Project Report for

Data Mining Group Project

Master's in Data Science and Advanced Analytics at NOVA IMS, Lisbon

Group H

Group members:
Henrique Vaz
Philipp Metzger

Link to GitHub repository:
https://github.com/ph1001/NOVA-Data-Mining-Project

Abstract

This report describes the stepwise creation of a customer segmentation on a dataset provided by Paralyzed Veterans of America (PVA). In chapter I, the information necessary for understanding the problem at hand is presented. In chapter II, some background knowledge on clustering is conveyed. In chapter III, the methodology of this project is described and in chapter IV, the results are presented and discussed. Finally, in chapter V, conclusions from the findings are drawn and presented.

I. Introduction

The goal of this project is to develop a customer segmentation based on a dataset provided by the non-profit organization Paralyzed Veterans of America (PVA). PVA provides programs and services for US veterans with spinal cord injuries or disease. The dataset contains information on individuals that have donated to PVA and that are classified as "Lapsed" donors[1], meaning that they made their last donation to PVA 13 to 24 months ago.

II. Background

In chapter 1 of [1], the goal of "clustering" is described as the discovery of groups of similar examples within the data. The customer segmentation in this project was done by clustering the observations contained in the dataset provided by PVA, resulting in subgroups of similar observations, to which then through interpretation of the characteristics of the different clusters, different marketing approaches were assigned.

In chapter 9.1 of [1], the intuition of clustering is described as the effort to find subgroups in the dataset at hand, such that the inter-point distances between points of the same cluster are small in comparison to their distances to points from different clusters.

There are several clustering techniques. As an example of a clustering algorithm, k-means, which is one of the most popular clustering algorithms, is described:

The goal of k-means is to partition the data into k subgroups, where every datapoint is allocated to one and only one subgroup. The process of k-means clustering can be summarised in the following way: k vectors from the same vector space as the points from the dataset are chosen in an appropriate manner. These k vectors are called centroids. For all points in the dataset, the nearest centroid is identified, using a distance measure such as the Euclidian distance, and the point is assigned to it. Then, all k centroids are updated in such a way that their new position represents the centre of the subgroup of points that were assigned to each one of them. Then, the assignment of every point to the nearest centroid is repeated and after that, the position of the k centroids is updated accordingly. These steps are repeated until a stopping criterion is fulfilled. The resulting k centroids represent the centres of k clusters in the dataset, which are comprised of the points that are nearer to the respective centroid than to any other one.

[1] In the dataset's metadata file also denoted as "Lapsing" donor

III. Methodology

III.1 Materials and Software

The materials used in this project are the dataset provided by PVA and the related metadata file, which describes that variables contained in the dataset. The dataset consists of 95412 observations and 475 variables.

The software that is used in order to complete the project is Python and more precisely Anaconda and Jupyter Notebook. In the latter, the code for this project is created.

In the following part, the steps conducted in our Jupyter notebook are described. The order of these descriptions is the same as the order of actions undertaken in the Jupyter notebook that has been handed in alongside with this report. The results and findings as well as their implications will be presented and discussed in the following chapter IV.

III.1 Imports and Organisation of Libraries and Data

As a first step, the necessary libraries were imported, and the dataset was loaded into main memory from the csv file provided.

Next, using the file "pva_metadata.txt", the features obtained from the imported dataset were split into metric and non-metric features.

III.2 Data Cleaning

In this step, again using the metadata file, the existence of cells was assessed, where spaces (" ") carry a meaning, such as for the feature 'MAILCODE', where a space means that the address of this individual "is OK", whereas the value "B" means that the address "is bad". Other features where something like this is the case are the features 'NOEXCH', 'RECINHSE', 'RECP3', 'RECPGVG', 'RECSWEEP' and 'MAJOR'. For these features, the spaces were replaced by a meaningful string, such as "Address is OK".

In the next step, the existence of duplicated observation was assessed and all remaining spaces in the dataset were replaced by NaN values[2]. It was also checked if the dataset contains any empty strings.

In the next part of the code, the percentage of missing values contained in each feature was assessed. Features that have more then 40 % missing values were discarded.

III.3 Data Transformation

In this section of the Jupyter notebook, all features containing time related information such as dates were transformed from their original string format to the datatype datetime.date[3]. For this purpose, using the metadata file, a list 'date_features' was defined, which contains all the features that are to be changed in this step. Each column represented by an element of this list, is then sent through a pipeline consisting of three functions that were defined in this step. Part of the functionality of this pipeline is to ensure that NaN values remain unchanged.

III.4 Further Data Cleaning

In this section of the code, the distributions of some features, where anomalies had caught the authors' eyes, were first visualised in order to then remove the unusual patterns that are likely to be errors stemming from a faulty process of data collection. Then, the unusual values were replaced by NaN values.

III.5 Feature Selection

As a first feature selection step, correlations between all metric features were assessed. Of feature pairs that were highly correlated, one was discarded, and one was kept in the dataset. Also, metric features that only contain a very small number of distinct values and thus carry little information for our analysis as well as features containing mostly zeros were discarded.

---

[2] NaN, short for "not a number", commonly denotes a missing value in a dataset.

[3] For the documentation of the Python library "datetime", see [2]

III.6 Further Feature Transformation

In this step, the date features whose transformation was described in III.3 were further transformed to integers representing their distance in days to the reference date stored in each observation's value of the variable 'ADATE_2'[4]. For better traceability, the resulting features were renamed in the following format: "<<original feature name>>>_rel_in_days".

III.7 Excluding features with very low Gini coefficient

In the following step, the Gini coefficient was assessed for every column in the dataset. In order to reduce dimensionality, all columns were discarded that had a Gini coefficient lower than $\frac{1}{2} * avg\_gini$, avg_gini being the average of all columns' Gini coefficients. The reasoning behind this step was that columns with a low Gini coefficient contain values that are quite similar to each other, thus containing little information and potential for clustering.

III.8. Filling the missing values

For filling the missing values in the dataset, two approaches were considered: The IQR method and KNN imputation. After testing both approaches, the authors came to the conclusion that IQR does not work so well, since even with very high values for the IQR multiplier, too many features were excluded. Because of this, the decision was made to use KNN imputation only.

