Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Group FB

Afonso Reyna, number: <20181197>

Cláudia Rocha, number: r20191249

Felix Gaber, number: <20221385>

<January>, <2023>

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

INDEX

1. Introduction

The aim of this project is to develop a customer segmentation for the A2Z insurance company using an Analytic Based Table (ABT) provided, that will allow us to segment customers of the company based on their characteristics and patterns through the process of Data Mining

By dividing their customer base into specific groups based on characteristics such as insurance coverage, customer tenure, age, and income, the company can better understand the different needs and preferences of their customers and develop more targeted and relevant offers. This will not only help to increase the effectiveness of marketing campaigns, but also lead to cost savings and greater customer satisfaction as customers receive fewer, more personalized offers that align with their interests and behaviors.

2. Data Exploration

2.1. General analysis

The team started the data exploration in order to gain a better understanding of the structure, content and characteristics of the raw data. First, we verified the shape of the provided dataset, which has 10296 rows and 13 columns, already defining the variable “CustID” as the index. In addition, we also checked the data types of each variable, proceeding to the descriptive statistics of numeric and categorical variables (Table 1 Descriptive statistics of numerical data using the ‘describe()’ method. With this initial analysis it was possible to gain a first insight on which variables had missing values, contained abnormal values, its standard deviation, discrepancies between the quartiles, among others to later proceed with the treatment.

Following, we also decided to replace to NaN possible missing or unavailable values that could not be identified as the NaN type, such as ["?","-"," ","null", "NK","--", "nan"]. When calculating the sum of missing observations per variable using ‘isna().sum()’ it was possible to conclude that the only variables that contained all observations were “CustMonVal”, “ClaimsRate” and “PremHousehold”.

The last part of this initial step was to perform a pandas profiling on the initial variables, to be able to visualize distributions and correlations, identifying patterns in the data, as well as any potential issues or problems that will later need to be addressed.

2.2. Coherence Checking

Additionally, we performed a data quality and integrity checking in order to verify the accuracy and consistency of the data. This step is crucial to ensure that the results of this cluster analysis, the aim of this project, are based on valid and reliable data.

First, the presence of 3 duplicates was found. Looking into specific variables, we checked that “FirstPolYear” and “BirthYear” contained only integer values as the presence of a decimal numbers in these variables could not be possible given that both variables represent years. Also, an inconsistency found in these two variables was that 1997 individuals had a value of ‘BirthYear’ bigger than

‘FirstPolYear’, meaning that they became customers before they were born and implying an error in processing, collection or entry of the data.

Regarding the “EducDeg” variable, we detected that 12 individuals born after 2000 had a high level of education or a value of “MonthSal” bigger than zero, which is not possible since customers that are sixteen years old do not have a bachelor's or master's degree neither are allowed to have a job in Portugal.

From this analysis, we concluded that the variable “BirthYear” leads to many issues which we will address in the further steps.

3. Data Cleaning

3.1. Fixing structural errors

We now proceed to fix the errors found in the previous steps, addressing inaccurate data, increasing its consistency and simplifying the interpretation.

The first change made was the removal of the duplicates. Then, we modified the name of the column “ClaimsRate” as it represents the ratio between the amount paid by the insurance company and the premiums, taking into consideration the last two years. We rename it to “ClaismRate_2y” to avoid interpretation problems later. Regarding “EducDeg” we replace the values, i.e., ‘b'1 - Basic'’, with its ordinal scale from 1 to 4 for simplification purposes. Additionally, the values of “GeoLivArea” were converted to string given that they represent different regions. However, these do not represent any particular order.

3.1.1. Variables FirstPolYear and BirthYear

As it was observed in step 2.2, variables “BirthYear” and “FirstPolYear” raise issues in the data regarding its accuracy and reliability. Regularly, a variable with this high proportion of incorrect data would be deleted. Nevertheless, from a business perspective of A2Z Insurance, both are strong predictors of risk, and the removal of these variables would result in a high loss of valuable information. The variable “BirthYear” results in a useful interpretation of customers in their health status, lifestyle and occupation as, for example, on one hand, younger individuals are more likely to engage in risky behaviors and, on the other hand, older customers tend to have more health issues. Regarding “FirstPolYear”, it represents the relationship between the customer and the company since customers that have been clients for more years are less likely to cancel their policies because of the longer history of paying premiums.

Earlier observations showed that some individuals' tenure with the company exceeded their age, suggesting that one of the recorded variables (“BirthYear” or “FirstPolYear”) may be inaccurate. To determine which variable was incorrect, we conducted a linear regression with age as the dependent variable and the other variables as independent variables. The results of the regression indicated that the values for "BirthYear” were more credible than those for "FirstPolYear”, as they were consistent with the estimates of the regression. This led us to conclude that the error in "FirstPolYear” may have been caused by the individual being included on a parent or guardian's insurance policy from birth. As such, the values recorded for "FirstPolYear” may correspond to the policy holder who added the individual to their contract.

To address this issue, we transformed the values that exhibited this discrepancy into NaN values and used the K-Nearest Neighbors (KNN) method for imputation in the next step. For observations that still exhibited the same issue after the imputation, we replaced the first policy year with the birth year.

3.2. Managing missing values

In this step of our project, we could notice that only five of the fourteen variables in the dataset do not have missing values, as visible in Figure 1. To address this issue, we used the K-Nearest Neighbors Imputer, that considers a number of neighbors, k, and uses those values to impute the missing value.

We considered this was the right approach because the percentage of missing data was low and did not have a significant impact on the data. Therefore, we defined k=6 for all the variables, categorical and continuous, and proceeded with the data treatment.

3.3. Outlier treatment

Regarding outlier management, we proceeded by plotting the data in boxplots and histograms to get a perspective on the distribution of the different variables and to visualize which records were unreasonably outside the interquartile range. Many records came to our attention, as most of the variables had values that fell far outside a reasonable range and would severely bias parameter estimations and visualizations. As we're dealing with clustering algorithms, it is very important to carefully handle outliers due to the sensibility that some algorithms have to these data points (such as kmeans for instance). The changes applied to variables can be seen in the annex with the boxplots before (Figure 2) and after the outlier removal.

3.3.1. Manual filtering

Manual filtering includes methods that handle outliers in a non-algorithmic manner, thoroughly examining each variable's situations and removing outliers by setting boundaries manually, variable by variable. This method for outlier handling, gives us the most elasticity and ability to tailor a specific approach to each variable. Although some values may fall outside the interquartile range (x1.5), they corresponded to real observation values that can be relevant to perform clustering (Figure 3).

3.3.2. Automatic filtering

Automatic filtering is the method that uses a specified interquartile range multiplier to delete abnormal observations that fall outside of this range. This approach is efficient and clean, but since it deleted almost 10% of all rows, we decided not to use it, due to the big loss of information that eventually could be important to our analysis(Figure 4 + Figure 5) .

3.3.3. DBSCAN

The last approach for outliers was Density-Based Algorithm(DBSCAN), that identifies groups of points through density and has the positive outcome of identifying outliers in more than one dimension. After several attempts, as DBSCAN is sensitive to the initial parameters, the best value choice for Eps was 200, resulting in identifying approximately 5% of the data as outliers.

Although we performed the cluster analysis with DBSCAN as an outlier treatment procedure, it ended up lowering the scores for most algorithms in the further steps. Therefore, our final decision was to

maintain the manual filtering that also had the benefit of allowing us to have a very customized procedure on each variable distribution.

4. Feature Engineering

4.1. Feature Creation

In order to improve the interpretability of the data and reduce its complexity, we proceeded to the creation of new variables. The first change was to subtract 2016, the current year, to “FirstPolYear” and “BirthYear” creating, respectively, the features “years_as_cust” and “Age” as later it can simplify our interpretation of the results.

Additionally, we created “annual_salary” multiplying “MonthSal” by the 12 months of the year. The team also created “prem_total_2016” by adding all the value of premiums together off the current year of the database. The new variable “premium_salary_ratio” was the ratio between “prem_total_2016” and “annual_salary”. Lastly, “estimate_paid_to_cust_15_16” represented an estimate of the amount paid by the ratio between “prem_total_2016” and “ClaimsRate_2y”.

Although some of the created variables are highly correlated with each other, for this step we decide to keep all of them and apply the changes later, when taking a further look into the dimensionality reduction.

4.2. Feature transformation

4.2.1. Log Transformation

When analyzing the distributions of all the variables (Figure 6), original and recently created, we noticed the presence of data that was not evenly distributed across the range of values. In the context of unsupervised learning, skewed data can lead to inaccurate or biased results of the analysis. It is pssible to notice that the variables 'PremHousehold', 'PremHealth', 'PremLife', 'PremWork', 'years_as_cust', 'prem_total_2016' and 'estimate_paid_to_cust_15_16' are skewed to the right, with a long tail of values on the right side of the distribution.

For this reason, the group decided to apply a log-transformation with ‘np.log()’. It is important to highlight that some of the variables had negative values, with the minimum being the value -75, and this transformation would create NaN in the data. Therefore, to avoid this issue, when applying the log transformation, we added 76.

Lastly, taking a look into the histograms (Figure 7) with the transformed variables, we can conclude that the distributions are significantly closer to the normal distribution, resulting in more linear relationships and making the cluster analysis more efficient and simpler.

4.2.2. MinMaxScaler

After performing the previous transformations, we conducted MinMax Scaling to normalize the numeric variables (values between 0 and 1). This is necessary to put all variables in the same level scale, in order to not give more importance to the ones with wider ranges of values.

4.2.3. Dummy encoding

The ordinal variables were transformed into dummy variables utilizing the pd.to_dummies() method, deleting the redundant variable.

4.3. Dimensionality Reduction

4.3.1. Correlation analysis

Before using clustering algorithms (Figure 8) on a dataset, we checked for the presence of strong correlations between variables in the dataset, as it can distort the clusters that are created, resulting in groups that do not accurately reflect the patterns present in the data. It can also be difficult to interpret the clusters that are formed, as it is not clear which variables are driving the patterns in the data. Therefore, this process can help prevent any potential issues that might arise and give insights into the relationships between the variables.

To start the process of reducing the dimensionality of the dataset, we proceeded to map the correlations between each variable in a heatmap to understand which variables had linear association and therefore would add redundant information to the models.

It was noticed that variables that we had created in the Feature Creation step, such as ratios regarding the premiums, monthly salary, sums of different variables had high correlations with each other, so it was important to highlight which of them we favored over others.

In the end, we decided to drop "BirthYear" and "FirstPolYear" due to their perfect correlation, naturally, with “Age” and “years_as_cust”, as these variables are derived from the subtraction of the current year with the year of the observation. We were more comfortable working with “Age” and “years_as_cust” because they are easier to interpret.

We also dropped “CustMonVal” and "estimate_paid_to_cust_15_16", because of their high correlation (-0.9 and 0.9 respectively) with “ClaimsRate_2y”. Of the three variables, “ClaimsRate_2y” proved to be the one with the least correlation to the other variables, therefore we opted to keep this one. “annual_salary” was preferred over “MonthSal” for interpretation purposes. Lastly The variable “prem_total_2016” which is the sum of all the premiums of the customer was eventually dropped as well due to its high correlations with all the premiums. The same case applied to “premium_salary_ratio" (Figure 9).

4.3.2. Visual analysis of non-metric variables

In the visual analysis of non-metric variables, we analyzed the variables “EducDeg”, “GeoLivArea”, and the binary variable “Children”. We applied violin plots to plot each categorical variable and compared the different distributions within each variable across the different characteristics. For EducDeg (Figure 10) and Children, the distributions were slightly similar, but there were also occasional clear distinctions. In contrast, the variable “GeoLivArea” has very similar distributions for all its values. Since it does not do very much and we predict it will only add noise to our analysis, we will exclude it.

4.3.3. Principal Components analysis

Another alternative of dimensionality reduction technique is the Principal Components Analysis, that identifies patterns in the dataset and creates new variables, the principal components (PC), that

capture the most information of the variables. After analyzing the Figure 11, that shows the eigenvalues and the variance of the principal components to choose the best number of PC to retain, we decided to proceed with 5 according to the Elbow method.

The interpretation of Principal Components (Table 2) leads us to the drawback of this approach. As we can see, the PC0 is a linear combination where the variables with most importance are 'MonthSal', 'PremMotor', 'years_as_cust', 'premium_salary_ratio' and 'Age'. This combination does not result into a clear or intuitive interpretation in terms of the original variables, as, from our business point of view, variables related to customer behavior and characteristics should not be analyzed together. Therefore, we decided not to proceed with this approach.

5. Clustering analysis

After performing the dimensionality reduction, the final variables to continue our research are present in the following table. Additionally, for business purposes, we believed there should be two areas under analysis – Demographic and Value, that contain, respectively, variables regarding the client's demographic characteristics and value of the individual for the company, containing variables regarding its behavior.

Variable	**Perspective**
ClaimsRate_2y	Value
PremMotor	Value
PremHousehold	Value
PremHealth	Value
PremLife	Value
PremWork	Value
Age	Demographic
years_as_cust	Demographic
annual_salary	Demographic
Children	
EducDeg	

After this division, it is important to enhance that we will perform the clustering analysis based on these perspectives and only later merge the best outcomes to reach a final clustering solution. However, we did not include the metric variables "Children" and "EducDeg" in the perspectives because most of the clustering algorithms are designed to work only with numeric data and will not perform efficiently with them.

It is also important to highlight that the outcomes that we are presenting in the further steps, are based on the dataset with the best attempt for data treatment from all the possible combinations from

above, that include the best option for outlier handling, missing values treatment, scaler and dimensionality reduction.

