**Abstract**

In this report we choose to analyze a set of data from the “Mall Customers” dataset. We wish to study our customers and cluster them in groups (segmentation) to be then able to target the groups of customers better by tailoring the marketing campaigns. After exploring the data and providing macro-trends we apply 3 main unsupervised-learning algorithms to the dataset: k-means, hierarchical clustering and DBScan. We then compare the results of the three algorithms and try to understand the reasons behind (any) difference in results.

**Goal:**

Our primary goal is to segment the customers, discovering groups within the unlabeled dataset. After selecting the “best” algorithm (where best, is intended as the “best” with regards to the dataset, and the application domain), we wish to understand the reasons behind the lesser performance of the remaining two algorithms.

**Description of Data Set**

This dataset is composed of 200 observations and 4 components (Gender, Age, Annual Income and Spending Score). Looking at the columns, gender is categorical while the remaining ones are numerical. We also notice that there is no missing data, and so proceed to the Exploratory Data Analysis.

Before applying any clustering method, we perform the Hopkins test to see whether the data set contains any meaningful (non-random) clusters. Indeed, the k-means algorithm or hierarchical cluster can return clusters even if data does not contain any cluster )e.g. on uniformly distributed data).

We perform the Hopkins test to see whether clustering is relevant for the data. As the Hopkins statistics is 0.85 (thus above 0.5), we can reject the null hypothesis and conclude that the dataset can be split into groups.

**Key Points**

***Key Points - EDA***

Looking at Figure 1 (next page) we notice that

* O1: there are more female customers than men, and women’s (spending) activity tends to be most in the 25-35 range, with a second peak at 45 while men have their first peak in the 25-35 range and tending to spend more than female in the 60-70 range.
* O2: there is a slightly negative correlation between Spending Score and Age , with older people spending less (irrespective of gender). The correlation between the other variables seem to be negligible There is one outlier in the male group with an annual income of about 140k$.
* O3: Men and women seem to follow similar trends across the different variables leading us to suggest that there is no statistically significant difference between men and women. This implies that groups formed will most differ by income or scoring group but primarily, not by gender.
* O4: Overall, the points are not densely allocated and the spatial density tends to vary. This could suggest that DBScan will not be very successful.

