Data Mining Project

MASTER'S DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

A2Z INSURANCE

Group B

Afonso Quintino, number: 20220698

Ana Sofia Mendonça, number: 20220678

Beatriz Sousa, number: 20220674

January, 2023

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

INDEX

1. Introduction

This report focuses on identifying customer segments as the final project in the Data Mining course. For the development of the project, a hypothetical situation was given, where we had to analyse a group of customers from an insurance company, A2Z Insurance, with different value and demographic information, and divide them into different segments, according to these variables. In other words, our main concern is answering the question "What are the different types of customers that this insurance company has?".

In the following pages, we present and discuss the entire process of the project development including data treatment, clusters created, decisions made, and major findings. We will be using various clustering techniques to find the most suitable method for our data, and we will be evaluating the results to make sure that the segments formed are meaningful and accurately represent the relationships within the data. Overall, this project will allow us to make more informed decisions and predictions based on the insights we gain from the clustering analysis.

2. Exploration

We started by checking the number of elements in our dataset. In table 1 is presented the initial 13 variables and a brief description. After checking a concise summary of the dataframe, we can see that variables *FirstPolYear*, *BirthYear*, *EducDeg*, *MonthSal*, *GeoLivArea*, *Children*, *PremMotor*, *PremHealth*, *PremLife* and *PremWork* have missing values. Regarding the types of variables, everything seems fine.

Following this, we inspected the main descriptive statistics for numerical and categorical variables, as shown in tables 2 and 3. The first problem found was the maximum value of the variable *FirstPolYear*, as this variable stores the year of the customer's first policy, the value 53784 does not make any sense. The same date problem happens with *BirthYear* since 1028 is not an admissible year to be born, and still have active insurance in our company. Looking at *MonthSal* we conclude that this company provides insurance to a wide range of customers, since the minimum and maximum values of gross monthly salary are far apart, and the standard deviation is big. Additionally, in *ClaimsRate*, the third quartile is far apart from the maximum value of this variable, which can indicate the presence of outliers. This also happens with *PremMotor*, *PremHousehold*, *PremHealth*, *PremLife*, and *PremWork*. Lastly, most of our dataset has a bachelor's or a master's degree and there are 4 classes of different education levels.

For non-metric variables, we created count plots, to extract useful information, shown in the figure 1. We see that most of the clients in our dataset have children and a greater number of clients have a bachelor's degree or a master's degree, being the ones with a Ph.D. a minority. Since we do not have information about the meaning of the codes for *GeoLivArea* we cannot extract useful information from this variable.

By analysing the histogram plots of the numeric features shown in the figure 2, we again notice, the presence of outliers in multiple variables. Extreme observations are present in almost every histogram plotted. When using boxplots to represent the numeric variables, shown in the figure 3, the same happens, as the distributions are highly skewed. So, after solving the problem with outliers, we should plot these histograms and boxplots again and see the differences and the real distributions of the data.

For relations between the different variables, we used a pair plot, a pair plot distinguishing the observations with or without children, and violin plots. The problem here was that no useful information was given as the dataset still has too many outliers and noise. So, we should redo these plots after the preprocessing phase. We finalized our exploration phase, by using pandas profiling, to have a report on the plots, missing values, outliers, etc.

3. Preprocessing

3.1. Duplicates

We started by checking the duplicated observations, and there were three, so they were dropped.

3.2. Data types

As seen previously, in the exploration phase, the variables were stored correctly. But, with pandas profiling, we noticed that variable *EducDeg* was 'rejected unsupported', which means that either the variable has mixed types or is constant (thus not suitable for meaningful analysis) or cannot be analysed (type is not supported, has mixed types, has lists, is empty or wrongly formatted) (pandas-profiling, 2023). It was assumed that the variable should be encoded in another way, and we did it with Ordinal Encoder, assigning to the old categories' integer numbers, maintaining the implicit order that is present in the different education categories. After the transformation: "b'1 - Basic'" corresponds to 0, "b'2 - High School'" corresponds to 1, "b'3 - BSc/MSc'" corresponds to 2, and "b'4 - PhD'" corresponds to 3, keeping the missing values unchanged.

3.3. Incoherences

During the exploration phase, it was noticed that there were two strange values, one in *FirstPolYear* (53784) and another in *BirthYear* (1028), and as the values are not admissible and unfeasible, we should search for other values higher than 2016 in *FirstPolYear* (the database is from 2016, so we cannot have information on the future), and lower than 1896 in *BirthYear* (assuming that it is not normal to have 120 years). Those observations were deleted. It was checked if there were customers that pay higher premiums than the income they receive (and we deleted one observation referent to CustomerID number 9150). We also checked for customers under 16 years that already have a salary (the minimum legal age to work in Portugal is 16 years old, so having a salary and being younger than that should not happen), and the observations were deleted, and for customers under 16 years that already had a degree (and there weren't).

Furthermore, we can check if there are clients that started to be our clients before they were even born, and there are 1997 people (almost 20% of our dataset). It is not sensible to remove that many observations, since there would be a huge information loss, and considering that we cannot talk to the person that gathered the data and ask how these two variables were recorded, we can assume some things: 1. assume that the insurance belongs to the object that is being insured (like the car or the house, for example), and that when it changes its owner it is acceptable to think that the insurance is older than the owner if the object is so; 2. assume that the value of the *BirthYear* was badly imputed and change it to the *FirstPolYear* (as we assume that the records from our own company are more trustable than the records we get from our customers); 3. assume that the value of the *BirthYear* was badly imputed and remove this column; 4. assume that the value of the *BirthYear* was badly imputed,

assign it to NaN, and then later treat them as missing values. The 4th option was used as it seems the most reasonable in this case.

3.4. Normalization

Since we will use methods that are based on measures of how far apart data points are, we need to scale our data. By scaling our variables, we can compare different variables on equal footing, because, for these algorithms, a change of "1" in any numeric feature is given the same importance (Cook, 2021). We started by trying min-max scaling, as it will preserve the shape of the dataset (no distortion) and it is the least disruptive to the information in the original data. Then, we tried the standard scaling, however, the problem with this type of scaling was that it should only be used when we know that the data distribution is normal. Lastly, we used robust scaling that scales the data according to the quantile range, so outliers will have less influence than in other scaling methods. We decided to keep the standard scaled dataset as most of the metric features have a bell-shaped distribution.

3.5. Missing values

As we know, missing values are usually expressed with NaN, but it was further checked if there could be other expressions for missing values (like empty strings, etc.), and none were found. We tried to detect some relationships between missing values such as missing together with other columns or other patterns and to achieve that a visual representation of missingness in data was done (with *msno.matrix* see figure 4). We could not find a pattern in the missing values, but it was noticed that some observations had multiple missing values in different features.

So, one thing we could do is just delete observations or features, but "as a general rule of thumb, only features that are missing in excess of 60% of their values should be considered for complete removal" (Kelleher, Namee, & D'Arcy, 2015), which is not the case, and we would be losing information.

Therefore, we tried to use an adequate measure of central tendency, which in our case is the mode for non-metric features and the median for metric features (as it is less sensitive to outliers or skewed distributions). However, imputing in this way can also affect the true relationship between columns, can bias the standard error, and if values are not missing at random can also bias the actual median/mode of the column. Another way to deal with missing values is to fill in with values from similar individuals (nearest neighbours). *KNNImputer* was used in the scaled data only for metric features. We also tried to fill the missing values a measure of central tendency of a certain subset, for example, replacing the missing values with the median/ mode value for customers in the same category as that of the given observation.

When we replace the missing data with some common value, we might add some bias to our data, so, different prediction models to input missing values were tried (linear regression, Naïve Bayes classifier for categorical data), therefore the information of other variables was used to predict the missing values in a variable.

3.6. Outliers

We tried three different methods to identify outliers: the IQR method, the z-score method, and manual filtering. Using the interquartile range, an observation was an outlier if it had an absolute value 1.5 times greater than the IQR (Chaudhary, 2019), but we would have to remove almost 30% of our data, which was considered too much, and therefore, was not applied. Regarding the z-score method, we

calculated the z-score for each observation and one observation was considered an outlier if it had a z-score value higher than 3, or lower than -3 (Geeks for Geeks, 2020). For the manual approach, we defined some conditions that our observations had to satisfy to not be considered outliers with help of a dynamic boxplot. The defined rules can be found in table 4.

We also tried DBSCAN which works by identifying points in the dataset that are closely packed together and labelling them as a cluster and identifying points that do not belong to any cluster and labelling them as outliers. The parameters we can change are MinPts (which is the minimum number of points required to form a cluster) and ε (which is the maximum distance between two points in a cluster). To identify outliers with DBSCAN, we set a large value for ε, a value of 1.2, and a small value for MinPts, we chose the number of metric features, causing the cluster 0 to be very large including most of the points in the dataset, and any points that are not included in this cluster were considered as outliers, and labelled as cluster -1. Around 4% of our data was removed by eliminating these 396 outliers.

Initially, we kept the manual filtering approach and stored the outliers, but this choice changed throughout the process of clustering, and we ended up choosing DBSCAN method to identify outliers.

3.7. New features

To extract relevant and more useful information that suits our investigation, we need to create several new variables using the ones initially given. We do this to make sure we improve our model's performance by finding potential new information. The information on the created variables can be found in table 5.

3.8. Variable selection

In favour of getting the best possible model for our problem, we need to select the variables that are statistically more relevant to our case. So, first, we checked for univariance of any feature and there were none. Knowing that the larger the input space, the more data and computing power is needed, we kept our focus only on relevant features, making sure there is no redundancy in our data.

We can identify highly correlated variables using Spearman's correlation coefficient and choose one variable from each correlated pair (choosing the variable that is more relevant to the problem). Also, we should remove any remaining redundant variables that are not contributing significantly to the clustering solution. A Spearman's correlation matrix was plotted (figure 5) for the numeric variables, as it accesses the strength and direction of the relationship between two variables that do not need to be necessarily linear. Looking at the correlation matrix we notice that: both *Age* and *BirthYear*, *FirstPolYear*, and *YearsClient* are perfectly negatively correlated, which was expected since those two pairs of features represent the same information but in different ways, so we will keep *Age* and *YearsClient*. *BirthYear* is correlated with *MonthSal* (-0.7), and with all the variables that represent income information (*Log10MonthSal*, *Log10AnnualSal*, etc.). The same happens with *Age* and *MonthSal* (0.7) as expected. Also, *CustMonVal* and *ClaimsRate* are highly negatively correlated (-0.99), and we dropped *ClaimsRate*, because it is referent from the 2 previous years, while the rest of the variables are annual. All the variables that represent income information are correlated, and we should choose *AnnualSal*, as the premiums information is also relative to a year and not a month. Regarding the premium variables, they are all correlated with each other, as expected, and we will keep the ratios on the total premiums (*PremMotorRatio*, *PremHouseholdRatio*, *PremHealthRatio*, *PremLifeRatio*, and *PremWorkRatio*) and drop the original ones and the ratios on the salary. That is because we want to

compare the proportion spent in each line of business and not the absolute value. We will also drop *TotalPremiumsRatio* because it is a highly correlated variable with many others, and this information will be present in *TotalPremiums* and *AnnualSal*.

This leaves us with the following metric features: *CustMonVal*, *PremWork*, *Age*, *PremMotor*, *AnnualSal*, *YearsClient*, *PremLife*, *PremHealth*, and *PremHousehold.* We were also left with the variables *Children* and *EducDeg,* as *GeoLivArea* is a useless variable since we do not have further information on the area codes.

3.9. One-hot encoding

Since many algorithms assume that the input data is numeric and cannot handle categorical variables directly, we performed one-hot encoding, which created new (binary) columns, indicating the presence of each possible value from the original data. However, this technique can also increase the dimensionality of the data, falling in the so-called 'curse of dimensionality'.

3.10. Principal component analysis

In PCA we search for k n-dimensional orthogonal vectors that can best be used to represent the data, so, the original data is projected onto a space, and then we can reduce the dimensionality of our data (Han, Kamber, & Pei, 2012). PCs are a new set of variables that are a linear combination of the original variables and are uncorrelated with each other. As we can see on the output PCA table 6, the proportion (explained variance ratio) is reducing along the components and can be useful for selecting the number of principal components to keep, because choosing the number of components that explain enough variance in the data, we can reduce the dimensionality while still retaining a significant amount of information. Therefore, we plotted (figure 6) this explained variance ratio against the number of principal components and used the elbow method to decide. The ideal number of principal components to retain is either 3 or 6, depending on the proportion of data variation that we want to keep. So, we performed PCA again with only 3 principal components, keeping in mind that the first three principal components explain only nearly 70% of the variance.