5.1. K-Means and Hierarchical clustering

Our first approach was to apply K-Means and later perform hierarchical clustering (Figure 12). As K-Means is an algorithm for partitioning the dataset and assigns each observation to its nearest centroids, it can be very sensitive to the initial placements of the cluster. For that reason, we chose k=20, creating 20 groups, and later applying Hierarchical clustering to build more robust final groups.

For the demographic perspective, first the dataset was partitioned into 20 groups, on which we performed the dendogram with ward linkage and euclidean distance, resulting into 3 final clusters divided into approximately 45%, 30% and 25% of all the individuals. This clustering solution had a Silhouette score of 0.728 and a Calinski and Harabasz score of 129.976.

In Table 3 that includes the average of all the data points in the cluster for each variable, it is noticeable that these clusters solutions differ from each other. However, when analysing its distributions (Figure 13), we noticed that, in general, they present different characteristics, but their distributions are similarly shaped.

Regarding the value perspective, the same process was applied, with the outcome of four clusters (Figure 14) that grouped approximately 24%, 32%, 25% and 19% of the customers. Although this clustering solution presented lower Silhouette and Calinski and Harabasz scores, of 0.673 and 112,850, respectively, the analysis of the the Table 4 and histograms (Figure 15) lead us to the conclusion that this method resulted in clusters with a bigger visual difference between them. It is important to highlight the dissimilarity of the variables PremMotor, PremWork and ClaimsRate_2y.

5.2. K-Prototypes

Given that our clustering methods in this project only take into consideration metric variables, the application of K-Prototypes was beneficial to create a solution that included the non-metric variables “Children” and “EducDeg”. Subsequently, we added these variables, without encoding, to our demographic perspective.

The first step was to perform the elbow method, to decide that we would choose three clusters for this approach that resulted in a Silhouette and Calinski and Harabasz score of 0.492 and 6710.18, respectively, on which this last one presents a significantly higher value compared to K-Means.

As we applied the algorithm, from Table 5 and Figure 16, it is possible to notice the fact that most of the individuals from the first and third clusters have the level ‘3’ of education. From this interpretation, we can also notice perceive that the third cluster, on which individuals mainly have no kids is also the one with a higher range of ages.

5.3. SOM

In the self-organizing map, there are three general outcomes of clustering solutions that we applied in our notebook. The first is to run the SOM alone over the data and determine the clusters directly.

Alternatively, with a larger map size, the SOM captures the topological structure of the data and then applies either K-means or hierarchical clustering to the SOM nodes.

We started by running the SOM alone over the data with a size of [5,5]. The histogram (Figure 21), as well as the U-matrix (Figure 19) and hit-map (Figure 20) for demographic features, are attached. For value features, we did not obtain notable results in this way. Therefore, we tested the combinations of SOM-K-means and SOM-hierarchical clustering. For this, we used a SOM size of [20,20], random initialization, gaussian neighborhood, and batch training parameter. However, before applying the algorithms on top of the SOM nodes, we analyzed the topological structure of the SOM using component planes (Figure 22 and Figure 23), U-matrix, and hit-map (Figure 24 and Figure 25). Based on the SOM visualizations and the dendrogram we used for hierarchical clustering, we decided to aim for four clusters for value features and three for demographic features. The clustering results of the nodes after K-means and hierarchical clustering (Figure 26 and Figure 27) are also attached in the annex.

Since we did not obtain good results subsequently, we decided to build a function that compared all combinations of different parameters and number of clusters for K in a range of one to five and stored the best combinations in terms of silhouette scores in a dataframe. With this, we tried the repeated process again.
We conducted clustering using Self-Organizing Maps (SOM) by running sompy on the data from df_dem and df_val. We intentionally gave the map a size of [2,2] so that the clusters would be output in a corresponding number. When using the value features, the results of the silhouette score were as follows: gaussian_rect score was 0.2018, gaussian_hexa score was 0.197, bubble_rect score was 0.2091, and bubble_hexa score was 0.401. When using the demographic features, the results were even worse: gaussian_rect score was 0.2277, gaussian_hexa score was 0.2398, bubble_rect score was 0.2661, and bubble_hexa score was 0.2512. In addition to the poor silhouette scores mentioned above, the cluster labels were too heavily skewed, so we did not pursue this approach further.

Subsequently, we attempted to combine SOM with both K-means and hierarchical clustering using our previously built function. The outcome for the silhouette score of value features (Table 8) and demographic features (Table 9) are attached.

5.4. Mean-Shift Algorithm

The Mean-Shift algorithm is a clustering algorithm that shifts iteratively the mean of each cluster towards the areas of more density, having the advantage of not needing to specify the number of groups upfront. However, as it is sensitive to the bandwidth of the kernel, we made several attempts to choose the value of this parameter that resulted in better and more meaningful clustering solutions.

For the demographic perspective (Figure 17), the best output of this algorithm was five clusters of approximately 32%, 19%, 19%, 16% and 14%, having a value of 0.777 for Silhouette score and 140170.84 for Calinski and Harabasz score, which we considered a promising result. Taking a deeper look into the characteristics of each cluster, in Table 6, we can notice that the average value of each group varies significantly, and it is possible to find the dissimilarities between them. Lastly, when interpreting the histograms is perceptible to find a difference in the distribution from the different groups.

Contrarily to the great result of the previous perspective, when performing Mean-Shift on the value perspective, the best outcome was five clusters, on which the first one contained 92% of the observations. The characteristics of these clusters can be revised in Figure 18 and Table 7, however, we decided that this option was not going to be used.

5.5. DBSCAN

The Density-based spatial clustering of applications with noise (DBSCAN) is an algorithm that clusters observations based on density and neighborhood of observations. It does not take as input the number of clusters, so it is important to specify reasonable values for the two main parameters of this model in order to get a satisfactory number of clusters: 'eps' and 'min_samples .

We attempted several different combinations of the two parameters but arrived at the conclusion that the best range of values were close to "eps" ~= 0.1 and min_sample ~= 100. We performed a fit with the 2 perspectives of demographic and value variables. The fit of the value perspective yielded a somewhat reasonable cluster solution with 3 clusters, however there was a big main cluster and 2 other clusters with significantly smaller count of observations. The demographic perspective did not yield good results regardless of the specified parameters. The same problem of 1 big cluster and then smaller clusters occurred, but in a more extreme way.

Our experience with this algorithm was that it was very sensible to the parameters specified. By changing even a fraction of a decimal the parameter of neighbourhood, the clusters could change from a total of 6 to 1 and vice-versa – even if the change implied the opposite would occur. The algorithm in the end generated unsatisfactory clustering solutions. We believe this is due to DBSCAN's known shortcomings with high dimensional data. Therefore, our final decision was to disregard this model in favor of other algorithms that we tested.

6. Final cluster solution

The project group carried out several clustering techniques for both perspectives and subsequently decided to combine the best solutions. The Mean Shift solution (silhouette score= 0.7765, calinski habarasz score= 140170) with 5 clusters was chosen for the demographic variables, and the KMeans solution (silhouette: 0.6732, calinski habarasz score: 112.85) with 4 clusters was chosen for the value variables, giving a total of 20 clusters. In order to merge these clusters,

Their centroids were calculated and then both manually (Table 10 and Table 11)and hierachical clustering using to the dendrogram clustered. Through the process of hierarchical clustering, we concluded that it would be sensible to merge the clusters, resulting in 6 clusters remaining, whose characteristics and the number of customers contained are shown in Table 12.

In the analysis that we carried out based on the radar plots (Figure 30), the distributions for the different clusters in the different features (Figure 31 and Figure 32) of the customers' value for the insurance company, there is no cluster that has either higher or lower values in all categories simoutaneaously. The clusters interchange their ranks for the different features, depending on which feature we are considering. The analysis of the preferred products of the customers shows a balanced distribution, as 2 clusters (Cluster 0 and Cluster 1) have a clear preference for the PremMotor category,

4 clusters (Cluster 0, 2, 3 and 5) have a clear preference for the Health category, the same is true for the Life category (with clusters 3 and 5), as well as with the PremWork category, where Cluster 5 stands out.

To clusters 0 and clusters 3, that represent an older generation with high income and low percentage of having children, we devised a strategy of offering discounts on bundled coverage packages that include both health and household insurance.
Cluster 1 represents a generation of mid-range aged / medium class clients that have very high motor premium spending and high percentage of having children. We would recommend devising a plan to get a discount on a family van (2% for instance) when purchasing car insurance from the company.
Cluster 2 represents young, highly educated, family-oriented customers with expenses on Health and Household insurance, as such, we labeled them as the academics. We would recommend creating social media content tailored to the needs and interests of academics. Publish regular posts that deal with health and wellness, such as healthy lifestyle tips, workout routines, and nutrition tips. You could also publish posts that deal with housekeeping, such as shopping tips, household budgeting, or DIY projects.
Cluster 4 represents middle aged, mid-range salary long customers with big expenses on motor insurance and a high percentage of having children. An approach could be E-Mail Marketing campaigns, sending personalized auto insurance quotes based on parents' age, income and lifestyle, send regular newsletters that address topics of interest to parents, such as family planning or financial management.
Cluster 5 is the cluster with the youngest customers, with the shortest time with the company, lowest salary and expenses all over the board when it comes to insurance. We labeled this group as Millennials. The strategy we propose is Online ad campaigns that are specifically tailored to the needs and interests of Millennials. For example, run display ads that focus on health and wellness topics, or social media ads that focus on household and financial topics. Ads that focus on life insurance or employment insurance, which are specifically tailored to Millennials' needs and interests.

7. Conclusion

We followed the usual Data Mining pipeline of understanding the data given, transforming and treating the data, selecting the best features and then fitting the clustering algorithms. After thorough trial and error and further consideration, we reached a solution with six clusters that encompassed the clusters created by the K-Means algorithm with the perspective of value and the clusters created by the Mean-Shift Algorithm with the demographic perspective.

We devised a plan tailored to each of the clusters that target their specific characteristics and insurance spending. Clusters that group younger customers (such as cluster 2 and 5) would have a more online centric approach, with ad campaigns tailored to the insurance needs of younger customers.

To customers belonging to clusters with expenses more pronounced in motor insurance, such as cluster 4 and 1, that represent a group of middle-aged customers with a high percentage of having children, we proposed an approach more focused in leveraging their interest in motor insurance and family security, such as providing a discount on a vehicle or sending personalized auto insurance quotas and newsletters that address topics such as family planning.

To older customers, belonging to clusters 0 and 3, that have high expenditure in Health and Household insurance, our strategy involves offering discounts on combined health and household insurance packages.

8. References

Homepage, Shap. (2023, January 1). *Welcome to the SHAP documentation—SHAP latest documentation*. https://shap.readthedocs.io/en/latest/

https://medium.com/analytics-vidhya/customer-segmentation-using-k-prototypes-algorithm-in-python-aad4acbaaede

https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html

9. Appendix

8.1 Explanation of previously unknown Methods

Shap:
SHAP is a method for explaining the predictions of a machine learning model. It decomposes the prediction of a model into the contributions of each feature, taking into account the interactions between the features. These contributions are based on the Shapley values from game theory, which measure the importance of each feature in determining the final prediction. SHAP can be used to understand the reasons behind a model's predictions, identify potential biases in the model, and improve model interpretability. It has been applied to a variety of machine learning models, including decision trees, random forests, and deep neural networks (Homepage Shap, 2023).

K-Prototypes algorithm:

K-Prototypes is a clustering algorithm that was developed to handle data sets with both numeric and categorical variables. It is an extension of the K-Means and K-Modes clustering algorithms, which are commonly used for numeric and categorical data, respectively. K-Prototypes combines the strengths of both algorithms and can handle mixed data types effectively. The algorithm works by partitioning the data into a specified number of clusters and iteratively updating the prototypes (i.e., the mean or mode of each cluster) until convergence is reached.

Calinski and Harabasz score:

The Calinski and Harabasz score or Variance Ratio Criterion is the score is defined as ratio of the sum of between-cluster dispersion and of within-cluster dispersion. Takes as input the parameters a list of n-dimensional data points and the predicted labels for each sample.

