5580 Data and Text Mining

Master of Science in Computing and Data Analytics

Department of Mathematics and Computing Science

Unsupervised Learning - Clustering

Group 4:
Bhavik Kantilal Bhagat (A00494758)
Jeevan Dhakal (A00494615)
Binziya Siddik (A00494129)

Date: January 19, 2026

Contents

1 Introduction

1.1 Objective

The primary objective of this analysis is to perform **unsupervised machine learning** on the retail transaction dataset (`sales219`) to identify distinct customer segments, and products for targeted marketing for those customer segments. Unlike supervised learning, which predicts known outcomes based on labeled data, this study utilizes **clustering algorithms** to uncover hidden patterns and natural groupings within the customer base and products without prior ground truth. [3]

By segregating customers based on their purchasing behavior, we aim to provide the business owner with actionable profiles—such as "VIP Whales" or "Weekend Warriors"—to enable targeted marketing strategies rather than a generic "one-size-fits-all" approach.

1.2 Scope of Analysis

The study requires clustering for both Customers and Products. To ensure analytical depth, the work was divided as follows:

- **Customer Clustering:** Analysis of purchasing behavior, based on preliminary factors like number of products bought, number of distinct products bought, revenue contribution, and number of visits. However, to conduct a thorough analysis, some other factors are considered, such as visit frequency (monthly), promotion sensitivity, and weekend shopping habits.
- **Product Clustering:** Analysis of inventory performance, revenue per product, and basket co-occurrence using Mean Normalization.

This report **focuses** on the **Customer Segmentation** process, detailing the end-to-end process from feature engineering to model validation, along with results. Additionally, the **Product Clustering** methodology and results will be **briefly** discussed.

2 Methodology and Data Preparation

The clustering analysis was conducted using a hybrid technology stack involving MySQL for data storage and a Python-based analytics stack (NumPy, pandas, Scikit-learn, Matplotlib, Seaborn, etc.) for analytical processing. This rigorous data pipeline ensured that raw transactional logs were transformed into a normalized, high-quality feature set suitable for the k-means algorithm. [5]

2.1 Data Acquisition and Storage

The raw dataset was provided as a structured SQL file. A local MySQL database named `mcda5580` was created, and the source script was executed to generate the `sales219` table and load the relevant data.

- **Volume:** The dataset comprises **3,678,038 rows** across 17 columns.
- **Granularity:** Each record represents an individual line-item transaction.

Handling a dataset of this scale (> 3.6 million rows) required database-level storage rather than spreadsheet software to preserve data integrity and query performance.

2.2 Data Integration (Python–SQL Bridge)

To enable advanced feature engineering, a connection was established between the local MySQL database and the Python environment using the `mysqlclient` and `sqlalchemy` libraries. The data were ingested into a pandas DataFrame for in-memory manipulation. This approach enabled SQL-based filtering alongside Python-based vectorized operations.

2.3 Data Aggregation

The transactional data were aggregated by `customer_id` to generate a customer-centric representation. This process transformed over 3.6 million transactions into unique profiles for **44,469 distinct customers** across **30,507 products**.

The initial customer-level aggregation included:

- **Total Revenue:** Sum of `selling_retail_amount`.
- **Product Diversity:** Count of distinct products purchased.
- **Visit Frequency:** Count of unique transaction identifiers.

2.4 Exploratory Data Analysis (EDA) and Feature Engineering

To satisfy the requirement for creativity with justification, behavioral features were explored beyond simple aggregation metrics. Because k-means clustering identifies latent behavioral patterns, features were engineered to capture *how* customers shop rather than solely *how much* they spend.

1. **Weekend Proportion (Lifestyle Metric):** The day of the week was extracted from transaction timestamps, and the following ratio was computed:

$$\text{Weekend Proportion} = \frac{\text{Visits (Saturday + Sunday)}}{\text{Total Visits}}$$

   This feature, bounded between 0 and 1, differentiates weekend-focused customers from those with flexible weekday schedules.

2. **Promotion Sensitivity (Price Metric):** Using the `PROMO_SALES_IND_CD` attribute, transactions were categorized by promotion type. Features were engineered to capture consumption patterns for `Promo B` (bundles) and `Promo C` (clearance), enabling differentiation between value-driven and premium shoppers.

3. **Visit Consistency:** The `visits_per_month` metric was calculated to normalize purchasing behavior across newly acquired and long-term customers.

2.5 Data Cleaning and Outlier Detection

Unsupervised learning algorithms such as k-means rely on **Euclidean distance** to determine cluster membership and are therefore highly sensitive to outliers.

