Task 02 - Customer Segmentation
Team Duo Dynamics

01. EDA - Exploratory Data Analysis

First, we examined the mean, standard deviation, and other statistical measures for each feature in the dataset to gain a clear understanding of their distributions. As observed, the ranges of the features vary significantly, making normalization essential before performing clustering to ensure fair comparisons between data points.

	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

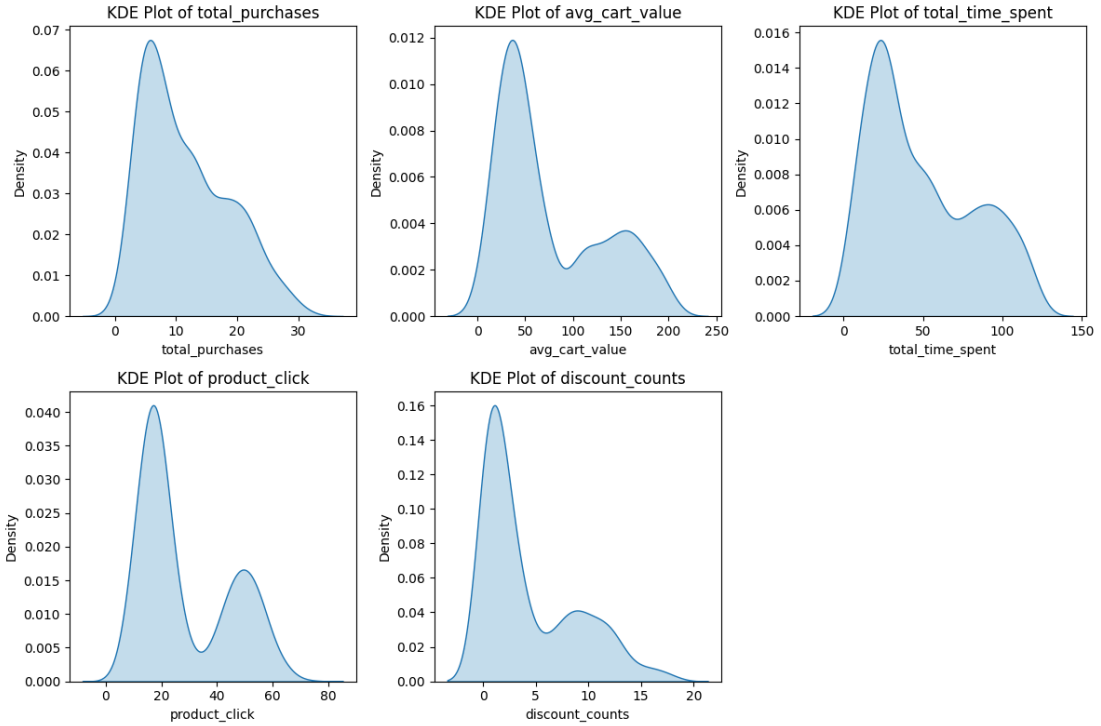
- ***Normalized dataframe***

	total_purchases	avg_cart_value	total_time_spent	product_click	discount_counts
count	9.970000e+02	9.970000e+02	9.970000e+02	9.970000e+02	9.970000e+02
mean	1.140289e-16	-9.175765e-17	1.425362e-16	-5.701446e-17	7.839489e-17
std	1.000502e+00	1.000502e+00	1.000502e+00	1.000502e+00	1.000502e+00
min	-1.659326e+00	-1.185808e+00	-1.352882e+00	-1.669510e+00	-9.562820e-01
25%	-7.957374e-01	-7.625296e-01	-8.255122e-01	-7.055247e-01	-7.328956e-01
50%	-2.200116e-01	-4.672234e-01	-2.743100e-01	-4.645283e-01	-5.095092e-01
75%	7.875086e-01	7.963867e-01	8.494826e-01	1.041699e+00	8.308092e-01
max	2.946481e+00	2.283719e+00	2.151713e+00	2.728673e+00	3.064673e+00

- Normalization is a technique that is applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale.
- In clustering analyses, standardization may be especially crucial in order to compare similarities between features based on certain distance measures. Another prominent example is the Principal Component Analysis, where we usually prefer standardization over normalization since we are interested in the components that maximize the variance

❖ *KDE Plots for features*

A KDE plot is a way to visualize the probability density function (PDF) of a continuous variable. It helps us understand the distribution of data points, including its shape, central tendency, and spread.



