Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Customer Segmentation

Group BK

Carolina Machado, number: 20210676

Francisco Calha, number: 20210673

Sara Arana, number: 20210672

January, 2022

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

INDEX

1. Abstract and Introduction

1.1. Abstract

The goal of this project is to develop a customer segmentation by achieving a possible clustering solution for the Insurance Company. To do so, we used a dataset with 10 290 observations and 13 features. Before developing our solution, we started by exploring the data pre-processing process. We went through feature engineering, outlier detection methods, encoding of categorical variables, and feature selection. After, we explored different algorithms, such as hierarchical, partition and density-based methods, among others. The best solution we obtained was using the K-Means algorithm, and we ended up with 5 customer groups. To have some visual information of our solutions we also explored the UMAP method.

1.2. Introduction

The goal of this project is to conduct a Customer Segmentation for the Marketing Department of an Insurance Company, in order to better understand the customer's different characteristics. We aim to explain and describe the different clustering solutions we obtained and, finally, to approach different Marketing techniques for each different group of clients.

2. Data Preparation

Before building our clusters, we started by looking at our data and understand the different processes we had to take into consideration for it to be operable.

2.1. Data Exploration

On a first instance, we saw that we had data regarding 10296 clients and 13 features (table 1). Then, we proceeded to understand how our dataset was built, by looking at the datatypes of our variables, and at the possible values our features could take. From here, we noticed we had some missing values represented by 'nan'. We also defined our variables according to their types, as we can see from table 2. Afterwards, we checked for possible inconsistencies in our data. We found that there were some records in which the variable 'FirstPolYear' happened before the 'BirthYear', and since it is not possible for someone to have an insurance before they are born, we figured it might have been an error. We also saw that there were some observations in which the client was under 16, and since it is not legal for someone under 16 to work, we would have to deal with these records later. For the 'EducDeg' variable, we saw that there were some observations completely missing (empty rather than with 'nan'), so we replaced them by 'nan', in order to fix this problem later on.

2.1.1. Feature Engineering

Regarding the observations in which the 'BirthYear' value was higher than the 'FirstPolYear', we thought that it might have been an error on the part of who put the data there – they might have switched both columns. As such, we decided to switch them back. Using this approach, we no longer had clients whose age was lower than 16, so, two of our problems were solved.

2.1.2. Further Data Exploration

After this step, we did some further data exploration (which we could not perform correctly before, due to the problems in our data) using the describe function. For the numerical features (Table 3), we can see that we have some extreme maximum and minimum values, which need to be treated carefully. Since these values are very distance from the others, they might jeopardize our clustering solution and, consequently, affect our analysis. Regarding the 'PremHousehold' variable, we can see that it might have outliers, since the mean and the median have sparse values. In the remaining premiums, and in 'CustMonVal', the difference is of about 10 units, so we need to better check if there are actually outliers. For the 'FirstPolYear' and 'BirthYear' variables we can see both have nonsense values, namely in their maximum and minimum, respectively. Regarding the 3 categorical variables, we plotted 2 bar plots (graph 1 and 2). From here, we can see that the vast majority of the customer base has 1 child, most live in area 4, and hold a master's degree.

2.1.3. Duplicate Values

In this step, we checked for the possible existence of duplicate values. In a first instance, we realized there were 3 rows fully duplicated, so we dropped them. Afterwards, we came to the conclusion that there were partially duplicated rows (Fig. 3), that the problem did not read as duplicates since they had NaN values in one column. We decided to eliminate these observations by their index, since we were dealing with just two observations.

2.2. Missing Values

2.2.1. Detecting Missing Values

For this portion of the pre-processing part, we realized we had missing values in almost all features. So, we proceeded to compute the percentage of missing values we had in each of them. By rule of thumb, 25 to 30% of missing observations is the maximum recommended; else, the variable should be dropped. However, this did not happen in our dataset, the variable with the most missing values, had only 1% of its data missing.

To better understand the missing values per column, we checked the nullity correlation between our variables (fig. 1) and we could see that there are no highly correlated variables, regarding their missing values. This means that the existence of a missing value in one variable, is not related to the existence of another in a different variable.

Regarding the rows, we decided to plot a missing values matrix (fig. 2), from which we can easily possibly identify patterns in our data. This visualization, showed to be very useful because we could see that the maximum number of missing columns in one row was 4 (number 9 on the sparkline on the right, it means that 9 of the 13 possible values were not missing). We decided to further explore these observations with 4 missing values, and we found an interesting resemblance between them – all of them had 'ClaimsRate' at 0, 'CustMonVal' at -25, 'PremHouse' at 0 (notice that these values were obtained from the dataset before the scaling process). There was a total of 12 observations following this pattern, so we needed to proceed carefully.

2.2.2. Filling Missing Values

Several approaches will be used to fill in the missing values in our dataset. First, we will replace all the missing values in the Prem features to 0. The reason for that choice is that is not mandatory for a client to have all the insurances. Due to this fact, it doesn't make sense to fill them with any other value. The second imputation method we chose to fill the remaining missing values was the K-Nearest Neighbors Imputer method, because of its simplicity and fast implementation.

Regarding the categorical variables, we will also deal with them in two different ways. For the 'EducDeg' and 'GeoLiv' features, we decided to fill the missing values with the mode of the column – since each of them have 4 labels, and the more labels one variable has, the harder it is to obtain accurate predictions. For the 'Children' variable, since it is a binary one, we decided to predict the missing values. For the predictions, we choose the DecisionTreeClassifier once it's less affected by outliers.

After this process, we realized there were some observations for which all premiums had 0 values (many of these observations corresponded to those observations in which there were 4 missing values). We eliminated these observations because they do not bring value to our problem – they refer to clients that do not have any insurance.

2.3. Encoding Categorical Variables

Since we are dealing with a dataset with categorical independent variables, we figured it was important to encode them using OneHotEncoder. By doing so, each feature value of each feature will be a new binary variable (with values 0 or 1).

2.4. Splitting Variables

From the Data Preparation process already done, we can identify two different segmentations. One regarding to the features related with client's information, composed by 'FirstPolYear', 'BirthYear', 'MonthSal', 'ClaimsRate', 'CustMonVal'. Another related to the insurances of the company, composed by 'PremMotor', 'PremHousehold', 'PremHealth', 'PremLife', 'PremWork'.

In order to have a better perception of our clusters, our objective is to build a clustering solution for each of these groups of variables, and, in the end, merged them into a final solution

2.5. Outliers

We decided to remove outliers from our dataset, since the presence of them can negatively impact a large number of clustering algorithm. With that being said, our goal is to either remove them for now, but add them at the final clustering solution – so that they are part of the solution but did not contribute to it.

The following steps were applied to both segments. On a first instance, by looking at each variables' boxplot and histogram, we figured we were not being able to have a good visualization of the distribution of our data due to the presence of extreme outliers. As such, we decided to manually remove these extreme observations from some variables. On a second step, we used the IQR method to remove more outliers not so extreme but also with high values. Afterwards, we opted to use the DBSCAN approach to remove outliers in base of density. In a first instance we ended up removing around 3 per cent of our data, however, across the clustering process we could identify that we were still handling with some noise. Afterwards, we decided to also implement Local Outlier Factor (LOF) for anomaly detection in other to reduce the noise and get a better visualization. With this done, we ended up removing around 5 per cent of our data.

Finally, we checked the absolute frequencies' distribution in the categorical variables (graph 3), which led us to the conclusion that there were no outliers to be dealt with.

3. Data Pre-Processing

3.1. Data Scaling

When trying to build a clustering solution, algorithms frequently require the computation of a distance metric. To be able to do so, we scaled all our data using MinMaxScaler, was the one leading to best visualizations and R-squared values.

3.2. Feature Selection

Feature selection is an important step in data pre-processing due to not only the fact that the reduction of the input space brings simplicity, but also to the fact that the model might be biased if we do not take redundancy into account. For this step, we plotted a Pearson Correlation Matrix for both datasets (fig. 4 and 5).

For the clients features matrix, we had two pairs of variables highly correlated (-1), so one for each pair would have to be removed. We decided to keep the ‘CustMonVal’ variable rather than ‘ClaimsRate’ because the latter had lower correlations with the other variables. Same reason for keeping ‘MonthSal’ over ‘BirthYear’. The final features for our clients’ dataset were the following 'FirstPolYear', 'MonthSal', and 'CustMonVal'.

For the prem features matrix we had several highly correlated features, however, we noticed the ‘PremMotor’ variable was highly correlated (always above 80%) with all of them, so we dropped it. The final features for our premium dataset were the following 'PremHousehold', 'PremHealth', 'PremLife', 'PremWork'.

4. Clustering

In the following section, we will approach several clustering algorithms for both our datasets. After figuring which cluster provides us the best solution, we will merge both clustering labels.

Note: the categorical features will not be used in the cluster building process because most clusters require distance calculations, and it would jeopardize the solution.

4.1. Clustering Algorithms

For the clustering algorithms' parameters, we defined most of them by trial-and-error, for other we computed them several ways and found the best one for our model. We decided to assess the quality of our clusters using the R-squared metric, which ranges from 0 (worst) o 1 (best) and it tells us how much of the response variable is explained by independent variables.

