Question 2

Customer Segmentation

Team Name: **DataMavericks**

Table of Contents

1 INTRODUCTION

The problem of customer segmentation in e-commerce platforms is a classic example of an unsupervised learning task, specifically a clustering problem. The goal is to identify distinct groups of customers based on their behavior, which can help in tailoring marketing strategies to meet the needs of each segment effectively.

In this context, clustering algorithms are suitable for segmenting customers into meaningful groups without prior knowledge of the segment labels. KMeans and Agglomerative Clustering are two popular algorithms that can be used for this purpose. KMeans is efficient for large datasets and works well with spherical clusters, while Agglomerative Clustering is more flexible and can handle clusters of varying shapes.

This report will outline the methodology used to identify these segments, evaluate the performance of the clustering models, and provide insights into the characteristics of each segment.

2 PROBLEM STATEMENT

Task: The identification of the customer segments

- Bargain Hunters
- High Spenders
- Window Shoppers

Dataset Features: total_purchases, avg_cart_value, total_time_spent, product_click, discount_count.

Algorithms Considered: KMeans and Agglomerative Clustering.

3 METHODOLOGY

3.1 Libraries used

1. Pandas (pd)
   - Purpose: Data manipulation and analysis.
   - Use: To read and handle the dataset (customer_behavior_analytcis.csv), perform exploratory data analysis, and manage data structures like DataFrames.

2. NumPy (np)
   - Purpose: Numerical computations.
   - Use: To perform various mathematical operations on the dataset, especially when working with arrays.

3. Matplotlib (plt) and Seaborn (sns)
   - Purpose: Data visualization.
   - Use: To create plots and visualizations that help in understanding the distribution of data and the characteristics of customer segments.

4. Missingno (msno)
   - Purpose: Handling missing data.
   - Use: To visualize and identify missing values in the dataset.

5. KNNImputer
   - Purpose: Imputing missing values.
   - Use: To replace missing values with values imputed using the K-Nearest Neighbors algorithm.

6. StandardScaler
   - Purpose: Data scaling.
   - Use: To standardize the features by removing the mean and scaling to unit variance, which is essential for many clustering algorithms.

7. KMeans and AgglomerativeClustering
   - Purpose: Clustering algorithms.
   - Use: To segment customers into distinct groups based on their behavior.

8. Silhouette Score, Calinski-Harabasz Score, Davies-Bouldin Score
    - Purpose: Evaluation metrics for clustering.
    - Use: To assess the quality and separation of the clusters obtained from the clustering algorithms.

9. PCA (Principal Component Analysis) and TSNE (t-Distributed Stochastic Neighbor Embedding)
    - Purpose: Dimensionality reduction.
    - Use: To reduce the dimensionality of the data for better visualization of clusters in a lower-dimensional space. PCA is linear and efficient, while TSNE is non-linear and can capture more complex relationships.

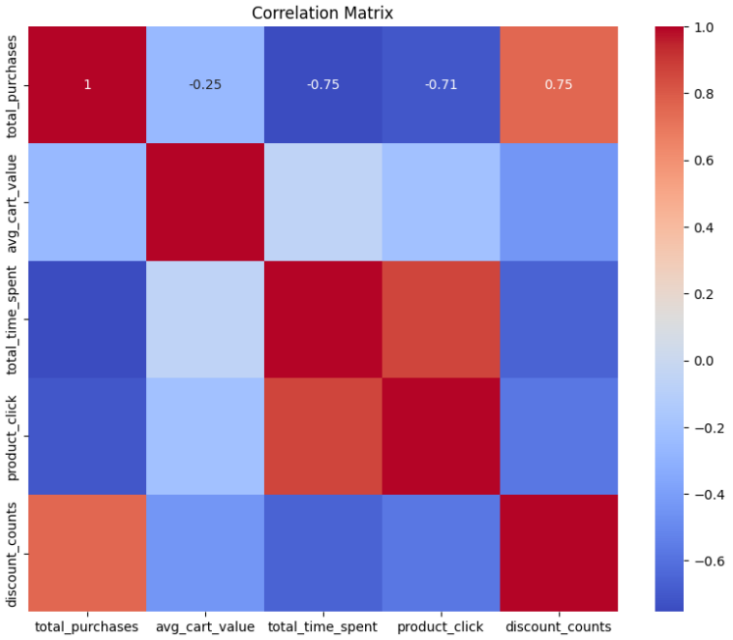
3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in understanding the dataset used for customer segmentation. It involves examining the structure, quality, and initial patterns within the data.

