IntelliHack 5.0 Task 02

Understanding the Data

The given dataset contains the data about customer behavior in an e-commerce platform. This dataset contains information about customer interactions, purchases and browsing patterns.

Our main goal is to identify the distinct customer segments based on their behaviour.

This dataset contains six features. Let's go through each feature.

01) customer_id

- This dataset includes 999 unique customers, each identified by a unique customer ID.

02) total_purchases

- This represents the total number of purchases made by a customer.
- The number of purchases ranges from 0 to 32.

03) avg_cart_value

- This indicates the average value of items in the customer's cart.
- The values range from 10.26 to 199.77.
- The average cart value is 75.45.

04) total_time_spent

- This represents the total time (in minutes) spent on the platform.
- The values range from 5.12 minutes to 119.82 minutes.
- On average, a customer spends 49.34 minutes on the platform.

05) product_click

- This shows the number of products viewed by a customer.
- The number of views ranges from 4 to 73, with an average of 28 views per customer.

06) discount_counts

- This represents the number of times a customer used a discount code.
- The maximum number of times a customer used a discount code is 21.

Summary Statistics of the data collected,

	total_purchases	avg_cart_value	total_time_spent	product_click	discount_counts
count	979.000000	979.000000	999.000000	979.000000	999.000000
mean	11.570991	75.457978	49.348759	28.237998	4.313313
std	7.016327	55.067835	32.730973	16.296384	4.532772
min	0.000000	10.260000	5.120000	4.000000	0.000000
25%	6.000000	33.130000	22.375000	16.000000	1.000000
50%	10.000000	49.380000	40.360000	21.000000	2.000000
75%	17.000000	121.255000	77.170000	45.000000	8.000000
max	32.000000	199.770000	119.820000	73.000000	21.000000

Data Cleaning and Preprocessing

Data duplicates

First we checked if there were any duplicates in the dataset and found that there are no any duplicates in the dataset.

Missing Values

Then we checked whether there are any missing values in the dataset.

There are few missing values from three columns in the dataset.

	Missing Values	% of Total Values
total_purchases	20	2.000000
avg_cart_value	20	2.000000
product_click	20	2.000000

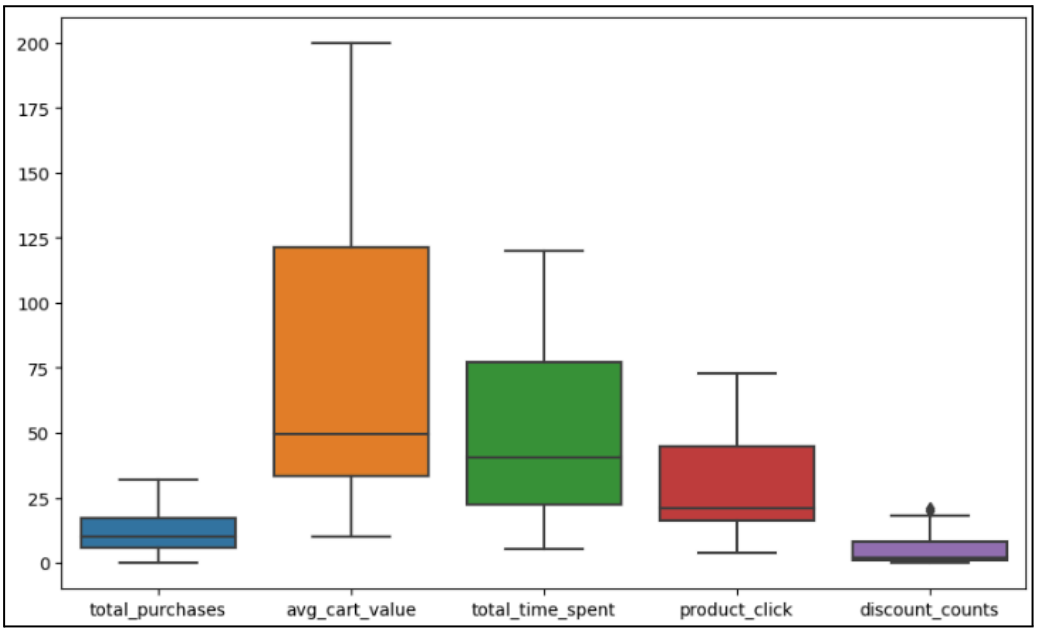
Missing values account for 2% of the data. Therefore, dropping entire columns is not advisable. Instead, it is better to impute the missing values using an appropriate value based on the respective column.

