Data Mining Project Report

Paralyzed Veterans of America

Customer Segmentation

and

Marketing Approach

Group AI:

Anastasiia Tagiltseva (m20200041)

John William Villalobos Ruiz (m20200540)

Gabriel Azenha Cardoso (m20201027)

Índice

1. Introduction

Our client, The Paralyzed Veterans of America (PVA), is a non-profit organization that provides programs and services for US veterans with spinal cord injuries or disease and is also one of the largest direct mail fundraisers in the United States of America.

A dataset was provided to analyse the results of one of the PVA's latest fundraising appeals. Our task is to develop a Customer Segmentation in which we'll be able to better understand how their donors behave and identify the different segments of donors/potential donors within their database.

The project can be found on, *https://github.com/johnruiz24/DM_Project_2020*, where a Jupyter notebook with our analysis and following documents can be found.

2. Data Understanding

The first step to tackle this task is to check and try to understand the dataset supplied. It can be observe that the shape of the dataset is of **95412** donors (rows) with **475** features (columns).

	ODATEDW	OSOURCE	TCODE	STATE	ZIP	MAILCODE	PVASTATE	DOB	NOEXCH	RECINHSE	...	AVGGIFT	CONTROLN	HPHONE_D	RFA_2R	RFA
0	2009-01-01	GRI	0	IL	61081			1957-12-01	0		...	7.741935	95515	0	L	
1	2014-01-01	BOA	1	CA	91326			1972-02-01	0		...	15.666667	148535	0	L	
2	2010-01-01	AMH	1	NC	27017			NaN	0		...	7.481481	15078	1	L	
3	2007-01-01	BRY	0	CA	95953			1948-01-01	0		...	6.812500	172556	1	L	
4	2006-01-01		0	FL	33176			1940-01-01	0	X	...	6.864865	7112	1	L	

5 rows × 475 columns

Figure 2.1. Dataset head

Through the pandas_profiling package, we got a quick overview of the whole dataset.

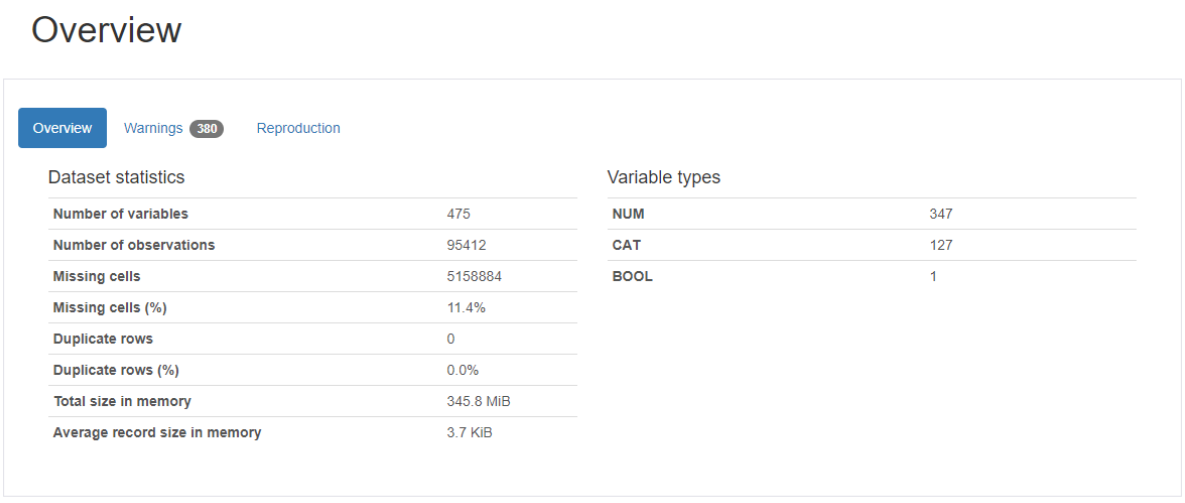


Figure 2.2. Initial Pandas Profiling report to raw data

According to the report, there are 11.4% missing values across all features. The package also includes correlation analysis and although this dataset is too big to fully understand all the correlations, we applied anyway the Phik coefficient to check the correlations between all the variables.

Phik is a new correlation coefficient that works consistently between categorical, ordinal and interval variables, captures non-linear dependency and reverts to the Pearson correlation coefficient in case of a bivariate normal input distribution.

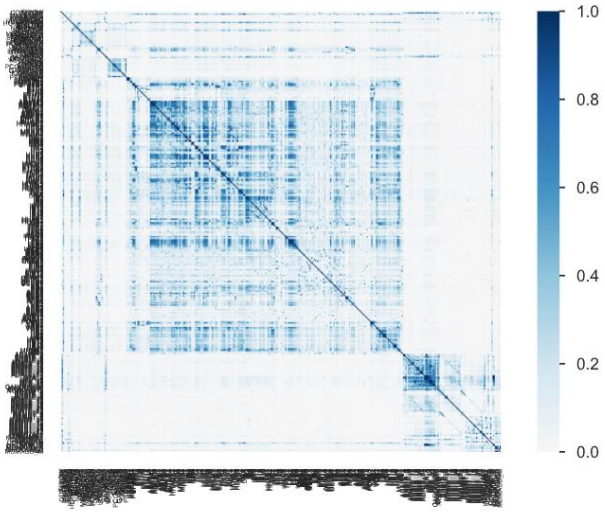


Figure 2.3. Phik Correlation heatmap

The correlation heatmap doesn't give us relevant information, there are no obvious patterns nor visible tendencies, the only observation noteworthy is that there are some highly correlated variables, which become redundant to the data exploratory analysis and clusterization, so that is something we'll have to check and analyse on the coming feature selection steps.

3. Data Preparation

3.1. Data Consistency

This dataset has a lot of information and consequently a lot of mistakes regarding the relational coherence between some of the features.

Promotion related corrections are the priority due to the importance of this data being trustworthy for our clusterization. From a deeper logical observation, we established our analysis priorities:

1. Check if "RDATE" (date the gift was received for promotion xxxx) entries were registered before the correspondent "ADATE" (date the xxxx promotion was mailed);
2. Check for empty "RAMNT" (dollar amount of the gift for promotion xxxx) entries with a correspondent "RDATE" and vice versa;
3. Check for empty "RFA_" (donor's recency/frequency/amount status as of xxxx promotion date) entries with a correspondent "ADATE" and vice versa;
4. Check for "DOB" (date of birth) entries with correspondent previous "FISTDATE" (date of first gift).

The result from those check steps is observable in fig. 3.1, where we can see the existence of 6999 entries where a $ gift from an x promotion was sent before the actual promotion and 281 entries where the donors first gift date is previous to its own date of birth.

```
RDATE before ADATE  6999
Empty amount with not empty date  0
Empty date with not empty amount  0
Empty RFA with not empty ADATE and vice versa  0
DOB is later than one of dates  281
```

Figure 3.1. Promotions coherence check

RDATE's inconsistency fixed by imputing mode on the rows with the spotted issue.

Correction in ZIP codes where it seems to exist some improper symbols. Those entries are going to be replaced with " ' ' " in order to avoid confusion and ease up the readability of this data.

Split MDMAUD (Major Donor Matrix code) into 4 columns, segregating the information by MDMAUD_R = 'Recency of Giving', MDMAUD_F = 'Frequency of Giving', MDMAUD_A = 'Amount of Giving' and MDMAUD_M where we simply identify the donor as 'major' or not.

Encode variables who can't properly be changed by ordinal encoder according to the metadata, so to do that we replaced the values of NOEXCH, GENDER, SOLP3, SOLIH and HOMEOWNR with values that logically make sense and better translate the information.

3.2. Missing Data

The dataset has 92 different variables that showcase missing values and before filling anything, we check one by one all those variables to understand what those missing values could mean.

We observed that MSA (MSA code), ADI (ADI code), DMA (DMA code) and GEOCODE2 (county size code) had the exact same number of missing values. With that in mind we checked how many rows had missing values from the 4 features combined and confirmed that all those 132 values from each feature were simultaneously missing which led us to conclude that these features are related. In this case, has those entries have a low record quality we decided to drop instead of imputing new values.

The **2 rows with nans in FISTDATE were deleted** as are donors without a recorded first donation so become irrelevant to our clusterization.

Accessing the metadata, we observed that some of the missing values in features were not actually empty but representative of zero, as is the case of:

- Variables that indicate the number of known times the donor has responded to other types of mail order offers which all have a correspondent value on HIT (mail order response which indicates the total number of known times the donor has responded to a mail order offer other than PVA's) and if the total mail responses is above zero we conclude that there was at least one response;
- Campaign related variables where a nan means that the donor didn't respond to that particular campaign;
- Gift related dates where the donors contributed at least once but don't contribute to any other campaigns.

The first two cases the nans were filled with '0' and the last case we filled NEXTDATE with a 'Timestamp(0)' and TIMELAG with '9999'.

Next, we addressed the numerical features of our nan list.

We began by converting “DOB” into “AGE” and reset the ages bellow 15 to 0 as those values are negative.

After that step we created a list with the numerical features we decided to impute and that imputation will be made applying the MICE method (Multiple Imputation by Chained Equations) provided by the Iterativeimputer package that deals with missing data by imputing data in a dataset through an iterative series of predictive models, where in each iteration, each specified variable in the dataset is imputed using the other variables in the dataset. Due to the information density of this dataset, this method’s imputation accuracy will be reliable and robust, thus the selection made.

3.3. Feature Engineering

In this step, all **the dates we deemed as logically relevant** for the clusterization were converted into months, which will facilitate a lot our final donor and campaign analysis. The relevant features selected for this transformation were “ADATE”, “RDATE_”, “FISTDATE”, “LASTDATE”, “MAXADATE”, “MINRDATE”, “NEXTDATE”, “MAXRDATE” and “ODATEDW”.

3.4. Outliers

3.4.1. Numerical Features

Prior to outlier detection and removal step, the data needed to be scaled and normalized has the outlier detection depends on densities and distances.

Univariate analysis was our first approach, so we started by applying two common methods on the numeric features in order to normalize the data, the Yeo-Johnson and the Quantile transformations.

In these QQ plots, the data follows a normal distribution if it shows a linear 45-degree pattern close to the red line, in this case we can observe that the Yeo-Johnson transformation doesn’t get us the results we want as there are few features following the desired distribution. Despite the fact our data doesn’t all follows a normal distribution it is scaled, so we are going to proceed with the outlier detection and assess the results.