1. Total Purchases: Right-skewed, meaning most users have low purchase counts, but a few users make many purchases

2. Average Cart Value: Right-skewed, likely due to some users spending significantly more than others.

3. Product Clicks & Discount Counts: Appear bimodal, meaning two distinct user behaviors exist (e.g., casual browsers vs. engaged shoppers).

4. Total Time Spent: Also skewed, indicating some users spend much more time on the platform than other

❖ *Handling missing values and outliers*

	name of column	types	unique_data	missing value	missing percentage	duplicated
0	total_purchases	float64	32	20	2.000000	0
1	avg_cart_value	float64	943	20	2.000000	0
2	total_time_spent	float64	953	0	0.000000	0
3	product_click	float64	64	20	2.000000	0
4	discount_counts	float64	21	0	0.000000	0

Next, we examined the dataset for missing values and duplicate entries, as these can negatively impact model performance if not handled appropriately.

For missing values, we applied the following strategy:

- If a feature had more than 50% missing values, it would be removed, as imputing such a large proportion of missing data could introduce bias and reduce reliability.
- If the percentage of missing values was relatively low, we imputed them using appropriate statistical measures to preserve data integrity.

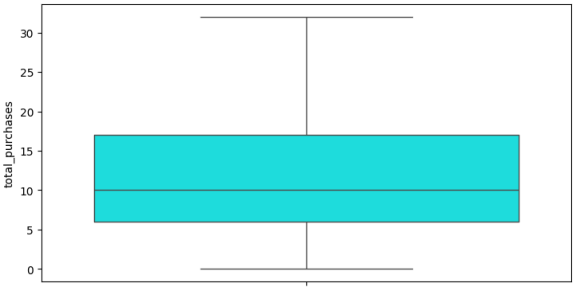
In this dataset, only the features 'total_purchases,' 'average_cart_value,' and 'product_click' contained missing values, with a missing percentage of just 2%.

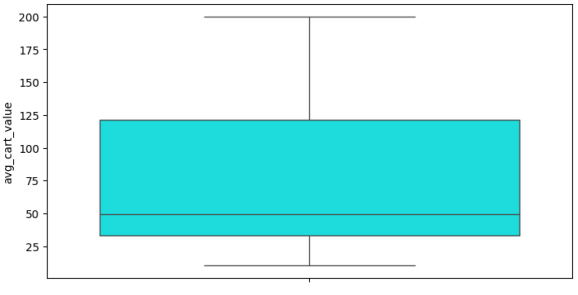
- 'Total purchases' was imputed with zero, as missing values likely indicate customers who had not made any purchases.
- 'Average cart value' and 'product click' were imputed using the median, as these features have skewed distributions, and the median is more robust to outliers compared to the mean.

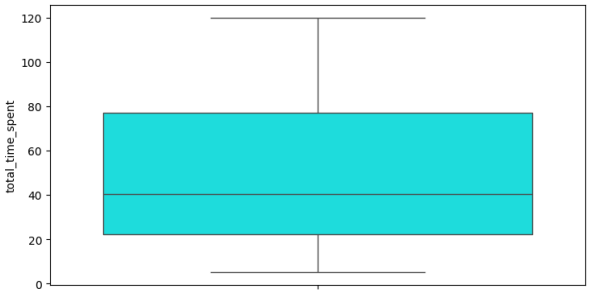
To clean the data, we need to identify and handle outliers. We used box plots to visualize the presence of outliers in the features.