We filled that missing data,

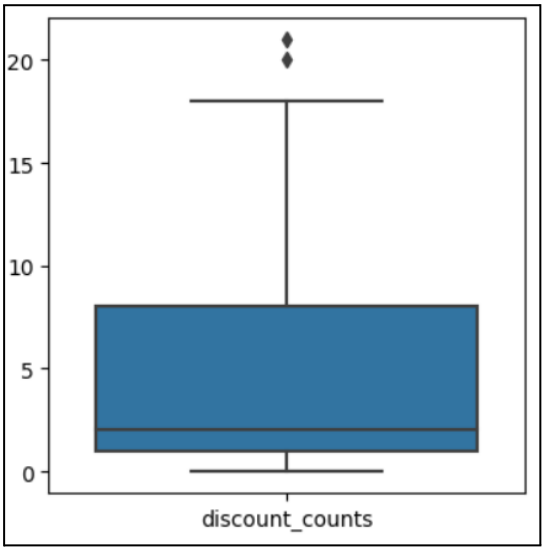
- Total_purchases -> median value (total_purchases are discrete values)
- Avg_cart_value -> mean (avg_cart_value are continuous values)
- product_click->median (product_click are discrete values)

Checking for Outliers

We plot the graph to check whether there are any outliers in the dataset.



We can see that there are small no of outliers in the discount_counts column.



So, we removed those outliers from the discount_counts.

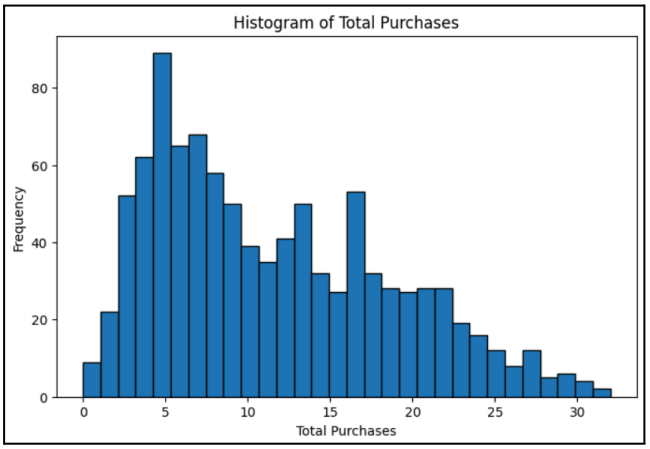
Feature Scaling

Most clustering techniques, including K-Means, are sensitive to scale. Therefore, it is essential to normalize the data to ensure accurate and meaningful clustering results.

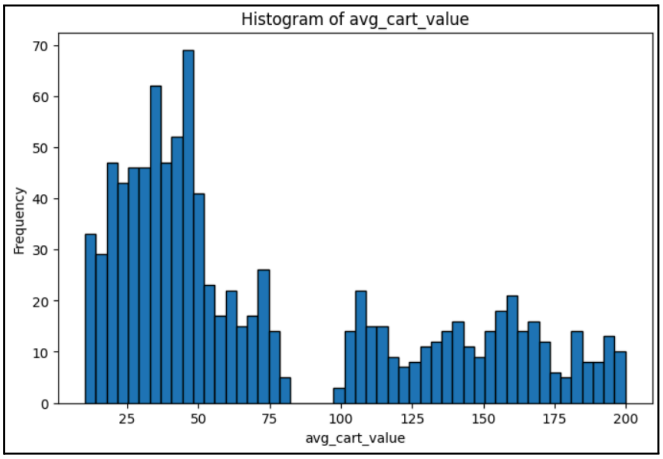
Data Visualization

First, we examine how each feature in the dataset is distributed.

