NOVA IMS Information Management School

Msc in Data Science and Advanced Analytics with Major in Business Analytics

January 2021

Fall Semester

Data Mining Curricular Unit

Customer Segmentation to analyse the PVA institution lapsed donors

Beatriz Chumbinho R20170867

Inês Costa R20170775

Rodrigo Matias R20170880

Paralyzed Veterans of America

Índice

1. Introduction:

This project aims to develop a Customer Segmentation for *PVA* (Paralyzed Veterans of America) recent fundraising appeals. In a way that it will be possible to understand how *PVA* lapsed donors behave and identify the different segments of donors and potential donors within their database. This segmentation enables the *PVA´s* association to perform more targeted campaigns in order to increase the offers.

2. Data and variables description:

The *PVA´s* dataset has 95412 observations and 476 variables (including various data types - *object, int, float, datetime*) and there are 0 duplicate records in the dataset.

Our team started to check the missing values of our data, we noticed there are a lot of missing values in the variables, mainly for the ones related with dates, which we will handle later in this document. On the other hand, the missing values per observation are not significant, this way, there was no need to remove any row.

3. Data preparation:

3.1. Data types modification:

To avoid future problems, we detected the need to change the datatype of some columns with dates from *object* to *datetime*.

3.2. Dropping variables:

3.2.1. Due to irrelevance: ***'DATASRCE','OSOURCE','TPE10','TPE11','TPE12','TPE13','CONTROLN','MSA','ADI','DMA','MAILCODE','Unnamed:0','ADATE_2','ZIP','MSA','ADI','DMA','LIFESRC','RAMNT_3','RAMNT_4','RAMNT_5','RAMNT_6','RAMNT_7','RAMNT_8','RAMNT_9','RAMNT_10','RAMNT_11','RAMNT_12','RAMNT_13','RAMNT_14','RAMNT_15','RAMNT_16','RAMNT_17','RAMNT_18','RAMNT_19','RAMNT_20','RAMNT_21','RAMNT_22','RAMNT_23','RAMNT_24','NOEXCH','MDMAUD','MDMAUD_R','MDMAUD_F','MDMAUD_A','CHILC1','CHILC2','CHILC3','CHILC4','CHILC5', 'ANC1','ANC2','ANC3','ANC4','ANC5','ANC6','ANC7','ANC8','ANC9','ANC10','ANC11','ANC12','ANC13','ANC14', 'ANC15'***

3.2.2. Due to a lot of missing values: ***'WEALTH1','WEALTH2','NUMCHLD'***

3.3. Metric vs Non-metric vs Date features:

In order to make it easier to declare specific instructions for specific groups of variables we divided them into 3 groups:

- *Metric Features*
- *Non-metric Features*
- *Date Features*

We also included the third group because, since our dataset is loaded with features related to dates, we considered they should have a special treatment, before we convert them into variables that belong to one of the other two groups.
This step was performed manually to avoid perturbations due to data types.

3.4. Imputation of Missing Data:

1. Non-Metric features:

In order to perform the imputation of the missing data for the *non-metric* features we used the mode through the following code:

```
df[Non_metric_features] = df[Non_metric_features].fillna((df[Non_metric_features].mode()))
df[Non_metric_features] = df[Non_metric_features].astype(str)
```

Fig. 1

2. Date Features:

For the variables referring to dates our team noticed that most of them had more than 95 000 rows with missing values so performing a fill on those variables would be a false solution.
Regarding the ***'DOB'*** variable which has a lot of missing values but is important in our point of view, we decided to transform it, in the *Feature Engineering -FE-* section, into a new metric variable (***'AGE'***) so the missing values fill in the ***'AGE'*** is after the *FE* section and we performed it it using a *Decision Tree - DT*.
In order to do that, we divided our dataset into 2 datasets - df_2 which corresponds to df dataset observations without missing values for age (train dataset) and df_3 being the df dataset observations with missing values for age (test dataset) - and each one of them with the variables divided into data (all variables but the age) and target (the age). Our team observed which variables mean the most regarding with a age of a person (e.g. ***'CHILD'***). Then the *DecisionTreeClassifier() algorithm* was applied and fitted to our data in order to perform the prediction.

3. Metric Features:

First of all we started to check which metric features had missing values to decide in which ones it worth to fill them. So, the variables ***'MBCRAFT','MBGARDEN','MBBOOKS','MBCOLECT','MAGFAML','MAGFEM','MAGMALE','PUBGARDN','PUBCULIN','PUBHLTH','PUBDOITY','PUBNEWFN','PUBPHOTO','PUBOPP'*** had some missing data but since we will transform them latter in one variable (***'OFFERS'***) and we won´t count with the null rows, our team filled those values with 0 so we assume that if the value is null the donor has not responded to that type of mail order offer.
Another *metric feature* with missing values is ***'TIMELAG'***, so to perform the imputation of this variable we used the *K-Nearest Neighbours Imputer - KNNImputer - algorithm*. Our team checked the 5 most correlated variables with ***'TIMELAG'*** in order to use them as neighbours to perform the *KNNImputer algorithm*.

```
imputer = KNNImputer(n_neighbors=2)
df_filled = imputer.fit_transform(df_metric)
```

Fig. 2

Regarding the ***'AGE'*** feature (only created in the feature engineering section) we decided to use a *Decision Tree* to classify the

missing data. In order to do that, we divided our dataset into 2 datasets - df_2 which corresponds to df dataset observations without missing values for age (train dataset) and df_3 being the df dataset observations with missing values for age (test dataset) -
and each one of them with the variables divided into data (all variables but the age) and target (the age).

```
df2 = df[df['AGE'].notnull()] #Rows with age not null
X_train = df2.copy()
y_train = df2['AGE']
X_train.drop(columns = 'AGE', inplace=True)
X_train = X_train[['socio_econ','CHILD','MAJOR','VETERANS','KIDSTUFF','NGIFTALL']]
```

Fig. 3

```
tree_age = DecisionTreeClassifier()
tree_age.fit(X_train, y_train)
```

Fig. 4

3.5. Outliers removal:

3.5.1. Methods used to detect outliers:

1. Manually:
   After the imputation, we coded boxplots and histograms, which allowed us to have a visualization in the data to, consequently see where the outliers were positioned. It is important to note that we only performed this task for the *Metric Features*.

