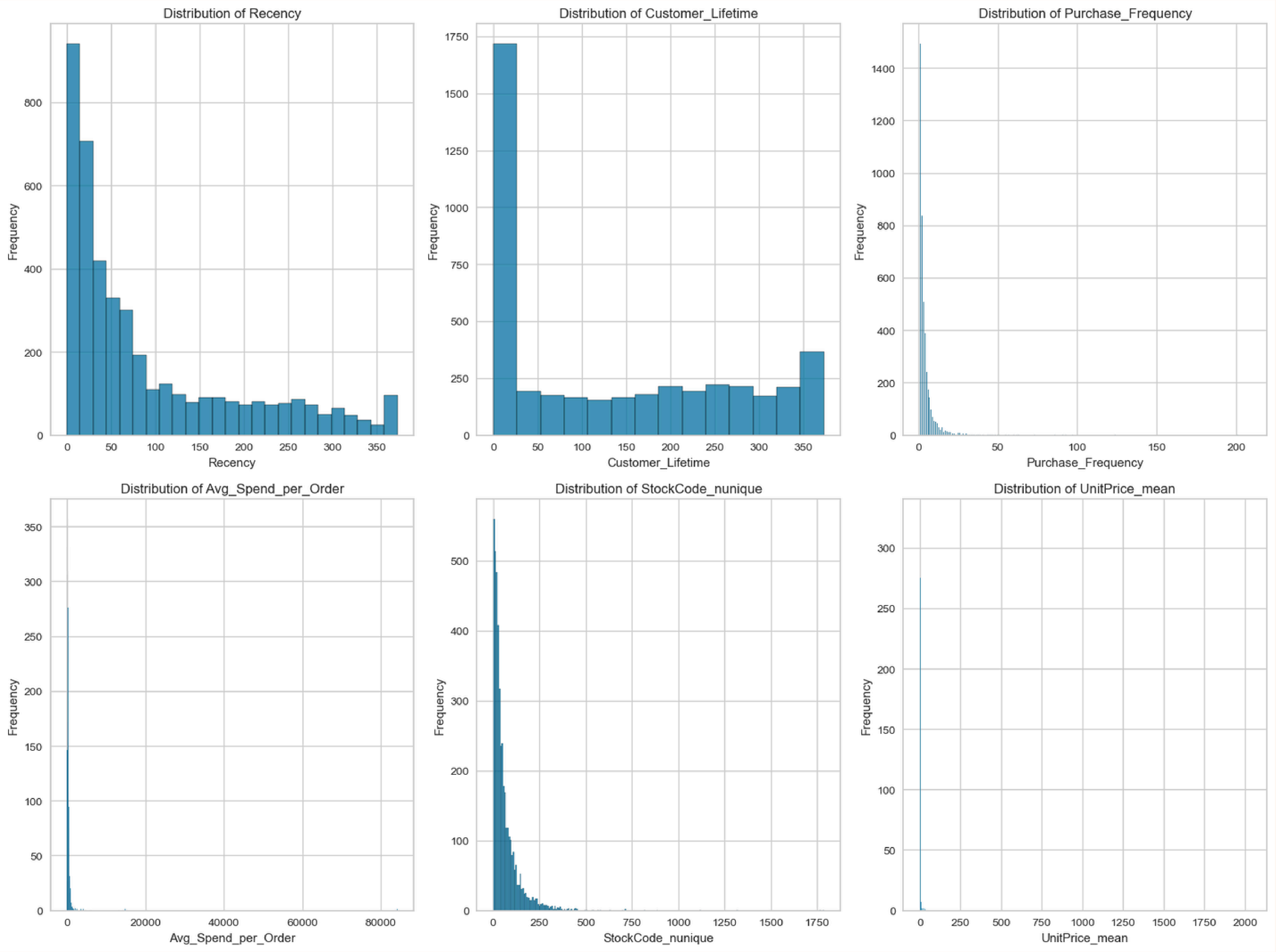


Feature Correlation

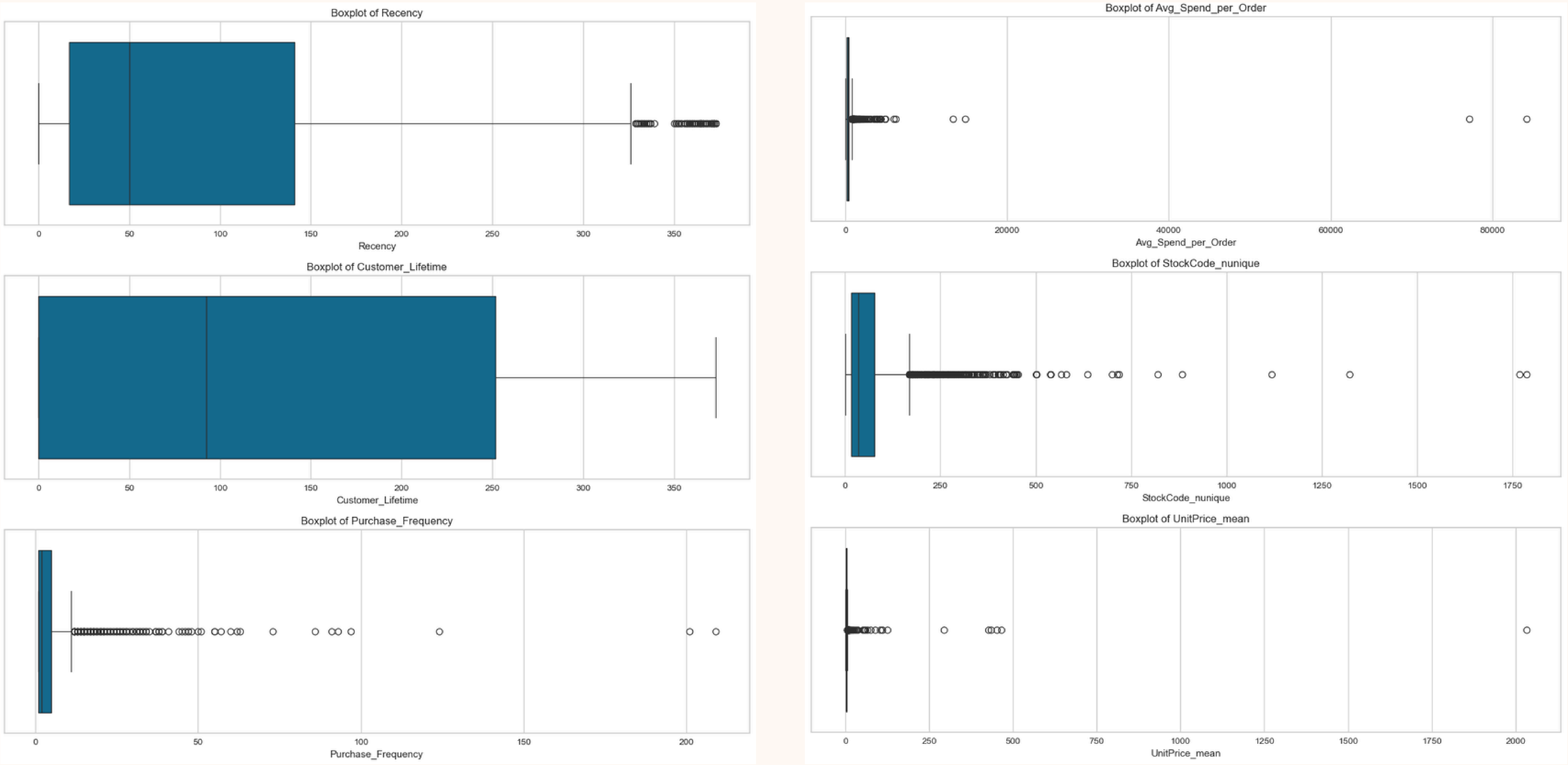
- Based on the correlation matrix, choose between highly correlated features
- From one's business knowledge, consider the features that makes most sense in forming customer segments, while still balancing the features
- Selected features
  - Recency
  - Customer Lifetime (frequency)
  - Purchase Frequency (frequency)
  - Avg Spend Per Order (monetary)
  - Unique stock code (product dimension)
  - Mean Unit Price (product dimension)