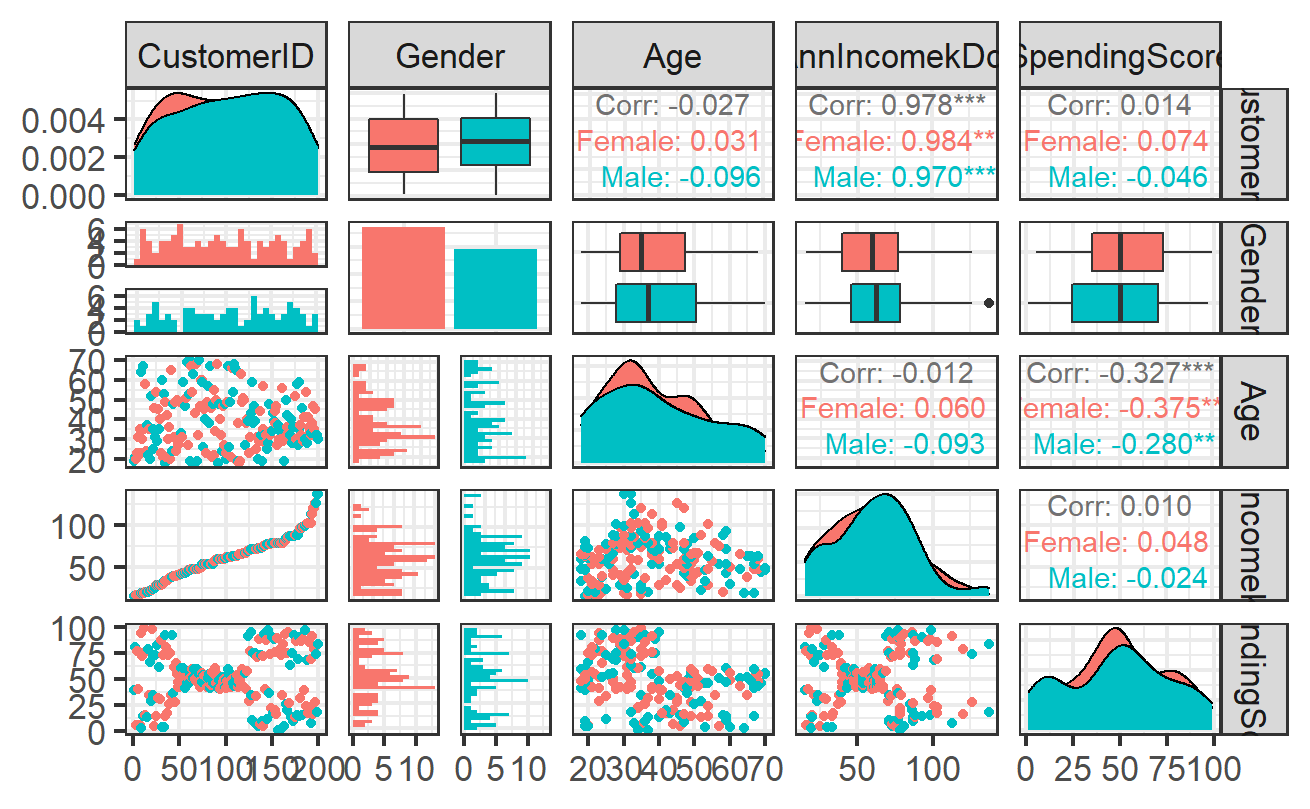


Figure 1: Scatterplot matrix and correlation value

***Key Points and Analysis - Unsupervised Learning***

***K-Means***

Looking at Figure 2, we select 5 as the “elbow point”. Running the algorithm we get 5 clusters of similar size (41, 37, 29, 40 53) respectively. We notice that the mean age-groups range from 24-55 across the groups. The “young” group (the 5th one) tends to have the lowest income and the highest spending score. On average we observe that this group is composed of more women. We find two groups (group 3 and group 1) with a similar average age (around 50), similar spending scores and income (both average). What differentiates them is the gender: the slightly younger group is composed by women, while the older one by men. We then have two further groups which have similar compositions in gender Income: Group 2 and Group 4. Indeed, although both groups are composed by both genders and have high income, the further divided by spending score: Group 2 has a low spending score while Group 4 has a high spending score. On average we notice that women tend to spend more (higher average spending score than men), however average income (primarily) and spending score tend to be the primary factors in determining most customer segments.

We summarize data in the following way (see Table 1):

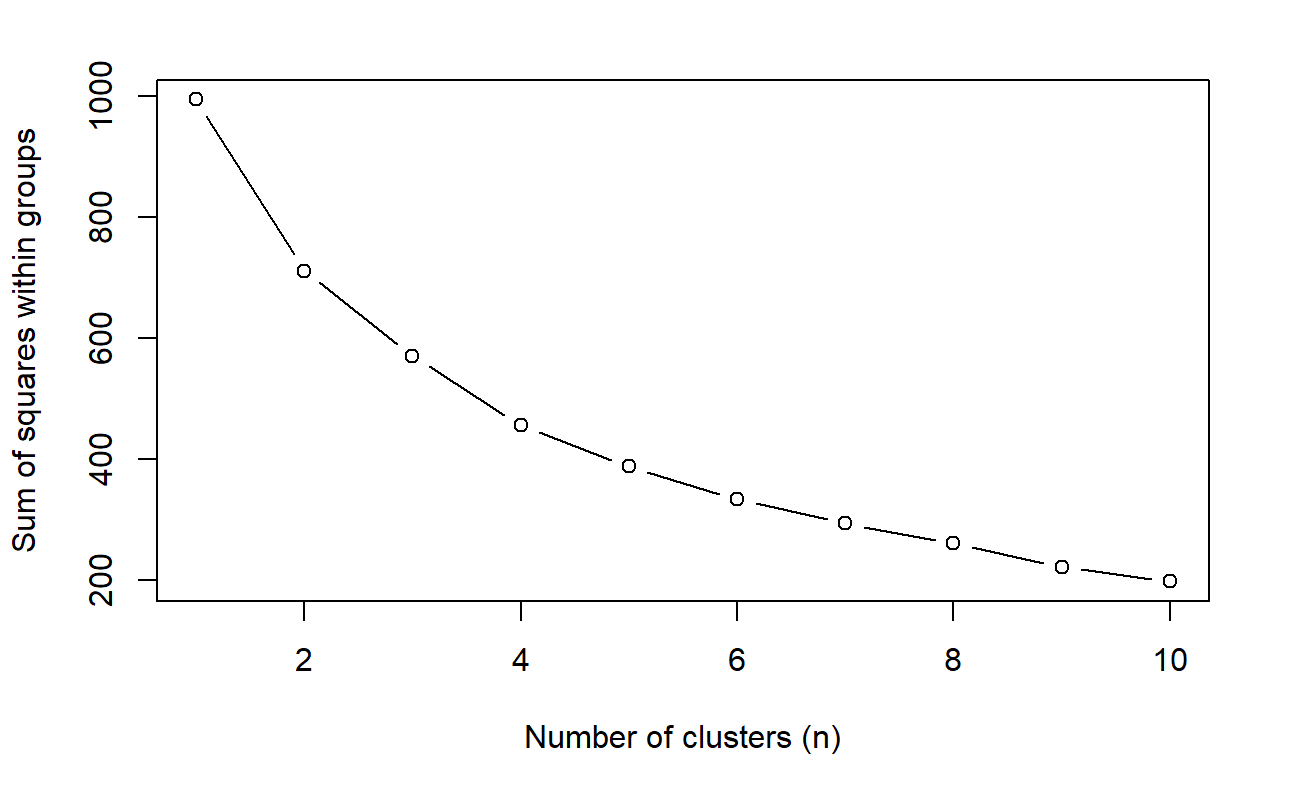


Figure 2: WSS Plot – K-Means

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GROUP | AVERAGE AGE | AVG INCOME | AVG SPENDING SCORE | GENDER COMPOSITION | NUMBER OF POINTS |
| 1K | 50 | Average (49) | Average (42) | Women | 41 |
| 2K | 40 | High (88) | Low (19) | Both genders | 37 |
| 3K | 55 | Average (48) | Average (40) | Men | 29 |
| 4K | 33 | High (86) | High (81) | Both Genders | 40 |
| 5K | 24 | Low (38) | High (61) | Slightly more women | 53 |

Table 1 – Results – K-means

Ein Bild, das Karte enthält.

Automatisch generierte Beschreibung

Figure 3: Cluster plot – K-means

Looking at the silhouette plots, we check how similar observations from clusters are observations from neighboring cluster. We notice that ther are a few “misclassified” values in group 2 and group 3 (negative silhouette score) and the average silhouette score is 0.28.