4.1.1. K-Means and Hierarchical Clustering

K-Means

For the different segmentations, first we try to assess what is the best number of clusters for K-means algorithm, using three different metrics – distortion, silhouette, and calisnki harabasz. We defined 3 as the optimal K for the clients' segmentation and ended up with a R-squared score of 52,04% (fig. 6). Regarding the prem features, we decide that the best number of clusters is also 3 with a R-squared score of 49.73% (fig. 7).

Hierarchical Clustering

To further explore the results from the agglomerative clustering, we first discover what is the best linkage method (fig. 8 and 9) for our segmentations. Afterwards, we plotted a dendrogram to better understand how our clusters were built and what is the best number of clusters (fig. 10 and 11). From this method, we obtained that 3 is the best number of clusters with a R2 score 10,64% for the clients' segmentation and for prem segmentation 3 is also the best number of clusters with a score of 45,04%.

4.1.2. Gaussian Mixture Algorithm

Afterwards, we decided to apply the Gaussian Mixture algorithm, to our data. This method allows for non-spherical shaped clusters, and it does not define one point to a certain cluster, but rather provides the probability of it belonging to the cluster. The algorithm was initialized 10 times, for several K's (1 to 8). To have the best solution possible, we applied both information criterions – BIC and AIC -, to see if they differed much. From fig. 12 and fig. 13, for clients features and prem features. respectively, we can see the values for BIC and AIC, and we confirmed the metrics did not vary much. For the clients features, we defined the number of components to 3, and obtained a R-squared of 51,73% and for the prem features, the number of components was set to 2, and we got a R-squared of 29.42%.

4.1.3. Self-Organizing Maps

Finally, we figured it was important to explore the Self Organizing Maps method, because it not only builds clusters but also helps us visualize our data. From the visualizations obtained from applying K-Means and Hierarchical Clustering on top of SOM, we understood a little bit better how our data was behaving, especially regarding outlier and coherence detection – with many misclassified observations. This analysis is what led us to remove almost 5% of our data.

For the clustering solutions of K-Means on top of SOM, we obtained a R-squared of 15,12% and 9,6% for clients' segmentation and premium segmentation, respectively. For the clustering solutions of Hierarchical Clustering on top of SOM, we obtained a R-squared of 16,41% and 1,7% for clients' segmentation and premium segmentation, respectively.

4.2. Individual Visualization and Interpretation

After exploring all these algorithms, we decided to perform some visualizations regarding each dataset's clustering solutions. To do so, we chose the Uniform Manifold Approximation and Projection (UMAP) approach, and we applied the clustering for which the R-squared score was the highest – in both cases, it was K-Means, with K equal to 3. Fig. 14 and 15 show how our clusters are distributed, using UMAP.

For the client behavior characteristics dataset (clients' features), we can see, in Table 4, the average values for all features used to define the consumer profile. From here, we defined 3 different clusters:

- Cluster 0: corresponds to the oldest customers, those that are clients for the longest time

- Cluster 1: corresponds to the lower income clients

- Cluster 2: corresponds to the most recent customers (whose 1st policy year was the most recent) and whose salary is the highest – our potential best clients

For the service characteristics, i.e., premiums, dataset (premium features), Table 5 provides the average values for all features used. From here, we defined other 3 clusters:

- Cluster 0: refers to the clients that spend the most on health insurance

- Cluster 1: might be the target customers we aim to focus on, because they have relatively good values in all the premiums, hence they are the most reliable clients

- Cluster 2: customers that spend the less in insurances (their premium values are symbolic)

4.3. Merging Labels and Interpretation

On a first instance, we started by visualizing the 9 clusters, that refer to the merged labels of the two segments (fig. 18). From this visualization, we can see that some of them have low cardinality and others are very sparse. So, we will try to understand which clusters would make sense to merge. To do so, we used the Hierarchical Cluster (HC) algorithm (fig. 16 and table 6 show the final solution using this method), which provided the dendrogram (fig. 17) that is helpful because it allows us to see the distances between clusters. However, we will question all merged solutions that the HC method made, to see if they make sense in our case.

- **(0,0)** with **(2,0)**: it is the first cluster merged when using the HC method, both clusters are made of very similar clients in almost every feature – the only difference is that some customers from cluster (2,0) are more recent to the company; we will trust HC rationale in merging these two clusters because both have clients with good salary and a major preference for the health insurance.

- **(1,0)** with **(1,2)**: the HC method also merged these two clusters and we figured it did so because they are almost similar in all features, except for the premium value in health insurance; however, the thought we could have very interesting conclusions by not merging these two groups. Both clusters have a relatively low salary, which could mean they are recent in the labor force (low salary could imply lower age), so, it would be interesting for the marketing department to find a way of capturing more customers from (1,2), so that they end up spending more in health insurances, as (1,0) does.

- **(2,1)** with **(2,2)**: from the UMAP visualization we can see that this merge is not beneficial, nevertheless we will interpret them further. Their values for monthly salary and first year policy are similar. We can identify them as older people (following the previously mentioned reasoning, because their income is higher), and relatively recent customers. However, cluster (2,2) spends less when compared to (2,1), and a possible goal for the marketing department is to find a way for them to start spending more.

- **(1,1)**: this group refers to the most important clients in our database; they are recent customers that already have insurances for all categories (and spent a decent amount on the premiums). So, we aim to keep them.

- **(1,1)** with **(2,1)**: these two clusters are very similar; however, (1,1) refers to the best clients and have lower salaries than (2,1). Our goal would be to increase the value in the premiums for cluster (2,1), so we will not join them for now.

- **(0,1)**: refers to the most trustworthy customers (older people that spend lot on all insurances), but they are a small percentage of our total clients so we should group them. In addition, since they are customers that already trust the firm, we can try to attract them to higher premiums. With that being said, we thought we might group them with cluster (2,1), since these are also customers for which we have the same objectives.

- **(2,2)** and **(1,2)**: recent clients that do not contribute with any significant premiums, because they might not be familiarized with our company, so we will group them.

- **(0,2)**: oldest customers (have been clients for a long time) but never spent much on insurance, so we decided not to focus much on them.

Finally, we figured it would be beneficial to group clusters (0,1) and (2,1) – previously grouped -, because both have low cardinality. To balance our clusters, we decided to group the previous cluster with cluster (0,0), which refers to older clients that have preference over the health insurance, and we could try to allure them to other insurances.

Fig. 19 refers to the visualization of the 5 clusters we defined manually, and table 8 refers to the centroids for each feature, of our final solution.

Note: the variable 'CustMonVal' is not relevant for our clustering solutions and, as such, we decided to exclude it from the final solution. We also checked if it was worth adding 'ClaimsRate' – which is highly correlated with 'CustMonVal' – to our model, but it ended up also not being relevant to the result.

After obtaining our results, we decided to incorporate the previously removed outliers to our clusters. To do so, we predicted the class label of each observation using the K-Nearest Neighbors classifier, with K set to 5. The solution obtained from this process can be seen in fig. 20. We can clearly see two 'groups' of outliers, being that one of them is significantly larger than the other. However, neither of them is closer to our clusters, so it is not useful to add them to the solution. We can even add that the majority of the outliers is related to cluster 0, which refers to the least valuable clients.

Regarding the importance of the categorical variables, to our clustering solution, we concluded that the only features that would be worth analyzing would be PhD and Children. However, their values were very low, when compared to the variables used in the clustering solution. We also explored clustering using categorical variables, since K-Means doesn't allow for features of that type. We tried K-Modes because it works in a similar way K-Means does but instead of calculating the mean value, it calculates the modes, and instead of assessing the distance between the observations, it assesses dissimilarities (the lower the value, the more similar they are). K-Prototypes was used due to its advantaged in working with different data types. It performs in a similar way to K-Means with numerical features, but also measures the distance between categorical ones by using the number of matching categories. The solutions obtained from these methods were not good nor helpful for our analysis, so we decided not to follow through with them.

4.4. Cluster Profiling

For the cluster profiling we considered the boxplots (fig. 22), in addition to the mean (fig. 21) of each cluster and all the information obtained previously. Our final clusters are defined in the following way:

- Cluster 0: refers to the company's oldest customers, that have relatively high salaries, but do not spend much on insurances; since these clients have not seemed very interested in the company's offers, over the years, our proposal is to offer them a campaign in which older customers have a discount on the insurances offered.

- Cluster 1: most recent customers, that do not have high salaries but spend a lot on insurances; the proposal is for the company to keep the same offer for these clients, since they are very reliable and trust the firm.

- Cluster 2: relatively recent clients, with average salaries, that spend the most on health insurance; our proposal is to offer them a bundle promotion, in which one of the services offered is the health insurance and the other one of the remaining insurances (to attract them to spend more on other premiums).

- Cluster 3: refers to the customers with highest average salary, and to the most trustworthy clients; we should try to keep their trust, by using marketing campaigns to try to attract them into upgrading their insurances.

