**Customer Behavior Analytics**

**Clustering Report**

# 1. Introduction

**Task Overview**

This report presents our approach to the hackathon challenge, which required identifying distinct customer segments based on behavioral data from an e-commerce platform. The goal was to leverage clustering techniques to uncover meaningful customer personas and provide actionable insights for businesses to optimize their marketing strategies, improve customer engagement, and enhance retention.

**Dataset Description**

The dataset provided for this task contains six key features that capture customer interactions, purchases, and browsing patterns:

* **customer\_id**: Unique identifier for each customer.
* **total\_purchases**: Total number of purchases made by the customer.
* **avg\_cart\_value**: Average value of items in the customer's cart.
* **total\_time\_spent**: Total time spent on the platform (in minutes).
* **product\_click**: Number of products viewed by the customer.
* **discount\_count**: Number of times the customer used a discount code.

**Hidden Customer Segments**

The dataset contains three hidden clusters, each representing a distinct customer type. Our task was to accurately identify and visualize these segments,

1. **Bargain Hunters**: Customers who prioritize discounts and make frequent purchases of low-value items.
   * High total purchases
   * Low average cart value
   * Moderate time spent on the platform
   * Moderate number of product views
   * High discount usage
2. **High Spenders**: Premium buyers who focus on high-value purchases and are less influenced by discounts.
   * Moderate total purchases
   * High average cart value
   * Moderate time spent browsing
   * Moderate number of product views
   * Low discount usage
3. **Window Shoppers**: Customers who spend significant time browsing but make very few purchases.
   * Low total purchases
   * Moderate average cart value
   * High time spent on the platform
   * High number of product views
   * Low discount usage

**Objective and Approach**

Our objective in this task was to develop a data-driven clustering model that effectively identifies and visualizes these customer segments. We applied K-Means clustering, a widely used unsupervised machine learning algorithm, to uncover the hidden structures in the data. The clustering results were then evaluated and visualized to ensure clear segment separations.

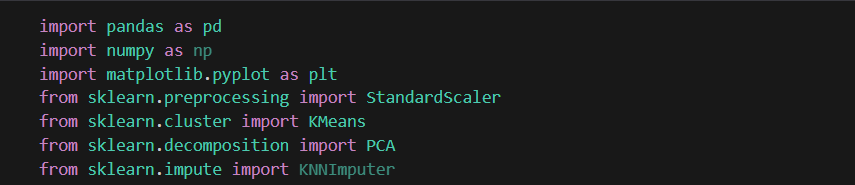
This segmentation approach provides businesses with a strong analytical foundation to enhance customer targeting, refine product recommendations, and improve retention strategies.

# 2. Methodology

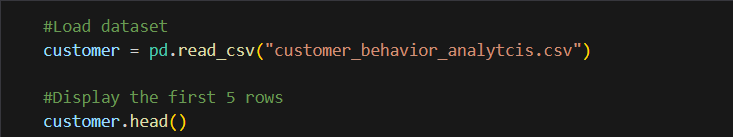
**2.1 Data Preprocessing**

Before applying clustering, the dataset needed to be cleaned and standardized to ensure meaningful results.

1. **Importing Required Libraries**

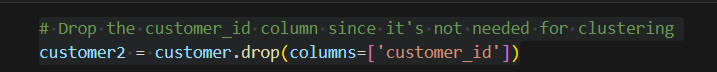
The project begins by importing necessary Python libraries such as pandas for data manipulation, numpy for numerical computations, and visualization tools like matplotlib .Additionally, machine learning libraries such as sklearn.preprocessing and sklearn.cluster are utilized.

1. **Loading the Dataset**  
   The dataset, **customer\_behavior\_analytcis.csv**, was loaded using pandas to inspect the data structure.