8.2 Tables

	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 1 Descriptive statistics of numerical data

	PC0	PC1	PC2	PC3	PC4
FirstPolYear	0.568684	0.280882	0.714730	-0.293954	-0.012459
BirthYear	0.909334	-0.359515	-0.141586	-0.003530	-0.020069
MonthSal	-0.882872	0.373761	0.161208	-0.000432	0.021423
CustMonVal	0.059654	-0.170803	0.435984	0.850485	0.013984
ClaimsRate_2y	0.047965	0.318225	-0.485908	-0.803377	0.063998
PremMotor	-0.501488	-0.751686	0.214017	-0.283449	0.204161
PremHousehold	0.365444	0.533505	-0.099790	0.353484	0.235055
PremHealth	0.152954	0.525179	-0.200241	0.089379	-0.789605
PremLife	0.438650	0.571443	-0.154650	0.234858	0.243463
PremWork	0.437707	0.540347	-0.142945	0.244469	0.140610
years_as_cust	-0.570384	-0.283479	-0.713299	0.291858	0.007939
prem_total_2016	0.428346	0.524047	-0.095956	0.363775	0.321879
annual_salary	-0.882872	0.373761	0.161208	-0.000432	0.021423
premium_salary_ratio	0.754828	0.079001	-0.090084	0.141522	0.149597
estimate_paid_to_cust_15_16	0.200576	0.525389	-0.498720	-0.593766	0.120664
Age	-0.909327	0.359520	0.141596	0.003513	0.020063

Table 2 Principal Components

demographic_labels_H	Age	years_as_cust	annual_salary
0	0.644440	0.754775	0.572283
1	0.233328	0.406413	0.260399
2	0.719583	0.329846	0.633532

Table 3 Mean of variables for demographic labels – K-Means

value_labels_H	PremMotor	PremHousehold	PremHealth	PremLife	PremWork	ClaimsRate_2y
0	0.254670	0.831348	0.624231	0.619386	0.488904	0.441434
1	0.669860	0.683489	0.524425	0.194657	0.245473	0.603539
2	0.340680	0.806378	0.718688	0.249715	0.484540	0.406990
3	0.701741	0.686393	0.489655	0.200663	0.214809	0.193711

Table 4 Mean of variables for value labels – K-Means

	Segment	Children	EducDeg	Age	years_as_cust	annual_salary
0	0	1	3.0	0.513038	0.641151	0.473004
1	1	1	2.0	0.231951	0.347271	0.257110
2	2	0	3.0	0.822202	0.552374	0.710110

Table 5 Mean of variables for demographic labels – K-Prototypes

demographic_labels	Age	years_as_cust	annual_salary
0	0.207275	0.363313	0.238801
1	0.623571	0.866565	0.555677
2	0.445740	0.609623	0.416177
3	0.736304	0.309821	0.647327
4	0.821135	0.664417	0.713373

Table 6 Mean of variables for demographic labels – Mean-Shift

value_labels	PremMotor	PremHousehold	PremHealth	PremLife	PremWork	ClaimsRate_2y
0	0.545834	0.737770	0.589653	0.267888	0.299457	0.415305
1	0.155002	0.815286	0.588559	0.726470	0.671340	0.448455
2	0.800619	0.000000	0.367628	0.111035	0.224732	0.769547
3	0.294480	0.642806	0.664491	0.399747	0.695575	0.727311
4	0.283610	0.000000	0.632686	0.379993	0.878810	0.858025

Table 7 Mean of variables for value labels – Mean-Shift

Combination	Silhouette Score	Number of Clusters
KMeans_gaussian_rect	0.3321	2
Hier_gaussian_rect	0.3222	2
KMeans_gaussian_hexa	0.3699	2
Hier_gaussian_hexa	0.3703	2
KMeans_bubble_rect	0.3095	2
Hier_bubble_rect	0.2832	2
KMeans_bubble_hexa	0	2
Hier_bubble_hexa	0	2

Table 8- Silhouette Score for all parameter & Cluster Combinations in SOM (value features)

Combination	Silhouette Score	Number of Clusters
KMeans_gaussian_rect	0.4441	2
Hier_gaussian_rect	0.4287	2
KMeans_gaussian_hexa	0.4587	2
Hier_gaussian_hexa	0.3776	2
KMeans_bubble_rect	0.4147	2
Hier_bubble_rect	0.364	2
KMeans_bubble_hexa	0	2
Hier_bubble_hexa	0	2

Table 9 Silhouette Score for all parameter & Cluster Combinations in SOM (demographic features)

demographic_labels	0	1	2	3	4
value_labels					
0	1290	237	203	376	318
1	1051	467	502	665	564
2	490	531	753	488	295
3	414	390	488	345	234

Table 10 Distance matrix of final clustering solution

demographic_labels	0	1	2	3	4
value_labels					
0	1290.0	203.0	NaN	NaN	931.0
1	1051.0	857.0	916.0	1010.0	798.0
2	490.0	826.0	753.0	488.0	NaN
3	NaN	NaN	488.0	NaN	NaN

Table 11 Distance matrix of final clustering solution after manually merging

product_labels	0	2	4
behavior_labels			
0	1290.0	NaN	758.0
2	NaN	NaN	2067.0
3	1955.0	1847.0	2184.0

Table 12 Distance matrix of final clustering solution after hierarchical merging

merged_labels	0	1	2	3	4	5
ClaimsRate_2y	0.806712	0.308761	0.675171	0.742124	0.897141	0.725876
PremMotor	280.896752	414.584167	288.117476	194.975156	377.700448	124.884851
PremHousehold	193.691140	96.305806	201.671304	300.692084	116.285057	424.110581
PremHealth	194.509786	120.213621	195.544638	191.610796	144.353771	184.708031
PremLife	34.472404	18.967581	22.962226	88.788762	17.693969	112.871249
PremWork	40.024596	15.702282	42.045881	53.711174	25.716045	82.209043
Age	66.554945	54.265602	31.220460	59.618734	49.697347	24.830233
years_as_cust	26.247711	30.667150	25.025575	33.857520	34.113698	22.682171
annual_salary	42338	34162	18903	37463	30940	15197