Looking at the dataframe with the correlations between PCs and metric variables (table 7), we see that PC0 is the most informative component, as it is highly correlated with 5 of our metric features. This component is affected by *PremMotor* since they have almost -1 in the correlation coefficient. PC1 explains the variables *Age* and *AnnualSal* (they are correlated as seen before, so this behaviour makes sense), while PC2 explains *PremHealth* and *CustMonVal*. Also, we can see that none of the Principal Components truly represent the variable *YearsClient*.

One of the main problems with PCA is that it is difficult to understand exactly what the transformed variables represent in terms of the original variables. PCA combines the original variables in a way that the PCs are ranked by their contribution to the variance in the data, rather than by their relevance to the context of the problem. Therefore, we decided not to use them.

3.11. Summary data exploration

After the preprocessing steps, we can redo the plots used in the exploration phase and see if we notice some additional issues (figures 8-10). Now, without too many outliers we can see the distributions of the variables. *PremHealth*, *PremMotor*, *CustMonVal*, and *AnnualSal* seem to have a bell-shaped distribution, while *PremHousehold*, *PremLife*, and *PremWork* have a higher concentration of small

values when compared to the high values. We do not notice any significant difference in the distribution of the metric features by each level of education, except for *PremMotor*, where we see that people with a Ph.D. tend to spend more on car insurance than people with only basic education. As for *Children*, we see that our customers without children tend to be older and to have a higher salary. Plotting again the correlation matrix for the chosen variables, we see that the significant correlations are *PremMotor* with the other premium variables and *Age* with *AnnualSal*. As these values are not high enough, we decided to keep these variables since they provide us with helpful information.

4. Modelling

4.1. Perspectives

Clustering by perspective refers to the practice of grouping data points into clusters based on how they are related to a particular point of view. This approach can be useful in a variety of contexts, such as when the goal is to understand how different groups relate to each other from a specific perspective. For example, in this case, we want to use clustering by perspective to group customer data by demographics, such as age, gender, and income level (to understand how these factors relate to customer behaviour) and to group customer data by value to the company, such as how much they spend in each type of insurance, etc. We divided our variables into demographic and value features according to their characteristics, as seen in table 8.

4.2. K-means

The first clustering algorithm we applied was K-means. Since the k-means algorithm needs the number of clusters we want to use a priori, we can decide it by plotting the distortion (a measure of how well the clusters fit the data, calculated as the sum of squared distances between each data point and its nearest centroid) against the number of clusters. Then, we can identify the "elbow" in the plot, that is the point at which the distortion begins to decrease more slowly and chose this number of clusters because as the number of clusters increases, the distortion will decrease at a diminishing rate. Another way to identify the ideal number of clusters is to compute the silhouette score for each point in the dataset, which is a measure of how well-separated the clusters are. We compute the nearest-cluster distance, subtract the intra-cluster distance and then divide this value by the maximum of these two values. Therefore, the silhouette score ranges from -1 to 1, and the bigger this score is, the better as it represents a better separation of clusters. We can choose the number of clusters that results in the highest silhouette score, because good clustering will result in data points that are well-separated from points in other clusters, and poorly separated from points within the same cluster.

For the demographic view, we started by evaluating the inertia plot and the best solution seems to be between 3 to 5 clusters, using the elbow method, while the best solution according to the silhouette plots is between 3 or 4 clusters, as after that the silhouette score starts to decrease due to some clusters beginning to pull to the negative silhouette coefficient values (indicating that the samples may have been assigned to the wrong cluster). After all, we decided to test the solution for both 3 and 4 clusters and chose the 4-cluster solution, based on the calculation of the r-squared, silhouette score, cluster characteristics, and violin plots. For visualization purposes, we made histograms and count plots for the distribution of the different variables in each cluster, where we can see most variables look well distributed between the clusters, except for *EducDeg* which looks the same for every cluster.

We followed the same process for the value features, ending up with 4 clusters, where the variables also look well distributed.

4.3. Hierarchical clustering

Hierarchical clustering is a method of clustering objects in such a way that each cluster is a hierarchy with several subclusters. In this method, each data point starts in its own cluster and then, at each iteration, the two closest clusters are merged into one (so this is a greedy algorithm). One of the advantages in using hierarchical clustering is that we do not need to specify *a priori* the number of clusters that we want to create, as we can use the dendrogram to help in this task. We only need to specify the linkage method, which is a way of determining how to merge the clusters in hierarchical clustering. There are multiple linkage methods such as single-linkage, complete-linkage, average-linkage and ward's method, and their usage will depend on the nature of the data and the requirements of the analysis: single-linkage clustering tends to produce long clusters, while complete-linkage clustering produces more compact clusters; average-linkage clustering and Ward's method tend to produce clusters that are more balanced and may be more interpretable in some cases.

We started by evaluating the r-squared scores for the different linkage methods. Overall, the best linkage method looks like Ward's, so that is what we will use. The core of hierarchical clustering lies in the construction and analysis of the dendrogram. The branches of the dendrogram represent the intermediate clusters that are formed as the objects are merged, and the length of the branches is proportional to the distance between the clusters that are being merged. So longer branches represent larger distances between clusters. For the demographic view, we decided to slice the structure with a cluster distance of 82, therefore the best option being 3 clusters, even though this is a subjective parameter, as we could choose to have a smaller distance between clusters, which would lead to a higher number of around 5 or even 6 clusters. Having this in mind, we decided to apply the algorithm to 3, 4 and 5 clusters and chose the 4-cluster solution, balancing the r-squared and the silhouette score. After applying the same steps for the value perspective, the best linkage method stayed the same and looking at the dendrogram we can say, again, that 3 clusters seem like a good option, but we can also try 4 and 5 clusters solutions. Considering the plots, r-squared and silhouette score, the final solution for the value view, was 5 clusters.

4.4. K-means and hierarchical clustering

We decided to aggregate the previous algorithms as we can take advantage of the strengths of each one. K-means is fast and efficient, but it can sometimes get stuck in local minima and produce suboptimal clusters, while hierarchical clustering is slower, but it produces a hierarchy of clusters that can be useful for understanding the structure of the data. Also, since we are not sure of what should be the optimal number of clusters to use, we can combine them and decide based on the dendrogram.

For the demographic view, we started by applying k-means in our data with 100 clusters, then we plotted the r-squared against the number of clusters used for different linkage methods to decide which one to use in the hierarchical clustering procedure. The average-linkage method was the best one, so we plotted a dendrogram for the k-means units. The best number of clusters was 3. The same was made for the value view, but the Ward’s linkage method was the best and the ideal number of clusters was 4. In the dendrogram for this perspective, we see that the blue cluster is joined late in the hierarchy, meaning that the observations in this cluster are more dissimilar to one another than the observations in the other clusters that are joined earlier in the dendrogram.

4.5. Self-organizing maps

Self-organizing maps are a type of unsupervised neural network that are particularly useful for clustering because they preserve the topological structure of the input data, meaning that similar observations in the input space will be mapped to nearby units in the SOM.

For the demographic view, we start by analysing the component planes (for a SOM with a 10x10 grid), and the higher values on *Age* and *AnnualSal* are represented on the top part, and it seems like these two features are correlated as the patterns are similar between the two component planes. Regarding *YearsClient*, this variable does not show any local correlations with the other variables but reveals higher values in the left region. We proceeded to visualize the U-Matrix and we can see possibly 4 clusters, one at each corner of the figure, as these are the zones with the lower distance between the units, as they are represented by darker colors. Another SOM representation is given by the hit map, we can notice again 4 clusters as explained before.

We can also use the emergent SOM approach, where a very large number of units is used, then we can interpretate the U-matrices and see the underlying structure of the data and combine it with other clustering algorithms (like k-means or hierarchical clustering). So, we started by training a SOM with a 50x50 grid and we plotted the distortion against the number of clusters to decide the number of clusters. After performing the k-means (with 4 clusters) on top of the 2500 units, we can see the 4 defined clusters and the respective units which they are associated with. For the hierarchical clustering on top of SOM units, after plotting the different r-squared values for the different linkage methods, we decided to choose the Ward's method, as it had the best results. After getting the dendrogram and slicing it, we ended up with the ideal number of 3 clusters and we applied the hierarchical clustering on top of our 2500 SOM units. Finally, we characterized the final clusters. For the value perspective, we essentially applied the same steps. The variables *PremHousehold*, *PremLife*, and *PremWork* have the same pattern in the component planes, which means that they are at some degree correlated. Also, we can see that *PremHealth* has the higher values on the bottom right of the figure, on the opposite side of the previous mentioned variables. Looking at the U-matrix, we can see possibly 2 to 3 clusters, one at each corner of the figure, and on the top right part of the matrix, we have units that are very distant from the rest. The hit plot also indicates the existence of 2 or 3 clusters. When using the emergent SOM approach, specifically the k-means on top of SOM units, we ended up with 3 different clusters represented on the self-organizing map, while for hierarchical clustering, we ended up choosing 3 clusters by using a Ward's dendrogram.

4.6. Mean-shift

This algorithm seeks to find the modes of a distribution, and it does that by shifting points in the distribution towards the mean of the points in their local region, until the points converge to a region where the mean does not change significantly anymore. Mean shift has the advantage of being able to find the modes of a distribution without having to specify the number of clusters in advance, unlike k-means for example, so we tried to use it. There is one fundamental parameter that is bandwidth that determines the size of the local region around each point that is used to compute the mean shift. Larger bandwidths result in larger regions, causing points to shift less, while smaller bandwidths result in smaller regions, which can lead to more sensitive shifts. So, when the bandwidth is bigger, we will have less clusters. To estimate this bandwidth, we used *estimate_bandwidth* function that needs the quantile as a parameter. The quantile represents the quantile of the distribution of distances between

points that is used to estimate the bandwidth (the distance between points falls below this quantile for most points). The results were 5 clusters for the demographic view and 4 clusters for the value view, and using violin plots, histograms, and count plots we were able to see the differences between the different clusters obtained.

4.7. Joining each view's solution

We started by checking the results from each of the methods we applied to see which ones have the highest r-squared and silhouette values (see tables 9 and 10). For both the demographic and the value perspectives we chose 4 clusters to compute using the k-means algorithm. We can have different plots to see the main characteristics of each cluster and the average of each variable by cluster can be useful to profile the clusters.

4.7.1. Demographic perspective

Cluster 0 represents clients that are younger as they have lower age values, lower years as clients and a lower annual salary. The clients represented by this cluster tend to be people that have children, rather than those who do not. This is our 2nd most frequent client. Cluster 1 seems to be clients that are older as they have higher age values and a bigger annual salary. From the older clients, this cluster represents the ones who recently joined the insurance as their years client is still on the lower side, compared to cluster 2. These clients can have children or not as their division in the *Children* values is almost the same. Cluster 2 is similar to cluster 1 by having older clients with higher salaries, but it differs in this cluster by having been clients for a longer time. They also almost equally distribute between having children or not. This is our least common client. Cluster 3 is our most common client, middle-aged with normal to low salaries who have been clients for a long time. These clients tend to have children. The education values stay the same because the variable wasn't used for clustering, as it is categorical (figures 11, 12, and 13).

4.7.2. Value perspective

Overall, the premiums spent on life, work and household tend to be lower as all their bar plots are skewed to the left. For cluster 0, these seem like our best clients as they tend to spend the most premiums throughout all the premium values apart from premiums spent in motors, where they have the lowest value compared to the other clusters. They have a normal distributed plot for *CustMonVal* which means it includes the clients who have the best monetary value but also the lowest. For cluster 1, these clients have a lower monetary value and spend the most premiums on motor insurance. These clients spend a low amount on household, life, and work insurance, they also tend to not spend a lot on health, but it is still more than they spend in previous mentioned variables, apart from motors. For cluster 2, these clients have a lower to normal monetary value, spending the most premiums on health. The other premiums spent tend to be lower. And finally, for cluster 3, these are customers with similar characteristics to cluster 1, as they spend the most premiums on motor related insurance, with other premiums having similar values to cluster 1. The difference seems to be that these clients have a higher monetary value (figures 14, 15, and 16).