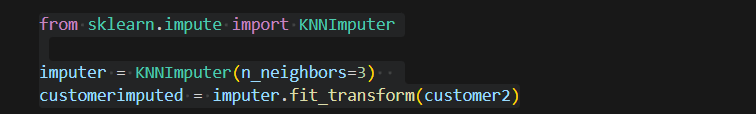


1. **Dropping Unnecessary Columns**

The **customer\_id** column is removed as it does not contribute to clustering.

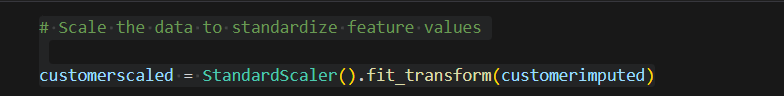


1. **Handling-Missing-Data**  
   Some attributes had missing values, which can affect clustering performance. Instead of removing missing data (which could reduce sample size), the K-Nearest Neighbors Imputer (KNN Imputer) was used to estimate missing values based on the nearest neighbors(03).



This technique ensures that missing values are filled based on patterns observed in the dataset.

1. **Feature-Scaling**  
   Since K-Means clustering relies on distance-based calculations, The dataset is standardized using **StandardScaler** to ensure all numerical features have a mean of zero and a unit variance.

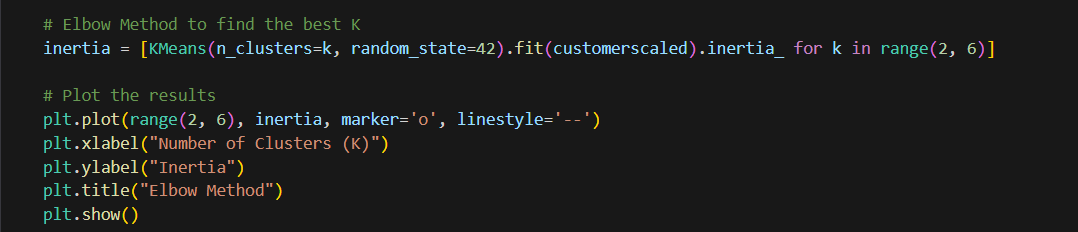
****

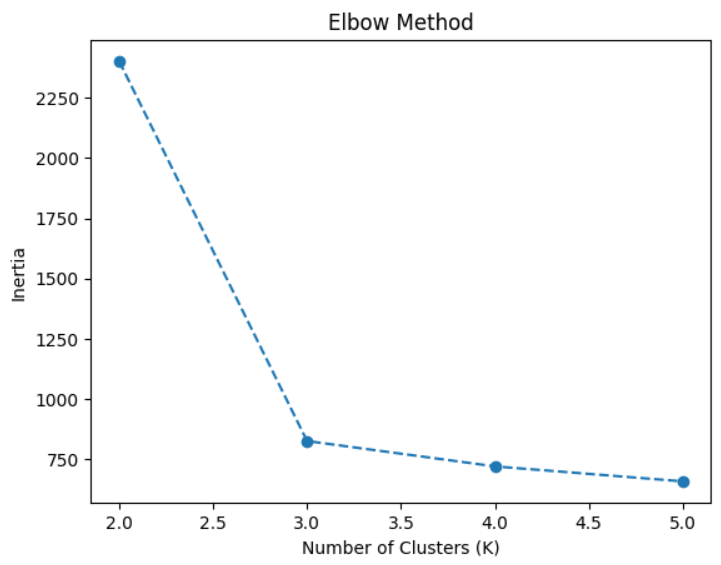
This transformation ensures that all features contribute equally to the clustering process.

* 1. **Clustering with K-Means**

1. **Elbow Method to Find Optimal K-Justification for choosing K=3**

The **Elbow Method** is a technique used to determine the optimal number of clusters (**K**) for K-Means clustering. The idea is to compute the **inertia** (sum of squared distances from each point to its assigned cluster center) for different values of K and find the point where the decrease in inertia slows down, forming an "elbow" in the graph.