The following plot is an example of what our team did. We divided the *Metric Features* into groups in order to be possible to visualize and interpret each plot.

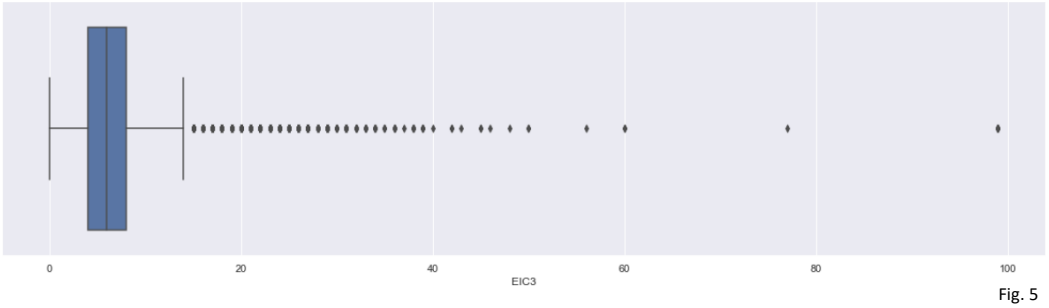


Fig. 5

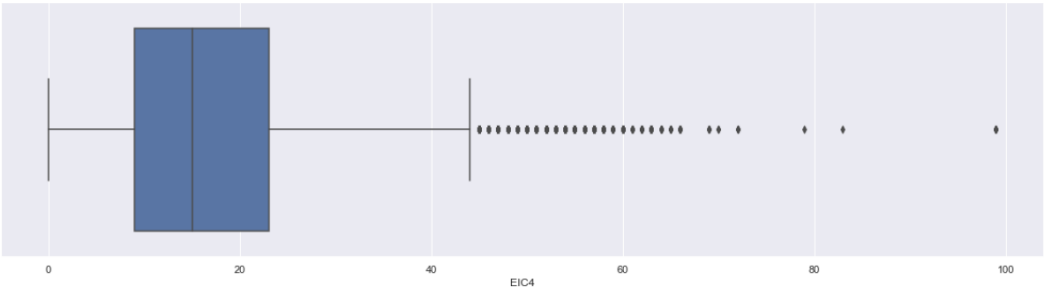


Fig. 6

We filtered df according to the plots obtained with those plots and we noticed that we still kept 99.9% of the data, very few rows (~100) were removed.

2. Local Outlier Factor (LOF):

   Is a density based algorithm which performs an unsupervised outlier detection by comparing the local density of a sample to its neighbours. For the contamination parameter we choose a value of 0.03 in order to not remove more than 3% of the observations. The code used to perform this step is provided below:

```
# model specification
model1 = LocalOutlierFactor(n_neighbors = 2, metric = "euclidean", contamination = 0.03)
# model fitting
y_pred = model1.fit_predict(df[Metric_features])
# filter outlier index
non_outlier_index = np.where(y_pred == 1) # negative values are outliers and positives inliers
# filter outlier values
non_outlier_values = df[Metric_features].iloc[non_outlier_index]
```

Fig. 7

Our team considered the visualization method ambiguous since some attributes were a little hard to fully visualize. This way, we believe that the best approach was to only use the *LOF method* to remove the outliers from our dataset. The percentage of data kept after removing outliers was 0.9699.

3.6. Coherence checking:

Age:
We started by checking if there were any persons with odd age values, such as bigger than 110 years, which we didn't have, but we had some rows were the age was smaller than 16, we even had rows with 3 and 4 years which we believe that it's an incoherence taking in consideration that this is lapsed donor dataset, so we proceed with the elimination of those rows.

Percent Children Age:
There are 155 rows which we deleted because the sum of ***'CHIL1'***,***'CHIL2'*** and ***'CHIL3'*** is 0.

Income Percentages:
Summing all the variables related with the income of the households (***'IC6'*** - ***'IC14'***), we noticed that 871 rows had a 0 percentage in all the income categories. So we proceeded with the elimination of those rows. There are also some other odd values such as 97,98, 99,101,102 or 103 but there are so many that we can't delete them all, we will just take this in consideration for future reference during our analysis.

We ended up with 91085 rows after performing incoherence checking meaning that we deleted 5% of the rows during this step.

3.7. Feature engineering:

In the present section we performed feature engineering. Our goal was to produce tools to retrieve the maximum information from the dataset. To reach our objective we created new variables that are transformations of variables already existing in the dataset and we also changed the way some variables were presented.