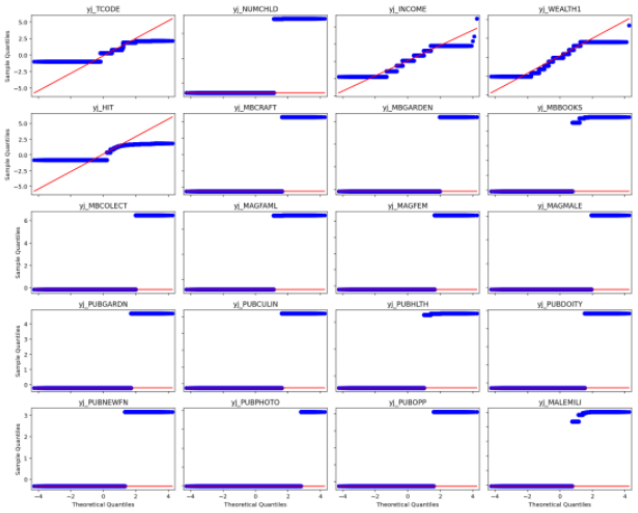


Figure 3.2. Plotted Yeo-Johnson data distribution

Based on the assumption that our data is Gaussian-like, we are going to use the Standard Deviation and apply the IQR (Interquartile Range) method by defining has abnormal values the ones outside the 2.5 x IQR and the ones with more than 4 standard deviations away from the mean. This criterion is more unrestrictive than the usual due to the high variation of the dataset.

For comparing purpose we applied each outlier detection method to each transformed dataset.

```
% of outliers after out_std & yeo-johnshon: 59.99
% of outliers after out_std & quantile-transformation: 49.02
% of outliers after iqr & yeo-johnshon: 99.94
% of outliers after iqr & quantile-transformation: 100.0
```

Figure 3.3. % of outliers detected

These results lead us to conclude that the techniques used to scale and transform the data are not appropriated to handle outliers, in that in mind we **decided to proceed with a multivariate analysis** the application of different techniques.

Multivariate normal distribution has increased our options and one of those is the Mahalanobis distance, which is an effective multivariate distance metric that measures the distance between a point and a distribution. Since our transformation techniques don't guarantees us a normal distribution on all features, the Mahalanobis method in theory seems to be efficient and robust enough to effectively work on both normal and non-normal data.

So, we set our outlier detection algorithm **assuming the input data is normally distributed** and from this assumption we know that we can define the "shape" of the data and can define the outlying observations as observations that stand far enough from the fit shape. Taking that into consideration we applied the "cov.EllipticEnvelope" method (from scikit learn package) that fits an elipse to the central data points ignoring the points outside the central mode and then the Mahalanobis distances obtained from this estimates are used to derive a measure of outlyingness.

```
cleaned_data=remove_outliers(main_data,'index',random_state)
```

```
Outliers Percentage: 4.83%
```

Figure 3.4. % of outliers detected and removed

Applying our Mahalanobis algorithm to the normalized data we detect **4.83%** of outliers and since the percentage is acceptable, we **proceed to the removal of those values**.

We also obtained the **chi-squared percentils** of each observation to the the center in order to check the fitness of our transformed data into the multivariate normal distribution and we plotted the transformed data with and without the outliers in order to compare both situations.

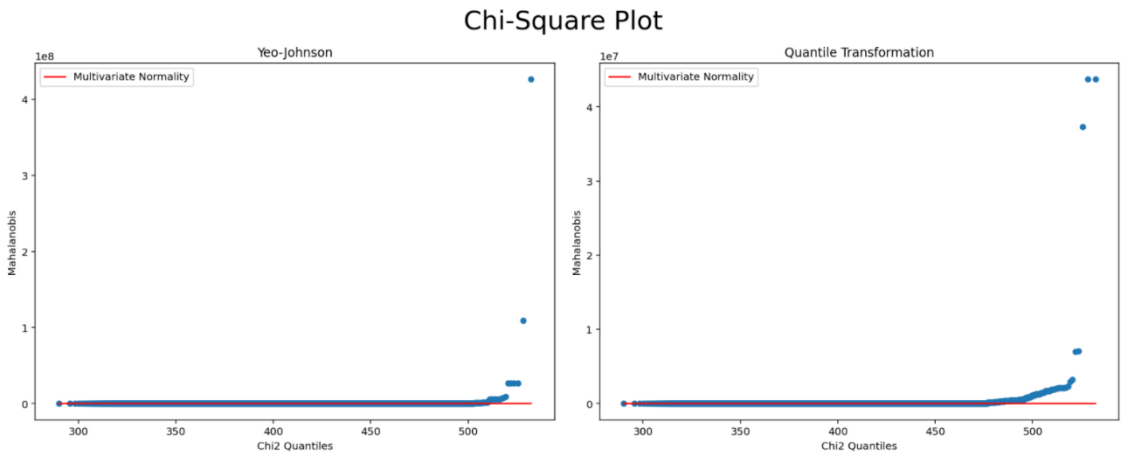


Figure 3.5. Chi-square plot before normalization

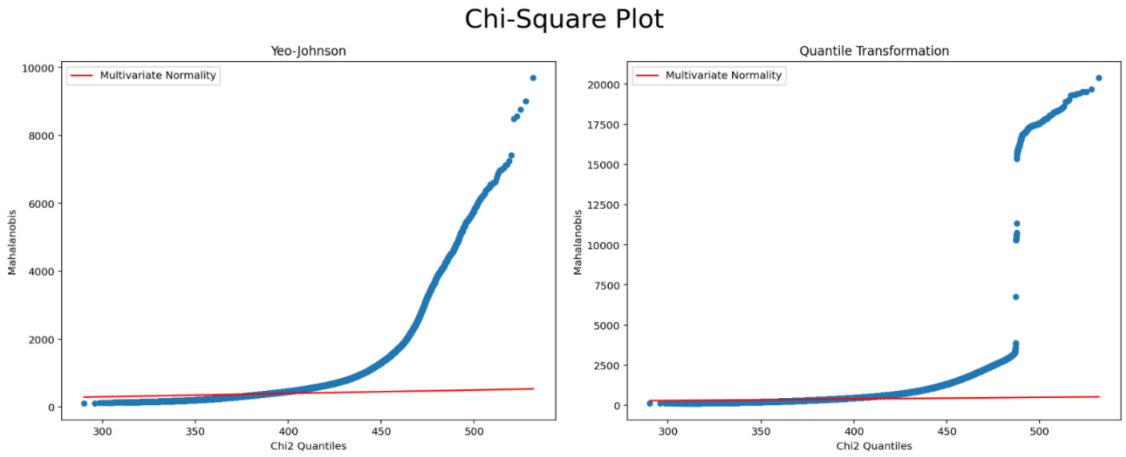


Figure 3.6. Chi-square plot after normalization

Above we can observe our previous statement that Yeo-Johnson and Quantile transformations provide a poor fit to a multivariate normal distribution since d2 doesn't match the Chi-Squared quantiles as it would be expected in the case of multivariate normality (the observations don't match the red line).

In addition, we can assess that our **data without the outliers is much closer to multivariate normality** even though visually looks otherwise. If we take a closer look at the scale on both plots, the data with the outliers has values on the scale of 1e8 which are much far from the centre of the distribution than the values represented on the data without the outliers.

3.4.2. Categorical Features

Last step is to **clean the categorical features**. In this case um created a function that defines a threshold to identify elements that that don't match the minimum number of occurrences that we established and compare it to the maximum number of occurrences per column.

Our algorithm detected 38 features with outlier percentages between 0.01 – 3.85%.

3.5. UMAP for Initial Data Distribution

UMAP (Uniform Manifold Approximation and Projection) is a powerful clustering and dimension reduction technique, that due to its speed and powerfulness is most effective on large datasets than other methods.

To better understand our data before stepping to the Dimensionality reduction part, we are going to apply UMAP to our data before and after the normalization process, for which we applied Yeo-Johnson transformation. Then we embedded both categorical and numerical features by fitting the values and segmented both in order to perform a fuzzy simplicial set embedding by using a initialization method and then minimalizing the fuzzy set cross entropy.

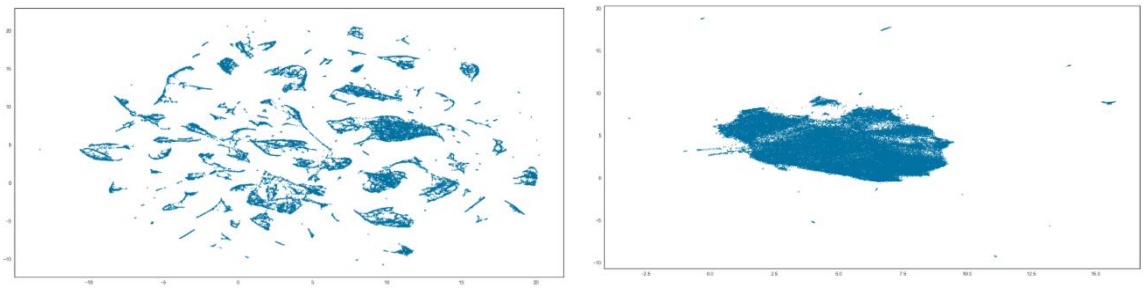


Figure 3.7. UMAP for dataset: Before Normalization VS After Normalization

As we can observe on the visualization above, the data due to the high number of dimensions has, potentially, a high number of clusters which doesn't apply after UMAP is applied, the difference is easily identified and the reduction of dimensions implies a decrease for the number of clusters, which servers our purpose in a marketing perspective.

4. Dimensionality Reduction

The provided **dataset has 475 dimensions, which is not suitable for clusterization**. Clustering usually depends on distance measures and those metrics don't work so perfectly in high dimensional spaces has the more dimensions there are, the sparser the data becomes (curse of dimensionality).

With that in mind, we scaled, fitted and joined the cleaned numerical features to the cleaned categorical data.

To properly select the variables, we grouped all of the 475 in 10 groups:

0. RFA (recency-frequency-donation amount, 6 features));
1. Characteristics of donor neighbourhood (286 features);
2. Demographics (28 features);
3. Donor interests (19 features);
4. Giving history (44 features);
5. Overlay data (13 features);
6. Promotion history (46 features);
7. Response to other types of mail orders (15 features);
8. Summary variables of giving history (13 features);
9. Summary variables of promotion history (5 features).

4.1. Metric Features

The groups who feature the metric variables are groups 1, 2, 4, 7, 8 and 9. To address these features, **we selected PCA (Principal Component Analysis)** which is a popular linear dimension reduction algorithm. PCA is method that reduces the number of variables while preserving as much information as possible, working as a projection-based method that transforms the data by projecting into a set of perpendicular axes.

It was observed that each one of the groups as a **list of relevant features assigned**. With PCA we're able to reduce the numerical features from 363 to 58 features. Even though we did significantly reduce the dimensions, we still must reduce more to proceed to the clustering.

4.2. Non-metric Features

The groups we targeted in this section are 0, 2, 3, 5 and 6.

To those groups, we are going **to apply two different techniques**, both like PCA but both address metric and non-metric data, MFA (Multiple factor analysis) and FAMD (Factor analysis of mixed data) respectively.

MFA is a multivariate data analysis method for summarizing a dataset in which the observations are described by sets of variables (numerical and categorical) structured into groups. In practice it builds a PCA/MCA (depending on the feature's type) on each group, then it constructs a global PCA/MCA on the results of the so called 'partial' PCAs/MCAs.

FAMD works as both PCA and MCA, dedicated to analysing the similarity between individuals by taking into account mixed types of variables and explore the associations between all those variables. This method is mixed between PCA and MCA where features are normalized during the analysis in order to balance the influence of each set.

Both methods were applied with a explained ratio cumulative percentage of 80%.