III.9 Removal of outliers

For outlier removal, DBSCAN was chosen.This density-based algorithm, which can also be used for clustering, pools observations, that share densely populated spaces with a sufficient number of other observations and marks the rest of the observations as outliers. As the value for the first parameter minPts, the approximate square root of the number of observations was chosen and in order to find a good value for the second parameter Eps, a k-distance graph was created. This graph shows the sorted distances of the points in the dataset to their kth (k = minPts) neighbor. With its help it can be seen which distance needs to be chosen in order to exclude the points that are quite far away from their kth neighbor. These points are regarded as potential outliers and not included in the process of clustering, but instead later added to the final result, by allocating them to their nearest cluster.

III.10 Data normalisation

For the step of feature scaling, normalisation was chosen as the technique applied. All columns corresponding to metric features were normalised.

III.11 Perspectives Creation

At this point of the project there are more than 200 remaining features, which represent very high dimensionality. In order to reduce this dimensionality but still keep most of the information, the data was divided into perspectives.
Using only our metric features and the respective description, groups of similar features were created. The idea is to have different sets of features that contain the same type of information.

III.12 Clustering

For the clustering, two approaches were followed:

III.12.i Approach 1

The basic idea of this first approach is to perform clustering on a dataset filled with clustering labels. The latter generated by clustering each perspective as a single dataset.
This process can easily be defined and better understood analysing it step by step:

1. Define which perspective are going to be used.

2. Apply different clustering solutions to each perspective and keep a suitable one. After getting the right solution, add the labels to each point clustered. Following that save the characterizations of those clusters on a Data Frame that

[4] 'ADATE_2' represents the dates on which the most recent promotion was sent to each individual. This variable was chosen to be the reference time for each observation.

contains the perspectives' columns grouped by label using the mean value of each label (getting the centroid of each cluster).

3. Concatenate in a new Data Frame all the labels obtained in the different perspectives clustered. This results in a *n* x *p* dataset, where *n* represents the number of rows in the perspectives (that is the same for all) and *p* represents the number of perspectives used.

4. After finding this *n* x *p* dataset the final phase of this approach is reached. In this step the user should find a suitable clustering algorithm to apply on this dataset. Having this result, the goal is to use the characterizations mentioned in step 2 to characterize our final labels.

The reason for this approach to be taken is related with the high dimensionality of the data. A way to work around that issue is to try to break data into different parts and apply clustering separately.

III.12.ii Approach 2

Summary of approach 2:
K-means clustering on different feature sets, representing different perspectives on the data with subsequent combination of those perspectives and hierarchical clustering on the found centroids in order to determine the perspective combination that leads to the best result.

In the following, the second approach will be described in detail:

First, the feature sets representing the different perspectives on the data were assessed again, drastically reducing their number and their number of features per perspective. For this, the component planes from approach 1 were used to identify the most promising features from the features used in approach 1. As an example, the component planes for the perspective 'donor_info' are presented in Figure 1. From these four features, three were selected to be used in this approach: 'ODATEDW_rel_in_days', 'INCOME' and 'NUMPROM', since these three features show clear and distinguishable patterns in the component planes.

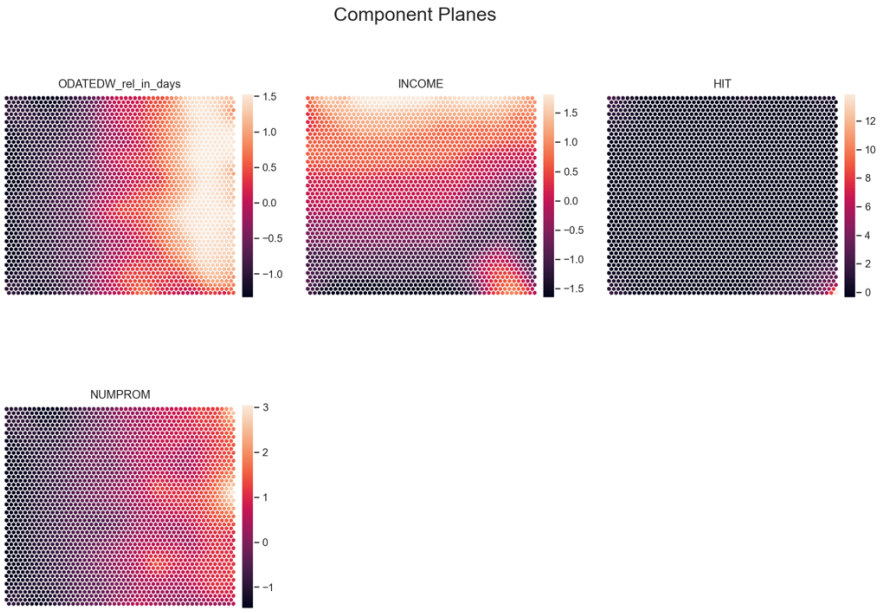

Figure 1: Component planes that were created in clustering approach 1

Then, on each of these reduced feature sets, k-means clustering was done iteratively with values of k ranging from 2 to to 9, each time saving the R2 score of the resulting clustering solution. Like this, the optimal value for k was identified for each perspective. Figure 2 shows the resulting R2 scores for each perspective and for varying values for k.

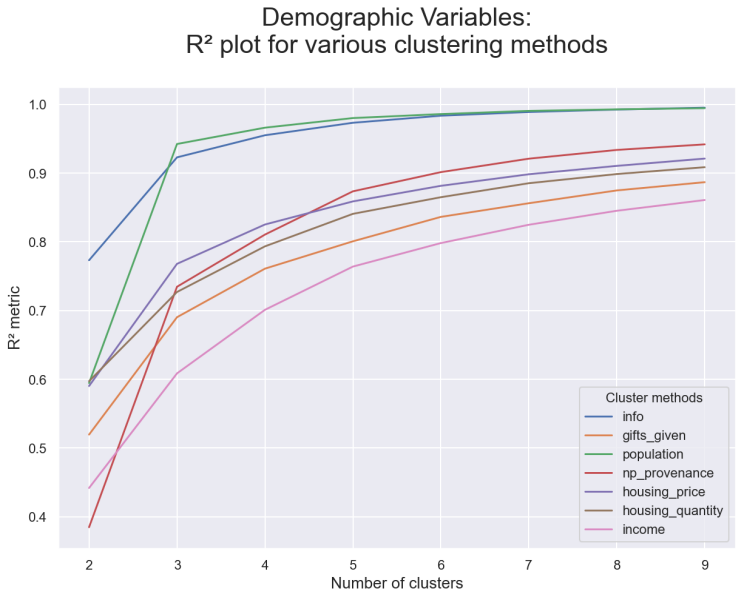

Figure 2: R2 scores for the different perspectives and varying values of k (numbers of clusters)

Then, with the optimal value for k for each perspective, k-means was run again once more on each perspective, yielding cluster labels for each perspective. In the next step, pairwise combinations of all perspectives were created and the centroids for each of the the resulting cluster label pairs were computed. Each of the resulting arrays of centroids were then the basis for hierarchical clustering, with the purpose of assessing which of the cluster pair combinations could be combined into merged, larger clusters. For this 'AgglommerativeClustering' was used on each result of each pair of perspectives and then a dendrogram for each fitted instance was created, in order to visually assess the clustering solution.