Table 13 Mean of numerical features in the final Clusters

8.3 Visualizations

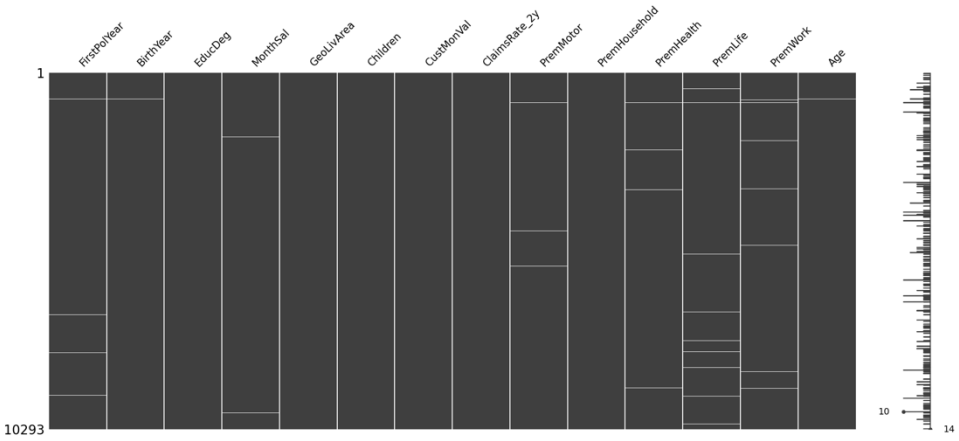

Figure 1 MSNO Matrix

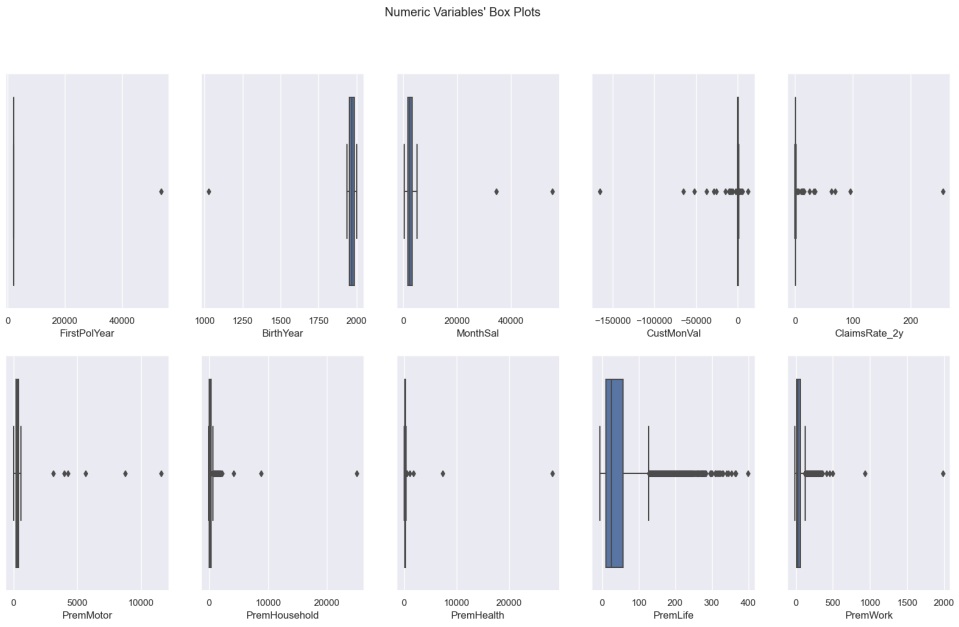

Figure 2 Boxplots of initial data

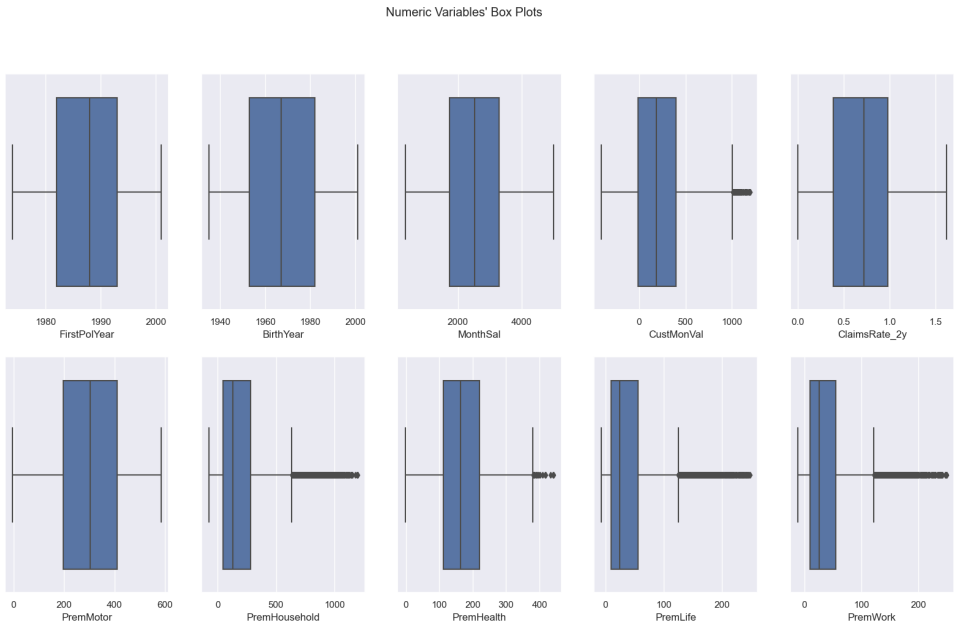

Figure 3 Boxplots of data treated with manual filtering

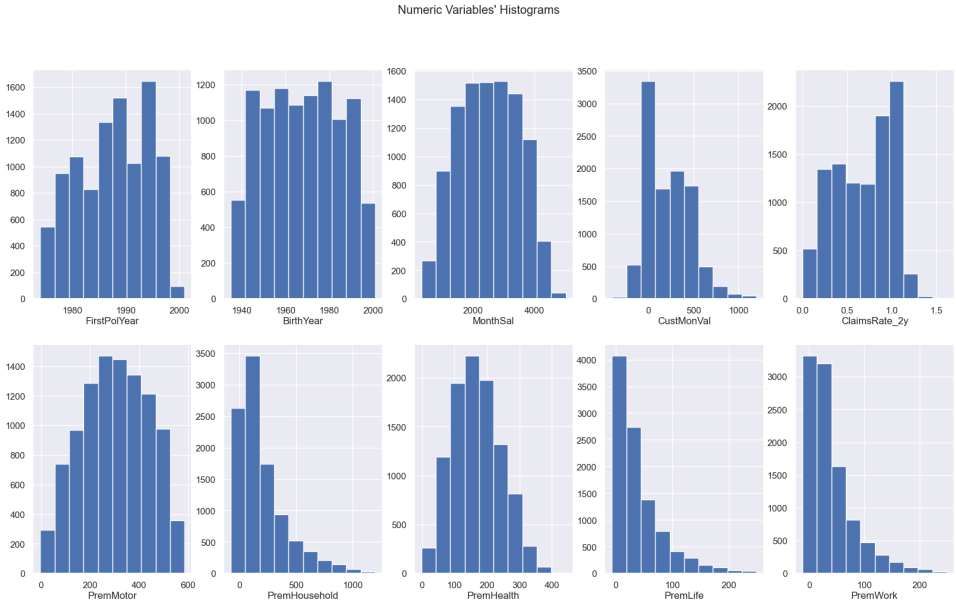

Figure 4 Histograms of data treated with manual filtering

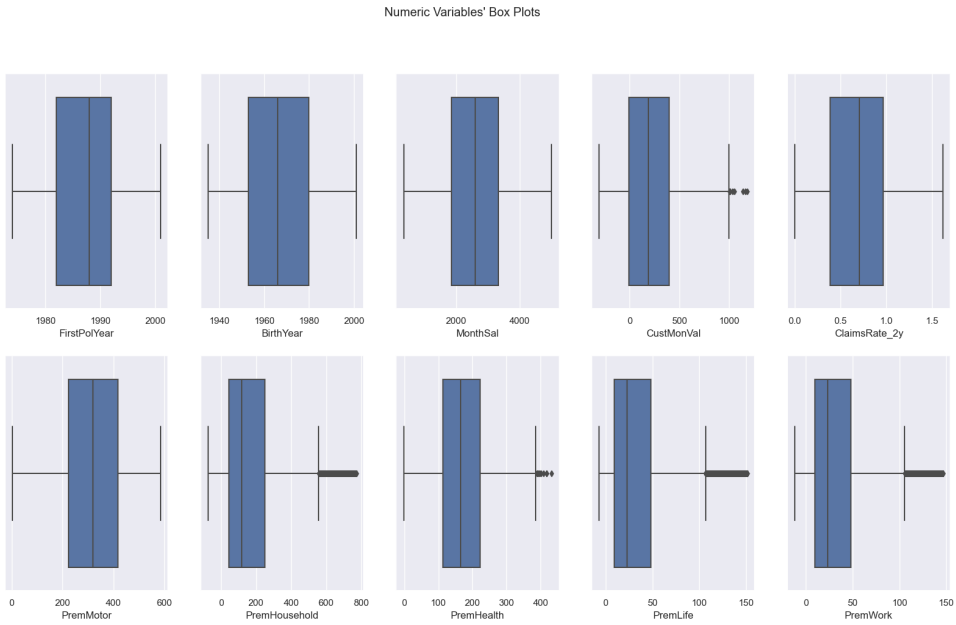

Figure 5 Boxplots of data treated with automatic filtering

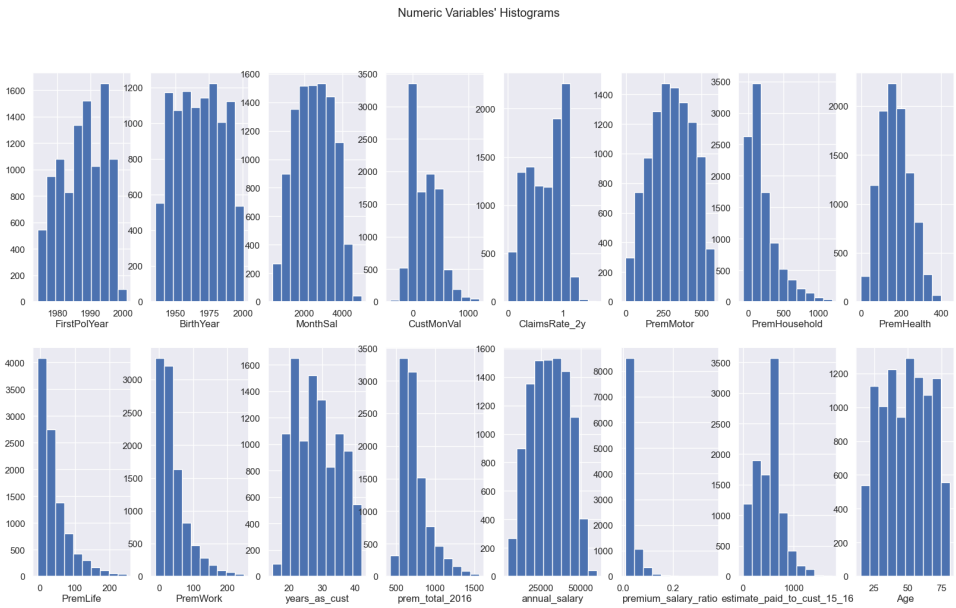

Figure 6 Histograms of data before log transformation

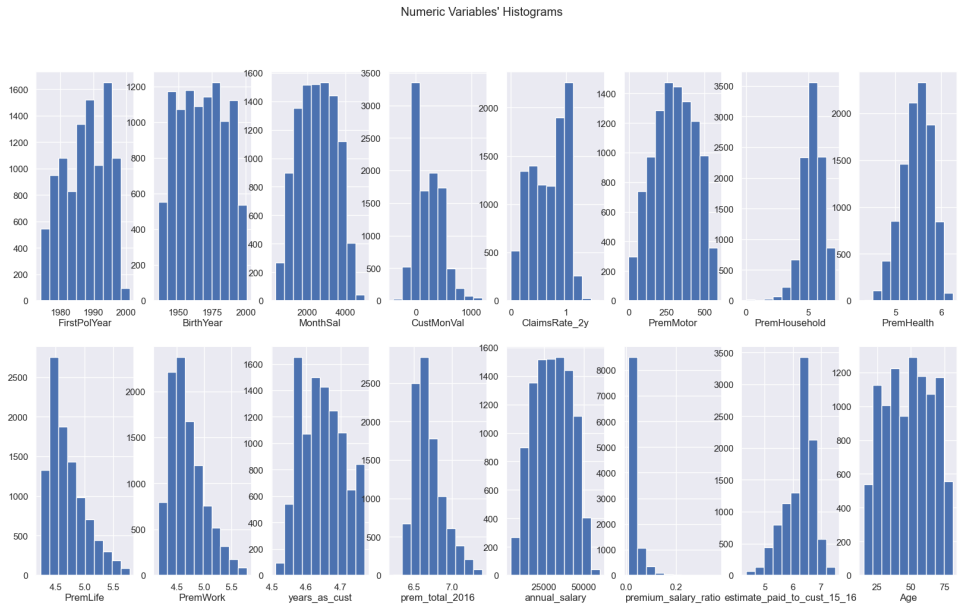

Figure 7 Histograms of data after log transformation

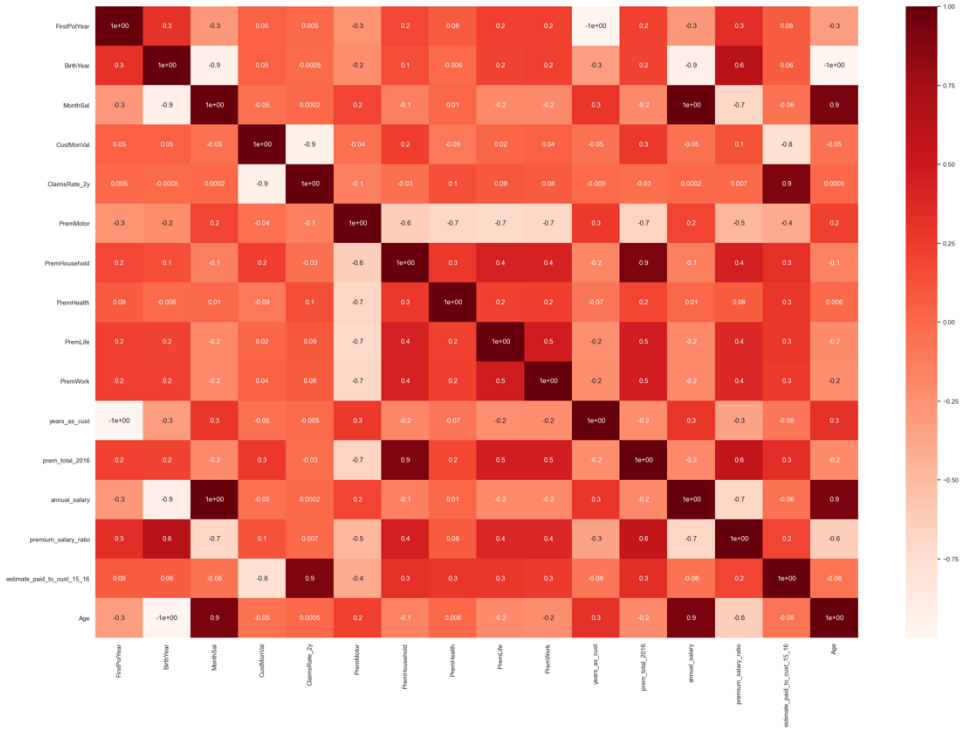