- **Identification:** EDA revealed extreme "whale" customers, including a single entity with more than 43,000 recorded visits. Inclusion of such extreme observations would compress the remaining data distribution and distort clustering results.
- **Action:** A **99th-percentile filter** was applied, removing the top 1% of records based on revenue and visit frequency.
- **Validation (DBSCAN):** As a validation step, the DBSCAN algorithm was applied to the unfiltered data. These extreme observations were consistently flagged as noise (label -1), providing empirical justification for their exclusion.

In parallel, EDA was conducted on the **product** dataset to assess inventory performance. After removing invalid transactions and handling missing values, nine key features were engineered, including revenue, customer reach, purchase frequency, and sales consistency metrics.

2.6 Normalization Strategy

Normalization is a critical prerequisite for k-means clustering. Because the algorithm relies on **Euclidean distance**, features with larger numeric ranges (e.g., revenue) can dominate similarity calculations if not appropriately scaled.

2.6.1 Customer Normalization

For customer segmentation, **min–max scaling** was applied to transform all features into the fixed range $[0,1]$. Following outlier removal, the normalized customer feature vector was computed as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

2.6.2 Product Normalization

For product-level clustering, a **log transformation** was applied to mitigate revenue skewness, followed by **robust scaling** to reduce sensitivity to extreme values. This ensured that high-selling items did not distort the feature space.

Using these normalization strategies, cluster counts ranging from $k = 2$ to $k = 10$ were evaluated using both the **elbow method** and **silhouette scoring**. The final model identified **six distinct product clusters** based on commercial performance characteristics.

3 Data Modeling and Validation

3.1 Modeling

To segment the customer base, we applied the **k-means algorithm** to the normalized dataset. Rather than arbitrarily selecting the number of segments, a two-stage validation process was employed, incorporating statistical, visual, and business validation to determine the optimal cluster count (k). [2]

Product segmentation was performed using a systematic **k-means clustering** approach on retail transaction data spanning **30,507 products**. Following rigorous data cleaning to remove invalid transactions and handle missing values, **nine key features** were engineered, including revenue, customer reach, purchase frequency, and sales consistency metrics.

Cluster counts ranging from $k = 2$ to $k = 10$ were evaluated using both the **elbow method** and **silhouette scoring**. The final model identified **six distinct product clusters** based on commercial performance patterns.

3.2 Stage 1: Statistical Validation (Elbow Method and Silhouette Score)

The algorithm was executed for values of k ranging from 2 to 10, and performance was assessed using two complementary metrics: compactness, measured by the sum of squared errors (SSE), and separation, measured by the silhouette score.

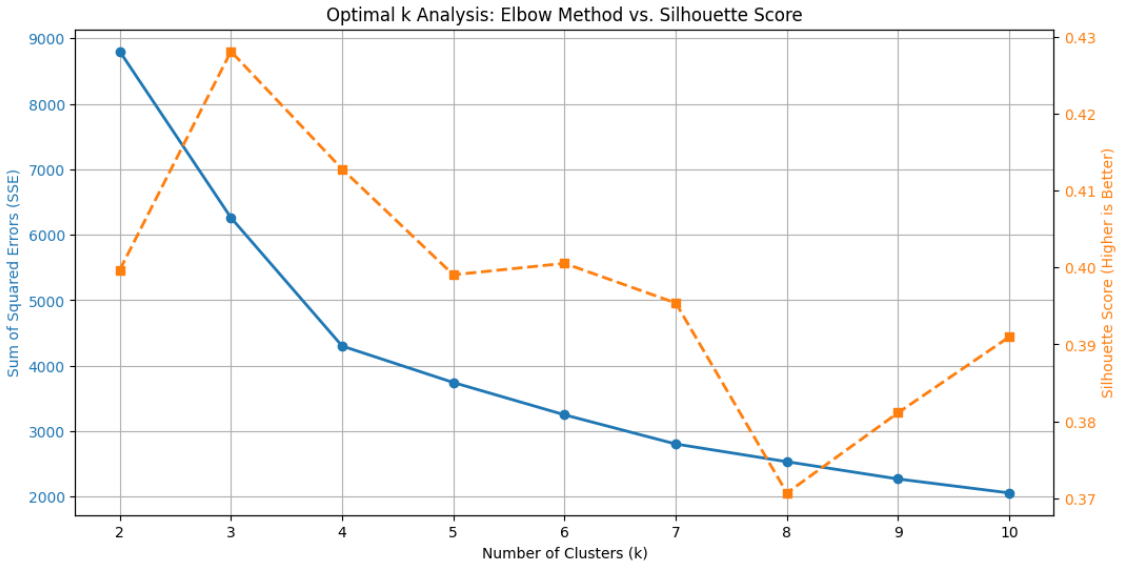

Figure 1: Dual-metric analysis comparing the elbow method and silhouette score [4]