Data Transformation

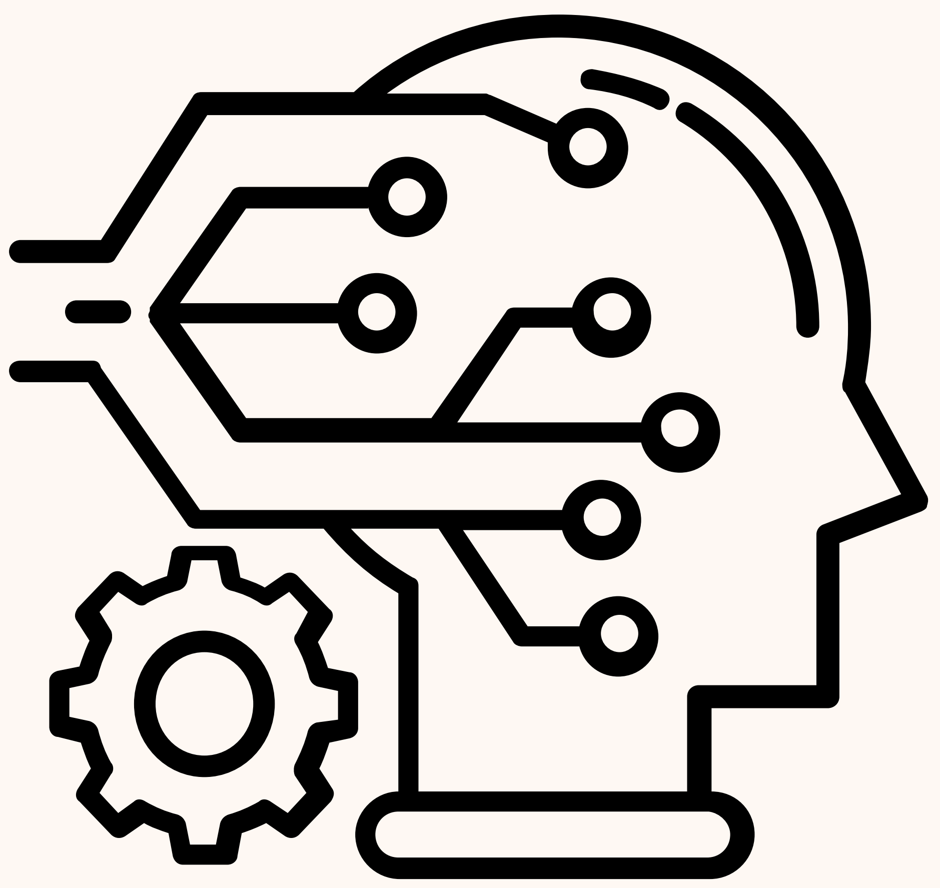
- Since most features have a right-skewed distribution, there are a small amount of high-value customers or transactions
- As such, it is important to employ retention strategies that are tailored to different customer segments
- Better understanding customer spending patterns can help with personalized marketing and product recommendations



Handling Outliers

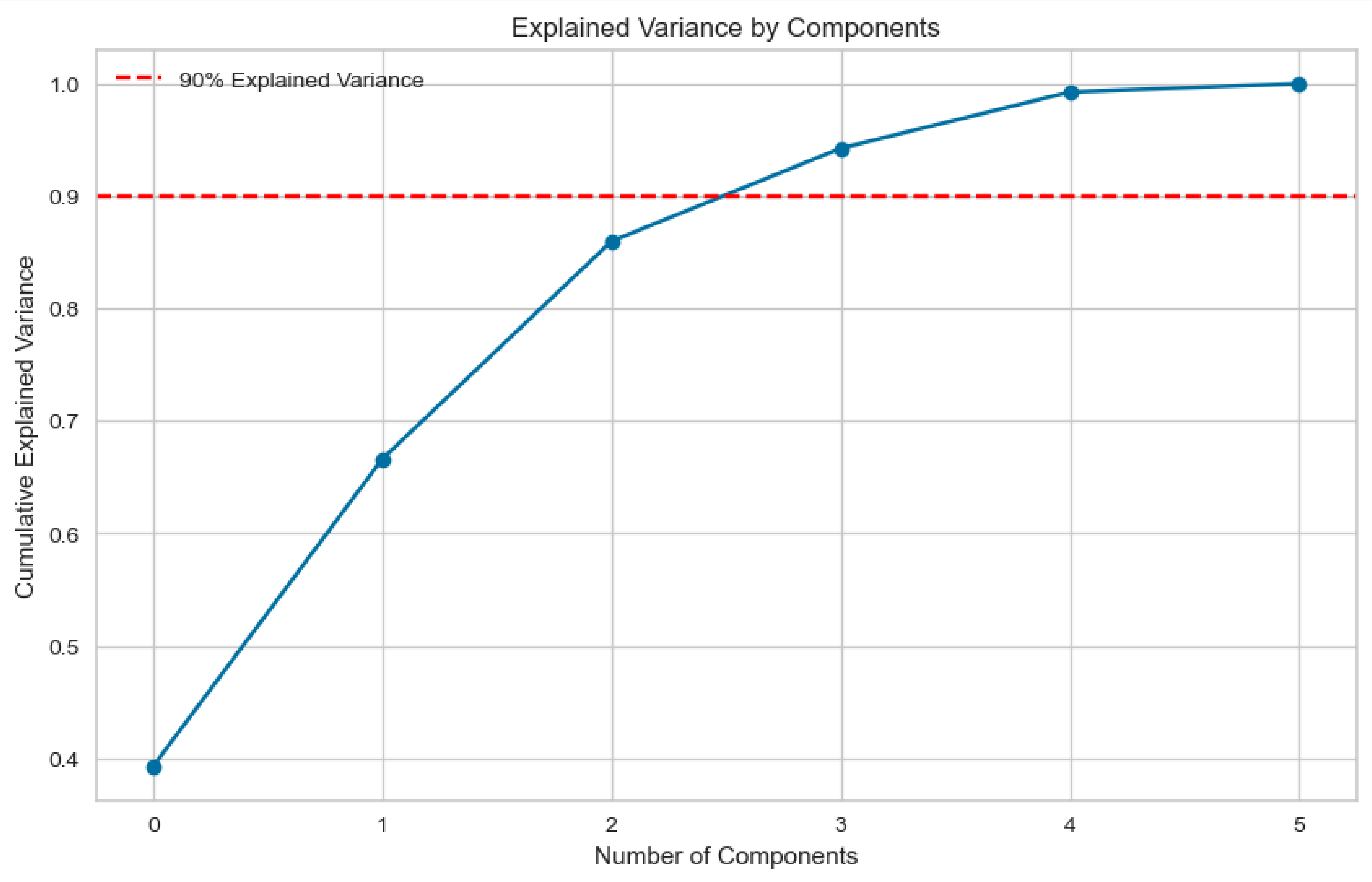


- Reduce skewness by applying log transformation to follow a more normal distribution
- Cap extreme outliers using percentiles to prevent distortion



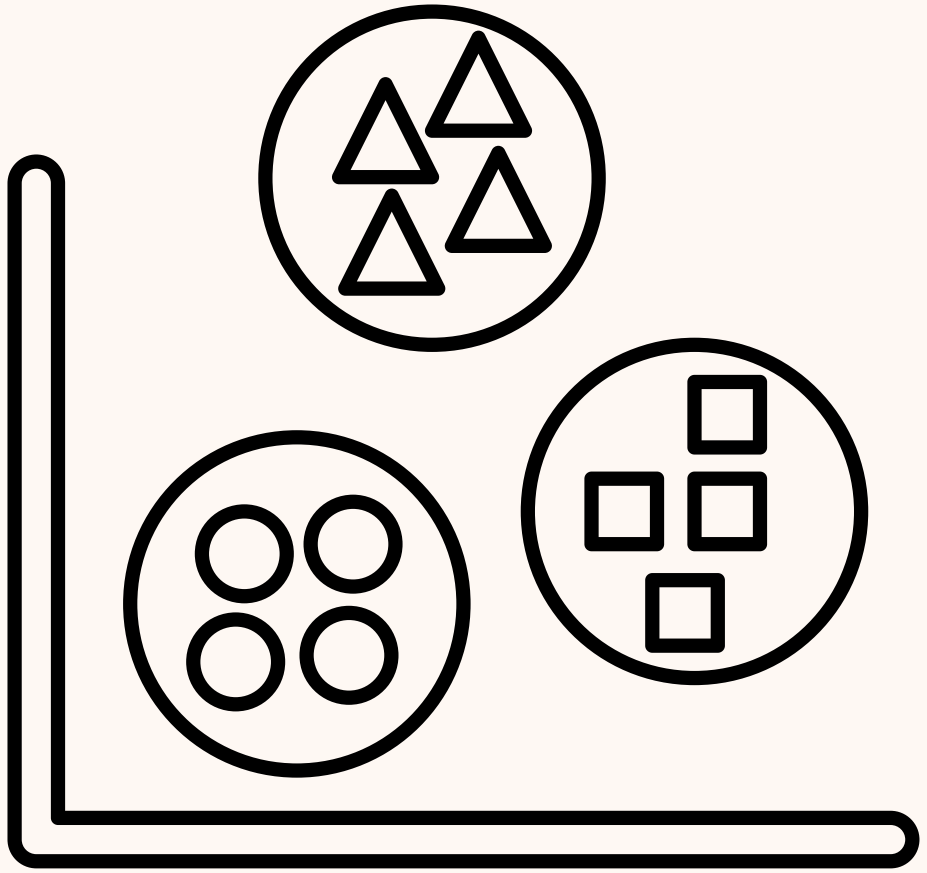
Model Creation

Principal Component Analysis (PCA)



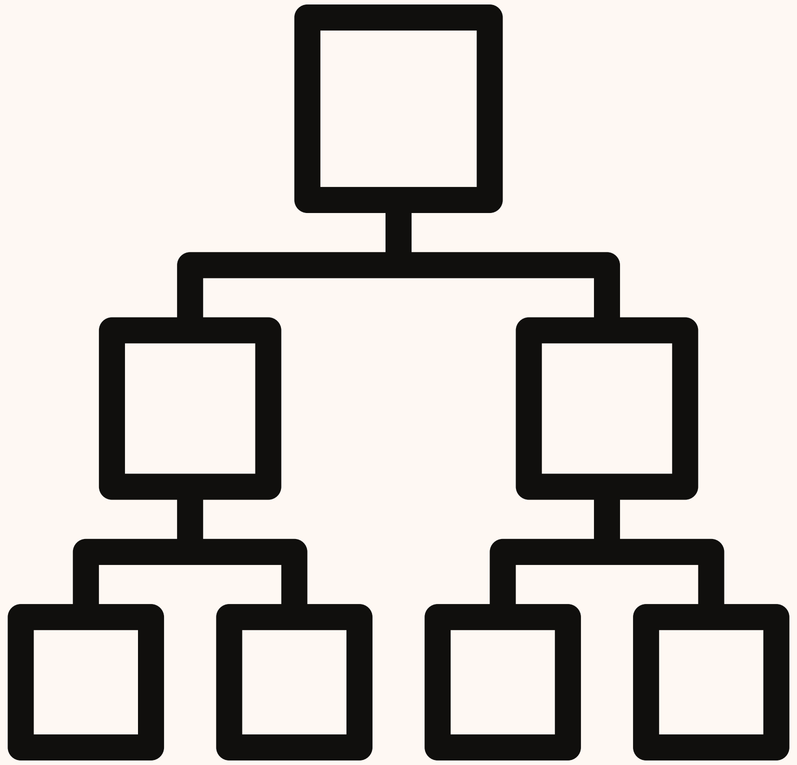
- Perform Principal Component Analysis (PCA) to reduce dimensionality while preserving data variance
- Reduce features to 4 components since this captures 95% of the total variance in the dataset
- Based on explained variance ratios, first component alone explains 39% of variance, with subsequent components contributing 33%, 19%, and 10% respectively - indicating a relatively balanced distribution of information across these components