- Cluster 4: relatively new customers, with average earnings, that do not spend much on the insurances (which could be related to the fact they do not trust the company yet); these clients have high margin of growth, so we want to focus on keeping their trust. We propose a marketing campaign in which the firm develops advertisement, where people (similar individuals to the ones from this cluster) present their testimony regarding the help received from the insurance company. The goal is that these clients understand the need to improve their insurances (since they already have the salary to do so)

5. Conclusion

In order to help the marketing department of the Insurance Company, we analyzed the historical data of the firm and with information regarding 10 296 clients, we were able to group them in only 5 clusters. We did so, by taking into consideration their personal information and the insurances they hold. To achieve this solution, we needed to treat and clean our initial data, to have the best result possible. We figured it would be better to divide our dataset into two different segments, and we ended up having a dataset related to the client's information and another related to the insurances. By doing this, we were able to analyze each segment individually. Regarding the actual clustering process for each segmented, several methods were explored, and we chose the K-Means algorithm for both segments. Besides being the algorithm that led to a better performance (according to R-squared), it also optimizes for compact clusters, and in a business context, that is what we aim for. Afterwards, we merged the labels from the individual clustering solutions – from the 2 segments -, to obtain our final solution. Merging these cluster manually, while taking into consideration the dendrogram provided by the HC merging method, ended up being the best approach. Finally, a customer profiling was done to understand the customers in each cluster and to identify the best marketing approaches for each one.

6. References

missingno library for missing values - https://github.com/ResidentMario/missingno

LOF and DBSCAN - https://nandhini-aitec.medium.com/day-26-anomaly-detection-implementation-of-dbscan-lof-cof-in-python-84102ce84bf2

R2 - https://towardsdatascience.com/multiple-linear-regression-beginners-guide-5b602d716aa3

Gaussian Mixture - https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

K-prototypes - https://towardsdatascience.com/the-k-prototype-as-clustering-algorithm-for-mixed-data-type-categorical-and-numerical-fe7c50538ebb

K-Modes - https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/

7. Appendix

Table 1

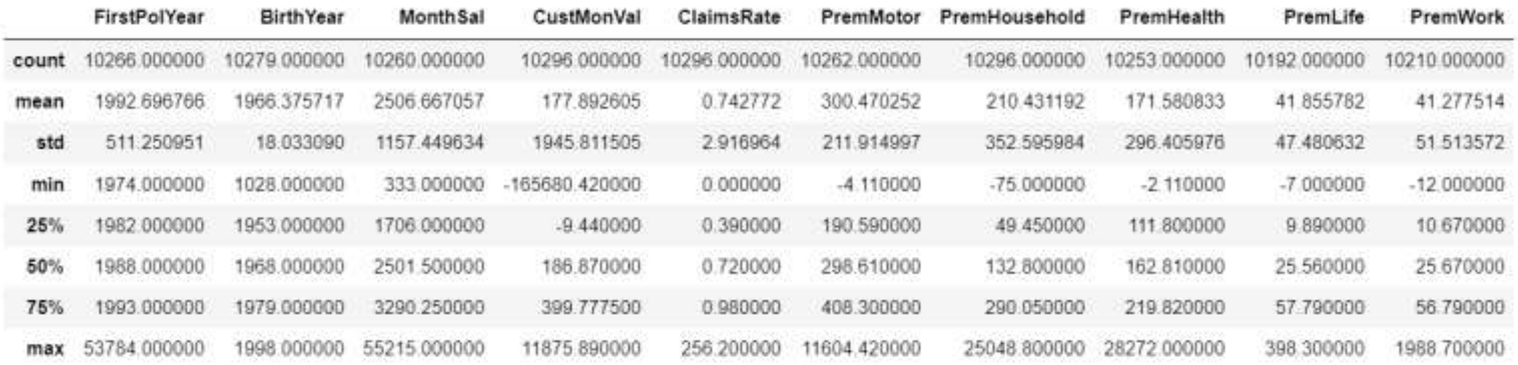
First Policy (FirstPolYear)	Year of the customer's first policy (May be considered as the first year as a customer)
Birthday (BirthYear)	Customer's Birthday Year (The current year of the database is 2016)
Education (EducDeg)	Academic Degree
Salary (MonthSal)	Gross monthly salary (€)
Area (GeoLivArea)	Living area (No further information provided about the meaning of the area codes)
Children (Children)	Binary variable (Y=1)
CMV (CustMonVal)	Customer Monetary Value Lifetime value = (annual profit from the customer) X (number of years that they are a customer) - (acquisition cost)
Claims (ClaimsRate)	Claims Rate (Amount paid by the insurance company (€)/ Premiums (€) Note: in the last 2 years)
Motor (PremMotor)	Premiums (€) in LOB: Motor[1]
Household (PremHousehold)	Premiums (€) in LOB: Household[1]
Health (PremHealth)	Premiums (€) in LOB: Health [1]
Life (PremLife)	Premiums (€) in LOB: Life[1]
Work Compensation (PremWork)	Premiums (€) in LOB: Work Compensations[1]

[1] Annual Premiums (2016); Negative premiums may manifest reversals occurred in the current year, paid in previous one(s)

Table 2

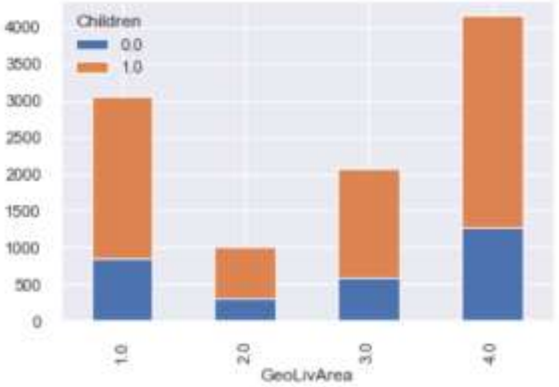
Categorical Variables	**Ordinal**	EducDeg
		GeoLivArea
	Binary	Children
Numerical Variables	**Discrete**	FirstPolYear
		BirthYear
	Continuous	MonthSal
		CustMonVal
		ClaimsRate
		PremMotor
		PremHousehold
		PremHealth
		PremLife
		PremWork

Table 3



	FirstPolYear	BirthYear	MonthSal	CustMonVal	ClaimsRate	PremMotor	PremHousehold	PremHealth	PremLife	PremWork
count	10266.000000	10279.000000	10260.000000	10296.000000	10296.000000	10262.000000	10296.000000	10253.000000	10192.000000	10210.000000
mean	1992.696766	1966.375717	2506.667057	177.892605	0.742772	300.470252	210.431192	171.580833	41.855782	41.277514
std	511.250951	18.033090	1157.449634	1945.811505	2.916964	211.914997	352.595984	296.405976	47.480632	51.513572
min	1974.000000	1028.000000	333.000000	-165680.420000	0.000000	-4.110000	-75.000000	-2.110000	-7.000000	-12.000000
25%	1982.000000	1953.000000	1706.000000	-9.440000	0.390000	190.590000	49.450000	111.800000	9.890000	10.670000
50%	1988.000000	1968.000000	2501.500000	186.870000	0.720000	298.610000	132.800000	162.810000	25.560000	25.670000
75%	1993.000000	1979.000000	3290.250000	399.777500	0.980000	408.300000	290.050000	219.820000	57.790000	56.790000
max	53784.000000	1998.000000	55215.000000	11875.890000	256.200000	11604.420000	25048.800000	28272.000000	398.300000	1988.700000