On application of the algorithm to the *Demographics* group, the number of dummy values created by the encoding of all features, when exploded, rises around 17k columns, which it is very hard to work on. Due to that fact, we decided to disregard the ZIP column as we are using STATE to provide similar information and decreasing the exploded columns account down to 1k.

4.3. Feature Selection

With the previous techniques we reduced the dimensions from 475 to 108.

As previously stated, on the raw dataset there were some high correlations noticeable and at this point we'll re-use the phik distance to complete our dimensionality reduction task by running it individually on each one of the groups to check for the correlations within and select the feature that make sense on a principal component, correlation and logical level.

After running our phik function and plotted the corresponding heatmaps we come up with the following feature selections per group:

0. RFA (recency-frequency-donation amount) - **RFA_2F, RFA_2A**
    - In this group although RFA_2F and RFA_2A are highly correlated but since we observed that MDMAUD_F and MDMAUS_R have a poor representation, we opted for the RFAs.
1. Characteristics of donor neighbourhood - **IC3, HHD2, ETHC3**
    - IC3 and HHD2 are highly correlated but are the two most relevant features, we'll take them both for clustering regardless;
    - ETHC3 as no high correlation problems with the other features already selected and it also explains information from other relevant features.
2. Demographics - **INCOME, N_ODATEDW, NUMCHLD, AGE**
    - INCOME and AGE are the most relevant features of the group;
    - According the component analysis both N_ODATEW and NUMCHLD are relevant features. They haven't correlations problems although N_ODATEW is correlated to one irrelevant feature and both make logical sense to proceed with our analysis.
3. Donor interests - **PETS**

- Component analysis indicates PETS has the most relevant feature and explains a huge part of the whole group, since according to phik, it correlates with a lot of other important features.

4. Giving history - **N_RDATE_7, RAMNT_7**
    - In this group we took a different approach, the logic was to select one campaign representative of the others, so we selected campaign *96G1* related features having in consideration the following:
        - This campaign is in the top 5 in terms of responses in donations;
        - Is the 5th most relevant of this group;
        - Doesn't have any correlation issues with other campaign features.
5. Overlay data - **FEDGOV, VIETVETS**
    - According to the component analysis, FEDGOV and VIETVETS are the more relevant features;
    - VIETVETS is correlated to other relevant features.
6. Promotion history - **N_ADATE_7**
    - N_DATE_7 is the feature related to promotion *96G1.*
7. Response to other types of mail orders - **MBBOOKS, MBGARDEN**
    - MBBOOKS and MBGARDEN have a huge combined importance.
8. Summary variables of giving history - **N_FISTDATE, MAXRAMNT, N_LASTDATE, N_MAXRDATE**
    - In this group we selected the top 4 most relevant features due to the logical sense they make to our final objective.
9. Summary variables of promotion history – **CARDPROM**
    - CARDPROM is the most relevant feature of the group.

Concluding, based on the phik, the component analysis and the logical relevancy we've selected the features above. They have shown a high adherence degree, are strongly supported by different dimensionality reduction techniques applied and explain a high % of the dataset variance.

5. Clusterization

With the dataset properly processed and the significant variables known and selected we proceed to the clustering and analysis part of our project. For this part we applied different clustering techniques, such as the **Silhouette Visualizer**, **K-means, K-prototypes, Hierarchical Clustering and SOM (self-organizing maps)** and even merged some of them to improve performance and validate results.

5.1. Number of clusters

The first concern was the **number of selected clusters** for our data, for that we defined a function where we apply **K-Elbow** to a given model to get the ratio per number of clusters and find the optimal number of clusters for the data we have. To calculate the ratio, 3 different metrics were selected, the Distortion, the Silhouette and the Calinski Harabasz measures.

Having that defined we called the function with k-means for initialization and display purposes:

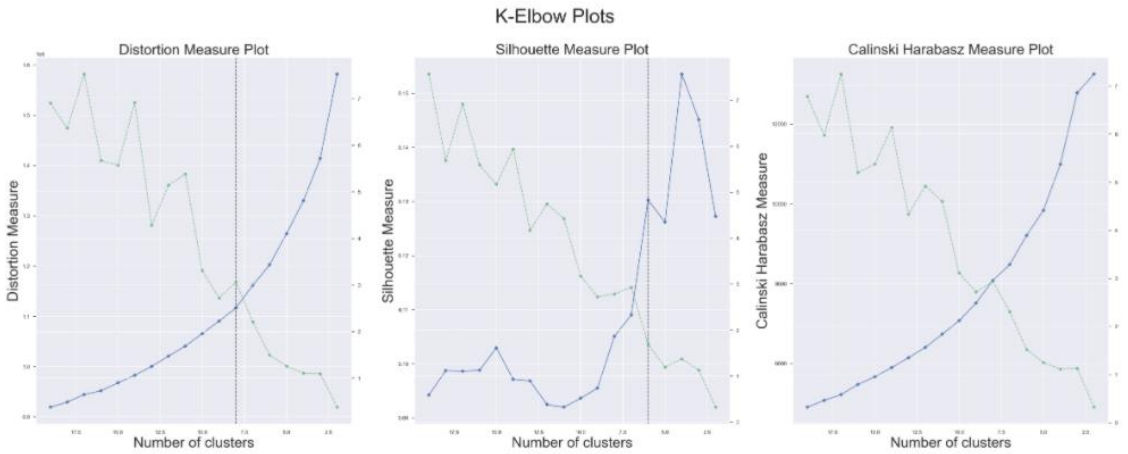


Figure 5.1. K-Elbow Plots

From the plots one can observe that both Distortion and Calisnki Harabasz display around 8 clusters where the Silhouette measure gives us approximately 6 clusters. To take a closer look, we plotted the Distortion and the Silhouette individually.

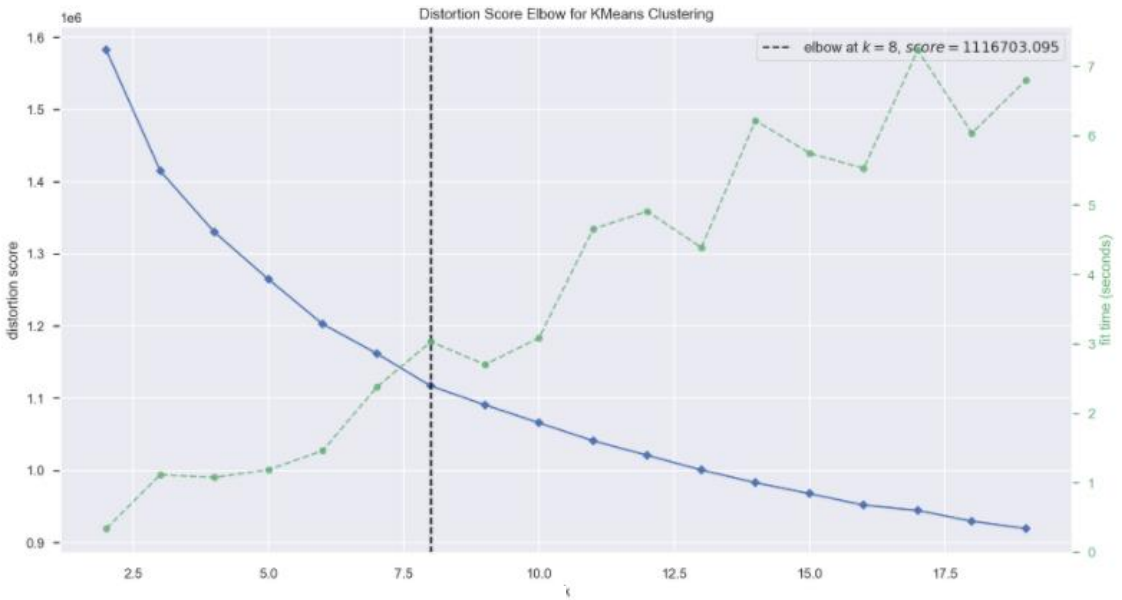


Figure 5.2. Distortion elbow plot for K-means

With the individual Distortion plot we can confirm that the **number of clusters recommended by the metric is 8**.

Next, plotted the Silhouette metric and checked the results. From our observation of those results, we confirmed that the highest score is attributed to 6 clusters. Bellow there is a comparison between the 8 clusters plot with the 6 clusters plot.

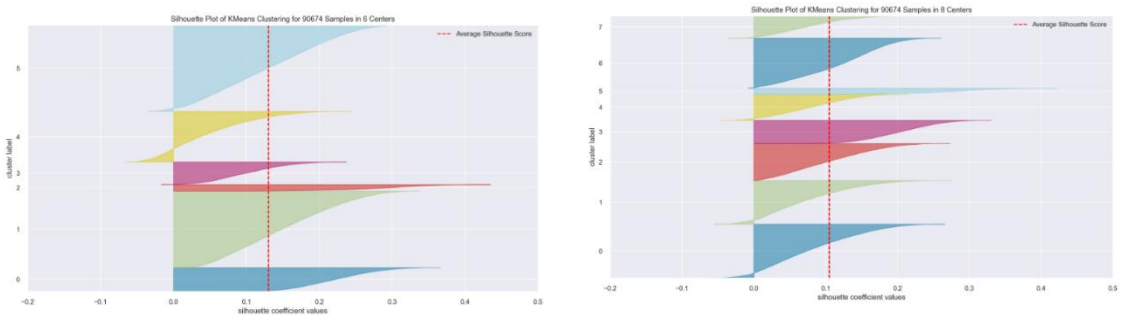


Figure 5.3. Silhouette plots for: 6 VS 8 clusters

The premises to consider for a visual analysis and comparison of Silhouette plots are the negative values per cluster, which represent the number of samples that might have been assigned to the wrong cluster and the size of each cluster has all the plots should have similar thickness.

Taking a closer look to the plot of n=6, it has negative values on 3 clusters, with the cluster 3 having a coefficient around -0.7/-0.8, which is closest to be the cluster with more misplaced samples in one cluster where in the **n=8 clusters plot** we observe 5 clusters with negative values where even though one on those has a high negative it isn't too thick. Concluding, both look to have an equivalent amount if misplaced samples but the **cluster 4 of n=6** seems to have more than half of the misplaced values from all the 14 cluster's misplaced values.

Moving on to the next premise, on the **n=6 plot** we can observe that there are 3 groups of thickness, clusters 5 and 1, clusters 4, 3 and 0 and cluster 2. On the **n=8 clusters plot** we can group the clusters into 4 different sizes, clusters 0, 1, 2 and 6, clusters 4 and 7, cluster 3 ad finally cluster 5. Comparing visually all the groups we observe that on **n=6 plot** the difference of size between them is significant whereas on **n=8 plot** there is an obvious difference between cluster 5 and the rest, but the other 7 clusters have similar sizes which means the data is consistently segmented.

Having in consideration that the other metrics pointed a number close to **8 clusters** and even though n=8 doesn't have the highest coefficient, the Silhouette plots support that n=8 is also a viable option. **In conclusion we selected n=8 for K-means clustering.**

5.2. K-means

K-means and K-prototypes were compared so we could select the best method to merge with other techniques and to proceed with our analysis.

