NOVA
IMS
Information
Management
School

MDSAA

Mestrado em Métodos Analíticos Avançados
Master Program in Data Science and Advanced Analytics

Data Mining Project Report

Paralyzed Veterans of America
Dataset Segmentation

Frederico José Jácome de Brito Santos 20200604
Svitlana Vasylyeva 20200617
Vasco Miguel Galvão Pestana r20170803
(Group X)

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

1. Introduction

Paralyzed Veterans of America is a veterans service organization founded in 1946, that helps its members – veterans of the armed forces who have experienced spinal cord injury or dysfunction.

The main aim of this project analysis of Paralyzed Veterans of America Dataset is to answer a research question:

How we can improve PVA organisation marketing strategy to recapture so-called “Lapsed” donors (individuals who made their last donation to PVA 13 to 24 months ago).

Our group was asked by the PVA 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.

The group was given an PVA dataset, consisting of 95412 observations of 476 Features.

The project files can be accessed through the link: https://github.com/fredericosantos/ims_DM_Project.

There are 6 Jupyter Notebooks, ordered by numbers, with a project analysis in the repository.

2. Exploratory Data Analysis

Notebook 1_EDA_FE.

Our dataset has 95412 observations and 476 features. We made a table with a brief clarification about the features that were dropped, due to various problems such as missing values, low variance, and noncompliance. This table can be found in the Appendix.

After the initial EDA, the most important features selected were:

- AGE (donors age, obtained using DOB)
- DOMAIN (Domain/Cluster code)
- STATE (State abbreviation)
- INCOME (Household Income)
- GENDER (gender)
- RECENCY
- FREQUENCY
- AMOUNT
- RAMNTALL (Dollar amount of lifetime gifts)
- NGIFTALL (Number of lifetime gifts to date)
- LASTGIFT (Dollar amount of most recent gift)
- AVGGIFT (Average dollar amount of gifts)
- LASTDATE_MONTHS (obtained from LASTDATE and ADATE_2 as the number of months between date of the last gift and the date when the last promotion (17NK) was sent)
- FIRSTDATE_MONTHS (obtained from FISTDATE and ADATE_2 in the same way as LASTDATE_MONTHS).

For better understanding of the relationship between our metric features, a correlation matrix was built (Fig 2.1).

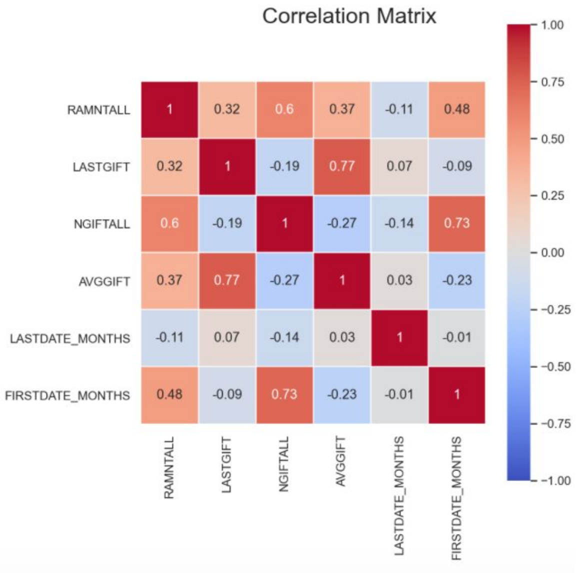


Fig. 2.1 - Correlation matrix for metric features

As we can see from Fig. 2.1, LASTGIFT is highly correlated with AVGGIFT (Pearson correlation coefficient is 0.77).

We also observed a high correlation between NGIFTALL and FIRSTDATE_MONTHS (0.73) and a high correlation between NGIFTALL and RAMNTALL (0.6). For the metric features, boxplots were built (Fig. 2.2) to do a first general pass on the data to understand its quality and possible outliers.

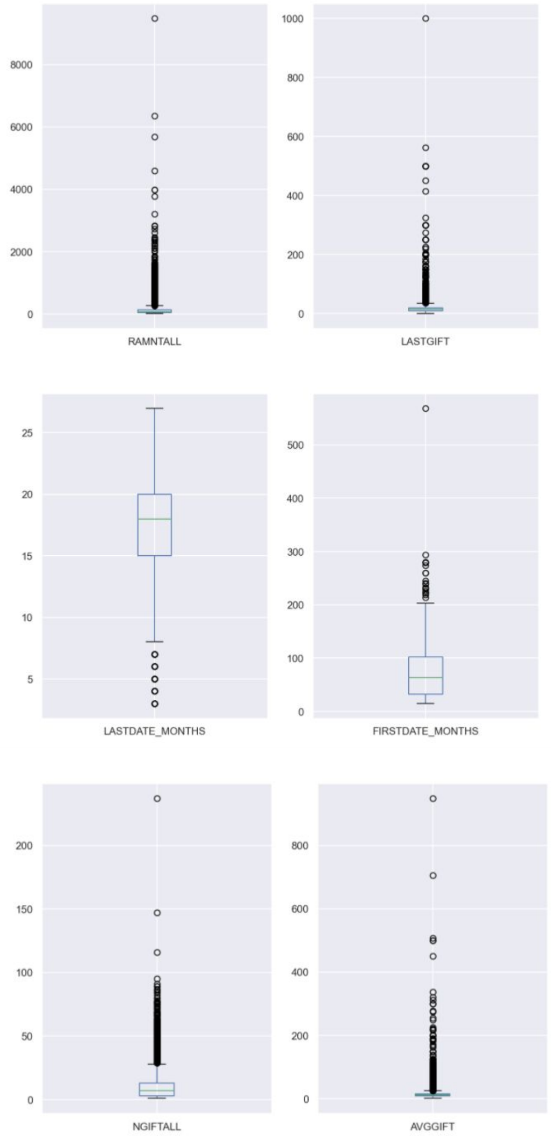


Fig. 2.2 - Boxplots for metric features

One of the metric features that we used in our project, AGE (obtained using DOB) was explored more thoroughly during the imputation part of our project.

