Final Report

Data Mining – 2020/2021

Universidade Nova de Lisboa – IMS

Master Degree Program in Data Science and Advanced Analytics

Catarina Natário Moreira m20201034
João Paulo Guerreiro Aredes m20200669
Maikel Sousa m20200735

Abstract

Clustering is an unsupervised process that creates clusters, a set of points, such that data points inside a cluster are close to each other and apart from data points in other clusters. There are many clustering techniques to group the data objects based on similarity, distance, and common neighbors. Clustering deals with finding a structure in a collection of unlabeled data points. The main goal of this project is to enhance clustering analysis and obtain the best results for the realization of the proposed problem. Firstly, we were provided with a dataset by the Paralyzed Veterans of America (PVA). PVA is a non-profit organization that provides programs and services for US veterans with spinal cord injuries or disease. Our task is to analyze a sample of the results of one of PVA's recent fundraising appeals. PVA asked us to develop a Customer Segmentation in such a way that it will be possible for them to better understand how their donors behave and identify the different segments of donors/potential donors within their database. Therefore, we define, describe, and explain the clusters that we choose.

In our project, we propose to iterate and optimize clustering results using various clustering algorithms and techniques. Specifically, we evaluate the K-Means available in the sklearn package.

This report involves the explanation of how we do the data pre-processing, data clustering using clustering algorithms, and data post-processing.

Keyword- Clustering, K-means

Contents

Chapter 1

1. Introduction

Clustering is the technique of grouping data so that the data in each group share similar characteristics and patterns. There are many techniques which are used to form clusters. Most commonly, hierarchical and partitional techniques are used to group similar data objects in one cluster. The clusters are formed by the similarity measures [1].

With the computer technology, programming language, software, and hardware tools improvements, the difficult tasks are easily solved. People are more interested in finding new way of performing the task in simple and easiest manner.

Thus, we deal with clustering in almost every aspect of daily life. Clustering is the way that we do research in several fields such as statistics, pattern recognition, machine learning and data mining. In the latter, clustering deals with very large data sets with different attributes associated with the data. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real life data mining problems.

As said before, most commonly, clustering methods are divided into two types: hierarchical and partitional clustering. To beyond these types, there are other different algorithms for finding clusters. On partitional clustering the goal is to create clusters that are coherent internally, and clearly different from each other. The elements within a cluster should be as similar as possible, and elements inside a cluster should not be similar, as much as possible, from elements in other clusters.

Hierarchical clustering builds a cluster hierarchy that can be represented as a tree of clusters. Each cluster can be represented as child, a parent and a sibling to other clusters. Hierarchical clustering is a method for clustering that has an intensive computational burden trying to find relevant hierarchies.

In our project, we used a clustering algorithm – k-means in conjunction with Hierarchical clustering to merge our clustering solutions.

As k-means is an unsupervised classification finding the appropriate number of clusters apriori to categorize the data, we came across with the difficult to know which the best number of clusters is to use.

We started by checking most of the clustering methods covered at class. We used the component planes of the SOM (Self-Organizing Maps) for several metrical features to assess their convergence by using a random initialization grid. After this assessment we found out that for the selected variables the K-means algorithm gave us good results and visually understandable centroids, which are the class representatives of each cluster. The algorithm is initialized with random centroid points. We found, by empirical evaluation, of both methods that the best possible number of clusters to understand PVA customers is 4.

In the project we evaluate the K-means clustering. The algorithm is initialized with random centroid points. We found, by empirical evaluation, that the best possible number of clusters for data is 4.
This report is organized by chapter 2 provides what we did in the pre-processing/data cleaning part, in the chapter 3 provides a brief review on the algorithm used. After this brief review, provides a brief explanation about we done. In chapter 3, we discuss the technique used in clustering evaluation and the results. Conclusion is presented in chapter 4.

Chapter 2

1. Data Cleaning

In this chapter, is detailed the conventions used for the cleaning and checking of potentially suspicious or out-of-range values on variables of this dataset.

We started by exploring and checking the variables included in the .txt file provided by the professors of the course. After that, we import the data and we started cleaning the data.

Firstly, we did some precious functions that helped us with exploration and validation of the data. After that, we created variables with lists of columns that related to each other. That is, we created several variables with several columns that had some kind of connection.

Posteriorly, since we are dealing with a big dataset, 475 columns, we went to check some details about each column of the dataframe.

We checked the Nan values that were present in the various variables and made different substitutions of them in the different variables created previously.