The clustering method selected was K-means, which is one of the simplest methods that basically allocates every data point to the nearest cluster while keeping the centroids as small as possible and where the centroid is found by an averaging of the data. To note that this method is only appliable to numerical features. The analysis of k-prototypes will be available on the Appendix chapter, where we have the dissection displayed.

After applying the method to our data and segmentate the observations through the 8 clusters, we've performed an inertia plot to validate the previous number of clusters selected.

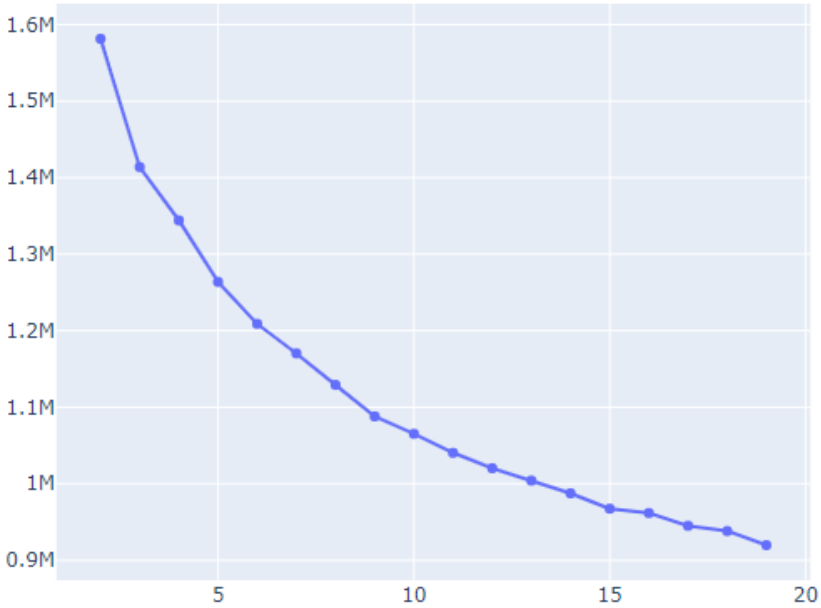


Figure 5.4. Inertia plot for K-means

Although the exact number of clusters is not totally clear, the plot confirms that 8 clusters is a suitable option for k-means run efficiently.

On the next step we've applied the UMAP embedding technique to the k-means data output. UMAP seeks to accurately represent local structure and better incorporate global structure. This technique has showed better performance to preserve both the local and global structure and it has proven to meet our needs.

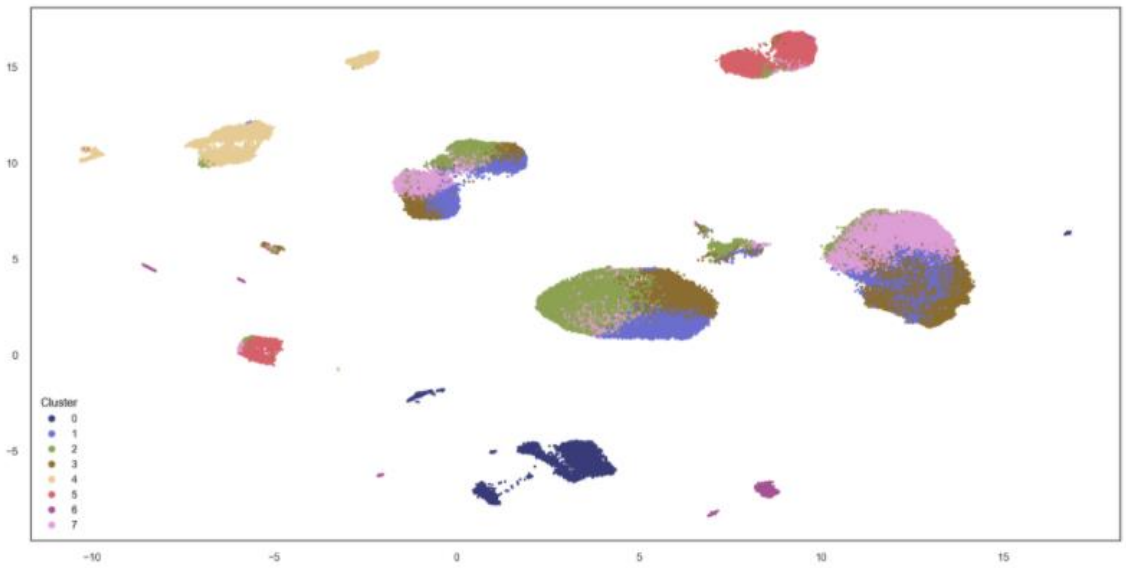


Figure 5.5. UMAP for K-means

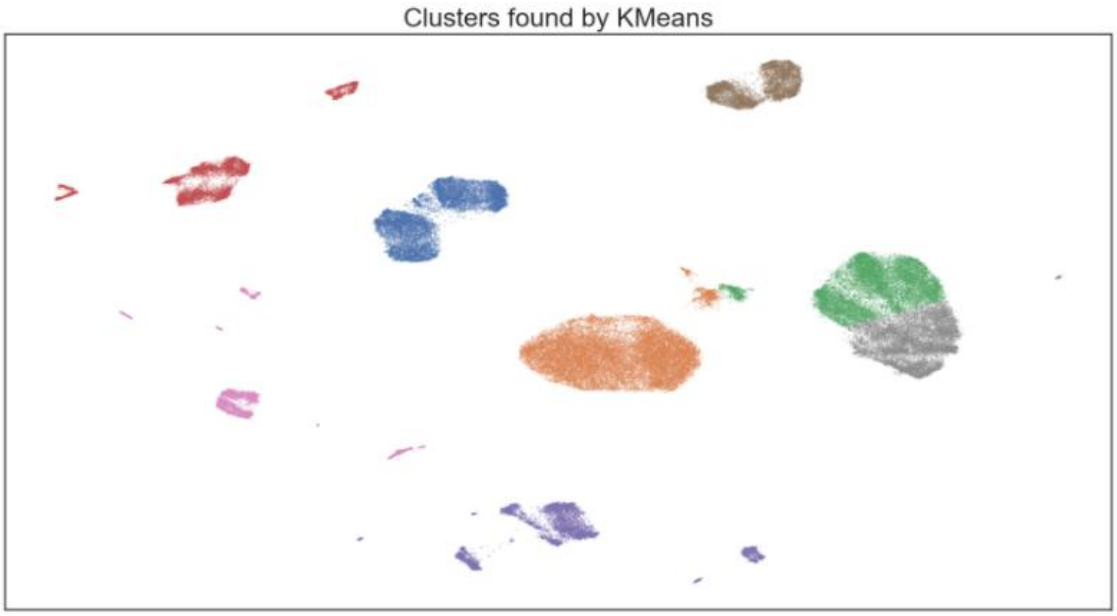


Figure 5.6. Scatter plot of K-means with UMAP embedding

As observed above, we can conclude the 8 clusters are clearly distinguishable and well segmented.

Now, to evaluate the cluster quality and to have a general idea of how the clusters are distributed in space, the **intercluster distance map in 2 dimensions** was plotted, where the closer the centres are to each other, the closer they are in the feature space. To have a visual perception of the cluster importance, they are sized according to a scoring metric where the highest is the score the biggest is the importance, however, if two clusters overlap in the 2D space, doesn't imply that they do overlap on the original feature space.

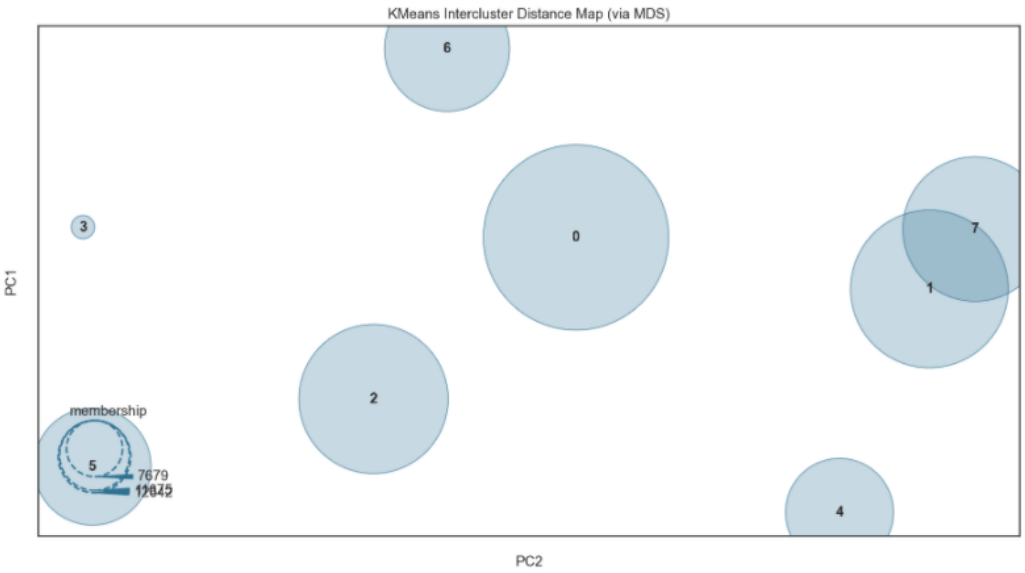


Figure 5.7. K-means intercluster distance

As observed, there is two overlapping clusters, and all the other centres are evidently separated. As it was stated this 2D overlapping doesn't mean the clusters overlap in a n-dimensional space but it could indicate that the 8 cluster solution might not be ideal.

To validate the measure our **cluster's classification quality we applied LGBM** (light gradient boosting machine) which treats the clusters as labels and builds a classification model. If the clusters are of high quality, the classification model will be able to predict them with high accuracy.

```
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
CV F1 score for K-Means clusters is 0.9753373323220856
```

Figure 5.8. F1 score of LGBM model for K-means

As observed the model as a F1 score of 97.5% which implies that we have good clusters.

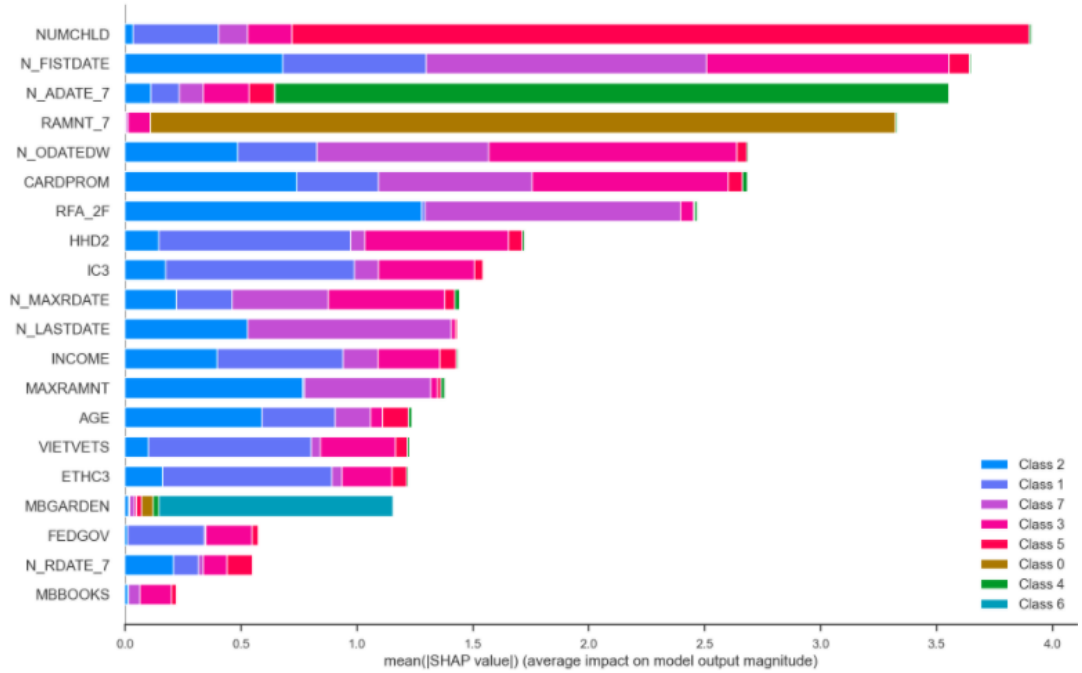


Figure 5.9. K-means Summary plot

This summary plot lets us know how much of each variable's observations are distributed by the clusters and have a broader idea of the importance each variable has. The top 7 variables are not only the most important but also have a huge weight compared to the rest.