The dataset contains 999 rows and 6 columns, namely: total_purchases, avg_cart_value, total_time_spent, product_click, discount_counts, and customer_id. Upon inspecting the dataset for missing values, it was found that there are 20 missing values distributed across the columns total_purchases, avg_cart_value, and product_click. The visual representation of the missing values is provided below, where the horizontal white lines indicate the presence of missing data in the respective columns.

This is the correlation matrix visualizes. This visualizes the relationships between different features in the dataset by showing their pairwise correlation coefficients. Correlation values range from -1 to 1, where:

- 1 indicates a perfect positive correlation.
- 0 indicates no correlation.
- -1 indicates a perfect negative correlation.



Total Purchases:

- Positively correlated with discount_counts (0.75): Customers who make frequent purchases tend to use discounts more often.
- Negatively correlated with avg_cart_value (-0.25): Frequent buyers generally purchase lower-value items.
- Strong negative correlation with total_time_spent (-0.75) and product_click (-0.71): Customers who buy frequently spend less time browsing and view fewer products.

Average Cart Value:

- Negatively correlated with discount_counts (-0.71): Customers who rely on discounts tend to have lower average cart values.
- Weak negative correlation with total_purchases (-0.25): Higher cart values are associated with fewer purchases.

Total Time Spent:

- Positively correlated with product_click (0.75): Customers who spend more time on the platform tend to view more products.
- Negatively correlated with total_purchases (-0.75): Customers who browse longer make fewer purchases.

Product Clicks:

- Positively correlated with total_time_spent (0.75): Indicates that customers who spend more time browsing also view more products.
- Negatively correlated with total_purchases (-0.71): Frequent buyers view fewer products compared to window shoppers.

Discount Counts:

- Positively correlated with total_purchases (0.75): Bargain hunters use discounts frequently and make more purchases.
- Negatively correlated with avg_cart_value (-0.71): Heavy discount users typically have lower average cart values.

3.3 Data Preprocessing

Data preprocessing is a critical step before applying clustering algorithms. It ensures the dataset is clean, consistent, and suitable for analysis.

1. Handling Missing Values with KNN Imputation

Missing values in the dataset can lead to biased or incomplete analysis if not handled properly. The KNN Imputer (K-Nearest Neighbors Imputer) was used to fill in missing values. KNN Imputer calculates the missing value of a feature by finding the K-nearest neighbors (based on other features) and averaging their values for that feature.

Why KNN Imputer?

- It preserves the relationships between features.
- It is more robust than simple imputation methods like mean or median replacement.

Effect of Not Handling Missing Values:

- Missing data can distort clustering results by introducing noise or reducing the dataset size if rows with missing values are dropped.
- Algorithms like KMeans may fail or produce unreliable clusters due to incomplete data.

2. Outlier Detection and Treatment

Outliers can significantly impact clustering algorithms, especially distance-based methods like KMeans.

Detected Outliers:

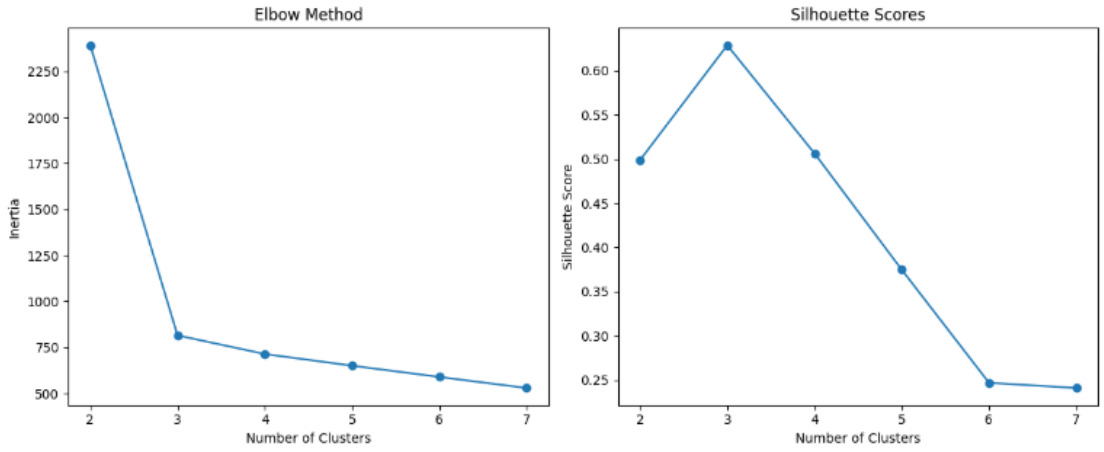
- Extremely high values in avg_cart_value (e.g., customers with unusually large cart sizes).
- Very high or low values in total_time_spent (e.g., customers spending an excessive or minimal amount of time on the platform).

Effect of Not Handling Outliers:

- Outliers can skew cluster centroids, leading to poor segmentation.
- They may cause algorithms like KMeans to form clusters around outliers instead of meaningful customer groups.

3.4 Model Selection

The problem requires segmenting customers into three clusters: Bargain Hunters, High Spenders, and Window Shoppers. While the problem specifies three clusters, it is essential to validate this assumption using clustering evaluation techniques. This ensures that the dataset supports the given number of clusters.



1. Elbow Method

The Elbow Method evaluates the optimal number of clusters by plotting the Within-Cluster Sum of Squares (WCSS) against the number of clusters. The goal is to identify the "elbow point," where adding more clusters results in minimal improvement in WCSS.

- Observation: The elbow point occurs at k = 3, indicating that three clusters provide a good balance between simplicity and accuracy.

2. Silhouette Scores

The Silhouette Score measures how similar a data point is to its own cluster compared to other clusters. It ranges from -1 (poor clustering) to 1 (excellent clustering).

- Observation: The highest Silhouette Score is achieved at k = 3, confirming that three clusters are well-separated and cohesive.

Both methods validate the assumption that the dataset naturally forms three distinct clusters.

3.4.1 Algorithms Used

To identify the best segmentation, two clustering algorithms were applied: KMeans and Agglomerative Clustering.

1. KMeans Clustering

KMeans is a centroid-based algorithm that assigns data points to clusters by minimizing the distance between points and their cluster centroids.

Advantages:

- Efficient for large datasets.
- Works well when clusters are spherical and evenly sized.

Why Used?

- KMeans is computationally efficient and provides clear cluster boundaries, making it a good initial choice for customer segmentation.

2. Agglomerative Clustering

Agglomerative Clustering is a hierarchical algorithm that builds clusters by successively merging or splitting groups based on proximity.

Advantages:

- Does not assume spherical cluster shapes.
- Provides a dendrogram for visualizing cluster relationships.

Why Used?

- This algorithm complements KMeans by handling non-spherical or unevenly sized clusters, ensuring flexibility in identifying natural groupings.

Why Use Two Algorithms?

- By applying these two algorithms, we ensure that the clustering results are robust, reliable, and reflective of actual customer patterns.

3.5 Model Evaluation

In this analysis, KMeans and Agglomerative Clustering were evaluated using three key metrics: Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index.

3.5.1 Evaluation Metrics

a. Silhouette Score

- Measures how similar a data point is to its own cluster compared to other clusters.
- Ranges from -1 (poor clustering) to 1 (excellent clustering).
- Higher values indicate better-defined clusters.

b. Calinski-Harabasz Index

- Measures the ratio of the sum of between-cluster dispersion to within-cluster dispersion.
- Higher values indicate better-separated clusters.

c. Davies-Bouldin Index

- Measures the average similarity ratio of each cluster with its most similar cluster.
- Lower values indicate better clustering.