Column	Dtype	Len	Unique	Unique_Val	Nan Val	%NaN
RAMNT_5	float64	95412	9	[nan, 50.0, 12.0, 8.0, 31.0, 10.0, 5.0, 13.0, ...	95403	0.9999
RDATE_5	object	95412	5	[nan, 2016-04-01, 2017-03-01, 2016-07-01, 2018...	95403	0.9999
RECPGVG	object	95412	1	[nan, X]	95298	0.9988
SOLP3	object	95412	4	[nan, 00, 12, 01, 02]	95232	0.9981
RAMNT_3	float64	95412	29	[nan, 50.0, 10.0, 2.0, 14.0, 20.0, 8.0, 7.0, 2...	95170	0.9975
RDATE_3	object	95412	14	[nan, 2016-07-01, 2016-06-01, 2017-04-01, 2018...	95170	0.9975
RAMNT_4	float64	95412	32	[nan, 10.0, 1.0, 15.0, 5.0, 2.0, 25.0, 20.0, 9...	95131	0.9971
RDATE_4	object	95412	21	[nan, 2016-09-01, 2017-04-01, 2016-06-01, 2016...	95131	0.9971
MAJOR	object	95412	1	[nan, X]	95118	0.9969
PLATES	object	95412	1	[nan, Y]	94852	0.9941
RDATE_6	object	95412	17	[nan, 2016-03-01, 2015-11-01, 2015-10-01, 2016...	94636	0.9919
RAMNT_6	float64	95412	40	[nan, 15.0, 21.0, 11.0, 7.0, 5.0, 10.0, 18.0, ...	94636	0.9919
HOMEE	object	95412	1	[nan, Y]	94525	0.9907
CARDS	object	95412	1	[nan, Y]	94371	0.9891
CHILD03	object	95412	3	[nan, M, F, B]	94266	0.9880
MAILCODE	object	95412	1	[nan, B]	94013	0.9853
PVASTATE	object	95412	2	[nan, P, E]	93954	0.9847
KIDSTUFF	object	95412	1	[nan, Y]	93876	0.9839
CHILD07	object	95412	3	[nan, M, B, F]	93846	0.9836
RECSWEEP	object	95412	1	[nan, X]	93795	0.9831
CHILD12	object	95412	3	[nan, F, M, B]	93601	0.9810

RECP3	object	95412	1	[nan, X]	93395	0.9789
BOATS	object	95412	1	[nan, Y]	93384	0.9787
CHILD18	object	95412	3	[nan, M, F, B]	92565	0.9702
PHOTO	object	95412	1	[nan, Y]	90626	0.9498
COLLECT1	object	95412	1	[nan, Y]	90210	0.9455
SOLIH	object	95412	7	[nan, 12, 00, 02, 01, 04, 06, 03]	89212	0.9350
RECINHSE	object	95412	1	[nan, X]	88709	0.9297
FISHER	object	95412	1	[nan, Y]	88282	0.9253
RDATE_15	object	95412	16	[2015-05-01, nan, 2015-06-01, 2015-08-01, 2015...	88150	0.9239
RAMNT_15	float64	95412	76	[11.0, nan, 9.0, 20.0, 30.0, 5.0, 16.0, 10.0, ...	88150	0.9239
RAMNT_23	float64	95412	76	[11.0, nan, 7.0, 10.0, 15.0, 3.0, 25.0, 21.0, ...	87553	0.9176
RDATE_23	object	95412	17	[2014-08-01, nan, 2014-07-01, 2015-02-01, 2014...	87553	0.9176
CATLG	object	95412	1	[nan, Y]	87547	0.9176
RDATE_20	object	95412	10	[nan, 2014-11-01, 2014-12-01, 2015-07-01, 2015...	87524	0.9173
RAMNT_20	float64	95412	69	[nan, 6.0, 16.0, 4.0, 11.0, 7.0, 25.0, 10.0, 3...	87524	0.9173
CRAFTS	object	95412	1	[nan, Y]	87236	0.9143
BIBLE	object	95412	1	[nan, Y]	86541	0.9070
RAMNT_7	float64	95412	75	[nan, 5.0, 20.0, 15.0, 23.0, 25.0, 50.0, 10.0,...	86517	0.9068
RDATE_7	object	95412	9	[nan, 2016-03-01, 2016-02-01, 2016-01-01, 2016...	86517	0.9068
RAMNT_17	float64	95412	74	[11.0, nan, 15.0, 10.0, 17.0, 20.0, 100.0, 25....	86011	0.9015
RDATE_17	object	95412	11	[2015-03-01, nan, 2015-02-01, 2015-05-01, 2015...	86011	0.9015
RDATE_21	object	95412	12	[nan, 2014-11-01, 2014-12-01, 2014-10-01, 2015...	85899	0.9003
RAMNT_21	float64	95412	77	[nan, 11.0, 22.0, 20.0, 15.0, 5.0, 10.0, 18.0,...	85899	0.9003

VETERANS	object	95412	1	[nan, Y]	84986	0.8907
RDATE_10	object	95412	8	[2015-12-01, nan, 2015-11-01, 2016-01-01, 2016...	84951	0.8904
RAMNT_10	float64	95412	92	[10.0, nan, 15.0, 9.0, 20.0, 30.0, 27.0, 5.0, ...	84951	0.8904
PCOWNERS	object	95412	1	[nan, Y]	84931	0.8902
WALKER	object	95412	1	[nan, Y]	84911	0.8899
RDATE_13	object	95412	14	[nan, 2015-08-01, 2015-07-01, 2015-09-01, 2015...	83162	0.8716

Figure 1 – Table with variables with Nan values

So, we did:

- All the columns with the 'RAMNT' subkey as a name are related with the dollar amount that the people donated to a specific campaign. Those values will be replaced for 0 and are stored in the previously defined list: *'money_received_gift_features'*.
- All the columns with the 'RFA' subkey as a name are related with recency and all nan values will be replaced by 0. Those values are stored in: *'rfa_promotion_features'*.
- All the columns with the 'RDATE' subkey as a name are related with the date the person received the gift and all nan values will be replaced by 0. Those values are stored in: *'date_received_gift_features'*.

- All the columns with the 'ADATE' subkey as a name are related with the date the gift was sent and all nan values will be replaced by the mode first and then by 0 in those cases where NaN values are the ones that repeat the most. Those values are stored in: *'dates_promotion_features'*.
- Replacing all the interest features that are mostly binary. Those values are stored in: *'interests_features'*.