Comparing k-means to k-prototypes, we confirm that the F1 score is close to 1 on both methods, which implies that k-means and k-prototypes have produced clusters that are easily distinguishable. Yet, the classification score on K-means is slightly superior and since k-means is a simpler and faster algorithm we choose it to merge with other techniques and aim for an improvement.

Since k-means doesn't work with categorical features, **RFA_2A and PETS were encoded** to include them on the clustering analysis. Both have a small number of unique values (4 and 2) which doesn't complicate too much the clustering.

5.3. K-means & Hierarchical Clustering

Even though k-means displayed a set of clusters with quality and is a simple and fast method we know that the initial centres are chosen from random observations, which means the final k-means solution can

be different each time the algorithm is computed. To get around this situation, k-means can be combined with other clustering methods that don't require a predefined number of clusters.

Having that in mind we applied a set of different combinations to improve and compare our clustering work that are going to be displayed on the appendix. As shown below, from all the combinations applied, K-means & Hierarchical Clustering was the method that better performed.

After running k-means we computed a R2 plot to check which hierarchical method we're going to use on our HC.

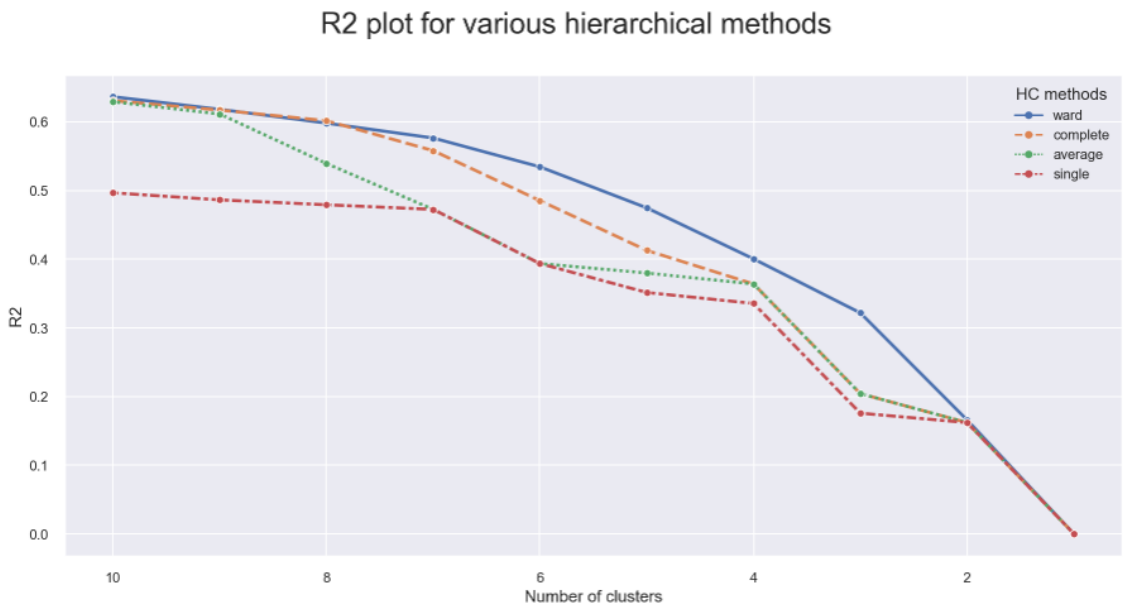


Figure 5.10. R2 plot for Hierarchical Clustering methods

The linkage method selection was based on a R2 plot, which compares the different clustering methods based on the R2 measure, calculated ratio between the sum of squares of the residual errors and the total sum of the errors. From the graph above we can conclude that the best hierarchical clustering linkage method is the ward method.

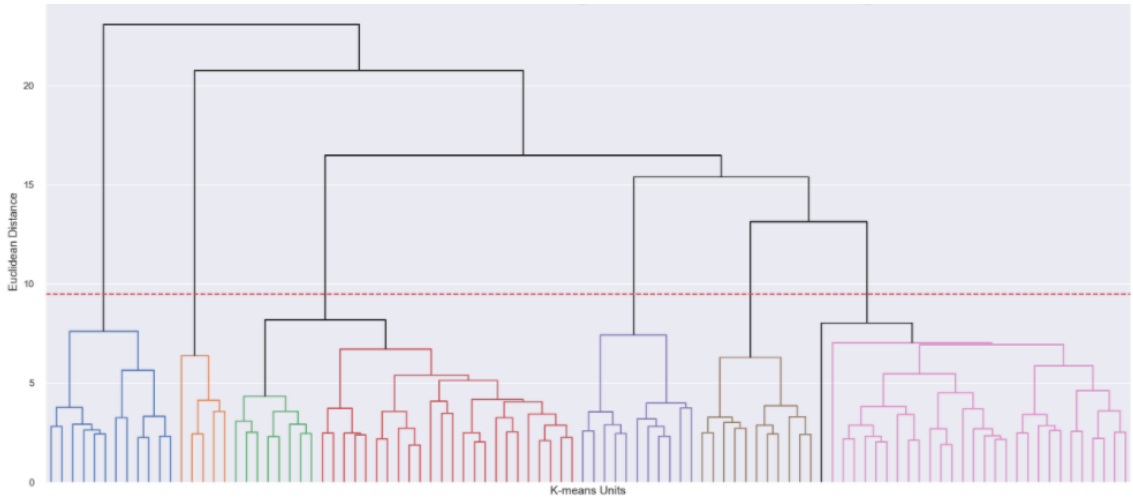


Figure 5.11. Hierarchical Clustering (Ward's Dendrogram)

The dendrogram was computed with the **Ward's linkage method**.

According to the dendrogram above the tree should be cut off on level 6, leaving us with that same number of clusters. For confirmation purposes, K-Elbow plots were computed. From those plots it's observed a consensus with the dendrogram as all 3 metrics applied suggested a optimal number of 6 clusters.

5.4. Accuracy Metrics

After trying out all the different methods, we based our final decision on this table, where we compared all the methods tested to a different variety of clusters. Has previously analysed, k-means + hierarchical clustering with 6 clusters is our optimal solution:

	KM	KM+HC	HC+SOM	KM+SOM	GMM	DBSCAN
Silhouette_Score	0.130104	0.237995	0.235647	0.196629	0.0895409	-0.10352
Calinski_Harabasz_Score	9213.6	22.1867	13.1287	13.6484	6309.01	1063.18
Davies_Bouldin_Score	1.97074	1.33865	1.12042	1.19123	2.32194	2.50095

Figure 5.12. Accuracy coefficients table for k=6

Looking at the table above, there is the visual confirmation that the algorithm with better Silhouette score is the K-means & HC combination, besides that, the other scores are balanced as it has intermediate scores on the Calinski Harabasz and Davies Bouldin measures. Considering those values, we selected this combination method to analyse and draw our marketing approach.

6. Marketing Approach

Finally, having the cluster method selected we gathered all the data and distributed it into 6 clusters, according our algotithm. Based on that information suggestions were drawn to lure customers from the different clusters.

Cluster	0	1	2	3	4	5
RFA_2F	1.895010	1.894850	1.810676	1.734504	2.363964	1.900046
IC3	386.739469	348.579127	354.708730	517.003997	378.654539	372.906525
HHD2	71.797667	69.118520	68.738060	81.653690	71.783186	73.445627
ETHC3	17.505768	20.131617	17.530343	10.694937	17.048553	17.728829
INCOME	3.821257	3.388881	3.746128	5.087765	3.780330	3.676076
AGE	59.749190	65.303547	56.209017	51.196153	59.247734	61.849607
N_ODATEDW	113.251458	151.997067	76.040343	111.104855	112.115237	125.219806
NUMCHLD	0.168373	0.072237	0.137165	0.529487	0.192972	0.257288
N_RDATE_7	612.000000	612.000000	612.000000	612.000000	58.100211	557.402129
RAMNT_7	0.000000	0.000000	0.000000	0.000000	15.528080	1.465988
FEDGOV	3.047440	2.719134	2.949983	4.004866	3.006830	3.155484
VIETVETS	29.701620	26.379025	27.722343	39.428861	29.783807	30.317446
N_ADATE_7	612.000000	59.117156	59.128179	59.560364	59.409661	105.622397
MBBOOKS	0.412054	0.492599	0.351459	0.571544	0.430274	2.495604
MBGARDEN	0.000000	0.000000	0.000000	0.000000	0.000000	1.106432
N_FISTDATE	109.533247	144.871419	75.779664	107.681265	108.312058	120.300787
MAXRAMNT	19.186434	20.337427	19.669591	22.087804	18.179142	18.670292
N_LASTDATE	60.160207	61.745771	61.727314	62.684046	57.803303	61.226284
N_MAXRDATE	72.382631	83.558595	64.400826	71.736010	72.558053	75.725590
CARDPROM	16.252625	25.956548	10.617300	18.288669	17.682975	20.259602
merged_labels	1.000000	0.000000	1.000000	1.000000	0.000000	0.000000

Figure 6.1. Final Cluster Centroids

Cluster 0. (Middle-class donors)

This is a medium-sized cluster - 7715 respondents.

They were not received promotion 96G1 (Date the 96G1 promotion was mailed is maximum). They have practically the largest personal income; donations occur quite often (RFA_2F) and Dollar amount of largest gift to date was in 4th place out of 6. For most clustering variables, this cluster is close to the central values. As a variant of a possible strategy, it is recommended to test the 96G1 campaign on them as a typical one according to the dataset sample to assess the response and the strategy for further communication according to the results obtained.

Cluster 1. (Donations are less often than average, above average)

Majority (donations are less than average, above average) maximum cluster - 29,320 donors. Donors of this cluster are the most experienced donors - they have the earliest date of the first donation, the maximum amount of CARDPROM. This is the oldest group (65.30) with the maximum Percent White Age 60+ among the neighbors. At the same time, they have the lowest personal income among other clusters and practically have no children. They donate not very often (RFA_2F 4th out of 6), but in large amounts (MAXRAMNT 2nd out of 6). To maximize the income for mailings for this cluster due to its significant volume, it is recommended to work on the percentage of responses, increasing the frequency, even though this may lead to a decrease in the size of the average donation because of the effect of volume, the total income from this cluster will increase.