Analysis of Figure 1:

- **Elbow Method (Blue Line):** The SSE decreases sharply from $k = 2$ to $k = 3$, followed by a smaller reduction at $k = 4$, after which the curve flattens. This indicates diminishing returns beyond four clusters.
- **Silhouette Analysis (Orange Line):** The highest silhouette score was observed at $k = 3$ (**0.428**), indicating strong separation. However, the score for $k = 4$ remained comparably high (**0.413**).

3.3 Stage 2: Visual and Business Validation

Although statistical validation favored $k = 3$, effective business segmentation requires actionable differentiation. To resolve the discrepancy between $k = 3$ and $k = 4$, clusters were evaluated using both Principal Component Analysis (PCA) visualizations and a business-oriented view (revenue versus visit frequency).

Figure 2: Comparative validation showing cluster overlap at $k = 3$ and VIP isolation at $k = 4$

3.3.1 Rejection of $k = 3$: The Lumping Problem

As shown in the **left panel ($k = 3$)** of Figure 2, the algorithm produced a dominant cluster containing over 26,000 customers.

- **Deficiency:** This solution merged frequent visitors with one-time shoppers.
- **Business Impact:** The resulting segment lacked sufficient granularity to support targeted marketing strategies.

3.3.2 Selection of $k = 4$: VIP Segment Isolation

The **right panel ($k = 4$)** demonstrates a critical improvement, in which the algorithm separated the dominant group and isolated a distinct high-value segment (Cluster 3, shown in teal/blue).

- **Quantitative Validation:** In the $k = 3$ model, the top segment exhibited an average revenue of $683. Under $k = 4$, the newly isolated VIP cluster achieved an average revenue of **$790**.
- **Conclusion:** By accepting a negligible reduction in silhouette score (0.015), the model achieved a highly pure VIP segment. This confirms $k = 4$ as the superior *business* solution, providing actionable insights that were obscured under $k = 3$.

Screenshots documenting the modeling and validation of the product clustering process are provided in the Appendix (Section 6).

4 Results and Comparative Analysis

To determine the final segmentation model, the cluster centroids and distributions of both the three-cluster and four-cluster solutions were analyzed. While statistical metrics derived from the elbow method indicated that $k = 3$ was acceptable, a more granular analysis of customer behavior demonstrated that $k = 4$ provided substantially higher business utility.

4.1 Analysis of the $k = 3$ Model (Baseline)

The three-cluster solution, although statistically stable, exhibited excessive generalization. As shown in Table 1, Cluster 0 contained nearly 60% of the total customer base (26,850 customers), combining multiple distinct behavioral profiles into a single mass-market segment.

Table 1: Summary of $k = 3$ Clusters

Cluster	Size	Revenue ($)	Visits/Month	Weekend Ratio	Profile Description
0	26,850	75.37	1.71	0.05	**Mass Market**
1	10,063	72.52	1.57	0.84	**Weekenders**
2	6,667	683.79	6.32	0.26	**High Value**