- Replacing all the past email features that are numeric, if the person has not responded to past offers then are assigned to 0. Those values are stored in: *'pastmail_offers_features'*.

Then, we checked again for the existence of NAN values. Although we have improved the number of Nan values, we still had a few. For example, we found, then, that the *basic personal features* list had values that contain discrepancies.

We found, then, that the *basic personal features* list has values that contain discrepancies.

Column	Dtype	Len	Unique	Unique_Val	Nan Val	%NaN
OSOURCE	object	95412	895	[GRI, BOA, AMH, BRY, nan, CWR, DRK, NWN, LIS, ...	928	0.0097
MAILCODE	object	95412	1	[nan, B]	94013	0.9853
PVASTATE	object	95412	2	[nan, P, E]	93954	0.9847
NOEXCH	object	95412	5	[0, 1, X, 0, 1, nan]	7	0.0001
RECINHSE	object	95412	1	[nan, X]	88709	0.9297
RECP3	object	95412	1	[nan, X]	93395	0.9789
RECPGVG	object	95412	1	[nan, X]	95298	0.9988
RECSWEEP	object	95412	1	[nan, X]	93795	0.9831
DOMAIN	object	95412	16	[T2, S1, R2, S2, T1, R3, U1, C2, C1, U3, nan, ...	2316	0.0243
HOMEOWNR	object	95412	2	[nan, H, U]	22228	0.2330
CHILD03	object	95412	3	[nan, M, F, B]	94266	0.9880
CHILD07	object	95412	3	[nan, M, B, F]	93846	0.9836
CHILD12	object	95412	3	[nan, F, M, B]	93601	0.9810
CHILD18	object	95412	3	[nan, M, F, B]	92565	0.9702
GENDER	object	95412	6	[F, M, nan, C, U, J, A]	2957	0.0310
WEALTH1	float64	95412	10	[nan, 9.0, 1.0, 4.0, 2.0, 6.0, 0.0, 5.0, 8.0, ...	44732	0.4688

Figure 2 – Table with *basic personal features* containing Nan values

Therefore, we made some replacements:

- In the 'OSOURCE' column, we replaced these values with the mode of the values in this column.
- In the ‘MAILCODE’ column, we replaced these values by 0 and we replaced ‘B’ values by 1.
- In the ‘PVASTATE’ column, we replaced the Nan values by 0.
- In the ‘NOEXCH’ column, we replaced it by 0 and ‘1’ or ‘0’ or ‘X’ VALUES BY 0.
- In those columns: 'RECINHSE', 'RECP3', 'RECPGVG', 'RECSWEEP', these values are replaced by 0 and ‘X’ values by 1.
- In the child columns ('CHILD03', 'CHILD07', 'CHILD12', 'CHILD18') the Nan values are replaced by 0.
- In the ‘GENDER’ column, we replaced the Nan, vales, the ‘C’ and ‘A’ values by ‘U’.
- In the ‘MAILCODE’ column, we replaced Nan values by G. This ‘G’ means that is good.
- In the ‘WEALTH1’ column, we replaced these values with the mode of the values in this column.
- In the ‘HOMEOWNR’column, we replaced these values by U. This ‘U’ means that is unknown.

After that, we verify that we still have some Nan values. So, we did some manual replacements taking as a reference the *Table 4*:

- In the ‘SOLIH’ and ‘SOLP3’ columns, we replaced these values by 13 since those can be solicited whenever.
- In the ‘MAJOR’ and ‘PEPSTRFL’ columns, we replaced these values by 0 and we replaced ‘X’ values by 1.
- In the ‘NUMCHLD’ column, we replaced the Nan values by 0.
- In the ‘GEOCODE’ column, we replaced the Nan values by 0.

After that, we verify again the Nan values.

Column	Dtype	Len	Unique	Unique_Val	Nan Val	%NaN
WEALTH2	float64	95412	10	[5.0, 9.0, 1.0, 0.0, nan, 3.0, 2.0, 6.0, 8.0, ...	43823	0.4593
DOB	object	95412	847	[1957-12-01, 1972-02-01, nan, 1948-01-01, 1940...	23883	0.2503
INCOME	float64	95412	7	[nan, 6.0, 3.0, 1.0, 4.0, 2.0, 7.0, 5.0]	21286	0.2231
DATASRCE	object	95412	3	[nan, 3, 1, 2]	21280	0.2230
NEXTDATE	object	95412	188	[2010-03-01, 2015-04-01, 2011-01-01, 2007-11-0...	9973	0.1045
TIMELAG	float64	95412	68	[4.0, 18.0, 12.0, 9.0, 14.0, 6.0, 8.0, 7.0, na...	9973	0.1045
DOMAIN	object	95412	16	[T2, S1, R2, S2, T1, R3, U1, C2, C1, U3, nan, ...	2316	0.0243
GEOCODE2	object	95412	4	[C, A, D, B, nan]	319	0.0033
MSA	float64	95412	298	[0.0, 4480.0, 9340.0, 5000.0, 2030.0, 3960.0, ...	132	0.0014
ADI	float64	95412	204	[177.0, 13.0, 281.0, 67.0, 127.0, 185.0, 91.0,...	132	0.0014
DMA	float64	95412	206	[682.0, 803.0, 518.0, 862.0, 528.0, 691.0, 509...	132	0.0014
FISTDATE	object	95412	176	[2009-11-01, 2013-10-01, 2010-01-01, 2007-02-0...	2	0.0000

Figure 3 - Table with variables with Nan values after replacing some of Nan values

After that, we replaced the values of the columns in the table above by the modes. We found that Nan values were no longer present.