Cluster 2 (New)

This is the second-largest cluster with 26,151 donors. The date of the donor's first gift is minimal - these are new donors who also received some CARDPROM. At the same time, they have a low Percent of Households w / Families and Buy Books (they do not work with book mailings). Since they have an average Dollar amount of largest gift to date and, at the same time, the Frequency code for RFA_2 is rather rare (5 out of 6), you need to work on the frequency of communication with them, using information that they are new - tell more about the organization and its needs, involving them in the active life of the foundation, taking into account the demographic characteristics of the cluster.

Cluster 3 (Young, rare but large donors)

This is an important 17262-donor cluster. This cluster has a maximum Average Household Income in hundreds and HOUSEHOLD INCOME, as well as the highest NUMBER OF CHILDREN and % Employed by Fed Gov, % Vietnam Vets, and Percent Households w / Families. They also have a maximum Dollar amount of largest gift to date, but a minimum Frequency code for RFA_2. As for recommendations for interaction with this cluster, local charitable family events are recommended in favor of the organization in assistance with the federal government (due to the high % Employed by Fed Gov) with the emphasis is on donation size.

Cluster 4 (frequent small donations)

This is a cluster of 8053 donors. This is the only cluster that performs well in the 96G1 mailing list. They have the maximum Frequency code for RFA_2, but the minimum Dollar amount of largest gift to date. The last donation was also closer to all other clusters (Date associated with the most recent gift). As a communication strategy with this cluster, attention should be paid to the maximizing donation sizes while maintaining donor loyalty (for example, using the success of the 96G1 campaign or indicating the need to replenish the fund for a certain amount for a limited period for a specific purpose).

Cluster 5 (respondents to other types of mail orders)

The smallest cluster is 2161 donors. These donors are characterized by a high percentage of participation in the Buy Books and Buy Gardening mailings. They have an average personal income and small but stable donations. As a strategy, it is recommended to maximize donations with books or garden supplies, such as cross-promotion with book coupons in exchange for a donation.

7. Conclusion

Finishing this project, we realized the importance of data preparation and preprocessing. The dataset provided was heavy, with 475 features and 95412 entries, and had a lot noise associated to it. While preparing the data, we always had the care of not removing any features or entries until the outlier removal and dimension reduction part, as our premise we deemed all information necessary and relevant for clustering purposes, although some features had a huge amount of missing data, we knew that the MICE imputation method would work well on this case due to the amount of information available.

On the dimension reduction section, note the divided features into groups that helped us to better understand the logic of all the variables and combining that with the Component Analysis and Phik techniques we easily selected the relevant and logical features to proceed our work. In this section we tried to have a unique approach compared to others and opted by applying first the PCA and then the Phik but for future projects we'll take into consideration applying first the Phik to remove highly correlated variables and then on top of it the PCA with a weigh matrix to have a more precise selection.

The clustering results made sense as we applied some different technique in order to perfect our data segmentation and after trying out diverse methods with diverse number of clusters, we found our optimal solution with the K-Mean agglomeration with Hierarchical Clustering on 6 clusters.

Finally, having the data distributed to each cluster, we propose a series of strategies to approach and conquer the different client's profile identified that can be enhanced by the marketing department in order to work more effectively.

8. Appendix

8.1. K-Prototypes

K-prototypes clustering method is a combination of k-means and k-modes as it works for mixed type datasets (categorical and numeric features), it measures the distance between numerical features using the Euclidean distance (same as k-means) and measures the distance between categorical features using the number of matching categories.

Considering that, the same methods applied on k-means were applied to k-prototypes.

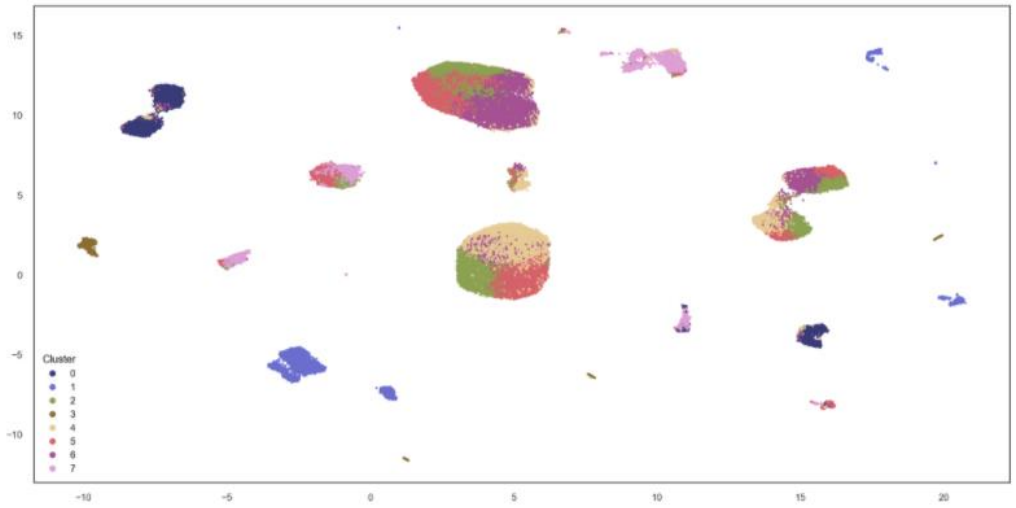


Figure 8.1. UMAP for K-Prototypes

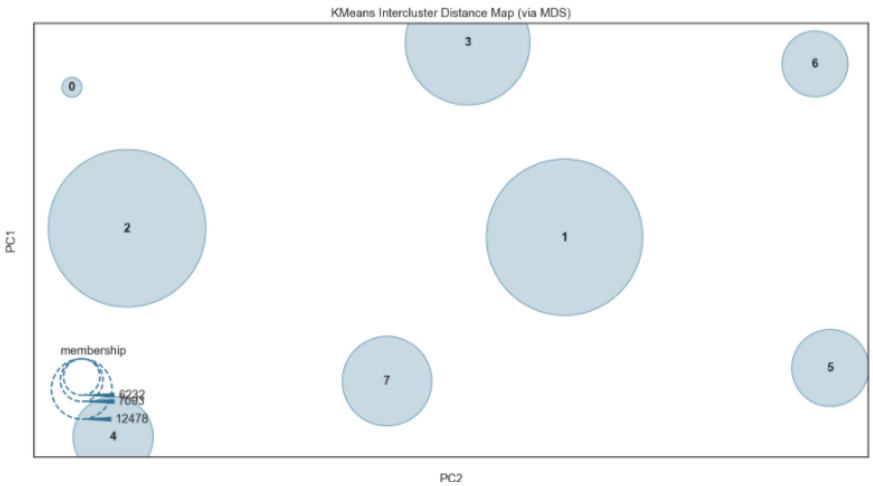


Figure 8.2. K-Prototypes intercluster distance map

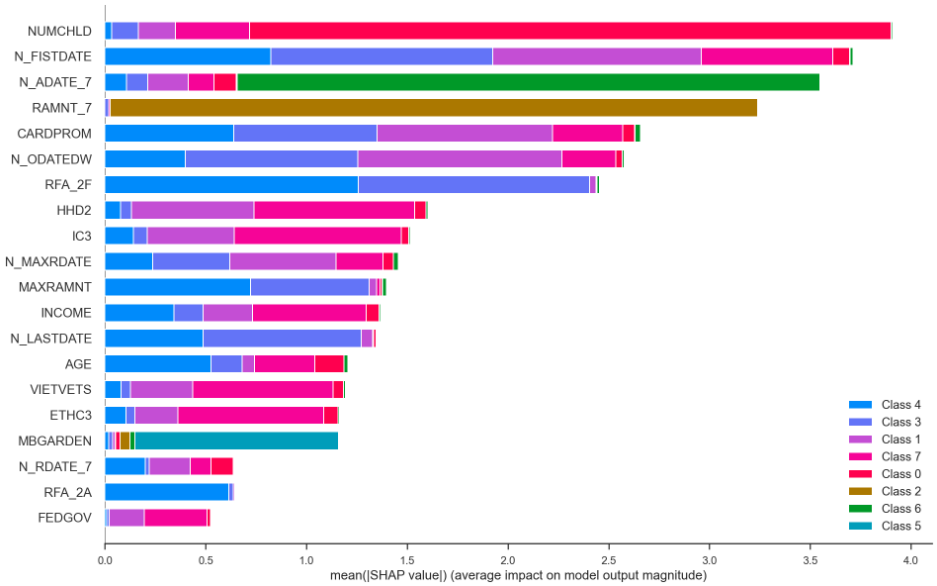
Also, there aren't any overlapping clusters, and all the centres are significantly distanced from each other, once again positive signs of our clustering quality.

Next step, **check the LGBM score**.

```
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
[LightGBM] [Warning] Unknown parameter: colsample_by_tree
CV F1 score for K-Prototypes clusters is 0.9737058157558366
```

Figure 8.3. F1 score of LGBM model for K-Prototypes

The 97.3% is slightly lower than the k-means but for an insignificant margin, this clustering can also be labelled as of high quality.



8.2. SOM & Hierarchical Clustering

Since SOM provides a powerful clustering algorithm but as the downside of the parametrization need (for example the definition of the grid of units that covers the feature space) and the hierarchical clustering

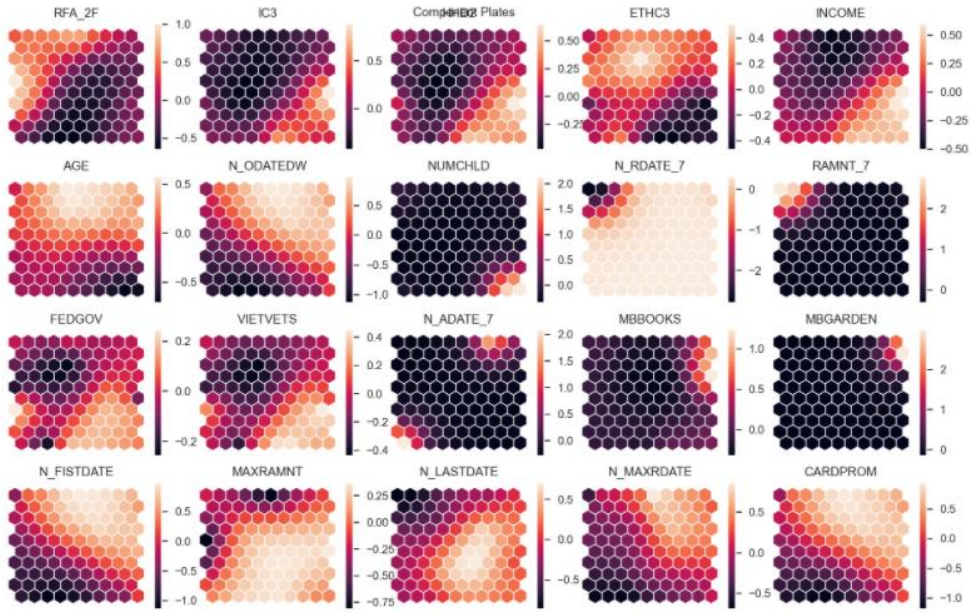


Figure 8.4. Summary K-Prototypes

doesn't need to be parametrized we decided to experiment the combination as it seems they in a way, complement each other.