Choosing a Clustering Algorithm



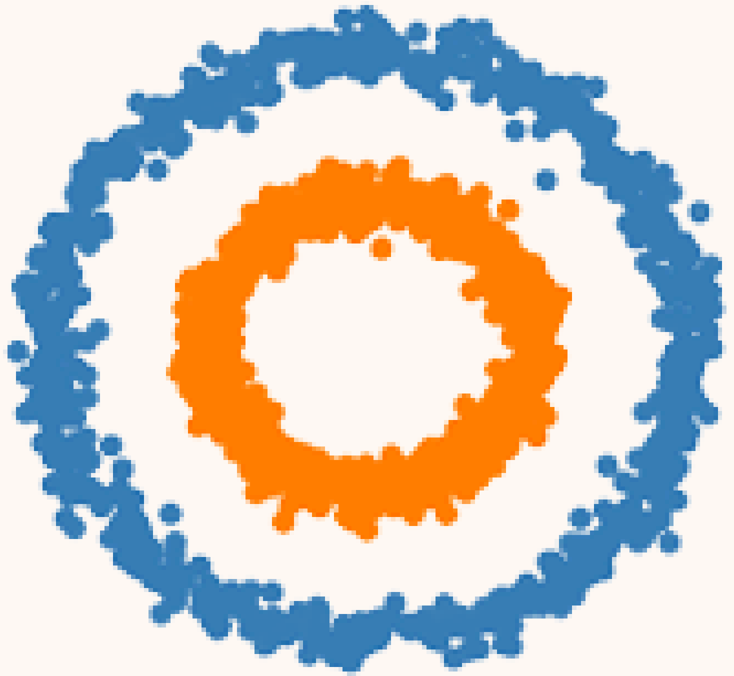
K-Means Clustering

- Model generates **balanced, interpretable clusters**
- Works well with large datasets and transformed features
- **Cluster centroids provide clear customer profiles**



Hierarchal Clustering

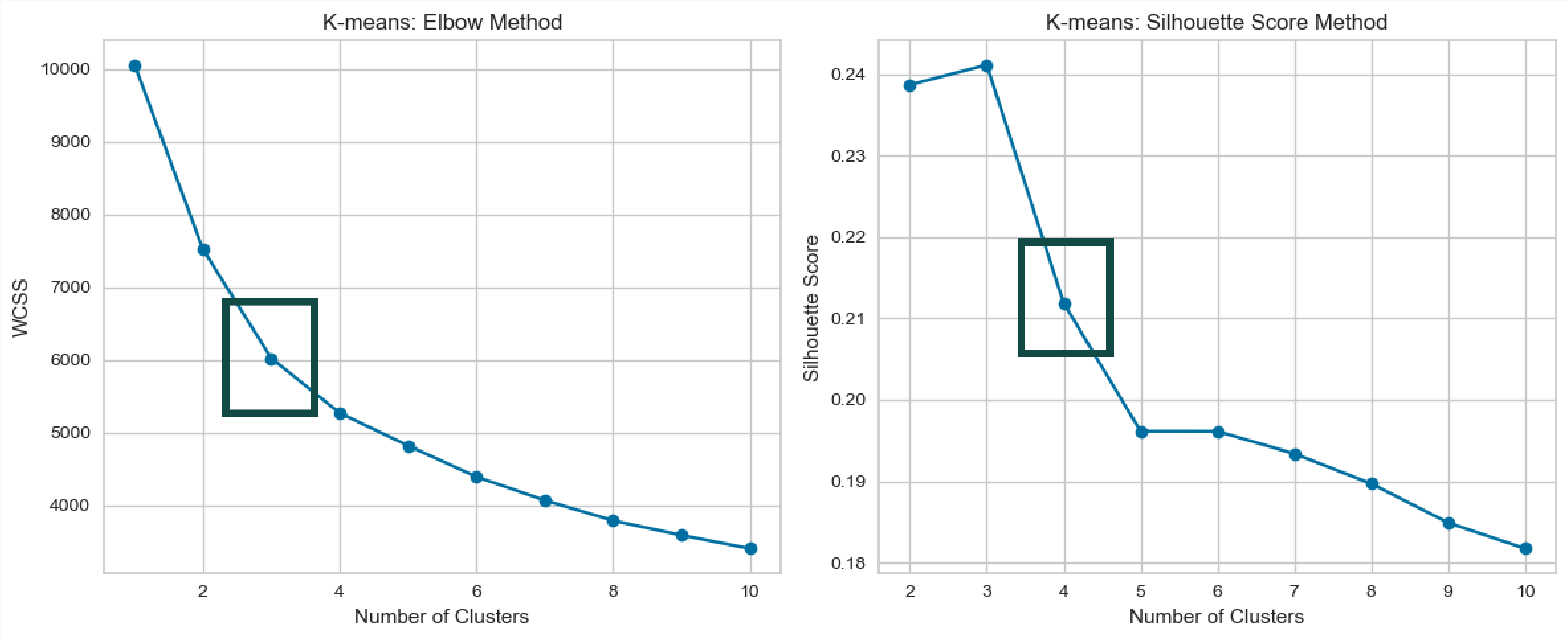
- Provides insight into **hierarchical relationships between customers**
- **More flexible** cluster shapes than K-means clustering
- No pre-specified cluster count



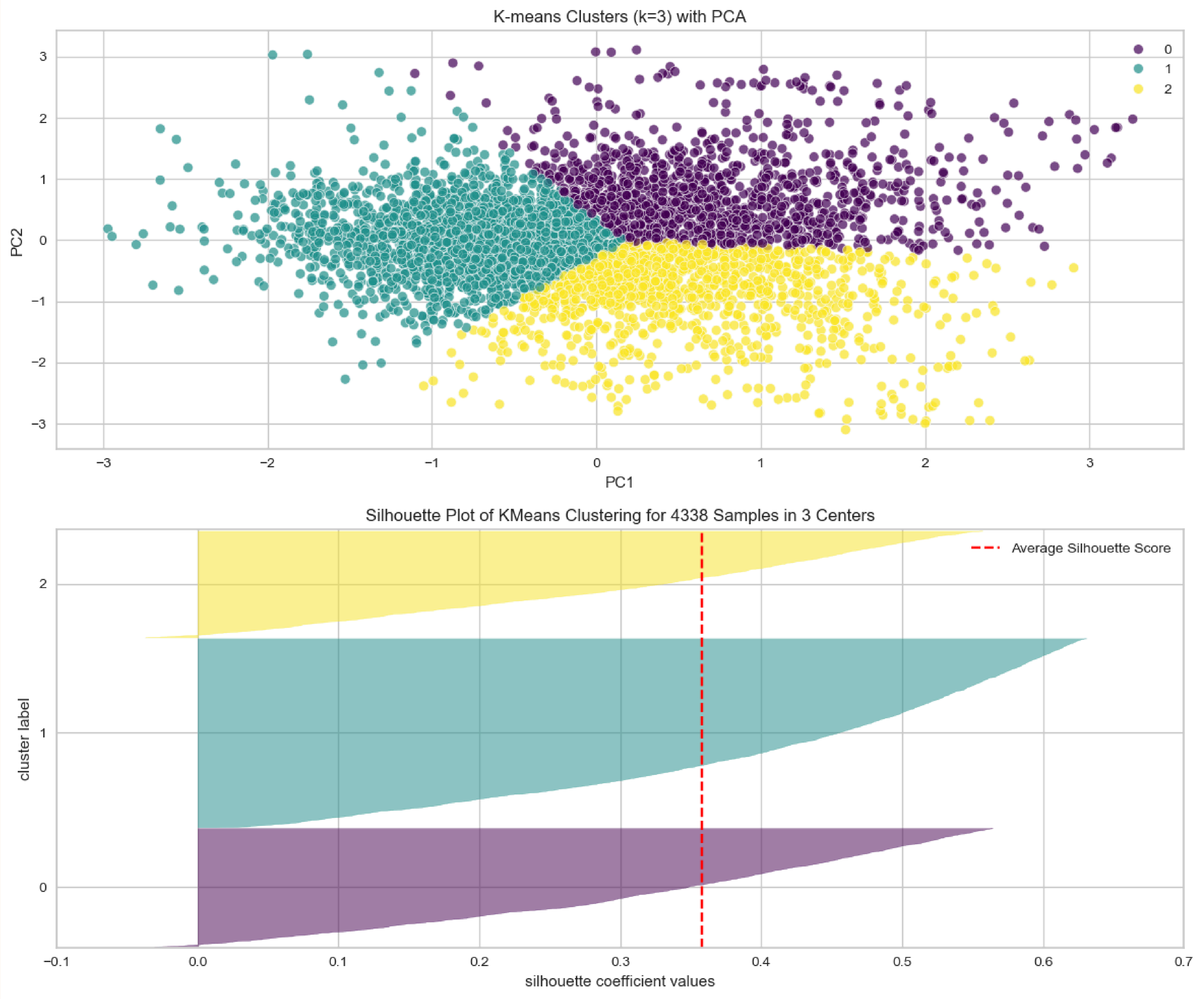
DBSCAN Clustering

- Identifies outliers (potentially high-value customers)
- Discovers **clusters of arbitrary shapes**
- Automatically determines number of clusters