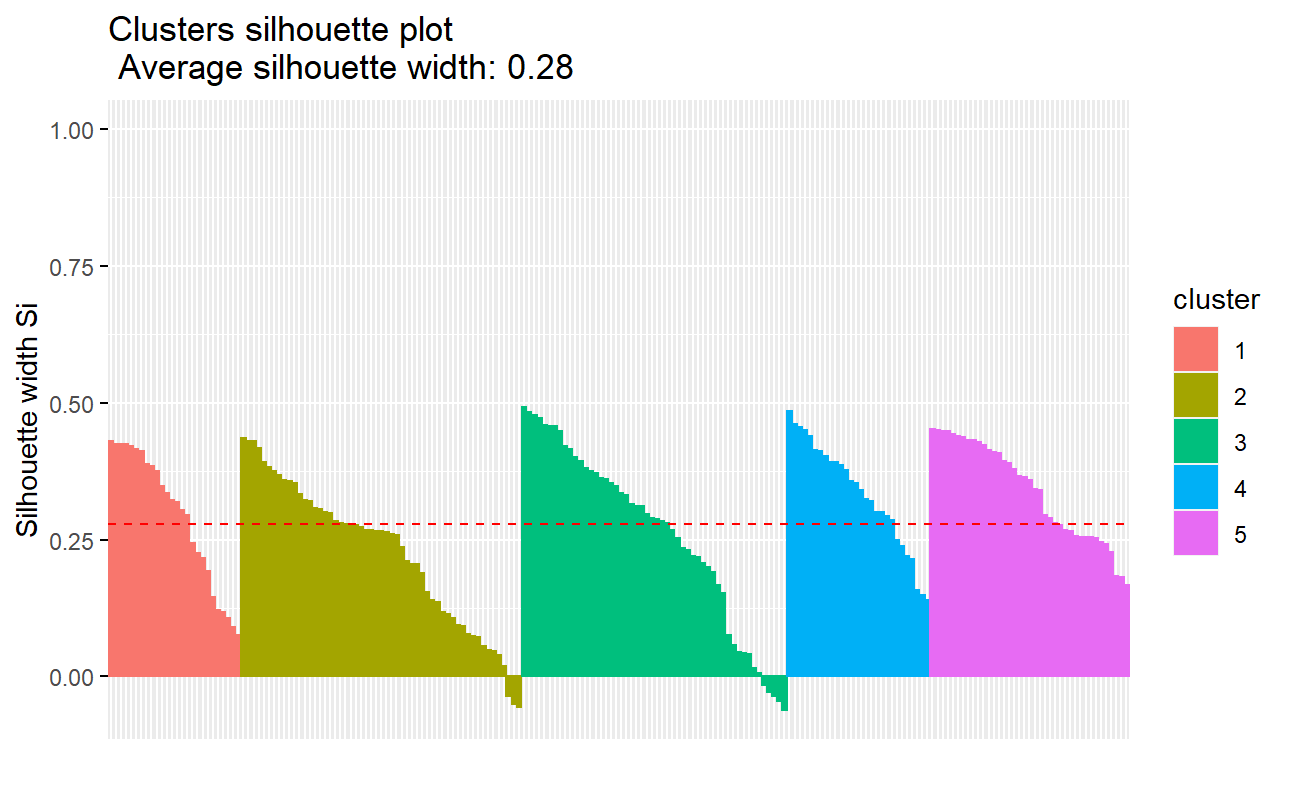


Figure 4: silhouette plot – K-means

***Hierarchical Clustering***

Looking at the dendogram in Figure 5, we select 4 as the number of clusters. Running the algorithm we get 4 clusters. We notice that the mean age-groups range from 33-60 across the groups.The first cluster and third cluster contain more points when compared to the k-means, while the third and fourth look similar in size to the other groups found in the previous analysis. We notice that we still have 2 clusters which have similar compositions in gender Income (Group 4H and Group 3H), have high income and are further divided by spending score: Group 4H has a low spending score while Group 3H has a high spending score. Looking at the two other groups both have average income and average spending score. The two groups differ in that one is composed by more women than men (1H) and vice versa (2H). Moreover, w.r.t. the K-means analysis Group 1H (the one with more women than men) is on average younger (around 31, instead of 50) and Group2H (more men than women) is older (around 60 instead of 55). We summarize data in the following way (see Table 2):