The SOM solution above reflects an interesting distribution of the clustering variables along the grid units. There are some remarks we can make observing these plates like the N_RDATE_7 and RAMNT_7, both features representing our selected promotion, complement each other and that N_ADATE_7 is redundant compared to them. MBBOOKS and MBGARDEN both from the same group don't really complement each other, the information each one has are, more or less, on the same area. We can also observe a lot of strong correlations between different group variables like INCOME, which is correlated with NUMCHLD, N_RDATE_7, FEDGOV, VIETVETS, MAXRAMNT and N_LASTDATE, which could be useful for the analysis. Then we proceeded to the Hierarchical Clustering, also with the ward's linkage method.

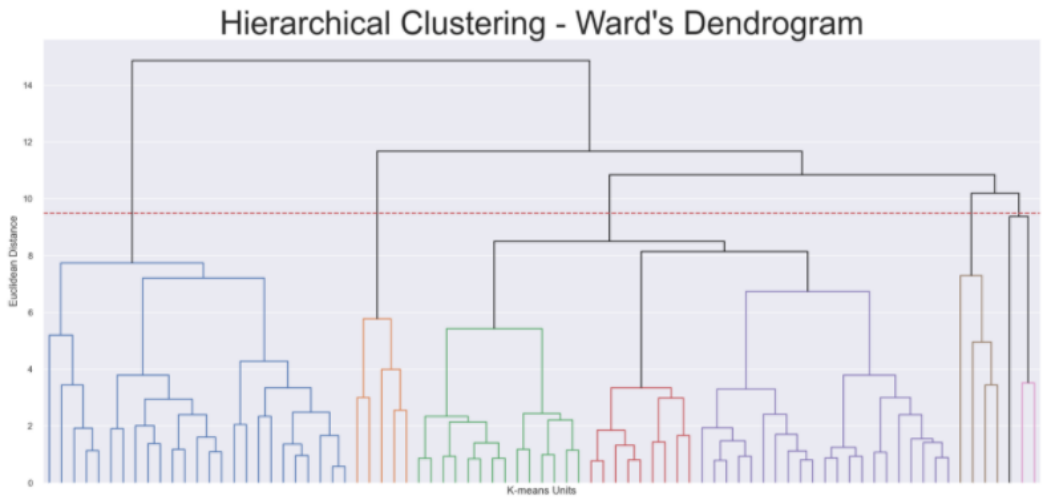


Figure 8.6. Hierarchical Clustering (Ward's Dendrogram)

As observed the dendrogram is suggesting a 5 cluster partition. We then proceeded to use some methods to evaluate the number of clusters before moving on.

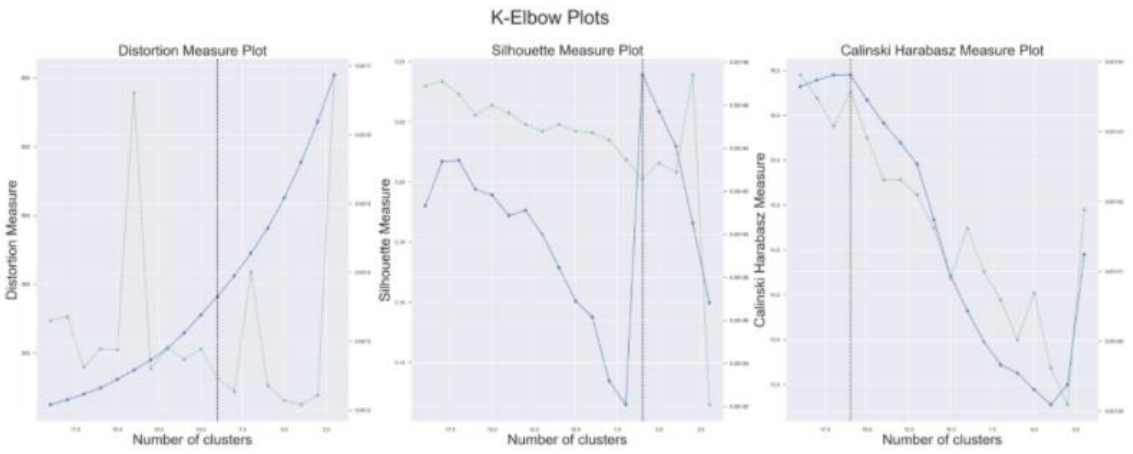


Figure 8.7. K-Elbow plots

We can observe that there isn't any consensus and some of the suggestions are completely out of what would be considered a acceptable range, so the Silhouette plots were the next option to analyse the situation.

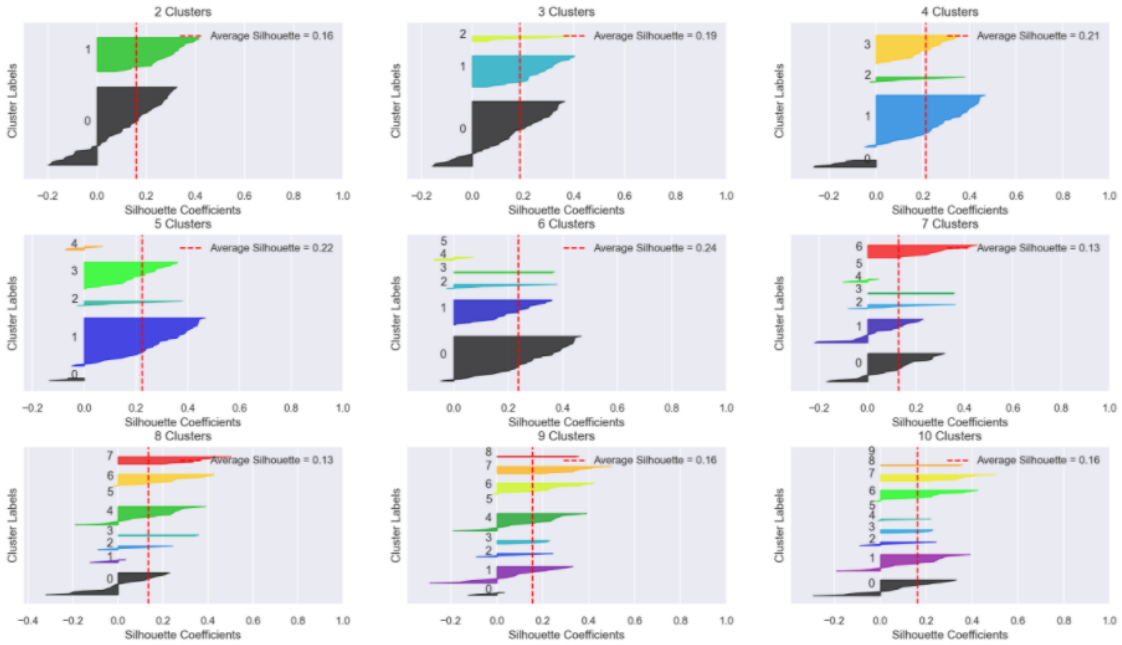


Figure 8.8. Clustering Silhouette plots

From observing these Silhouette plots, we can't conclude much based on the cluster sizes so we just take the into consideration the solution with better coefficient, which is the 6 clusters solution.

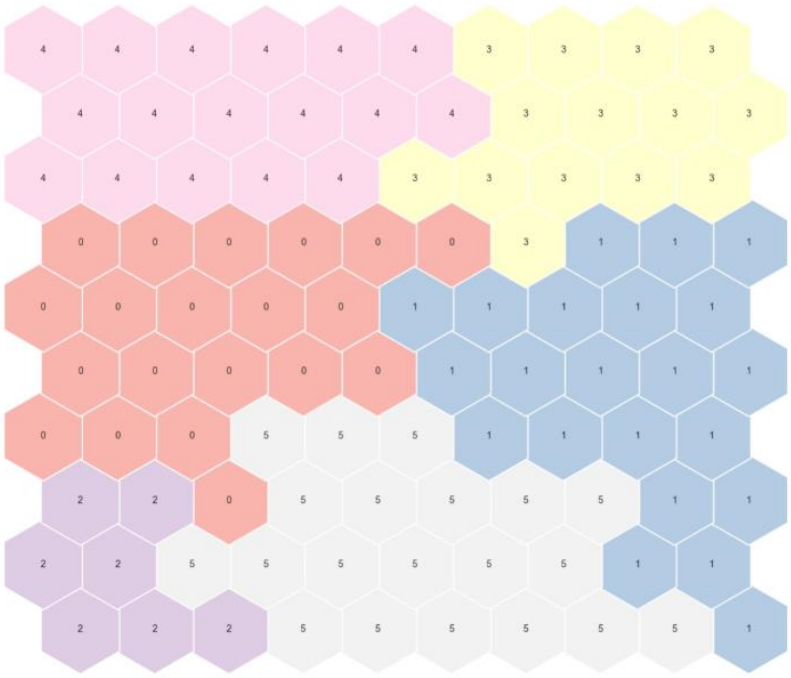


Figure 8.9. HIT Map View

From this hit map view, we can see there are some observations that could be assigned to the wrong cluster, which explains why it didn't score as high as other methods.

8.3. SOM & K-means

As previously with SOM got some promising results, it was decided to try out the SOM & K-means approach.

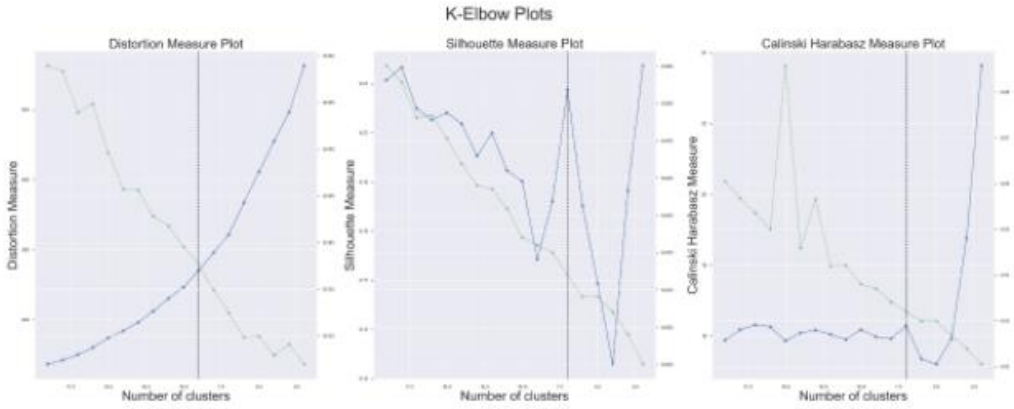


Figure 8.10. K-Elbow plots

As usual the K-elbow plots don't have a consensus on the optimal number of clusters since the Distortion plot suggests approximately 9 clusters and the Silhouette and the Calinski Harabasz suggest 7.