4.7.3. Merging views

We tried two methods for merging the perspectives. One was to manually see the clusters that had less observations when compared to the remaining ones and change one of the labels (either the

demographic label or the value label) to merge the two clusters. We found that cluster (2,0) and cluster (1,0) should be merged as they have little observations. And we ended up with 15 clusters, which is still a large number. Therefore, we tried a more intuitive merging option, that included using hierarchical clustering applied to the centroids of the 16 clusters created before. Seeing the Ward's dendrogram (figure 17) was not sufficient to decide the ideal number of clusters, so we decided to try with 4, 5, and 6 final clusters and to see which are the results that make sense in the segmentation of the customers. Analyzing parallel coordinates, TSNE, count plots, and histograms, as well as the r-squared and silhouette score results (table 11) for all the three possibilities, we decided to keep the 5-cluster solution.

5. Results

Cluster 0 (figures 18-27) represents an older client, with a higher annual salary than other clusters of clients, there is a high range of years as client so we cannot be sure if they are older or newer clients, still, they tend to be clients for more than 20 years which is already significant. It makes sense that these clients spend the most premiums on health, compared to younger clients. The rest of the premiums spent tend to be on the lower side. They do not tend to have children.

When looking at Cluster 1, this seems to be a middle-aged client that has been a client for a long time and commonly has children. Their salary has a normal distribution, so it has a big range but is most commonly average. They spend most premiums on motor related insurance, while other values are low. Still, they are the second most valuable customer when having into account the monetary value.

For Cluster 2, it seems like these are customers that recently joined the company. It is hard to specify what age range the client belongs to, but the higher values are still middle-aged individuals, with more younger individuals than cluster 1, they also have normal distribution when it comes to salary. This makes sense, as a new client could belong to any age or salary group. Again, this customer seems to spend most of their premiums on motors. By analysing the radar chart, we can see that clusters 1 and 2 have the highest *PremMotor* values.

When it comes to Cluster 3, we can see that it amounts to slightly younger than average customers with a long-time connection to the company. Despite having very low-income levels, these individuals are the ones (by far) who bring the most value to the company. They pay high values in basically all the premium types, particularly in household, life, and work, which were well above the average. On the opposite side, these clients show very low levels when it comes to the premium on motor.

Finally, Cluster 4 gathers the youngest group of clients in the company as well as the ones with the lowest income. However, these individuals have been clients of the company for a long time despite having a low monetary value. The premiums paid by these clients seem to be around the average values except for motor, which is lower, and health, which seems to be higher than the average of the companies' clients.

6. Marketing strategies

For Cluster 0, it would be important to market our health insurance as it is the client that spends most on it, at the same time, as they are older, we must direct our marketing to platforms mostly used by that range of people like newspapers, TV, or *Facebook*. Moreover, these customers have a high salary

so we should reach out to them individually and propose customized insurance policies that are tailored to their specifics need, including insurances of other types beside health. We can create partnerships with higher luxury brands or even private institutions like hospitals to better cater for these clients' needs.

For Cluster 1 and Cluster 2 the biggest difference is that in the first, we have a long-time client whereas, in the second, the clients recently joined the company. Therefore, the strategy used can be essentially the same but may be delivered in different ways or have more perks for loyal clients. As both clusters spend more on motor and are likely to have children, we should advertise spending premiums in areas like life or household. For example, specifically, put advertisements where we know parents drive or in car repair shops. For Cluster 1, as they have been clients for a longer time, we can directly reach out to them and let them know our different insurance types offering promotions, as they already trust the company.

As Cluster 3 is the one that brings more value to the company, it is important to make sure it continues to be the case. Since these customers have been with the company for a very long time, we can assume it will be harder for them to change insurance companies and try to create innovative packages of premiums options while still making sure these clients are satisfied. For example, if we take either household, life, or work and couple them with motor, perhaps we might see an increase in the premiums paid in the latter category. It is also important to have in mind that these clients have lower incomes, so the company cannot raise the prices too much otherwise they will be in a complicated position and be enticed to trade the company for a more competitive one.

For Cluster 4, the company needs to be more active when it comes to announcing packages directly to the clients since they have been with the company for a long time but represent the lower monetary value, as they have low incomes. Special offers such as cheaper premiums for motor, household, life, and work might be the key to having the clients diversely increase their payment of premiums. This way they can pay less for more, thus increasing their value to the company but not compromising their finances.

7. Conclusion

In conclusion, we are aware the preprocessing and treatment of the features is a crucial step in obtaining these results, so we tried our best to treat the data, without removing a significant portion or without distorting the dataset. Regarding the context of the problem, some features could be better collected, like *GeoLivArea* could have more information about the categories. After trying many clustering algorithms, we ended up choosing the k-means algorithm for both the clustering perspectives, because of its simplicity and good results. Finally, we merged the clusters and tried to find useful insights to give to the insurance company, such as possible marketing solutions and data-driven advice. Finally, we used the cluster labels and a supervised method (decision tree) to classify new customers that were not considered during the segmentation process (the removed outliers). Overall, this study demonstrates the usefulness of clustering techniques in data mining and highlights the importance of carefully selecting the appropriate method depending on the task.

8. References

Chaudhary, S. (2019, September 28). *Towards Data Science*. Retrieved from Why “1.5” in IQR Method of Outlier Detection?: https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097

Cook, A. (2021, December 14). *Kaggle*. Retrieved from Scaling and Normalization: https://www.kaggle.com/code/alexisbcook/scaling-and-normalization

Geeks for Geeks. (2020, August 27). Retrieved from Z score for Outlier Detection – Python: https://www.geeksforgeeks.org/z-score-for-outlier-detection-python/

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining. Concepts and Techniques.* Elsevier.

Kelleher, J. D., Namee, B. M., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics.* Massachusetts: The MIT Press.

pandas-profiling. (2023). Retrieved from Concepts - Data Types : https://pandas-profiling.ydata.ai/docs/master/pages/getting_started/concepts.html

9. Appendix

Table 1 - Initial 13 variables

Variable	Description
CustID	ID
FirstPolYear	Year of the customer’s first policy, and this value considers when the customer is client for the first year
BirthYear	Customer’s birthday year (the database was collected in 2016)
EducDeg	Customer’s academic degree
MonthSal	Customer’s gross monthly salary (in euros)
GeoLivArea	Living area, but no further information was provided about the meaning of the area codes
Children	Binary variable: =1 if the customer has children, and =0 otherwise
CustMonVal	Customer monetary value, calculated by the formula: $lifetime\ value\ =\ (annual\ profit\ from\ the\ customer) \times (number\ of\ years\ that\ they\ are\ a\ customer) - (acquisition\ cost)$
ClaimsRate	Claims rate referent to the last 2 years, calculated by the formula: $amount\ paid\ by\ the\ insurance\ company\ /\ premiuns$
PremMotor	Premiums (in euros) in line of business: Motor (negative premiums are reversals that occurred in the current year, paid in previous one(s))
PremHousehold	Premiums (in euros) in line of business: Household (negative premiums are reversals that occurred in the current year, paid in previous one(s))

PremHealth	Premiums (in euros) in line of business: Health (negative premiums are reversals that occurred in the current year, paid in previous one(s))
PremLife	Premiums (in euros) in line of business: Life (negative premiums are reversals that occurred in the current year, paid in previous one(s))
PremWork	Premiums (in euros) in line of business: Work compensations (negative premiums are reversals that occurred in the current year, paid in previous one(s))

Table 2 - Descriptive statistics for numerical variables

Variable	Count	Mean	Std	Min	25%	50%	75%	Max
FirstPolYear	10266.0	1991.062634	511.267913	1974.00	1980.00	1986.00	1992.0000	53784.00
BirthYear	10279.0	1968.007783	19.709476	1028.00	1953.00	1968.00	1983.0000	2001.00
MonthSal	10260.0	2506.667057	1157.449634	333.00	1706.00	2501.50	3290.2500	55215.00
GeoLivArea	10295.0	2.709859	1.266291	1.00	1.00	3.00	4.0000	4.00
Children	10275.0	0.706764	0.455268	0.00	0.00	1.00	1.0000	1.00
CustMonVal	10296.0	177.892605	1945.811505	-165680.42	-9.44	186.87	399.7775	11875.89
ClaimsRate	10296.0	0.742772	2.916964	0.00	0.39	0.72	0.9800	256.20
PremMotor	10262.0	300.470252	211.914997	-4.11	190.59	298.61	408.3000	11604.42
PremHousehold	10296.0	210.431192	352.595984	-75.00	49.45	132.80	290.0500	25048.80
PremHealth	10253.0	171.580833	296.405976	-2.11	111.80	162.81	219.8200	28272.00
PremLife	10192.0	41.855782	47.480632	-7.00	9.89	25.56	57.7900	398.30
PremWork	10210.0	41.277514	51.513572	-12.00	10.67	25.67	56.7900	1988.70