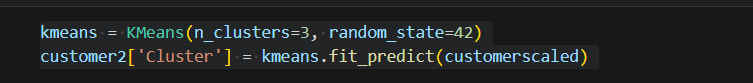
We calculated the inertia values for **K=2 to K=5** and plotted them to visualize the rate of decrease. The plot revealed a sharp drop in inertia up to **K=3**, after which the reduction in inertia slowed down-significantly.



 In our case, the "elbow" appears at **K=3**, indicating that the optimal number of clusters is 3. Given that the task specified 3 clusters, the Elbow Method confirmed that **K=3** is indeed the optimal number of clusters for this dataset, aligning with the requirements. This ensures that the clustering model is both accurate and efficient for the given problem.

1. **Performing K-means clustering**

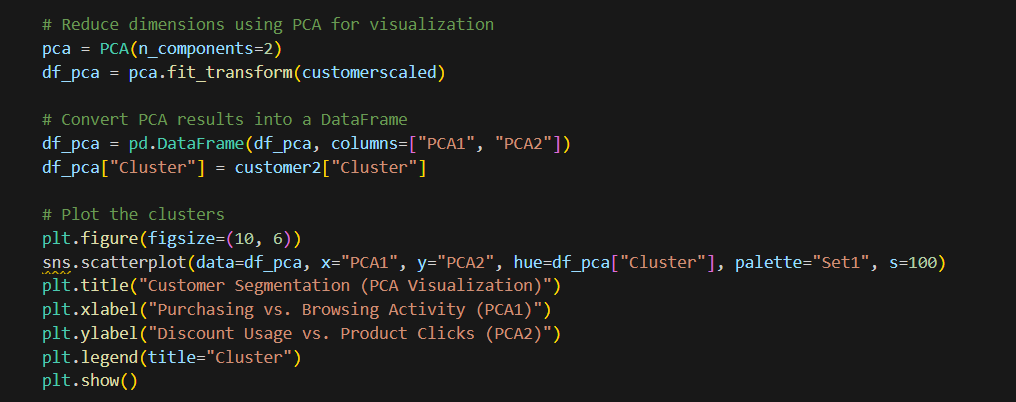
In this step, we applied the KMeans clustering algorithm with 3 clusters to the scaled customer data (customerscaled). The model was trained to assign each customer to one of the 3 clusters based on their similarities. The resulting cluster labels (0, 1, or 2) were stored in a new column, 'Cluster', in the customer2 DataFrame, allowing us to identify the cluster each customer belongs to.