To analyse further we'll check the Silhouette plots having in mind 8 is still our preferred solution.

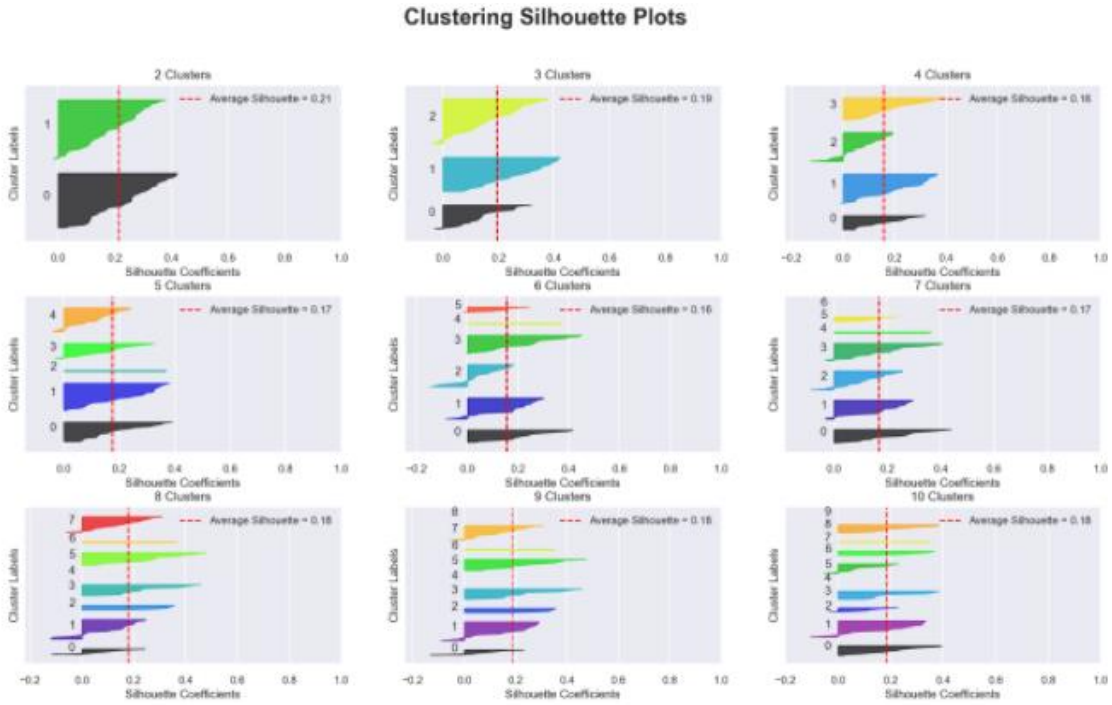


Figure 8.11. Clustering Silhouette plots

Comparing the average silhouette values, it is observed that the 9 clusters have a better score than the 6 clusters and the same value as the 8 clusters solution. Comparing the plots we observe that there isn't a significative difference between them, so for sake of consistency we're going to proceed with 8 clusters.

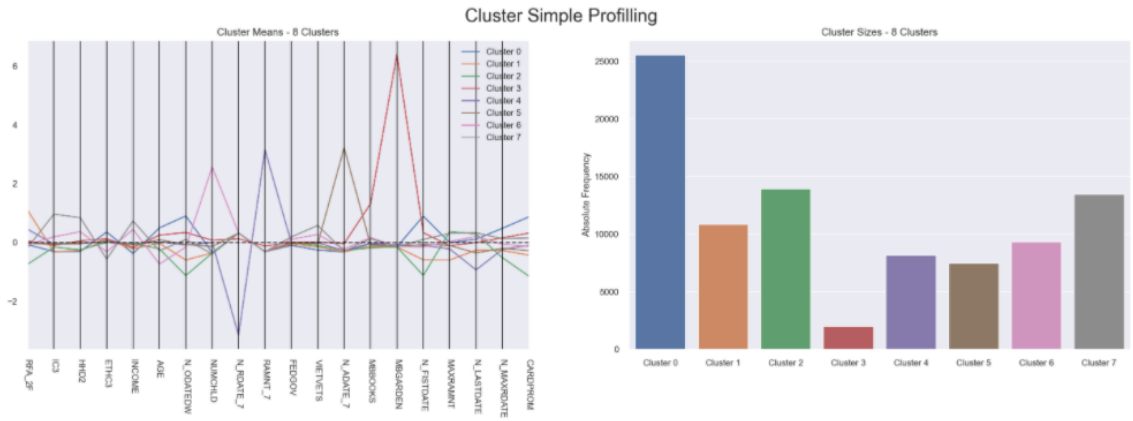


Figure 8.12. SOM & K-means Cluster Simple Profiling

Observing the cluster distribution after applying the algorithm, it can be remarked that 6 of the clusters are very well balanced and consistent between them, the other 2 clusters, cluster 0 and 3, are respectively the ones the highest and lower number of donors.

Is interesting to note that cluster 0 represents N_ODATEDW, N_FISTDATE and CARDPROM which makes sense has the total number of promotions received increase the likelihood of a first donation happen. Also is observable that cluster 4, 5 and 6 (all 3 with a balanced number of observations) individually, represent almost entirely features RAMNT_7, N_ADATE_7 and NUMCHLD in that order.

8.4. Gaussian Mixture Model

GMM was also one of the options we've decided to explore, as is very similar to k-means but it has the advantage of accounting the variance which works better when the data doesn't have a circular shape and performs a soft classification, which provides the probability that a given data point belongs to each of the possible clusters.

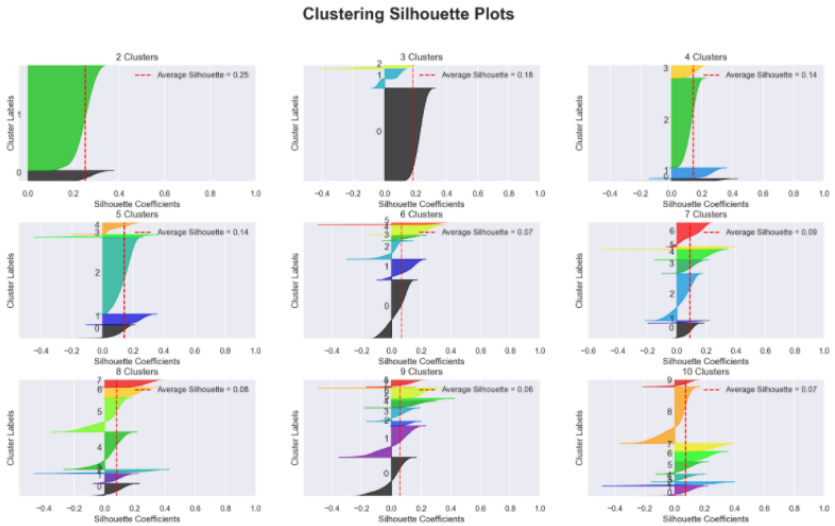


Figure 8.13. GMM Silhouette plots

We can observe that 2 clusters has by far the highest Silhouette score but we don't consider that a feasible solution so we'll apply an AIC/BIC Elbow test to evaluate our options.

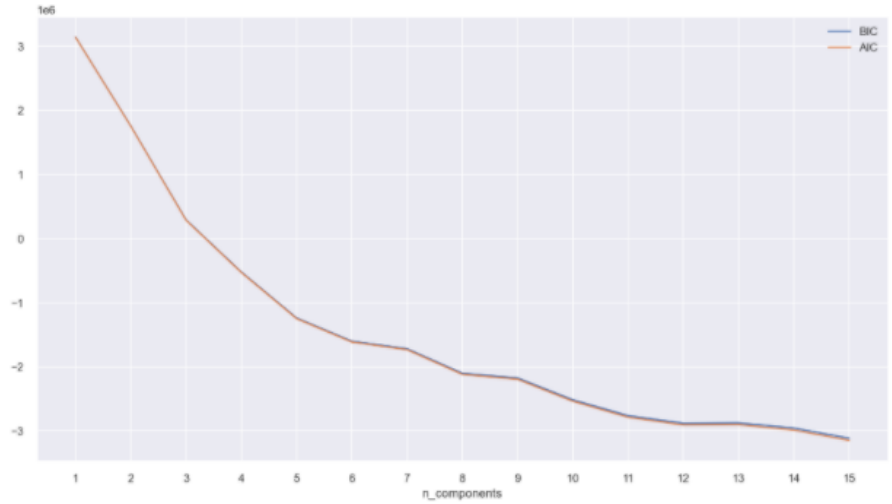


Figure 8.14. AIC/BIC Elbow plot

We can consider 6 the optimum number of clusters, but visually any value between 5 and 8 could be taken into consideration.

We clustered the data and evaluated the r2 for comparing with the other methods. The cluster solution with r2 was 0.9998 which looked promising and gave this method basis of worth to proceed with comparing.

8.5. Accuracy Metrics

Above although we only displayed the table with our selected method, as we stated the comparation was on the method and number of clusters basis. Here we'll share all the different tests made, taking all the try outs into consideration.

N = 5	KM	KM+HC	HC+SOM	KM+SOM	GMM
Silhouette_Score	0.123635	0.211798	0.223448	0.192365	0.0968578
Calinski_Harabasz_Score	9844.72	21.3917	12.9446	12.8738	5323.15
Davies_Bouldin_Score	2.08232	1.48007	1.38095	1.38171	2.716
N = 6					
Silhouette_Score	0.130104	0.237995	0.235647	0.196629	0.0895409
Calinski_Harabasz_Score	9213.6	22.1867	13.1287	13.6484	6309.01
Davies_Bouldin_Score	1.97074	1.33865	1.12042	1.19123	2.32194
N = 7					
Silhouette_Score	0.100278	0.237863	0.125975	0.180871	0.0874229
Calinski_Harabasz_Score	8293.71	20.0488	13.2217	13.7969	5380.57
Davies_Bouldin_Score	2.28326	1.1734	1.23103	1.14436	2.92833
N = 8					
Silhouette_Score	0.112416	0.189642	0.13388	0.157847	0.078439
Calinski_Harabasz_Score	7733.77	18.4646	13.4774	13.1749	6182.16
Davies_Bouldin_Score	2.05538	1.27619	1.20845	1.29503	2.47229
N = 9					
Silhouette_Score	0.101084	0.186222	0.155074	0.175582	0.0525386
Calinski_Harabasz_Score	7549.5	17.2327	13.8247	14.6557	5138.1
Davies_Bouldin_Score	2.13662	1.28227	1.19662	1.17261	2.99488
N = 10					
Silhouette_Score	0.0995755	0.190079	0.160387	0.190558	0.0748834
Calinski_Harabasz_Score	7106.32	16.3174	14.2117	14.9236	4058.49
Davies_Bouldin_Score	2.18049	1.18085	1.08338	1.02733	3.19851

Figure 8.15. Accuracy metrics table

As observable both the Calinski Harabasz and Silhouette highest scores are associated to the 5 and 6 clusters solution and although the Davies Bouldin top score is for n = 10, that value isn't a viable option. So, our focus turned to the 6 and 5 cluster solution.

Our final option tended for the 6 clusters solution as on our try outs that solution performed well and consistently.