The plan was then to choose the most promising few of these clustering solutions and their perspectives pairs in order to then assess, which was would be the best for the final clustering. But, taking into consideration all the result, the authors came to the conclusion to use only one feature perspective for the final clustering. The reasons for this and the results are further described in IV.11.ii. After the final clustering and interpretation described in IV.11.ii, the outliers whose removal is described in III.9 were each allocated to their closes cluster.

IV. Results and Discussion

IV.1 Results and discussion: Imports and Organisation of Libraries and Data

We identified 398 metric and 77 non-metric features in the dataset.

IV.2 Results and discussion: Data Cleaning

In this step it was found out, that no duplicated observations or empty strings exist in the dataset. Furthermore, 3011889 spaces were converted to NaN values.

67 metric and 30 non-metric features were discarded from the dataset, due to them having a percentage of missing values higher than 40 %.

IV.3 Results and discussion: Data Transformation

The result of this step is the updated dataset, where the features defined in the list 'date_features' have been changed to objects of the type datetime.date. This facilitated their further processing and enabled us to do calculations such as addition or subtraction of different date columns.

IV.4 Results and discussion: Further Data Cleaning

The variables with the names 'AGE90x', $x \in \{1, 2, 3, 4, 5, 6, 7\}$ serve as a good example for the process of removing values that are likely to be faulty. Figure 3 shows the distributions of these features before any values were removed. It is apparent, that there are values similar to zero, whose frequencies don't integrate well with the rest of the frequencies of these distributions. These values are indeed the value zero. After all of these zeros were replaces by NaN, the distributions

were checked again. The resulting distributions, visualised as histograms, are presented in Figure 4.

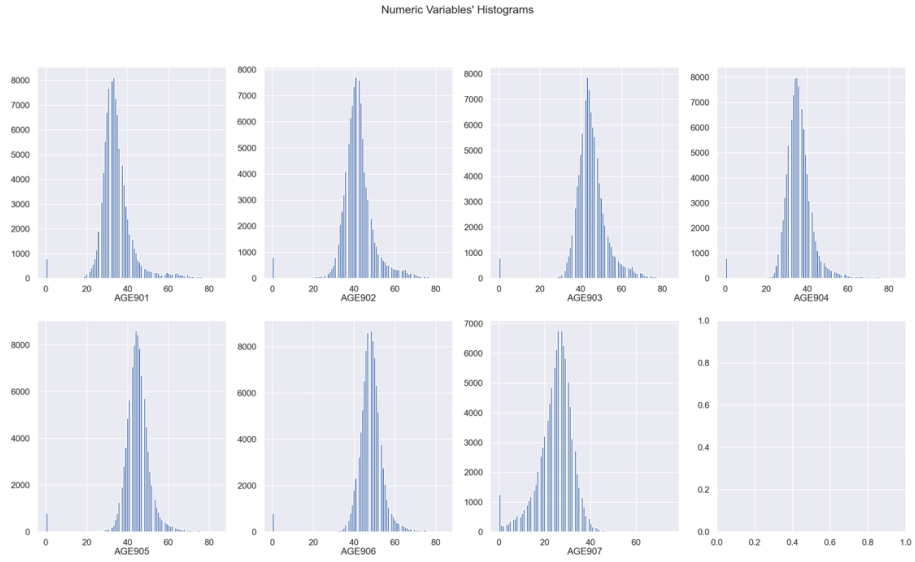

Figure 3: The distributions of the variables 'AGE90x', x ∈{1, 2, 3, 4, 5, 6, 7} before the removal of the value zero

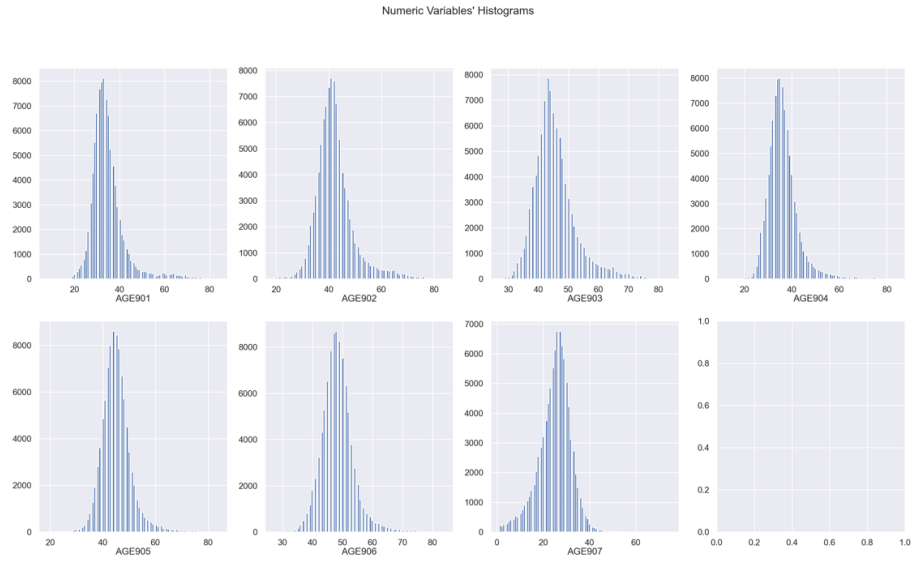

Figure 4: The distributions of the variables 'AGE90x', x ∈{1, 2, 3, 4, 5, 6, 7} after the removal of the value zero

IV.5 Results and discussion: Feature Selection

In this step, 92 features were discarded. The number of metric features was decreased from 331 to 249 and the number of non-metric features dropped from 47 to 37.

IV.6 Results and discussion: Further Feature Transformation

The values in days resulting from this step range from -122 days (= approximately -0.33 years) to 31928 days (= approximately 87 years). After the discovery that negative values exist, they were located in the dataset. They belonged to the feature 'DOB', or more precisely to the newly created feature 'DOB_rel_in_days'. There were five observations present in the dataset that had values equal to or smaller than zero in this column[5]. It was decided to remove these observations, since it is highly unlikely that any promotion was mailed to an individual that wasn't born yet or that was born on the day of the mailing.