Graph 1



Graph 2

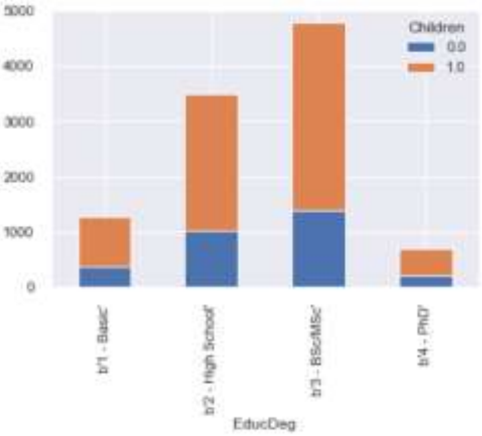


Fig. 1

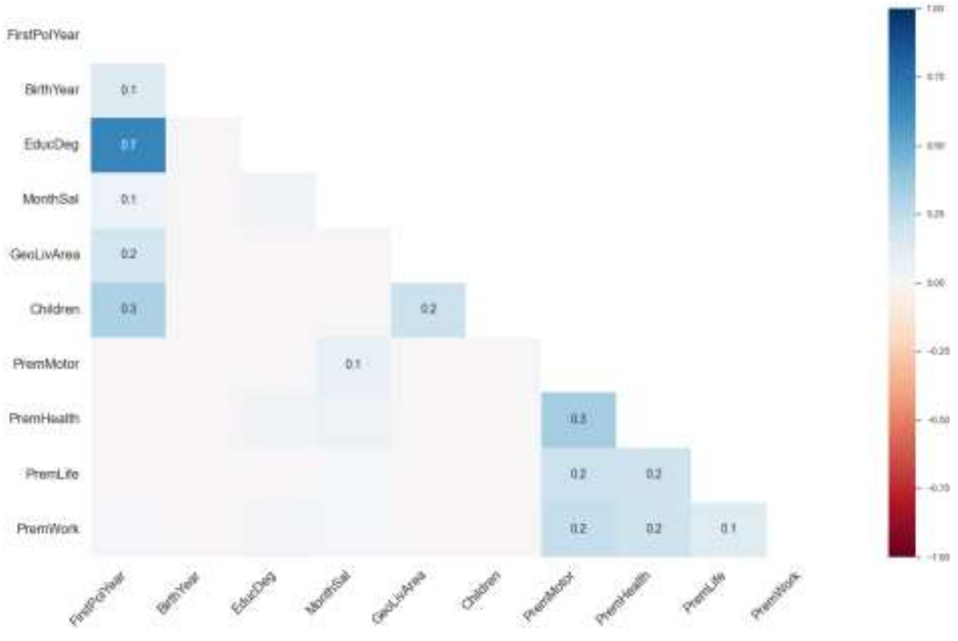
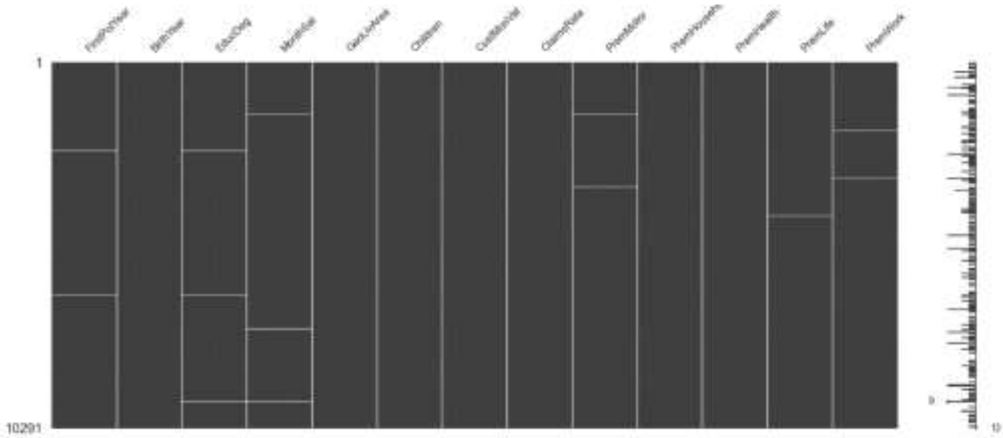


Fig. 2



Graph 3

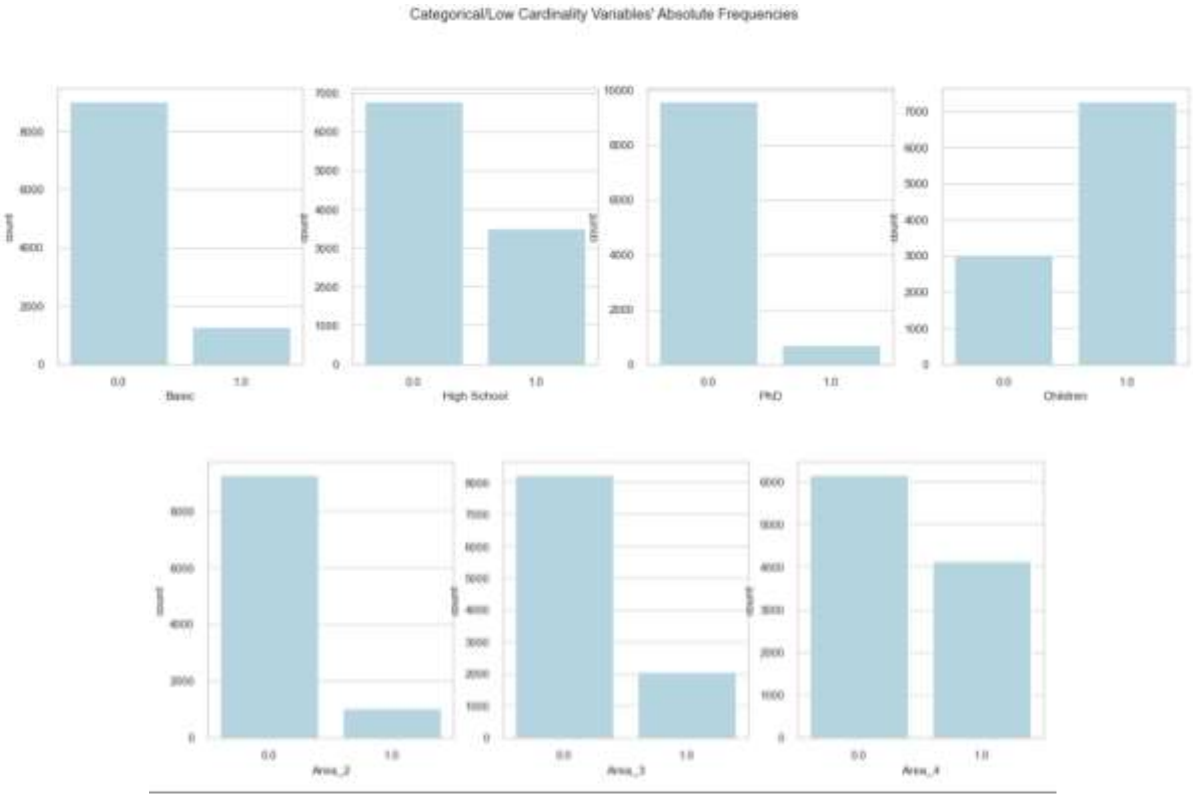
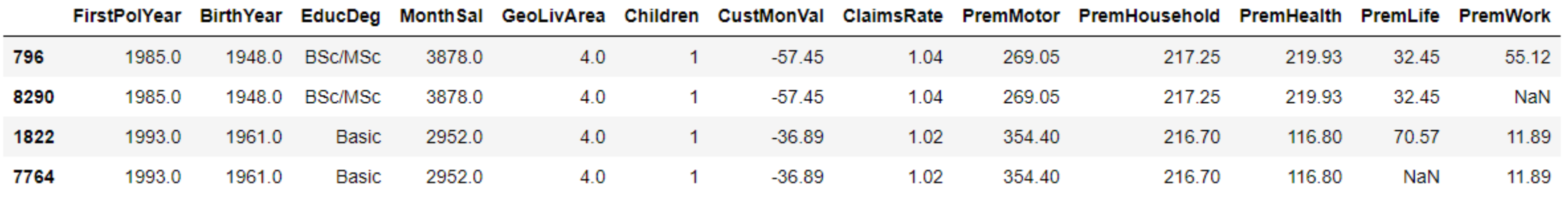


Fig. 3



	FirstPolYear	BirthYear	EducDeg	MonthSal	GeoLivArea	Children	CustMonVal	ClaimsRate	PremMotor	PremHousehold	PremHealth	PremLife	PremWork
796	1985.0	1948.0	BSc/MSc	3878.0	4.0	1	-57.45	1.04	269.05	217.25	219.93	32.45	55.12
8290	1985.0	1948.0	BSc/MSc	3878.0	4.0	1	-57.45	1.04	269.05	217.25	219.93	32.45	NaN
1822	1993.0	1961.0	Basic	2952.0	4.0	1	-36.89	1.02	354.40	216.70	116.80	70.57	11.89
7764	1993.0	1961.0	Basic	2952.0	4.0	1	-36.89	1.02	354.40	216.70	116.80	NaN	11.89

Fig. 4 - Correlation Matrix for 'clients_features'

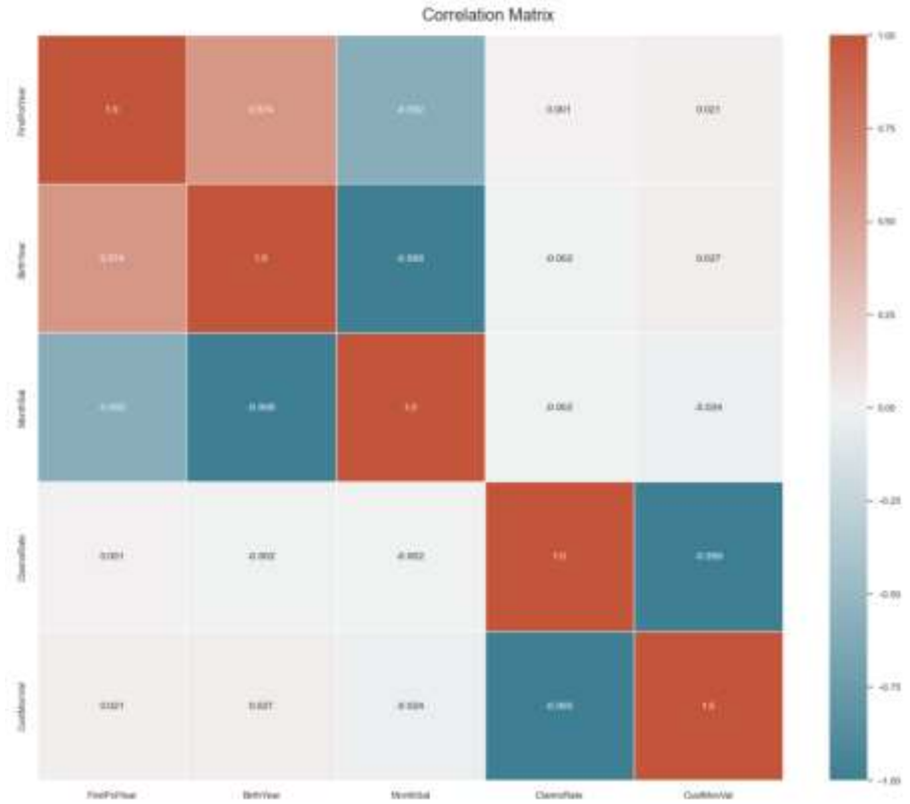


Fig. 5 - Correlation Matrix for 'prem_features'