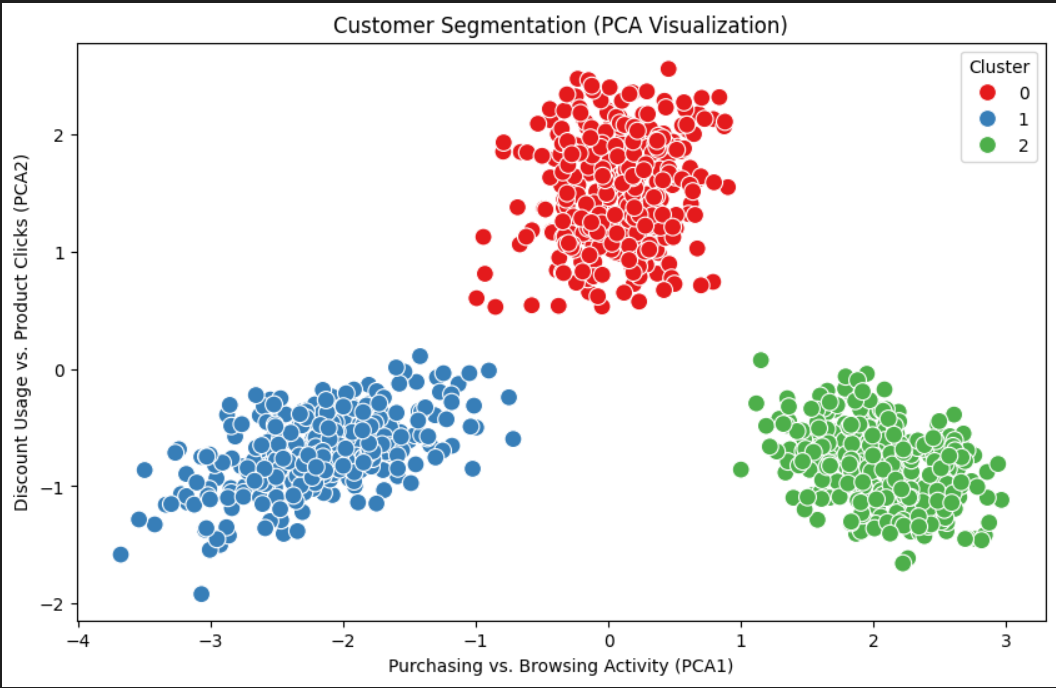


1. **Visualizing segments**

In this step, we applied **Principal Component Analysis (PCA)** to reduce the customer data (**customerscaled)** to two dimensions for visualization. Using **n\_components=2**, the data was transformed into two principal components, stored in df\_pca. The cluster labels from KMeans were added to the DataFrame to color-code the data points based on their cluster.

We then created a **scatter plot** with **PCA1** and **PCA2** on the x and y axes, respectively, and used color coding to represent the different customer clusters. The axes, **PCA1** and **PCA2**, represent the most significant features captured by PCA, which combine factors like purchasing activity, browsing, discount usage, and product clicksThis visualization helps to clearly display the customer segments in a 2D space, making it easy to observe how the clusters are separated.



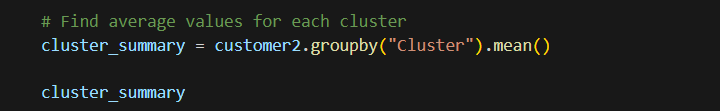


**Interpretation**

* **Cluster 0 (Red):** These customers have higher values on PCA2 (Discount Usage vs. Product Clicks) and moderate values on PCA1 (Purchasing vs. Browsing Activity). This suggests they are active in using discounts and clicking on products, but their purchasing behavior might not be as pronounced.
* **Cluster 1 (Blue):** These customers have lower values on both PCA1 and PCA2. This suggests they are less active in both purchasing/browsing and discount usage/product clicks.
* **Cluster 2 (Green):** These customers have higher values on PCA1 (Purchasing vs. Browsing Activity) and moderate values on PCA2 (Discount Usage vs. Product Clicks). This suggests they are more active in purchasing and browsing, with moderate engagement in discounts and product clicks.

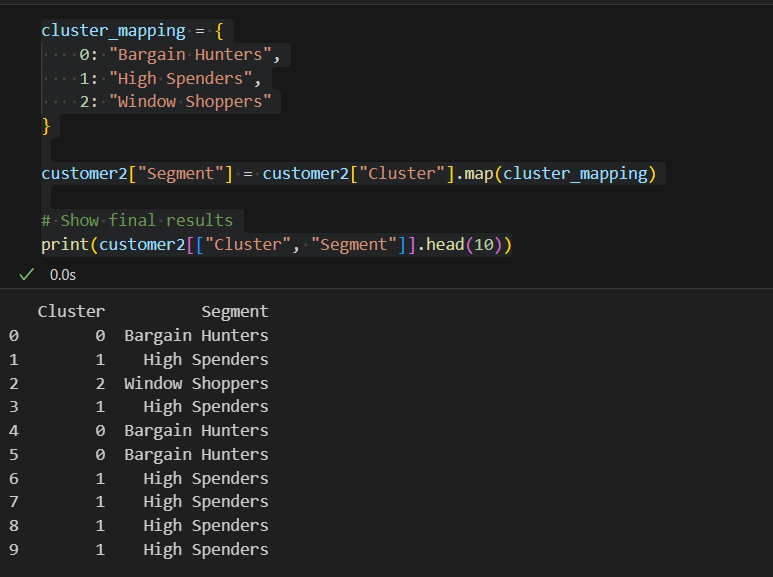
**Summarizing the Average Values for Each Cluster**

This step calculates the average values for each cluster based on the features in the customer2 DataFrame.