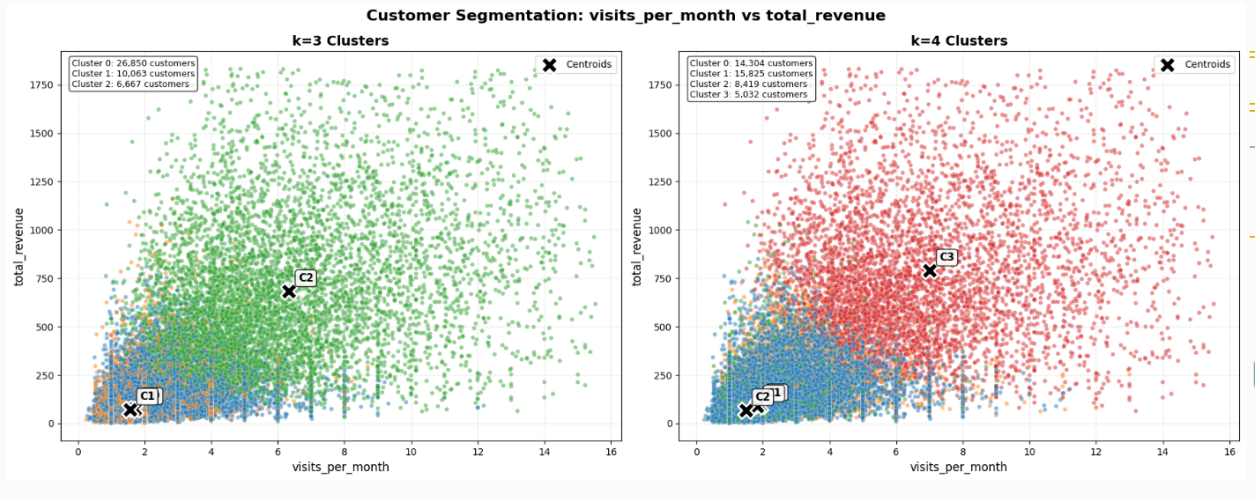

Figure 3: Cluster separation comparing visits and total revenue. Centroid 3 ($k = 4$) exhibits a substantially higher revenue position than Centroid 2 ($k = 3$).

Critique of the $k = 3$ Model: Although Cluster 2 exhibited an average revenue of $683, Figure 3 (left panel) illustrates that this cluster encompassed a wide behavioral range, combining moderate spenders with true high-value customers. This overlap reduced the effectiveness of targeted premium marketing strategies.

4.2 Analysis of the $k = 4$ Model (Selected Solution)

Increasing the segmentation resolution to $k = 4$ enabled the algorithm to partition customers into more behaviorally coherent groups. Table 2 presents the refined cluster metrics.

Table 2: Summary of $k = 4$ Clusters

Cluster	Size	Revenue ($)	Visits/Month	Promo B Usage	Profile Description
0	14,304	94.43	1.91	**0.78**	**Promo Hunters**
1	15,825	88.54	1.82	0.48	**Casual Browsers**
2	8,419	69.03	1.48	0.44	**Weekend Warriors**
3	5,032	**790.83**	**7.01**	5.42	**VIP Whales**

4.2.1 Key Improvements in the $k = 4$ Model

1. **Identification of VIP Whales (Cluster 3):** Under the $k = 3$ model, the highest-value segment averaged $683 in revenue. The $k = 4$ solution isolated a smaller, purer segment of 5,032 customers with an average revenue of **$790.83**. This distinction enables targeted premium retention strategies that were previously diluted.
2. **Refinement of Weekend Warriors (Cluster 2):** The weekend activity ratio increased from 0.84 in the $k = 3$ model to **0.90** under $k = 4$, confirming a highly distinct segment that almost exclusively shops on weekends. This insight supports weekend-specific staffing and promotional strategies.
3. **Segmentation of the Mass Market (Clusters 0 and 1):** The large mass-market cluster from the $k = 3$ solution was divided based on promotion sensitivity. Cluster 0 demonstrated substantially higher Promo B usage (0.78 average) compared to Cluster 1 (0.48 average), indicating that Cluster 0 is price-sensitive, whereas Cluster 1 exhibits general disengagement.

4.3 Visual Validation of Separation

Figure 3 visually reinforces the superiority of the $k = 4$ solution. Additional visualizations illustrating customer and product clustering outcomes are provided in the Appendix (Section 6).

In the $k = 4$ cluster plot (right panel), the red cluster (Cluster 3) clearly isolates the high-frequency, high-revenue quadrant, while the blue and orange clusters (Clusters 0 and 1) effectively partition the dense lower-left region. This confirms that $k = 4$ captures the underlying structural variance of the customer base more effectively than $k = 3$.

5 Summary

This report presents a comprehensive data mining analysis of the `sales219` retail transaction dataset. By applying unsupervised machine learning techniques, including **k-means clustering**, and rigorous validation methods such as **DBSCAN** and **silhouette analysis**, the business entities were partitioned into actionable and interpretable segments. In the future, **hierarchical clustering** can be employed to find nearest cluster which can be used to recommendation systems. [1]

The analysis was structured around two strategic domains:

1. **Customer Segmentation:** Identification of purchasing behaviors to optimize marketing expenditure.
2. **Product Segmentation:** Categorization of inventory performance to support supply chain and merchandising decisions.