K-Means Clustering



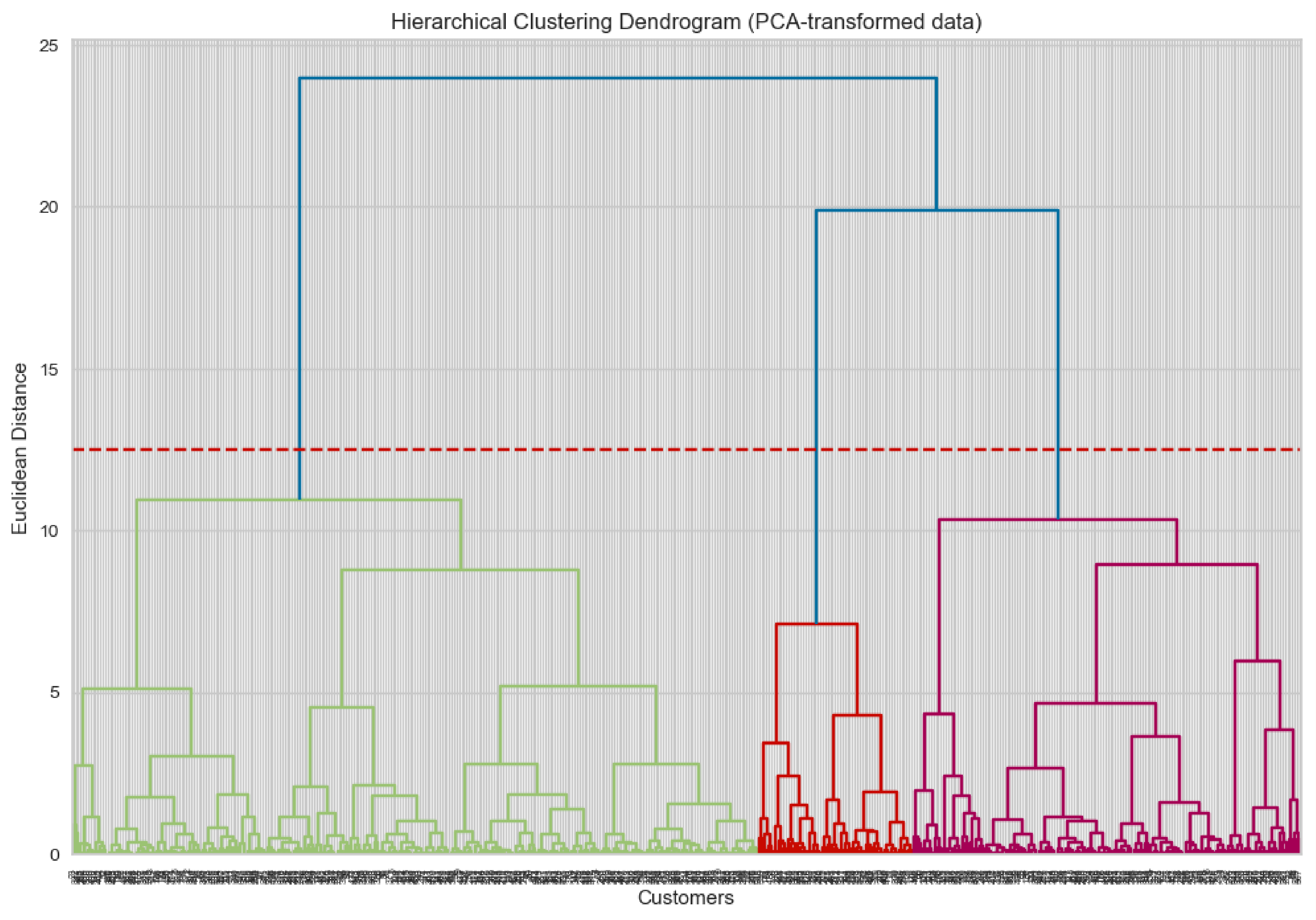
- Use elbow method and WCSS to determine e at which point adding more clusters doesn't significantly improve compactness
- Check silhouette score to ensure clusters are well-separated and prevent overlapping groups
- From the graphs above, the optinal number of clusters seem to be 3, balancing WCSS reduction and silhouette score



K-Means Clustering

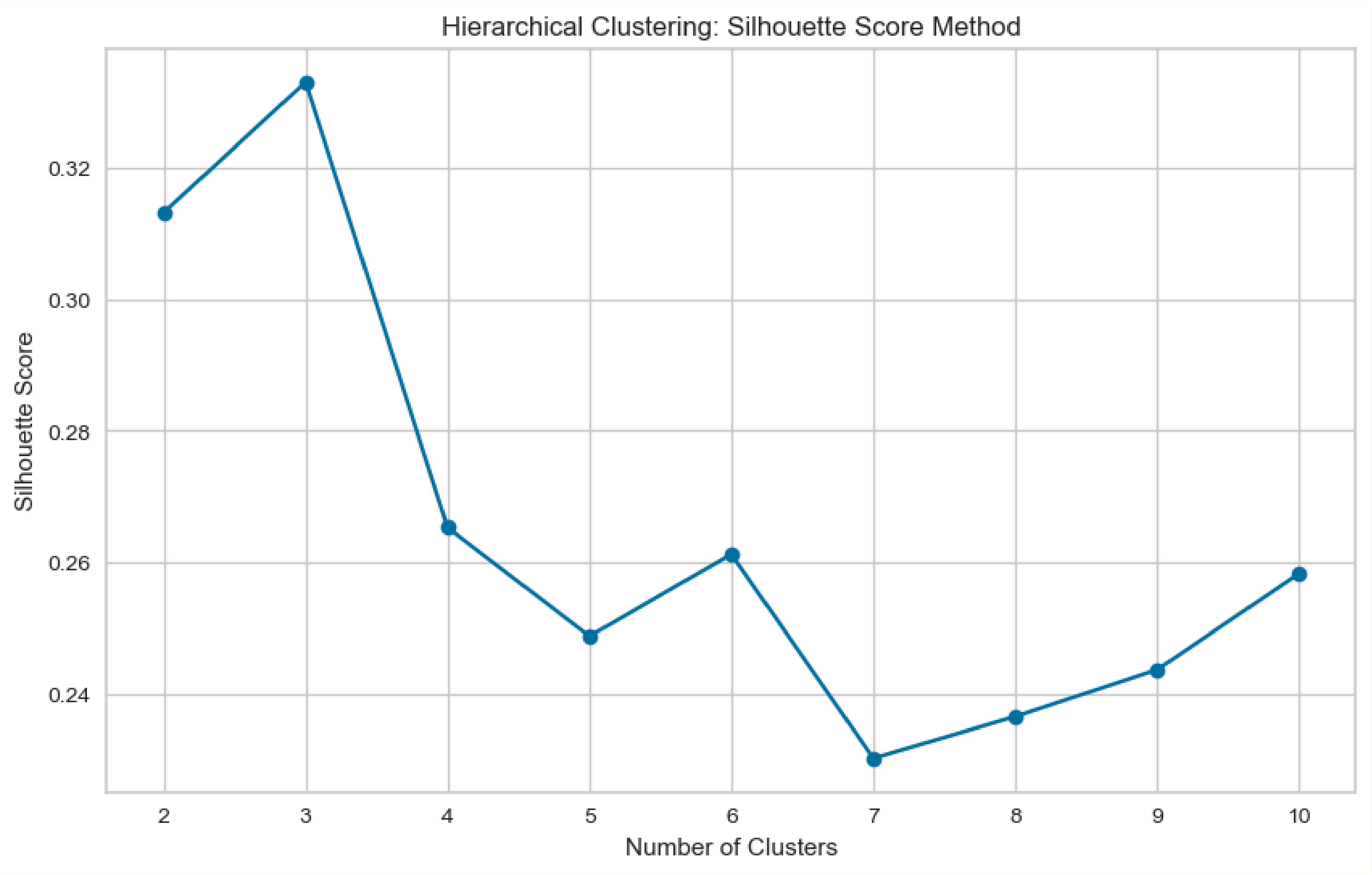
- K-means clustering identified 3 distinct customer segments in the dataset, with clear separation visible in the PCA visualization
- The yellow cluster shows the highest cohesion (with a silhouette score of around 0.4 to 0.5), representing our most well-defined customer segment with consistent purchasing behaviors
- All three clusters demonstrate positive silhouette coefficients, confirming the validity of our segmentation approach with an average silhouette score of ~0.35
- This three-segment model provides an optimal balance between statistical validity and practical application, allowing us to create targeted marketing strategies tailored to each customer group's unique characteristics