After eliminating all Nan values, we started with exploring the different variables and we took group of features and checked for weird values, wrong records an outliers. After that, we tried to fix them. An example was that many of the variables has a percentage greater than 100%. So, we create a function to correct this.

So, let's check out what was said.

Starting with the variable 'POP90C1', 'POP90C2', 'POP90C3', we find that the percentage is not exactly 100%. As we can see in the graph below:

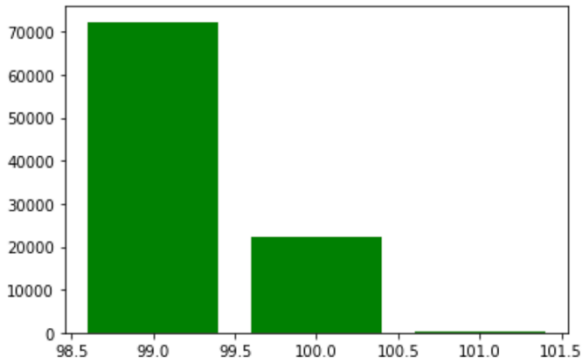

Figure 4- Percentage of the variables 'POP90C1', 'POP90C2' and 'POP90C3'

After fixing those values, we have percentages between 0 and 100% and, therefore, we can now view the histogram and boxplot of the variables.

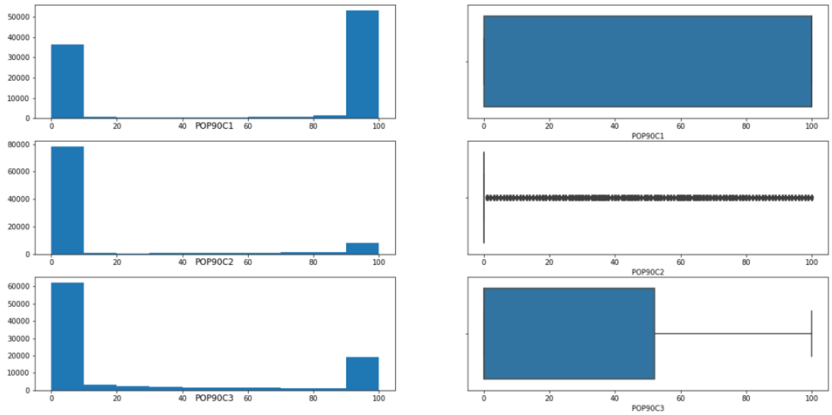

Figure 5 – Histogram and boxplot of variables 'POP90C1', 'POP90C2' and 'POP90C3'

We repeat the same process for the ‘POP90C4’ and ‘POP90C5’variables. After checking the percentages (if they do not have a values different than to 100%), we have:

```
100.0    94613
0.0        799
dtype: int64
```

Figure 6 – Percentage of the ‘POP90C4’ and ‘POP90C5’ variables after fixed

And as we can see above, we have records that show 0% in terms of male/female proportion in the donor's neighborhood. As they are not meaningful, we drop them. Therefore, we have a histogram and a boxplot like this:

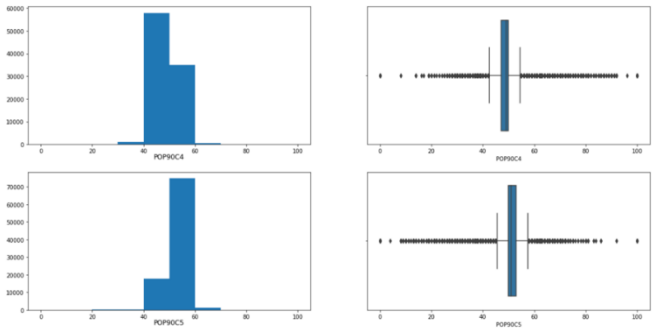

Figure 7 - Histogram and boxplot of variables ‘POP90C4’ and ‘POP90C5’

We then proceeded to the study of the variable 'ETHx', where x is a value between 1 and 16.

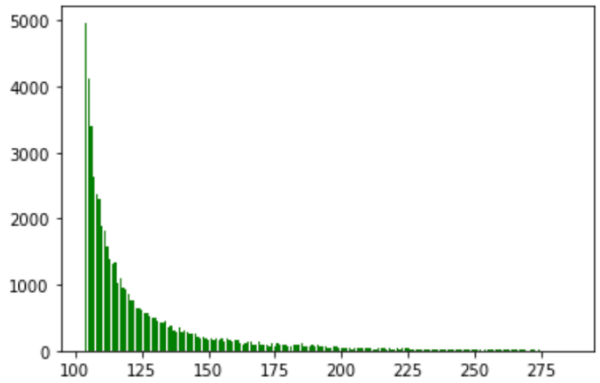

Figure 8 - Percentage of the variables ‘ETHx’, where x is between 1 and 12

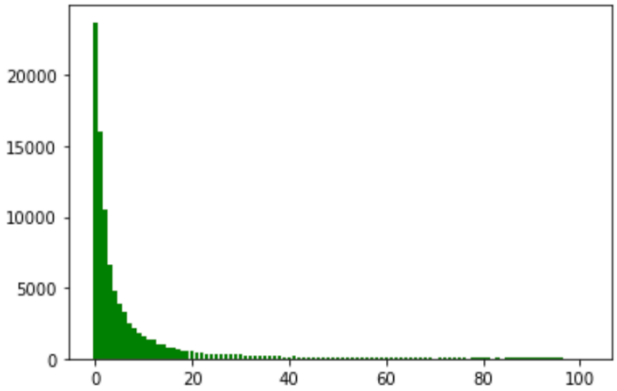

Figure 9 - Percentage of the variables ‘ETHx’, where x is between 13 and 16

Then, we made the histogram and boxplot of these variables to explore and better analyze these variables. Also, a methodology very much like the one above, we analyzed the percentage of the variable ‘AGE907’, that represents the population under age 18.