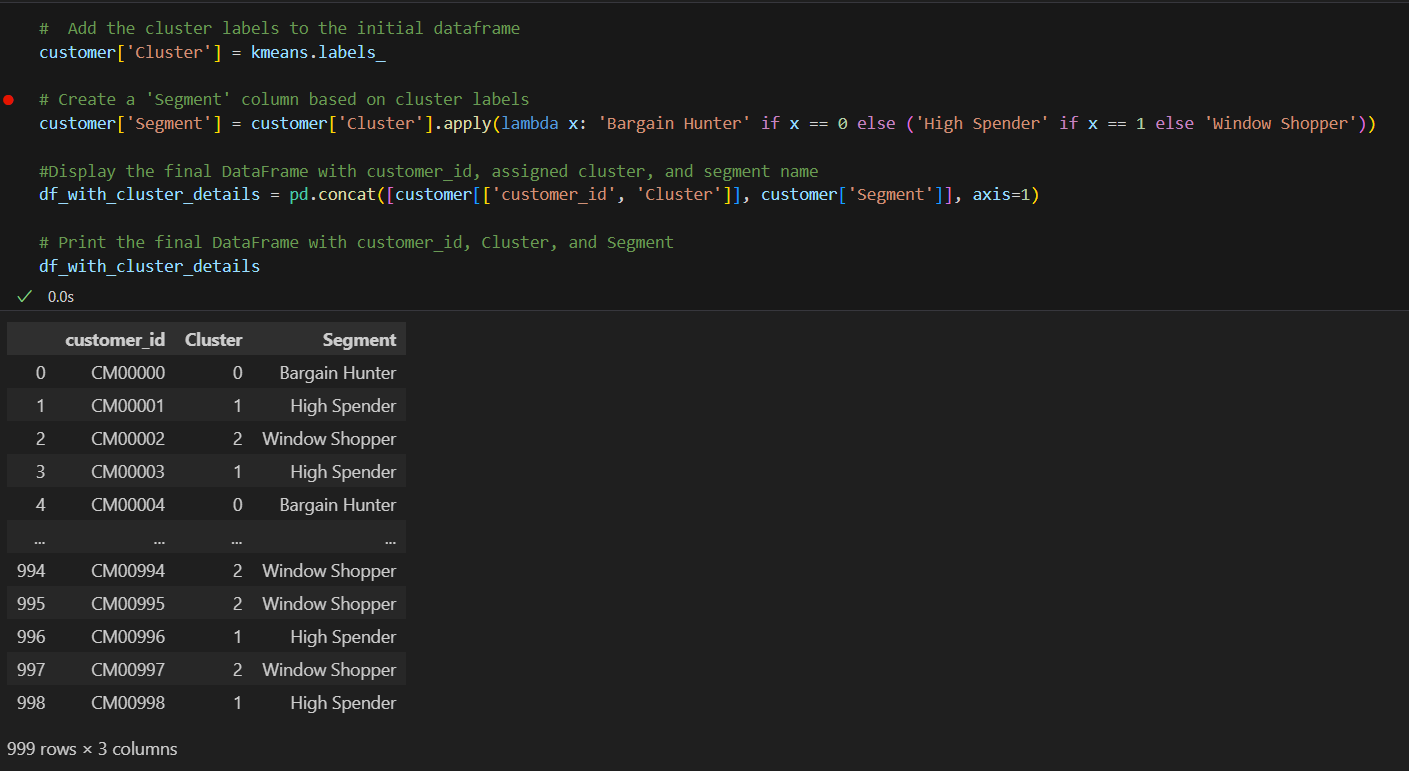
1. **Mapping Cluster Labels to Descriptive Segment Names**

This step maps the cluster labels (0, 1, and 2) to more descriptive segment names (such as "Bargain Hunters", "High Spenders", and "Window Shoppers"). The map() function is used to assign the appropriate segment name to the Segment column based on the Cluster labels.