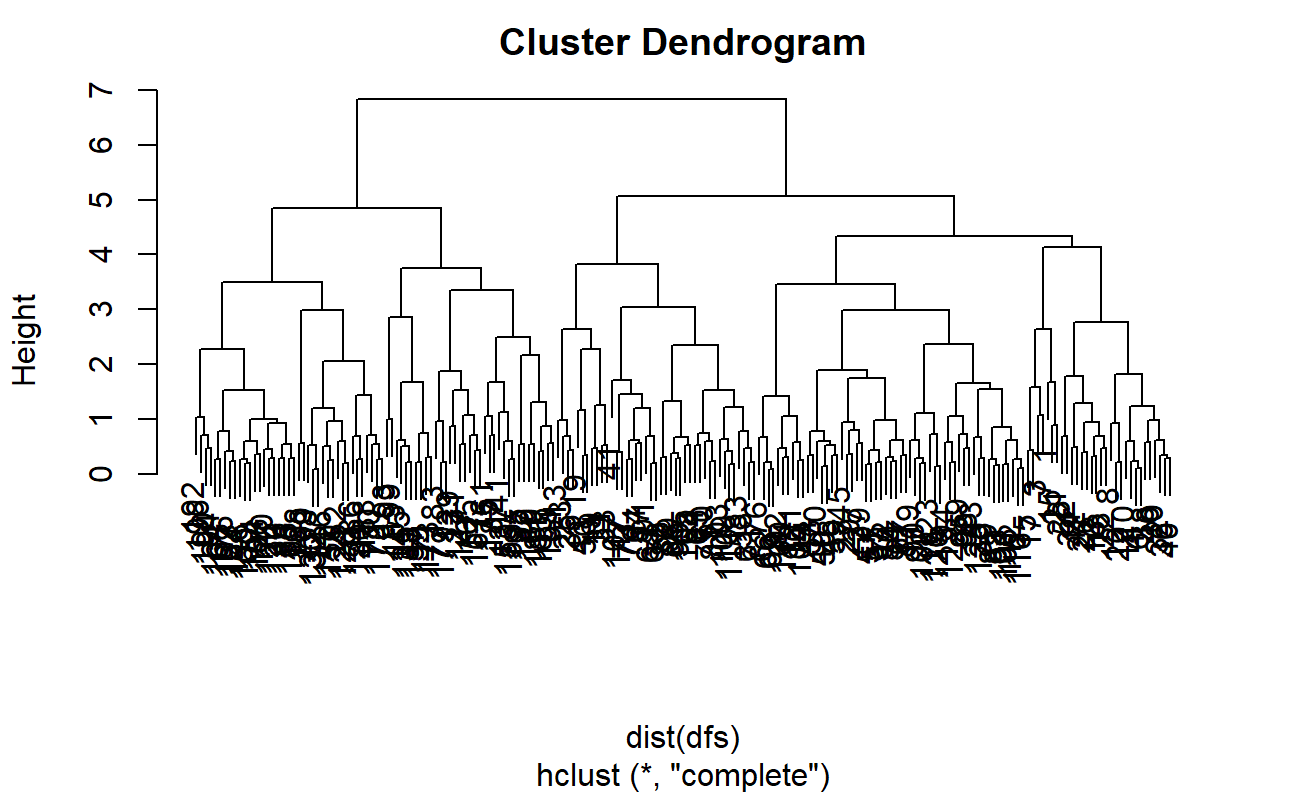


Figure 5: dendogram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GROUP | AVERAGE AGE | AVG INCOME | AVG SPENDING SCORE | GENDER COMPOSITION | NUMBER OF POINTS |
| 1H | 31 | Average(44) | Average(54) | More Women | 85 |
| 4H | 42 | High(88) | Low(18) | Both genders | 35 |
| 2H | 60 | Average(47) | Average(41) | More Men | 39 |
| 3H | 33 | High (86) | High(82) | Both Genders | 41 |

Table 2- Results - Hierarchical

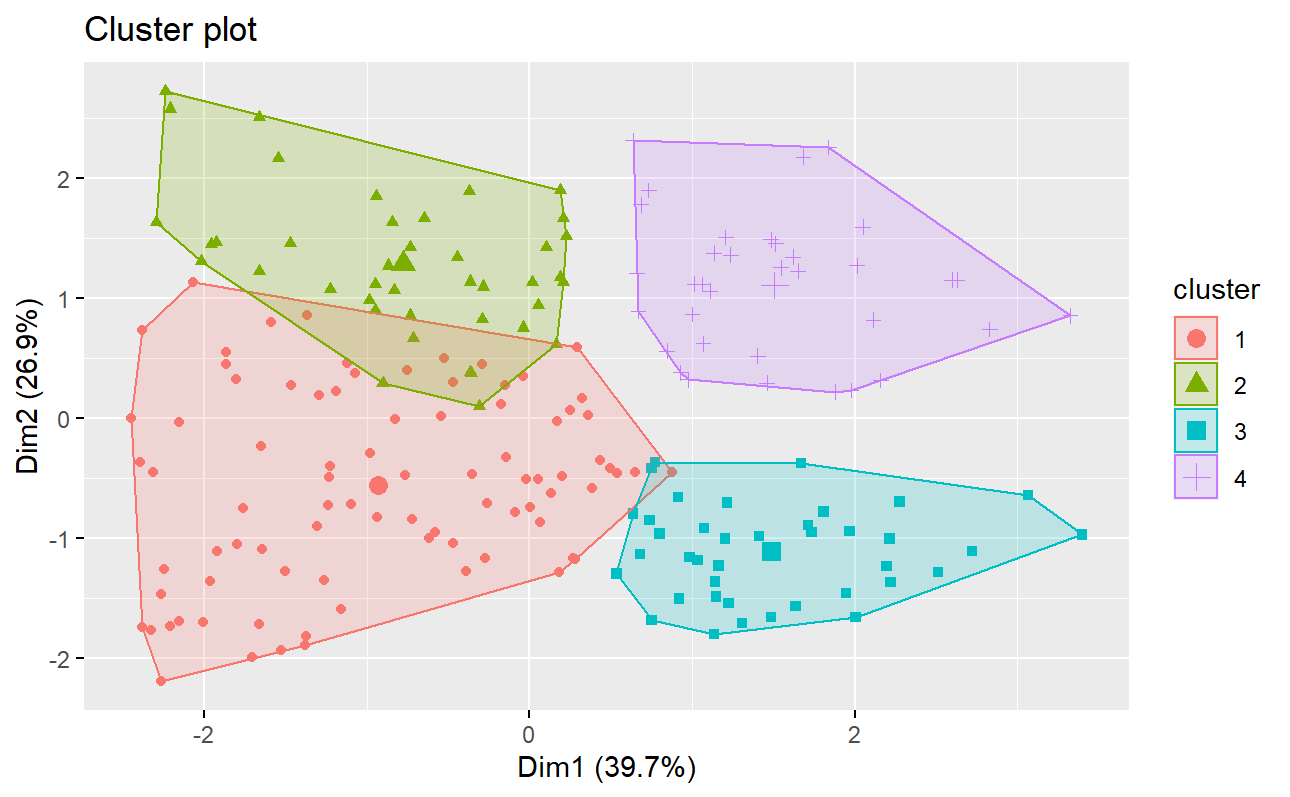


Figure 6: Cluster plot – Hierarchical

Looking at the silhouette plots, we check how similar observations from clusters are observations from neighboring cluster. We notice that there are a few “misclassified” values in group 1H and group 2H (negative silhouette score – we can also see it from Figure 6 where the groups slightly overlap) and the average silhouette score is 0.29 (slightly better than the k-means algorithm).