Table 3 - Descriptive statistics for non-numerical variables

Variable	Count	Unique	Top	Freq
EducDeg	10279	4	b'3 - BSc/MSc'	4799

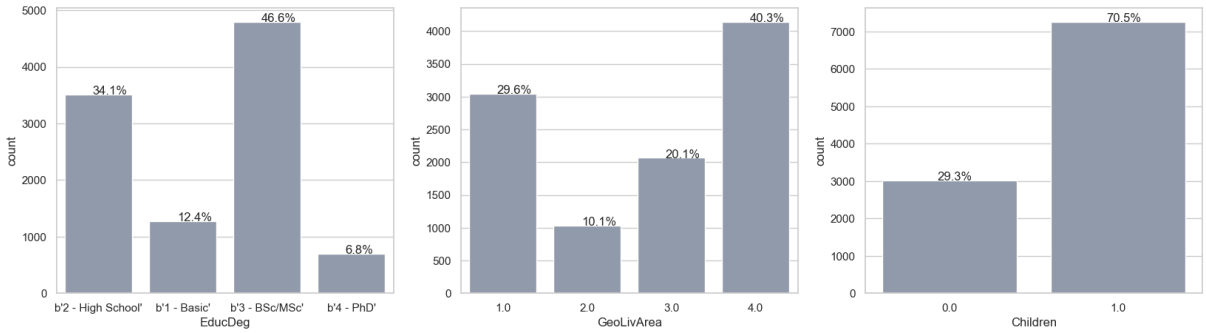

Figure 1 - Bar plots for non-metric variables

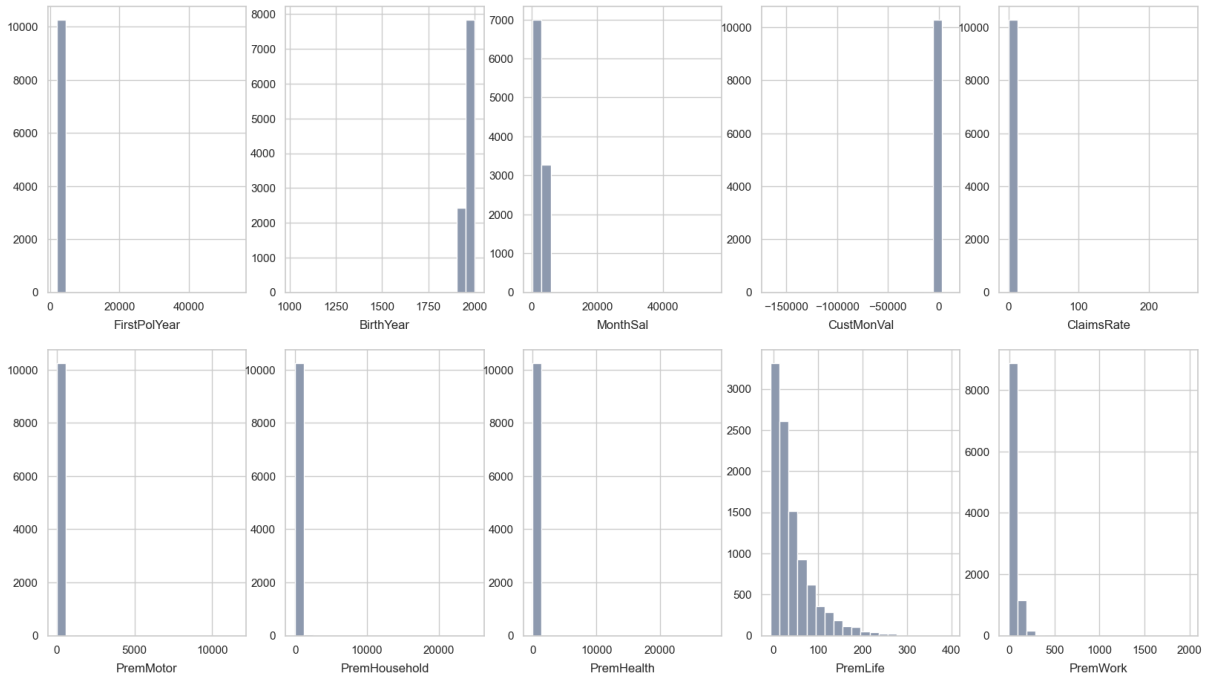

Figure 2 - Histograms for metric variables

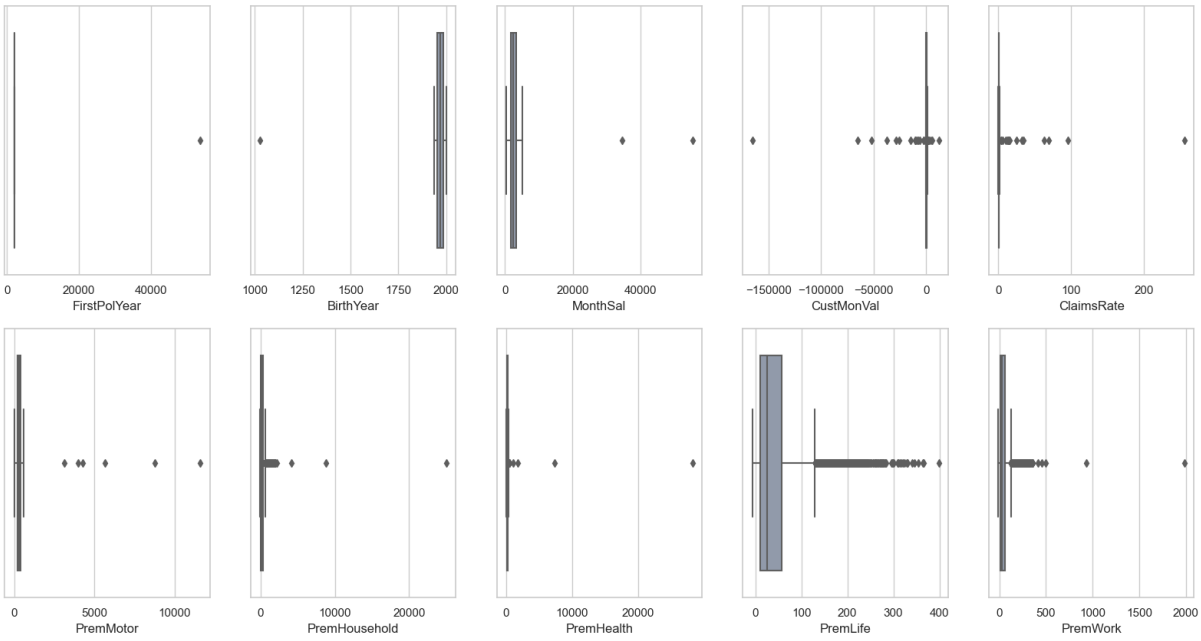

Figure 3 - Boxplots for metric variables

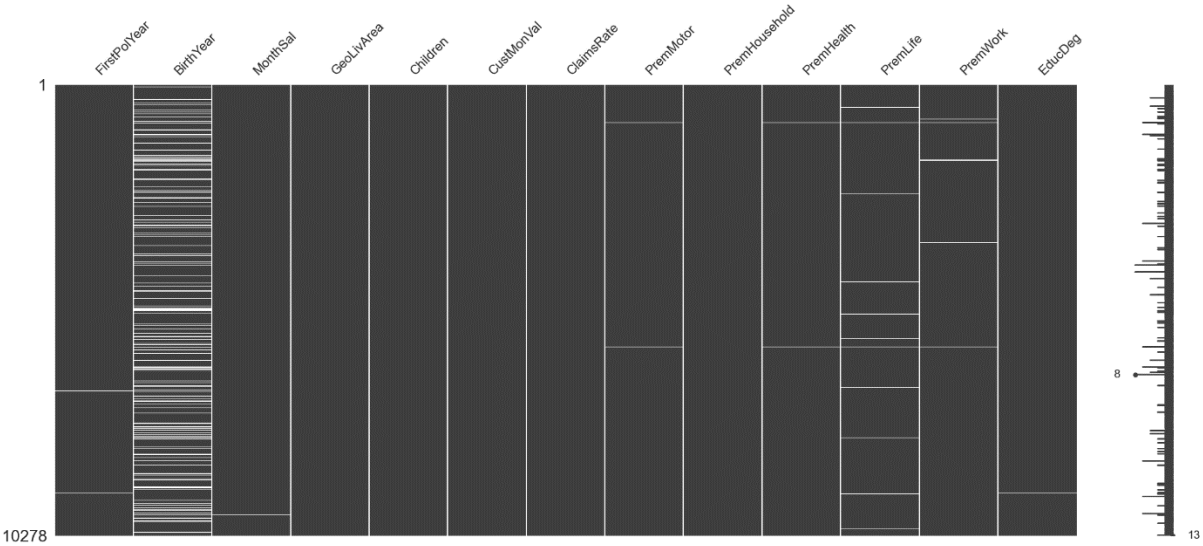

Figure 4 - Missing values

Table 4 - Rules for manual removal of outliers

Variable	Rule
MonthSal	There are two obvious extreme values (34490 and 55215) that are above 30000, so we should remove them. We must keep in mind that these outliers are 'true' outliers since it is possible to customers to have this kind of salary, it is just not usual (like the Bill Gates effect).
CustMonVal	We can remove a small number of outliers until the boxplot does not look so flat. In the negative side of the boxplot (customers that bring loss to the insurance company) we can see a gap between the -14714 and -26130, so we can remove observations with values smaller than -

	25000. On the positive side of the boxplot (good customers that provide profit to the company, as their annual is bigger than the acquisition cost) we notice that the biggest value is 11875, so this observation should also be removed.
ClaimsRate	The only extreme value in is 256 so that observation can be removed.
PremMotor	We can notice 3 major outliers, which were expected because of the huge difference between the percentile 75 and the maximum value (as checked in the exploration phase), therefore, we can remove the observations that have *PremMotor* above 5000.
PremHousehold	As we can see in the flat boxplot, there are 3 major outliers, so they should be removed (observations that have a value greater than 4000).
PremHealth	Regarding the boxplot, there is a gap between the points and the maximum value, so this maximum observation can be considered as an outlier (so, values higher than 2000 are removed).
PremLife	In the boxplot, we did not notice any big gap between observations, even though they are outside the IQR fence, so we decided to not remove any observation.
PremWork	There are 2 super extreme observations, so they were removed (observations that have higher values than 900).

Table 5 - New features

Variable	**Description**	**Formula**
Age	Gives the age of the client, based on the *BirthYear* and considering the current year of the database (2016)	data['Age'] = 2016 - data['BirthYear']
YearsClient	Gives the total years that our client has been our client, based on *FirstPolYear* and considering the current year of the database (2016)	data['YearsClient'] = 2016 - data['FirstPolYear']
TotalPremiums	Gives the total amount of money paid in premiums (annually), based on the sum of all premiums paid (*PremMotor*, *PremHousehold*, *PremHealth*, *PremLife,* and *PremWork*)	data['TotalPremiums'] = data['PremMotor'] + data['PremHousehold'] + data['PremHealth'] + data['PremLife'] + data['PremWork']
AnnualSal	Gives the annual salary of the customer (assuming 14 months)	data['AnnualSal'] = data['MonthSal'] * 14
TotalPremiumsRatio	Gives the part of the annual salary that is used to pay the premiums	data['TotalPremiumsRatio'] = data['TotalPremiums'] / data['AnnualSal']

Log10MonthSal	Gives the logarithm of *MonthSal*, made to minimize the difference among the amounts received	data['Log10MonthSal'] = np.log10(data['MonthSal'])
Log10AnnualSal	Gives the logarithm of AnnualSal, made to minimize the difference among the amounts received	data['Log10AnnualSal'] = np.log10(data['AnnualSal'])
PremMotorRatio	Gives the part of the premiums that is for motor insurance	data['PremMotorRatio'] = data['PremMotor'] / data['TotalPremiums']
PremHouseholdRatio	Gives the part of the premiums that is for household insurance	data['PremHouseholdRatio'] = data['PremHousehold'] / data['TotalPremiums']
PremHealthRatio	Gives the part of the premiums that is for health insurance	data['PremHealthRatio'] = data['PremHealth'] / data['TotalPremiums']
PremLifeRatio	Gives the part of the premiums that is for life insurance	data['PremLifeRatio'] = data['PremLife'] / data['TotalPremiums']
PremWorkRatio	Gives the part of the premiums that is for work insurance	data['PremWorkRatio'] = data['PremWork'] / data['TotalPremiums']
PremMotorSalRatio	Gives the part of the salary that is spent in motor insurance	data['PremMotorSalRatio'] = data['PremMotor'] / data['AnnualSal']
PremHouseholdSalRatio	Gives the part of the salary that is spent in household insurance	data['PremHouseholdSalRatio'] = data['PremHousehold'] / data['AnnualSal']
PremHealthSalRatio	Gives the part of the salary that is spent in health insurance	data['PremHealthSalRatio'] = data['PremHealth'] / data['AnnualSal']
PremLifeSalRatio	Gives the part of the salary that is spent in life insurance	data['PremLifeSalRatio'] = data['PremLife'] / data['AnnualSal']
PremWorkSalRatio	Gives the part of the salary that is spent in work insurance	data['PremWorkSalRatio'] = data['PremWork'] / data['AnnualSal']

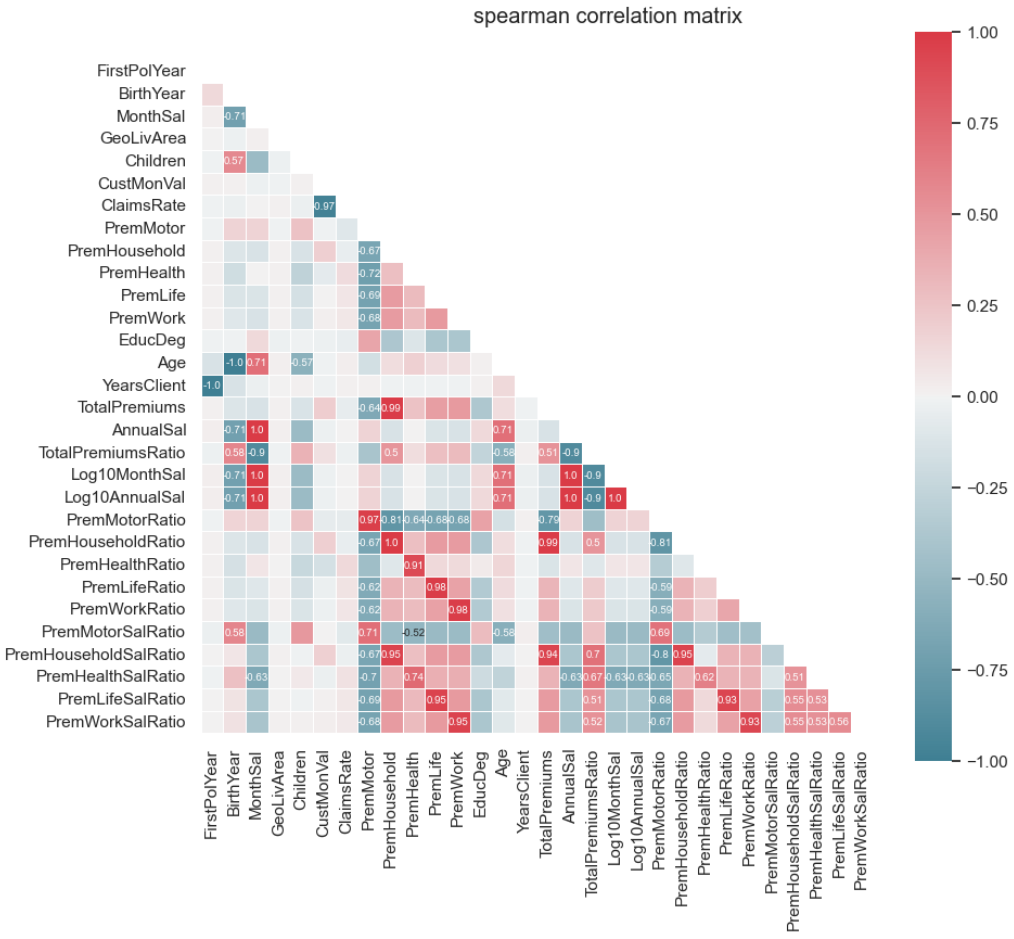

Figure 5 - Spearman's correlation matrix for metric variables

Table 6 - Results for PCA

	Eigenvalue	Difference	Proportion	Cumulative
1	2.992202	0.000000	0.332433	0.332433
2	1.741686	-1.250516	0.193501	0.525934
3	1.170524	-0.571162	0.130045	0.655979
4	1.014436	-0.156087	0.112704	0.768683
5	0.850315	-0.164121	0.094470	0.863153
6	0.544857	-0.305457	0.060534	0.923686
7	0.461375	-0.083482	0.051259	0.974945
8	0.215097	-0.246278	0.023897	0.998842
9	0.010419	-0.204678	0.001158	1.000000