Variable Name	Description	Creation	Example of Code
T CODE *-Already existing*	Donor title code: to understand if the donors are related with the army, if the women are married and so on.	We mapped each number code of *TCODE* to be a title, with characters.	`df["TCODE"] = df["TCODE"].map({'0':'Undifined','1':'Man', '2': 'Womar'`

AGE
-New variable
	The age of each donor.	We computed the difference between the currente datetime and the *DOB* variable.	`now = pd.Timestamp('now')`
			`df['DOB'] = pd.to_datetime(df['DOB'], format='%m%d%y')`
			`df['DOB'] = df['DOB'].where(df['DOB'] < now, df['DOB'] - np.timedelta64(100, 'Y'))`
			`df['AGE'] = (now - df['DOB']).astype('<m8[Y]')`

PVASTATE *- Already existing*	Indicates whether the donor lives in a state served by the organization's EPVA chapter.	We mapped each character to 0 or 1.	`df["PVASTATE"] = df["PVASTATE"].map({'P':1,'E':1,' ':0})`
MAJOR *- Already existing*	Indicates whether the donor is a major donor or not.	We mapped each character to 0 or 1.	`df['MAJOR'] = df['MAJOR'].map({' ':0,'X':1})`

CHILD
-New variable
	Indicates whether the donor has children or not.	We used *CHILD03, CHILD07, CHILD12, CHILD18* variables. We mapped each character to 0 or 1 and then we sum and then we mapped each number to 0 or 1 according if the donor has or not children.	`df['CHILD18'] = df.CHILD18.map({'B':1,'M':1,'F':1,' ':0})`
			`df['CHILD'] = df['CHILD03']+df['CHILD07']+df['CHILD12']+df['CHILD18']`
			`df['CHILD'] = df.CHILD.map({1:1,0:0,3:1,2:1,4:1})`

GENDER *- Already existing*	Indicates the donor gender – male (M), female (F), unknown(U).	We mapped each character to M,F or U.	`df['GENDER'] = df.GENDER.map({'J':'U',' ':'U','C':'U'`
VETERANS,COLLECT1, BIBLE, CATLG, HOMEE, PETS, CDPLAY,STEREO, PCOWNERS,PHOTO, CRAFTS,FISHER, GARDENIN,BOATS, WALKER, KIDSTUFF,CARDS,PLATES	This set of variables indicates whether the donor has or not a specific interest.	We mapped each character to 0 or 1.	`df['VETERANS'] = df.VETERANS.map({' ':0, 'Y':1,'N':0})` `df['COLLECT1'] = df.COLLECT1.map({' ':0, 'Y':1,'N':0})`

- Already existing			
PEPSTRFL *- Already existing*	Indicates PEP Star RFA Status.	We mapped each character to 0 or 1.	`df['PEPSTRFL'] = df['PEPSTRFL'].map({' ':'0','X':1})`
OFFERS *-New variable*	Total number of known times the donor has responded to other types of mail order offers.	We used all the variables correspondent to mail order offers. We mapped each character to 0 or 1 and then we sum and then we mapped each number to 0 or 1 according if the donor has or not responded.	`df['PUBOPP'] = df.PUBOPP.map({None:0,' ':0,0:0,1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1})` `df['OFFERS'] = df['PUBOPP']+df['PUBPHOTO']+df['PUBNEWFN']+df['PUBDOITY']+df['PUBHLTH']`
HOMEOWNR *- Already existing*	Indicates if the donor ir home owner or not.	We mapped each character to 0 or 1.	`df['HOMEOWNR'] = df.HOMEOWNR.map({'H':1,' ':0,'U':0})`
RECINHSE *- Already existing*	Indicates if donor has given to PVA's In House program.	We mapped each character to 0 or 1.	`df['RECINHSE'] = df['RECINHSE'].map({' ':'0','X':1})`
RECPGVG *- Already existing*	Indicates if it is a planned Giving Record.	We mapped each character to 0 or 1.	`df['RECPGVG'] = df['RECPGVG'].map({' ':'0','X':1})`
RECPGVG *- Already existing*	Indicates if it is a sweepstakes Record.	We mapped each character to 0 or 1.	`df['RECSWEEP'] = df['RECSWEEP'].map({' ':'0','X':1})`
Urbanicity *-New variable*	Urbanicity level of the donor's neighborhood.	We divided the *DOMAIN* variable into 2 new ones, the Urbanicity took the first position of the split.	`df['Urbanicity'] = df.DOMAIN.astype(str).str[0]`
socio_econ *-New variable*	Socio-Economic status of the neighborhood.	We divided the *DOMAIN* variable into 2 new ones, the Urbanicity took the second position of the split.	`df['socio_econ'] = df.DOMAIN.astype(str).str[1]`
Gifted *-New variable*	Indicates if the donor gave or not in the last promotions	We used the not null *RDATE* variables, summing them.	`df['Gifted'] = (df['RDATE_3'].notnull() + df['RDATE_4'].notnull() +`

Table 1

Dropped variables:

'CHILD03','CHILD07','CHILD12','CHILD18','MBCRAFT','MBGARDEN','MBBOOKS','MBCOLECT','MAGFAML','MAGMALE','PUBGARDN','PUBCULIN','PUBHLTH','PUBDOITY','PUBNEWFN','PUBPHOTO','PUBOPP','DOMAIN','RDATE_3','RDATE_4','RDATE_5','RDATE_6','RDATE_7','RDATE_8','RDATE_9','RDATE_10','RDATE_11','RDATE_12', 'RDATE_13','RDATE_14','RDATE_15','RDATE_16','RDATE_17','RDATE_18','RDATE_19','RDATE_20','RDATE_21' 'RDATE_22','RDATE_23','RDATE_24','LASTDATE', 'ADATE_3','ADATE_4','ADATE_5','ADATE_6','ADATE_7','ADATE_8','ADATE_9','ADATE_10','ADATE_11','ADATE_12', 'ADATE_13','ADATE_14','ADATE_15','ADATE_16','ADATE_17','ADATE_18','ADATE_19','ADATE_20','ADATE_21','ADATE_22','ADATE_23','ADATE_24','DOB','MAGFEM'