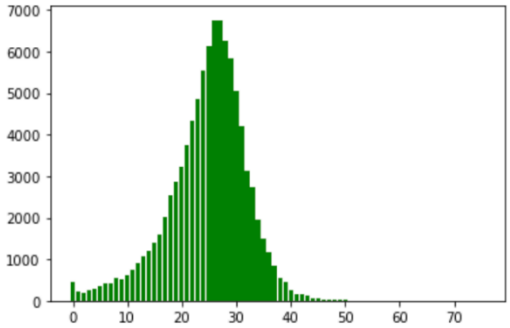

Figure 10 - Percentage of the variables ‘AGE907’

After that, we will represent the histogram and the boxplot of variable ‘AGE907’.

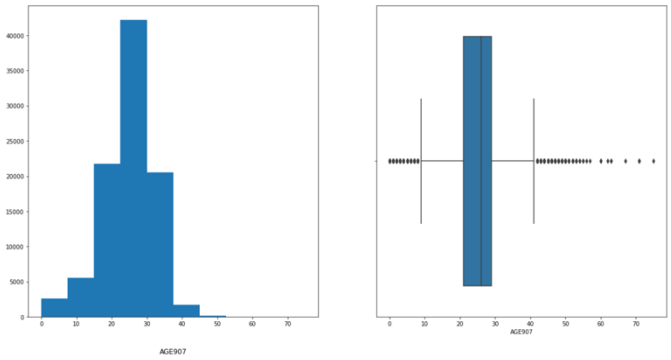

Figure 11 - Histogram and boxplot of variables ‘AGE907’

After that, we analyze the percentage of the variable ‘CHIL1’, ‘CHIL2’ and ‘CHIL3’.

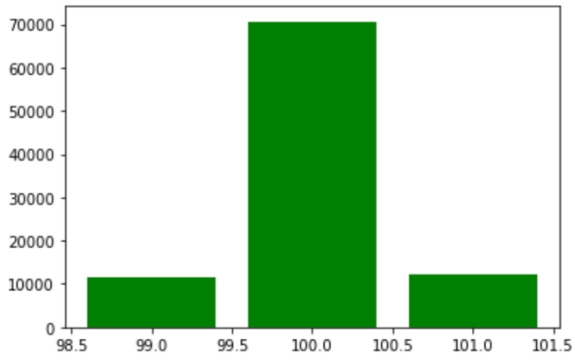

Figure 12 - Percentage of the variables ‘CHIL1’, ‘CHIL2’ and ‘CHIL3’

As we can see the percentage is greater than 100% and as we did before, for male/female neighborhoods, we scaled back the values of the neighborhoods close to 100%. And analyzed the histograms and the boxplots of these variables for any extreme value.

We then proceed to the analysis of the following variables: ‘AGE90x’, where x is a value between 1 and 7.

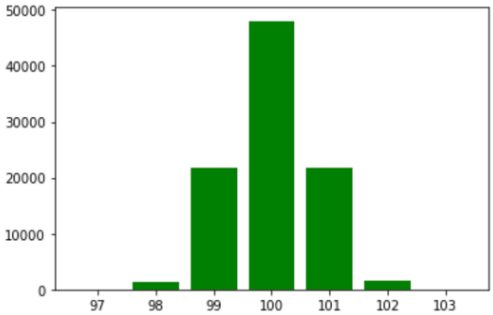

Figure 13 - Percentage of the variables ‘AGE90x’, where x is between 1 and 7

By seen the bar plot above we can notice that several records are outside 100% when checked, but those values are not insignificant, therefore, we scaled to not drop any important information.

For the variables ‘CHILCx’, where x is a value between 1 and 5 and for the variables ‘MARRy’, where y is a value between 1 and 4, we proceed in the same way as the previous ones. The process was analogous, we checked the percentages and tried to find any extreme values within the records for them to be dropped, we did not found any particularity that gave us justification to drop values based on this feature.

For the variables ‘ICx’, being complementary by definition where x is a value between 6 and 23, we calculate the percentage.

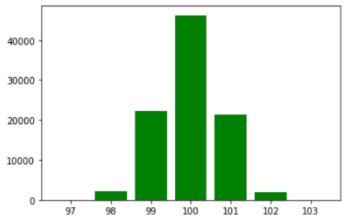

Figure 14 - Percentage of the variables ‘ICx’, where x is between 6 and 14

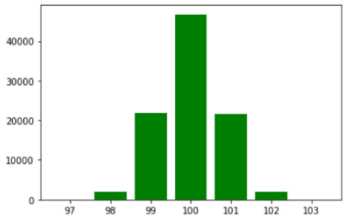

Figure 15 - Percentage of the variables ‘ICx’, where x is between 14 and 23

Subsequently, we scaled to not have percentages greater than 100%.

For the ‘LSCx’ variables, where x is a value between 1 and 4 and the ‘POBC1’ and the ‘POBC2’ variables, we calculated the percentage.