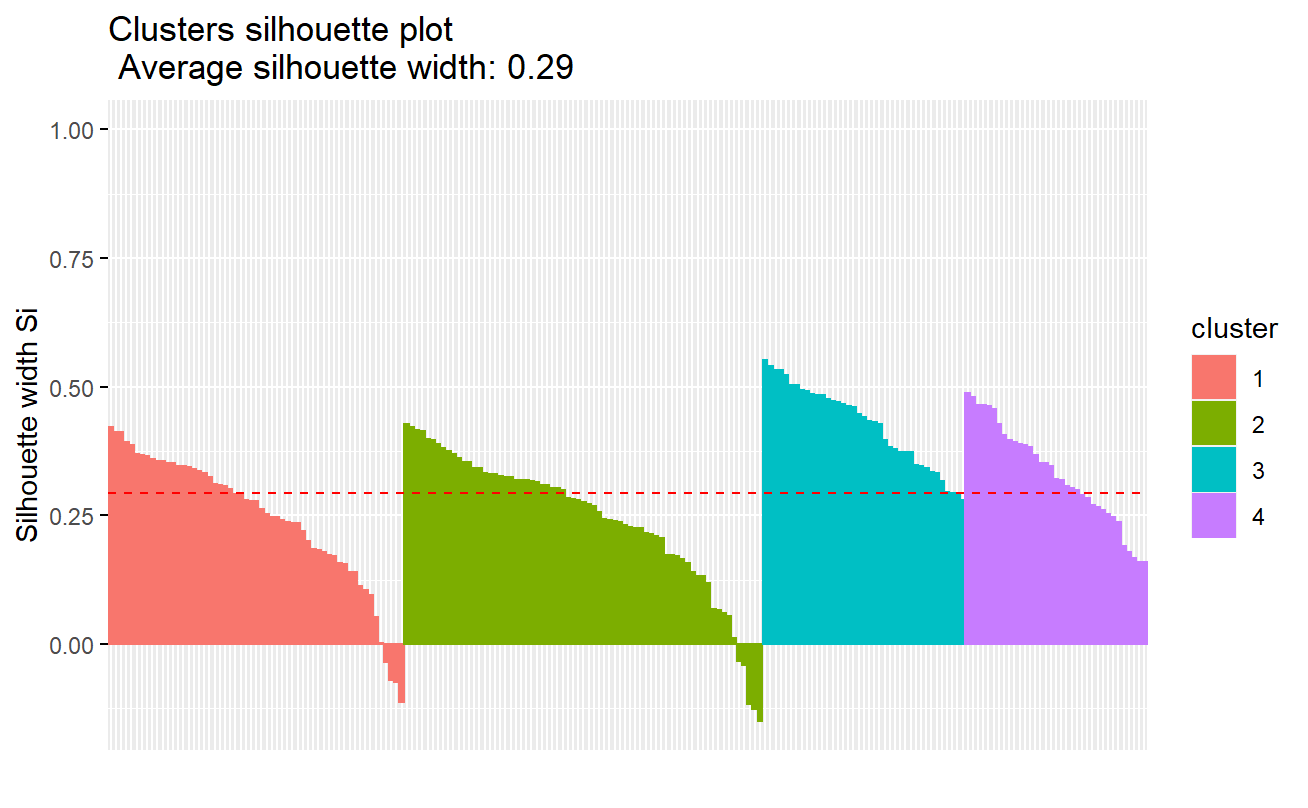


Figure 7: Silhouette plot – Hierarchical

***DBScan Algorithm***

Taking 3 as the number of minimum points, and looking at Figure 8, we take 0.8 as the distance eps. Running the algorithm we get 9 clusters and 16 anomalies (most men, with an average income and high . Looking at Table 3, we notice that cluster size varies significantly. Indeed, one group tends to have around 50% of the data (Group 2DB) and is composed of women with an average income and average spending score (similar to group 1K and 1H). On the other hand the remaining data is split between the 8 remaining groups, with cluster size varying from very small (4) to small (35) and the clusters being divided by gender (6 of which are men, 2 women). These findings already indicate that the algorithm probably does not perform as small clusters are mostly irrelevant seeing as they will not help in identifying a correct target market. Looking at the silhouette scan in Figure 10, we notice a very low average silhouette score (0.14) with several clusters possibly having “misclassified” points (negative silhouette score).