Metric	KMeans	Agglomerative Clustering
Silhouette Score	0.6285	0.6766
Calinski-Harabasz Index	2543.10	3151.12
Davies-Bouldin Index	0.5447	0.4674

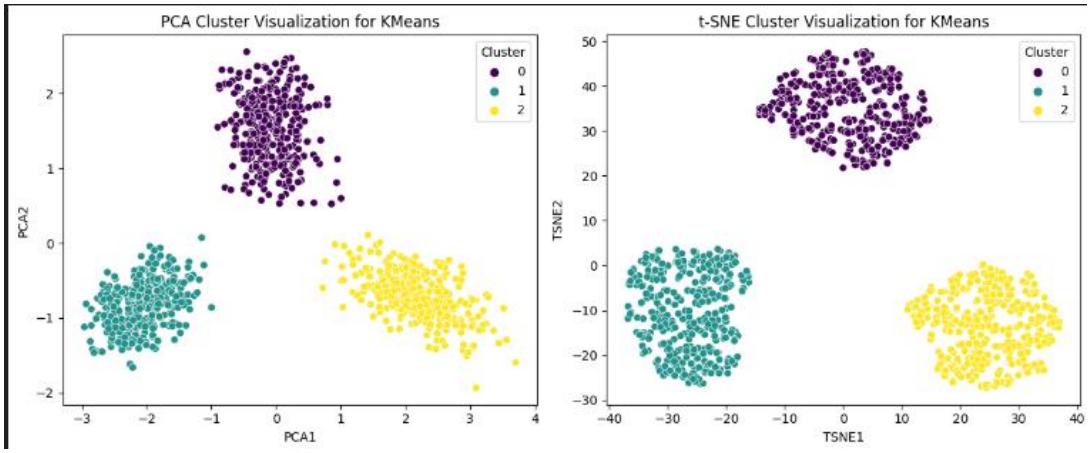
3.5.2 Observations:

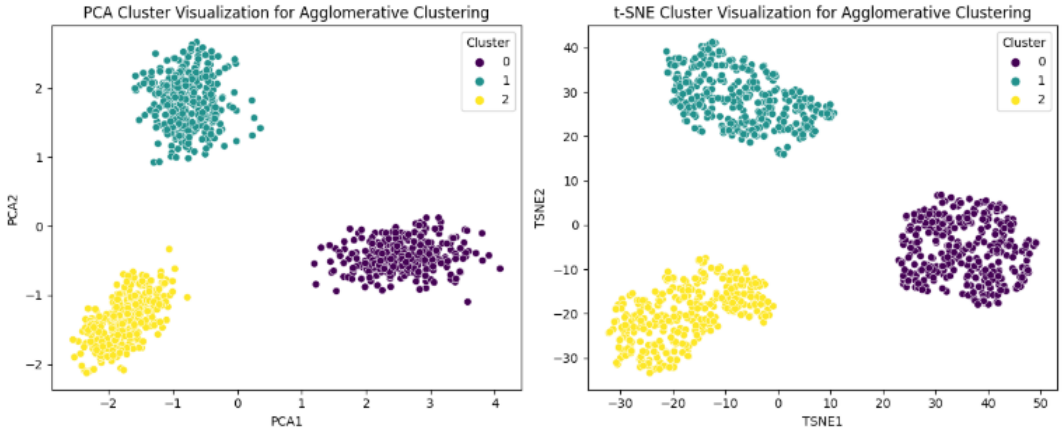
Agglomerative Clustering performed slightly better than KMeans across all metrics:

- Higher Silhouette Score (0.6766 vs. 0.6285).
- Higher Calinski-Harabasz Index (3151.12 vs. 2543.10).
- Lower Davies-Bouldin Index (0.4674 vs. 0.5447).

Both algorithms produced meaningful clusters, but Agglomerative Clustering provided more cohesive and well-separated clusters.

