Task 02: Customer Segmentation

Team Insighta

1. Task Overview

The dataset we were given consists of customer interaction data from an e-commerce platform. It contains six key features:

1. customer_id : Unique id for the customer.
2. total_purchases : Total number of purchases made by the customer.
3. avg_cart_value : Average value of items in the customer's cart.
4. total_time_spent : Total time spent on the platform (in minutes).
5. product_click : Number of products viewed by the customer.
6. discount_count : Number of times the customer used a discount code.

Assigned task was to cluster customers into three predefined segments:

1. Bargain Hunters
   - total_purchases: High (frequent purchases).
   - avg_cart_value : Low (they buy cheaper items).
   - total_time_spent : Moderate (they spend some time browsing but focus on purchasing).
   - product_click : Moderate (they view a reasonable number of products).
   - discount_count : High (they frequently use discount codes).

2. High Spenders
   - total_purchases: Moderate (they make fewer but high-value purchases).
   - avg_cart_value: High (they buy expensive items).
   - time_spent: Moderate (they spend time browsing but focus on high-value items).
   - product_click : Moderate (they view a reasonable number of products).
     discount_usage: Low (they rarely use discount codes).

3. Window Shoppers
   - total_purchases: Low (they make very few purchases).
   - avg_cart_value: Moderate (they view items of varying prices).
   - time_spent: High (they spend a lot of time browsing).
   - product_click : High (they view a large number of products).
   - discount_usage: Low (they rarely use discount codes).

2. Data Preprocessing

- **Handling Missing Data**

The dataset contained 20 missing values in features total_purchases, avg_cart_value, and product_click. To handle these missing values without losing valuable data, **median imputation** was applied. The median was chosen over the mean because it is less affected by extreme values and skewed distributions, ensuring a more stable representation of customer behavior.

- **Removing Outliers**

Outliers were identified and removed using the **Z-score method** to enhance clustering accuracy. Any data point with a |Z-score| greater than 3 was considered an outlier. This helped eliminate extreme values that could distort the clustering process, allowing the model to focus on meaningful patterns instead of anomalies.

- **Feature Selection**

To improve clustering performance and reduce redundancy, a **correlation matrix** was analyzed to identify highly correlated features. The matrix showed that:

- total_time_spent and product_click were highly correlated
- discount_count and total_purchases were also strongly correlated

To avoid redundancy and multicollinearity, one feature from each pair was removed

- Removed total_time_spent (kept product_click).
- Removed discount_count (kept total_purchases).

By removing these redundant features, the model became more efficient, reducing noise and improving clustering accuracy without losing essential behavioral patterns.

- **Feature Scalling**

After handling missing data, outliers, and performing feature selection, **Min-Max Scaling** was applied. This scaling technique transforms the features to a fixed range between 0 and 1, ensuring that all features contribute equally to the clustering process.

3. Clustering Techniques

To segment customers effectively, multiple clustering techniques were explored, each with unique methodologies and assumptions. The following methods were applied:

- **K-Means Clustering**

K-Means is a centroid-based clustering algorithm that partitions data into **K clusters**, where each data point is assigned to the nearest cluster center. The algorithm iteratively updates the cluster centroids until convergence. It works best when clusters are well-separated and spherical in shape.

- **Gaussian Mixture Model (GMM)**

GMM is a **probabilistic clustering** method that models data as a mixture of multiple Gaussian distributions. Unlike K-Means, which assigns each point to a single cluster, GMM provides a probability distribution over clusters for each data point, allowing for more flexible cluster assignments, especially when the clusters have elliptical shapes.

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

DBSCAN is a **density-based** algorithm that groups together points that are closely packed while treating low-density points as noise. It does not require predefined cluster numbers but relies on distance-based density estimation. This method is useful for detecting arbitrarily shaped clusters and handling outliers effectively.