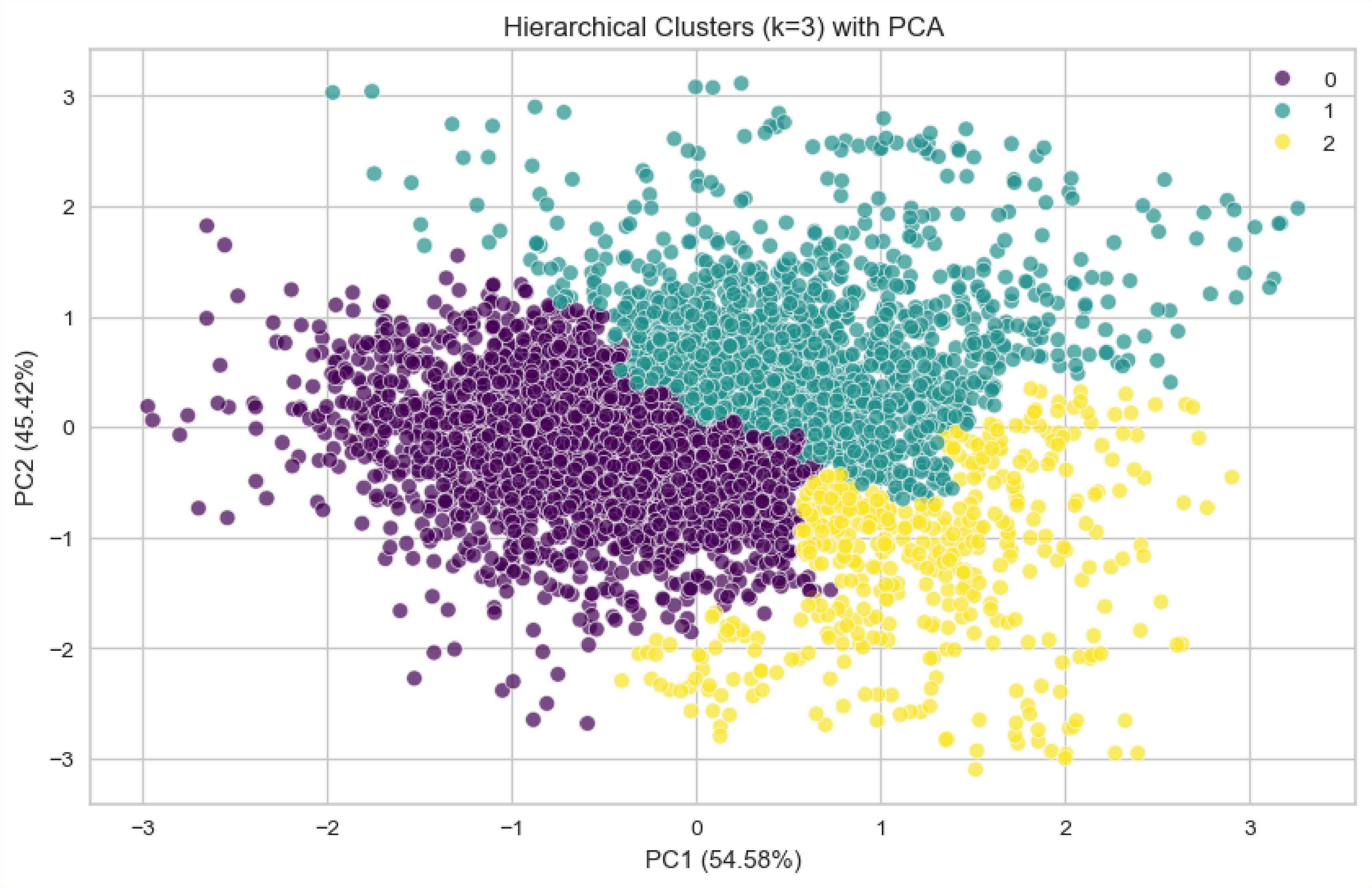
Hierarchal Clustering



- Use dendogram to better understand optimal split of the major customer clusters - there are three major customer clusters (green, red, and pink branches) when cutting at the 12.5 Euclidean distance threshold
- Each cluster shows distinct internal structures with multiple sub-clusters, suggesting potential for more granular segmentation if business needs require it

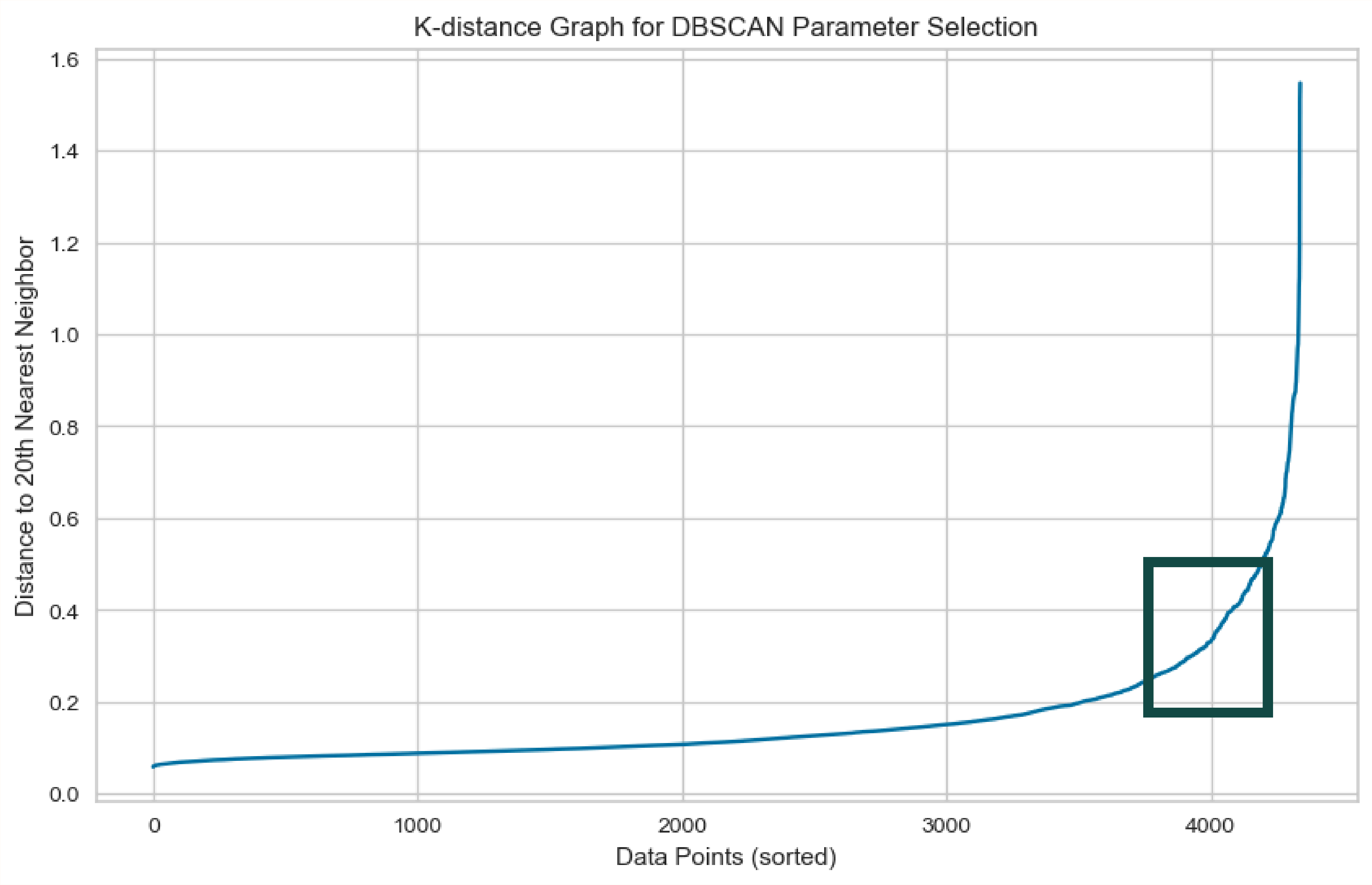
Hierarchal Clustering





- Silhouette score analysis confirms that 3 clusters provide the optimal segmentation solution with the highest score (~0.33), aligning with our K-means findings and validating the robustness of our customer segmentation approach
- Blue cluster shows the widest distribution across the plot's upper region, suggesting customers with more varied behaviors, while the purple and yellow clusters form more concentrated groupings with some overlap, indicating opportunities for specialized marketing strategies