Figure 8 Initial correlation matrix

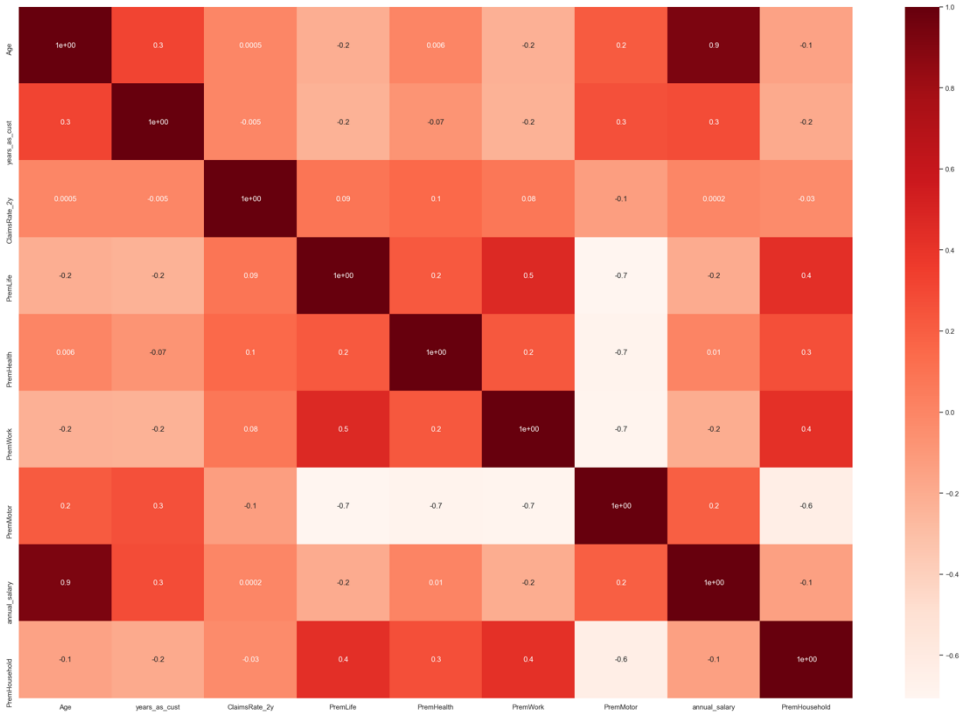

Figure 9 Final correlation matrix

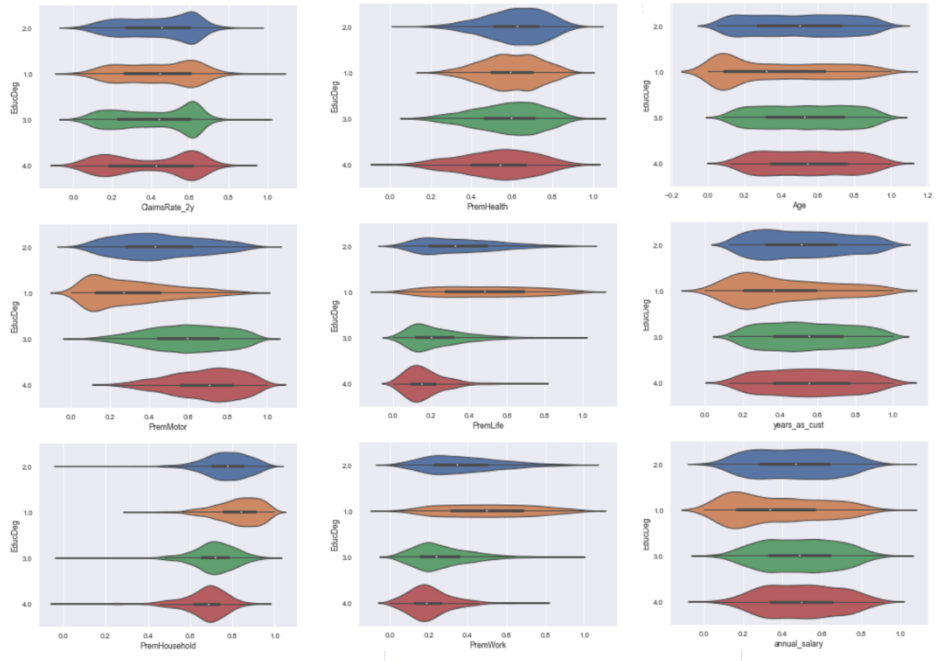

Figure 10 Violin Plots

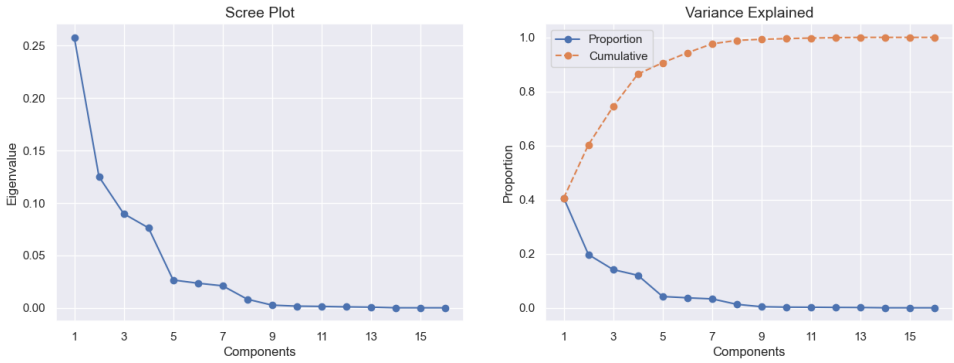

Figure 11 Principal Component evaluation

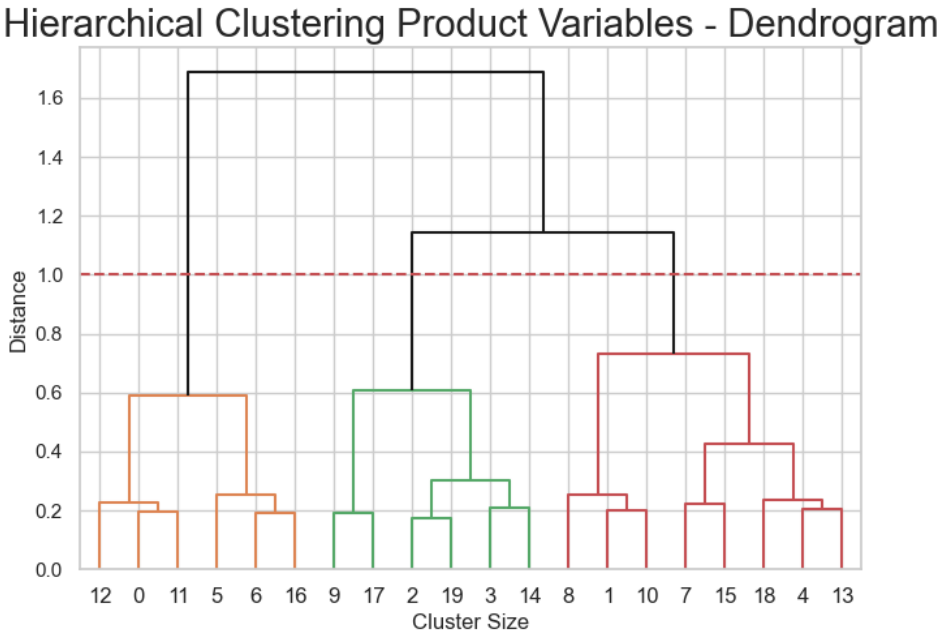

Figure 12 Dendogram of K-Means for demographic labels

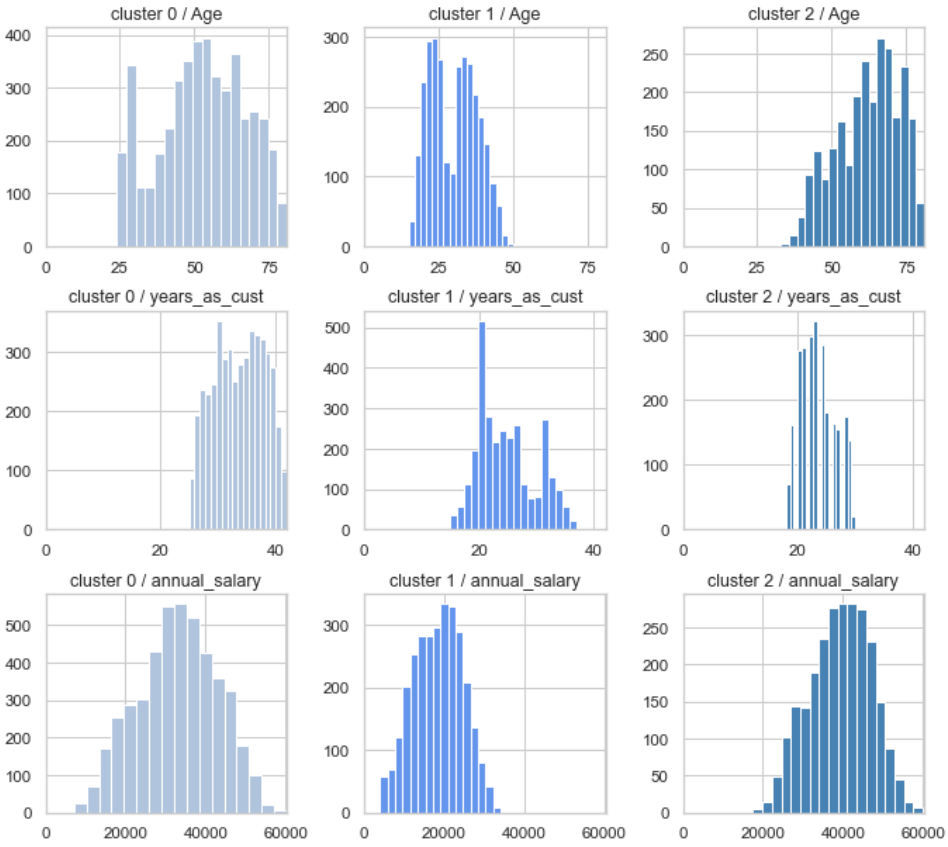

Figure 13 Histogram for demographic labels – K-Means

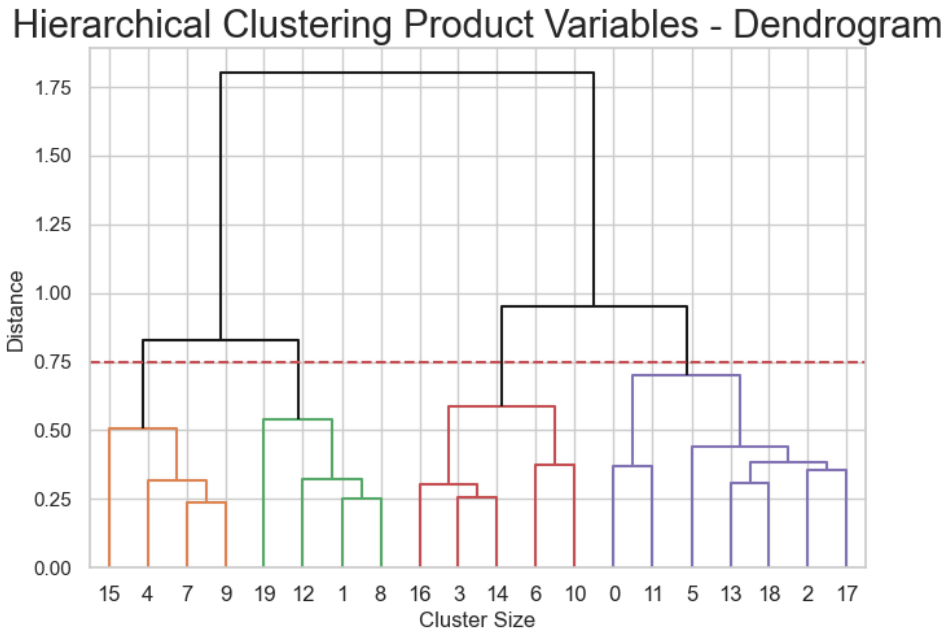

Figure 14 Dendogram of K-Means for value labels

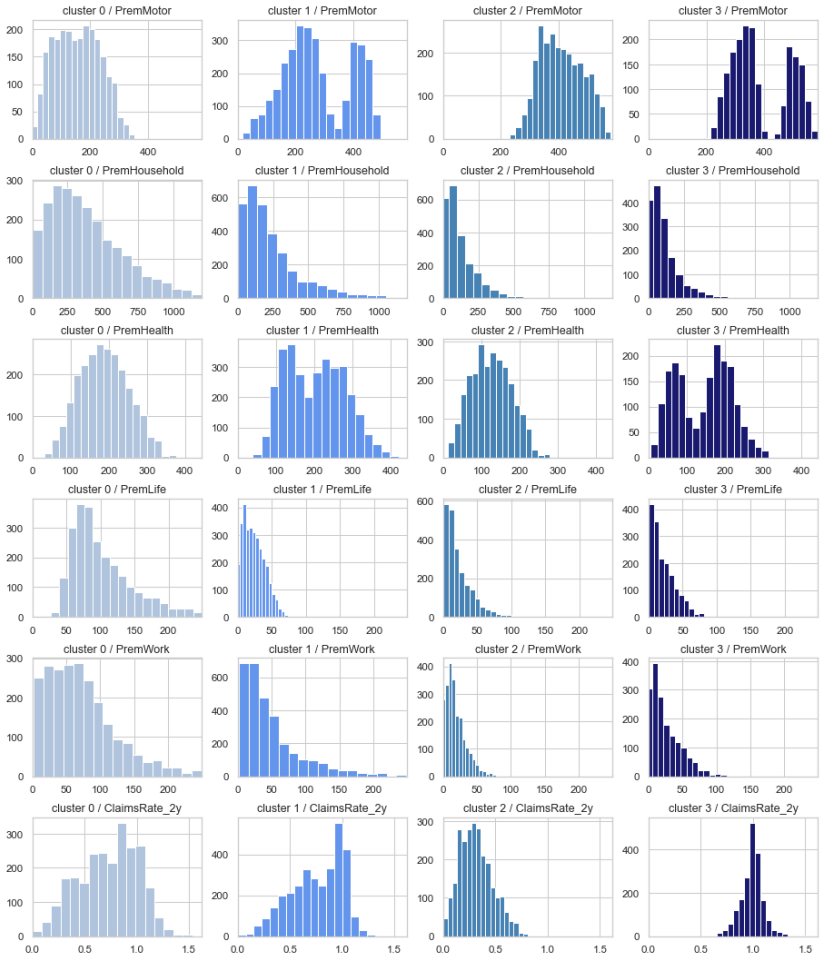

Figure 15 Histogram for value labels – K-Means

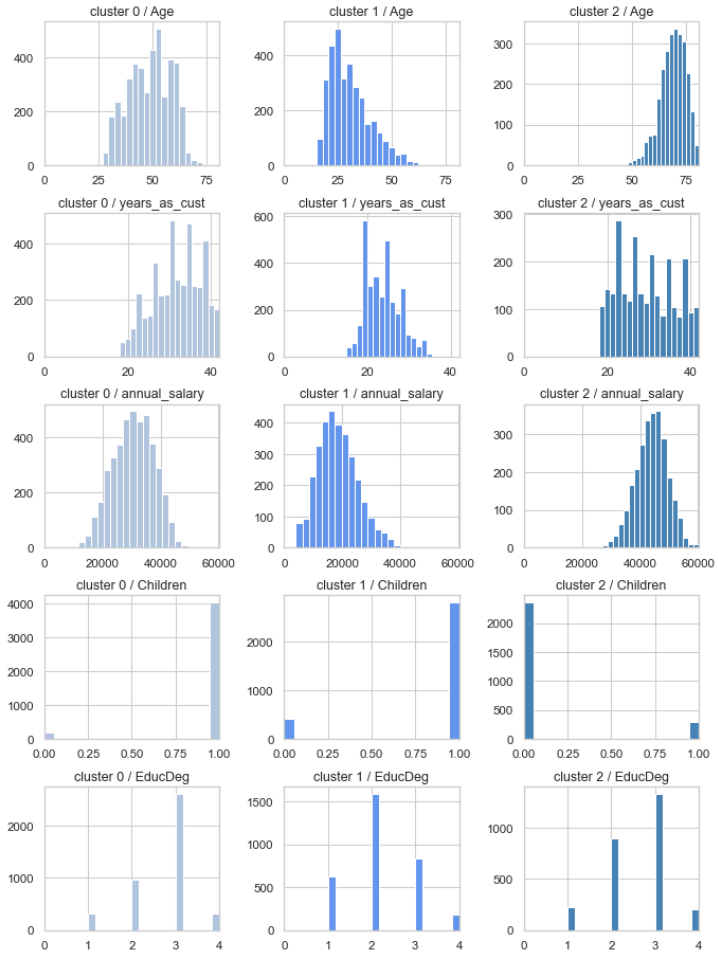

Figure 16 Histogram for value labels – K-Prototype