- **Hierarchical Clustering (Agglomerative)**

Hierarchical clustering builds a **nested hierarchy** of clusters using a bottom-up (agglomerative) approach, where individual points are initially treated as separate clusters and then progressively merged based on similarity. This method does not require specifying the number of clusters beforehand and provides a hierarchical structure useful for understanding relationships between clusters.

4. Performance Metrics

To assess the effectiveness of different clustering techniques, two key metrics were used:

1. Silhouette Score

- The Silhouette Score measures how well each data point fits within its assigned cluster compared to other clusters.
- A **higher Silhouette Score** (closer to 1) indicates **well-separated**, while a lower score (closer to -1) suggests poor clustering.

2. Davies-Bouldin Index

- The Davies-Bouldin Index (DBI) evaluates clustering quality by analyzing the similarity between clusters.
- A **lower DBI** indicates that clusters are **well-separated**, whereas a higher DBI suggests overlapping clusters.

5.Performance of Clustering Techniques

Each clustering method was applied to the dataset, and the performance was analyzed using these metrics.

	Gaussian Mixture	K-Means	Hierarchical Clustering	DBSCAN
Silhouette Score	0.6637	0.6635	0.6608	0.4121
Davies-Bouldin Index	0.6058	0.6058	0.6058	0.9127

DBSCAN performed the worst due to its inability to form three distinct clusters.

Hierarchical Clustering behaved similarly to K-Means, indicating that the data structure did not benefit from a hierarchical approach.

I selected the Gaussian Mixture Model (GMM) for segmentation, as it showed the best results, with the highest Silhouette Score and relatively low Davies-Bouldin Index, making it the preferred choice for segmentation.

6. Improving the Model

I performed **hyperparameter tuning** for the Gaussian Mixture Model (GMM). After evaluating various configurations, the final optimal model was selected as:

GMM = GaussianMixture(n_components=3, random_state=42, covariance_type='full', init_params='kmeans', max_iter=500)

Instead of removing highly correlated features such as total_time_spent and discount_count, I employed an **autoencoder** for dimensionality reduction while preserving key information. This approach allowed me to capture complex relationships within the data and compress it into a lower-dimensional representation before performing clustering.

The autoencoder was trained using Mean Squared Error (MSE) as the loss function, and Adam optimizer was applied to minimize the error during the training process.

The encoded features from the autoencoder were then fed into the Gaussian Mixture Model (GMM) for clustering, allowing for efficient segmentation while retaining crucial information from the original features.

	Silhouette Score	**Davies-Bouldin Score**
Gaussian Mixture	**0.6637**	**0.6058**
Gaussian Mixture (with autoencoder)	**0.7275***	**0.4661***