5.1 Key Findings: Customer Segmentation ($k = 4$)

Using a normalized dataset with outlier protection (top 1% removed), four distinct customer personas were identified:

- **VIP Whales (11%):** The most valuable segment, generating an average of **$790 per month** with a high visit frequency (approximately seven visits per month). These customers exhibit strong engagement with `Promo B` bundle offers.
- **Weekend Warriors (19%):** A segment revealed through targeted feature engineering. These customers conduct approximately **90% of their visits** on Saturdays and Sundays, representing a concentrated demand period that introduces specific staffing and inventory pressures. The methodology supporting this finding is detailed in the Appendix (Section 6).
- **Promo Hunters and Casuals:** The mass-market segment was further divided based on promotion sensitivity. Promo Hunters actively engage with clearance-based promotions (`Promo C`), whereas Casual customers exhibit low engagement levels and an elevated risk of churn.

5.2 Key Findings: Product Segmentation ($k = 6$)

Product-level clustering revealed distinct performance-based groupings that differentiate high-revenue flagship products from low-velocity inventory. These clusters provide actionable insight into which products warrant premium positioning, routine replenishment, or targeted clearance strategies.

- **High-Value Drivers (Clusters 0 & 4):** The "Foundation Pillars" and "Premium Niche" products (28% of items) generate the vast majority of revenue through high traffic and exceptional customer spend, requiring strict inventory protection.
- **Core Stabilizers (Clusters 1 & 2):** "Mass Market Essentials" and Average products provide essential inventory depth and stability; while individual revenue is lower, they are crucial for bundling and customer retention.
- **Candidates for Rationalization (Clusters 3 & 5):** The "Inactive" and "Neglected" segments represent dead weight with near-zero engagement, identifying immediate opportunities for clearance sales or discontinuation to free up capital.

5.3 Strategic Recommendation

Integrating insights from both segmentation models yields a unified strategic framework. The business should prioritize **VIP Whales** with exclusive promotions centered on **flagship products**, while leveraging the predictable shopping behavior of **Weekend Warriors** to reduce **slow-moving inventory** through time-restricted weekend flash sales. This dual strategy aligns customer value with inventory optimization, maximizing revenue while minimizing operational inefficiencies.