Boxplots show quite a big number of outliers that we are going to detect in the nexts section of our report.

3. Data preparation

Notebooks 1_EDA_FE, 2_NN, 4_Outliers.

Data preparation is an important step in a data mining project, when we convert disparate, raw, messy data into a clean and consistent view.

Data preparation was made in three steps: coherence checks (where we also explain some slight feature engineering), missing values imputations and outliers detection.

3.1. Coherence Checks

Choosing between two features with the same meaning - ODATEDW and FISTDATE, the last one was chosen because of the non-compliance of ODATEDW with LASTDATE feature (3.8% of inconsistencies).

The date of the first gift (FISTDATE) and that person's birthday date had 278 observations dropped (0.29% of the dataset) since the first gift can not be before the date of birth.

Despite the fact that we should have only Lapsed donors in our dataset we had 82,309 lapsed, 7,173 inactive and 3,885 donors based on calculations made, so the percentage of not lapsed donors is 11.84%.

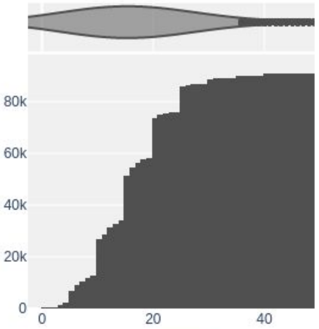


Fig. 3.1 - Last Gift Cumulative Distribution

We noticed the feature AMOUNT had outdated categories (donation of less than 1$) in comparison with the last donation each person had done (Fig. 3.1), so we engineered a new AMOUNT category:

Last Gift ($)	AMOUNT
] 0, 3]	A
] 3 - 5]	B
] 5 - 10]	C
] 10 - 15]	D
] 15 - 20]	E
] 20 - 25]	F
] 25 - Inf]	G

As for the INCOME, before we imputed it, we used K-Bins discretizer to bin the variable into 3 categories. Fig 3.2 shows its distribution.

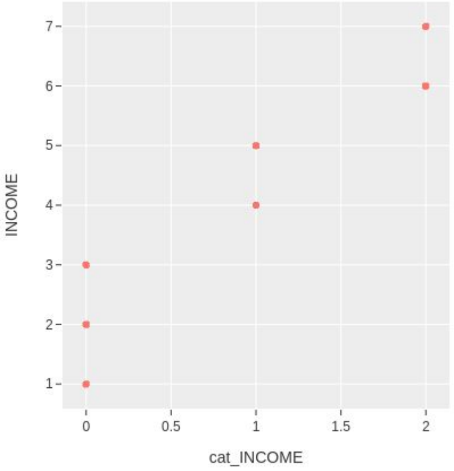


Fig. 3.2 - Distribution of INCOME (7 categories) into cat_INCOME (3 categories)

3.2. Missing values

For a better understanding of the pattern of missing values in features important for our analysis, the library missingno was used (Fig. 3.3).

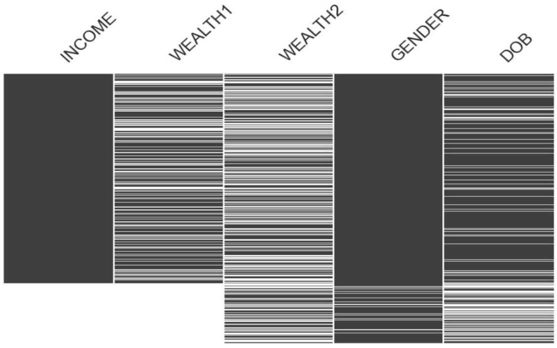


Fig. 3.3 - Missing Values for INCOME, WEALTH1, WEALTH2,GENDER, DOB

Observations that had NaNs in DOB and GENDER and INCOME at the same time were dropped (1767 rows, 2.14% of the dataset). These values should be evaluated in the future to assert the reason why they’re missing, possibly implementing measures to avoid these breaks in data.

3.3. Imputation of Missing Values

Notebooks 1_EDA_FE, 2_NN

Two missing values in FISTDATE were obtained from the ODATEDW feature.

AGE feature (derived from DOB and ADATE_2) had 22,116 NaNs, so K-Nearest Neighbors imputer (“KNNImputer”) from scikit-learn was used to impute missing values on this metric feature.

GENDER values were encoded as 1 for 'M', 0 for 'F', values 'A', 'C', 'U' were treated as NaNs, and all NaNs were imputed with most frequent value for this categorical feature.

Regarding the missing values for DOMAIN and INCOME (2,225 NaNs and 21,228 NaNs respectively), we decided to be creative. Deep Neural Networks with Entity Embeddings were used to train on the dataset, with all the other features as train and validation data, and impute the missing values in INCOME and DOMAIN, on a round-robin basis.

The results were indeed interesting and it is clear this method achieved an obvious improvement over more standard imputing methods. For comparison, we show the distribution of missing values and predictions for KNNImputer, Simple Imputer (scikit-learn) and our method.

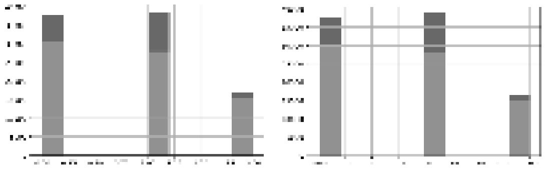


Fig. 3.4 - KNN & Simple Imputer for Income (Darker grays are predictions)

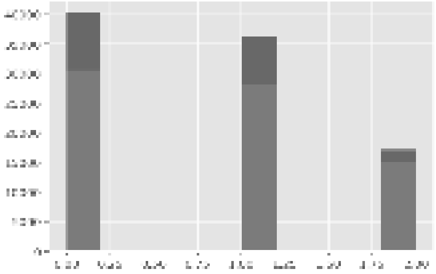


Fig. 3.3 - Our Method (DNN Imputer) (Darker grays are predictions)

3.4. Outliers Detection

Notebook 4_Outliers

We used two algorithms, Isolation Forest (scikit learn) and HDBSCAN (McInnes et al.) to detect outliers. We combined them by making an intersection of the outliers detected by each algorithm.

Isolation Forest works by randomly selecting a feature and then randomly choosing a split value for a decision tree. "The number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node." This path length serves as the decision function and shorter paths are chosen as outliers.

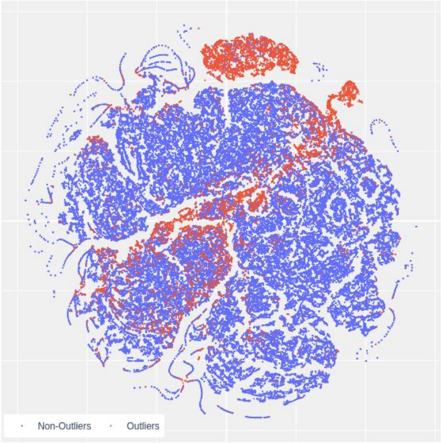


Fig. 3.5 - Outliers detected by Isolation Forest (10% contamination rate)

The HDBSCAN algorithm builds on top of DBSCAN to convert it into a hierarchical algorithm and then extracting a flat clustering based on the stability of the clusters.

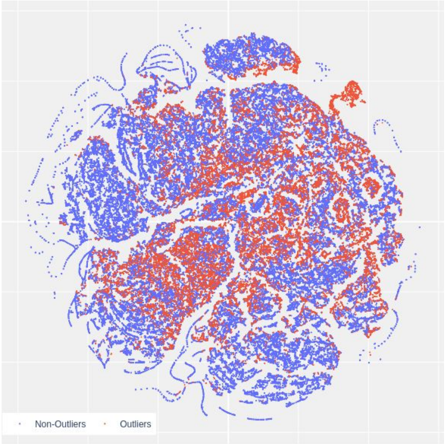


Fig. 3.6 - Outliers detected by HDBSCAN (20% upper quantile)

We then merged both solutions into one, ending up with 3.34% of the dataset as outliers. We labeled them and avoided using them during clustering.

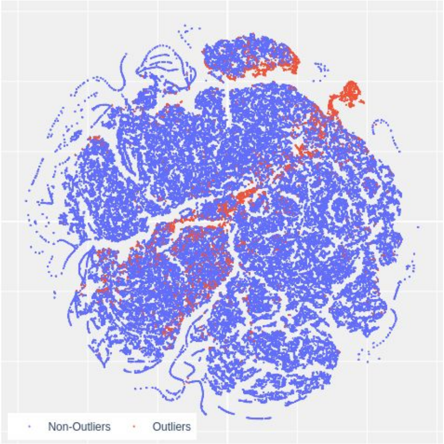


Fig. 3.7 - Final Outlier Solution

We plotted the outliers into a 3D plane to understand them a little better. It became clear that most of the outliers are actually the biggest donors.

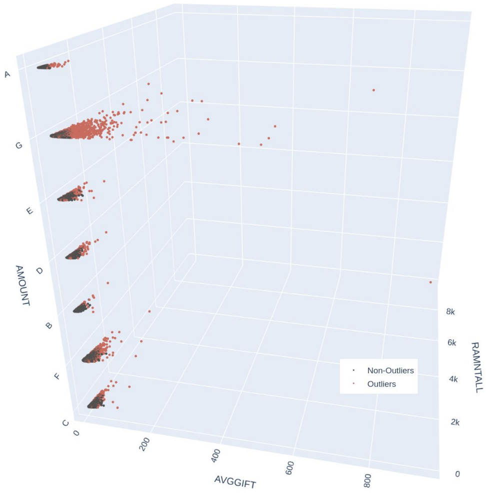


Fig. 3.8 - Visualizing Outliers by Money features

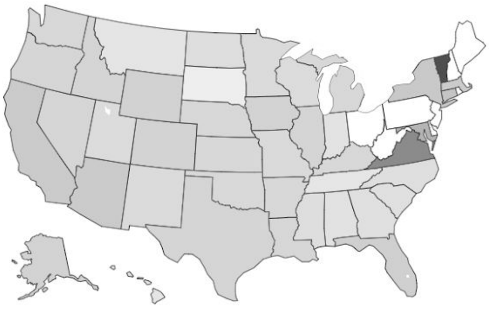


Fig. 3.9 - Outliers by State (darker means more outliers)

4. Socio-Economic Status Index

Notebook 3_PCA

We utilized the Census features (286) to build a Socio-Economic Status Index (SES), based on the Canadian Socioeconomic Factor Index.

The SEFI (1996) is a score derived from Canadian Census data that reflects non-medical social determinants of health and is used as a proxy measure of socioeconomic status (SES). The SEFI is derived from six census measures such as age, single parents status, female labour force participation, unemployment and education. The intent of SEFI is to reflect some degree of material and/or social deprivation, but it does not include a measure of income or wealth, substituting for income are highly correlated variables that also reflect SES - educational attainment and employment. (Manitoba Centre for Health Policy (MCHP), 2009)

In our work we developed methodology for constructing our SES index based on SEFI using following features:

- AGEC6 - Adults Age 65-74 (%)
- HHD7 - Single Parent Households (%)
- HHD11 - Single Female Householder (%)
- LFC3 - Females in Labor Force (%)
- LFC4 - Adult Males Employed (%)
- LFC5 - Adult Females Employed (%)
- EC2 to EC8 - Education Features (%)

These features are explained more thoroughly in our 3_PCA notebook. The SES index proceeds by running a Principal Component Analysis and then using the first principal component binned to 4 categories (4 types of SES index).

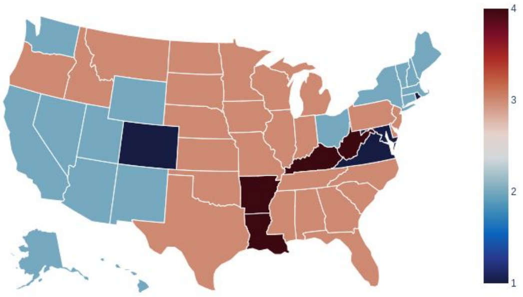


Fig. 4.1 - Mode of SES index by State

5. Clustering Customer Features

Notebook 5_Clustering

We tried several methods of clustering and, based on multiple performance criteria for clustering solutions without known targets, as our final algorithm, we used K-Means & Hierarchical Clustering methods to cluster our customer features. We will briefly explore each method tried and provide a visualization, over the same T-SNE, of how well each solution clustered.

HDBSCAN (McInnes et al., 2017)

We find it important to note that density based clustering, when compared against distance based clustering using metrics based on distances, will always underperform. We point this out as HDBSCAN gives outstanding visual results, even if it deems many points as outliers. But based on the metrics learned on this course, we did not feel comfortable selecting this algorithm.

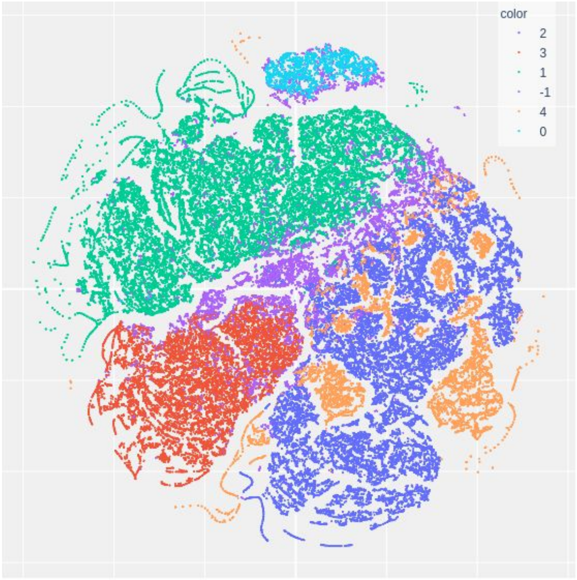


Fig. 5.1 - HDBSCAN Solution (5 clusters & Outliers)

K-Means

The Elbow Method to select the number of clusters was used:

1. Compute clustering algorithm for different values of n, varying n from 1 to 10 clusters;
2. For each n, calculate inertia;

3. Plot the curve of inertia according to the number of clusters n;
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

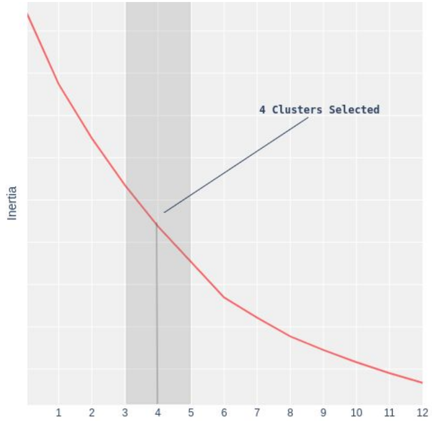


Fig. 5.2 - Inertia per number of clusters in K-Means

Our final solution for the K-Means clustering.

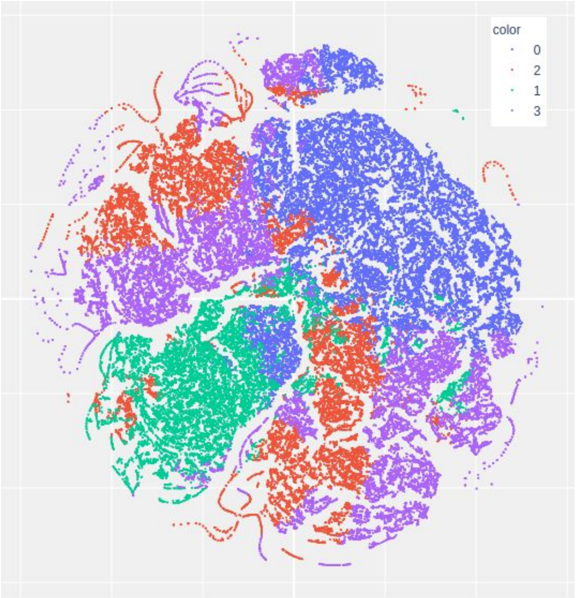


Fig. 5.3 - K-Means Solution (4 clusters)

K-Means & Hierarchical Clustering

In order to use Hierarchical Clustering, due to computational constraints, we mixed both a K-Means solution with a Hierarchical one. The dataset was clustered with 100 K-Means centroids and Hierarchical Clustering was then applied onto those clusters. We selected the *Ward* linkage method as it achieved the best results in the R^2 score (Fig 5.4).

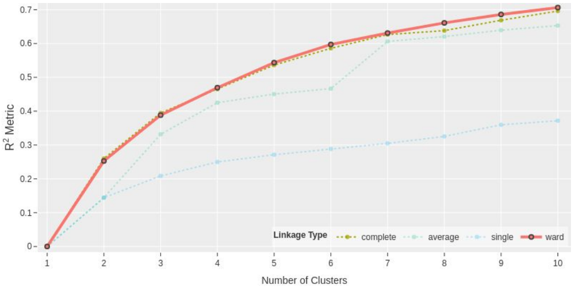


Fig. 5.4 - Assessing the best linkage type for hierarchical clustering

Based on a dendrogram graph of euclidean distances between clusters, 3 clusters were chosen for our solution.

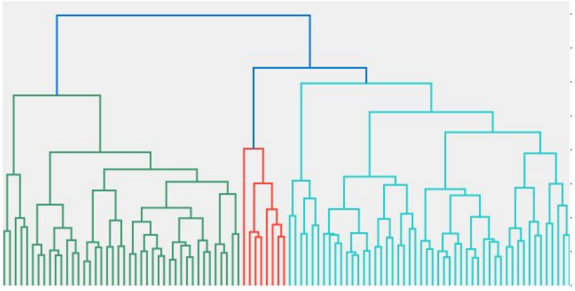


Fig. 5.5 - Dendrogram of Euclidean distances between clusters

Our final solution for this method can be seen in Fig. 5.3, with a clear distinction between the isolated cluster 2 (green) and the other two clusters.

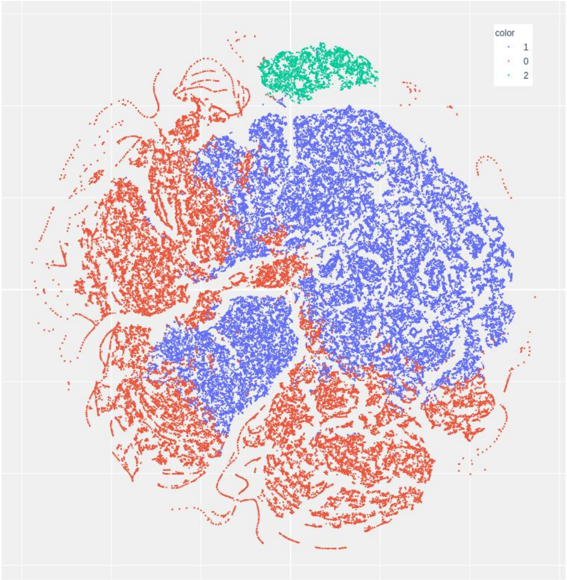


Fig. 5.5 - K-Means & Hierarchical Clustering Solution (3 Clusters)

DBSCAN

While the more advanced version of this algorithm, at least visually, shined, DBSCAN proved extremely hard to tune and never achieved decent results. This is most likely due to the algorithm's constraints on clustering around different density sizes and the distribution density of this dataset. By stark contrast, we can see the benefits of using HDBSCAN instead of DBSCAN.

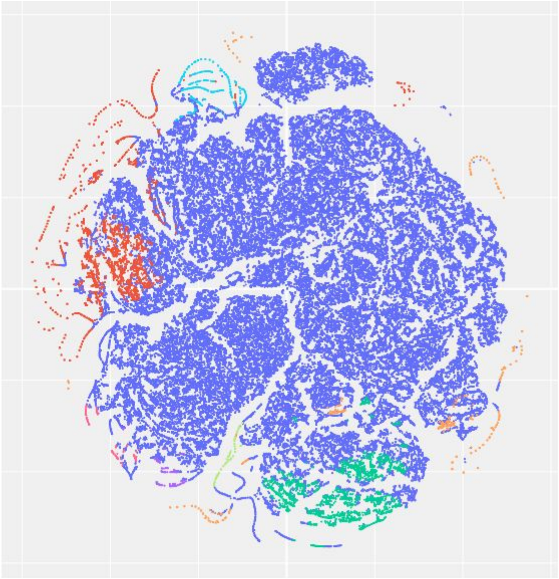


Fig. 5.5 - DBSCAN Clustering Solution (Catastrophic Failure with who cares how many Clusters)

Self-Organizing Maps

We selected a grid of 50x50 neurons to map the features and then applied Hierarchical Clustering onto those 2500 neurons using the same cluster selection technique we applied with the K-Means & Hierarchical Clustering solution. We also experimented with using K-Means to cluster the neurons but it performed worse than using hierarchical clustering.

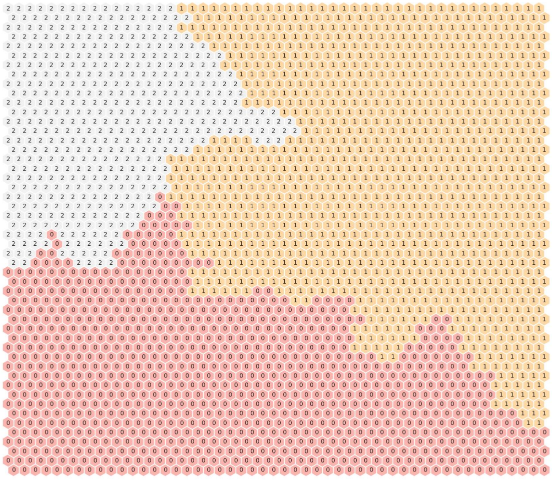


Fig. 5.6 - Hierarchical Clustering applied to SOM neurons (3 clusters)

It visually appeared to not be a great solution as we can see below (Fig. 5.7)

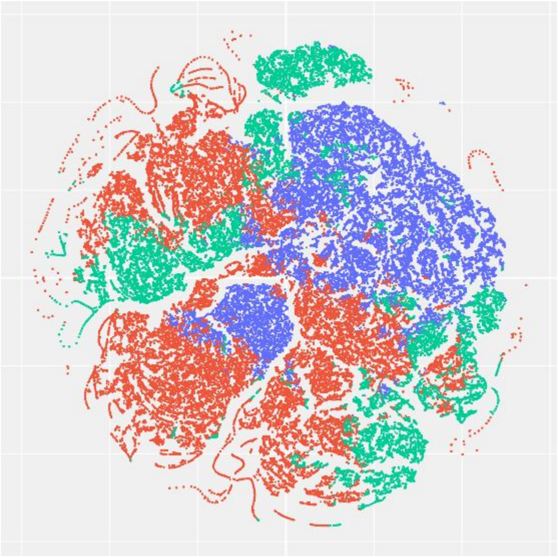
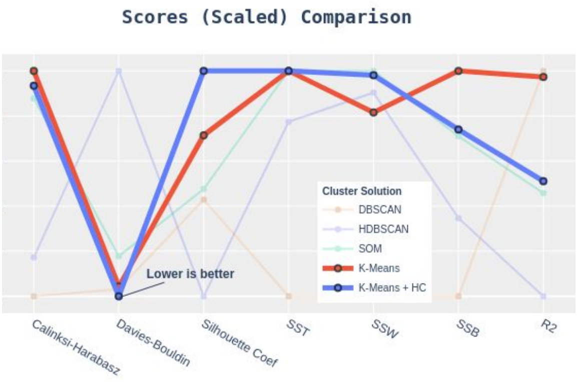


Fig. 5.7 - SOM & Hierarchical Clustering Solution (3 Clusters)

Comparing Clustering Solutions

As we noted previously, density based solutions are hard to measure against distance based solutions. Without targets, it's hard to compare both methods. We have found one method, parallel analysis, to measure the performance of density based methods better but it is not available in Python as far as we could research.

We constructed a combination of clustering performance evaluation metrics. The K-Means and K-Means & Hierarchical Clustering methods tied in first in terms of scoring metrics, so we decided to select the K-Means & Hierarchical Clustering approach based on the visual results of both methods.



6. Clustering Analysis

Notebook 6_Clustering Analysis

Nowadays, with the rise of machine learning in businesses of all shapes and sizes, customer nano-segmentation is more than possible and it is, therefore, extremely difficult to say there is an optimal solution to the number of clusters we want to keep for our final solution. Given this argument, however, we chose to assume PVA is somewhat "oldschool" since they still mail their promotions and, therefore, perhaps a more generalized cluster solution is required. For the final clustering solution, we merged the SES Index with the Customer Features clusters. We obtained the centroids of each feature for each pair of clusters and created a dendrogram (Fig. 6.1) to select how many clusters we would end up with.

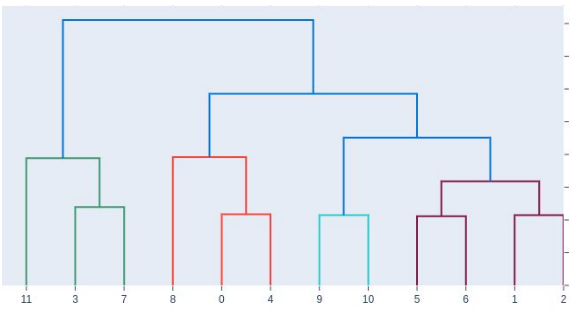


Fig. 6.1 - Dendrogram for Merging the 2 Cluster Solutions

We selected 4 clusters as our final Customer Segmentation. Marketing around 4 clusters of customers seems reasonable and not micro-targeted. It obviously depends on what kind of techniques, computing power, size of company and marketing budget we have available to determine how finely grained our customer segmentation can be, but we believe this is the optimal solution given the distances between our clusters and the fact that our client is PVA.

We took the centroids of the features of each final cluster and plotted (Fig 6.2) how both are merging to create a macro-view of how well these clusters ended up merging.

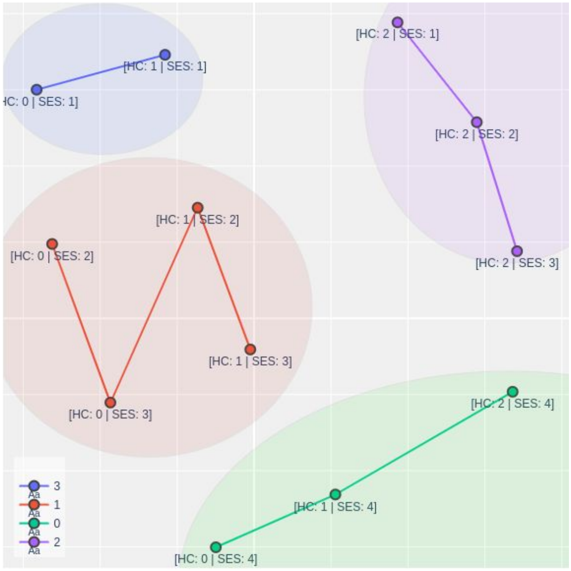


Fig. 6.2 - T-SNE of Clusters' Centroids

Cluster 0 - The Unwealthy Zero

This cluster is composed of SES Cluster 4, which, according to the index, are the people who have the lowest social-economic status. Based on Fig. 6.3 (INCOME), we can see that this is the cluster where its population average is poorest. This cluster also shows the lowest education percentage amongst clusters and the lowest employment rate (Fig. 6.6).

Interestingly enough, this cluster incorporates a part of the HC 2 cluster (Fig. 6.2), which is the cluster where customers donate the most and are active donors. For the **Unwealthy Zero**, we suggest focusing on a marketing approach based on lower donation amounts.

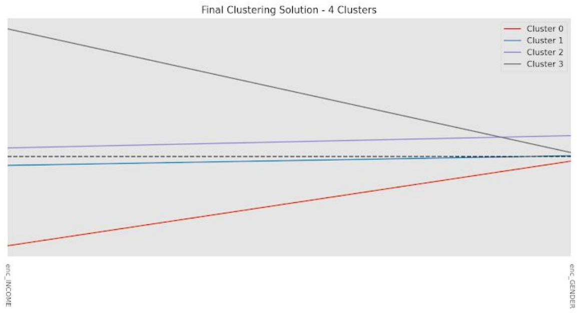


Fig. 6.3 - Final Clustering Solution - 4 Clusters

Cluster 1 - The Big One

This cluster, as we can see in Figure 6.3, is the largest and possibly less meaningful. It incorporates 50% of HC 0 and 66% of SES 2 and 3.

According to Fig. 6.5, these customers tend to be more towards the Inactive side (if LASTDATE is higher, it means it has been more months since they last donated).

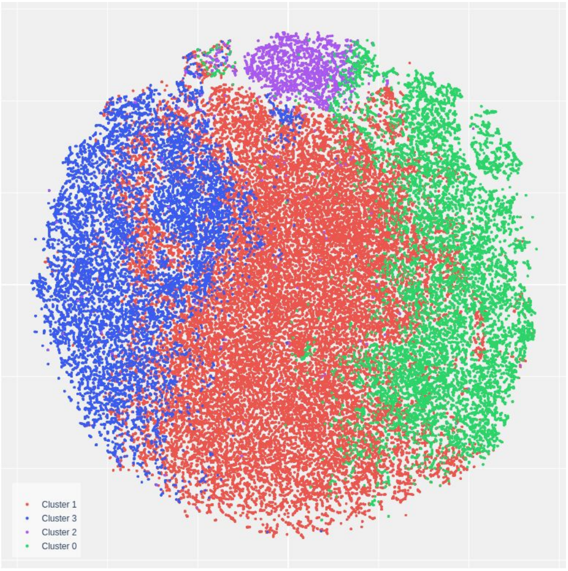


Fig. 6.4 - T-SNE of Customer Segmentation

Cluster 2 - The Spending Two

As we can see, HC cluster 2 (customer features clusters) aggregated all SES Indexes except for SES 4 (Fig. 6.2), which means this cluster contains the population that are better off in life.

These customers tend to be slightly better educated than the average population (Fig. 6.6). They tend to have average INCOME (Fig. 6.3) but they are the cluster that spends the most (Fig 6.5 - RAMNTALL), are the most frequent (Fig 6.5 - NGIFTALL) and active (Fig 6.5 - LASTDATE) donors.

We do not believe that we need a specific marketing strategy for these donors as they are already highly involved with PVA, perhaps appreciation marketing would work best for this cluster.

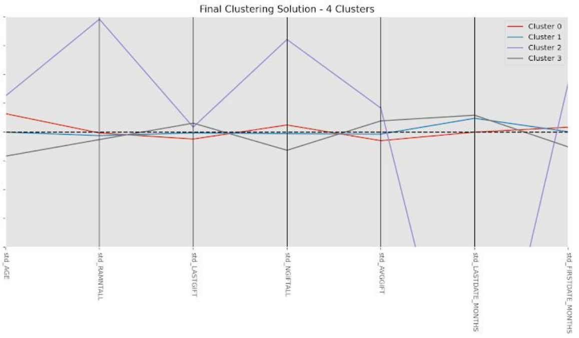


Fig. 6.5 - Final Clustering Solution - 4 Clusters

Cluster 3 - The Youth Three

This cluster is mainly composed by the younger population (Fig. 6.5 - std_AGE) living in young neighborhoods (Fig. 6.6 - Aged 65-74). These customers have the highest employment rate of all clusters as well as the highest amount of education (Fig. 6.6).

Unsurprisingly, the **Youth Three** also scores the highest in the Social-Economic Index, containing only SES 1 in their cluster solutions (Fig. 6.2).

This cluster is also the wealthiest of all clusters, with INCOME higher than others (Fig. 6.3).

These customers are also the most inactive of all and have not donated a lot (Fig 6.5 - NGIFTALL, LASTDATE). As a marketing strategy, we would recommend to appeal to younger audiences through “woke” marketing, possibly through the use of social media.