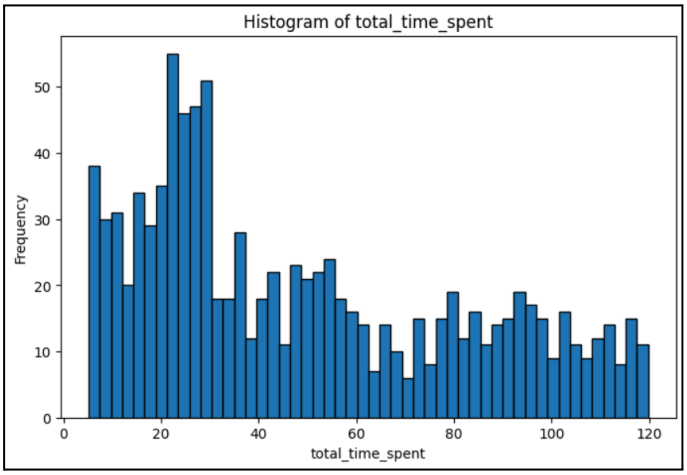
01) Total_purchases



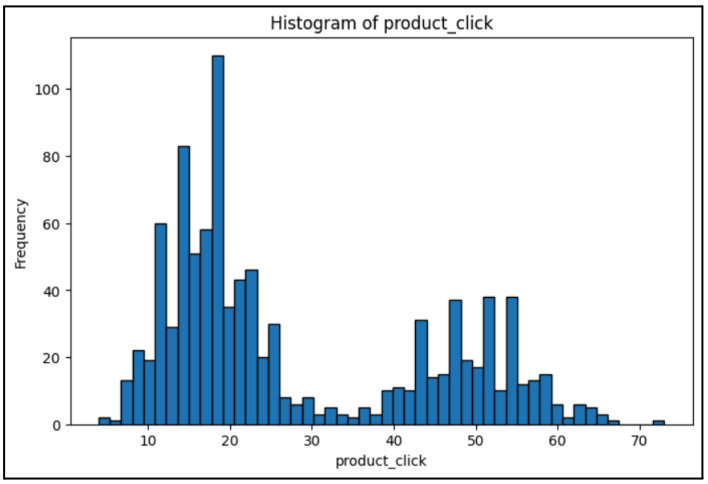
02) avg_cart_value



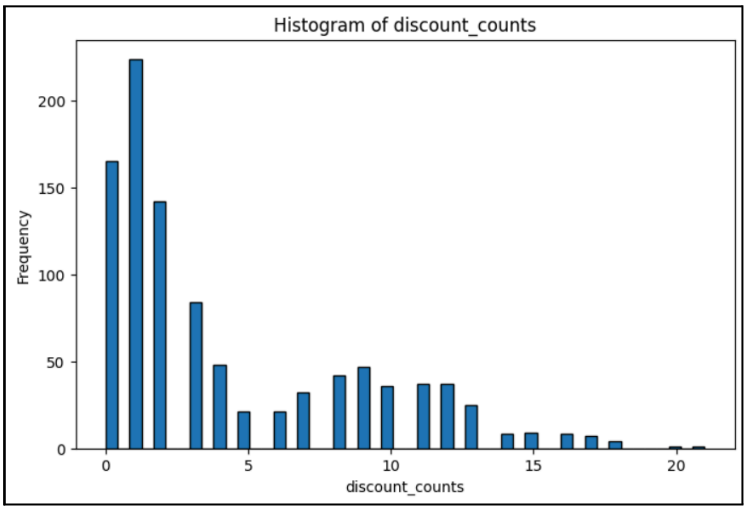
03)total_time_spent



04)product_click



05)discount_counts



Cluster Identification

We divided the dataset into 3 clusters to find the customer classification according to the given three groups. These are the values we got after getting 3 different clusters.