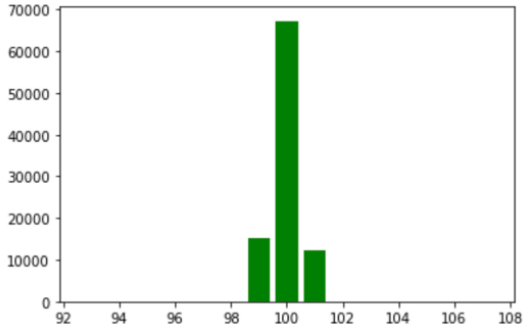

Figure 16 - Percentage of the variables ‘LSCx’, where x is between 1 and 4

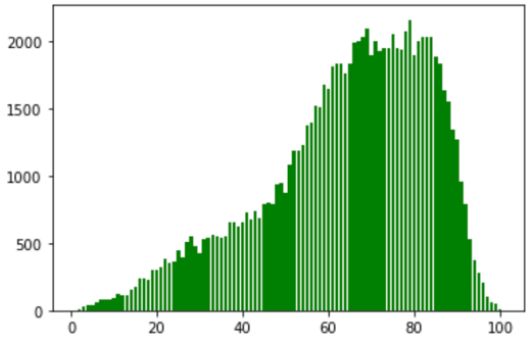

Figure 17 – percentage of the variables ‘POBC1’ and ‘POBC2’

After all of the manual validation of each feature, we calculated the proportion of records remaining from the original data (1% loss) and we saved our cleaned dataframe into a variable.

2. Literature Review

Clustering objects into groups is usually based on a similarity metric between objects, with the goal that objects within the same group are very similar, and objects between different groups are less similar.

In this chapter, we elaborate more details about the algorithms used.

Hierarchical Clustering

Hierarchical clustering is a commonly used clustering method of cluster analysis which seeks to build a hierarchy of clusters. It clusters similar instances in a group by using similarities of them.

The hierarchical clustering algorithms do not require knowing the pre-specified number of clusters as an input parameter, which is an advantage over partitioning algorithms. However, this advantage came with the cost of the algorithm complexity. The hierarchical decomposition can be represented by dendrogram. The basic agglomerative, hierarchical clustering algorithm works as following ways. Initially, each object is placed in a unique cluster. For each pair of clusters, some value of dissimilarity or distance is computed. For instance, the distance may be in minimum distances (Single linkage) in the current clustering are merged, until the whole data set form a single cluster.

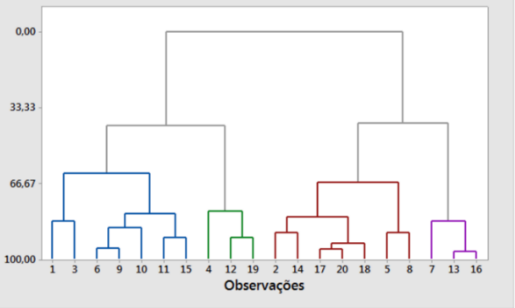

Figure 18 – Dendogram example

Self-Organizing Maps

Self-organizing map (SOM) is a neural network-based dimensionality reduction algorithm generally used to represent a high-dimensional dataset as two-dimensional discretized pattern. Reduction in dimensionality is performed while retaining the topology of data present in the original feature space.

K-Means Algorithm

The k-means clustering algorithm is known to be efficient in clustering large data sets. This algorithm is one of the simplest and the best-known unsupervised learning algorithms and it solves the well-known clustering problem.

The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into k clusters, where k is a predefined constant. First of all, k centroid point is selected randomly. These k centroids are the means of k clusters. Then, each item in the dataset is assigned to a cluster which is nearest to them. Then, means of all clusters are calculated again with new points added to them, until values of means do not change.

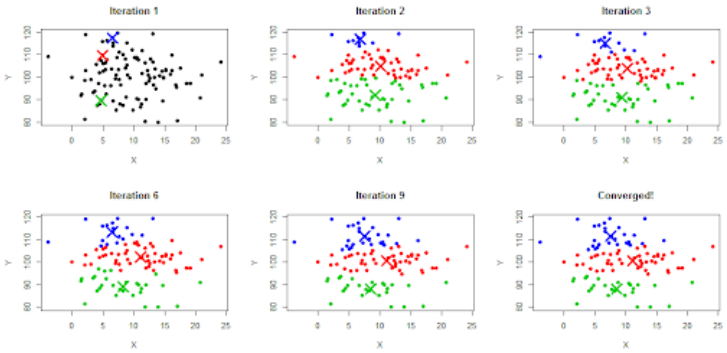

Figure 19 – K-means example

Chapter 3

1. Results

Assessing the result of our clustering solution, in practical terms, can be done by describing each cluster on terms of the categorical features.