[5] These observations have the following indices: 22984, 40565, 52252, 60753, and 94452.

IV.7 Results and discussion: Excluding features with very low Gini coefficient

In this step, 71 features were excluded that all have a Gini coefficient lower than 0.1956. The authors decided that dropping these features is adequate and helpful for the later task of making the final feature selection for clustering. The remaining numbers of features after this step were 249 metric and 178 non-metric features.

IV.8. Results and discussion: Filling the missing values

Since KNN imputation was used on the whole set of metric features, the computation time when running it is very long. Because of this, the authors decided to save a copy of the data that is the result of this step in a newly created csv file called 'donors_after_KNN_imputation_with_date_features.csv'. The resulting dataset contains no more NaN values in any of the metric features.

IV.9 Results and discussion: Removal of outliers

Figure 5 shows the k-distance graph created for this application. Just by looking at it, it can be seen that that there are approximately 10000 observations that are much further away from their kth neighbour than the other observations. These observations have an approximate distance of 400 to 500 to their kth neighbour. To be conservative, 500 was chosen in order to be sure to exclude most outliers. Running DBSCAN with minPts = 300 and Eps = 500 resulted in the removal of 7547 observation, which accounts for 7.91 % of the dataset's observations.

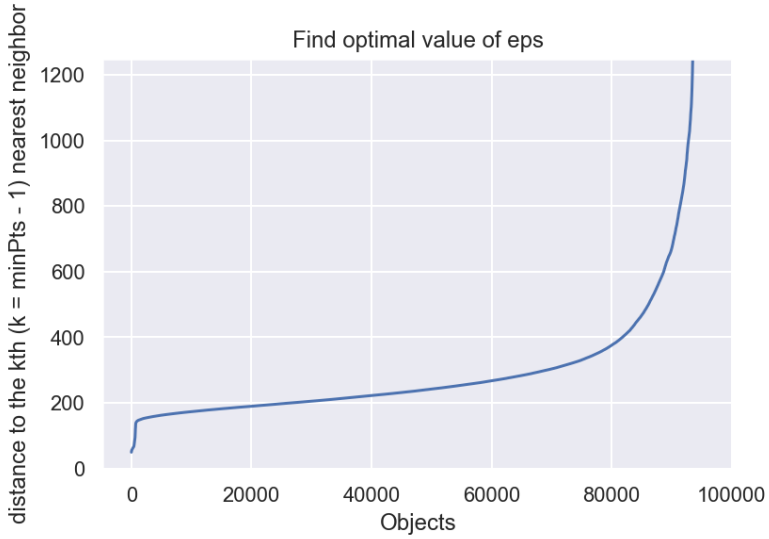

Figure 5: k-distance graph used for determining a good value for the DBSCAN parameter Eps, which was used for outlier detection

IV.10 Results and discussion: Data normalisation

As the result of this step, all metric features were normalised.

IV.11 Results and discussion: Perspectives Creation

As stated in III.7, different sets of features were generated. Each set of features represents a perspective/category. The result of this process is a total of 25 different perspectives, which can further be used for clustering.
The decision of using or not a specific perspective will depend not only on the average value of GINI Index of the perspective's features, but also on some intuition of what the authors consider is relevant for the result. This decision process led to a reduction from 25 to 15 perspectives to be clustered.
The perspectives that went on to be clustered further ahead were the following:

1. Donor Info: personal information about the donor.
2. Donor Gifts Given: information of gifts given by a donor.
3. NB[1] Population: information about population on the donor's nb.
4. NB Housing Price: information about house pricing in donor's nb.
5. NB Income: information about the incomes on donor's nb.
6. NB Job: information about donor's nb's people jobs.

7. NB Education: information about donor's nb's people education level.
8. NB Military Service: information about presence of military service in donor's nb.

IV.12 Results and discussion: Clustering

IV.12.i Results and discussion: Clustering Approach 1 - Description of the Results

It is important for the reader to keep in mind that this approach follows a single procedure and applies it to the different existing perspective. In the end it stores the results of each perspective's clustering solution in a final Data Frame composed only by labels.

Hierarchical Clustering on top of KMeans:

In this approach the first option to be taken was to apply hierarchical clustering on top of kmeans. For the kmeans algorithm the *n_clusters* chosen was 1000. This way we would end we 1000 centroids and further on apply hierarchical clustering on those values.
After running kmeans several Nan values appeared. In addition to this, the application of hierarchical clustering on those values wasn't either performant or suitable. Thus, this solution was discarded.

KMeans on top of SOMs:

In a second attempt, that intended to follow the same logic but with different techniques, KMeans was tested on the result of different *Self Organizing Maps* (SOM) results (units).
For the first and unique SOM created the chosen map size was 50x50, thus resulting in 2500 units as expected. This huge number of units allowed the user to have a good amount of data to cluster in further actions. Other relevant observation to point out is that the same number of epochs was chosen for the unfolding and finetuning phases, 100. The value represents a balance of good performance in short time and a good minimization of the quantization error.

Right after getting the result of SOM, KMeans is applied on top of that result. The values of *n_clusters* for KMeans tested ranged in the interval [3, 7]. Although the results were not stored, the authors still found out that the best results were shown with *n_clusters* = 6 based on the different r2 scores obtained for this same dataset. Adding to the fact that the r2 scores were higher, keeping a higher number of clusters in this phase will ensure more variance in the final labels' dataset.
After the k-means algorithm ran, the labels are added to each perspective's Data Frame resulting of a Data Frame that has the perspective's columns plus a column with a label assigned to each observation.
It's also important to store each perspective clustering description. For this, perspectives characterization Data Frames were created. These Data Frames are shown in Table 1 to Table 8.

label_donor_info	ODATEDW_rel_in_days	INCOME	HIT	NUMPROM
0	0.948415	-1.064547	-0.053114	0.890212
1	0.093702	0.657660	-0.108082	0.170011
2	1.221805	0.614783	-0.049823	1.100804
3	-0.862996	-0.963426	-0.203642	-0.820379
4	0.416597	-0.060402	3.529037	0.425168
5	-1.016564	0.713129	-0.093790	-0.981619