4. Feature Selection

4.1 Correlations

In order to understand the relationship between the numerical variables, we decided to compute the correlations using the *Pearson* method.
To avoid redundancy we decided to drop some of the most correlated features, we consider the variables with a correlation above 0.7 very correlated variables, taking into account this specific dataset.
To perform this task we used the following code, which allow us to compare the correlation of attribute with the remaining:

```
#the matrix is symmetric so we need to extract upper triangle matrix without diagonal (k = 1)
sol = (corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))
                  .stack()
                  .sort_values(ascending=False))
```

Fig. 8

Output:

HHAGE1	HHAGE3	0.993967
HV1	HV2	0.993472
ETH2	ETHC5	0.989762
POP901	POP902	0.988626
	POP903	0.981371
HHD7	HHD9	0.980798
LFC3	LFC5	0.979790
AFC1	AFC2	0.979374
POP902	POP903	0.978398
AGE905	AGE906	0.976823
IC1	IC3	0.975413
DW4	DW5	0.974199
IC2	IC4	0.973510
IC3	IC4	0.971566
RHP1	RHP2	0.971507
LFC2	LFC4	0.971312
DW5	DW6	0.971251
IC14	IC23	0.969128
AGE902	AGE903	0.966790

Fig. 9

Dropped variables:

'HC18','HU2','AGE906','HHAGE3','HV2','ETHC5','POP902','HHD9','POP903','LFC3','AFC2','IC3','DW5','IC4','LFC2','RHP2','IC23','HHP1','AGE903','HHN4','HHP2','AGE905','LSC2','OCC9','IC20','HHN3','HUPA2',
'AGE904','ETHC4','DW2','AFC5','IC19','IC21','CARDPROM','HUPA6','AGE902','HV4','HC8','HVP2','IC22',
'HVP1','AGE907','DW6','HHD3','LFC1','HHAGE1','HHD5','RP2','CARDGIFT','HHN6','HUR2','MARR3','HVP4','HHD4','TPE4','HHD11','ETH13','AGEC6','HHD3','HVP6','IC15','ETHC3','RP1','IC18','RP4','HC5','IC16','HC6','HHD1','LSC3','HC21','HHAGE2','CHIL1','HUPA1','OEDC2','IC17','HC18','HU2','AGE906','HHAGE3','HV2','ETHC5','POP902',
'HHD9','POP903','LFC3','AFC2','IC3',
'DW5','IC4','LFC2','RHP2','IC23','HHP1','AGE903','HHN4','HHP2','AGE905','LSC2','OCC9','IC20','HHN3','HUPA2',
'AGE904','ETHC4','DW2','AFC5','IC19','IC21','CARDPROM','HUPA6','AGE902','HV4','HC8','HVP2','IC22','HVP1','AGE907','DW6','LFC1','HHAGE1','HHD5','RP2','CARDGIFT','HHN6','HUR2','MARR3','HVP4','HHD4',
'TPE4','HHD11','ETH13','AGEC6','HHD3','HVP6','IC15','ETHC3','RP1','IC18','RP4','HC5','IC16','HC6','HHD1',
'LSC3','HC21','HHAGE2','CHIL1','HUPA1','IC17','HVP5','OEDC3','HHD2','HHD6','IC6','ETHC6','IC14','OEDC1','MC2','ETHC2','HVP3','HC7','AFC3','IC1','VOC2',
'VC3','OCC1','NUMPROM','POP90C5','HU1','HC19','HHN5','VC1','DW3','LFC4','MAXRAMNT','HC19','RP3', 'ETH1','LASTGIFT','HUPA3','HV3','EC1','POP90C3','DW7',
'SEC4','HHD10','MARR4','OCC11','HC11','HV1','VC3','DW4','AGEC7','OCC2','EC7','LFC6','ETH5','MALEMILI','VOC1','POP90C4','HC4','LSC1','AC2','AC1',
'DW1','MINRAMNT','HU3','ETH4','AFC4','HHAS2','MC1'

NOTE:
Accordingly to the changes performed we updated the list of *Metric* and *Non-metric features*.

4.2. Standardize the data frame:

The goal of this section is to standardize the range of the continuous variables in order that each of them contributes the same to our analysis, we choose the *MinMax scaler* since this method will preserve the shape of our dataset. In the first approach we used the *Standard Scaler*, which standardizes features by removing the mean and scaling to unit variance, in order to remove the differences between the features but, since our data do not have a normal distribution, the *StandardScaler() algorithm* was hard to interpret. For this reason our team opted to perform the standardization using the *MinMaxScaler() algorithm*. Below it is possible to observe a sample that shows how the features look like after standardization:

	HIT	RAMNTALL	TIMELAG	MALEVET	VIETVETS	WWIIVETS	POP90C1	ETH2	CHIL2	AGE901
0	0.000000	0.023965	0.003676	0.393939	0.343434	0.181818	0.0	0.010101	0.424242	0.464286
3	0.008299	0.010135	0.008272	0.232323	0.141414	0.313131	0.0	0.000000	0.353535	0.380952
4	0.248963	0.025443	0.012868	0.282828	0.090909	0.535354	1.0	0.989899	0.434343	0.392857
6	0.000000	0.009924	0.003676	0.333333	0.363636	0.343434	0.0	0.000000	0.414141	0.416667
9	0.000000	0.001584	0.006434	0.282828	0.515152	0.141414	0.0	0.262626	0.424242	0.321429
...	...	...	...	...	...	...	...	...	...	...
95388	0.008299	0.011613	0.003676	0.303030	0.434343	0.484848	1.0	0.020202	0.373737	0.440476
95392	0.000000	0.001267	0.008732	0.363636	0.242424	0.353535	0.0	0.000000	0.404040	0.404762
95397	0.000000	0.012352	0.009191	0.323232	0.343434	0.363636	1.0	0.010101	0.353535	0.321429
95402	0.000000	0.002745	0.001838	0.202020	0.212121	0.636364	0.0	0.020202	0.414141	0.404762
95407	0.000000	0.001267	0.008732	0.363636	0.474747	0.111111	1.0	0.101010	0.373737	0.333333