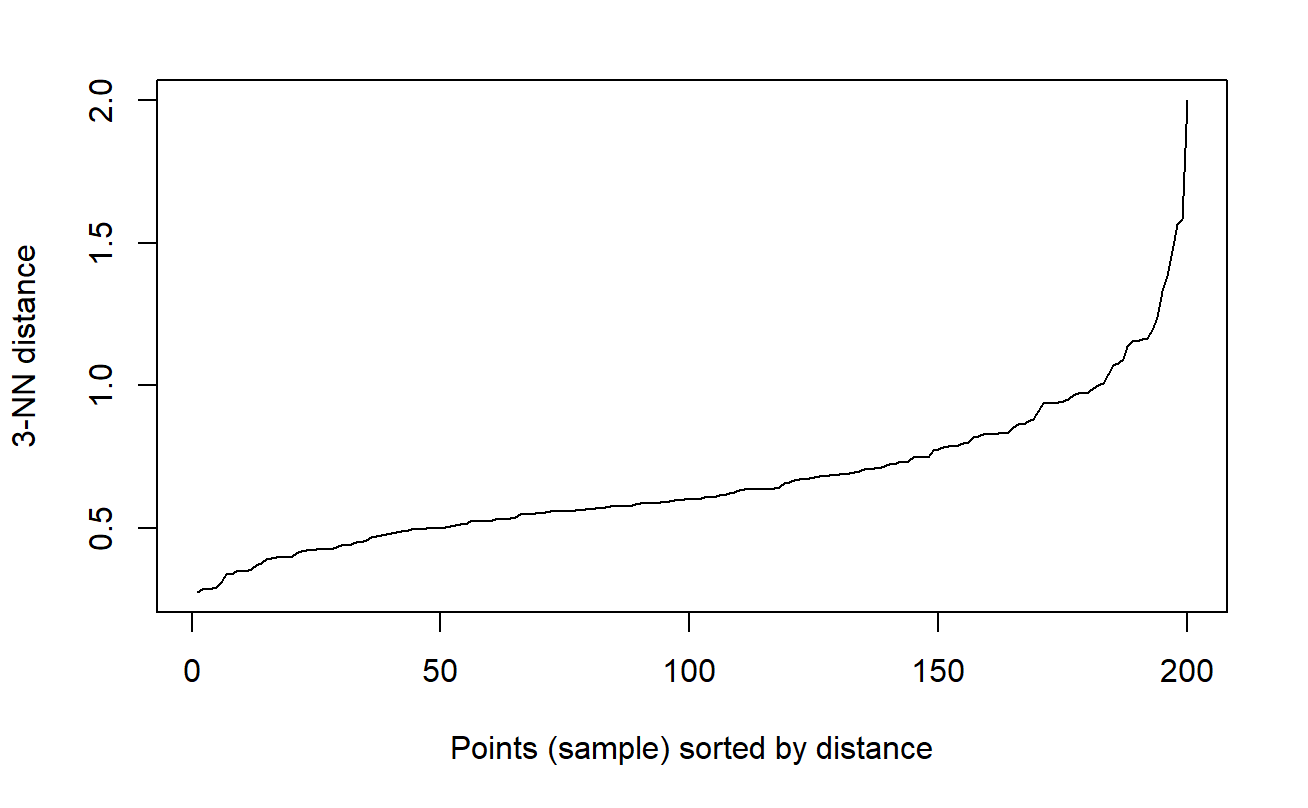


Figure 8: choosing eps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GROUP | AVERAGE AGE | AVG INCOME | AVG SPENDING SCORE | GENDER COMPOSITION | NUMBER OF POINTS |
| oDB/Anomalies | 40 | Average(58) | Average(38) | Slightly More men | 16 |
| 1DB | 26 | Low(27) | High(75) | Men | 10 |
| 2DB | 37 | Average(55) | Average(57) | Women | 92 |
| 3DB | 61 | Low(25) | Low(6) | Men | 4 |
| 4DB | 52 | Average(64) | Low(37) | Men | 35 |
| 5DB | 21 | Average(56) | Average(53) | Men | 10 |
| 6DB | 35 | High(78) | High(88) | Men | 12 |
| 7DB | 21 | High(76) | Low(8) | Men | 4 |
| 8DB | 43 | High(94) | Low(21) | Women | 13 |
| 9DB | 29 | High(91) | High(69) | Women | 4 |