To visualize the high-dimensional data in two dimensions, PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding) were applied





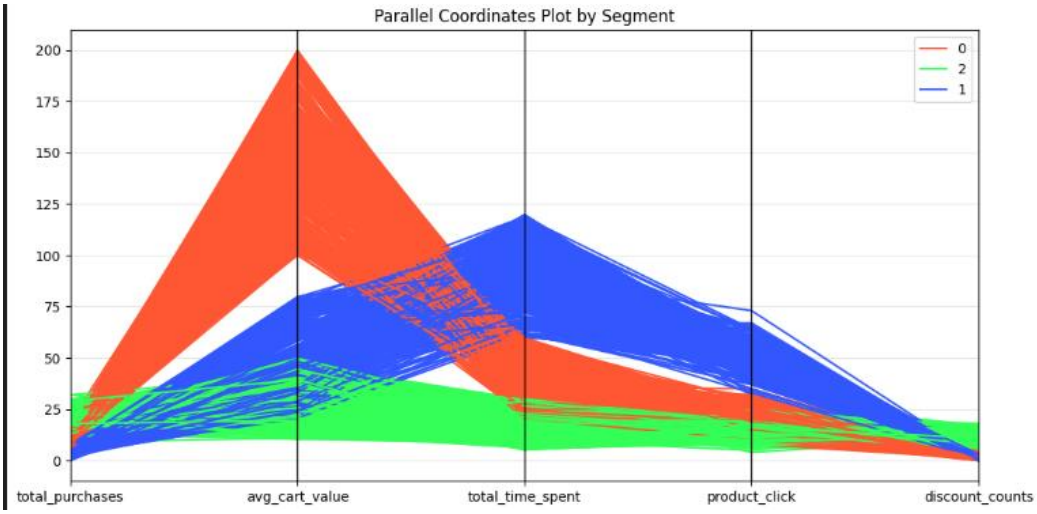
Visualizations:

- The PCA plot (left) shows distinct clusters with minimal overlap.
- The t-SNE plot (right) provides a clearer separation of clusters, highlighting their distinctiveness.

while both algorithms performed well, Agglomerative Clustering showed slightly better results based on evaluation metrics and visualizations, making it the preferred choice for this analysis.

3.5.3 Parallel Coordinates Plot

This Visualizes the average behavior of each cluster across all features.



Key Observations:

- Cluster 0 (red): Represents Bargain Hunters, characterized by high total_purchases, low avg_cart_value, and high discount_counts.
- Cluster 1 (blue): Represents Window Shoppers, with low total_purchases, high total_time_spent, high product_click, and low discount_counts.
- Cluster 2 (green): Represents High Spenders, showing moderate total_purchases, high avg_cart_value, and low discount_counts.

3.6 Cluster Identification and Results

Using the results obtained from both KMeans and Agglomerative Clustering, the customer segments were identified based on their behavioral patterns across the dataset features. The clusters were automatically labeled using ranking logic and mapped to the three predefined customer segments: Bargain Hunters, High Spenders, and Window Shoppers.

Cluster	total_purchases	avg_cart_value	total_time_spent	product_click	discount_counts	Segment
0	10.177177	147.229405	40.389730	19.903303	1.945946	High Spenders
1	4.860661	49.064583	90.144865	49.737538	1.024024	Window Shoppers
2	19.700906	30.466308	17.466858	14.981269	9.906344	Bargain Hunters

4 CONCLUSION

Both KMeans and Agglomerative Clustering performed well in identifying the three distinct customer segments: Bargain Hunters, High Spenders, and Window Shoppers. The evaluation metrics, including Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index, demonstrated that the clusters were cohesive and well-separated. While Agglomerative Clustering slightly outperformed KMeans in terms of metrics, both algorithms validated the segmentation effectively.

4.1 Challenges Faced During the Process

- Ensuring accurate imputation without introducing bias in the missing values handling
- Ensuring that the chosen number of clusters aligns with the dataset's natural structure.
- Balancing computational efficiency with clustering accuracy.

4.2 Suggestions for Improving Model Performance

- Use Advanced Clustering Techniques (DBSCAN or Gaussian Mixture Models (GMM))
- Hyperparameter Optimization
- Ensemble Clustering Methods
- Feature Engineering
- Scaling Data Appropriately (Min-Max scaling or Robust scaling)