Table 1: Characterization of Donor Info perspective clustering solution

label_donor_gifts_given	RAMNTALL	NGIFTALL	MINRAMNT	MAXRAMNT	LASTGIFT	TIMELAG	AVGGIFT
0	-0.270285	-0.074628	-0.430099	-0.381641	-0.399264	-0.369715	-0.491745
1	-0.661403	-0.880916	1.308238	0.156902	0.340756	-0.469723	0.807033
2	1.280530	0.241033	-0.128602	1.308406	1.094414	0.101725	0.810855
3	1.153563	1.856939	-0.593872	-0.370650	-0.515136	-0.252327	-0.592330
4	1.093221	-0.565451	3.124801	3.973047	3.923472	0.298523	4.185117
5	-0.333410	-0.370647	-0.221190	-0.116752	-0.046454	1.429568	-0.150715

Table 2: Characterization of Gifts Given perspective clustering solution

label_nb_population	POP901	POP90C1	POP90C2	POP90C3
0	1.910426	-1.132283	-0.323972	1.597293
1	1.665667	0.794876	-0.429263	-0.570312
2	-0.377133	-1.202912	2.475314	-0.480547
3	-0.290377	-1.161609	-0.354443	1.651381
4	-0.381744	0.804994	-0.431359	-0.629828
5	2.579611	-1.185609	1.583834	0.199264

Table 3: Characterization of NB Population perspective clustering solution

label_nb_house_price	HV1	HVP3	HVP5	HUR1	HUR2	HUPA1	HUPA3	HUPA4	HUPA5	HUPA7	RP3
0	1.337050	1.517746	0.857603	-0.365365	1.026033	-0.403246	-0.508120	-0.324967	-0.433727	-0.474848	0.867614
1	-0.409693	-0.509485	-0.213373	0.361798	-0.749720	2.109352	-0.473369	0.002832	2.132063	-0.408206	-0.033136
2	-0.312038	-0.298381	-0.153430	-0.005099	-0.437008	-0.469711	1.823274	-0.201575	-0.385912	1.641331	-0.622367
3	-0.756256	-0.831134	-1.386595	-0.215328	-0.278219	-0.252557	-0.003044	0.603461	-0.136356	0.022925	-1.101491
4	0.825493	1.031279	0.641783	1.822652	-1.142178	0.751178	-0.465425	0.009705	0.475400	-0.435651	0.752980
5	-0.248139	-0.353880	0.477996	-0.346108	0.451839	-0.233725	-0.392510	-0.156292	-0.265599	-0.362534	0.414602

Table 4: Characterization of NB House Price perspective clustering solution

label_nb_income	HHD4	IC2	IC6	IC19
0	-0.212854	1.245555	-0.840018	0.928059
1	0.055858	-0.775551	0.724265	-0.888266
2	1.408817	1.130784	-1.063678	1.432294
3	-1.168247	-1.098665	1.770193	-0.942556
4	0.642584	-0.118910	-0.384068	-0.019993
5	-0.936583	-0.076002	-0.128683	-0.082682

Table 5: Characterization of NB Income perspective clustering solution

label_nb_job	EIC1	EIC2	EIC3	EIC4	EIC5	EIC6	EIC7	EIC8	EIC9	EIC10	...	EIC13	EIC14	EIC15
0	0.193928	4.347373	-0.041476	-0.610856	-0.086205	0.114884	-0.066168	-0.118523	-0.305984	-0.233747	...	-0.166372	0.179167	-0.150042
1	-0.323345	-0.146761	-0.145615	-0.055346	0.734703	0.416567	0.772289	0.017632	0.456511	0.328431	...	-0.186469	-0.339668	-0.031816
2	-0.332548	-0.164730	-0.467217	-0.207382	-0.299096	-0.088439	-0.047059	-0.189900	0.534007	-0.078756	...	0.771618	0.349314	0.891519
3	-0.065047	-0.113692	0.758483	-0.442927	-0.179563	-0.039633	-0.219938	0.527117	-0.027799	0.366001	...	-0.169881	-0.302490	-0.063805
4	0.619322	-0.143296	-0.033443	0.971671	-0.189116	-0.289026	-0.179100	-0.303654	-0.610330	-0.378095	...	-0.333731	-0.282586	-0.560556
5	-0.153740	-0.101860	-0.215277	-0.503771	-0.043810	0.055026	-0.307572	0.002779	-0.082065	-0.144570	...	0.163820	0.781532	0.066804

Table 6: Characterization of NB Job perspective clustering solution

label_nb_education	EC2	EC3	EC4	EC6	EC7	EC8	SEC1	SEC3	SEC4	SEC5
0	1.237925	1.100177	0.194074	-0.591162	-0.829621	-0.647224	-0.454889	-0.471082	0.387240	-0.400486
1	-0.539518	-0.548801	-0.224751	0.899195	0.355340	0.117769	0.252477	0.367182	0.294874	0.179628
2	-0.525255	-0.770265	-1.095389	0.191933	1.183650	1.185358	0.884068	-0.555182	-1.285103	2.731876
3	-0.037738	0.187382	0.932902	-0.115730	-0.547851	-0.514326	-0.105972	0.343024	0.384374	-0.285413
4	-0.274041	-0.051594	-0.002595	-0.210764	-0.137895	-0.152187	-0.382174	-0.724744	-1.075975	-0.225351
5	-0.804733	-1.131141	-1.265711	-0.038721	1.701508	1.726969	0.784922	0.777637	-0.089945	0.239723

Table 7: Characterization of NB Education perspective clustering solution

label_nb_military_service	AFC2	AFC3	AFC6	VC1	VC2	VC3	VC4
0	-0.133569	-0.102620	-0.293662	-0.433319	-0.585503	-0.227902	0.847762
1	-0.110596	-0.070633	-0.320268	1.122049	-0.445359	-0.749392	-0.109788
2	-0.147336	-0.111052	-0.336176	-0.093652	0.868477	-0.068271	-0.412112
3	6.457539	4.953319	2.520774	0.856976	-0.172931	-1.087599	2.146463
4	0.292014	0.168620	1.601405	0.277104	0.002642	-0.278217	0.522414
5	-0.173152	-0.116821	-0.076462	-0.910983	-0.056561	1.378350	-0.641656

Table 8: Characterization of NB Military Service perspective clustering solution

Whenever a single perspective has been put through SOM and KMeans, its resulting labels' column is added to the final Data Frame. This final Data Frame is described in III.8 a) step 3.
Having obtained the final Data Frame (Table 9), filled only with labels and having the same number of columns as number of perspectives, different clustering algorithms were directly applied on it.