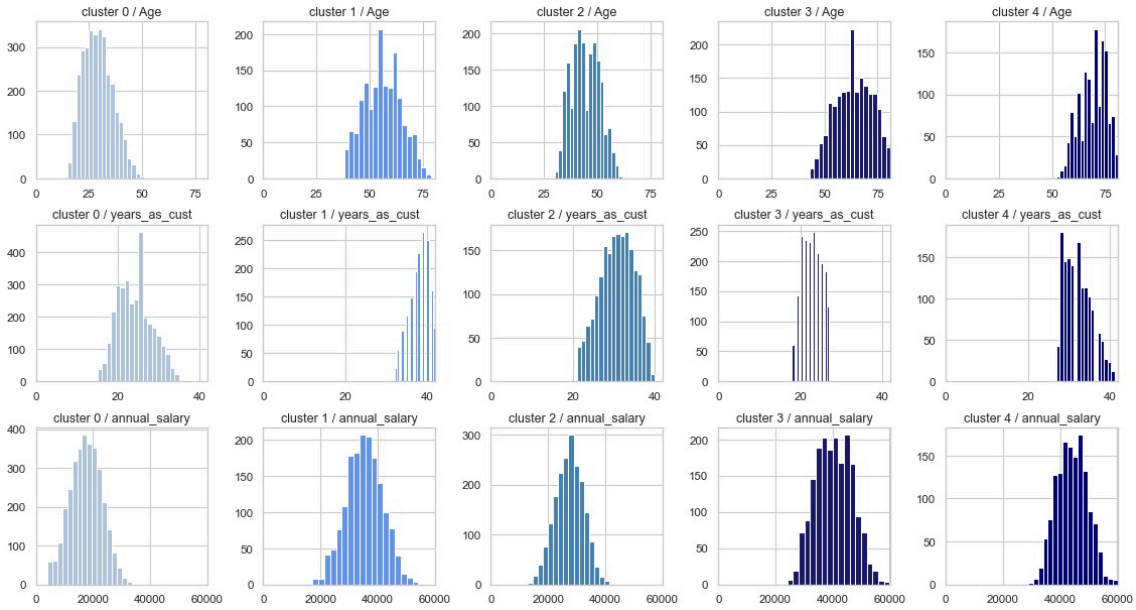

Figure 17 Histograms Mean Shift Demographic

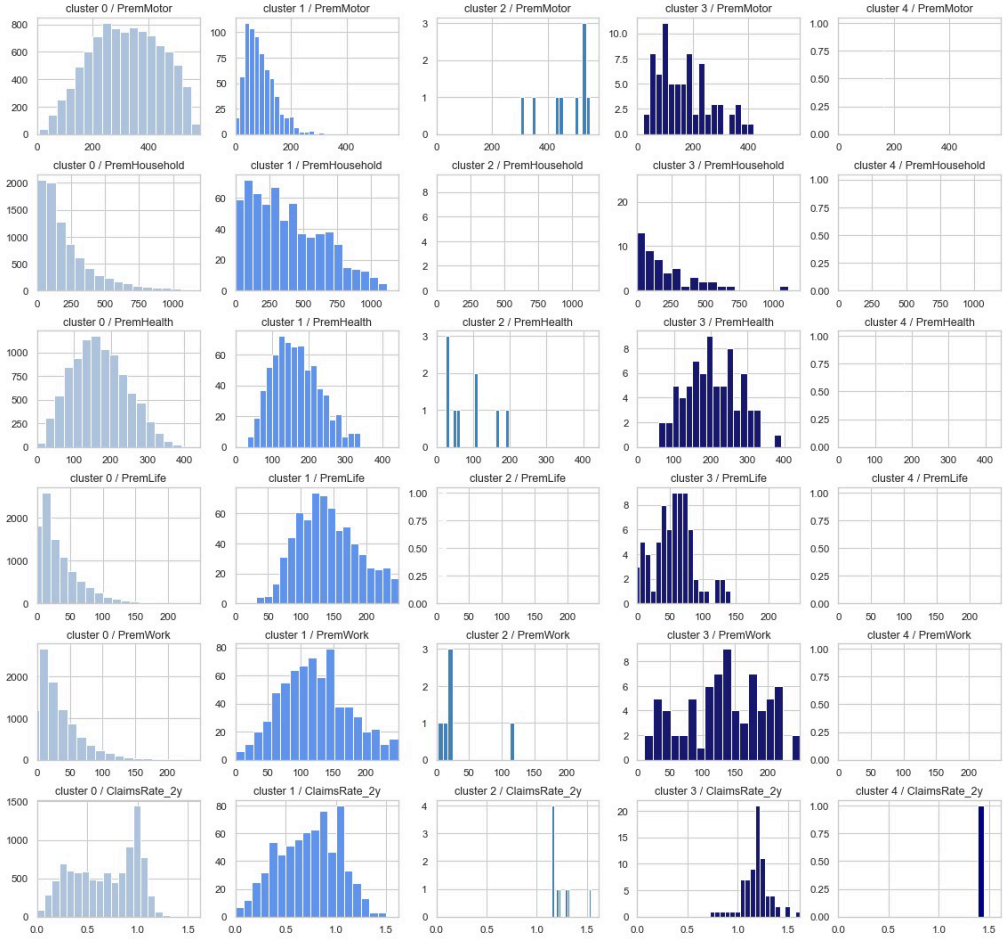

Figure 18 Histograms Mean Shift Value

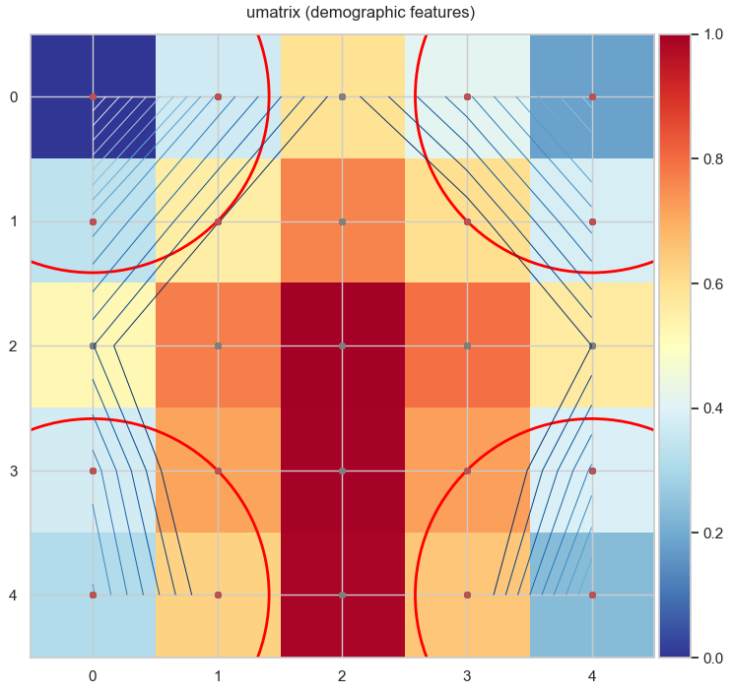

Figure 19 U-Matrix SOM size [5,5] (demographic)

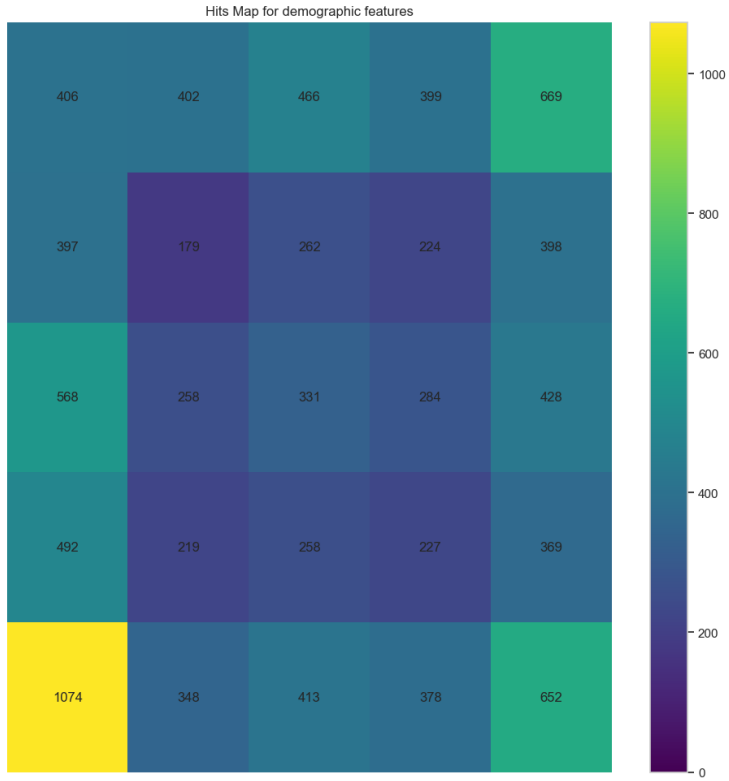

Figure 20 Hit Map for SOM size [5,5] (demographic)

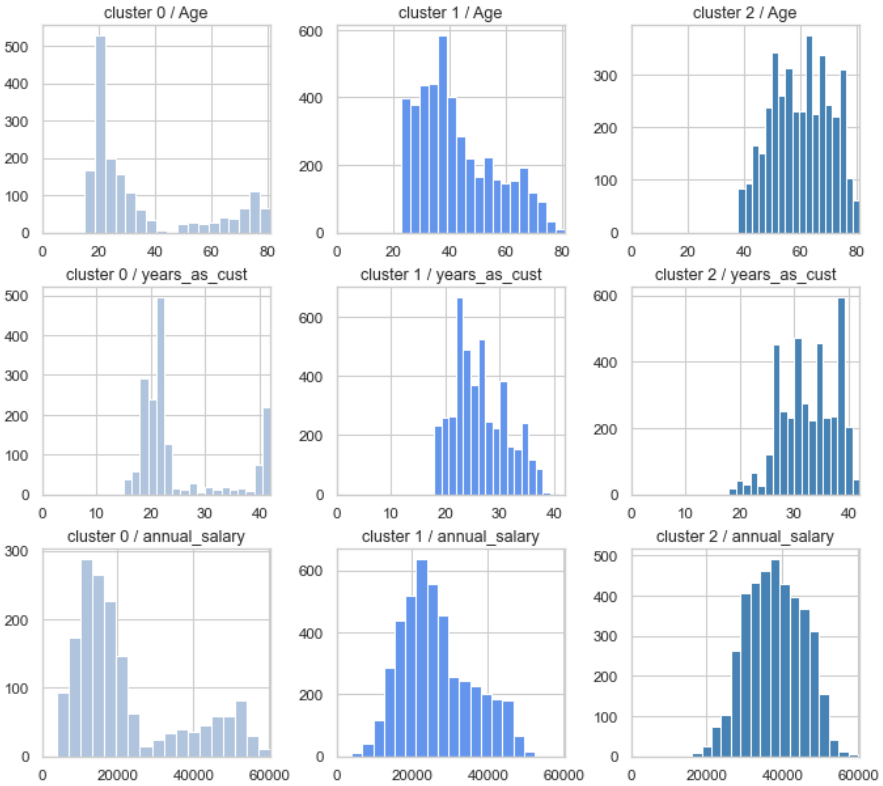

Figure 21 Cluster Histograms for SOM size [5,5] (demographic)

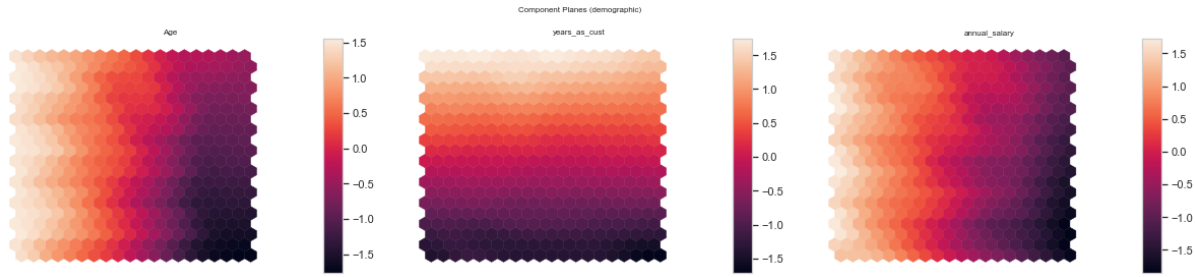

Figure 22 Component Planes for SOM size [20,20] (demographic)

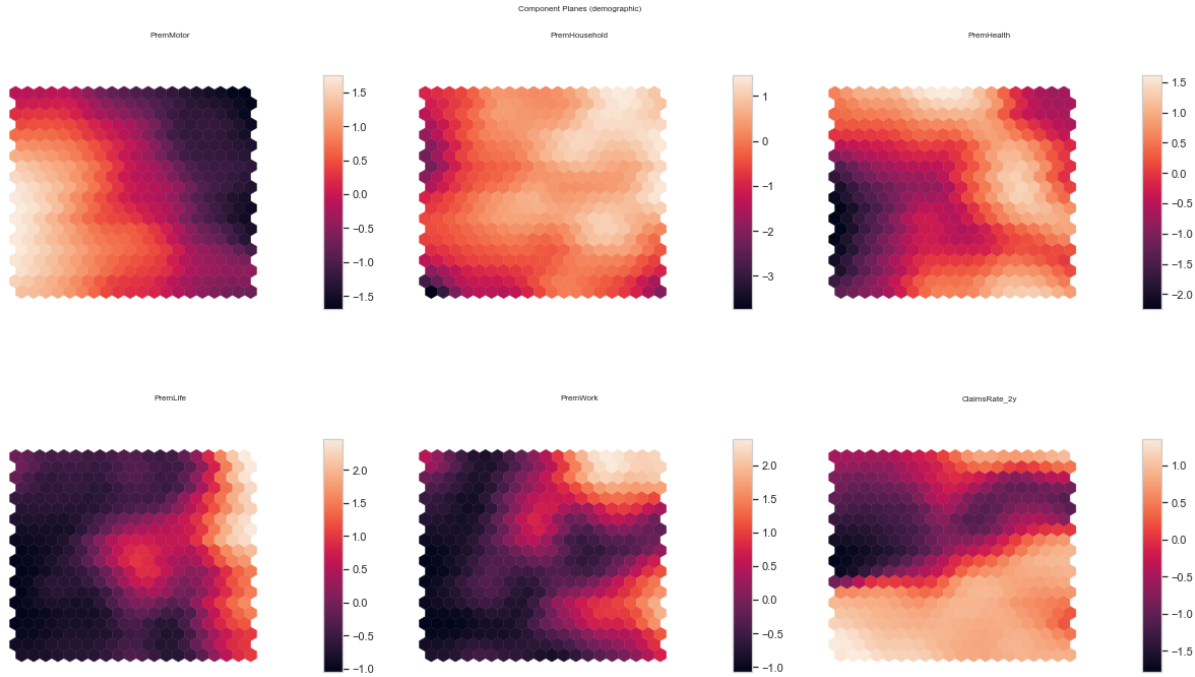

Figure 23 Component Planes for SOM size [20,20] (Value)

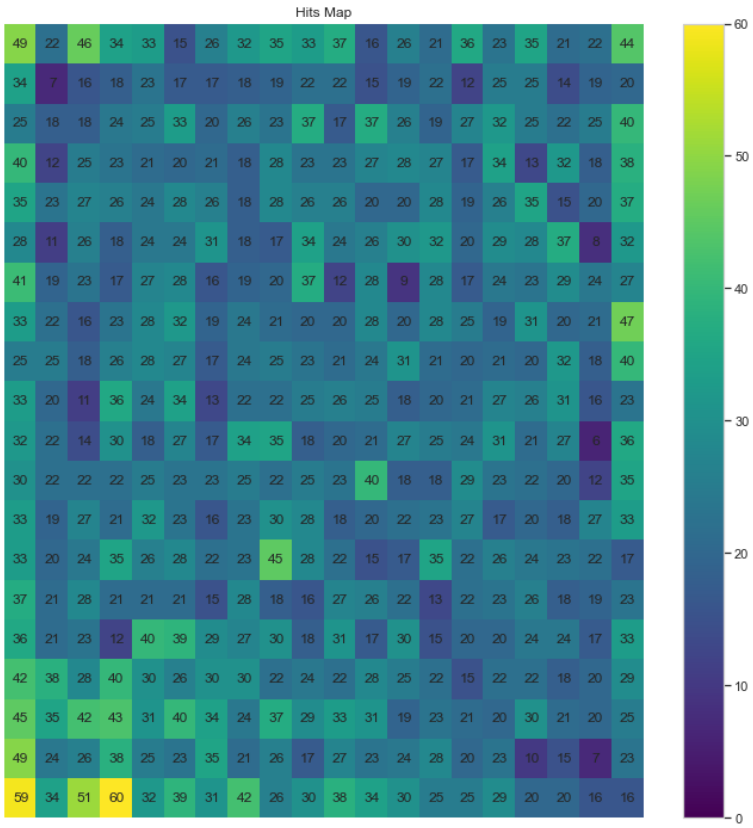

Figure 24 Hits Map for SOM size [20,20] (demographic)