Fig. 10

4.3. Encoding:

In order to try to gain explainability of the clusters (that will be built) we used a simple code to perform *One Hot Encoding* in the *Non-metric features*.
The result of encoding the categorical variables with *One Hot Encoding* was the creation of a column for each different category in every categorical variable. These new columns are binary, having the values 0 or 1 depending on its category in that variable. At the end of this step, these initial categorical features are removed from the datasets because they are already represented with the new created columns. Below it is possible to observe a sample that shows how the features look like after encoding:

	x0_Army	x0_Couple	x0_Doctor	x0_Elite	x0_Judge	x0_Man	x0_Religion	x0_Undifined	x0_Woman	x1_AA	...
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
9	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
95388	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
95392	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...
95397	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...
95402	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
95407	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...

Fig. 11

4.4. Dimensionality Reduction:

4.4.1. First Approach:

In order to reduce the dimensionality of the dataset, our team performed the *Principal Component Analysis*. *PCA* transforms a large set of variables into a smaller one, preserving as much information as possible.

There are as many *PC´s* as there are variables in the data and *PCs are* constructed in such a way that the first principal component accounts for the largest possible variance in the data set. The goal is to maintain as many *PC´s* as needed to preserve a large variance (~80%), representative of the original variables. Since we choose to

standardize using the *minmax Scaler*, our reference threshold value to check in the *Eigenvalue* was 0.012448. According to the generated table (showing *Eigenvalue* and the variance) we should maintain approximately 20 components.
After using those components we noticed that a lot of them were meaningless so we reduced the number of *PCs to* maintain to only 12 but they are still being poor in terms of information provided, so hard to interpret each *PC*.

4.4.2. Final Approach:

Once the *PCA* approach did not led us to successful results, due to lack of meaningful interpretation, our team made the decision to use only some of the *metric features* as the cluster input, to make this choice we used not only our domain knowledge but also taking in consideration the variables correlation between themselves.
Thus, in order to build the input of the *clustering algorithms*, our team selected 19 metric variables. This selection was made keeping in mind the two perspectives - or views- we want to study, being them the Socio Economic view and the *PVA* interest view.
Regarding the first perspective, we decided to choose the following variables: ***'POP90C1'***, ***'ETH2'***, ***'CHIL2'***, ***'AGE901'***, ***'MARR1'***, in order to describe the social context of the donors, since the ***'POP90C1'*** and ***'ETH2'*** indicates the residence area and the respective ethnical context, ***'CHIL2'***, ***'AGE901'***, ***'MARR1'*** indicates if the donor lives in a place where people have children, the median age and also the marital status. The remaining variables,***'RHP4'***, ***'IC2'***, ***'HHAS4'***, ***'EIC9'***, ***'EC8'***, ***'VOC3'***,***'MHUC1'***, give information about the economic level of the place where the donor lives. So the median number of people per room, the average income of the families, the indicator of poverty, the level of the qualification of the professions, if the education is high, if the families have 3 vehicles and the mortgage value of the houses.

For the second perspective our team choose the following variables: *T*aking into consideration the *RFA- Recency Frequency Amount-* analysis our team considered useful to maintain the following 3 variables:***'TIMELAG','NGIFTALL', 'RAMNTALL',*** meaning the period between the first and the second gift, the number of lifetime gifts and the total amount gifted. Being this study about *PVA*, our team believes that it is worth analyzing who are the ones who relate and identify more with the cause. To answer our question, the variables ***'MALEVET'***, ***'VIETVETS'***, ***'WWIIVETS'***, were used to enable us to know who lives in places that have veterans as neighbours. We also considered the ***'HIT'*** variable an important one since it can show people who are willing to accept more e-mail offers in general.

4.5. Correlation checking - Pre clustering features:

The correlation between variables can vary between -1 and 1, being -1 a perfect negative correlation (when one occurs the other does not) and 1 a perfect positive correlation (if one occurs so also the other). For this reason, all the correlations with a value of 1 happen when a variable is compared to itself.
As it is possible to confirm, there are not highly correlated features within the ones that will be used in the clustering step. This way, the maximum correlation we can observe between variables has a value of 0.7 and the minimum correlation appearing has a value of -0.6.

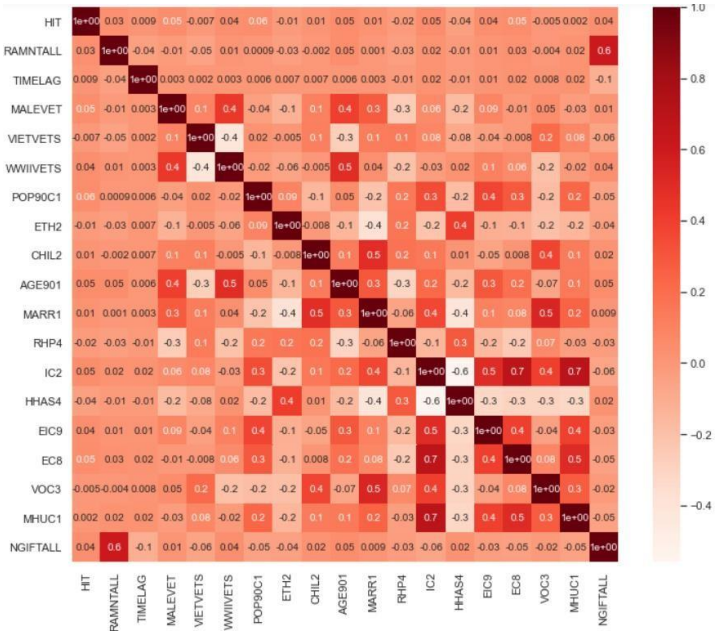


Fig. 12

Note: Descriptive statistics of the attributes that will be the clustering algorithms input