	_donor_info	_donor_gifts_given	_nb_population	_nb_house_price	_nb_income	_nb_job	_nb_education	_nb_military_service
0	2	3	3	3	4	4	3	2
2	0	3	0	2	1	4	0	2
3	0	0	3	2	4	4	0	2
4	4	3	4	5	1	5	0	4
6	2	0	2	1	4	4	4	0
...	...	...	...	...	...	...	...	...
95400	1	1	4	2	1	4	0	5
95401	1	3	3	2	1	4	0	2
95403	5	1	4	0	2	0	5	4
95404	3	0	4	3	5	5	3	1
95405	2	3	4	0	2	2	1	1

Table 9: Final Data Frame with labels only

The first one that was tried was a KMeans. The number of clusters chosen to use in this phase was based on an inertia plot, using the elbow method. The inertia plot is shown in Figure 6.

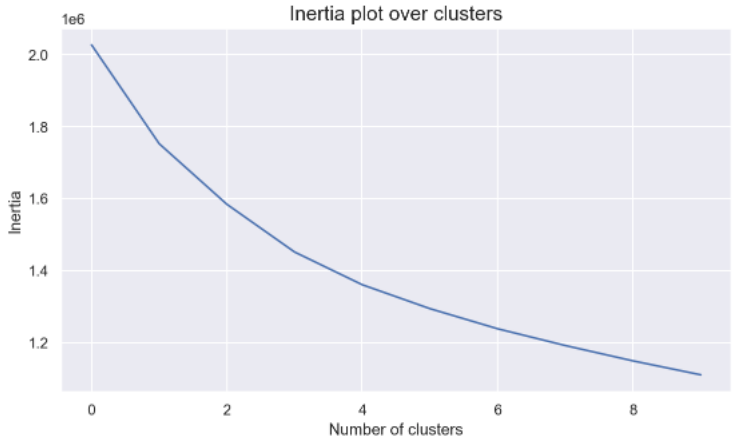

Figure 6: Inertia plot

From the plot above, the number of clusters chosen was 5. This number is right after the great decay in SSW and represents a decent number of clusters to describe in the end.
Using *n_clusters* = 5 the obtained r2 was 0.32, a value that's higher the ones in the tries with *n_clusters* < 5.

Besides Kmeans, other algorithms were tested. One of these other was Meanshift clustering. To initialize this algorithm the bandwidth was estimated, having as a result of that a value around 8. Using that bandwidth value, the result of the algorithm retrieved only one single cluster, which is not suitable.

Moreover, DBSCAN was also ran over the final Data Frame (Table 9). To find the optimal eps value to use in DBSCAN a k-distance graph was plotted, with parameter *n_neighbors* = 300 that is roughly the square-root of the number of observations of the data. This plot is shown in Figure 7.

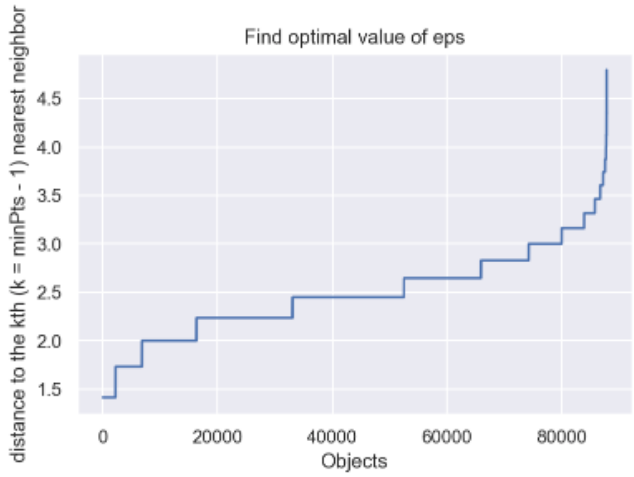

Figure 7: K-distance graph for finding the optimal value for Eps for DBSCAN

The value that was picked was 3. After that value the eps starts increasing too much and would not suit the purpose of the application. However, even picking the right value for the eps, the number of clusters retrieved with DBSCAN was also 1. Once again, this is not a solution that would fit the needs of this project.

Finally, the last attempt to cluster the final Data Frame (Table 9), was using hierarchical clustering. The type of linkage chosen was the only one available throughout all the process, single. Using that linkage, the r2 scores were retrieved, from different numbers of clusters combinations. The results for 2 to 9 clusters were all bellow 0.0001. Thus, this solution also ends up discarded.

To conclude this final clustering phase, the chosen model was the KMeans and those are the results to take into consideration.

labels	_donor_info	_donor_gifts_given	_nb_population	_nb_house_price	_nb_income	_nb_job	_nb_education	_nb_military_service
0	3.0	2.0	4.0	0.0	1.0	2.0	4.0	3.0
1	1.0	1.0	3.0	3.0	3.0	3.0	2.0	1.0
2	2.0	1.0	3.0	3.0	3.0	3.0	2.0	5.0
3	4.0	1.0	3.0	3.0	3.0	3.0	2.0	1.0
4	2.0	5.0	3.0	3.0	3.0	3.0	2.0	2.0

Table 10: Final Clustering Solution

IV12.ii.1 Results and discussion: Clustering Approach 2 - Description of the Results

The result of the feature perspectives selection process, conducted by assessing the component planes from approach 1 resulted in seven perspectives that are as follows:

- donor_info_features_2: 'ODATEDW_rel_in_days', 'INCOME', 'NUMPROM',
- gifts_given_features_2: 'RAMNTALL', 'NGIFTALL',
- population_features_2: 'POP90C1', 'POP90C2', 'POP90C3'
- provenance_features_2: 'ETH2', 'ETH5',
- housing_price_features_2: 'HVP3', 'HVP5', 'RP3',
- housing_quantity_features_2: 'HU4', 'HU5',
- income_features_2:' HHD4', 'IC6'.

As an example, the dendogram resulting from pairing donor_info_features_2 and gifts_given_features_2 is presented in Figure 8.