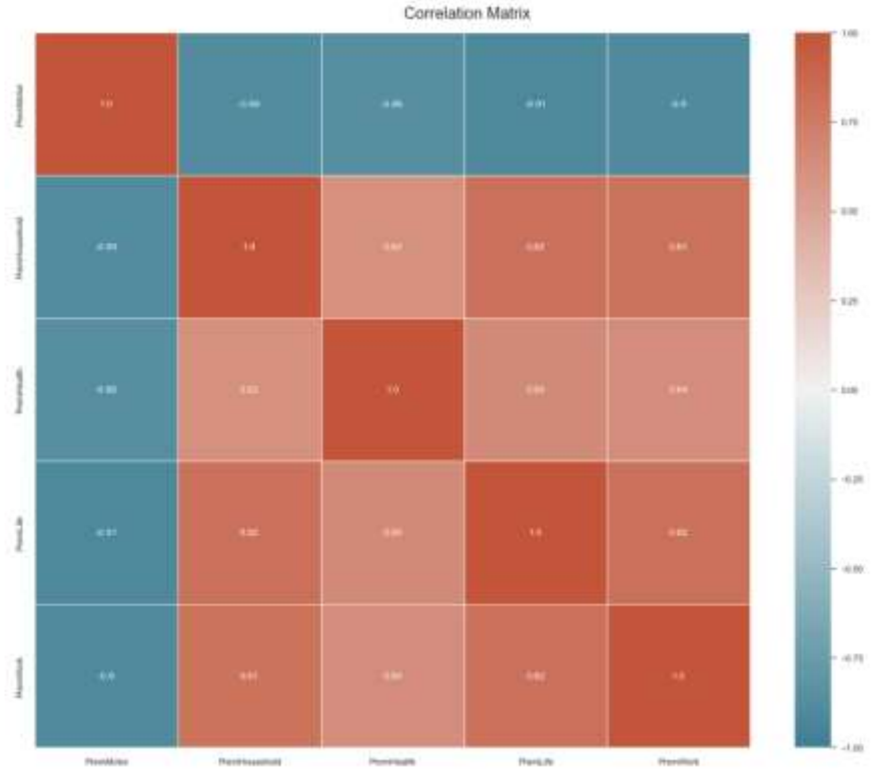


Fig. 6 - ‘clients_features’

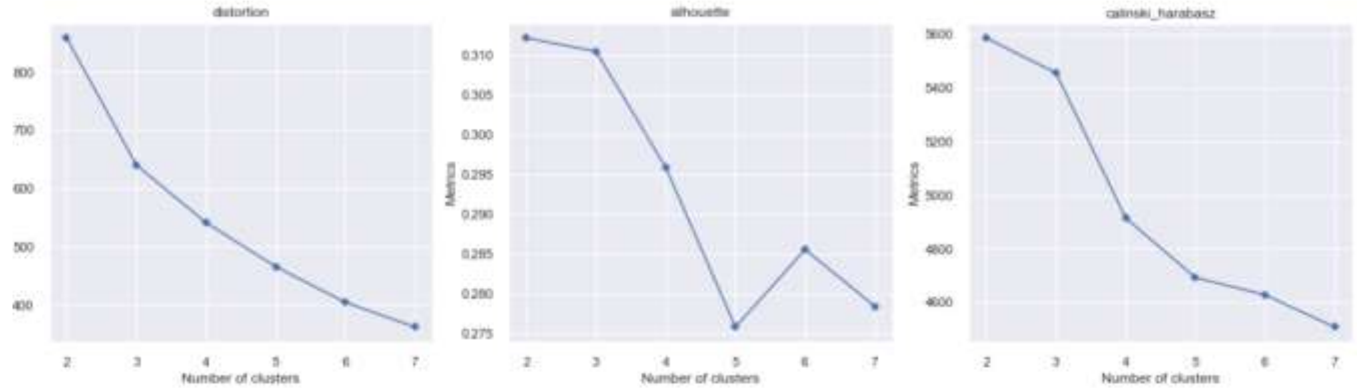


Fig. 7 - ‘prem_features’