	HIT	RAMNTALL	TIMELAG	MALEVET	VIETVETS	WWIIVETS	POP90C1	ETH2	CHIL2	AGE901
count	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000
mean	0.013787	0.009584	0.007504	0.307880	0.299910	0.330099	0.589310	0.075591	0.387506	0.410331
std	0.037845	0.012111	0.006528	0.115700	0.152542	0.178706	0.479761	0.169340	0.064739	0.099247
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.002851	0.003676	0.252525	0.202020	0.212121	0.000000	0.000000	0.363636	0.357143
50%	0.000000	0.006862	0.006434	0.313131	0.292929	0.323232	1.000000	0.010101	0.393939	0.392857
75%	0.012448	0.012458	0.010110	0.373737	0.393939	0.434343	1.000000	0.050505	0.424242	0.440476
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig. 13

MARR1	RHP4	IC2	HHAS4	EIC9	EC8	VOC3	MHUC1	NGIFTALL
92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000	92549.000000
0.586840	0.109783	0.256704	0.108905	0.068911	0.073670	0.202054	0.384119	0.058936
0.131151	0.029573	0.114672	0.100995	0.046998	0.070828	0.109525	0.167328	0.058226
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.525253	0.100000	0.182000	0.040404	0.040404	0.030303	0.121212	0.285714	0.013699
0.616162	0.100000	0.236000	0.080808	0.060606	0.050505	0.191919	0.380952	0.041096
0.676768	0.125000	0.308667	0.151515	0.090909	0.101010	0.272727	0.428571	0.082192
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig. 14

5. Clustering:

5.1. Cluster Perspectives:

5.1.1. Socio-economic perspective:

With the usage of the Socio-Economic perspective we aim to understand from which donors can the institution expect to earn more or to earn less. This way, to take the

maximum benefit from each donor, there is the need to segment the marketing campaigns accordingly with the socio-economic environment of each one. In this perspective it is possible to identify the level of wealth and also the conditions that surround each donor.

Taking into consideration the characteristics of our dataset, as the size, we believe the best approach is to use SOM as a clustering algorithm. In SOM, each neuron is a vector in the input space and, during the training, their position is adjusted and with them also came their neighbors. The input patterns are compared to each neuron and are finally assigned to one of them. Thus, the 'winner' neuron is updated and, with him, their neighbors.

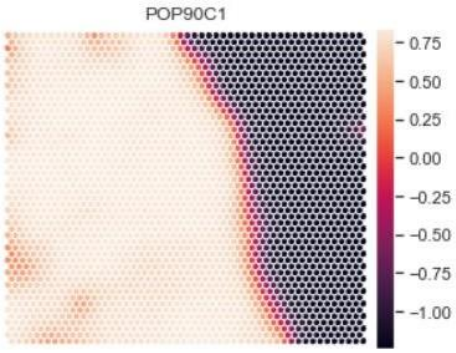


Fig. 15

Component Planes: In order to have an overview about our features distribution after applying the SOM algorithm, we plotted the component planes for all the features in this perspective. Our goal, besides to have an overview, was to check if the variables were representative / showed differences and also we confirmed if they seem much correlated with each other, which was not the case. E.g. We provide an example regarding the ***'POP90C1'*** variable, which shows to be a good feature to maintain in our perspective since it shows differences (the more light means people living outside the city, the transition colors indicates peripheries and the dark colors indicates living in the city). Note: In the appendices it is provided all the component planes.

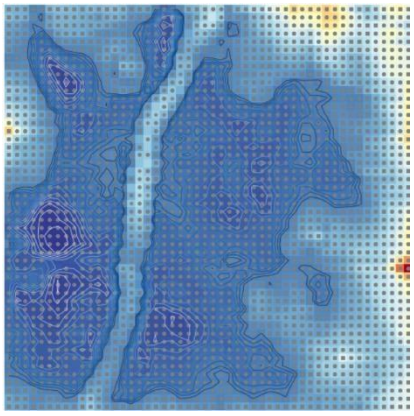


Fig. 16

U-Matrix: This matrix was performed in order to provide us a visualization about the distribution of the density in our data. This matrix suggests we consider 2 or 3 clusters and, after a trial and error process, our team decided to keep 3 clusters.

Hierarchical Clustering Algorithm: Our strategy passes through performing the SOM algorithm and then applying a hierarchical technique on the top, as the good practices recommend. This way, we used the AgglomerativeClustering algorithm before assigning labels to each observation. In the hierarchical clustering we can also confirm that 3 is a good number of clusters for our solution, so we will use it as input for the nodes and labels steps.

The final clustering result from the Socio-Economic perspective is represented in the table below:

label	POP90C1	ETH2	CHIL2	AGE901	MARR1	RHP4	IC2	HHAS4	EIC9	EC8	VOC3	MHUC1
0	0.579990	0.078228	0.391689	0.327674	0.590603	0.164485	0.248148	0.112550	0.067283	0.069957	0.199898	0.371539
1	0.000000	0.050505	0.362412	0.140975	0.556780	0.309764	0.199455	0.187022	0.027242	0.042853	0.046526	0.303030
2	0.977418	0.023058	0.394950	0.364649	0.638943	0.153612	0.559596	0.037539	0.127085	0.192264	0.328396	0.837931

Fig. 17

Clusters interpretation:

- 0- live in an urbanized area and have an income within the average;
- 1- live outside urbanized areas, have an income within the average, and are the ones presenting less schooling;
- 2- live in an urbanized area and have an income above the average and present more schooling. This cluster is the one with more Majors.