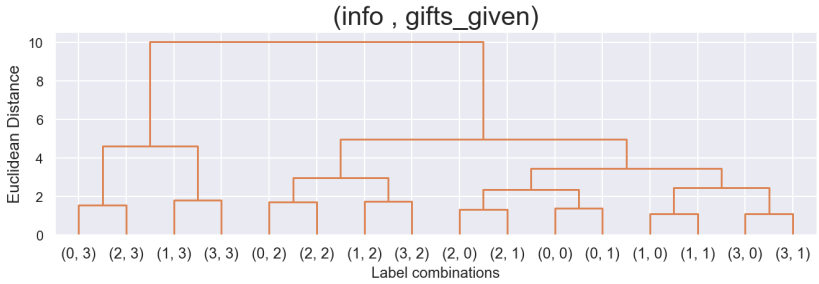

Figure 8: Dendogram for pairing the feature sets of donor_info_features_2 and gifts_given_features_2

The feature perspective pairs that the authors deemed as promising for further investigation are (donor_info_features_2, gifts_given_features_2), (gifts_given_features_2, housing_price_features_2), and (gifts_given_features_2, income_features_2). The analysis of the result yielded though, that none of the results of these feature perspective pairs merged by hierarchical clustering were satisfying. Instead, it was noticed, that the features in donor_info_features_2 yielded quite good results on their own. In Figure 9, the profiling of the clustering solution that was acquired by using only the features from 'donor_info_features_2' is presented. It is apparent that the four different clusters have distinct characteristics, which will be further described later on.

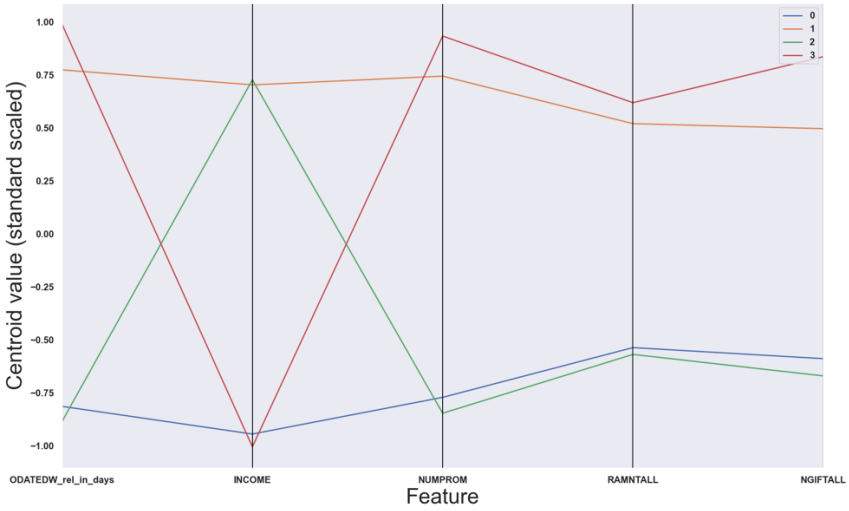

Figure 9: Profiling of the clustering on the features from 'donor_info_features_2'

In a next step, as a basis for interpretation, more features were included in the visualisation presented in Figure 9. These features are POP90C1, HVP3, HVP5, RP4, and EC7. These features, again, were chosen on the basis of the component planes from clustering approach 1. The result of the visualisation with these new features for interpretation included is presented in Figure 10. It can be seen that also for these features that were not used for this clustering, the clusters show a clear pattern.

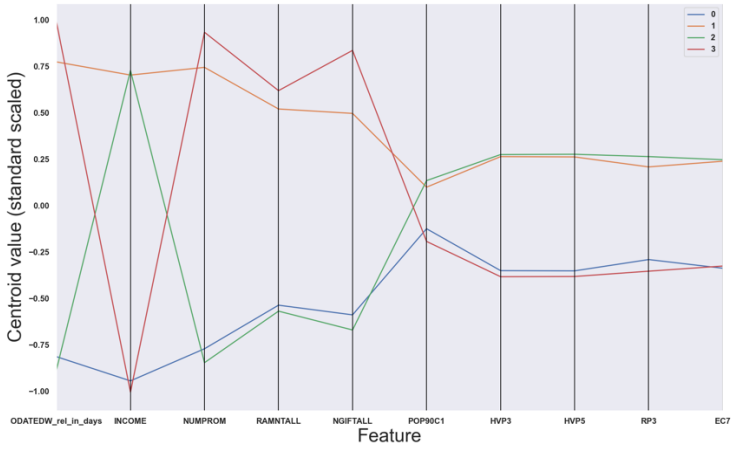

Figure 10: Visualisation of the clustering solution created with the features from 'donors_info_features_2', extended by more features for interpretation purposes

Next, the absolute frequencies per cluster were visualised. The result is presented in Figure 11. It can be seen that the clusters' frequencies are all in a similar range and no cluster has very little observations allocated to it.

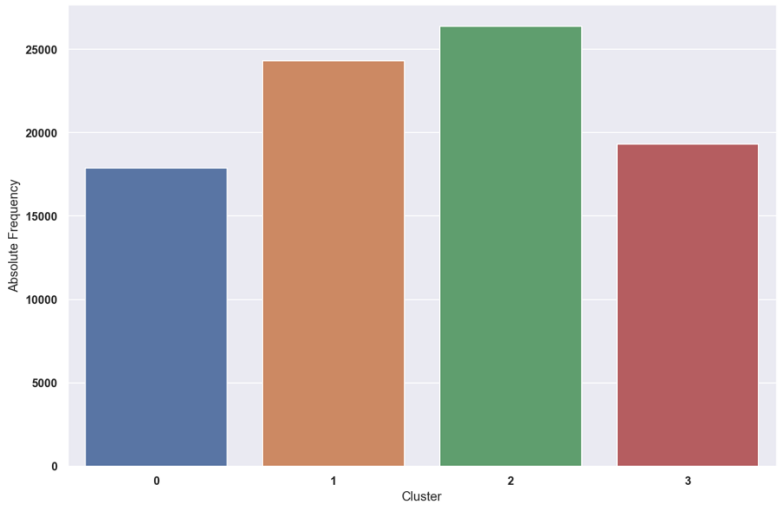

Figure 11: Visualisation of the absolute frequencies of observations in the clusters created with the features from 'donors_info_features_2'

Next, a categorical feature was included in the interpretational analysis. The features chosen is 'HOMEOWNR', which represents a flag stating whether or not it is knows that the respective observation owns a home or not. A visualisation of the absolute frequencies of this variable in the four clusters was created. The result is displayed in Figure 12. It is apparent, that in the clusters 0 and 4, significantly less individuals are known to own a home than in the clusters 1 and 2.