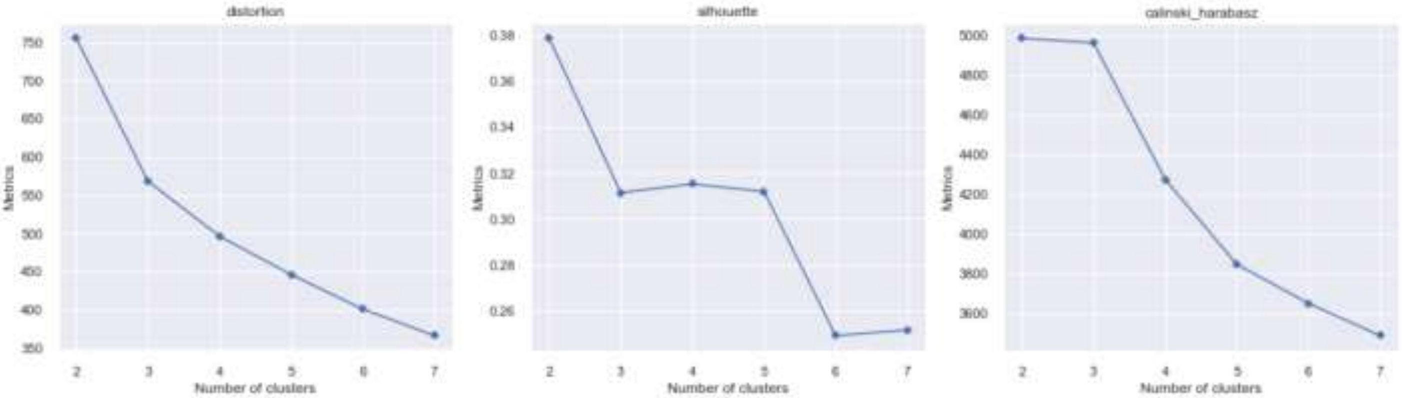


Fig. 8 - ‘clients_features’ best linkage

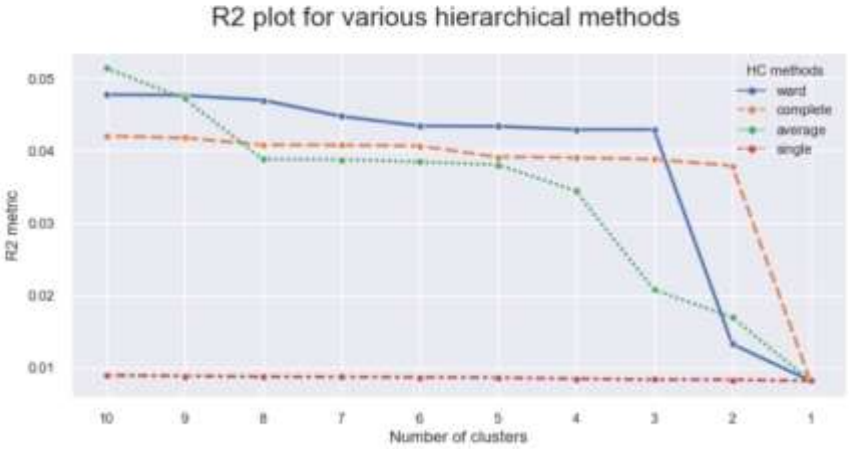


Fig. 9 - ‘prem_features’ best linkage

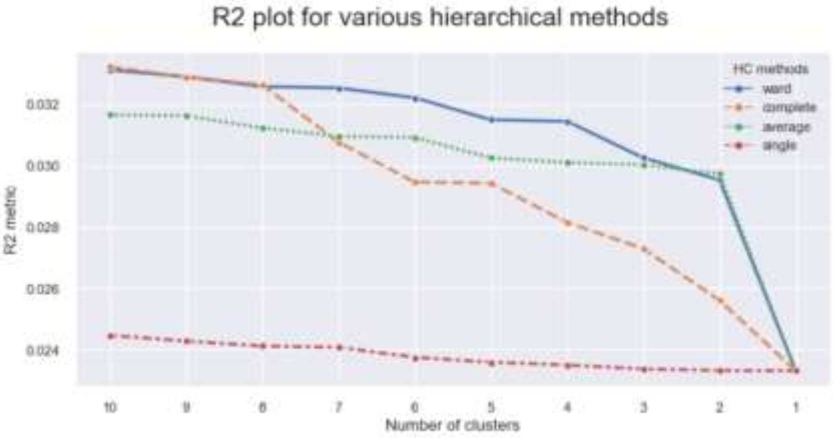


Fig. 10 - ‘clients_features’ dendrogram

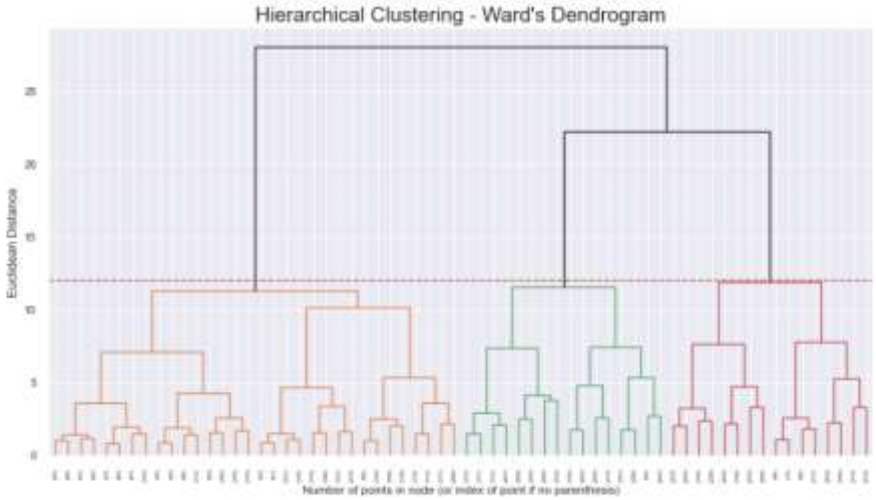


Fig. 11 ‘prem_features’ dendrogram

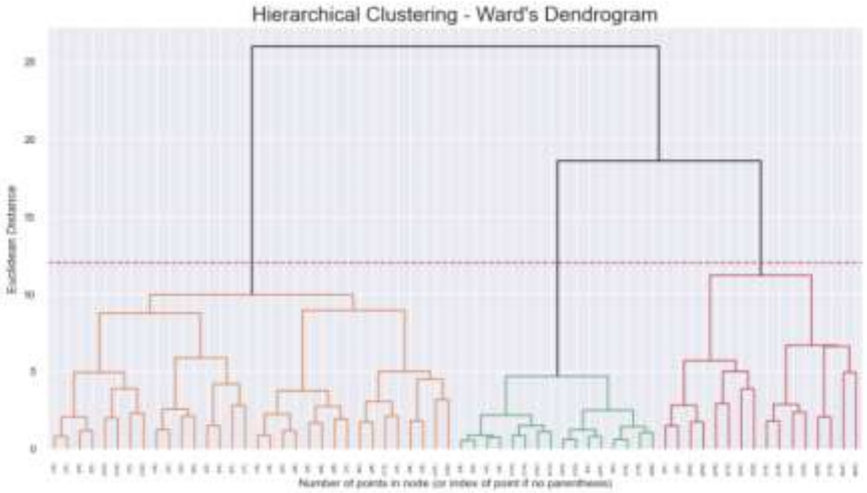


Fig. 12 - 'clients_features' AIC and BIC values for Gaussian Mixture

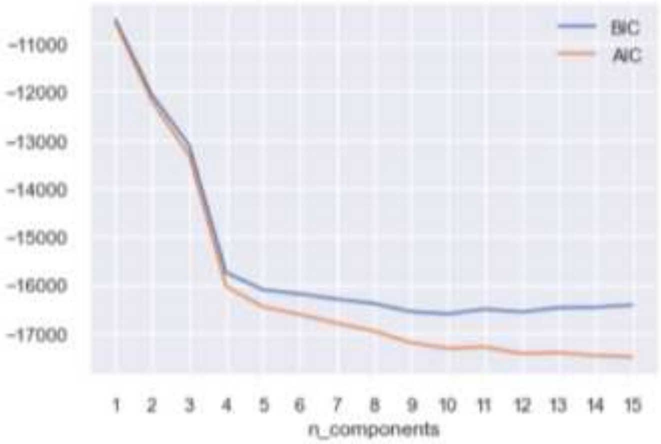


Fig. 13 - 'prem_features' AIC and BIC values for Gaussian Mixture

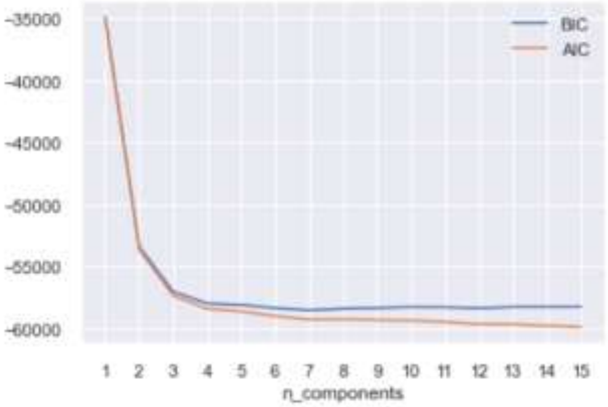


Fig. 14 - 'clients_features' UMAP visualization

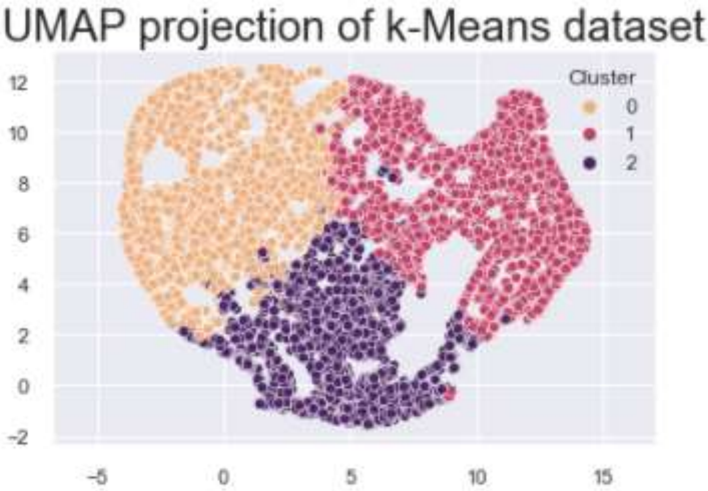


Fig. 15 - ‘prem_features’ UMAP visualization

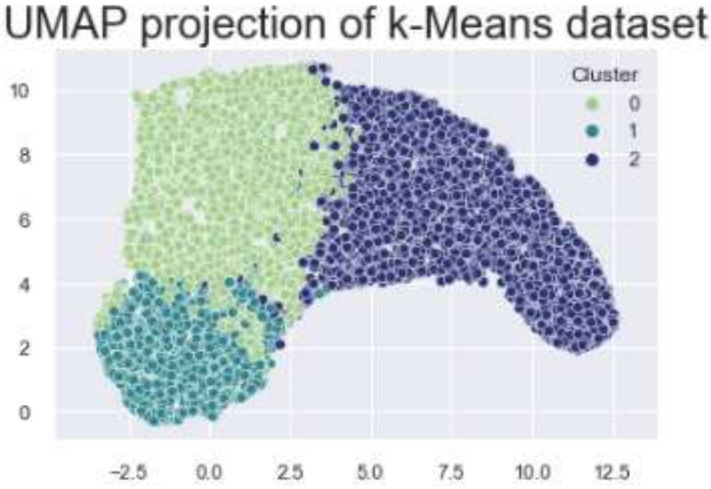
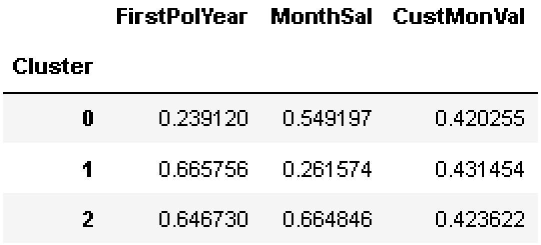
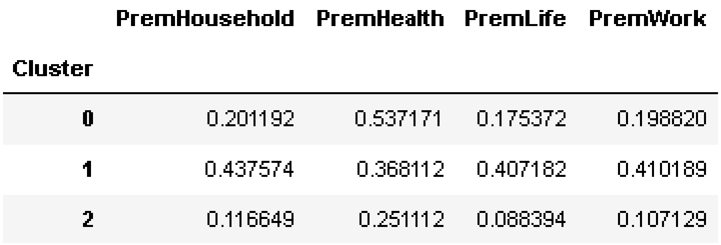


Table 4 - ‘clients_features’ – centroids



	FirstPolYear	MonthSal	CustMonVal
Cluster			
0	0.239120	0.549197	0.420255
1	0.665756	0.261574	0.431454
2	0.646730	0.664846	0.423622

Table 5 - ‘prem_features’ – centroids



	PremHousehold	PremHealth	PremLife	PremWork
Cluster				
0	0.201192	0.537171	0.175372	0.198820
1	0.437574	0.368112	0.407182	0.410189
2	0.116649	0.251112	0.088394	0.107129

Fig. 16 - Hierarchical Clustering merged solution – UMAP Visualization

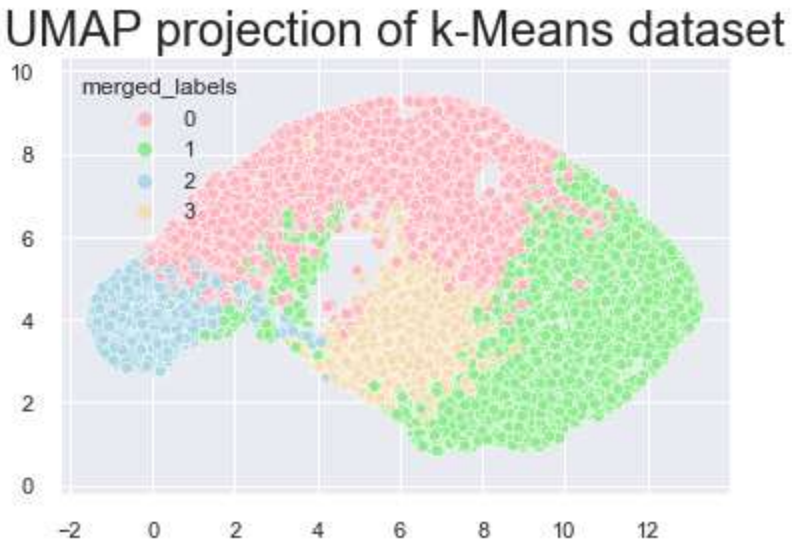
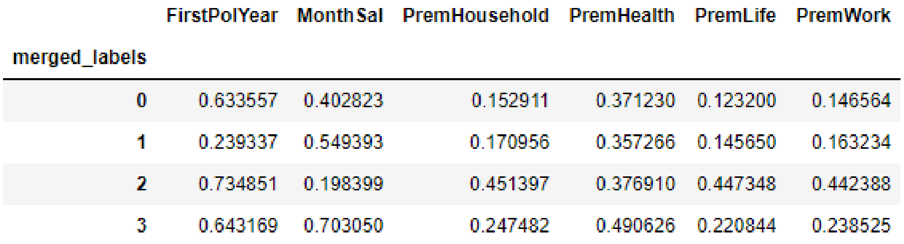