State

STATES	0	1	2	3
More common	VT MS AK WY ID AR MT	AA DC RI CT MD HI CA	DE WV	NH WV SD ND MD
Less common	MD CA AA DC RI WV	MS ID WY AR MI SD NH VT	AA DC ME VT OH AK	CA CT AK HI OH AA DC DE MA

Table 1 – Categorical feature ‘State’ described with clusters

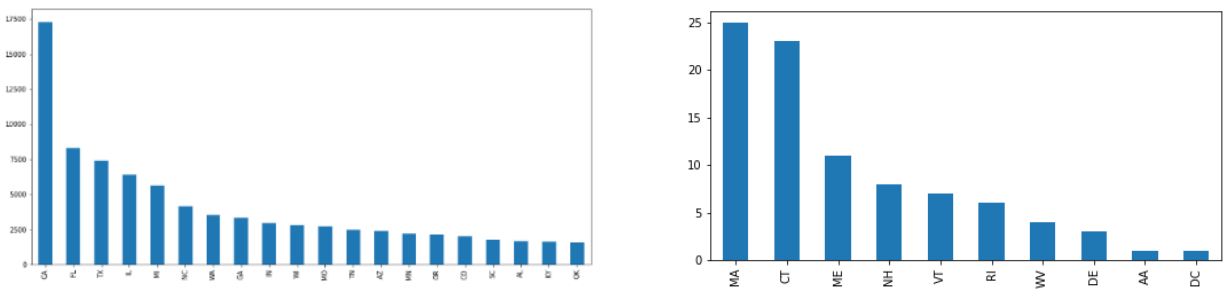

Figure 20 – Categorical feature ‘State’ described with clusters

Urbanicity Level

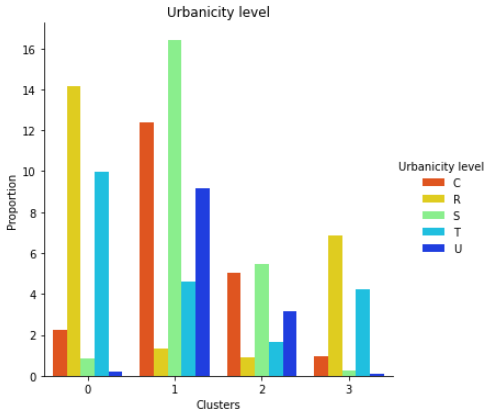

Figure 21 – Categorical feature 'Urbanicity Level' described with clusters

We can see that clusters 0 and 3 are characterized by being migratorily Rural/town while clusters 1 and 2 are more urban/city.

Social-Economic

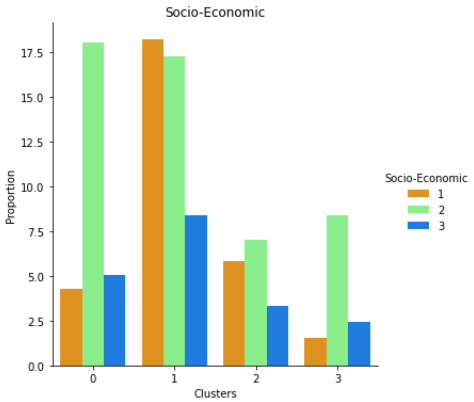

Figure 22 – Categorical feature 'Social-Economic' described with clusters

Cluster 1 is the one with the most % of people highest with the highest social economic status, while clusters 0 and 3 are characterized for more middle-class subjects. Interestingly enough, the proportion of individuals, throughout the clusters, with lowest social economic status is always fairly equal to the proportion of dataset in each cluster.

Wealth

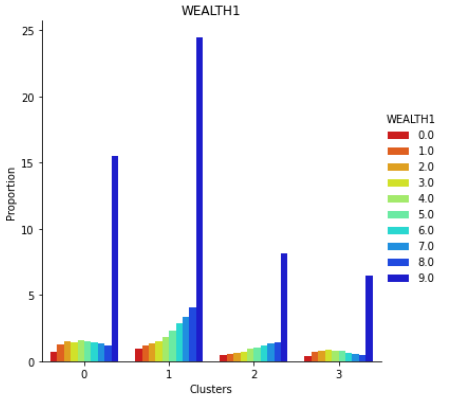

Figure 23 – Categorical feature 'Wealth' described with clusters

Shows us that the cluster 0 is where we can find individuals with lower wealth, cluster 1 in the other hand is where the must wealthy persons are, the others cluster are well distributed.

NUMCHLD

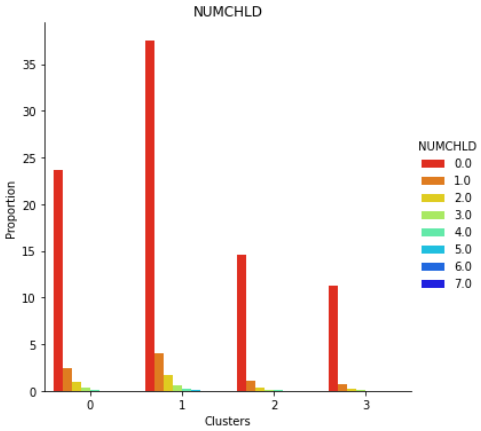

Figure 24 – Categorical feature 'Numchld' described with clusters

Cluster 1 is the one that agglomerates the number of people that have more kids, meanwhile clusters 2 and 3 are the ones with people with fewer kids.

Income

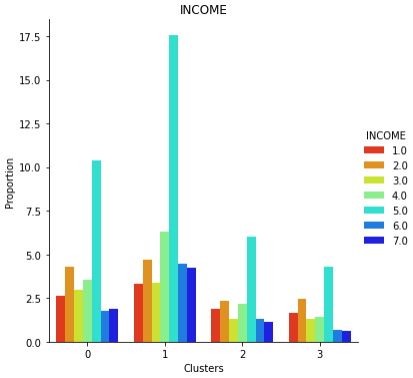

Figure 25 – Categorical feature 'Income' described with clusters

Cluster 1 is the cluster with higher incomes. Clusters 2 and 4 have lower.