DBSCAN Clustering

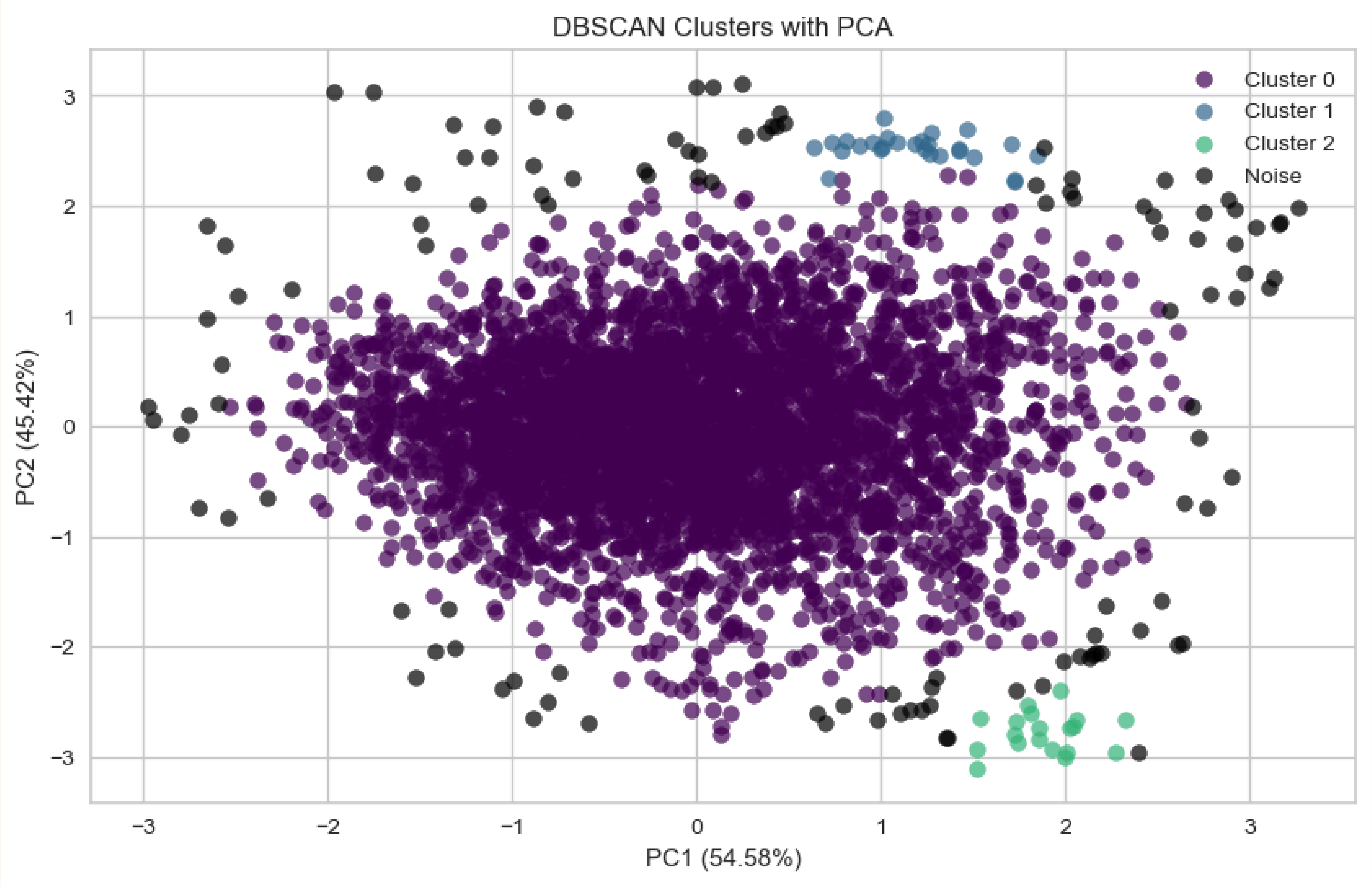



```
Best DBSCAN parameters: eps=0.35, min_samples=15
Number of clusters: 3
Silhouette score: 0.4056
```

- K-distance graph shows a gradual increase until around the 4000th data point, followed by a sharp elbow inflection point - showing an optimal epsilon (eps) value of 0.35 for neighborhood density
- Parameter tuning shows that min_samples=15 is the best cluster definition, ensuring clusters contain related customers rather than random groupings
- From the parameter tuning strategy, the best parameters achieved the highest silhouette score of 0.4056
- These optimized parameters effectively balance cluster density requirements while minimizing noise, creating a more robust clustering strategy than traditional methods

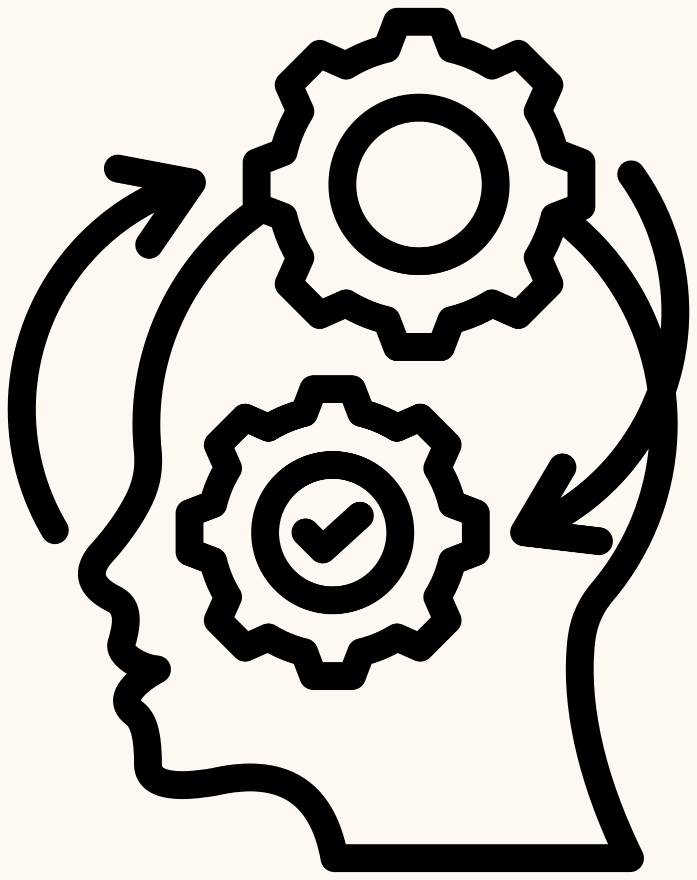
DBSCAN Clustering

- DBSCAN identified 3 distinct customer segments
- Outliers were automatically flagged as noise points (black dots), representing around 5-10% of customers with unique purchase patterns
- The purple cluster is clearly the most dominant, forms a dense central group containing most customers
- Meanwhile, blue and green clusters represent more specialized customer segments with unique purchasing behaviors
- Since DBSCAN doesn't force every customer into a cluster, it gives a more realistic representation of the customer base by distinguishing between true segments and anomalous behaviors - further validated by the higher silhouette score

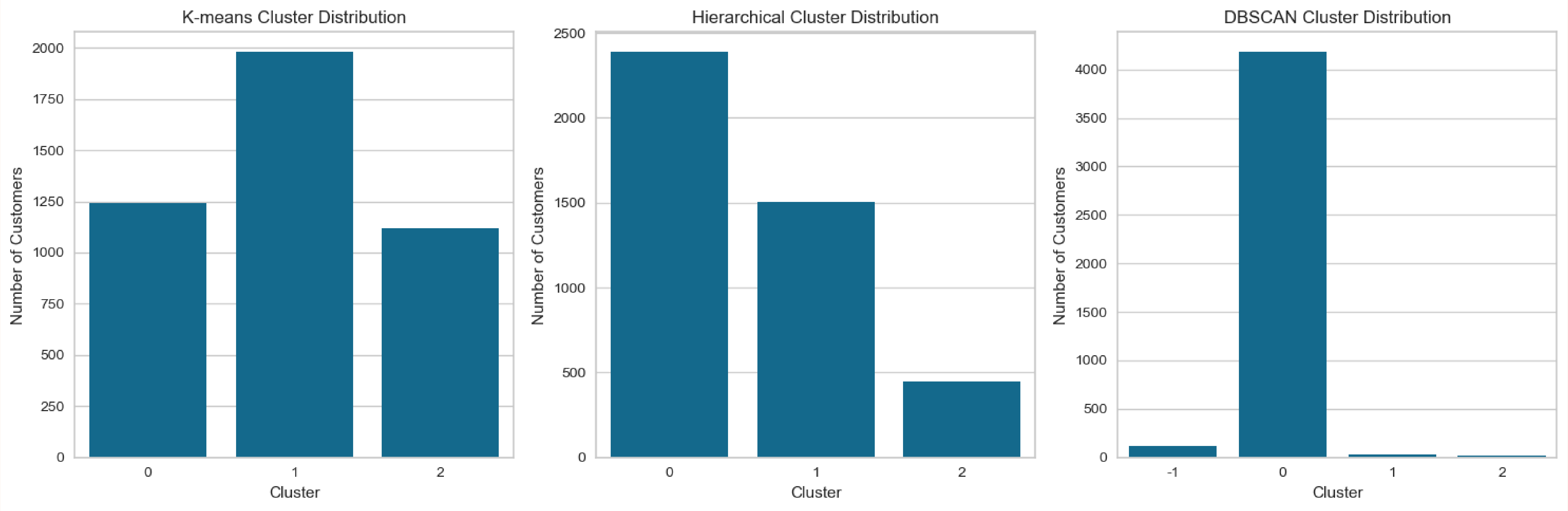



```
Percentage of noise points: 2.67%
DBSCAN Silhouette Score: 0.4056
DBSCAN Calinski-Harabasz Score: 155.19
```

Model Interpretation



Model Comparison



	Algorithm	Optimal Clusters	Silhouette Score	Calinski-Harabasz Score
0	K-means	3	0.357699	2705.160839
1	Hierarchical	3	0.332999	2143.074360
2	DBSCAN	3	0.405579	155.188028

- K-means produced the most balanced customer segments, while DBSCAN identified one dominant segment containing over 90% of customers (4000+), suggesting different approaches to defining customer similarities
- DBSCAN achieved the highest silhouette score (0.4056) despite its imbalanced distribution, indicating that its density-based approach identified the most statistically cohesive segments, though potentially less practical for marketing implementation than the other two

K-Means Cluster Characteristics

	cluster_kmeans	Recency	Purchase_Frequency	Avg_Spend_per_Order	UnitPrice_mean	StockCode_nunique	Count
0	0	147.530596	2.393720	278.397221	9.092798	21.159420	1242
1	1	44.911021	6.569767	588.202776	2.980703	106.530334	1978
2	2	111.822898	2.293381	270.583863	1.972570	26.650268	1118