1. **Assigning Cluster Labels and Segment Names to Customers**

This code assigns cluster labels and segment names to the original customer data based on the results from the KMeans clustering algorithm. First, the cluster labels generated by KMeans are added to the customer DataFrame in a new column called Cluster. Then, a Segment column is created, where each customer is assigned a segment name: 'Bargain Hunter' for cluster 0, 'High Spender' for cluster 1, and 'Window Shopper' for cluster 2, using the apply() function. The customer ID, cluster label, and segment name are then combined into a new DataFrame, df\_with\_cluster\_details, providing a clear overview of each customer’s assigned cluster and segment. Finally, the resulting DataFrame is printed, showing the customer ID, cluster, and segment information.



# 3. Challenges Faced

### ****Struggling with Visualizing Clusters Using PCA****

The challenge we faced was visualizing the customer clusters using **Principal Component Analysis (PCA)**. PCA helps reduce complex data with many features into just two dimensions for easier visualization. However, understanding how to use PCA to display the clusters in 2D was tough. Sometimes, when reducing data to two dimensions, important patterns might get lost, making it harder to clearly see the clusters. Since we were new to PCA, it took extra effort to understand how it affects the clustering results and how to create a clear and meaningful visualization without losing key information.

# 4. Insights and Findings

 The PCA has effectively separated customers into three distinct behavioral groups based on their purchasing, browsing, discount usage, and product interaction patterns.

 The clear separation of clusters allows for tailored marketing strategies.

* **Cluster 0:** Focus on discount-driven campaigns and promotions.
* **Cluster 1:** Investigate the reasons for low engagement and try to re-engage them with personalized offers or content.
* **Cluster 2:** Focus on nurturing loyalty and providing a premium experience.

 The Cluster 2 represents the most valuable segment based on their high purchasing and browsing activity.

 Cluster 1 presents an opportunity to understand and address the factors contributing to their low engagement. This could involve improving the user experience, offering personalized recommendations, or addressing potential issues.

# 5.Suggestions for Improvement

* Including behaviors like purchase frequency and time spent browsing gives more depth to clustering. For instance, frequent buyers who browse little can be classified as High Spenders, while those who browse but rarely buy might be Window Shoppers. This provides a clearer picture of customer behavior.
* Tuning KMeans can improve clustering. Using KMeans++ for initialization and running multiple attempts with different initializations helps find better and more stable clusters, avoiding poor results.
* After clustering, examine customer behaviors in detail and consult with marketing teams to ensure labels (e.g., Bargain Hunters, High Spenders) reflect the group’s true characteristics.
* Customer behavior changes over time, so it's important to update clusters periodically to reflect these shifts. This keeps customer segments relevant and helps businesses stay on target with personalized strategies.

# 6. Instructions for Running the Program

Install dependencies-

pip install pandas numpy matplotlib seaborn scikit-learn

Ensure customer\_behavior\_analytcis.csv is in the same directory as the script.

Run the Jupyter Notebook or Python script.

Analyze the visualizations and clustering results.

# 7. Conclusion

In this machine learning task, we used **KMeans clustering** to group customers based on their shopping habits, how often they browse, and how they use discounts. We found three main customer groups: **Bargain Hunters**, **High Spenders**, and **Window Shoppers**. By using **PCA** (a technique to simplify the data), we could visualize and understand these groups better.

Although we faced some challenges in visualizing the clusters and giving them clear labels, we were able to improve the results by analyzing the data more deeply. The insights from this task can help businesses create better marketing strategies that are more targeted to each customer group.

This task showed how powerful clustering can be for understanding customers and how using the right data analysis techniques can lead to more effective and personalized marketing. It also demonstrated that with continuous improvement, customer segmentation can evolve to match changing customer behaviors.