Major

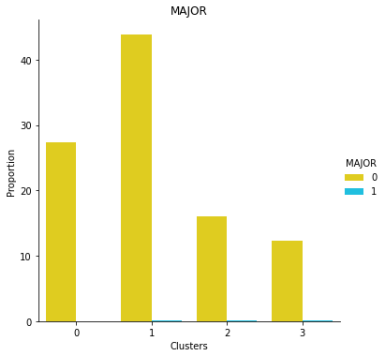

Figure 26 – Categorical feature 'Major' described with clusters

We can see that the must major donors (50%) are in the cluster 1.

Gender

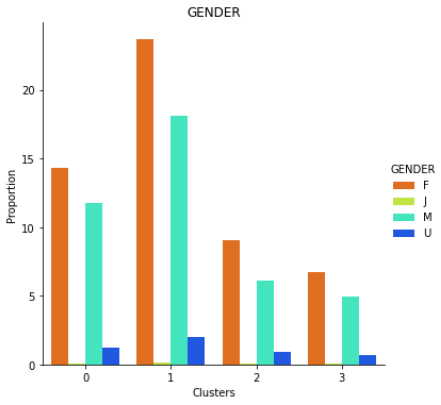

Figure 27 – Categorical feature 'Gender' described with clusters

Gender is equally distributed throughout the clusters except joint accounts that are more heavily represented on cluster 2.

CENTROIDS

merged_labels	POP90C1	ETH1	MARR1	AGE907	AGEC4	HVP5	IC7	month_no_donation	NGIFTALL	NUMPROM
0	1.627882	89.765599	62.696651	26.534860	14.727191	61.278992	19.258862	66.341745	5.744094	37.326358
1	98.033034	81.864771	55.953192	23.812458	13.960144	84.405207	15.299263	66.185645	5.482042	36.590369
2	98.083954	84.369999	55.865107	23.077545	13.619365	83.037208	15.977040	59.494613	19.541249	72.495830
3	1.271631	91.009980	62.424677	26.090328	14.447843	57.691999	19.814330	60.178224	19.803821	71.974582

Figure 28 – Centroids

Chapter 4

1. Conclusions

In our project, we did in our input data some clustering approaches. In the K-means algorithm, the initial centroids of the cluster are chosen as random and it is required to specify the number of clusters. Using empirical analysis, we found that the best number of clusters for the data set is 4.

We started by using the elbow method and from there we found the number of clusters to do the inertia. Then we merged and found a value of k = 4.

In order to further improve the search quality results, we performed hierarchical clustering on the input data. As a result of this work, we obtained a R^2 value of 63%.

We can conclude that the cluster with the best donors is cluster number 2, it includes the highest number of recurrent donors, and is alongside cluster 3, part of the cluster that not only donate more but also received more promotions. For PVA to be able to increase the amount of donations, they should focus their marketing efforts on cluster 2.

Cluster 2 is characterized by having few child's and a lot of neighbors, and more than likely would live in a city/urban area. Contrary to what could be assumed, the people who donate the most are not the ones who have higher income/wealth values.

Cluster 1 is the other cluster with people from urbanized areas, this cluster is filled with donators having high income, wealth, and the highest social economic status. However, people from it donate very few times. This is a great cluster to invest resources in since people from it have the money to spend.

Cluster 0 is characterized by having a larger number of people from rural areas, with lower wealth, this is the cluster with a low number of donations, and since people from it seem to not have that much money, we would recommend avoiding spending substantial resources to get people from it to donate.

Cluster 3 is the rural version of cluster 2, people from it donate a lot but have a lower wealth/income status. We would recommend not spend extra resources in this cluster, but to keep them on the lookout for future promotions.

REFERENCES:

[1] Jiawei Han, Michelinekamber, Jian Pei, "Data Mining- Concepts and Techniques", Morgan kaufman Publishers, 3rd edition,2013.

[2] Parul Agarwal,M Afshar Alam, Ranjit Biswas ,Analysis the agglomerative hierarchical Clustering Algorithm for categorical Attributes- International journal of innovation, Management and Technology,Vol 1, No 2 june 2010 , ISSN:2010-0248

[3] Zhao, Ying, and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. Proceedings of the eleventh international conference on Information and knowledge management. ACM, 2002.

[4] When using the K-Means Clustering Algorithm, i. and Singh, P., 2021. *When Using The K-Means Clustering Algorithm, Is It Possible To Have A Set Of Data Which Results In An Infinite Loop?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/60312401/when-using-the-k-means-clustering-algorithm-is-it-possible-to-have-a-set-of-dat> [Accessed 4 January 2021].

[5] En.wikipedia.org. 2021. *Self-Organizing Map*. [online] Available at: <https://en.wikipedia.org/wiki/Self-organizing_map> [Accessed 4 January 2021].

[6] 2021. [online] Available at: <https://web.itu.edu.tr/uzunper/documents/Document_Clustering.pdf> [Accessed 4 January 2021].

[7] Han, J. and Kamber, M., Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann Publishers, 2006.

[8] Borgatti, S.P., "How to Explain Hierarchical Clustering", Connections, 17(2):81-84, 1994.

[9] Python package scikit-learn: different clustering algorithm implemented in python. http://scikit-learn.org/stable/modules/clustering.html# clustering Last accessed: 02/19/2015

[10] En.wikipedia.org. 2021. *Self-Organizing Map*. [online] Available at: <https://en.wikipedia.org/wiki/Self-organizing_map> [Accessed 4 January 2021].