6 Appendix

6.1 Source Code - GitHub

GitHub repository

6.2 YouTube Video

YouTube Video

6.3 Chat Session Links

Notebook LM

DeepSeek 1

DeepSeek 2

6.4 Additional Plots

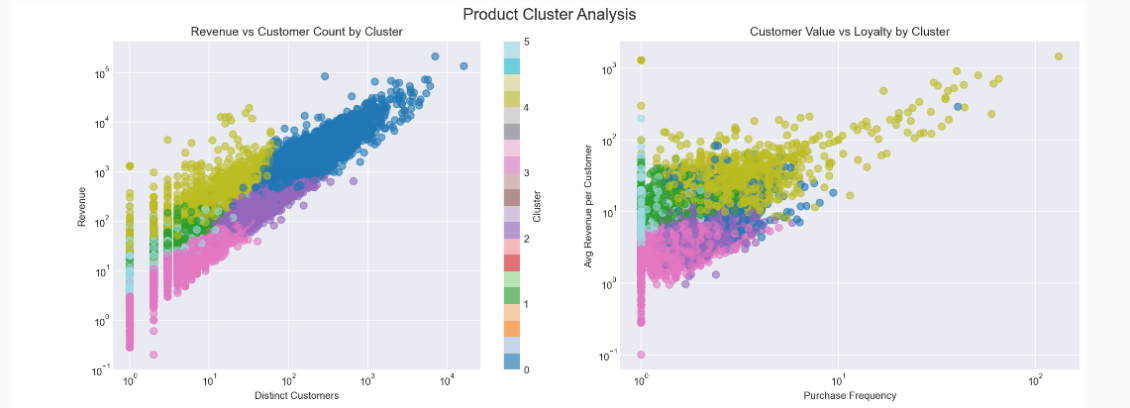

Figure 4: Dual-metric analysis comparing the elbow method and silhouette score

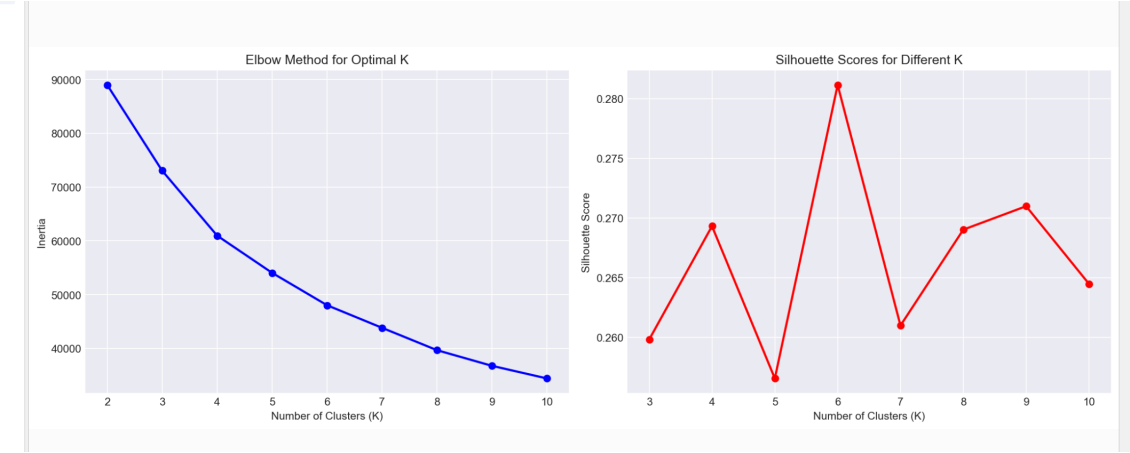

Figure 5: Product Clusters Analysis

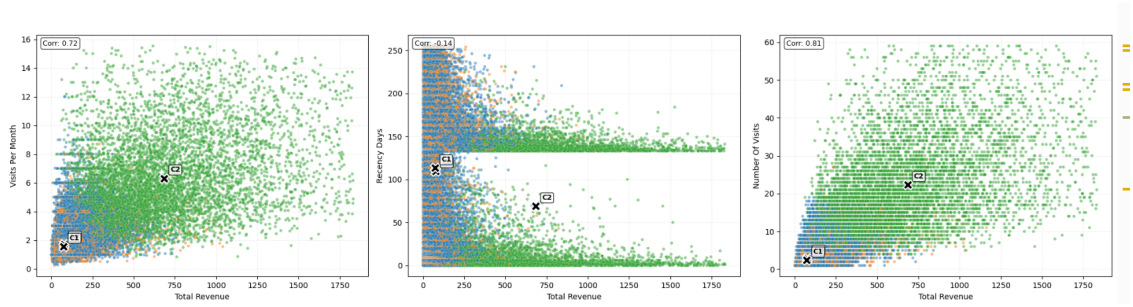

Figure 6: Various plots for features for 3 clusters

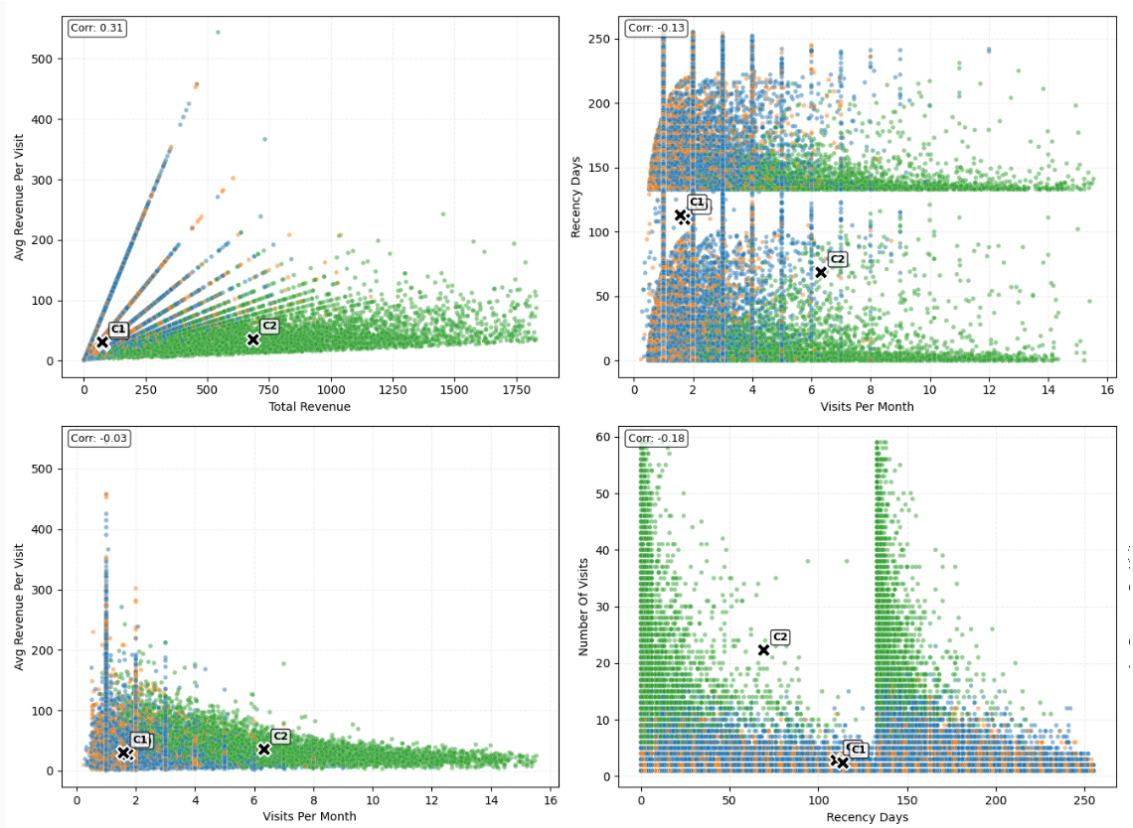

Figure 7: Various plots for features for 3 clusters

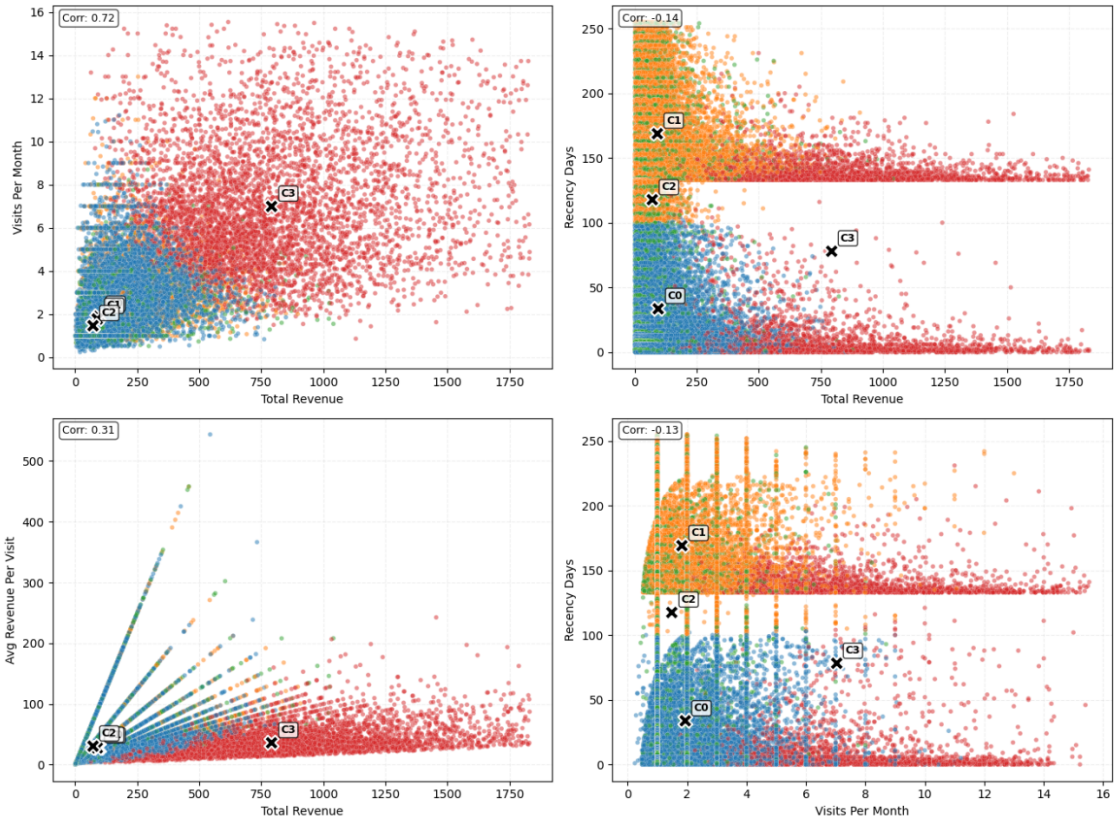