Table 6 - Hierarchical Clustering merged solution – centroids



merged_labels	FirstPolYear	MonthSal	PremHousehold	PremHealth	PremLife	PremWork
0	0.633557	0.402823	0.152911	0.371230	0.123200	0.146564
1	0.239337	0.549393	0.170956	0.357266	0.145650	0.163234
2	0.734851	0.198399	0.451397	0.376910	0.447348	0.442388
3	0.643169	0.703050	0.247482	0.490626	0.220844	0.238525

Fig. 17 - Dendrogram from merged solution from HC

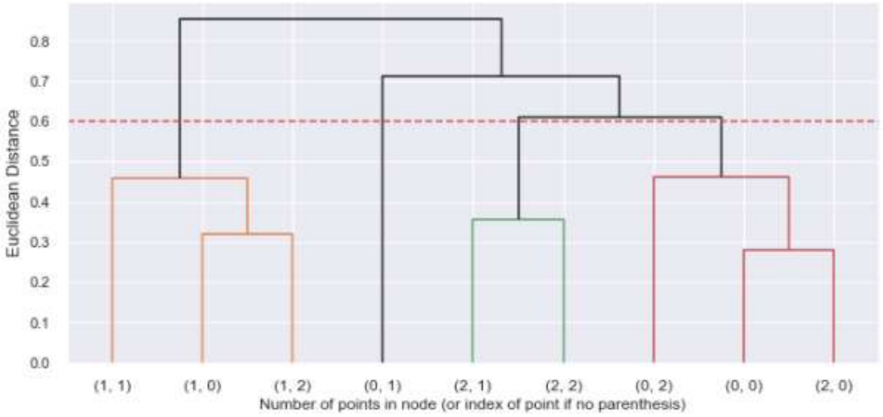


Fig. 18 - Manual merged solution – UMAP Visualization (1st)

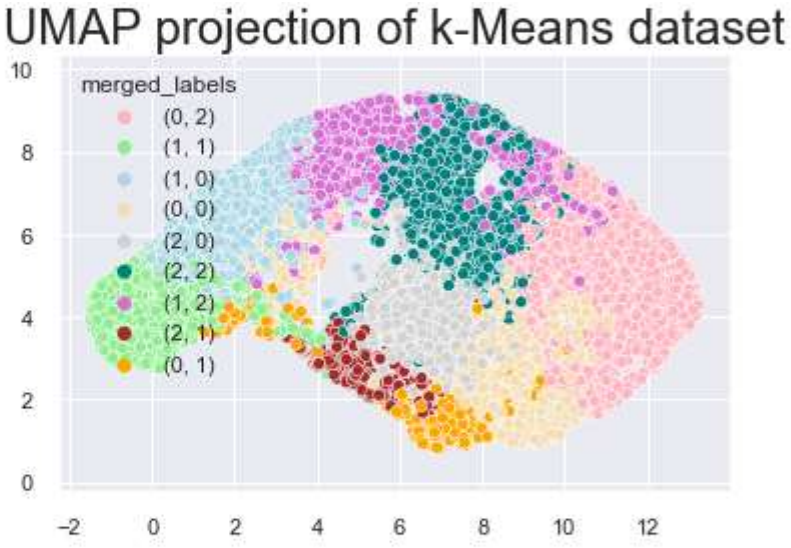
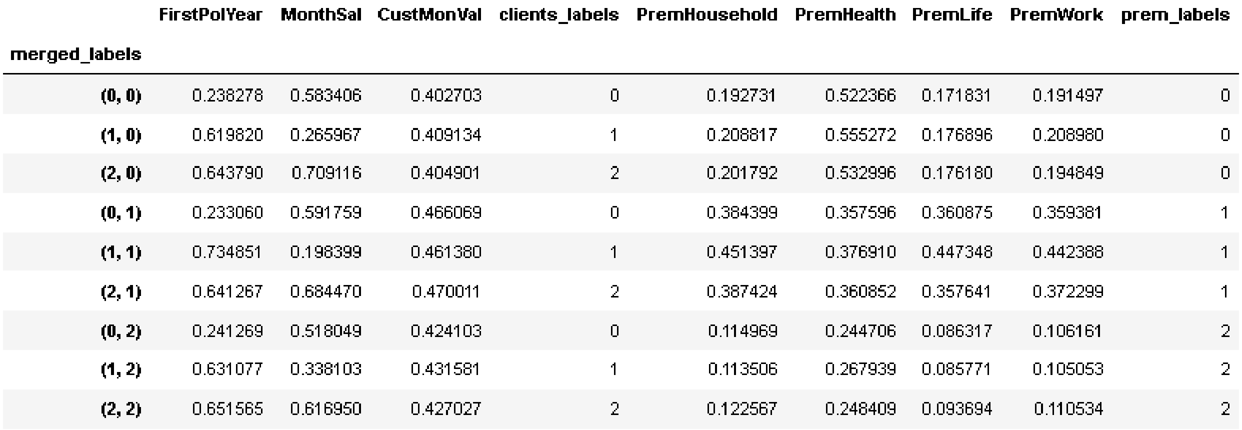


Table 7 - Manual merged solution – centroids (1st)



merged_labels	FirstPolYear	MonthSal	CustMonVal	clients_labels	PremHousehold	PremHealth	PremLife	PremWork	prem_labels
(0, 0)	0.238278	0.583406	0.402703	0	0.192731	0.522366	0.171831	0.191497	0
(1, 0)	0.619820	0.265967	0.409134	1	0.208817	0.555272	0.176896	0.208980	0
(2, 0)	0.643790	0.709116	0.404901	2	0.201792	0.532996	0.176180	0.194849	0
(0, 1)	0.233060	0.591759	0.466069	0	0.384399	0.357596	0.360875	0.359381	1
(1, 1)	0.734851	0.198399	0.461380	1	0.451397	0.376910	0.447348	0.442388	1
(2, 1)	0.641267	0.684470	0.470011	2	0.387424	0.360852	0.357641	0.372299	1
(0, 2)	0.241269	0.518049	0.424103	0	0.114969	0.244706	0.086317	0.106161	2
(1, 2)	0.631077	0.338103	0.431581	1	0.113506	0.267939	0.085771	0.105053	2
(2, 2)	0.651565	0.616950	0.427027	2	0.122567	0.248409	0.093694	0.110534	2

Fig. 19 - Manual merged solution – UMAP Visualization (2nd) – Final Solution

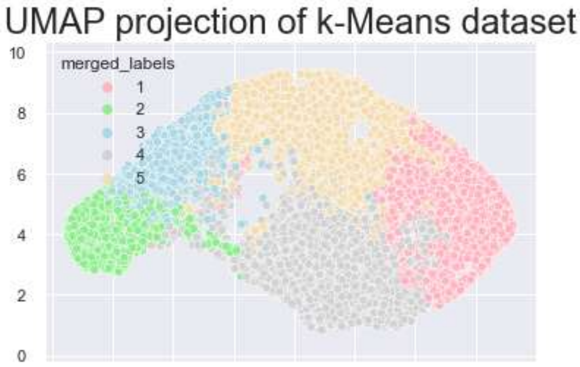
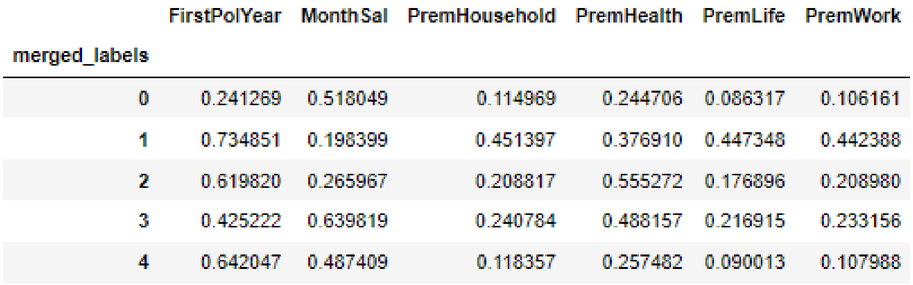