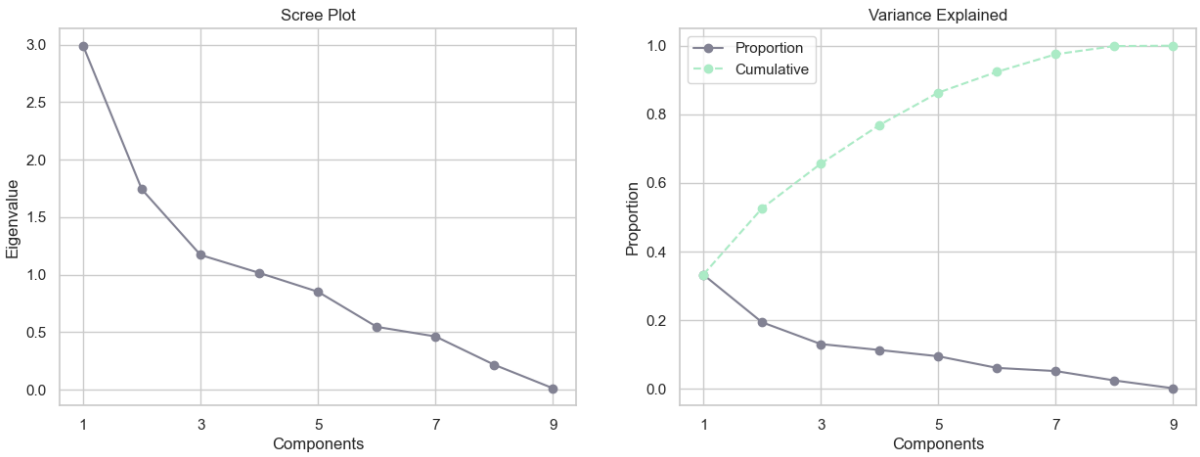

Figure 6 - Variance explained against number of Principal Components

Table 7 - Correlations between Principal Components and metric variables

	PC0	**PC1**	**PC2**
PremMotor	-0.960052	-0.177515	0.153764
PremHousehold	0.753163	-0.027184	0.374780
YearsClient	-0.012732	0.171638	0.161727
PremLife	0.739280	-0.002832	0.025073
Age	0.004335	0.917576	0.200648
PremWork	0.750202	-0.023347	0.055228
PremHealth	0.505049	0.346421	-0.513500
AnnualSal	-0.347313	0.838499	0.121116
CustMonVal	0.133131	-0.119215	0.811083

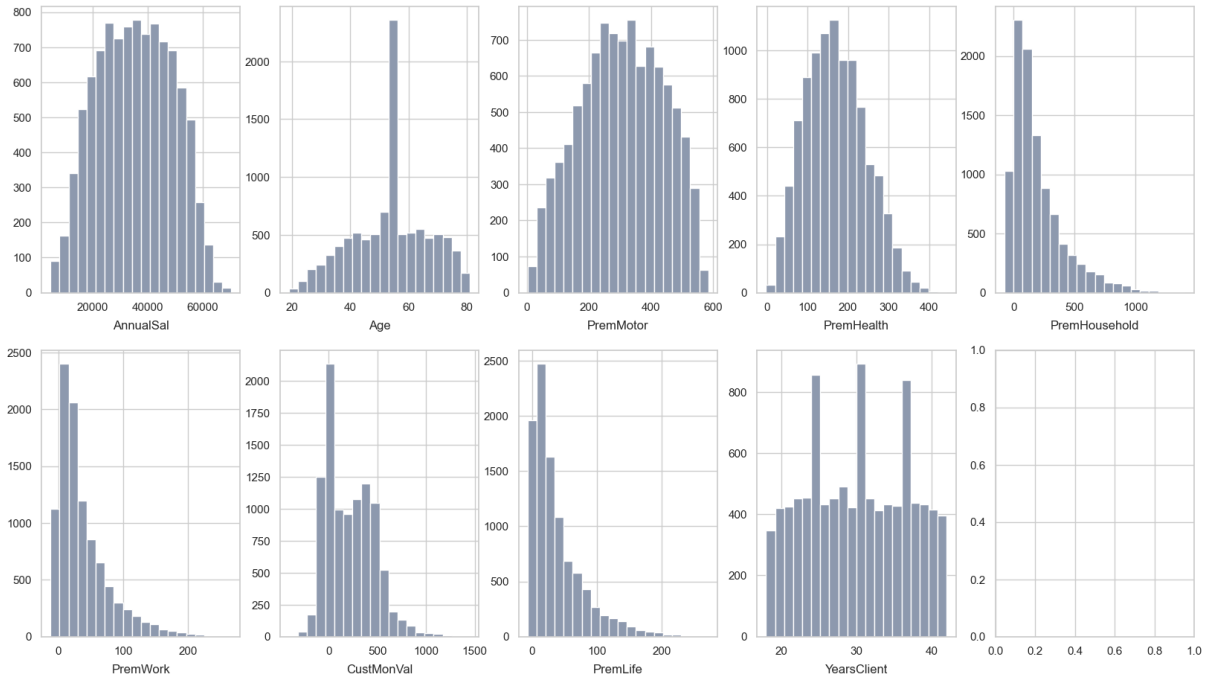

Figure 7 - Histograms for metric variables

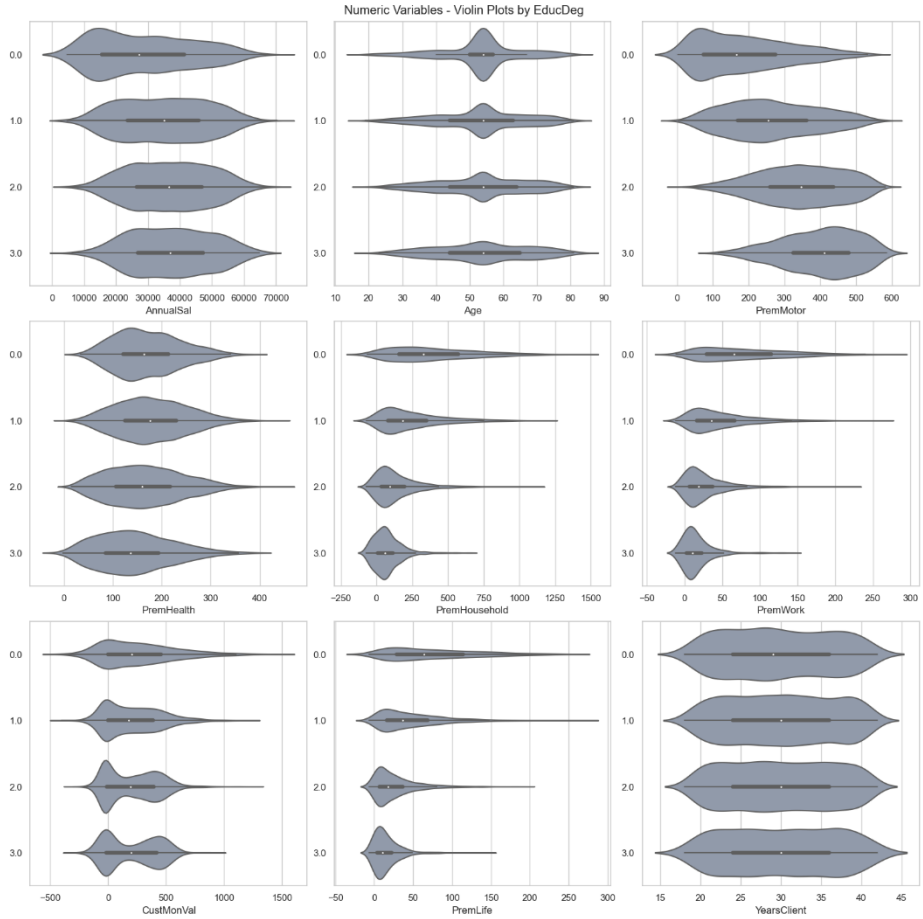

Figure 8 - Violin plots for numeric variables by Education

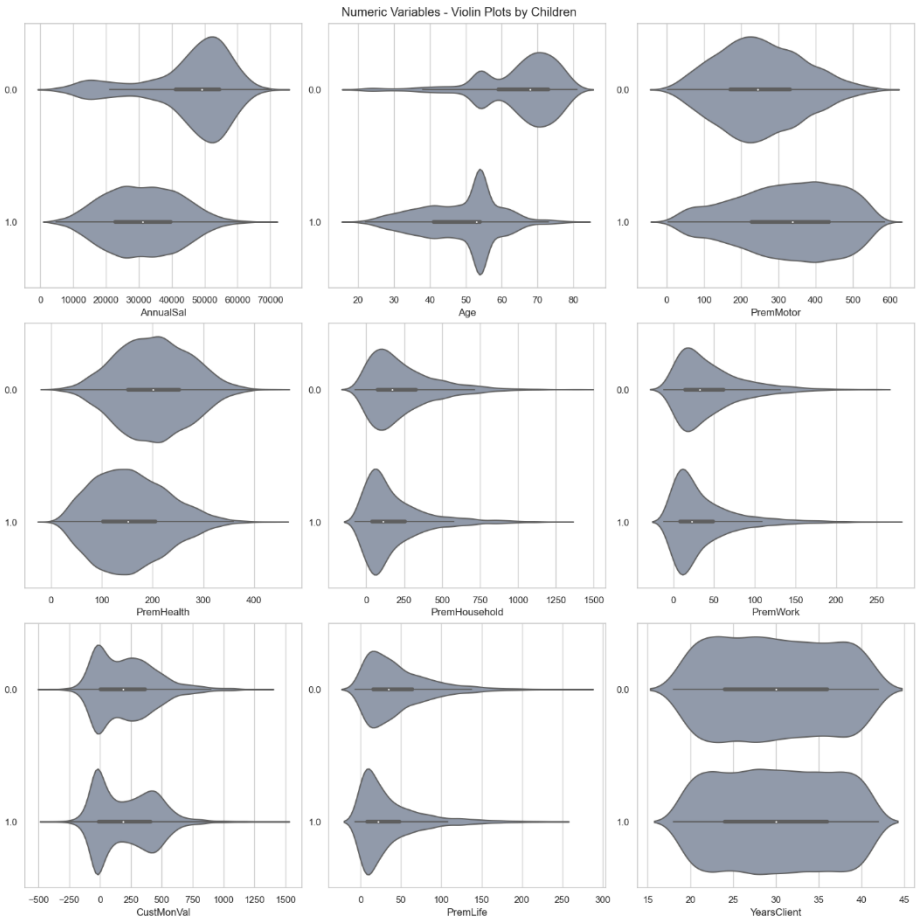

Figure 9 - Violin plots for numeric variables by Children

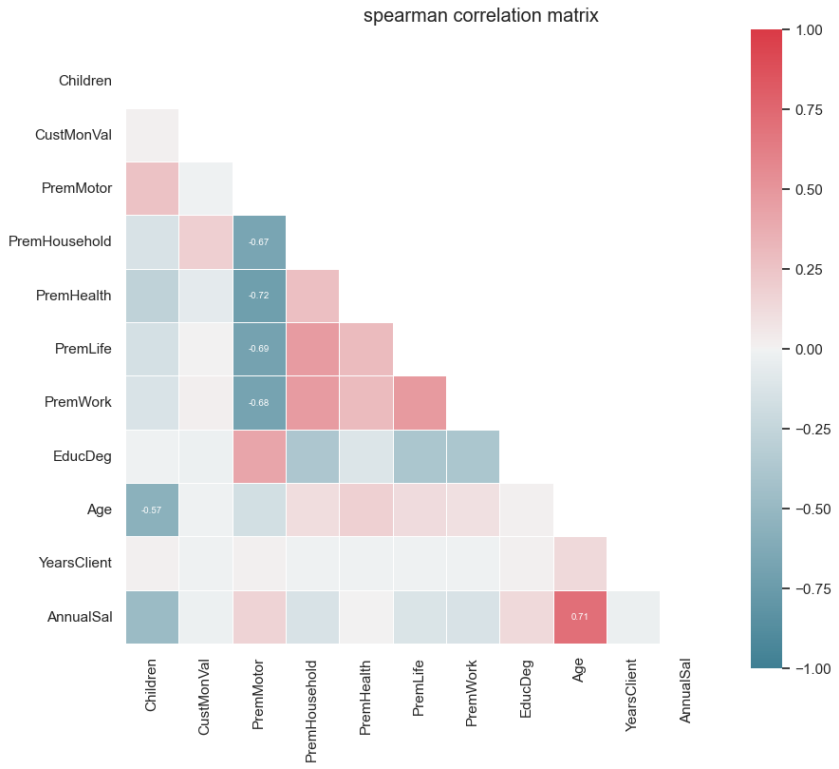

Figure 10 - Spearman's correlation matrix for numerical variables

Table 8 - Variables by perspective

Demographic features	Value features
ohc_Children_1.0	*CustMonVal*
ohc_EducDeg_1.0	*PremMotor*
ohc_EducDeg_2.0	*PremHousehold*
ohc_EducDeg_3.0	*PremHealth*
Age	*PremLife*
YearsClient	*PremWork*
AnnualSal	