Table 3: results – DBScan

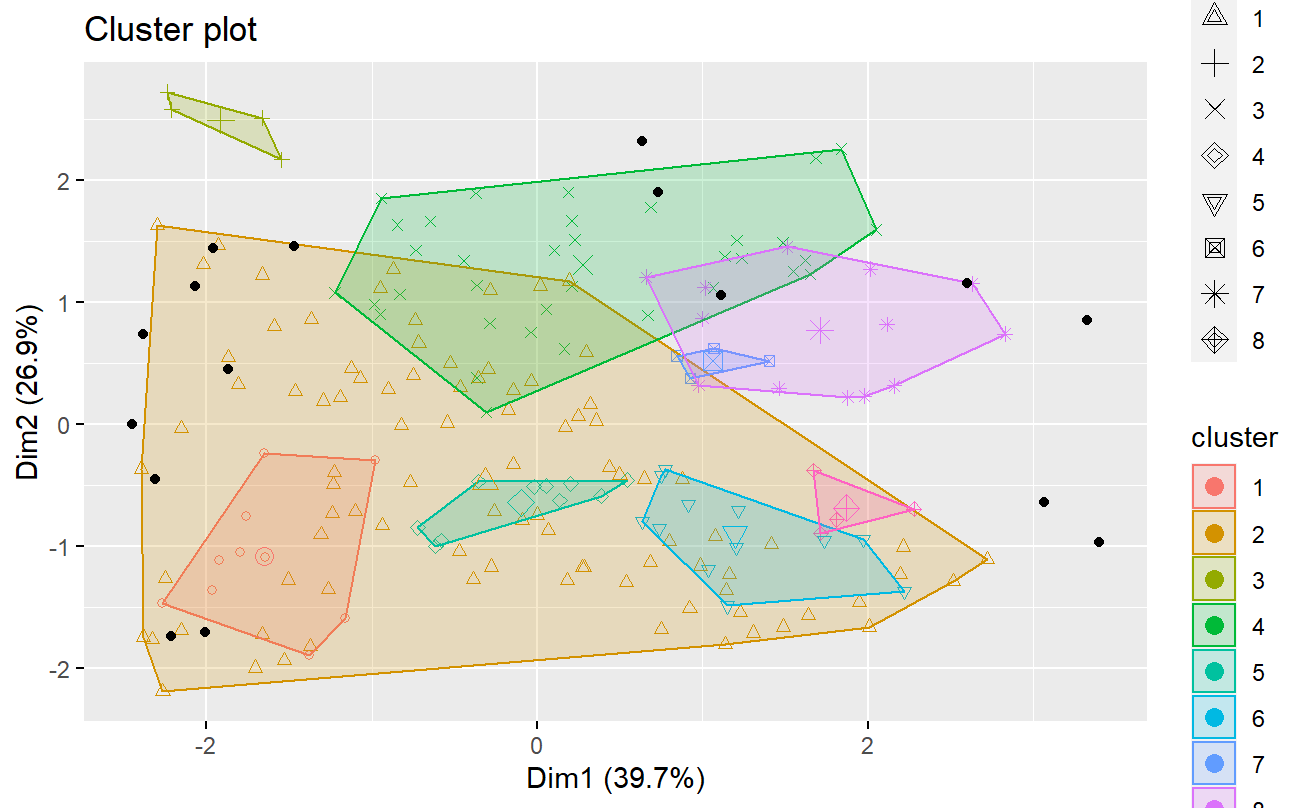


Figure 9: cluster plot - DBScan

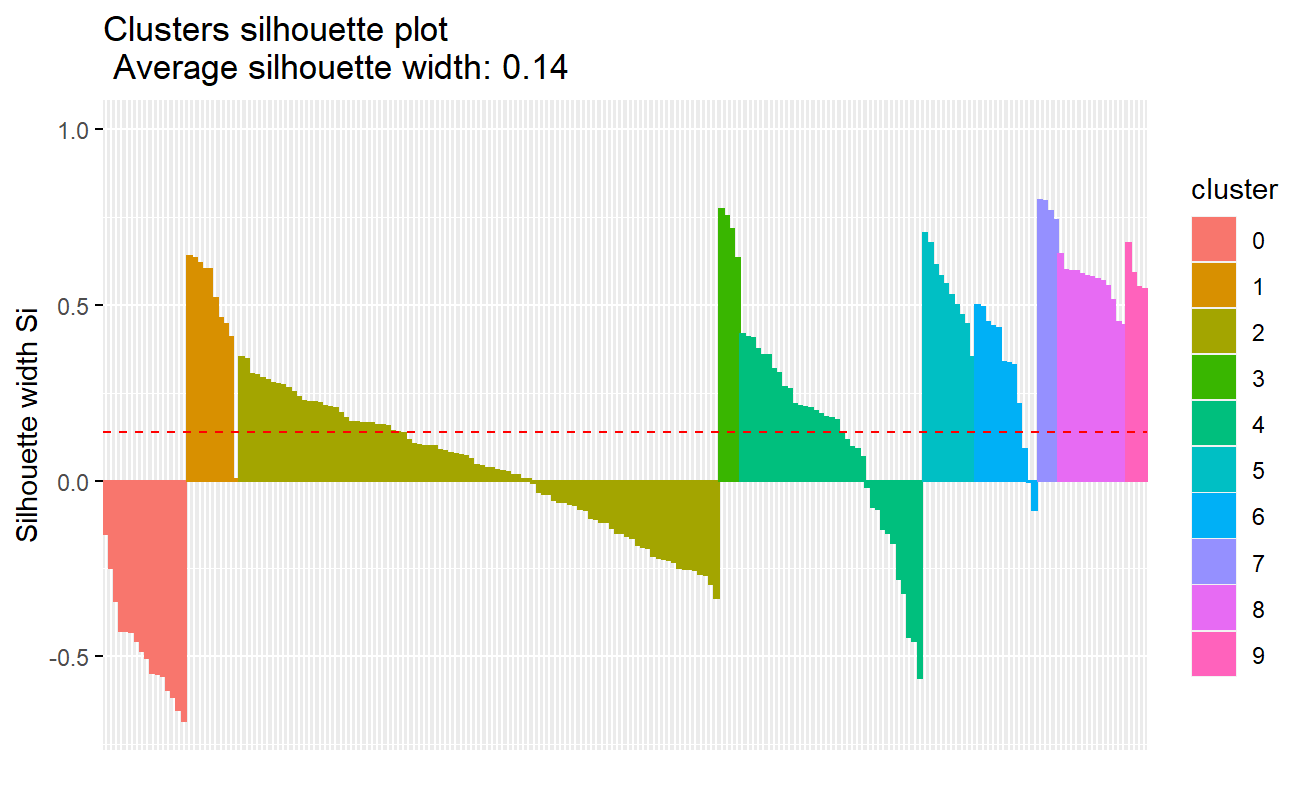


Figure 10: Silhouette plot - DBScan

***K-Means Algorithm- A bit of theory***

The k-means algorithm is a partitional algorithm which consists of the following steps:

* Split samples into initial groups by using seed points. The nearest samples to the seed points create the initial clusters.

Until the centroid points does not: converge do the following actions recursively:

* Calculate the sample distances to the group’s centroid and assign the nearest samples to their cluster.
* Recalculate the updated cluster centroids.

The k-means algorithm seeks to minimize an objective function (inertia) over all the clusters. The inertia (or distortion score) measures the cluster similarity and is composed by two summations - first summing the intracluster distances (distances between a data point and its centroid) up across every data point in the cluster and then by performing the summation of this result across every cluster.A low inertia (or Within-Cluster Sum of Squared error) value means that the data points within the clusters are similar to each other, and is an indication that the clusters are well-formed. The k-means algorihtm is guaranteed to converge, can be run in parallel and has a computational cost in the order of O(kn), where k is the number of clusters and n the number of samples. Moreover this dataset tends to perform well in data-sets with low noise.