Table 8 - Manual merged solution – centroids – Final Solution



merged_labels	FirstPolYear	MonthSal	PremHousehold	PremHealth	PremLife	PremWork
0	0.241269	0.518049	0.114969	0.244706	0.086317	0.106161
1	0.734851	0.198399	0.451397	0.376910	0.447348	0.442388
2	0.619820	0.265967	0.208817	0.555272	0.176896	0.208980
3	0.425222	0.639819	0.240784	0.488157	0.216915	0.233156
4	0.642047	0.487409	0.118357	0.257482	0.090013	0.107988

Fig. 20 - Final Solution Visualization – with Outliers

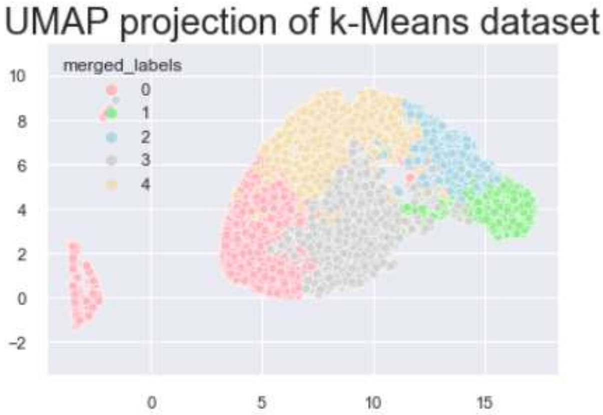


Fig. 21 - Clusters Visualization of Final Solution

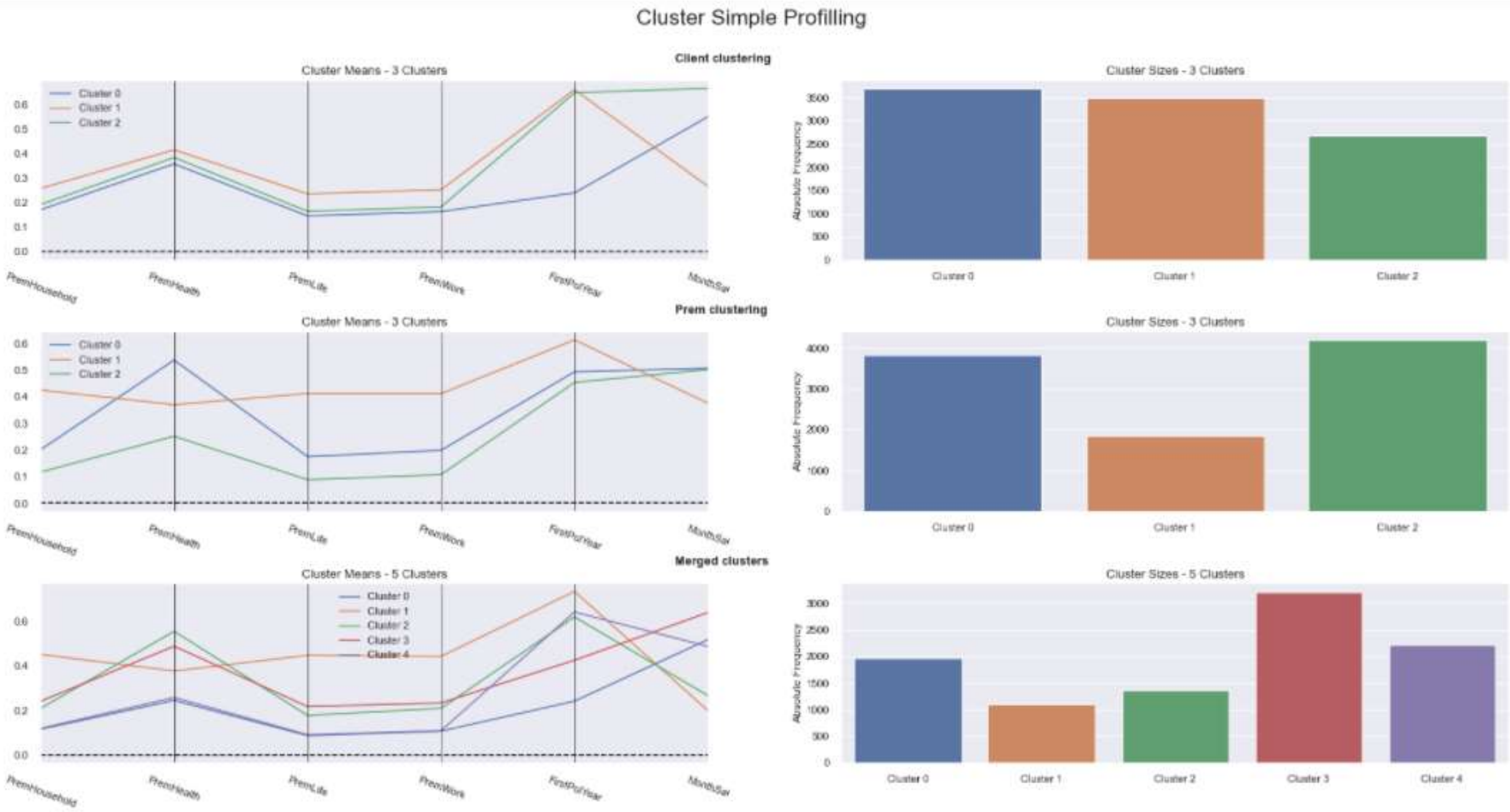
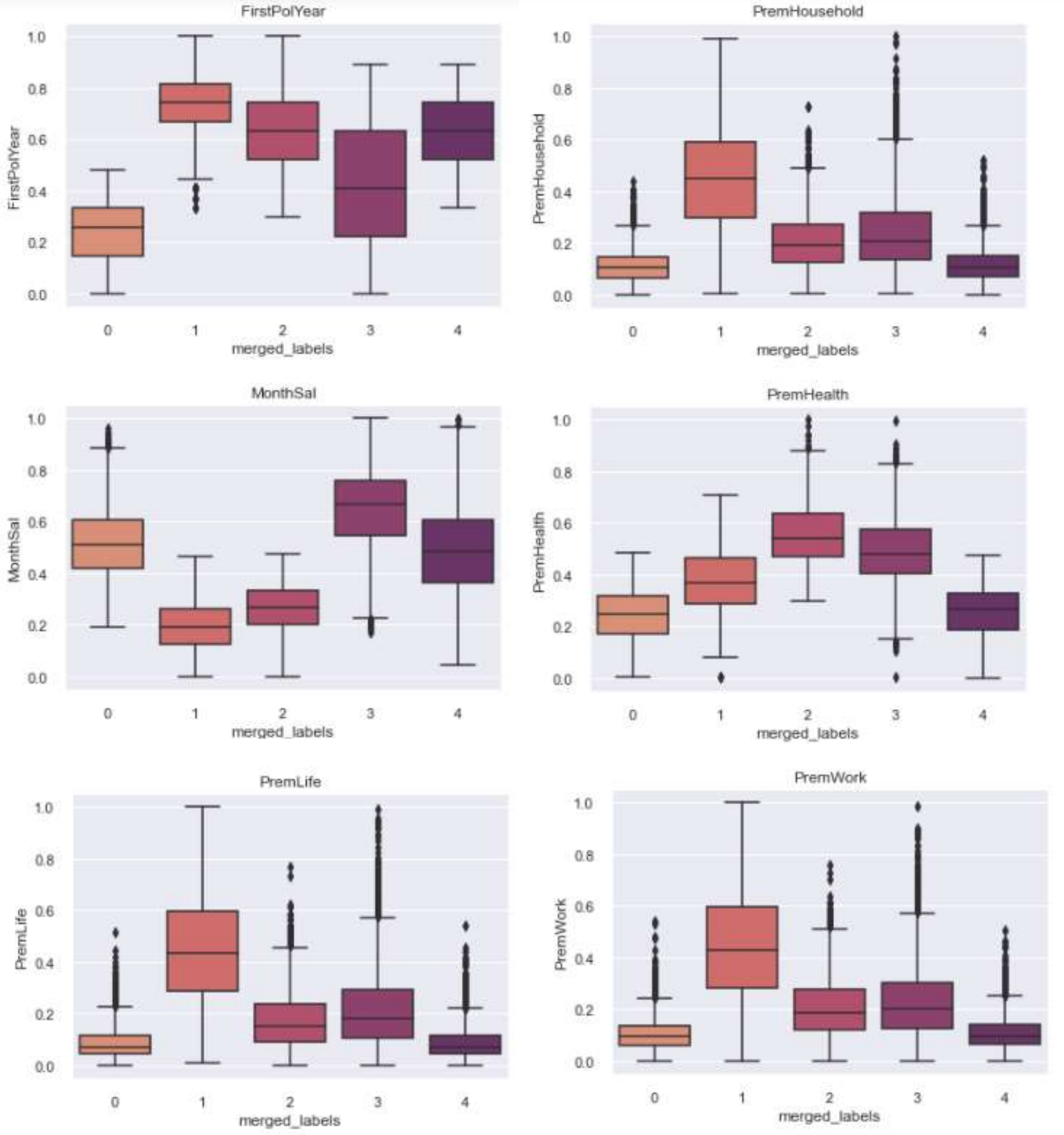


Fig. 22 - BoxPlots of the Final Visualization


FirstPolYear
PremHousehold
MonthSal
PremHealth
PremLife
PremWork
FirstPolYear
PremHousehold
MonthSal
PremHealth
PremLife
PremWork
merged_labels