Cluster	total_purchases	avg_cart_value	total_time_spent	product_click	discount_counts	Cluster
0	19.551515	31.228020	17.433697	15.090909	9.915152	0.0
1	10.169643	144.977412	40.560149	19.931548	1.949405	1.0
2	4.909366	49.270237	90.236314	49.456193	1.030211	2.0

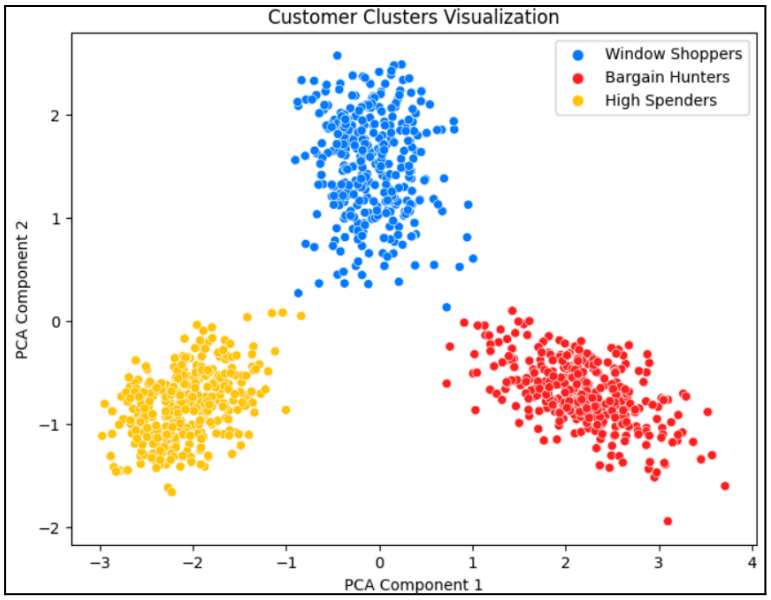
According to the given problem, we need to identify which customer group are suitable for each above group using given data

Feature	Bargain Hunters	High Spenders	Window Shoppers
Total Purchases	High	Moderate	Low
Avg Cart Value	Low	High	Moderate
Time Spent	Moderate	Moderate	High
Product Clicks	Moderate	Moderate	High
Discount Usage	High	Low	Low