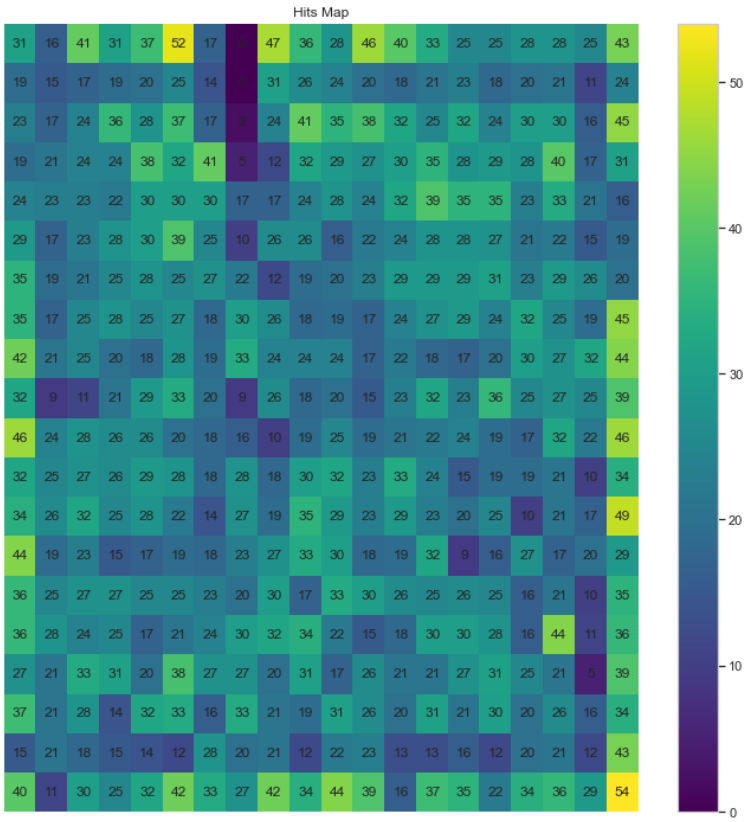

Figure 25 Hits Map for SOM size [20,20] (value)

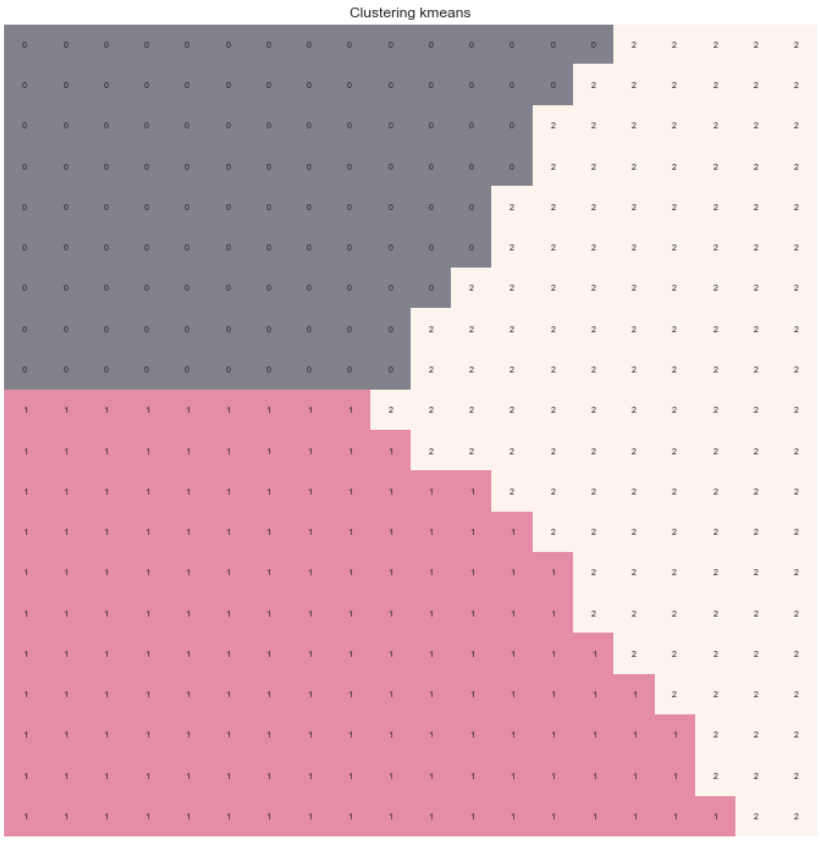

Figure 26 Clustering Nodes SOM - Kmeans (demographic)

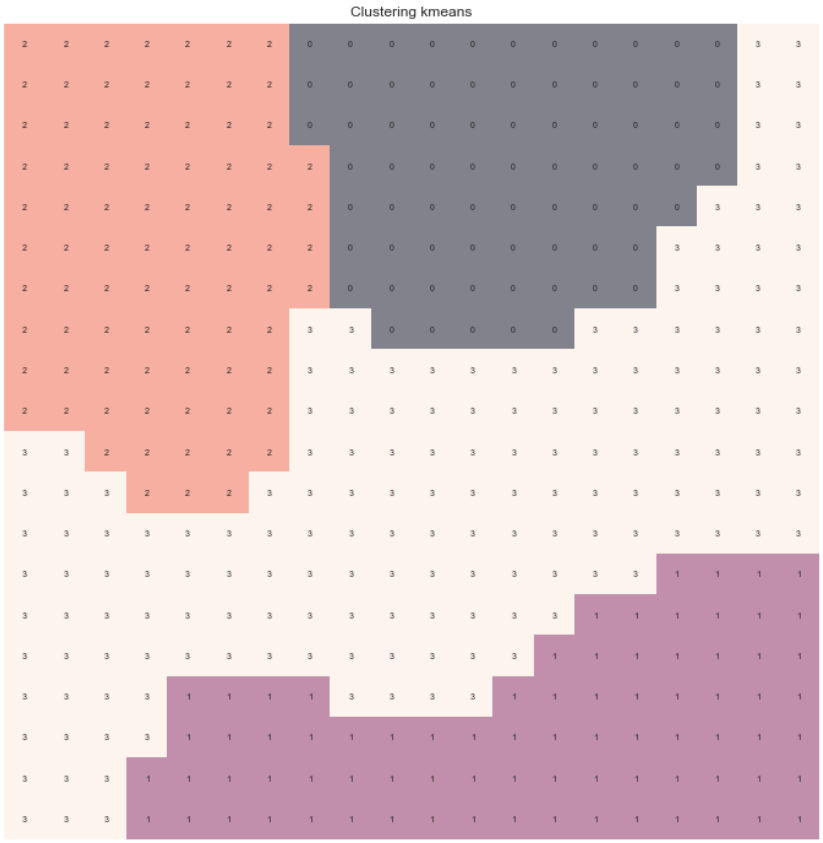

Figure 27 Clustering Nodes SOM - Kmeans (value)

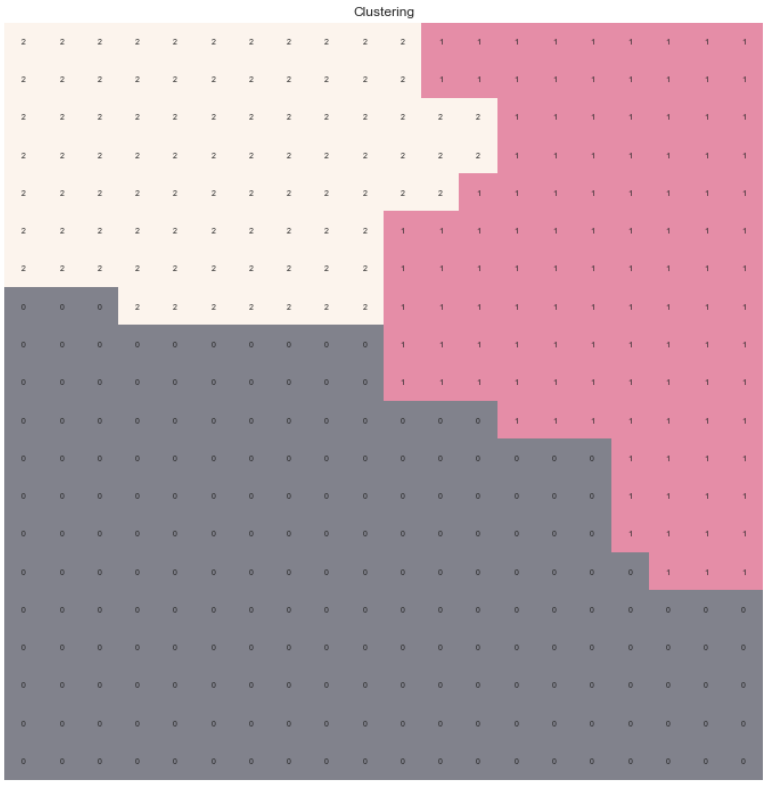

Figure 28 Clustering Nodes SOM - Hierarchical (demographic)

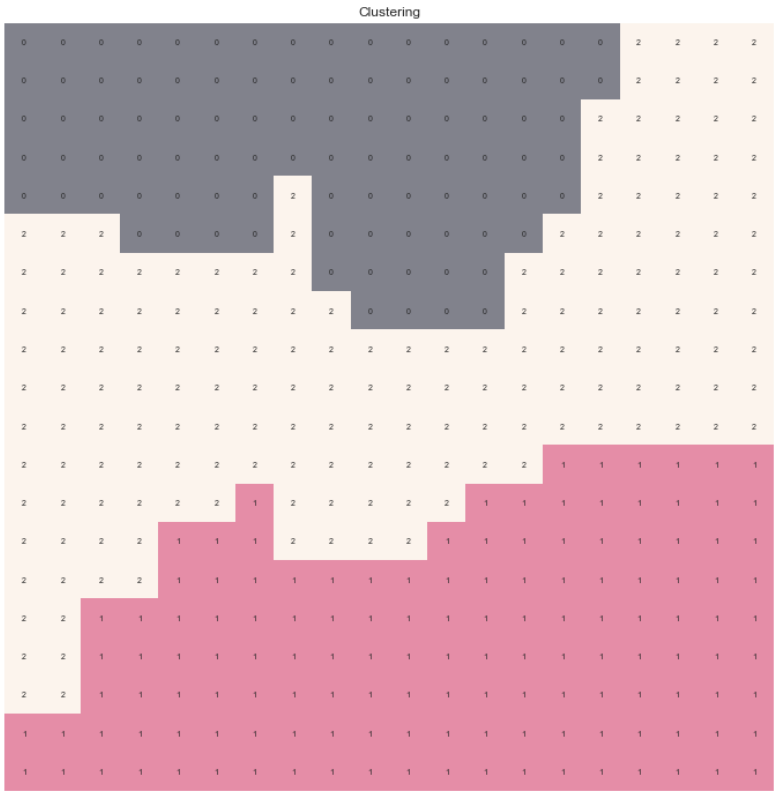

Figure 29 Clustering Nodes SOM - Hierarchical (value)