On the other hand, the algorithm is however quite sensitive to outliers and N/A data, in which case the DBScan algorithm is often seen to perform well (as it can also identify outliers). In our case, we use the elbow method to select the number of clusters. Moreover, this algorithm is stochastic (non-deterministic) implying that different initial points could lead to different results and could converge to a local minima (and not the global one). This means that several runs are performed (with different initial point) before choosing the run which results in minimum resultant inertia.

We choose to use the Euclidean metric as a distance metric and the Elbow method to select the number of clusters. As the resulting algorithm assumes spherical-like groups only and tends to become inflated in highly multi-dimensional analysis, it performs well when the data-groups are hyperspherical clusters, or well-isolated groups, converging fast and providing well-interpretable clusters.

Going back to the elbow method and looking at an inertia vs number of clusters plot, generally the clusters’ coherency tends to increase (thus inertia decreases) as we increase the number of clusters. On the other hand, the performance of the algorithm tends to decrease (longer run-time) and the amount of information contained in the clusters may decrease (e.g. we get 10 clusters instead of 5 well-defined meaningful clusters, which would allow us to cater to the groups accurately strategy-wise).

***Hierarchical Clustering - A bit of theory***

Looking at hierarchical algorithms, we will use an agglomerative one with complete linkage, recursively merging points into a dataset into larger subsets until a subset contains all of them. Complete linkage clustering tends to make highly compact clusters which is what we are looking for from a marketing-point of view. Moreover, although complete linkage clustering behaves poorly when outliers are present, we have seen from the previous analysis that we are dealing with a “clean dataset”.

The algorithm can be summarized in the following steps:

* A similarity matrix (distance in this case) is calculated

While a cluster which contains all observations does not exist:

* Single points will be merged based on the distance between the most dissimilar samples (the most distant elements in each cluster).
* Update the similarity matrix.

The resulting structure can be visualized on a dendogram. Compared to the K-means algorithm, there is generally no clear winner although Hiearchical Clustering tends has less assumptions on the shape of the dataset, performs better on a global cluster and is a deterministic algorithm. This algorithm is however more computationally intensive, especially so with the size of the dataset. Both perform quite well on datasets with low noise.

***DBScan - A bit of theory***

Density-Based Spatial Clustering of Applications with Noise (DBScan) is a partitional algorithm which does not require the number of clusters to be pre-specified- it is able to find the number of clusters on its own, based on two prescribed parameters (discussed later). This algorithm assumes that natural clusters are composed of densely located points and uses two parameters: distance ( eps ) and a minimum number of points(n) within the distance to find out “dense regions”. The minimum number of points is the fewest number of points required to form a cluster, while the distance eps, is the maximum distance two points can be from one another while still belonging to the same cluster. Points which are within the distance (eps) but find less than n points are treated as border points. Those instead who are outside the distance (eps) are treated as noise or outliers.

When compared to the K-Means scan, the DBScan algorithm has more relaxed assumptions regarding the shape of the clusters, works eacwell with different shapes of clusters and does not require that the clusters are convex (e.g. it is able to differentiate elongated clusters of different shapes). Another advantage is that it also allows for points not to fit in any clusters, therefore is more robust to noise. However, as the algorithm must execute a neibhourhood query for each point, the computational cost of the algorithm is higher than K-Means (in the order of n. log(n)). When compared to Hierarchical Clustering, the algorithm is not entirely deterministic, as it starts with a random point. This implies that border points that are reachable from more than one cluster can be part of either cluster. Moreover, the DBScan algorithm performs quite poorly on datasets with large differences in density (an appropriate n-eps combination is difficult to find) or with sparse datasets.

***Silhouette Plots - A bit of theory***

In this study, the silhouette plot will be used as a validation technique. For a given point in a dataset, the silhouette coefficient is the ratio of two numbers: the difference between the average interclustering distance of the point (w.r.t. all points in the cluster) and the average intracluster distance (w.r.t all points in the neighboring clusters) and the maximum between the interclustering distance and the intraclustering distance).This number can take a value from -1, (where -1 indicates that the point in the cluster is dissimilar and has probably been incorrectly assigned), to 1 (an indication that the clusters are correctly assigned/the point is similar to all other points in the cluster).

**Conclusion and Further Works**

Overall, we notice that the DBScan does not perform well on the dataset, mainly resulting in different clusters with very few observations. Focusing on K-Means and Agglomerative Hierarchical Clustering (AHC) we notice that AHC performs slightly better (this could be due to the fact that the clusters are not spherically shaped). When compared to K-Means, the AHC tends to lend less importance to classifying the dataset by average income, using average spending and average age to differentiate the clusters. On the other hand, K-Means tends to cluster by average income and average spending score (and then subdivide by “age” and/or“gender”). Overall both methods tend to agree on two groups with average income and average spending score (one of which is women-dominated and the other male-dominated). They tend to slightly differ on the average age of this group and the size of the cluster. Both groups also highlight the existence of two high average-income mixed-gender groups, similar in size, and further differentiate the groups by average spending score (one has a low spending score while the other one a higher one). K-Means also comes up with a fifth group with low income and higher spending (mainly composed of women), this groups seems to be distributed between group 1H and group 2H (in the hierarchical cluster), which tend to be less divided by gender (when compared to the k-Means method which clearly divides the groups by gender).

Focusing on K-Means and the Agglomerative hierarchical clustering, we suggest that looking into the different variables independently (e.g. gender vs average age …) could be interesting. Indeed seeing as the variables are rather uncorrelated, selecting less variables could have resulted into a different choice of clusters being preselected. For example, further works could explore if excluding the gender variable (which was originally categorical in the dataset) could result in different group-types and/or whether the average silhouette score would be higher.