Figure 8: Various plots for features for 4 clusters

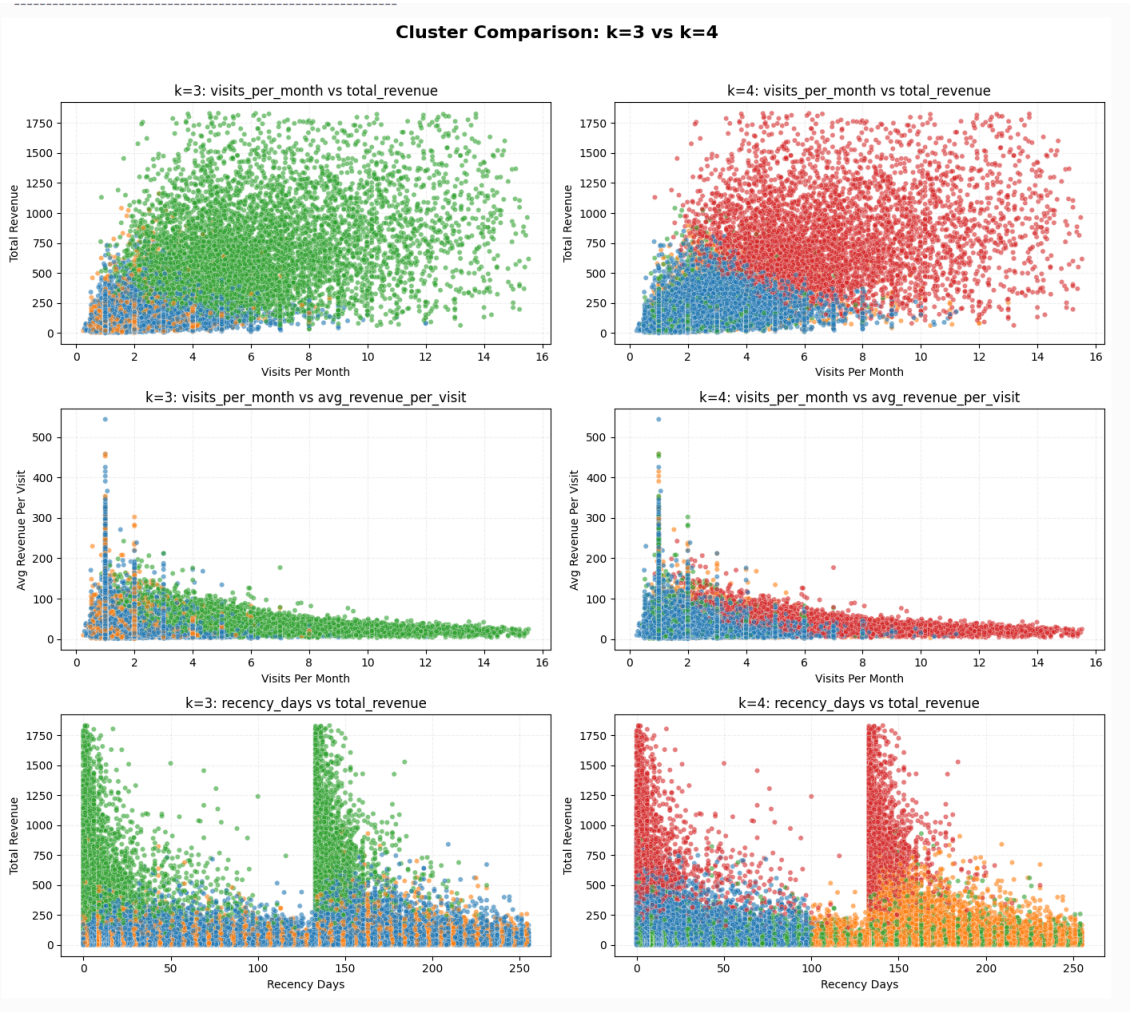

Figure 9: Comparison between 3 and 4 clusters

Bibliography

References

[1] Data Analytics Pro. **Introduction to Hierarchical Clustering**. YouTube video. Accessed: April 2, 2025. Oct. 2022. URL: https://www.youtube.com/watch?v=RDZUdRSDOok (visited on 04/02/2025) (cit. on p. 11).

[2] Data Science Basics. **K-Means Clustering Explained**. YouTube video. Accessed: April 2, 2025. Mar. 2023. URL: https://www.youtube.com/watch?v=4b5d3muPQmA (visited on 04/02/2025) (cit. on p. 6).

[3] Data Science Dojo. **4 Basic Types of Cluster Analysis used in Data Analytics**. YouTube video by Data Science Dojo. Accessed: January 19, 2026. 2020. URL: https://www.youtube.com/watch?v=Se28XHI2_xE (visited on 01/19/2026) (cit. on p. 2).

[4] GeeksforGeeks. **What is Silhouette Score?** GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/ (visited on 04/02/2025) (cit. on p. 6).

[5] Prashant, S. **K-Means Clustering with Python**. Kaggle Notebook. Kaggle. Dec. 2020. URL: https://www.kaggle.com/code/prashant111/k-means-clustering-with-python (visited on 04/02/2025) (cit. on p. 3).