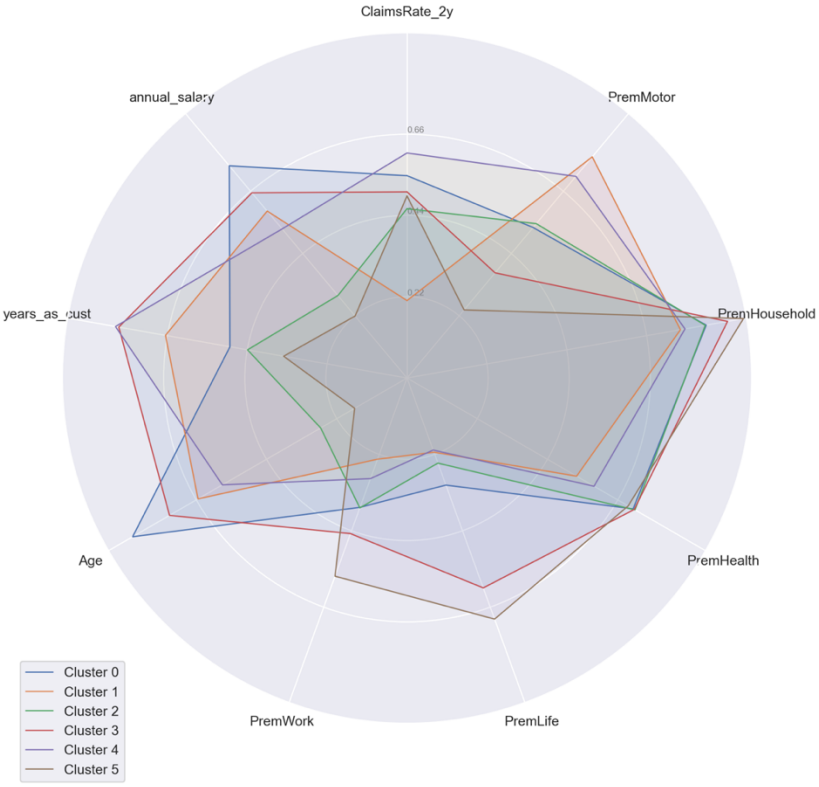

Figure 30 Radar Plot Final Clustering

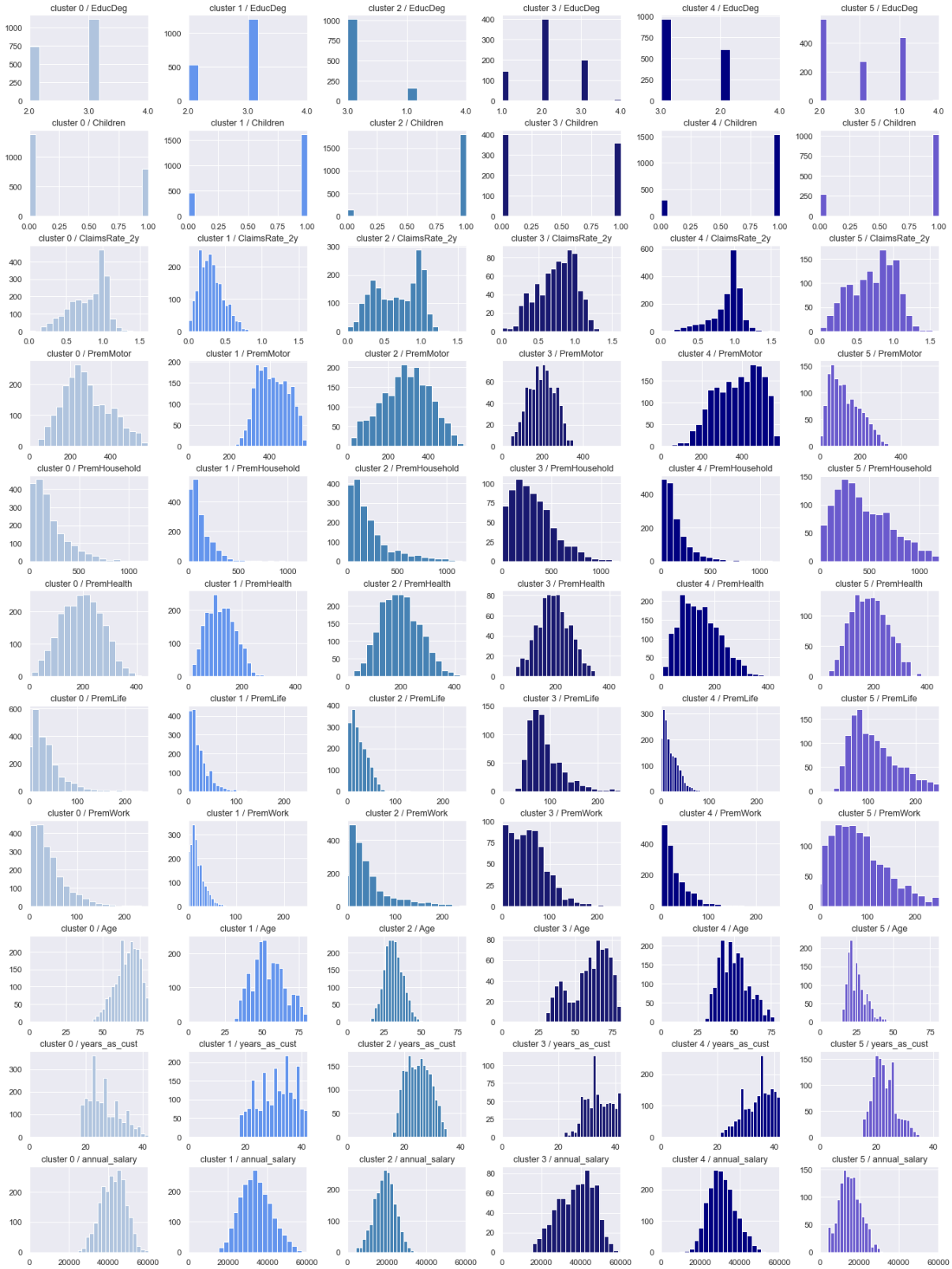

Figure 31 Histograms Final Cluster Variables

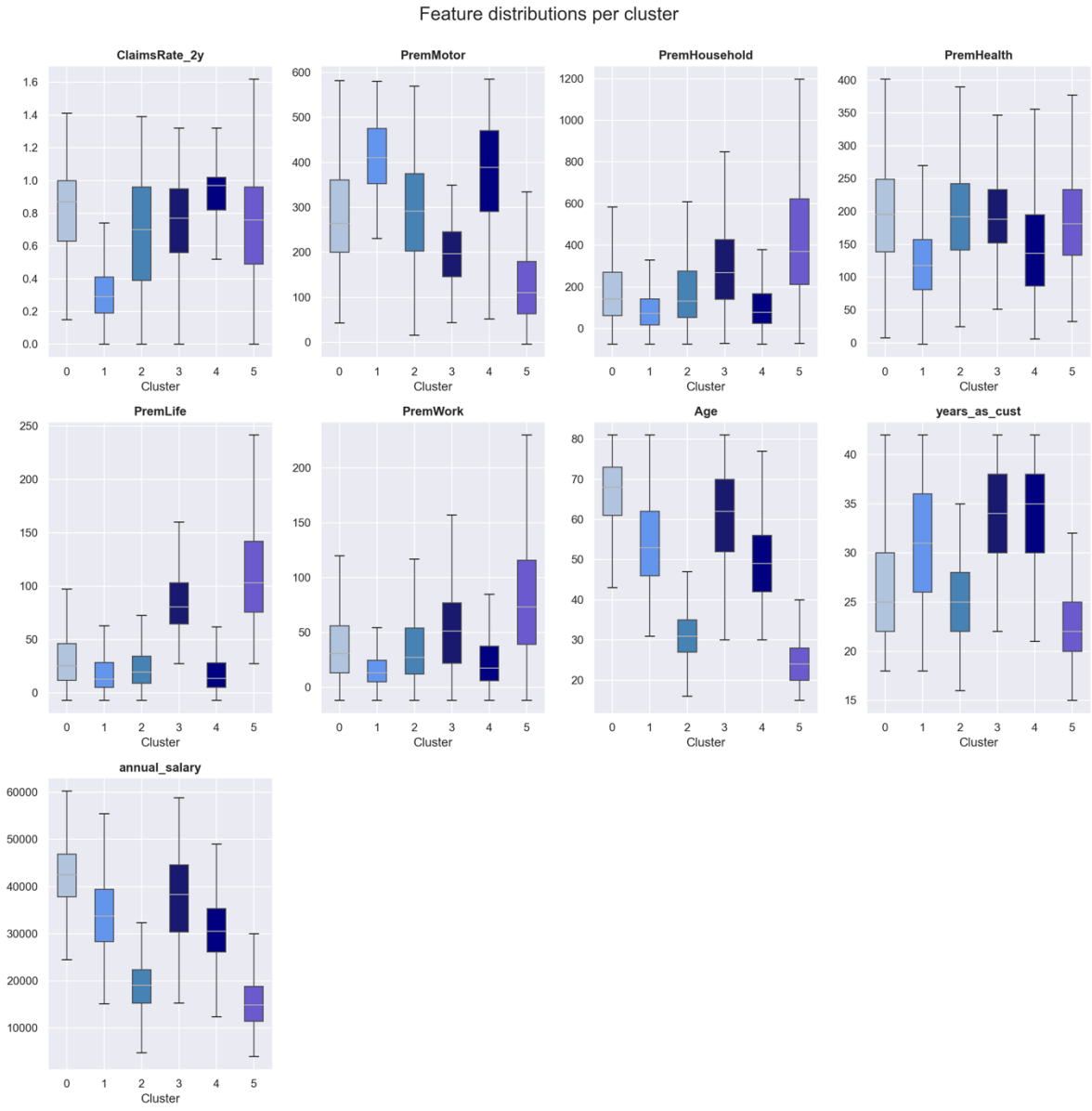

Figure 32 Boxplot Final Clustering

|^

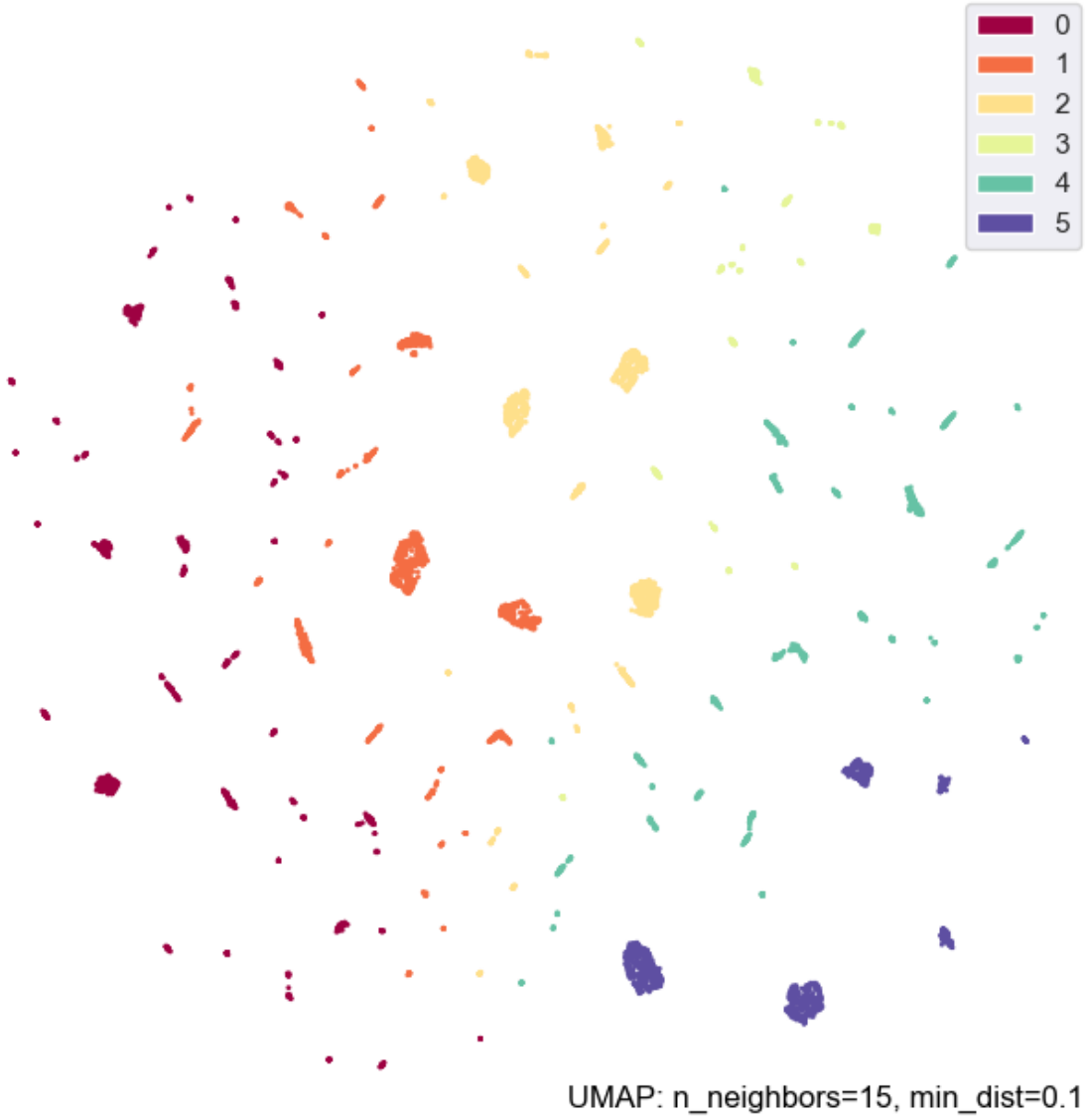

Figure 33 UMAP Final Clusters

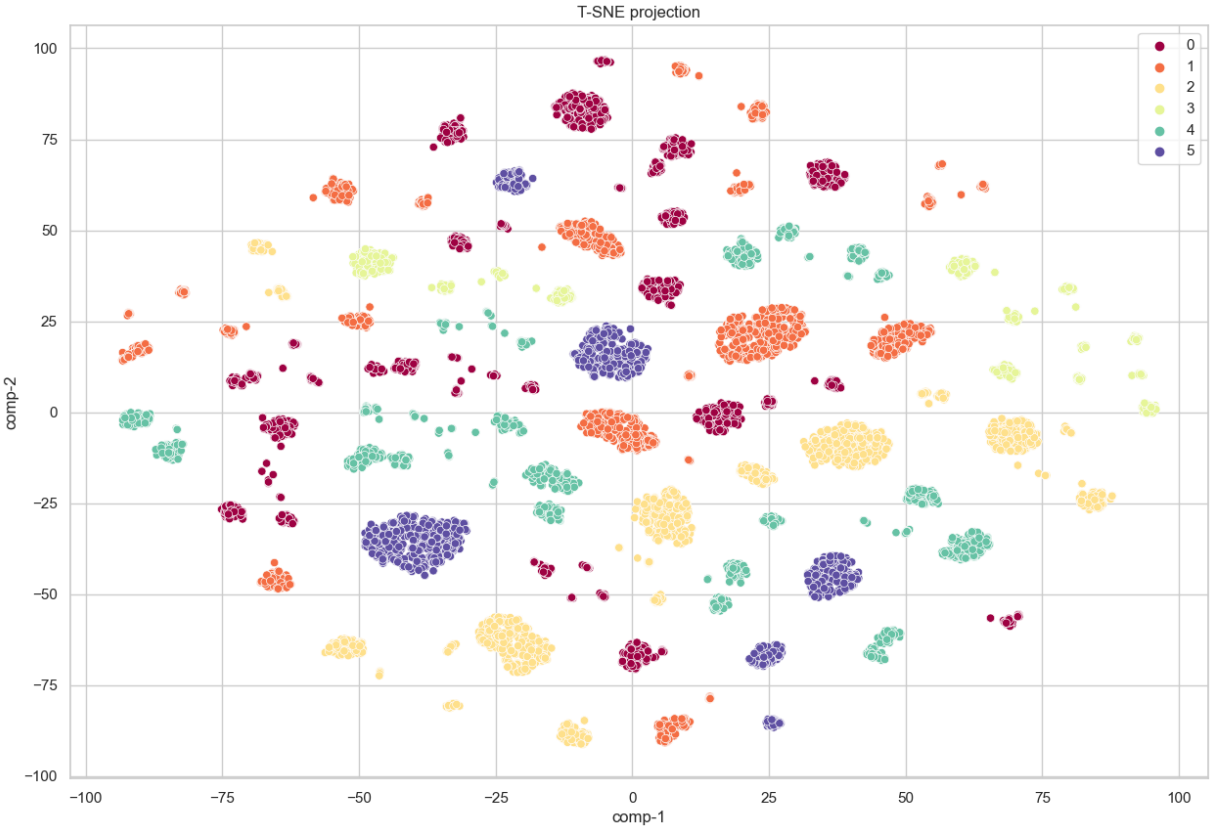

Figure 34 UMAP 2 Final Clusters