Table 9 - R-squared and silhouette score for each clustering method: demographic view

Method	Clusters	R-squared	Silhouette
K-means	4	0.7865446508312557	0.3498298698690985
Hierarchical	4	0.6212821095933949	0.12270995563494541
K-means + hierarchical	3	0.5288909173136533	0.19981715373142317
Self-organizing map	3	0.6283814710017077	0.30942326966145933
Mean-shift	5	0.829392391870688	0.29360730177677896

Table 10 - R-squared and silhouette score for each clustering method: value view

Method	Clusters	R-squared	Silhouette
K-means	4	0.5957485389297719	0.2525488705627774
Hierarchical	5	0.6240056393658103	0.2008261121905221
K-means + hierarchical	3	0.35115600171481576	0.14320787851287442
Self-organizing map	3	0.5957485389297719	0.2525488705627774
Mean-shift	4	0.30961229307866067	0.38304544185720957

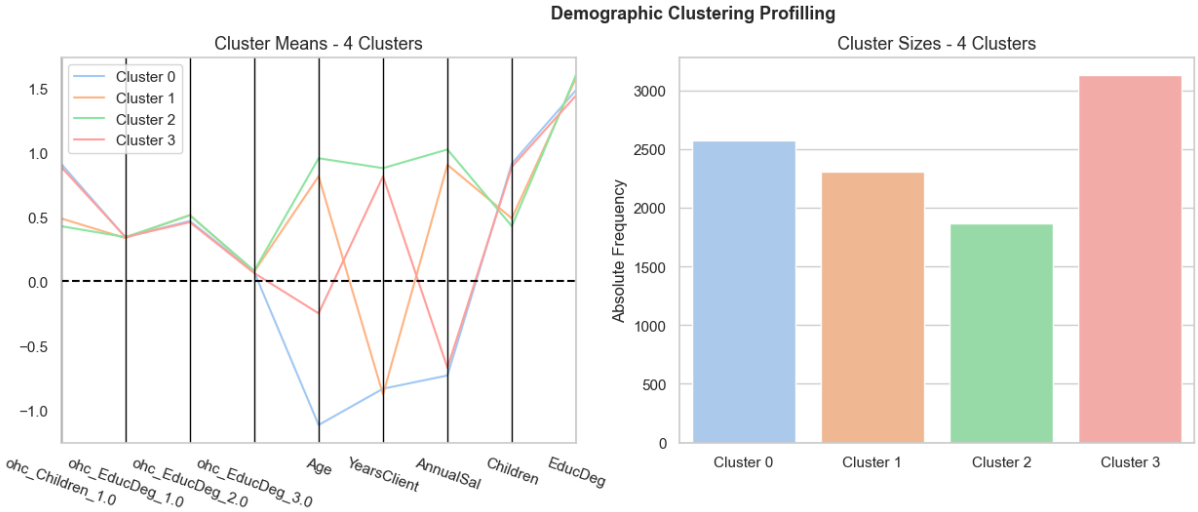

Figure 11 - Demographic clustering profiling

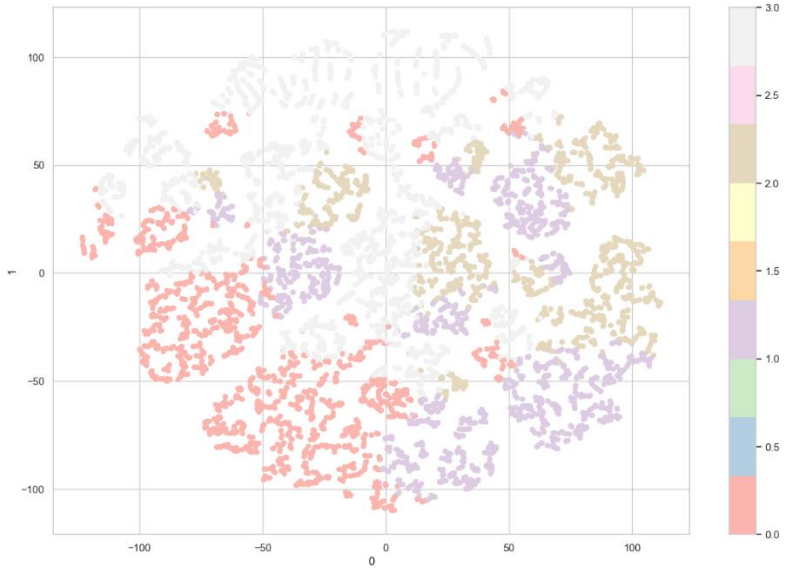

Figure 12 - TSNE demographic clusters

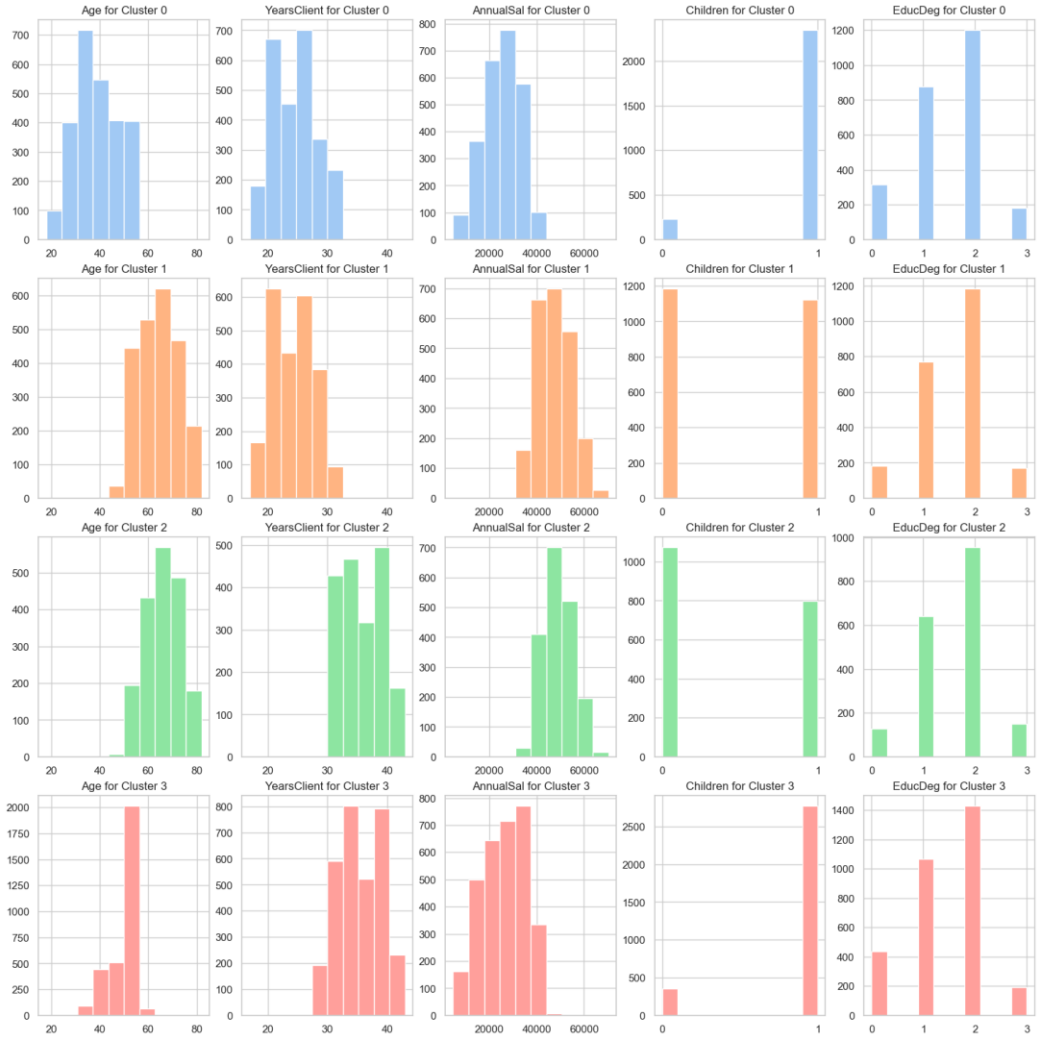

Figure 13 - Variables distributions by demographic clusters

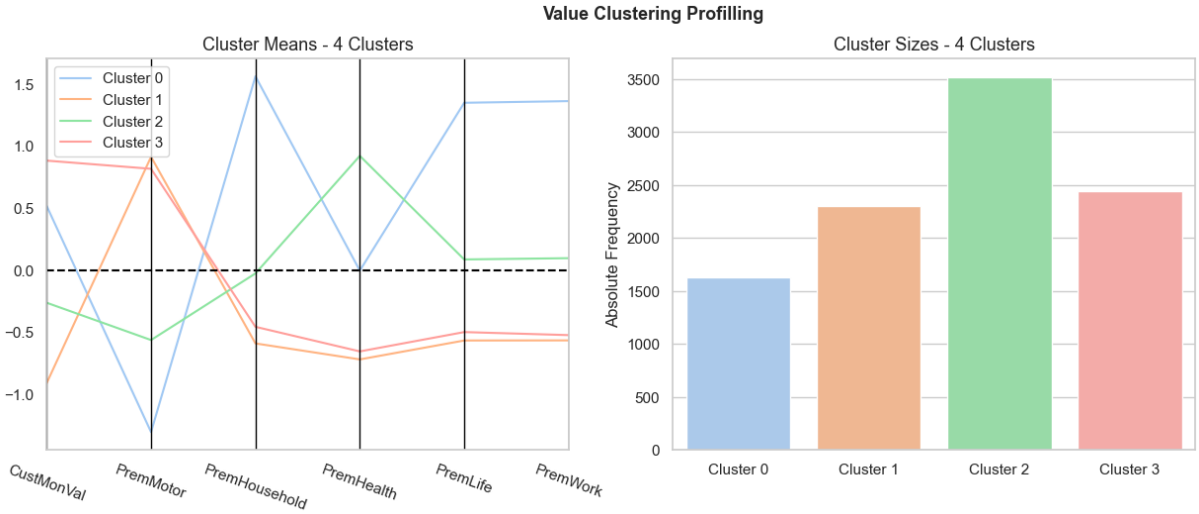

Figure 14 - Value clustering profiling

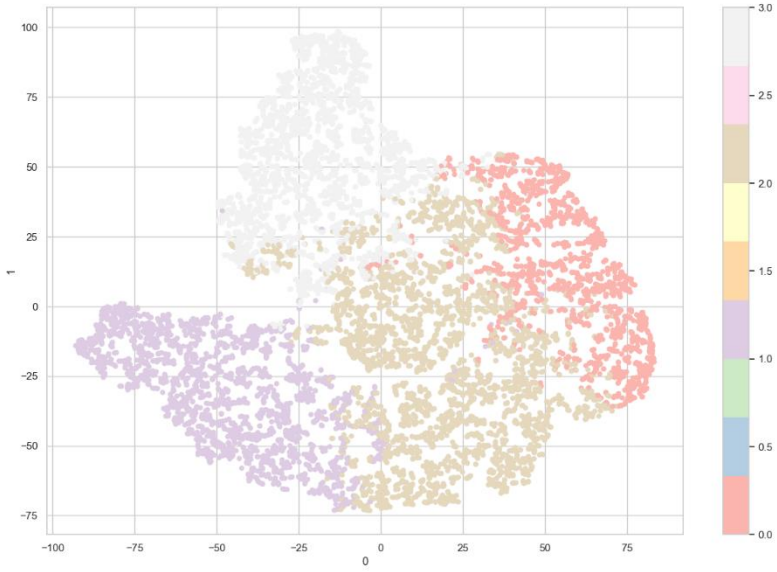

Figure 15 - TSNE value clusters

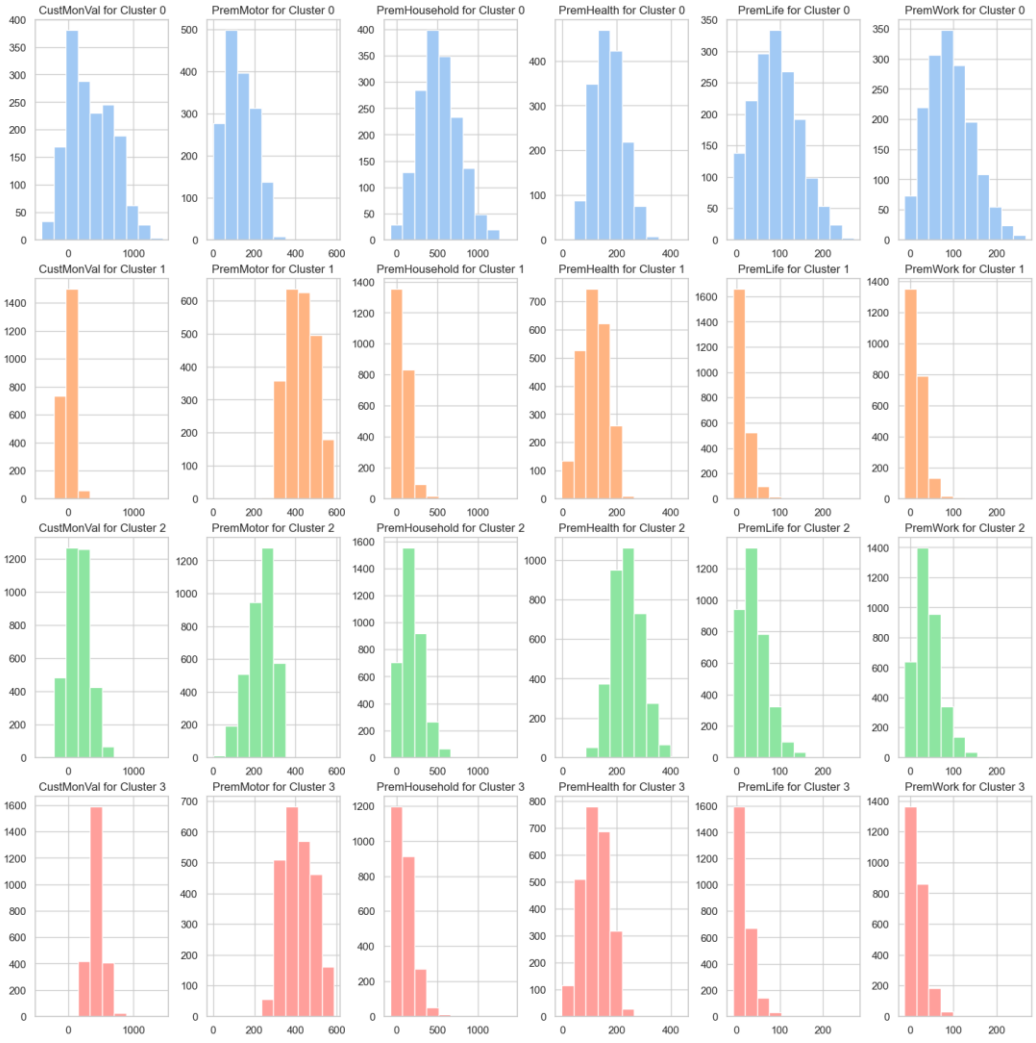

Figure 16 - Variables distributions by value clusters

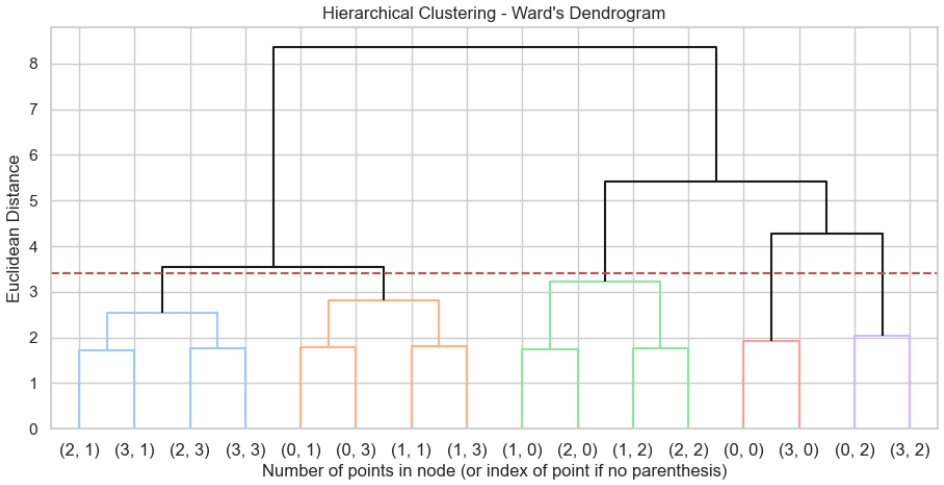

Figure 17 - Ward's dendrogram for merging the final clusters

Table 11 - Results of the clustering merging (for 4, 5, and 6 clusters)

Clusters	R-squared	Silhouette
4	0.467540	0.185527
5	0.536259	0.142250
6	0.573465	0.135531