Cluster Characteristic Interpretation

- Cluster 0 (Value Shoppers): Moderate recency (148 days) with low purchase frequency (2.4 orders) but high average spend ($278) and highest unit price ($19), indicating customers who make occasional but significant purchases
- Cluster 1 (Frequent Browsers): Best recency (145 days) and highest purchase frequency (6.6 orders), with high product variety (107 unique items) but moderate spend per order ($88), representing engaged customers who regularly purchase a diverse range of items
- Cluster 2 (Budget Buyers): Most recent activity (112 days) with moderate frequency (2.3 orders) but lowest average spend ($71) and product price ($2), suggesting price-sensitive customers who purchase lower-cost items relatively often
- All three segments show distinct purchasing behaviors that require tailored marketing approaches, with the most valuable customers appearing in Clusters 0 and 1 but for different reasons, high-value transactions compared to frequent engagement

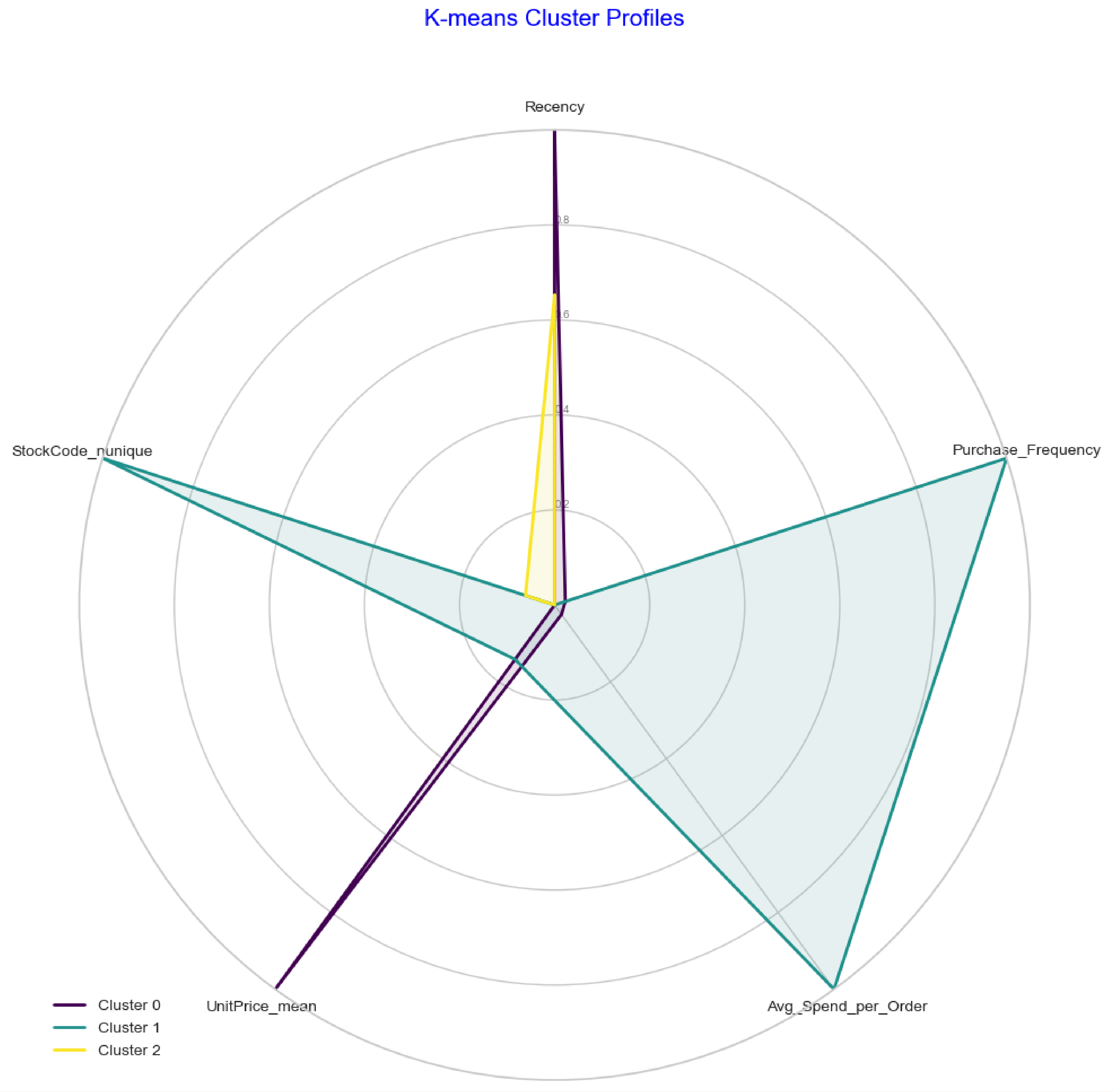
Hierarchal Cluster Characteristics

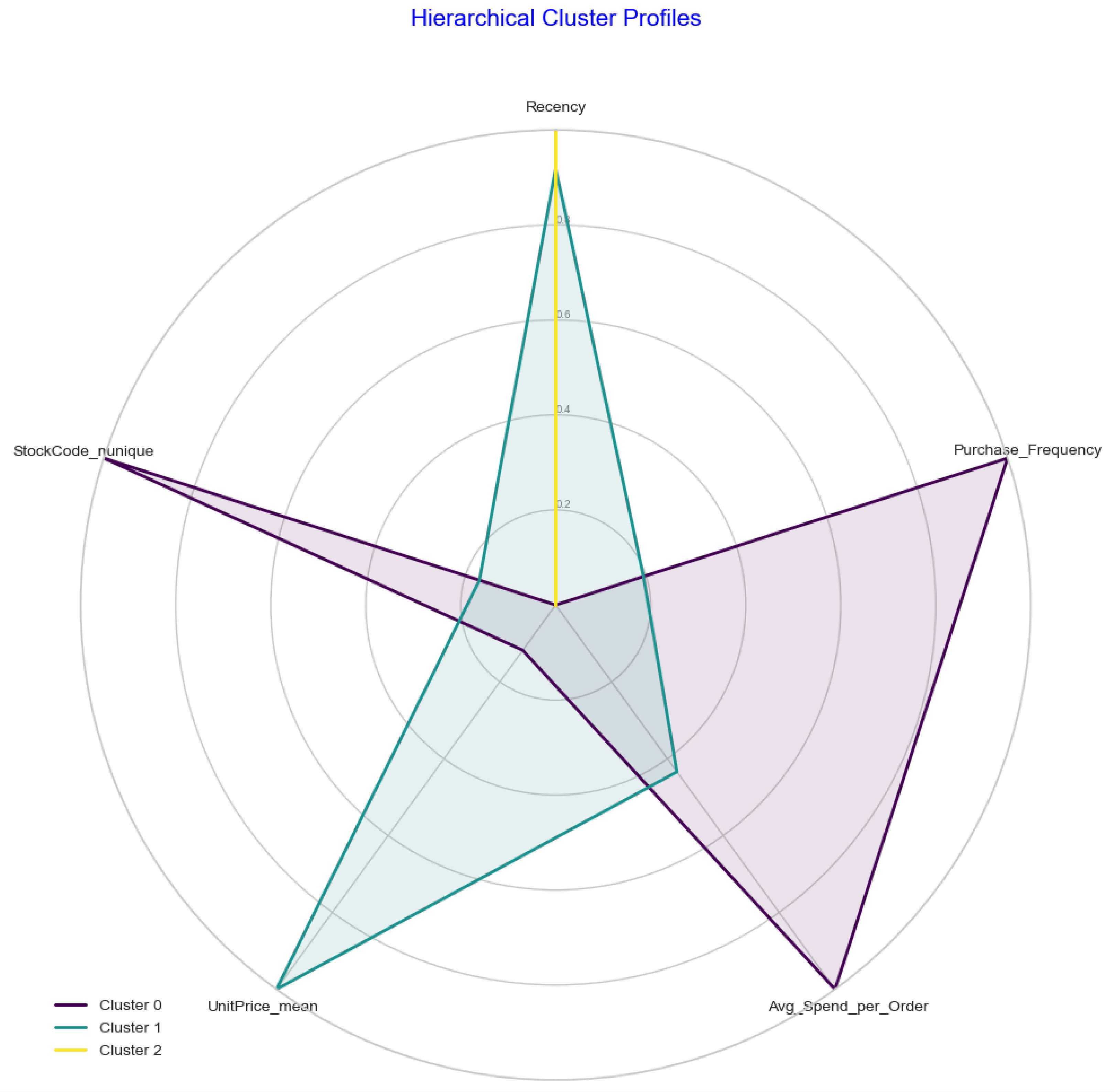
	cluster_hierarchical	Recency	Purchase_Frequency	Avg_Spend_per_Order	UnitPrice_mean	StockCode_nunique	Count
0	0	51.826415	5.855765	539.326544	2.608113	93.568553	2385
1	1	138.289509	2.521912	309.436910	8.195184	25.503320	1506
2	2	145.894855	1.718121	132.978536	1.861571	11.684564	447

Cluster Characteristic Interpretation

- Cluster 0 (Premium Shoppers): Moderate recency (52 days) with highest purchase frequency (5.9 orders) and significantly highest average spend ($539) and unit price ($43), representing your high-value, loyal customer segment.
- Cluster 1 (Mid-Market Regulars): Longer since last purchase (138 days) with moderate frequency (2.5 orders) and average spend ($309), indicating a solid mid-tier segment with growth potential.
- Cluster 2 (Infrequent Bargain Hunters): Longest inactive period (146 days) with lowest purchase frequency (1.7 orders), average spend ($133), and product variety (12 unique items), suggesting a segment that makes occasional low-cost purchases.
- Hierarchical clustering reveals a more pronounced segmentation by value, with clearer distinction between high-value customers (Cluster 0) and lower-value customers (Cluster 2) than the K-means approach