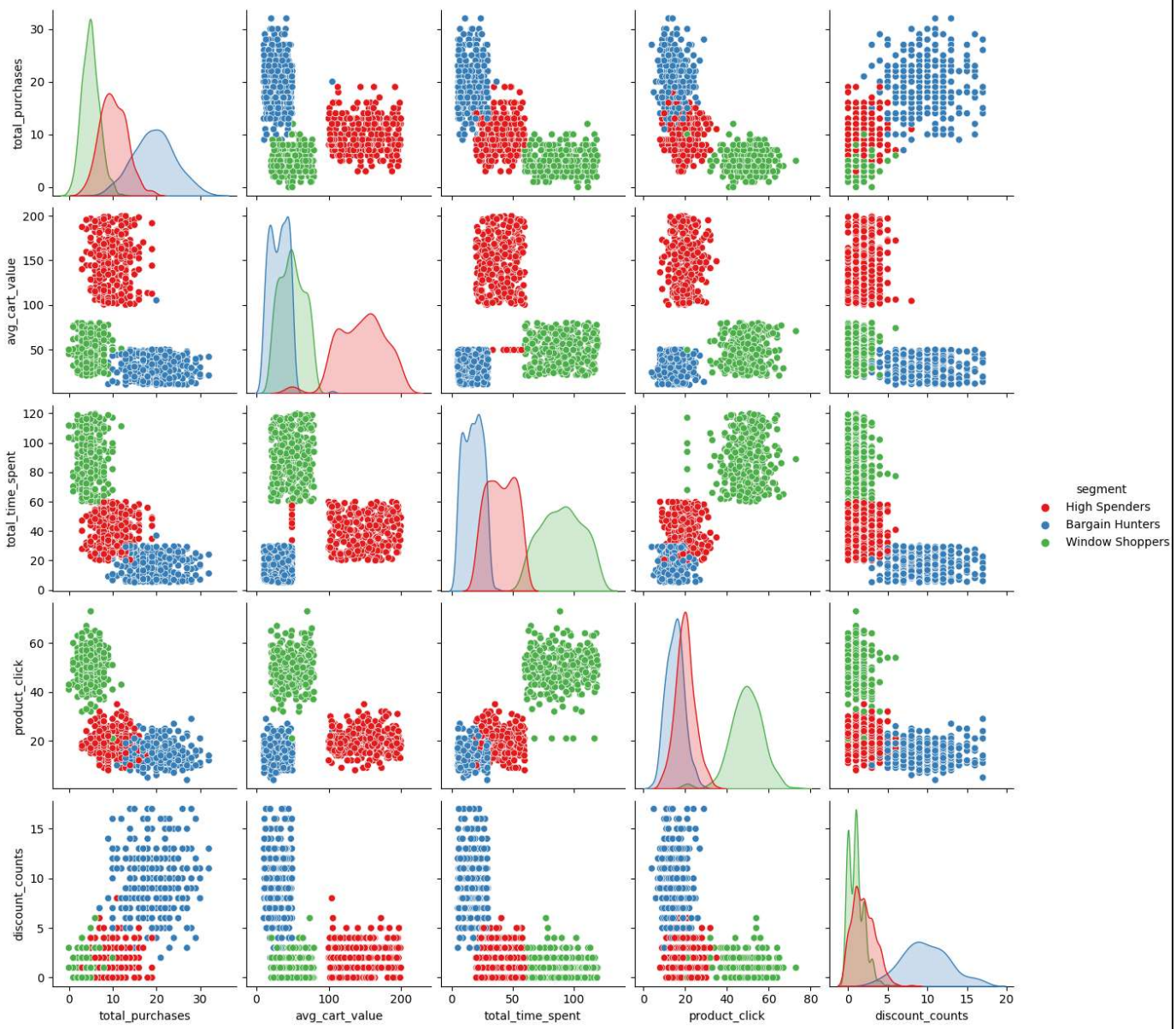
*The GMM with encoding scores keep varying because of the stochastic nature of neural network training in the autoencoder. I got Silhouette scores ranging from 0.66 to 0.76 across different runs. The mentioned value is the one I obtained while writing this report.

- Silhouette Score increased to 0.7275, indicating better-defined clusters.
- Davies-Bouldin Score decreased to 0.4661, showing improved compactness and separation.
- Autoencoder significantly enhanced the clustering quality compared to the previous approach.

7. Final Results

This is the customer segmentation obtained after applying autoencoding.

You can see the distinct separation of clusters representing different customer behaviors.



8. Conclusion

In this study, customer segmentation was performed using various clustering techniques to identify distinct groups based on purchasing behavior, browsing patterns, and discount usage. Initially, traditional clustering methods such as K-Means and Gaussian Mixture Models were applied after careful feature selection and outlier removal. While these methods provided reasonable segmentation, the correlation among certain features led to some redundancy, which impacted clustering performance. To address this, an autoencoder-based approach was introduced to compress features while retaining essential information.

Best approach: Gaussian Mixture with Autoencoder-based feature compression.

The final clustering results demonstrated clear distinctions among Bargain Hunters, High Spenders, and Window Shoppers, with improved Silhouette and Davies-Bouldin scores.