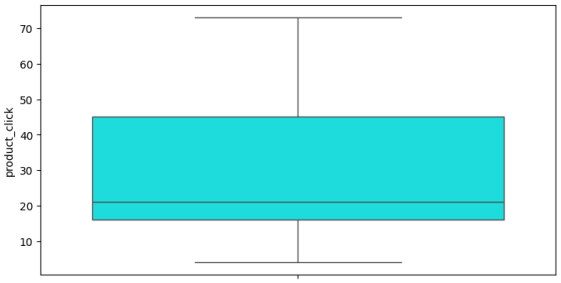
From the box plots, we observed that only the **discount_counts** feature contains outliers. To address this, we applied the **Interquartile Range (IQR) method** as the threshold to detect and remove these outliers.

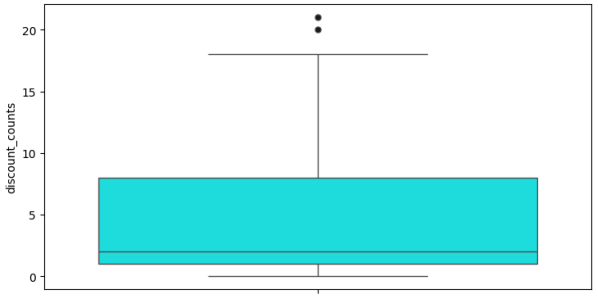
By removing values that fall outside the IQR-defined range, we ensure that extreme values do not skew the analysis while preserving the integrity of the dataset.

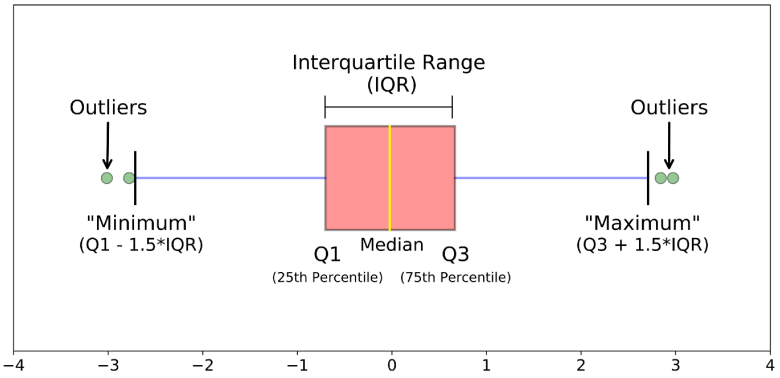




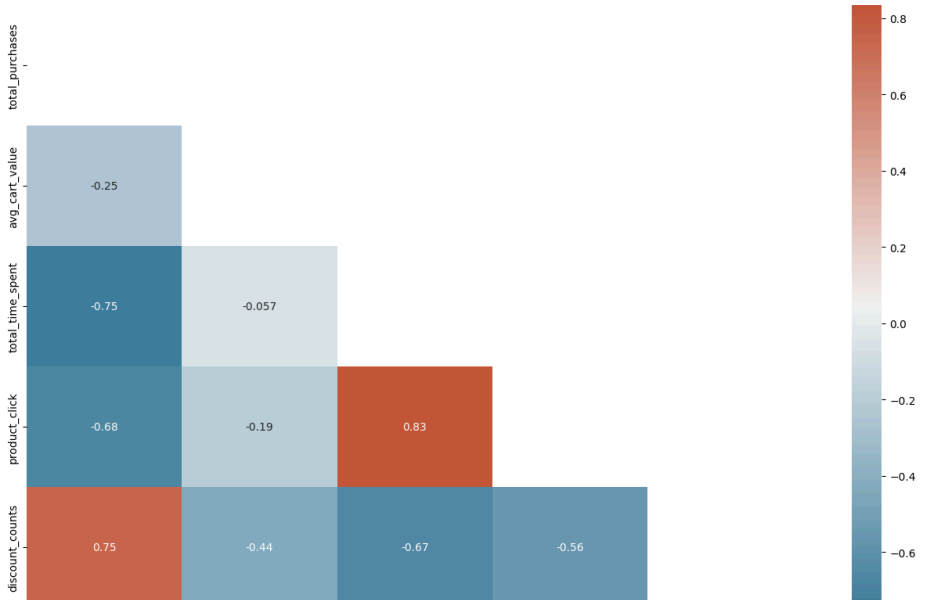








❖ *Correlation matrix heatmap*



Strong Positive Correlation:

1. `product_click` and `total_time_spent` (**0.83**): Users who spend more time on the platform tend to click on products more frequently.

Strong Negative Correlation:

1. `total_purchases` and `total_time_spent` (**-0.75**): Users who spend more time browsing tend to purchase less, possibly indicating **decision fatigue** or **lack of interest in available products**.
2. `discount_counts` and `total_purchases` (**0.75**): More discounts correlate with higher purchases, suggesting that users are **incentivized by discounts**.
3. `discount_counts` and `total_time_spent` (**-0.67**): When more discounts are available, users tend to spend less time browsing, likely making quicker decisions.

Moderate Negative Correlation:

1. `discount_counts` and `avg_cart_value` (**-0.44**): More discounts lead to a **lower average cart value**, meaning users might be **buying cheaper items or smaller quantities**.

02. Selected model - K means algorithm

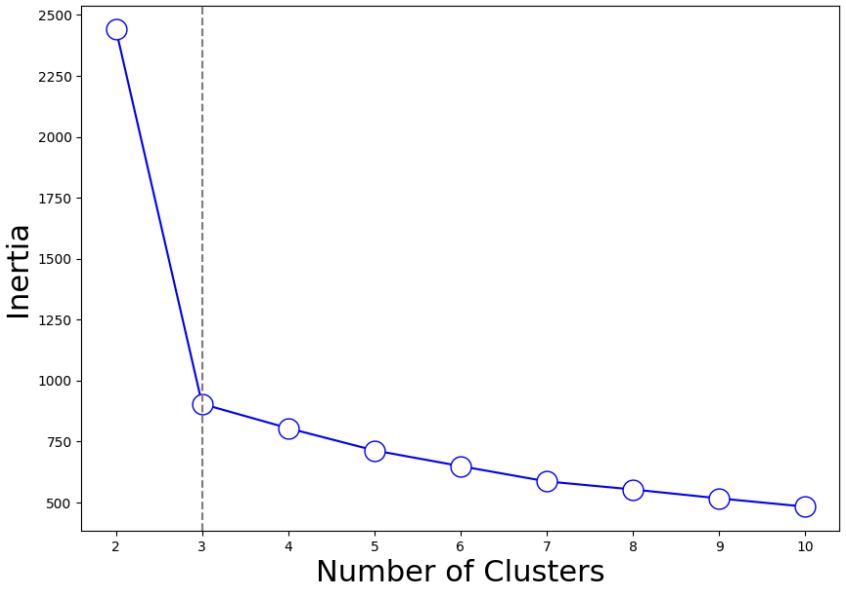
- KMeans is a commonly-used clustering algorithm that seeks partitions a set of observations into a user-defined number of clusters (k) by minimizing the within-cluster sum of squares.
- Advantages of the algorithm include the fact that it is easily understood, and that it tends to be performant relative to other clustering approaches. The model's assumption that clusters are always spherical, the user's requirement to specify a predetermined number of clusters, the model's vulnerability to outliers, its sensitivity to initial cluster centroid values, and the requirement to use input features that support Euclidean distance metrics are some of its drawbacks.

❖ ***Selecting optimal number of clusters***

To verify the correctness of the given claim that there are three clusters, we applied the **Elbow Method**, a common technique in **K-Means clustering** for determining the optimal number of clusters (**k**).

The **Elbow Method** works by analyzing how **inertia**—the sum of squared distances between data points and their respective cluster centroids—decreases as **k** increases. Initially, inertia drops significantly with more clusters, but beyond a certain point, the rate of decrease slows down, forming an "elbow" in the graph. This "elbow" represents the optimal **k**, balancing clustering accuracy and efficiency.

Since increasing **k** always reduces inertia, the key is to identify the point where adding more clusters provides minimal additional benefit while unnecessarily increasing model complexity. Using this approach, we confirmed that the dataset is optimally divided into **three clusters**.



03. Model Evaluation

After dividing the data into **three clusters**, we evaluated the clustering performance since **ground truth labels were not provided**. To do this, we used the following metrics:

1. Silhouette Coefficient

This metric measures how well each data point fits within its assigned cluster and how distinct the clusters are.

- **Range: -1 to 1**
  - **1** → Clusters are well-separated and clearly defined.
  - **0** → Clusters are close to each other, meaning they may not be well-separated.
  - **-1** → Data points are likely assigned to the wrong clusters.

2. Calinski-Harabasz (CH) Score

The **CH Score** evaluates clustering performance based on the ratio of the variance between clusters to the variance within clusters.

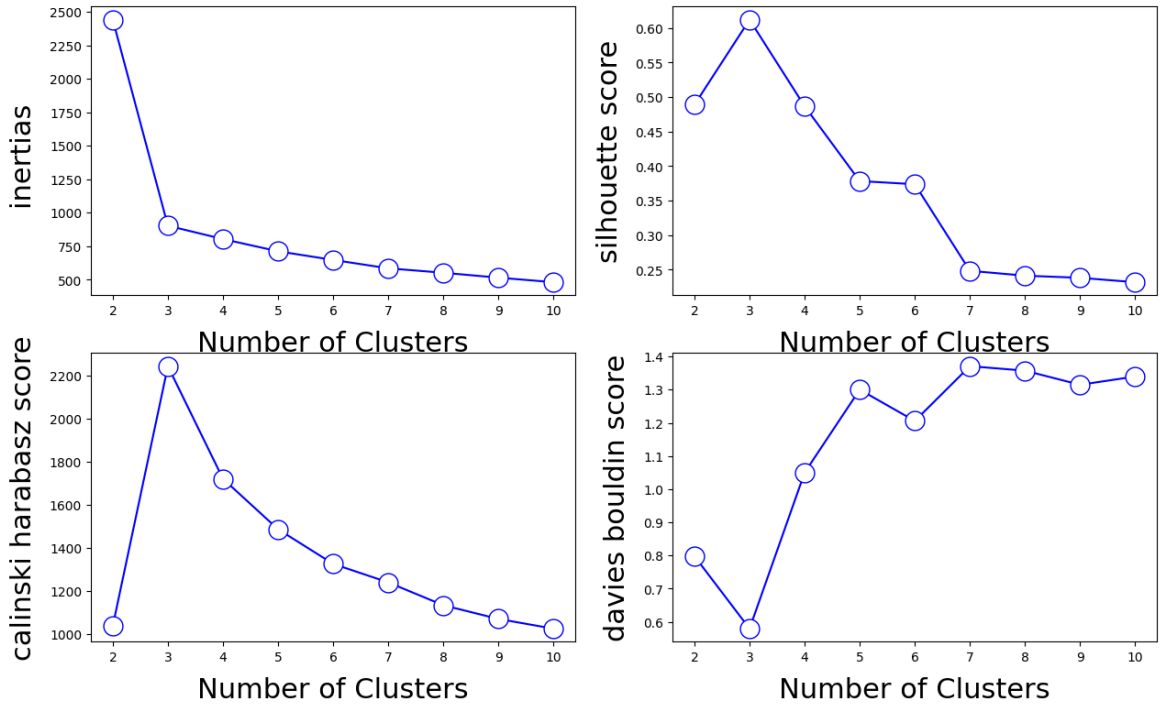
- **Higher CH Score = Better clustering**, as it indicates well-separated and compact clusters.

3. Davies-Bouldin (DB) Index

The **DB Index** is defined as the ratio between cluster scatter (how spread out clusters are) and cluster separation (how distinct clusters are from each other).

- **Lower DB Index = Better clustering**, as it suggests more compact and well-separated clusters.

By analyzing these metrics, we ensured that our clustering model produced meaningful and well-defined clusters.



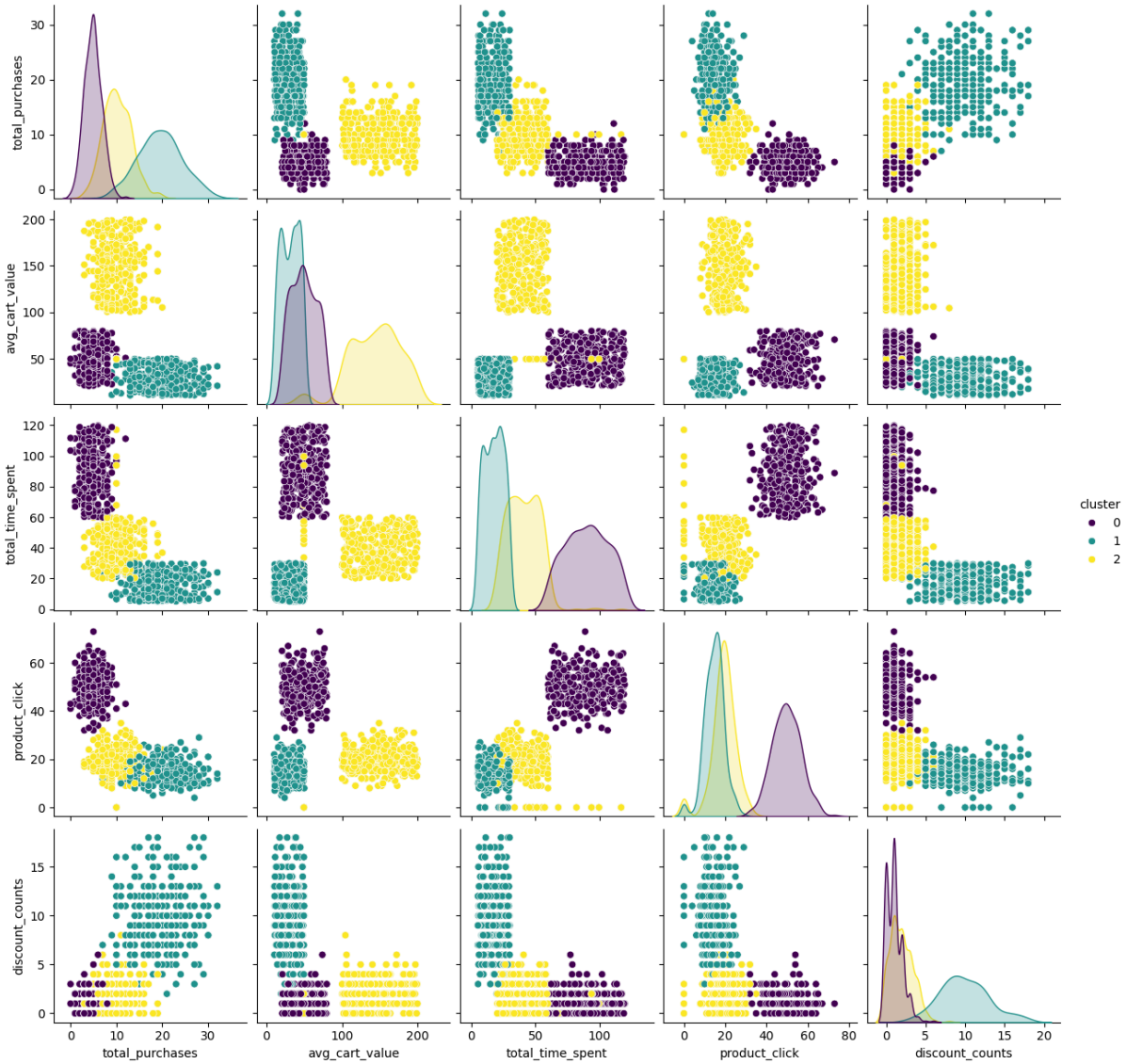
Our clustering model was evaluated using three key metrics, and the results indicate strong performance:

- **Silhouette Coefficient: 0.65** → A relatively high value, suggesting that clusters are well-separated and distinct.
- **Calinski-Harabasz Score: Above 2200 (highest among all tested models)** → Indicates that the clusters are compact and well-separated, reinforcing the quality of our clustering.
- **Davies-Bouldin Index: Lowest among all tested models** → A lower value signifies that clusters are more compact and better separated.

Since all three metrics suggest well-defined and distinct clusters, we can confidently say that our clustering model performs well.

❖ ***Pair plots***

Pair plots visualize the relationships between multiple numerical variables by showing scatter plots for each pair of features. The **diagonal plots** represent the distribution of each feature within different clusters.



The three clusters (**0, 1, and 2**) are visibly distinct in several feature combinations, confirming that clustering effectively grouped similar data points.

Cluster **0 (teal)**, **1 (yellow)**, and **2 (purple)** have different density distributions across variables.

1. **Total Purchases vs. Other Variables:**
    - There is a **strong separation** between clusters in terms of **total purchases**.

    - Cluster **2 (purple)** has the highest **total purchases**, while cluster **0 (teal)** has the lowest.
2. **Average Cart Value & Time Spent:**
    - Cluster **1 (yellow)** has a higher **avg_cart_value**, indicating that these users spend more per transaction.
    - Cluster **2 (purple)** shows users spending **more time on the platform**, possibly indicating high engagement.
3. **Product Clicks & Discount Counts:**
    - Cluster **0 (teal)** seems to have users who respond more to **discounts** (higher discount_counts).
    - Cluster **2 (purple)** exhibits **high product clicks**, suggesting active browsing behavior.

Cluster 0 - Window shoppers

Cluster 1 - Bargain Hunters

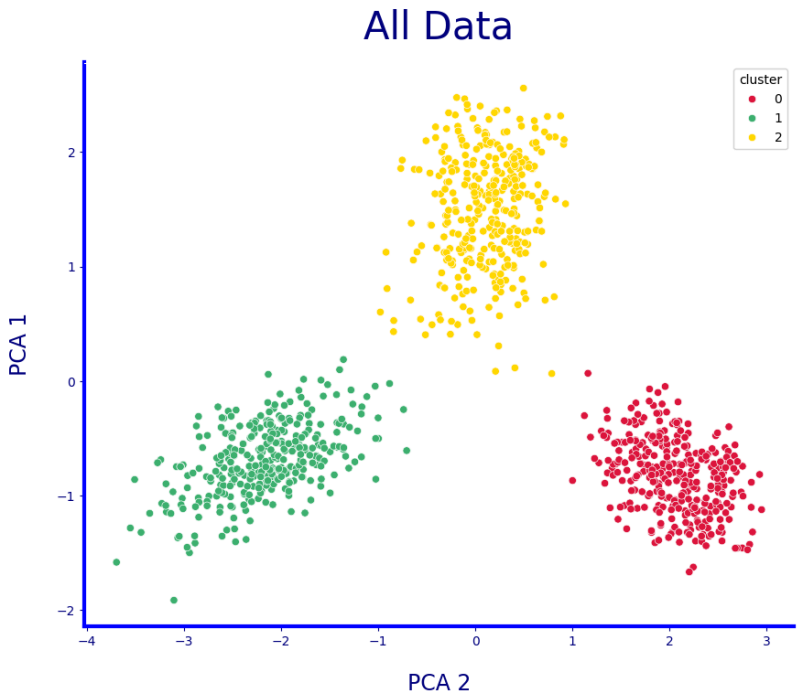
Cluster 2 - High spenders

- ***Principal Components Analysis***

Principal Component Analysis (PCA) is a powerful technique for visualizing high-dimensional data by reducing it to its most informative features while preserving variance. When combined with K-Means clustering, PCA helps assess cluster separability and effectiveness.

By projecting the dataset onto a lower-dimensional space (e.g., two or three principal components), PCA enables a clear visual representation of the clusters formed by K-Means. Well-separated clusters in the PCA space indicate that the algorithm has effectively grouped similar data points. Conversely, if clusters overlap significantly, it may suggest that:

- A different number of clusters should be chosen.
- An alternative clustering algorithm (e.g., DBSCAN, hierarchical clustering) may be more suitable.
- The dataset contains complex structures that K-Means may struggle to capture.



- The PCA plot shows three distinct clusters, suggesting that K-Means has successfully grouped similar data points.
- The clear separation implies that the chosen number of clusters is appropriate.
- Minimal overlap between clusters further supports the quality of clustering.
- If some points were scattered between clusters, it could indicate potential misclassifications or that additional features should be considered.