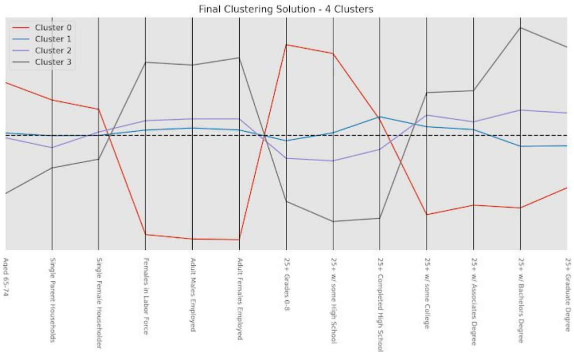


Fig. 6.6 - Final Clustering Solution - 4 Clusters

7. Outlier Classification

Notebook 6_Clustering Analysis

We applied a Semi-supervised learning method to select outliers. We used a Decision Tree Classifier, where it classified 95% of our customers correctly, to fill in the outliers.

8. Conclusion

Regarding the methods applied in this project, we're very pleased with the Deep Neural Network Imputer we developed and it showed promise to be applied in future projects. We're very satisfied with our outlier method as well, as it proved to segment an important part of the customer base.

We would like to find better evaluation metrics for the different kinds of clustering solutions (density vs distance). HDBSCAN seemed to be the most interesting to select based on visual analysis but, unfortunately, we could not argue to select it due to the lack of good performance metrics. It is a true clustering solution, whereas K-Means and Hierarchical Clustering are more of partitioning solutions. *"Not all those who wander are lost"*. That is our sentiment towards outliers; they should be outliers.

Regarding the results achieved, we're very happy to be able to showcase good clustering solutions with such a difficult dataset. Based on Fig. 6.3, even though we did not use the Income and Gender features during our clustering process, because they were not a metric feature, the final cluster solution successfully segmented the customers around these categorical features in a meaningful way. Both Fig. 6.5 and 6.6 also confirm that our clustering is successfully selecting different kinds of customers.

Our Marketing Strategy consists of the following:

- The **Unwealthy Zero**: focus on micro-donations;
- The **Big One**: Use the General Marketing approach of the company;
- The **Spending Two**: Focus on Appreciation Marketing;
- The **Youth Three**: Focus on Woke & Social Media Marketing

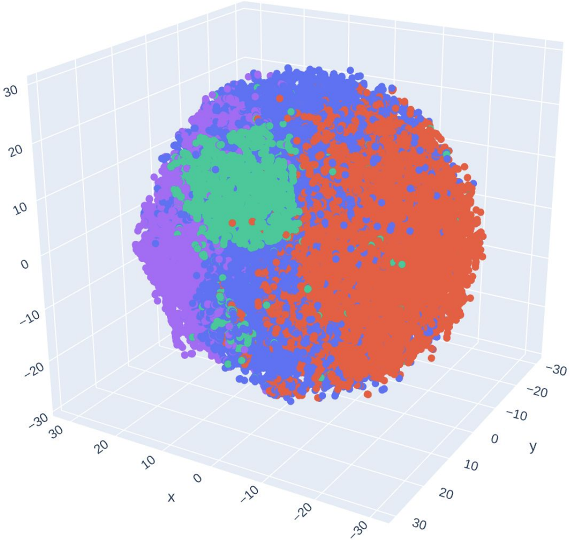


Fig. 8.0 - One last look at our 3D T-SNE planet

8. Bibliography

"KNNImputer." K-Nearest Neighbors imputer, https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html.

Manitoba Centre for Health Policy (MCHP). Composite Measures/indices Of Health And Health System Performance, August 2009, http://mchp-appserv.cpe.umanitoba.ca/reference/Chip.pdf#Page=108&View=Fit.

McInnes, L., et al. "hdbscan: Hierarchical density based clustering." Journal of Open Source Software, The Open Journal, vol. 2, no. 11, 2017, hdbscan.readthedocs.io/.

scikit learn. "Isolation Forest." Scikit Learn, https://scikit-learn.org/stable/auto_examples/ensemble/plot_isolation_forest.html#sphx-glr-auto-examples-ensemble-plot-isolation-forest-py.

9. Appendix

Features dropped because of missing values:
PVASTATE, NUMCHLD, WEALTH1, MBCRAFT, MBGARDEN, MBBOOKS, MBCOLECT, MAGFAML, MAGFEM, MAGMALE, PUBGARDN, PUBCULIN, PUBHLTH, PUBDOITY, PUBNEWFN, PUBPHOTO, PUBOPP, WEALTH2, GEOCODE, LIFESRC
Low variance columns (dtale)
MBCOLLECT, PUBPHOTO, ETH12, TPE5, TPE6, AFC3, HC15
Non-compliant with other features:
TCODE - noncompliant with GENDER column, ODATEDW - noncompliant with LASTDATE, some ADATE_X - non-compliant with ADATE_2
Redundant or irrelevant features:
OSOURCE, GEOCODE, ZIP, MAILCODE, NOEXCH, RECINHSE, RECP3, RECPGVG, RECSWEEP, MDMAUD (redundant), HOMEOWNR, CHILD03, CHILD07, CHILD12, CHILD18, WEALTH1, HIT, DATASRCE, MALEMILI, MALEVET, VIETVETS, WWIIVETS, LOCALGOV, STATEGOV, FEDGOV, SOLP3, SOLIH, MAJOR (redundant), COLLECT1 - PLATES, LIFESRC, PEPSTRFL, CENSUS FEATURES (except those that were mentioned in main part, and were used for AGE imputation, SES index construction) CARDPROM, MAXADATE, NUMPROM, CARDPM12, NUMPRM12, CARDGIFT, MINRAMNT, MINRDATE, MAXRDATE, NEXTDATE, TIMELAG, CONTROLN, HPHONE_D, GEOCODE2