Customer Segmentation

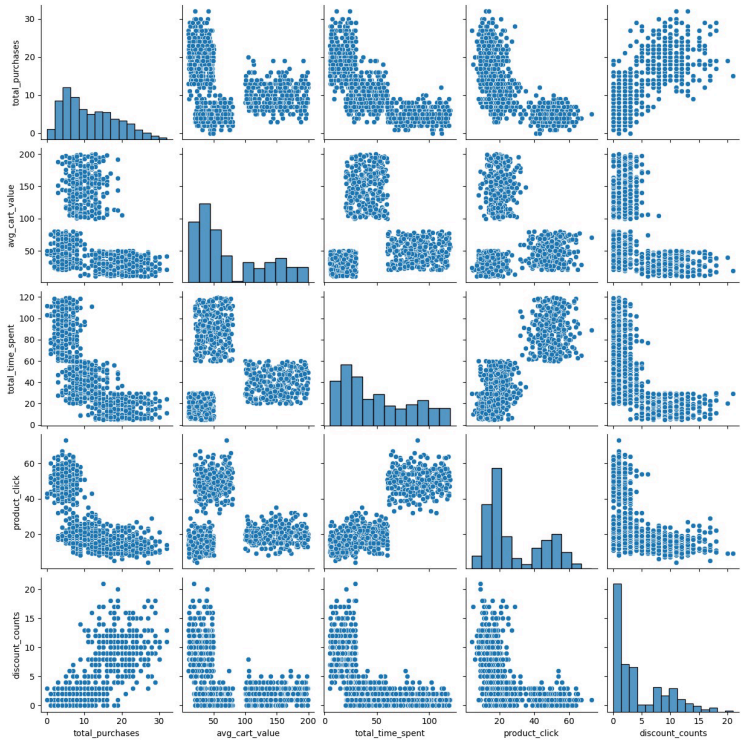
Data Cleaning and Exploratory Data Analysis (EDA)

To begin, we cleaned the dataset by removing missing values and performed Exploratory Data Analysis (EDA) using these methods:

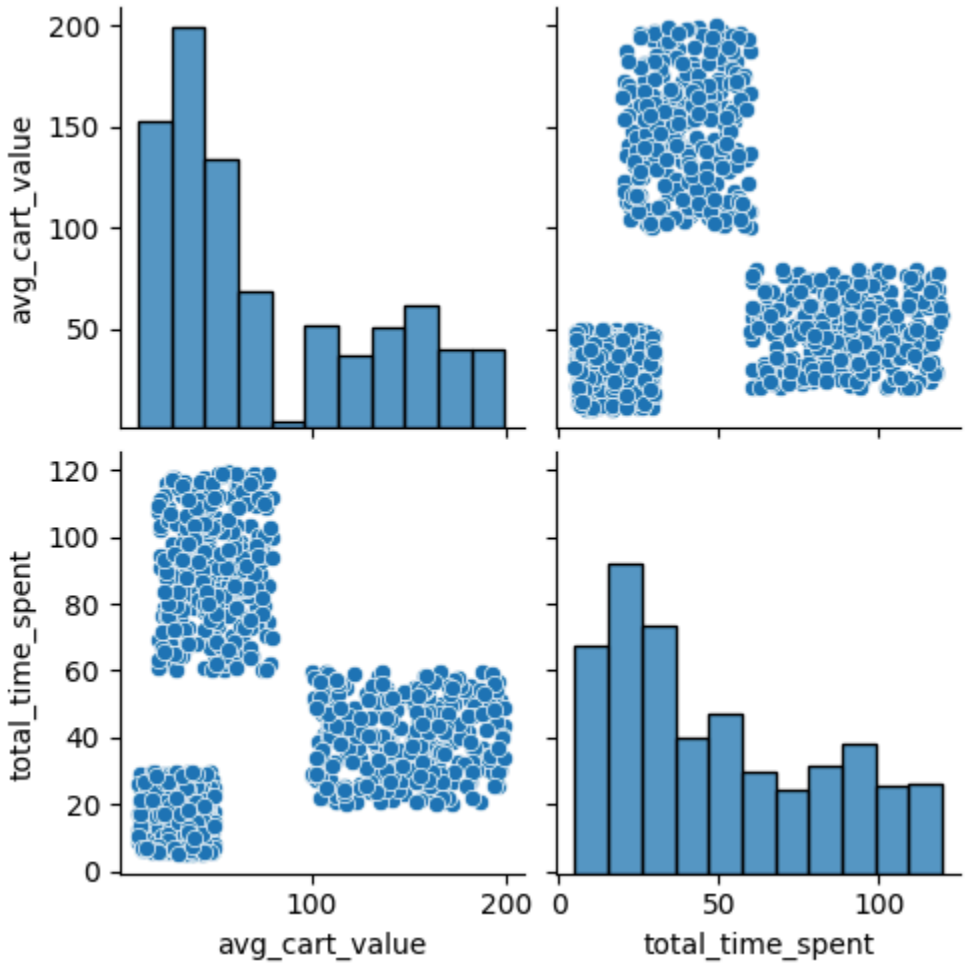
1. **Feature Selection**: We removed irrelevant features (`total_purchases`, `discount_counts`, `product_click`) that did not contribute significantly to clustering.
2. **Correlation Analysis**: A correlation matrix heatmap helped us visualize relationships between features.
3. **Distribution Analysis**: Histograms (with KDE enabled) showed the distribution of each feature.
4. **Outlier Detection**: Box plots helped identify outliers in the dataset.
5. **Feature Interaction Analysis**: A pairplot was used to explore relationships between different features.
6. **Data Scaling**: We used `StandardScaler` to normalize the data for better clustering performance.

Feature selection

When visualizing the data, we observed that using all features resulted in unclear clusters.



However, when we simplified our dataset by selecting only `avg_cart_value` and `total_time_spent`, the three customer segments became distinct:



And these three clusters matched the cluster descriptions given to us.

1. **Bargain Hunters**
    - High `total_purchases` (frequent purchases)
    - Low `avg_cart_value` (cheaper items)
    - Moderate `total_time_spent` (some browsing but focused on purchasing)
    - Moderate `product_click` (reasonable number of products viewed)
    - High `discount_count` (frequent use of discount codes)
    - **Behavior**: These customers are deal-seekers who frequently purchase low-value items and rely on discounts.

2. **High Spenders**
    - Moderate `total_purchases` (fewer but high-value purchases)
    - High `avg_cart_value` (expensive items)
    - Moderate `time_spent` (browsing but focused on high-value items)
    - Moderate `product_click` (reasonable number of products viewed)
    - Low `discount_usage` (rare use of discount codes)
    - **Behavior**: Premium buyers who focus on high-value purchases and are less influenced by discounts.
3. **Window Shoppers**
    - Low `total_purchases` (very few purchases)
    - Moderate `avg_cart_value` (varying item prices)
    - High `time_spent` (spend a lot of time browsing)
    - High `product_click` (view a large number of products)
    - Low `discount_usage` (rare use of discount codes)
    - **Behavior**: These customers browse extensively but make very few purchases.

Models Used

To identify the customer segments, we used two clustering algorithms: **KMeans Clustering** and **Gaussian Mixture Model (GMM)**.

Comparison of Clustering Models

We evaluated both methods using silhouette scores, Calinski-Harabasz Index, and Davies-Bouldin Index:

Python

```python
# Compare KMeans and GMM
def compare_clustering(scaled_data):
    # KMeans clustering
    kmeans = KMeans(n_clusters=3, random_state=42)
    kmeans_labels = kmeans.fit_predict(scaled_data)

    # Gaussian Mixture Model
    gmm = GaussianMixture(n_components=3, random_state=42)
    gmm_labels = gmm.fit_predict(scaled_data)

    # Get scores for both models
    silhouette_kmeans = silhouette_score(scaled_data, kmeans_labels)
```

```
silhouette_gmm = silhouette_score(scaled_data, gmm_labels)

ch_kmeans = calinski_harabasz_score(scaled_data, kmeans_labels)
ch_gmm = calinski_harabasz_score(scaled_data, gmm_labels)

dbi_kmeans = davies_bouldin_score(scaled_data, kmeans_labels)
dbi_gmm = davies_bouldin_score(scaled_data, gmm_labels)
```

Both methods provided similar results since the clusters were clearly distinguishable due to our feature simplification. Therefore, we decided to proceed with **KMeans Clustering** for simplicity and efficiency.

Cluster Creation Using KMeans

Python
```
# Create clusters using KMeans
kmeans, clusters = make_clusters(data, scaled_data,
num_clusters=3)
```

We used **Principal Component Analysis (PCA)** to visualize the clustered data in a two-dimensional space.

Clustering Evaluation

Python
```
# Evaluate clustering performance
evaluate_clusters(scaled_data, clusters, kmeans)
```

Metrics such as **Silhouette Score**, **Davies-Bouldin Index**, and **RMSE** confirmed that the clusters were well-separated and meaningful.

Final Conclusion

By simplifying our dataset and selecting only the most relevant features, we achieved clear cluster separations. The **KMeans Clustering** method proved to be the most efficient and interpretable approach for identifying **Bargain Hunters, High Spenders, and Window Shoppers** in our dataset.