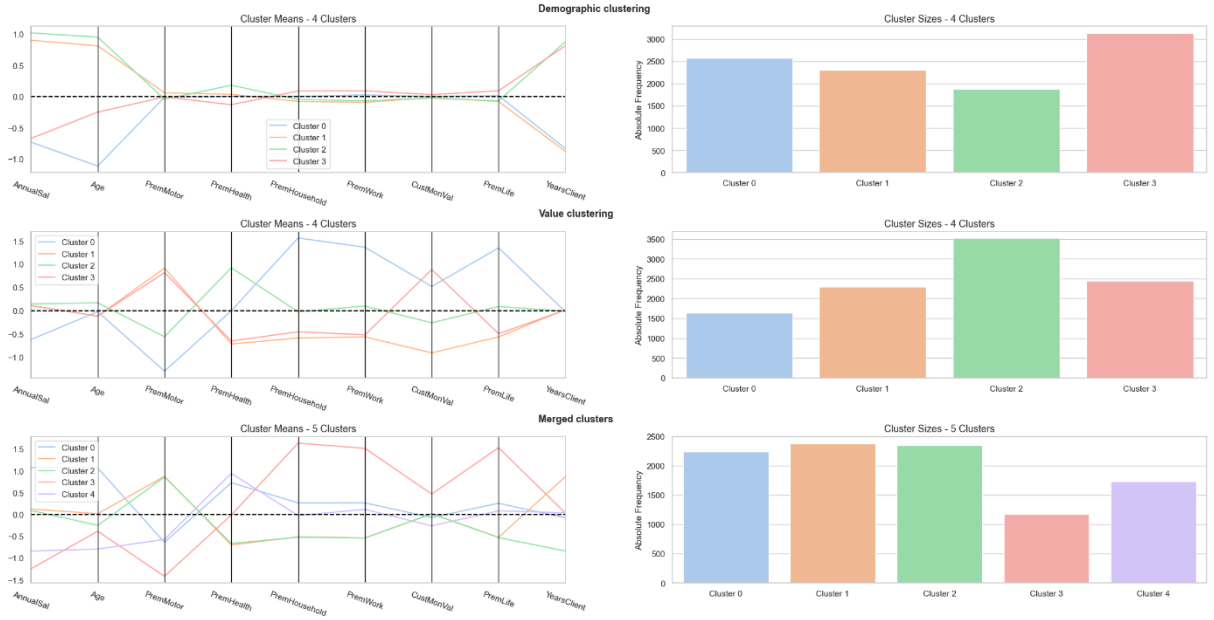

Figure 18 - Final clusters profiling

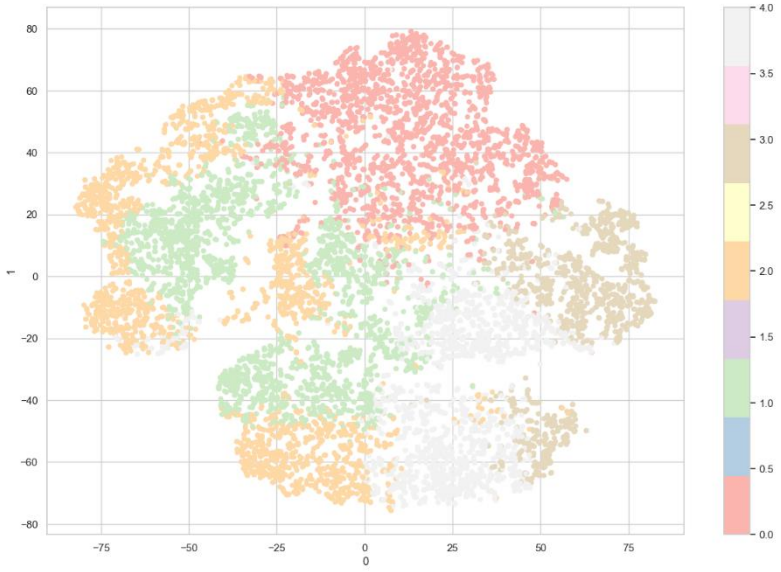

Figure 19 - TSNE final clusters

Figure 20 - Demographic variables distributions by final clusters

Figure 21 - Value variables distributions by final clusters

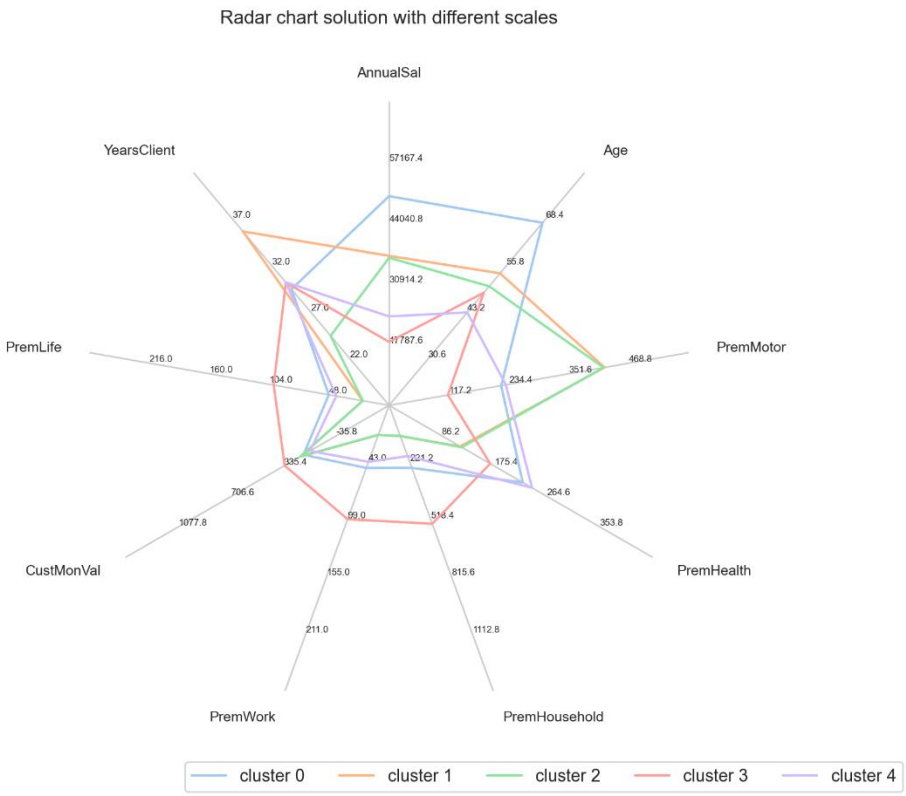

Figure 22 - Radar plot by final clusters

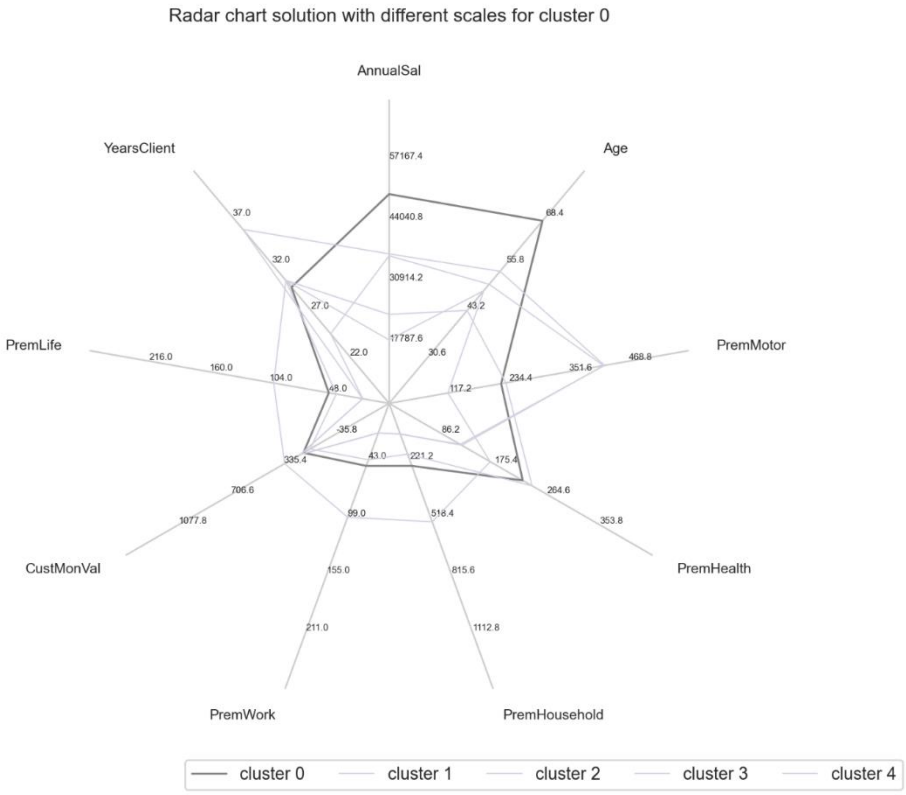

Figure 23 - Radar plot by final clusters (highlighting cluster 0)