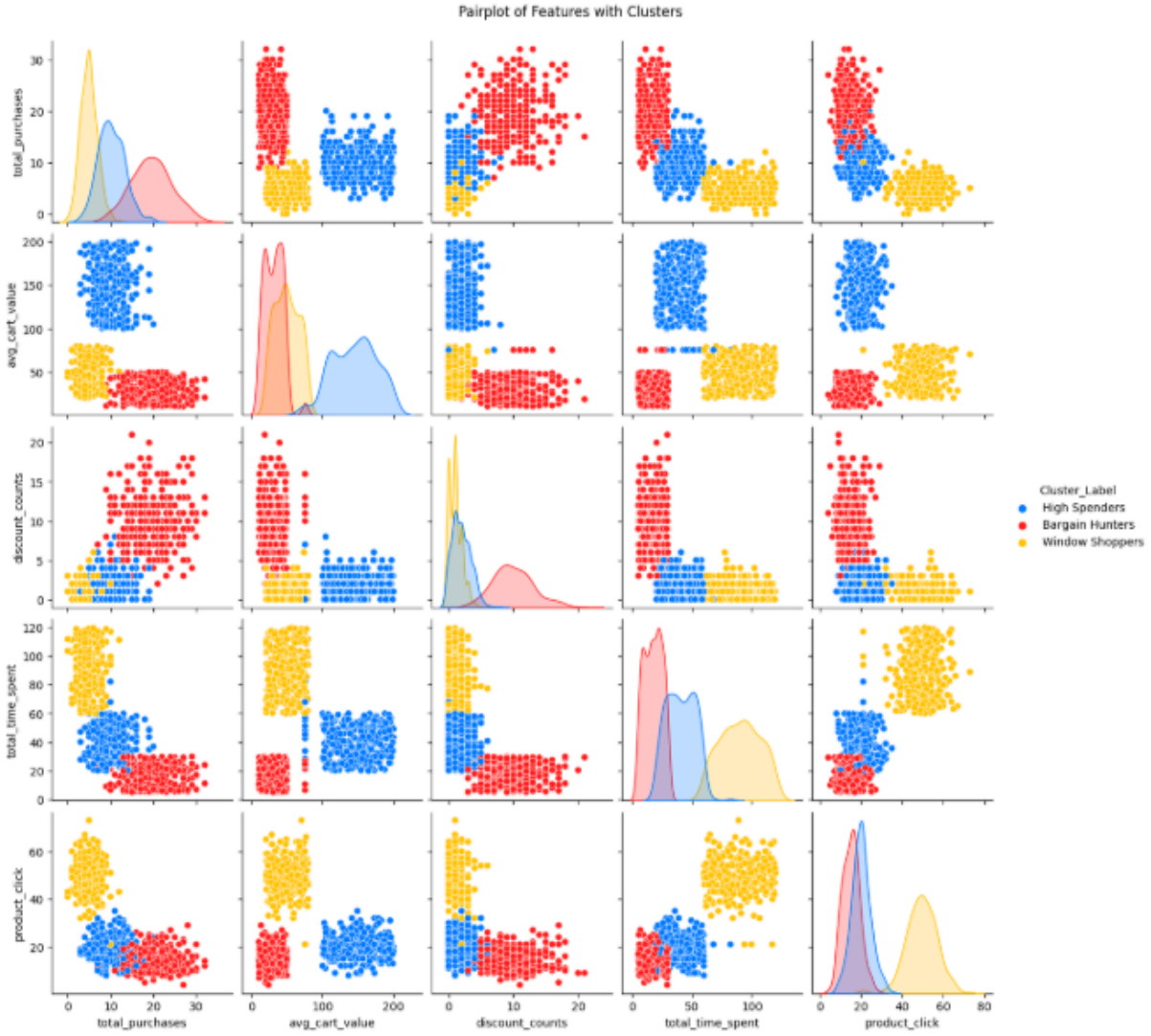
Comparing these data, finally we come up with these cluster assignments.

1. Cluster 0 - Bargain Hunters
2. Cluster 1 - Window Shoppers
3. Cluster 2 - High Spenders

This is how the these 3 clusters are arranged in the dataset.



After we dividing the clusters, we check how the each column data distribution



Model Selection

For this, we used 5 different clustering techniques

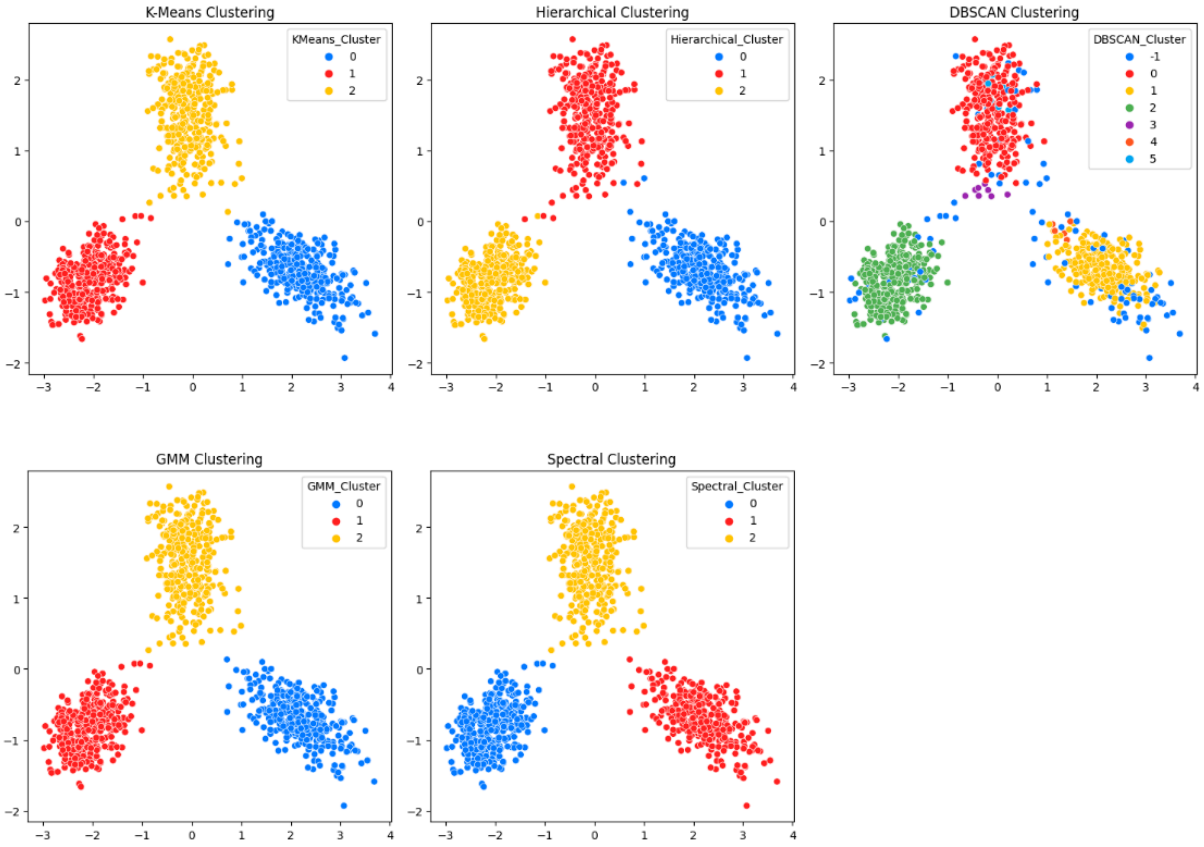
01) Kmeans clustering

02)Hierarchical Clustering

03)DBSCAN

04)Gaussian Mixture Model

05 Spectral Clustering



Here, we used two different metrics to evaluate those different models. They are,

- silhouette_score
- davies_bouldin_score

This is how we got the values for each metric

Clustering Algorithm	Silhouette Score	Davies-Bouldin Index
K-Means	0.614	0.568
Hierarchical	0.612	0.572
DBSCAN	0.231	1.719
GMM	0.614	0.568
Spectral	0.614	0.568

Silhouette Score

The Silhouette Score measures how similar each point is to its own cluster compared to other clusters.

K-Means, GMM, and Spectral Clustering: These algorithms all got a score of 0.614, which is pretty good. This means they did a nice job of grouping the data into clusters. The points are well separated, and most points are in the right groups.

Hierarchical Clustering: It got a score of 0.612, which is almost the same as the others. This means it also did a good job of clustering the data.

DBSCAN: It got a much lower score of 0.231, which means it didn't do as well. It might be having trouble with noisy data or figuring out how dense the clusters are, so it's not grouping the points as accurately as the other methods.

Davies-Bouldin Index

The Davies-Bouldin Index is used to evaluate the average similarity ratio of each cluster with the cluster that is most similar to it.

K-Means, GMM, and Spectral Clustering: These algorithms all got a DBI score of 0.568, which is low and good. This means the clusters they created are well-separated and distinct, so the groups are clear and not overlapping too much.

Hierarchical Clustering: It got a slightly higher DBI score of 0.572, which is still pretty good but not as great as the others. This means its clusters might be a little less compact or slightly less well-separated, but the difference is small.

DBSCAN: It got a much higher DBI score of 1.719, which is bad. This means its clusters are not well-defined—they might be overlapping a lot, or DBSCAN might be struggling with noise (outliers) in the data.

According to the these values, we can say that,

- K-Means, GMM and Spectral Clustering are the best performing algorithms to these datasets
- Hierarchical Clustering is a moderate performing algorithm
- DBSCAN is not suitable for this dataset