Cluster	Name
0	Urbanized_Median_Class
1	Unurbanized
2	Urbanized_High_Class

Table 2

5.1.2. PVA Interest perspective:

For the PVA Interest we want to study how donors were grouped according to an *RFA* point of view. We also aim to understand who lives in areas where there is a strong presence of veterans (from different types) because probably those people are more sensitive to the campaigns.Besides the different types of veterans and donors our team also believes that it is important to understand who accepts or not e-mail offers, in general.

For the exact same reasons from the perspective above -Socio-Economic- we believe the best approach is to use SOM as a clustering algorithm as well.

Component Planes: In order to have an overview about our features distribution after applying the SOM algorithm, we plotted the component planes for all the features in this perspective. E.g. we provide an example below regarding the ***'WWIIVETS'*** variable, which shows to be a good feature to maintain in our perspective since it shows differences (the more dark means people living in areas with a stronger presence of veterans of WWII, the transition colors indicates moderate presence and the light colors indicate absence of these people).
Note: In the appendices it is provided all the component planes.

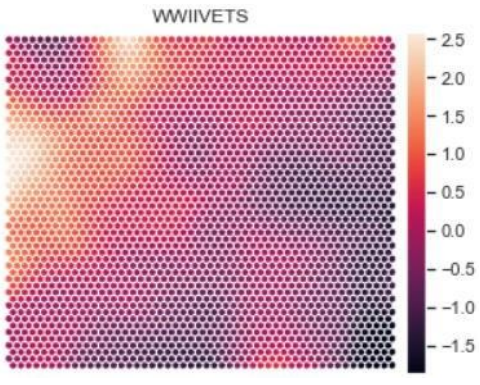


Fig. 18

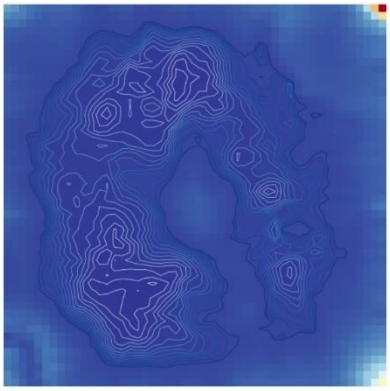


Fig. 19

U-Matrix: This matrix was performed in order to provide us a visualization about the distribution of the density in our data. The interpretation of this matrix is ambiguous in a matter of visualizing clusters. Thus, our team approached solutions with 2, 3 and 4 clusters and after a trial and error process, we ended up by choosing the solution with 3 clusters.

Hierarchical Clustering Algorithm: Our strategy here also passes through performing the SOM algorithm and then applying a hierarchical technique on the top, as the good practices recommend, with AgglomerativeClustering algorithm.

The final clustering result from the PVA_interest perspective is represented in the table below:

label	HIT	RAMNTALL	TIMELAG	MALEVET	VIETVETS	WWIIVETS	NGIFTALL
0	0.015156	0.009609	0.007481	0.312989	0.197776	0.451240	0.058293
1	0.009295	0.009837	0.070974	0.180606	0.029495	0.229091	0.042740
2	0.012717	0.009498	0.007494	0.305310	0.389915	0.228222	0.059145

Fig. 20

Clusters interpretation:

- 0- in this cluster are the donors who donate a smaller amount, those who donate less often, basically the RFA is small and exists many donors from the WWII;
- 1- in this cluster are the donors who donate a bigger amount, those who donate more often and where exists more majors donors and more vietnam veterans;
- 2- In this cluster are the donors who give a large amount, those who have the smallest value of ***'timelag'***, it is those who take less time to donate, where there are more male veterans, so, more veterans in general. Due to the fact that they have a high ***'ngiftall'*** value and a low ***'timelag'*** value, we can assume that these refer to more "recent" (and good) donors. Through the *encoding* performed previously we can affirm that we were able to analyze the ***'income'*** for this cluster, thus concluding that these are the donors who have the highest ***'income'*** values.

Cluster	Name
0	Bronze_WWII
1	Gold_VIET
2	Silver_NEW

Table 3

5.2. Joining both perspectives:

A simple code in Python allowed us to combine the information from both perspectives in order to understand the position of each donor - observation - in each view. The result table was the following one at right.
In order to provide an easier interpretation we build more explanative tables as shown below:

	socio_economic_label	PVA_interest_label	counts
0	0	0	40342
1	0	1	23
2	0	2	47439
3	1	0	3
4	1	2	30
5	2	0	1514
6	2	1	2
7	2	2	1731

Fig. 21

Sub-segment	Socio-economic	PVA Interest	Count
0	Urbanized_Median_Class	Bronze_WWII	40342
1	Urbanized_Median_Class	Gold_VIET	23
2	Urbanized_Median_Class	Silver_NEW	47439
3	Unurbanized	Bronze_WWII	3
4	Unurbanized	Silver_NEW	30
5	Urbanized_High_Class	Bronze_WWII	1541
6	Urbanized_High_Class	Gold_VIET	2
7	Urbanized_High_Class	Silver_NEW	1731

Table 4

In this table we can understand where do our donors are positioned regarding both our prespectives, we can see that the majority of them are in the sub-segments 0 and 2 and for that reason we must be extremely careful when approaching them

6. Marketing Approaches:

This marketing approach intends to apply the previously performed cluster analysis. We have different marketing approaches for different sub-segments but we will join a few in order to make it simpler to our marketing team.

- Sub-segment 6 and 1 are our best customers and are very sensible to campaigns regarding the Vietnam war, so we can send them promotions on the 30th of April which marks the end of the Vietnam war;
- Sub-segment 2, 4 and 7 are our Silver customers and we should interact with them on a regular basis in order to get their engagement back, also it would be interesting to send them a questionnaire to understand what went wrong in our relationship;

- Sub-segment 0, 3 and 5 are in theory our worst customers but we may try one last time to get them back, since they might be sensible to WWII related promotions, we can send them a promotion on the 2nd of September which marks the end of this war.