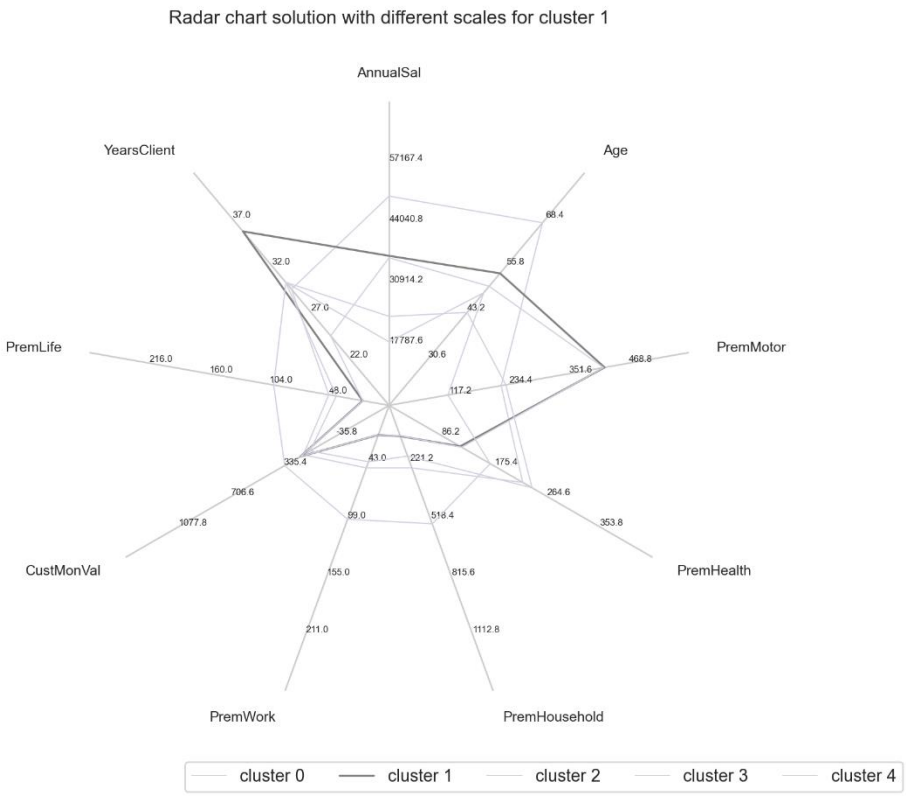

Figure 24 - Radar plot by final clusters (highlighting cluster 1)

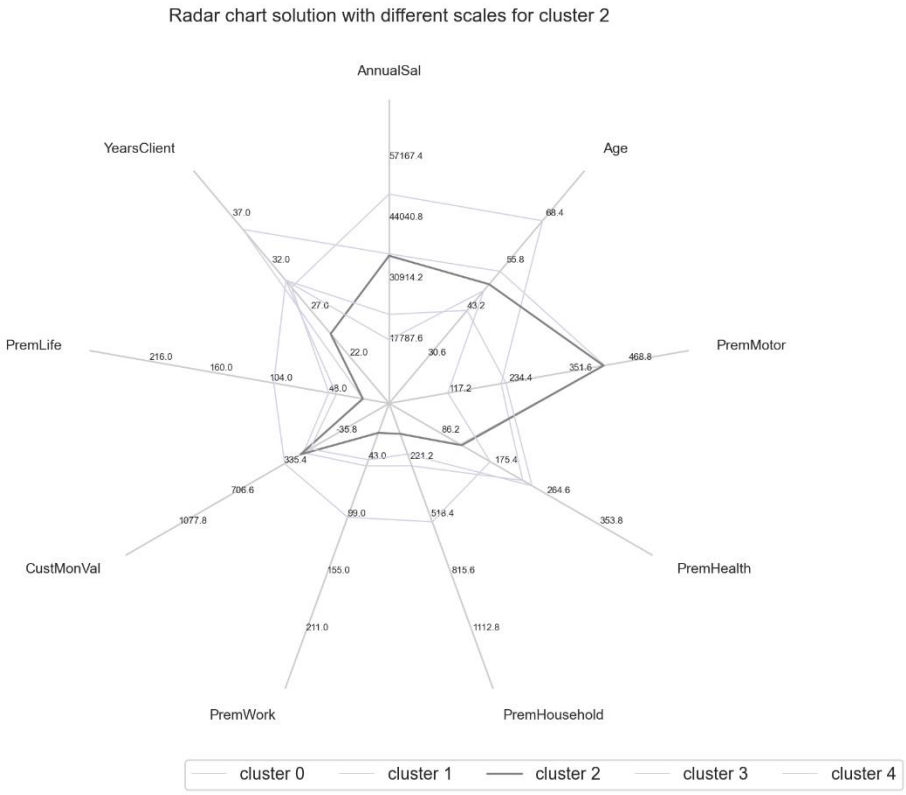

Figure 25 - Radar plot by final clusters (highlighting cluster 2)

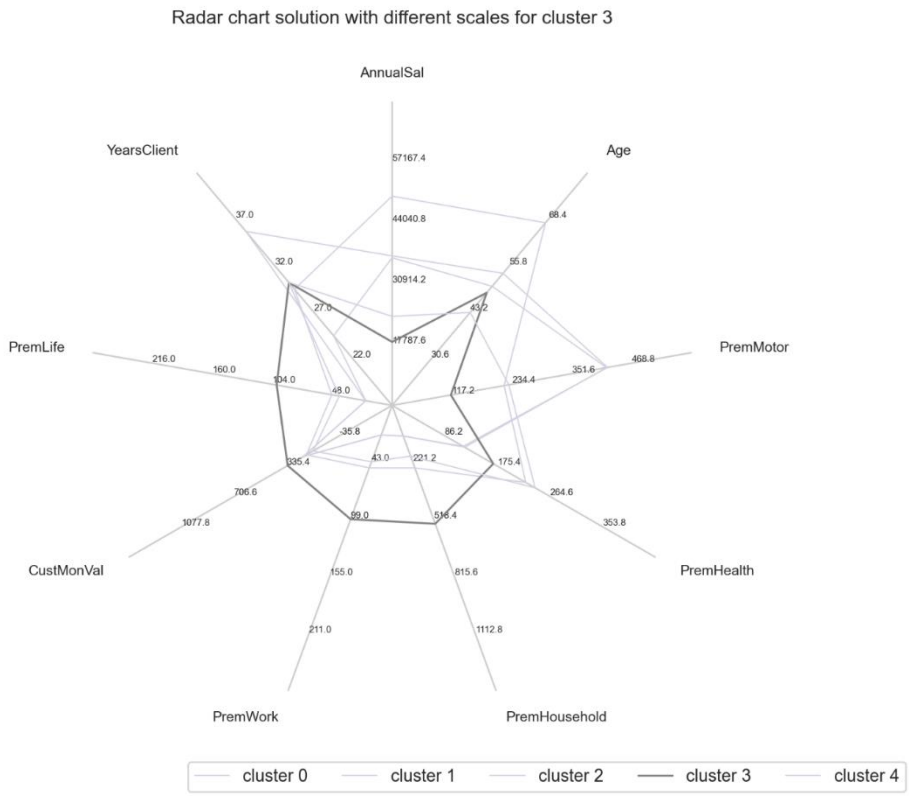

Figure 26 - Radar plot by final clusters (highlighting cluster 3)

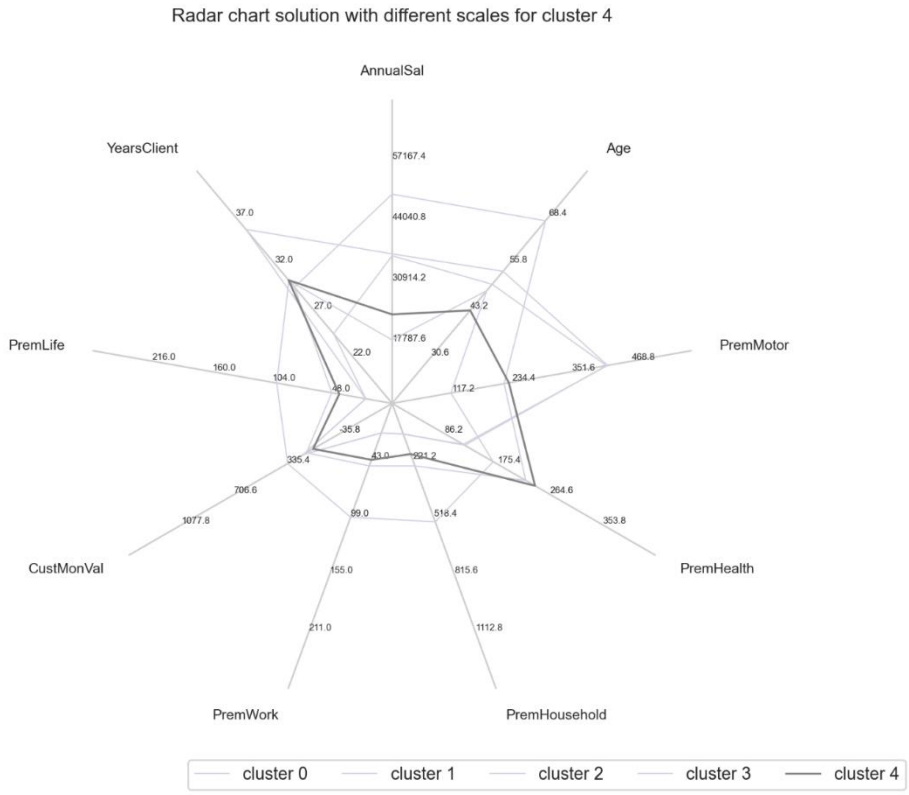

Figure 27 - Radar plot by final clusters (highlighting cluster 4)