Cluster Profile Comparison





Model Decision

Cluster Size Comparison – K-means vs Hierarchical:

	K-means	Hierarchical
0	1242	2385
1	1978	1506
2	1118	447

Clustering Similarity (Normalized Mutual Information): 0.4871
(0 = completely different clusters, 1 = identical clusters)

Cluster Overlap (Cross-tabulation):

Hierarchical	0	1	2
K-means			
0	8	1190	44
1	1789	189	0
2	588	127	403

Cluster stability (higher = more similar assignments): 82.23%

Score Comparison:

	Algorithm	Optimal Clusters	Silhouette Score	Calinski-Harabasz Score
0	K-means	3	0.357699	2705.160839
1	Hierarchical	3	0.332999	2143.074360
2	DBSCAN	3	0.405579	155.188028

Decision Summary:
Better Silhouette Score: K-means
Better Calinski-Harabasz Score: K-means

Recommended algorithm: K-means

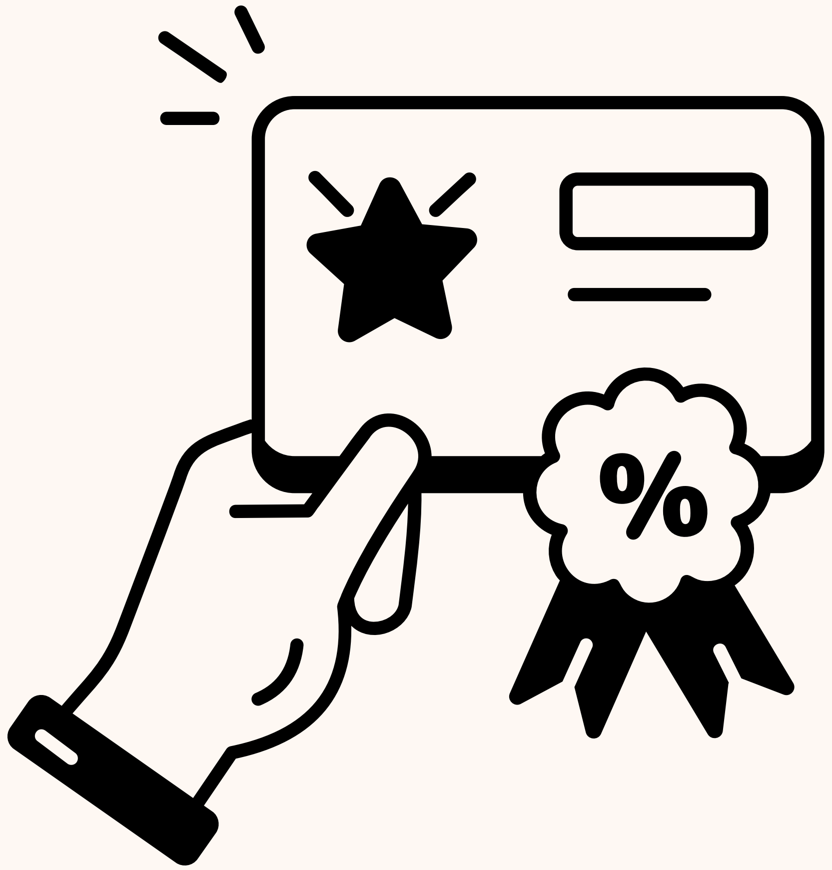
- Make use of K-means due to better silhouette score and Calinski-Harabasz score

Model Implementation

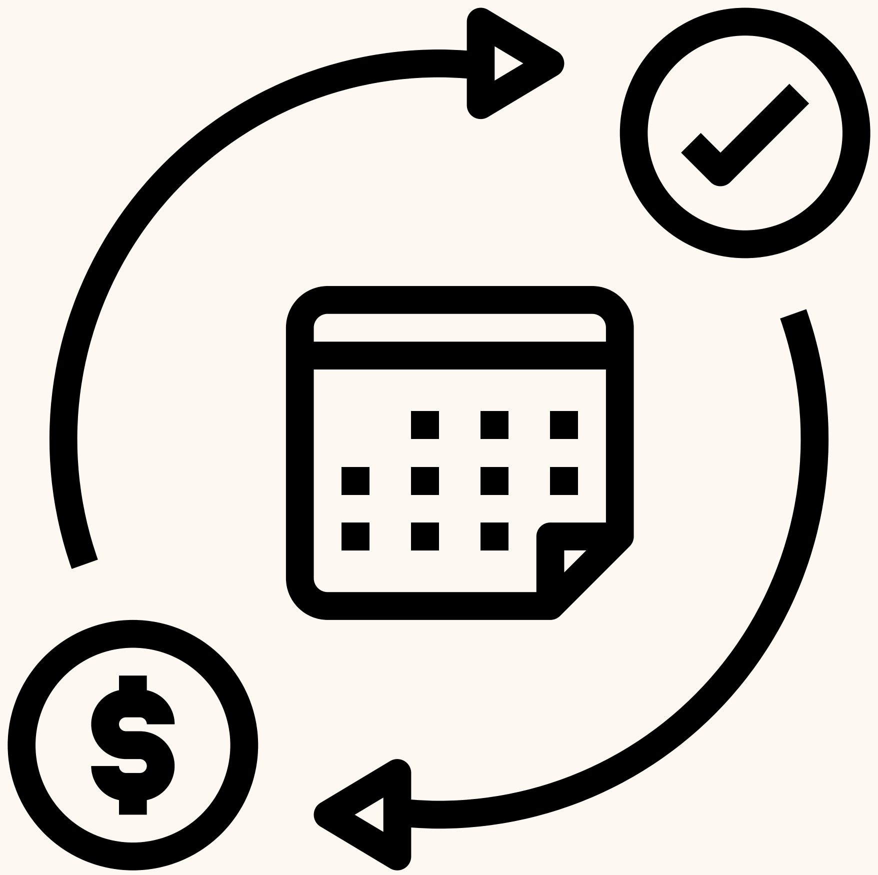
Marketing Strategies

- For Cluster 0 (Value Shoppers): Launch a premium loyalty program offering exclusive access to new high-value products, personalized product recommendations based on previous high-value purchases, and VIP customer service to encourage continued premium spending
- For Cluster 1 (Frequent Browsers): Implement a subscription model or bundle discounts for frequently purchased items, cross-sell related products based on their diverse purchase history, and create a tiered rewards program that increases benefits with purchase frequency
- For Cluster 2 (Budget Buyers): Deploy targeted price promotions and flash sales notifications, introduce a budget-friendly product line, and develop a "buy more, save more" program to encourage incremental spending without sacrificing price sensitivity
- Cross-Segment Strategy: Create a data-driven customer journey model that aims to migrate Budget Buyers to Frequent Browsers, and Frequent Browsers to Value Shoppers through targeted incentives and personalized communication strategies

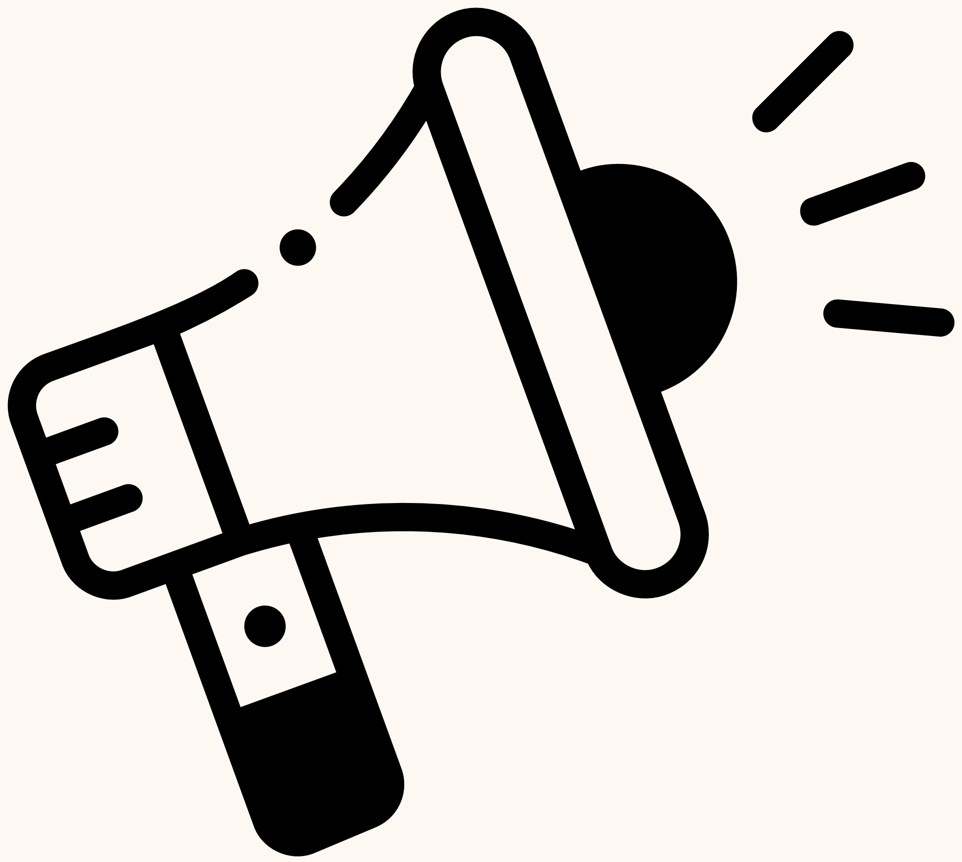
Marketing Strategies



Value Shoppers



Frequent Browsers



Budget Buyers