7. Conclusion:

From the analysis of clusters performed previously the different customers profile were identified from two perspectives: Socio-economic and PVA Interest.
After approaching each customers' groups taking into account its characteristics, possible combinations were made for the study of various campaigns.
Taking into consideration the proposed marketing plan it was concluded that the main focus should be the customers belonging to the sub-segment 6 and 1 and the sub-segment 2,4 and 7. In our opinion, it would be more profitable to invest in customers from these sub-segments for the reasons presented before.

8. References:

- Lucas Bação, Fernando.2020,November 23 - "Self-Organizing Maps";

- Lucas Bação, Fernando.2020, November 9 - "Hierarchical Clustering";

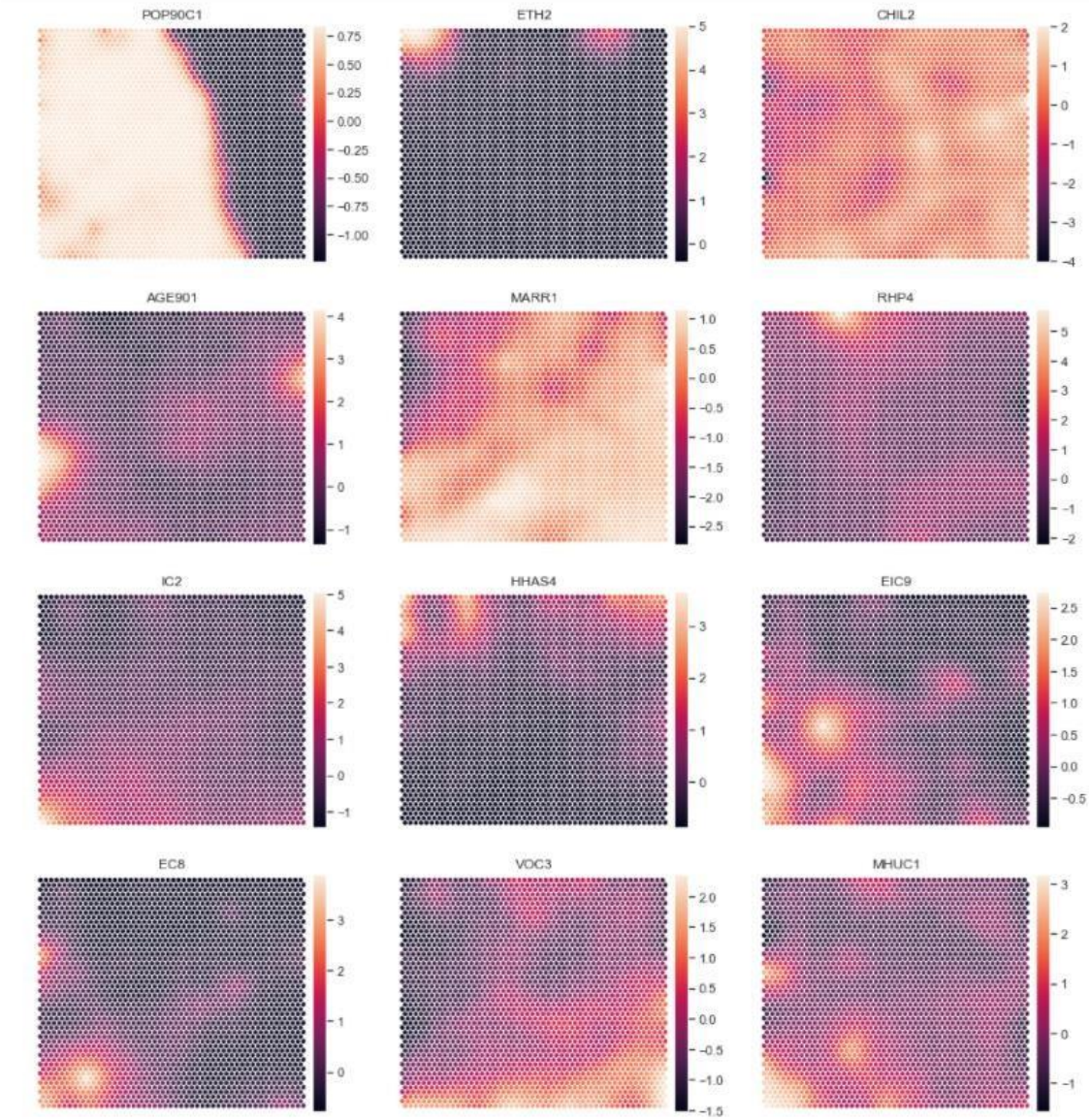
- Silva, David. "Self-Organizing Maps", notebook;

- Silva, David. "data-preprocessed3", notebook;

- Standardization in Cluster Analysis. (2019, August 3) Alteryx Community https://community.alteryx.com/t5/Alteryx-Designer-Knowledge-Base/Standardization-in-Cluster-Analysis/ta-p/302296

9. Appendices:

- ✓ Component Planes for the Socio-Economic perspective clustering:


POP90C1
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
ETH2
5
4
3
2
1
0
CHIL2
2
1
0
−1
−2
−3
−4
AGE901
4
3
2
1
0
−1
MARR1
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
RHP4
5
4
3
2
1
0
−1
−2
IC2
5
4
3
2
1
0
−1
HHAS4
3
2
1
0
EIC9
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
EC8
3
2
1
0
VOC3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
MHUC1
3
2
1
0
−1

✓ Component Planes for the PVA_Interest perspective clustering: