Machine Learning Project Report (Customer Segmentation)

Group members: Tommaso Malenchini, Ferdinando Giordano.

Introduction: The project began by examining a database of orders placed by customers in Brazil. The database contained various order details, including the total cost and information about the customers and sellers. our goal was to segment the customers using three clustering methods (K-means, Hierarchical clustering, and DBSCAN). These methods were chosen because they are effective at grouping similar data points and have been widely used in similar customer segmentation tasks.

EDA and Preprocessing the dataset: We started by reading the dataset into our notebook and importing all the libraries needed such as (pandas, NumPy, seaborn, matplotlib and others).

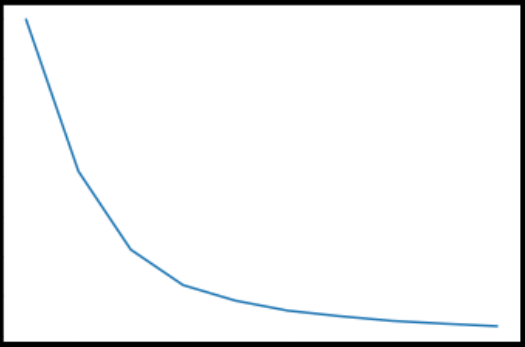
In the first part of the project we performed an EDA (exploratory data analysis) In order to visualize the dataset and get the relevant information needed to decide which features are useful and which ones are not, this process was crucial in as we did get the touch with the dataset itself and be more confident when preprocessing it.

In the preprocessing phase the first thing we did is to drop the duplicates where they weren't needed, we also checked for categorical variables, as preprocessing them would be beneficial for our analysis. Going further we noticed that Some columns in the dataset were not relevant for our project, therefore we dropped them, this operation results in a smaller dataset which is more comfortable in terms of computational time and storage.
Successively we proceeded with the One-Hot Encoding process. When performing clustering algorithms such as K-means and Hierarchical clustering converting categorical variables into numerical form is useful since these algorithms require numerical input data. Additionally, one hot encoding can help prevent bias in the clustering process by ensuring that the categorical variables are treated equally. Without one hot encoding, the clustering algorithm may give more weight to certain categories over others, which could affect the resulting clusters. Finally, we reduce other columns by using the 'get_dummies' function.

RFM Analysis: The customer segmentation we want to perform is based on the RFM (Recency, Frequency and Monetary value) Strategy. This technique basically identifies the value of a customer to a certain business, so that a business can take track of their most valuable customers and target their marketing efforts towards them. This strategy is composed by three factors:
-Recency: which refers to the last time a customer made a purchase;
-Frequency: refers to the number of purchases a customer has made;
-Monetary: This refers to the amount of money a customer has spent;
Intuitively these three factors will give a precise value to a customer;
Using our dataset we calculated both the three factors thanks to the following columns: (order_purchase_timestamp, customer_id, Latest Purchase and payment_value).

K-Means: The first Clustering method we used was The K-means algorithm, K-means is one of the most used algorithms in clustering, The k-means algorithm is used for grouping data into k groups based on how similar the data points are. Each data point is then assigned to the nearest centroid after initializing k "centroids" in the data. The algorithm then reassigns each data point to the closest centroid after iteratively adjusting the positions of the centroids to be the means of the points in their respective clusters. Repeating this procedure until the centroids stop moving is necessary for the clusters to be deemed complete. The centroids and the data points given to them produce the final clusters.

In order to determine the optimal number of clusters to minimize the WCSS(within-clusters-sum-of-squares),we begin by applying the Elbow method. By applying the Elbow method, we obtain this graph:

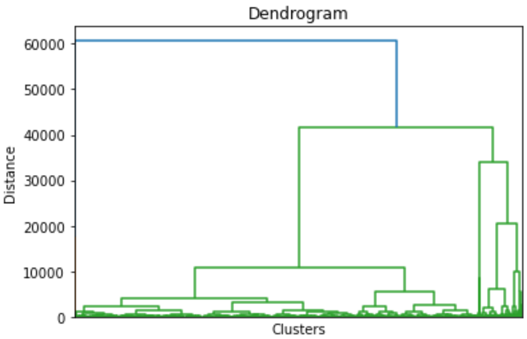


What this graph is telling us is that the optimal number of clusters is either 4 or 5. As we want to make sure about which is the optimal number of clusters we perform the Grid Search method, using the Grid Search method helped us since otherwise we had to find the optimal number of clusters manually.

We got that the best parameters are 'full' as algorithm and 5 as n_clusters setting to 3 the iteration to find the centroids.

Finally, we wanted to 'evaluate' the performance of the k-means, and we did it by finding the silhouette score of our k-means, the silhouette score is a metric that measures how similar the points within a cluster are to each other, and how dissimilar the points in different clusters are. The silhouette score for k-means is: 0.7496956778380279.

Hierarchical Clustering: The second clustering method we decided to apply was the Hierarchical clustering, a benefit of this method which differences it with k-means is that when performing it there is no need to specify the number of clusters . Clusters are created through hierarchical clustering in the form of a hierarchy tree (called a Dendrogram) which is a tree-like structure with the full data as the root node, a subset of similar data is formed in this manner, and branches are created from the root node to form various clusters. This method is much more interpretable than k-means since we can immediately see which is the best number of clusters by looking at the longestbranch of the tree.



Given this dendrogram ,the silhouette score for the Hierarchical clustering is :
0.7904401494191384

DBSCAN Clustering : The third and last clustering method we applied is the DBSCAN clustering, this method is a density-based algorithm, DBSCAN algorithm is a good choice for clustering data when the data points in the dataset have variable densities, and when the number of clusters is not known in advance. It is also relatively robust to noise and outliers in the data. We have seen that this algorithm has some advantages when performing customer segmentations but in our case, we got a silhouette score of : -0.61958 which is not really desirable.

Comparison Between The Methods: Having the silhouette score as comparison method, the hierarchical and k-means clustering seems to be a lot better than the DBSCAN since its silhouette score is close to -1, which means that the data points in the cluster are very poorly matched to the cluster; However, the silhouette score can't define exactly which algorithm is more suitable for a task, in order to find the best algorithm for a specific task many other variables are needed, such as the domain of the data, the time complexity of the algorithm and many other factors.
In conclusion, both the k-means and hierarchical seem to suit our task.