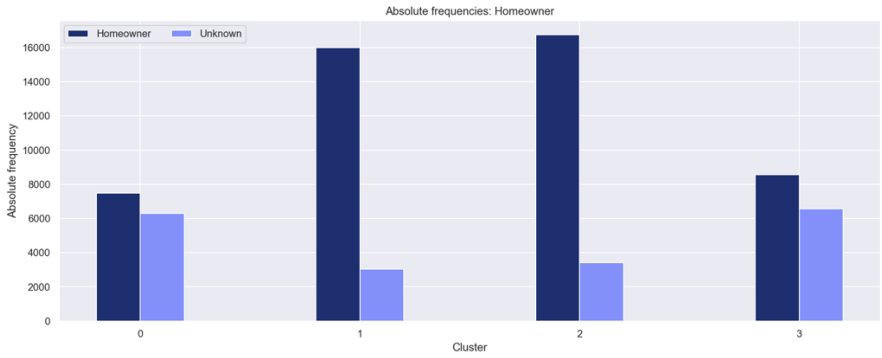

Figure 12: Absolute frequencies of the variable 'HOMEOWNR' for the clusterin solution obtained by using the features from 'donors_info_features_2'

IV.12.ii.2 Results and discussion: Clustering Approach 1 – Interpretation

The final clustering solution contemplates 5 different clusters. Since this solution is difficult to analyse graphically it is important to well describe the clusters.
This clusters will be described based on a tuple that contains its respective perspectives' labels. To describe this clusters, the reader must take into consideration Tables 1 to 8.

(donor_info, gifts_given, nb_population, nb_house_price, nb_income, nb_job, nb_education, nb_military_service)

Cluster 0: (3, 2, 4, 0, 1, 2, 4, 3)

Donors in this cluster are in a first analysis mainly describe by a low income however they appear to do regular donations. Still in this cluster, donors tend to live in highly populated neighbourhoods. Most of this donor's neighbourhood habitants work in health services and other professional services. It is relevant to say that this cluster has the highest percentage of males in active military service.

Cluster 1: (1, 1, 3, 3, 3, 3, 2, 1)

This group of donors have close to the highest income. The average amount gifted is also considerable. These donor's average home value is some of the greatest and this neighbourhood's habitants have the highest rents to pay.

Cluster 2: (2, 1, 3, 3, 3, 3, 2, 5)

These donors are the oldest of all and are also the one that have received promotions. The amount dispended by these donors is also considerable and seem to donate frequently.
Most of these donors live in the rural area and their houses value tends to be below the average and there are a lot of construction workers here. Finally, this group has the highest number of WW2 Vietnam veterans.

Cluster 3: (4, 1, 3, 3, 3, 3, 2, 1)

Donors in these group are very responsive to promotion offers and they are average donors and most of them hold bachelor's degrees. Might also be relevant to say that in this group there is the highest percentage of general Vietnam veterans.

Cluster 4: (2, 5, 3, 3, 3, 3, 2, 1)

These donors are also the oldest of all, although these ones have donated very few amounts and don't donate often. As in cluster 2, their houses' value is normally low. In this group of donors there is also high presence of Vietnam veterans.

In the cluster description above, the reader can infer that there are some perspectives that don't have great variance from cluster to cluster. More specifically, population, house price, income, job and education don't change values significantly between clusters. One possible approach to solve this issue would be do delete those perspectives and stick with the remaining. However, since another approach was tried in this project, the authors will keep the second approach's result and design a marketing strategy based on those results.

IV.12.ii.2 Results and discussion: Clustering Approach 2 – Interpretation

Interpreting Figure 10 and Figure 12, the following characterisations of and appropriate marketing strategies for the four clusters can be derived:

Cluster 0:

Characterisation:
The average individual in this cluster is younger than the average. It has a below-average household income and has not donated much in the past but has not received many promotions either. The ratio $\frac{money\ donated}{promotions\ received}$ is higher than in cluster 1 and 3, meaning that the effect of the promotions is higher in this cluster than in the other two mentioned. The average individual of this cluster lives in a not-so-urbanised area that has below-average home values and rents. Is not so likely to hold a bachelor's degree and is a little bit more likely to be homeowner than not to be one.

Marketing strategy:
Even though the average individual in this cluster has little money, the effect of promotions is higher than in clusters 1 and 3. **Increase the number of promotions sent per time span, but with caution!**

Cluster 1:

Characterisation:
The average individual in this cluster is older than the average. It has an above-average household income, has received a lot of promotions and has also donated more than the average person, but the ratio $\frac{money\ donated}{promotions\ received}$ is lower for this cluster than in clusters 0 and 2, meaning that the effect per promotion is not as high in this cluster as it is in two clusters mentioned. The average individual of this cluster lives in a rather urbanised area that has above-average home values and rents. Is likely to hold a bachelor's degree and is very likely to own a home.

Marketing strategy:
Keep the number of promotions sent steady or even consider reducing the amount of promotions sent, but with caution!

Cluster 2:

Characterisation:
The average individual in this cluster is younger than the average. It has an above-average household income, has not donated much in the past, but has received very little promotions as well. The ratio $\frac{money\ donated}{promotions\ received}$ is the highest in this cluster, meaning that promotions seem to be quite effective in comparison to the other clusters. The average individual of this cluster lives in rather urbanised area that has above average home values and rents. Is likely to hold a bachelor's degree and it is very likely to own a home.

Marketing strategy:
Send more promotions!

Cluster 3:

Characterisation:
The average individual in this cluster is older than the average. Even though it has a below-average household income, it donates significantly more than the other clusters' average individuals. The average individual of this cluster lives in a not-so-urbanised area that has below-average home values and rents. It is less likely to hold a bachelor's degree and is a little bit more likely to be homeowner than not to be one.

Marketing strategy:
Keep sending the same volume of promotions. They are likely to be rewarded with donations.

V. Conclusions

For the first approach there was not a relevant enough difference in the 5 clusters found. Probably, with some perspectives removal, the result could be more concrete and allow us to take solid conclusion. That not being the case, the authors will use the second approach defined and make that count as the final.

As for the second approach, the clustering solution presented in III.11.ii and IV.11.ii yield results that can be interpreted in an intuitive way. It can be concluded that it is possible to segement the dataset at hand into four cluster that carry a meaning in terms of their characterisation as well as their appropriate marketing strategy. The features used for this clustering were 'ODATEDW_rel_in_days', which is a transformation of the original variable 'ODATEDW', 'INCOME', 'NUMPROM' and 'RAMTALL' and the features used for the interpretation of the results obtained were 'POP90C1', 'HVP3', 'RP3' and 'EC7'. Marketing approaches were defined that range from the recommendation to cautiously decrease the number of promotions mailed in a time frame to the increase of the promotiuons mailed in order to leverage the effectivity of the promotions in certain clusters.

VI. References

[1] C. Bishop, Pattern recognition and machine learning, Springer, 2006.
[2] [Online]. Available: https://docs.python.org/3/library/datetime.html. [Accessed 30 Dec. 2020].