Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

A2Z Insurance – Customer Segmentation

Group Y

Lukas Stark, number: 20220626

Felix Gayer, number: 20220320

David Halder, number: 20220632

January 2023

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

INDEX

LIST OF FIGURES & TABLES

1. Introduction

Market segmentation is a crucial tool for businesses because it allows them to identify and target specific groups of customers. By understanding the unique characteristics and needs of different customer segments, businesses can tailor their products, marketing, and sales efforts to better serve those customers. This can lead to increased customer satisfaction, loyalty, and ultimately, sales.

The goal of the data mining project is to segment A2Z's customer database to better understand their customers, identify potential cross-selling opportunities and provide general business and marketing suggestions.

The customer segmentation process for A2Z Insurance incorporated the CRISP-DM (Cross-Industry Standard Process for Data Mining) model as a framework. The standard CRISP-DM model was adapted to align with the specific requirements and objectives of the A2Z Insurance project.

1. In the **Business Understanding phase**, the goal and motivation of the project are established, as well as an overview of the data mining process is provided.

2. In the **Exploratory Data Analysis & Data Pre-Processing phase,** the dataset is introduced and general information about it is provided. Data cleaning and feature engineering were performed to handle missing values, inconsistencies, outliers, duplicates, and encode categorical values. Dimensionality reduction and feature selection were used to identify the relevant variables for customer segmentation. The features were then split into three different sets: one set including all features, one set including only the value features, and one set including only the demographic features.

3. In the **Modelling phase**, the methodology and different algorithm categories are introduced, including centroid-based clustering, density-based clustering, distribution-based clustering, and neural network methods. The feature segmentation approach was determined based on the results of the previous phase.

4. In the **Evaluation phase**, the performance of different approaches is evaluated, and the clusters are interpreted and explained. The process is reflected upon, and potential improvements are identified. First, the different results of the Perspectives *Value* and *Demography* were evaluated. The best results were merged. Second, the merged result and the Perspective *All-Features* is appraised. At the end of this step, the best working algorithm and the resulting clusters regarding our data was assessed.

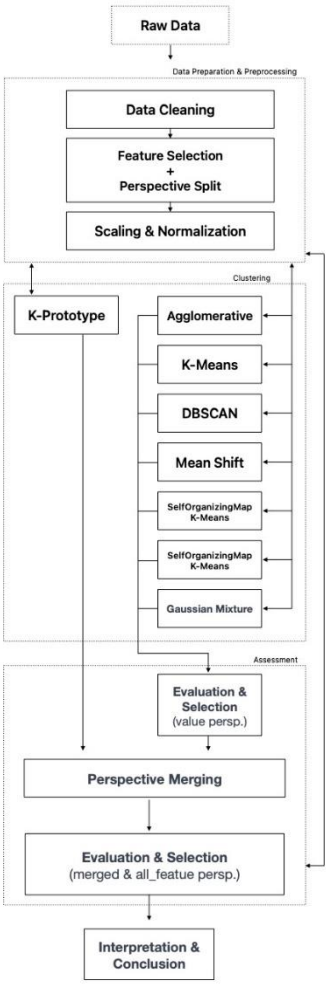


Figure 1.1 – Project Process

5. In the **Conclusion phase,** business suggestions and marketing strategies and approaches are provided based on the results of the project.

2. Exploratory Data Analysis and Data Pre-Processing

2.1. Exploratory Analysis

The provided data contains 10296 observations and 13 variables. It is split up in demographic information about the customer like the birth year or the education as well as product related variables related to the individual policy values or the monetary value a customer generates. All the features except education seem to be in a numeric manner. All the attributes are of the type "float64" and only the column "EducDeg" is of type object. Furthermore, "GeoLivArea" and "Children" should also be considered nominally scaled. Next, missing values are checked. Since "EducDeg" isn't numeric, empty strings and other common unwanted strings like "null" or "None" are considered as well. Three columns aren't showing any missing values and the feature with the most missing values is "PremLife" with 104 empty cells. To put it in a relational view no column has more than 1% missing values. Regarding duplicate rows we identified a total of 3 observations without any subset chosen. Following the descriptive statistics were analysed. "FirstPolYear" with a max value of 53784 or "CustMonVal" with a value of -165680 show extreme outliers which will be addressed later.

To get a better understanding about the distributions and outliers of each specific column bar charts and boxplots were examined[1]. Most attributes show outliers, which is conspicuous in both visualizations. The data also contains illogical values. There are 1997 observations where the "FirstPolYear" is lower than the "BirthYear"[2]. Since we can assume that it's impossible to sign an insurance policy before getting born, this needs to be addressed. Finally, the data was checked for underaged customers, and for customers who paid more in insurance, than they earned in a year[3]. Finally, a correlation heatmap was plotted. With all the outliers in mind "ClaimsRate" and "CustMonVal" still show an almost perfect correlation which could probably lead to a redundant variable even after pre-processing.

To get a deep understanding of the structure of the data the output-layers of a self-organizing map were analysed. The component planes confirm the relationships observed in the correlation matrix. Moreover, we can already examine, that there 3-5 clusters within the data[4].

2.2. Data Pre-Processing

Based on the gained insights, the data was pre-processed. Several pre-processing functions aggregated in one function called "preprocessing" were created. This ensured that multiple versions could be tested, to assess their impact on the created clusters. First, the non-logical observations which had a higher "BirthYear" than "FirstPolYear" were treated. Since the observation count for this problem is very high (1997) dropping the observations or imputing those errors with a central tendency like the mean isn't a solution. The first approach was imputing via "K-NN-imputation" which reflected in rectifying about 500 of those non-logical observations. Consequently, it was tried to switch the "BirthYear" with the "FirstPolYear" values, given that the observation was affected by

[1] See Appendix 6.1
[2] See Appendix 6.2
[3] See Appendix 6.3
[4] See Appendix 6.10 – 6.13

this problem. The underlying assumption is that there was an error or misunderstanding during data collection. This approach worked well and did not lead to any consequential problems.

Several approaches were tested regarding missing values. For the premiums it was tested to replace the empty values with the number zero, assuming the respective customer did not take out an insurance in the area without value. However, since the number of assumptions should be held to a minimum and the imputation with zero did not change the results significantly, the KNN-Imputer was used to impute the missing values. This has the advantage that it takes multiple variables into account to infer from the behaviour of similar customers to the ones with missing values. Next the missing values of the categorical variables were imputed. First imputation via mode was performed and subsequently "EducDeg" is manually imputed due to the ordinal scaling of this attribute. Finally, all three duplicate rows found, got dropped.

Subsequently outliers were handled. Considering the large number of outliers in the dataset simply using the interquartile-range method and dropping all of them will lead to a large amount of lost information. Therefore, first flooring and capping the outliers to the whiskers was applied, since that ensures that the rank of the capped observations remains very high or very low while making the values less extreme. By replotting the bar charts and box plots it can clearly be observed that they have improved but is not ideal. Particularly for the columns "PremHousehold", "PremLife" and "Premwork" a normal flooring and capping method seems too inordinate which is indicated by the large bars on the right-hand side of the bar plot[5]. More than 600 outliers got trimmed down to the upper whisker limit. To solve this problem, we added the possibility to our outlier-removal function to set cut-off-limits manually and drop the observations above/below these limits. Trade-off of this procedure is a 4% decrease in observations. While this is not ideal, since information is lost by doing so, the improvement of the results upheld the decision made. Furthermore, these observations were saved and will be added to the created clusters at the end of the process. The final distribution can be seen in the Appendix[6].

Finally, it was tried to normalize three skewed premium columns ("PremHousehold", "PremLife" & "Premwork") with scikit-learn's PowerTransform[7]. However, this did not add any value to the clustering solutions and complicated the interpretation which is why it was removed again.

Before scaling the data, the final step was creating features to add value to clustering or for interpretation. A full list and description of the calculated variable can be found in the Appendix[8].

Finally, the data was scaled with the "StandardScaler", since many implemented algorithms use distance metrics which are prone to different scales[9]

[5] See Appendix 6.5
[6] See Appendix 6.6
[7] See scikit-learn [1], 2023
[8] See Appendix 6.7
[9] See scikit-learn [2], 2023

2.3. Dimensionality Reduction & Feature Selection

After conducting a Principal Component Analysis (PCA) on our data, it was determined that using principal components does not add any value to the project. First, 4 principal components would have been needed to explain 80% of the variance in the data, which is a relatively large number. This suggests that the data may not be highly amenable to dimensionality reduction through PCA. Secondly, it was found that interpretation is significantly more complex, as principal components are linear combinations of the original features, thus an additional analysis of the loading scores would have been needed.

For the feature selection two perspectives need to be considered: the business- and the data-perspective. From the data-perspective it is important that variables used are relevant, not redundant and provide some discriminatory value. To assess that first a correlation matrix was created. “PremTotal” has a nearly perfect collinearity with “PremHousehold” and will therefore be removed. With a value of -0.94 also “CustMonVal” and “ClaimsRate” are highly correlated the same, thus “CustMonVal” will not be considered for the clustering, to reduce redundancy. Even though “MonthSal” and Age are also highly correlated they are both considered for the clustering, as they added value to the results. The attribute “GeoLivArea” did not show any discriminatory value and also did not add any value to the results, thus it will not be considered further. The other transformed variables are either very high correlated with the premium-values or are combined through the two clustering-perspectives which are introduced later.

From the business side, it is important that as much information regarding the socio-demographics and the value of the customers as well as the information what products/insurances were purchased and have proven to be valuable to the company. It was decided to not consider “GeoLivArea” in this context as it does not provide any valuable insights regarding the understanding of customer behaviour or segments and should therefore not be considered for any marketing strategies.

2.4. Clustering Perspectives

Clustering by perspective refers to the process of dividing a dataset into two or more subsets based on certain characteristics or features, and then applying a clustering algorithm to each subset independently. After applying the clustering algorithm to each subset, you could then combine the results to get a more complete understanding of the patterns and relationships in the data. We therefore created two separate perspectives. The first one contains socio-demographic information about the customer and is a mixture between numeric and categorical data which will later be used for the K-Prototypes algorithm. Secondly the values/product perspective focuses on the value of the customer to the company as a whole and within the different premium categories. A detailed overview of the perspectives and the respective features can be found in the Appendix[10].As a benchmark, the clustering algorithms were also tested taking all metrical variables into account.

[10] See Appendix 6.9

3. Modelling

3.1. Introduction/Methodology

Contemplating the business situation, it was decided that not any number of clusters are economically sensible. On the one hand it is too expensive to develop a very high number of strategies for many small clusters. On the other hand, too few big clusters might be too generic to create adequate strategies. Consequently, it was decided to find a segmentation solution anywhere between 3 and 7 Clusters.

As described earlier various pre-processing approaches were tested. To find interpretable and actionable clusters, these approaches were then tested with multiple clustering algorithms of different categories. The strategy behind that approach was to try at least one approach per category on our data to see what clustering technique is best suited. This is conducted on the different perspectives as well and on the whole dataset, filtered for metrical variables. Following the cluster methods used will be stated, shortly explained and the parameter values chosen will be elaborated. The final number of clusters chosen can be found in Appendix[11].

3.2. Algorithms

3.2.1. Partitional: Hierarchical Algorithms

The first algorithm used was an agglomerative hierarchical clustering algorithm. This algorithm calculates the distance between all the given variables, combines the closest variables, combines them to a cluster and repeats these steps until every variable is within one cluster. Since this algorithm utilizes distances, it only works for numerical variables. Thus, it was used on the value set and on the all-features set.

An important parameter for this clustering technique is the Linkage/Aggregation rule. These rules specify how the distance between the variables/clusters is measured. To find the optimal rule, they were tried, plotted, and assessed for the explained variance per number of clusters. For the distance metric, the Euclidean distance was selected.

3.2.2. Partitional: Centroid Based Algorithms

Next Partitional Cluster Algorithms were tested on the different datasets. For the value- and the complete numerical- set K-Means was chosen and for the demographic perspective K-Prototypes was used, since this algorithm can also handle categorical data.

The main disadvantage is that the K-means algorithm needs a number of clusters as input. Therefore, the algorithm was executed multiple times with a different number of k and the elbow plot and the average silhouette score was used to find the optimal number of clusters. Furthermore, the initial seeds can be a problem. To solve this problem we used the hyperparameter “k-means++”, which

[11] See Appendix 6.14

places the seeds based on a probability distribution regarding their contribution to inertia, which can speed up convergence[12].

The K-Prototype Algorithm combines the K-Means and the K-Modes algorithm. To do so it divides its dissimilarity coefficient into two parts. For the numerical part the squared Euclidean distance and for the categorical data the Hamming distance is used. Moreover, a parameter "γ" is included in the equation to control the relative weight of dissimilarity between categorical and numerical data. Based on that dissimilarity function the algorithm divides the dataset into k - clusters to minimize the value of a cost function, which calculates the sum of the within cluster distance[13].

Since you also need to give the K-Prototype Algorithm a number of clusters beforehand, we used the result of the cost-function to visualize an elbow criterion for various numbers of clusters.

3.2.3. Density Based Algorithms

To account for some of the weaknesses of the partitional clustering algorithms, like the sensitivity to the cluster-shape, initial seeds or outliers, also two density-based algorithms were tested.

DBScan is a density-based clustering algorithm, which creates clusters based on the number of objects close to point x. The algorithm has two input parameters: "ϵ", which specifies the neighborhood of point x, and "MinPts", which specifies how many points need to be within ϵ so that point x can be marked as a core object. One peculiarity of the DBscan algorithm is that it can detect multidimensional outliers. Outliers found, will be removed for the interpretation of DBScan.

To find the optimal value for ϵ we used the K-Nearest Neighbor Algorithm as proposed by Rahmah and Sukaesih Sitanggang[14]: The distances are calculated, sorted, plotted and at the point of the maximum curvature we can extract the optimal ϵ.

To find a suiting value for "MinPts" it was looped through a range from 1 to 20. It was quickly clear that DBSCAN can only find distinct clusters for small values of minimum points within a neighbourhood. That can either mean that the combination between epsilon and "MinPts" is not adjusted for the dataset or that there are several different densities within the data, which makes the detection of density-based clusters very difficult.

Mean shift is also a density-based clustering algorithm, but it utilizes centroids to find clusters by shifting a sliding window towards dense regions until it reaches convergences

The most important parameter of the algorithm is the bandwidth. It defines the size of the sliding window, hence the size and number of clusters. To estimate the bandwidth the "estimate_bandwith" function from sklearn.cluster was used[15]. As the bandwidth is calculated upon a quantile of the pairwise distances, the quantile needs to be defined before.

---

[12] See scikit-learn [3], 2023
[13] See Jia & Song, 2020, p.2
[14] See Rahmah and Sukaesih Sitanggang, 2016
[15] see scikit-learn [4], 2023

3.2.4. Distribution Based Algorithms

To exploit the advantages of distribution-based algorithms against the k means algorithm, in particular the "lack of flexibility in the cluster shape and the lack of probabilistic cluster assignment"[16] the Gaussian Mixture Model was tested on our data.

The algorithm first chooses starting guesses regarding the location and the shape of the clusters based on the number of clusters predefined. Next, two steps are repeated until the clusters are converged. First, it finds weights for each point which encode the probability of membership to each cluster. Next, based on these weights and all data points, the location shape and normalization of the clusters is updated[17]

The most important hyperparameter is the covariance type. In order not to go beyond the scope of the project they will not be explained in detail. The three covariance types ("diag", "spherical", "tied" and "full") were tested and based on the clustering results the type "full" was chosen.

To find out the appropriate number of clusters, the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) were examined. The AIC shows the relative amount of information lost by the model while representing the data generation process. BIC works in a similar way but penalizes complex models. At the optimal number of clusters, AIC and BIC are minimized.

3.2.5. Self-Organizing Maps

Finally, Self-Organizing Maps (SOM) are used on their own and together with the k-means agglomerative hierarchical clustering (Emergent SOM). In this method a very large number of SOM-Units is used, to accurately detect the structure of the underlying data.

First clustering solely with a Self-Organising Map was undertaken. This was done by first examining the average silhouette score per number of neurons. Next a SOM was built with the appropriate number of neurons, which is similar to the number of clusters retrieved.

For the K-Means Algorithm first the elbow and silhouette scores to define the optimal number of clusters, was analysed. Then we visualized the clusters obtained on a Hit-Map and finally obtained the cluster labels for the data, by obtaining the best matching unit for every observation in the original dataframe.

A similar approach was taken for hierarchical clustering. Based in the large SOM we first chose the distance metric, then the appropriate number of clusters and finally visualized a Hit Map with the number of clusters chosen.

3.3. Performance Evaluation & Cluster Interpretation/Explanation

To determine which of the clustering algorithms was the most effective, the performance of each algorithm was assessed using the R Square score and Silhouette scores. In addition, the duration of the

[16] VanderPlas, 2018, para. 9
[17] See VanderPlas, 2018

clusters was compared to the average duration of the entire dataset, and the UMAP and T-SNE plots were analysed visually. The profile of the clusters was analysed using a radar plot, and the feature importance and composition of the clusters were examined using a classification tree. The analysis has been evaluated on the entire feature set.

In the first step, the perspectives of value and demography were analysed in more depth and the best algorithm was selected based on these results. The best results are then combined.

In the second step, the combined results were compared to the perspective of all features and evaluated to determine which approach was most suitable for our business needs.

3.4. Perspectives - Value & Demography

In analysing the performance of various algorithms on the value perspective, it can be observed that the R2 scores and silhouette scores are not completely aligned. The “KMeans” and “SOM + KMeans” algorithms, both using 4 clusters, were the top performers according to R2 scores, but the “SOM” algorithm with 2 clusters performed better according to silhouette scores. However, the “SOM” algorithm did not perform as well on R2 scores. Additionally, it was noted that the two density-based algorithms did not yield satisfactory results.

To assess the performance of each algorithm, the results of all algorithms were compared. This included evaluating the mean values of the clusters in relation to the mean values of the total data, examining the distribution of the clusters, and analysing the UMAP and TSNE projections. Additionally, a classification tree was used to explain the functioning of each algorithm. The visualizations of the results for each algorithm can be found in the Appendix. While a more detailed description is not feasible within the constraints of this report, “KMeans” was determined to be the best performing algorithm for the Value perspective.

For the Demography perspective, the “KPrototype” algorithm was found to be the only one capable of handling categorical as well as numerical data. Upon evaluation, it demonstrated strong performance with an R2 score of over 0.6 and a Silhouette score of over 0.3. While “KPrototype” outperformed all other algorithms in the Value perspective in terms of both scores, it should be noted that only four features were used for clustering in the Demography perspective, compared to seven in the Value perspective. It also has to be noted that the algorithm is not very efficient in terms of computational time, therefore it might not be advisable to utilise it for very large datasets.

The two results were merged after the evaluation using hierarchical clustering.

3.5. Perspectives - Merged & All Features

In the following analysis, we compared the merged perspectives with the approach of clustering all numerical features together. The first step was to determine the best algorithm for the All-Features perspective.

Upon examining the R2 scores and silhouette scores, it was found that the scores for most algorithms behaved similarly. However, the scores for “SOM” (2 clusters), “MeanShift” (4 clusters), and “DBSCAN” (3 clusters) performed poorly. The best performance in terms of R2 was seen with “GMM” (4 clusters), while “KMeans” performed best in terms of silhouette score. After conducting a detailed

content analysis using the same methods as for the Value perspective, “KMeans” was determined to be the best algorithm for the All Features perspective.

In comparison with the merged variant, it was more difficult to make a decision between the two approaches, as both formed well-separated, distributed, and logically interpretable clusters. However, the merged version performed better in terms of the business objective and was therefore chosen. The resulting profile is described in detail below.

The analysis of the clusters reveals that they are distributed around the mean value of the overall data for all features, with no cluster oriented completely towards the mean. The features “ClaimsRate” and “CustMonVal” exhibit little variance, while the averages of “Children”, “Age”, “MontSal”, and “PremMotor” are well separable around the average of the total data. Clusters 0 and 1 exhibit similar behavior, while cluster 3 is significantly different from the other clusters in most features. In terms of distribution, clusters 1 and 3 are relatively balanced with 2786 and 2612, respectively. Cluster 0 is the largest with 3330, while cluster 3 is the smallest with 1145.When visualized using UMAP and TSNE, all clusters can be largely separated from each other, with mixing occurring at the edges, particularly between clusters 1 and 3. This is also evident in the scatter pair plot[18].

	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Age	middle age (≈ 52)	young (≈ 35)	old (≈ 70)	young (≈ 31)
Education	good	good	good	bad
Children	yes	yes	no	yes
Monthly Salary	middle (≈ 2700)	low (≈ 1600)	high (≈ 3700)	low (≈ 1200)
First Policy Year	early (≈ 1986)	early (≈ 1988)	early (≈ 1986)	not so early (≈ 1993)
Health Prem	middle/low (≈ 125)	high (≈ 195)	high (≈ 200)	middle/high (≈ 165)
Household Prem	low (≈ 130)	low (≈ 150)	low (≈ 200)	high (≈ 540)
Life Prem	low (≈ 25)	low (≈ 30)	low (≈ 40)	high (≈ 110)
Work Prem	low (≈ 25)	low (≈30)	low (≈40)	high (≈ 105)
Motor Prem	high (≈ 390)	middle (≈ 300)	middle (≈ 265)	low (≈ 100)
Total Prem	low (≈ 690)	low (≈ 710)	middle/low (≈ 750)	high (≈ 1020)
Claims Rate	middle (≈ 0,65)	middle (≈ 0,69)	middle (≈ 0,7)	middle (≈ 0,69)
Customer Monetary Value	low (≈ 220)	low (≈195)	low (≈200)	middle (≈ 300)

Table 5.1: Abstract Final Clustering Solution

4. Conclusion and segmentation-based strategy

4.1. Acquiring new customers

The results of the cluster analysis reveal that there are several opportunities for new customer acquisition through the creation of bundled insurance packages. In particular, cluster 3 demonstrates that household, life, health, and work insurance policies are often taken out together, and can potentially be targeted to a young audience with low educational qualifications, children, and low incomes through the use of social media as a marketing channel.

[18] See Appendix 6.20-6.23

Additionally, cluster 0 exhibits a higher total premium and customer monetary value than the other clusters, indicating that bundle solutions could potentially save on marketing and acquisition costs by bundling multiple insurance policies, and potentially increase customer monetary value even further.

The analysis of clusters 1 and 2 suggests that health and motor insurance are of particular interest to these target groups. Bundle solutions could also be pursued for these groups, targeting young people with good educations, children, and low incomes with gentrified appeals, and older individuals with good educations, no children, and high incomes through channels such as newspapers and estate agents. However, it should be noted that both of these groups have relatively low customer monetary values and should not be heavily advertised with large budgets.

Finally, the analysis of cluster 0 shows that there is a group of middle-aged, well-educated individuals with children and average wages who are primarily interested in motor insurance. One hypothesis regarding this particular behaviour could be, that the family car got exchanged for a more luxurious vehicle, since the children are slowly becoming independent, thus the motor-insurance rate went up. Consequently, it would make sense to partner with local car-dealerships of medium to high class brands, to place our product right at the source.

4.2. Up-selling/Cross-selling

For the existing clients in cluster 0, it appears that Health, Household, Life, and Work insurance may be viable opportunities for up- or cross-selling due to the relatively low level of investment in these areas to date. In cluster 1, Motor insurance presents a potential chance for upselling, as there is interest in this type of insurance despite it currently performing in the midfield. Additionally, Household, Life, and Work insurance may also offer chances for cross- or up-selling due to the low level of investment in these areas. For customers in cluster 2, the same types of insurance as those in cluster 1 may be viable opportunities due to the similarity in their insurance profiles. In cluster 3, only Motor insurance appears to have potential for cross-selling, possibly due to the low income and young age of this group, who may not own a car. This group may be the least promising for upselling efforts.

4.3. Retention

Cluster 0 represents an older version of cluster 1. Both clusters are well educated and have children, but cluster 0 is middle-aged while cluster 1 is younger. The income level of cluster 0 is middle range, while that of cluster 1 is low. However, the data suggests that there is a correlation between age and income, meaning that as cluster 1 ages, it is likely to also have a middle income. It is worth noting that the older generation in cluster 0 places less value on health insurance compared to the younger generation in cluster 1. This finding suggests that it may be necessary for the insurance company to take proactive measures to retain customers in cluster 0 as they age. Additionally, the data indicates that there is strong potential for upselling motor insurance to the older generation in cluster 0, as this group exhibits a stronger preference for such products compared to the younger generation in cluster 1.

5. References

Jia, Ziqi; Song, Ling (2020): Weighted k-Prototypes Clustering Algorithm Based on the Hybrid Dissimilarity Coefficient. In *Mathematical Problems in Engineering* 2020, pp. 1–13. DOI: 10.1155/2020/5143797.

Rahmah, Nadia; Sitanggang, Imas Sukaesih (2016): Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra. In *IOP Conf. Ser.: Earth Environ. Sci.* 31 (1), p. 12012. DOI: 10.1088/1755-1315/31/1/012012.

scikit-learn [4] (2023): sklearn.cluster.estimate_bandwidth. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.cluster.estimate_bandwidth.html, updated on 06/01/2023., checked on 06/01/2023.

scikit-learn [3] (2023): sklearn.cluster.KMeans. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html, updated on 1/5/2023, updated on 06/01/2023., checked on 06/01/2023.

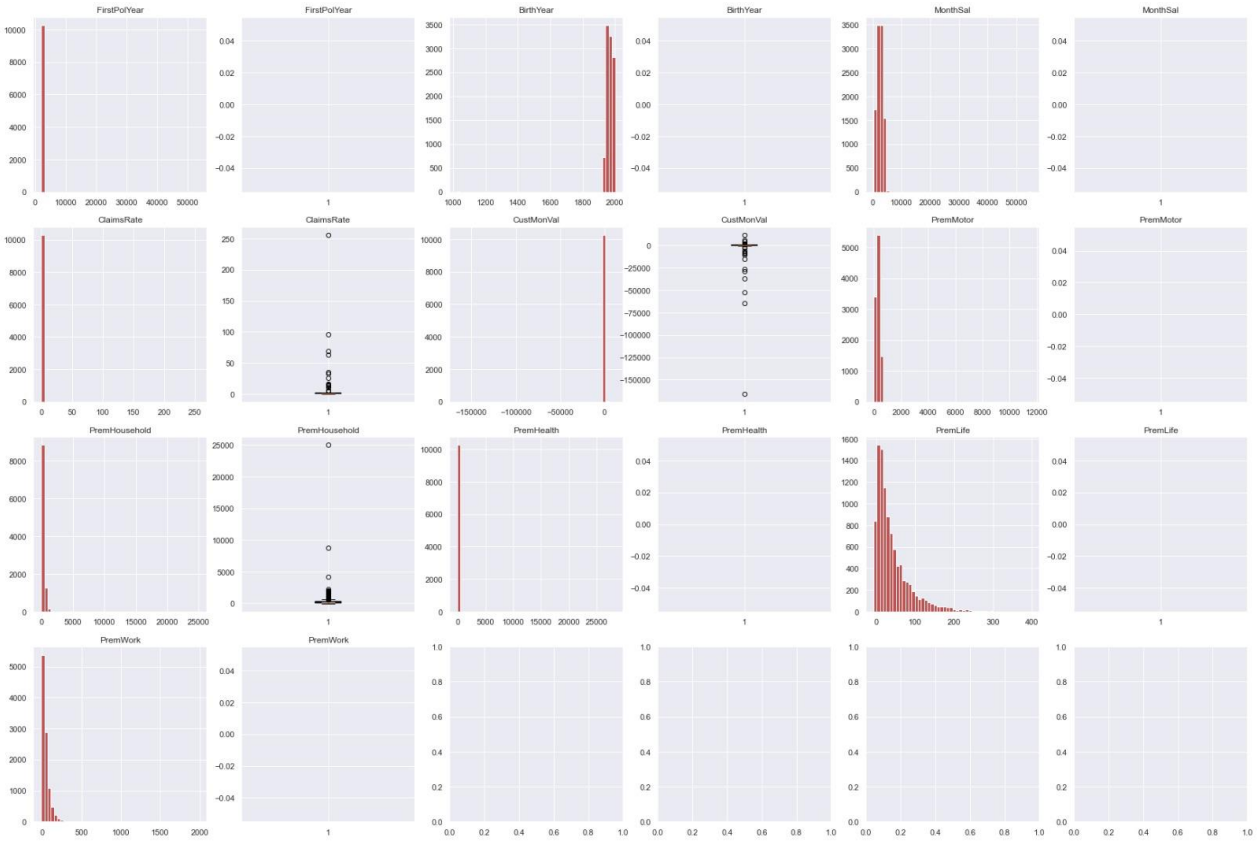
scikit-learn [1] (2023): sklearn.preprocessing.power_transform. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.power_transform.html, updated on 06/01/2023., checked on 06/01/2023.

scikit-learn [2] (2023): sklearn.preprocessing.StandardScaler. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html updated on 06/01/2023., checked on 06/01/2023.

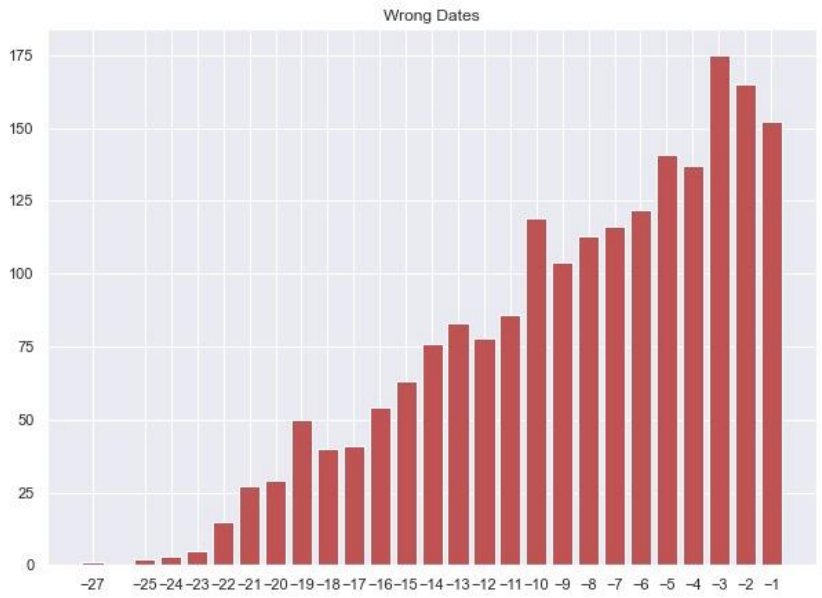
VanderPlas, Jake (2018): In Depth: Gaussian Mixture Models | Python Data Science Handbook. Available online at https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html, updated on 06/01/2023., checked on 06/01/2023.

6. Appendix

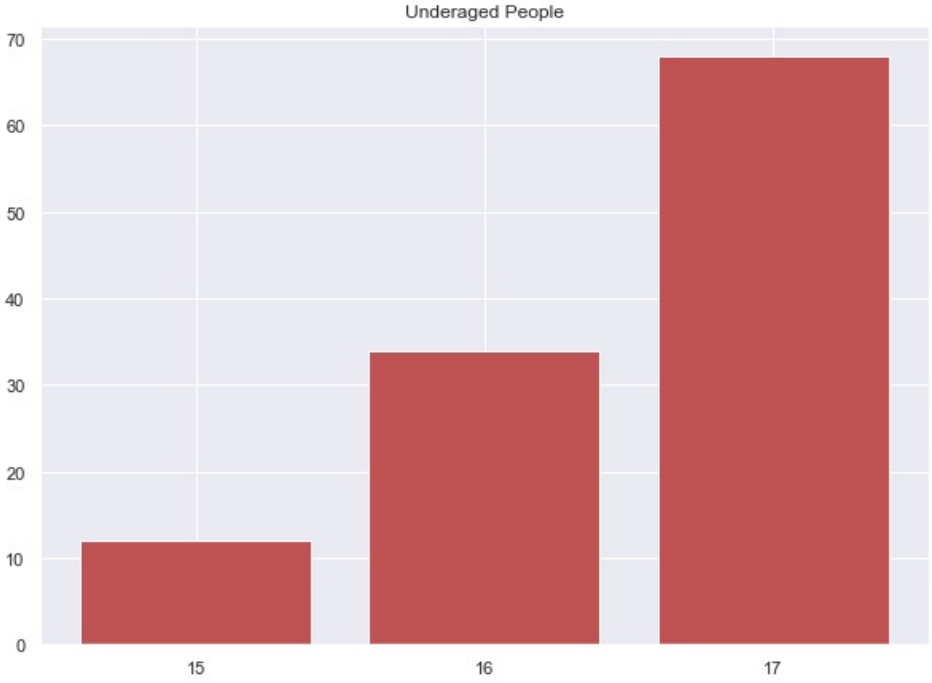
6.1. Distribution and Boxplots before Pre-processing



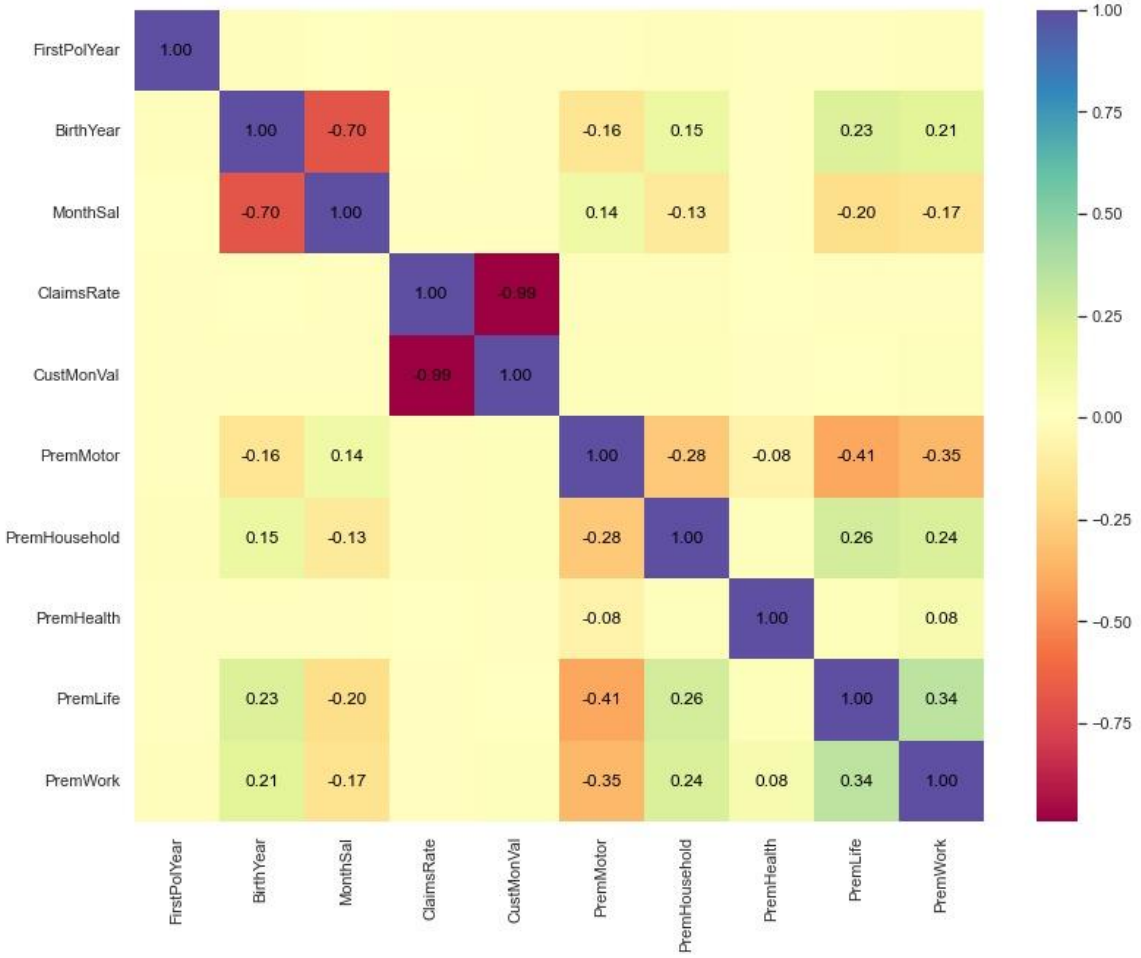
6.2. Frequency of wrong “FirstPolYear” Dates



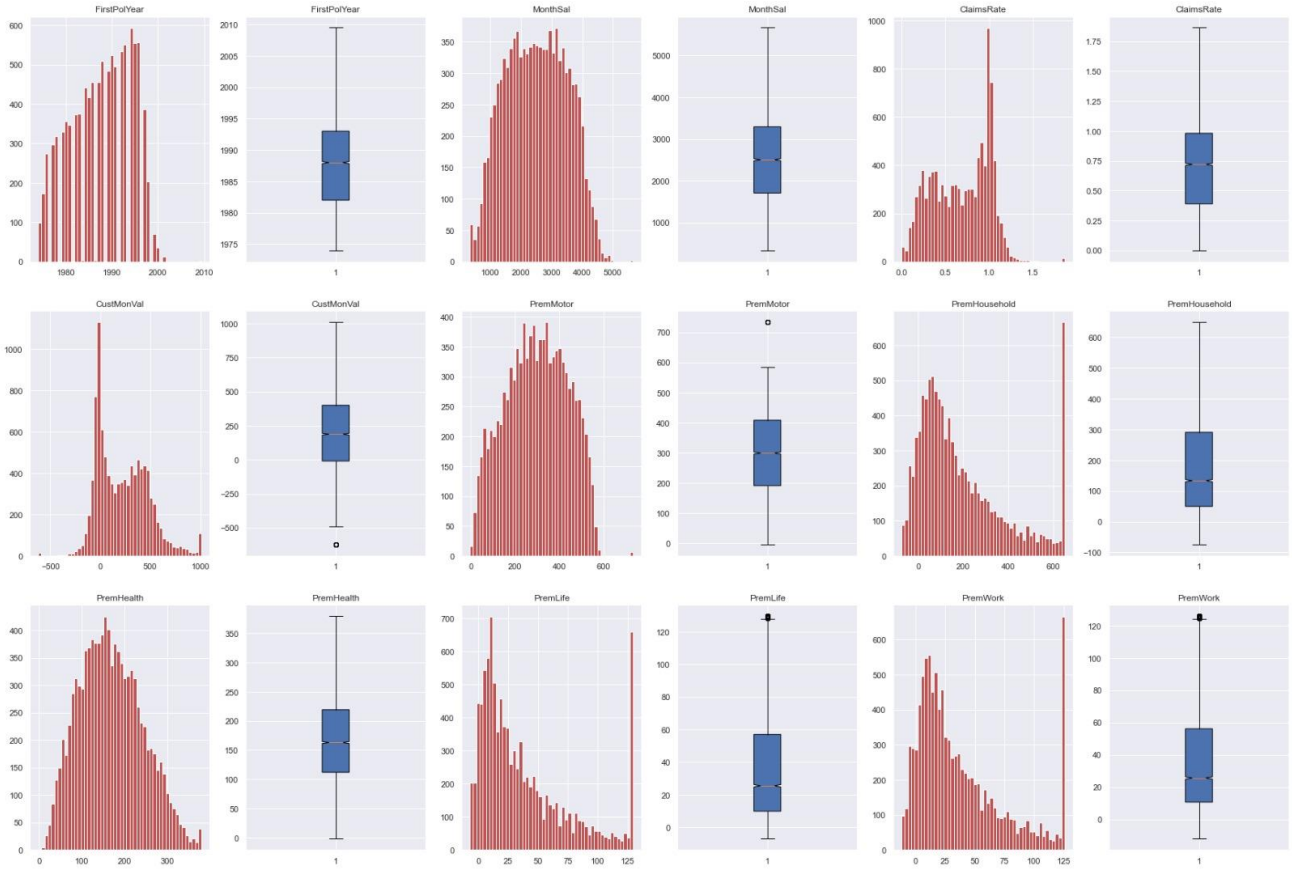
6.3. Frequency of underaged people



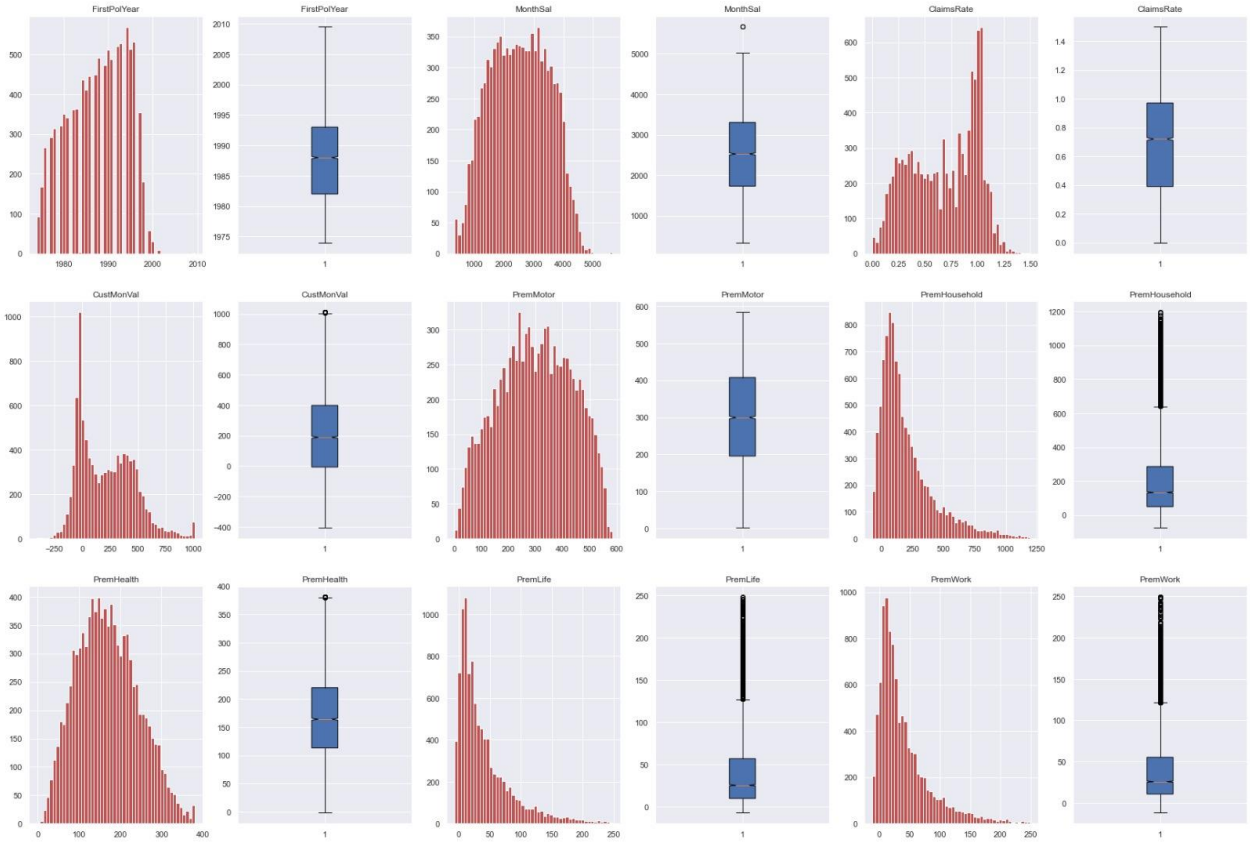
6.4. Initial Correlation Matrix



6.5. Distributions and boxplots after the first pre-processing iteration



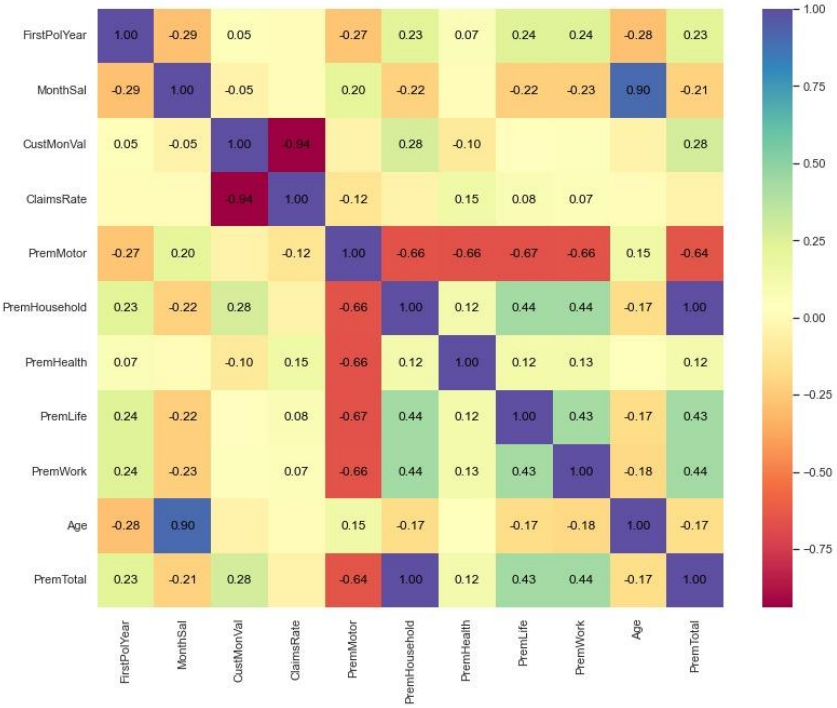
6.6. Distributions and Boxplots after the second pre-processing iteration



6.7. Created Features with explanations

Variable Name	Description	Calculation
Age	The Age of the customer	2016 - BirthYear
PremTotal	Sum of the premiums of the year 2016	$\sum PremiumVariables$
Motor_Prem_Share	Share of Motor Premiums on the overall premium value	PremMotor/PremTotal
Household_Prem_Share	Share of Household Premiums on the overall premium value	PremHousehold/PremTotal
Health_Prem_Share	Share of Health Premiums on the overall premium value	PremHealth/PremTotal
Life_Prem_Share	Share of Life Premiums on the overall premium value	PremLife/PremTotal
Work_Prem_Share	Share of Work Premiums on the overall premium value	PremWork/PremTotal
Motor_Sal_Prop	Share of the yearly salary on motor premiums	PremMotor/(MonthSal)*12
Household_Sal_Prop	Share of the yearly salary on on household premiums	PremHousehold/(MonthSal)*12
Health_Sal_Prop	Share of the yearly salary on health premiums	PremHealth/(MonthSal)*12
Life_Sal_Prop	Share of the yearly salary on life premiums	PremLife/(MonthSal)*12
Work_Sal_Prop	Share of the yearly salary on life premiums	PremWork/(MonthSal)*12

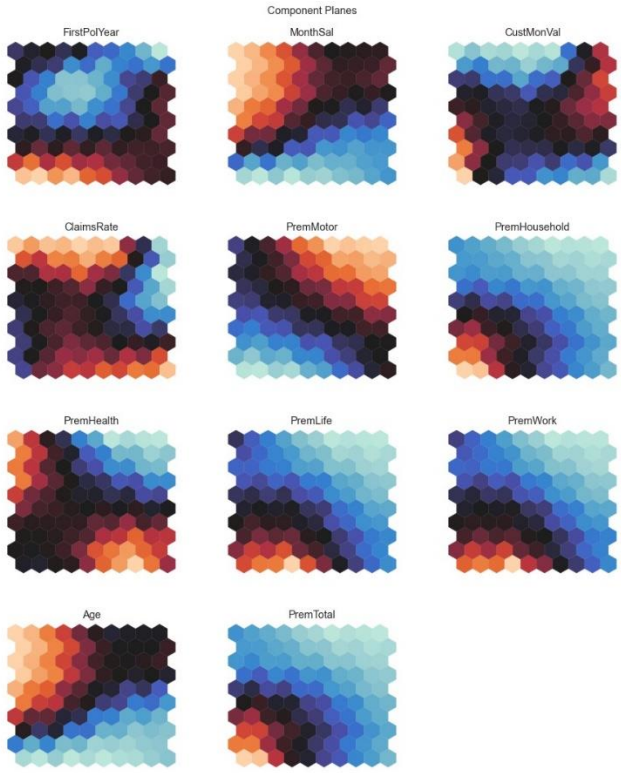
6.8. Correlation matrix after the pre-processing



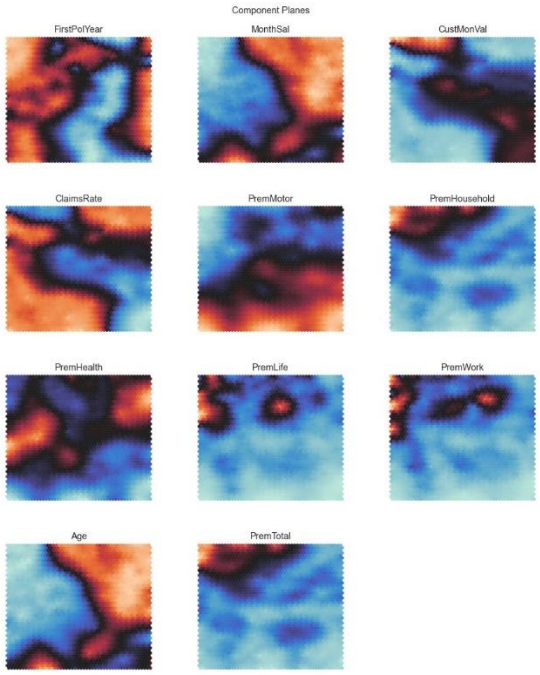
6.9. Perspective Split -Feature List

Demographic Perspective	Values/Product Perspective
Age	FirstPolYear
MonthSal	ClaimsRate
EducDeg	PremMotor
Children	PremHousehold
	PremHealth
	PremLife
	PremWork

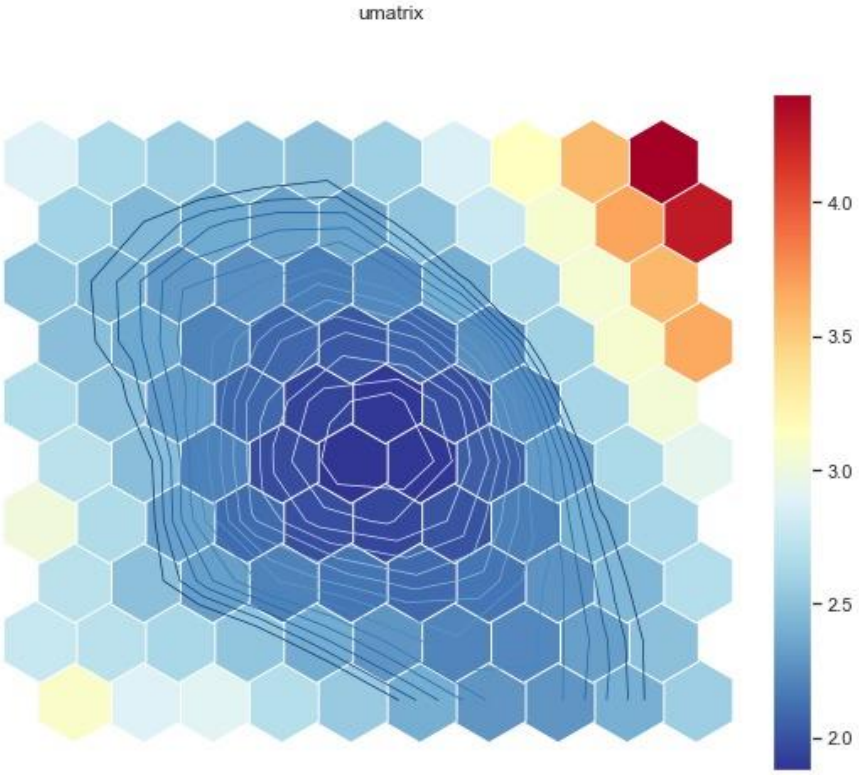
6.10. Pre-processing component planes - small grid



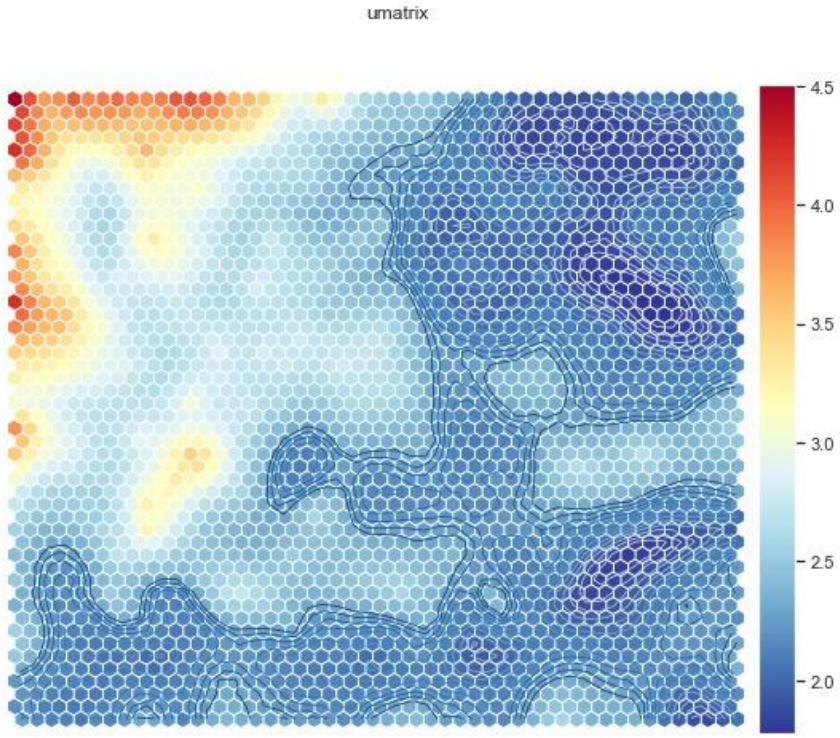
6.11. Pre-processing component planes – large grid



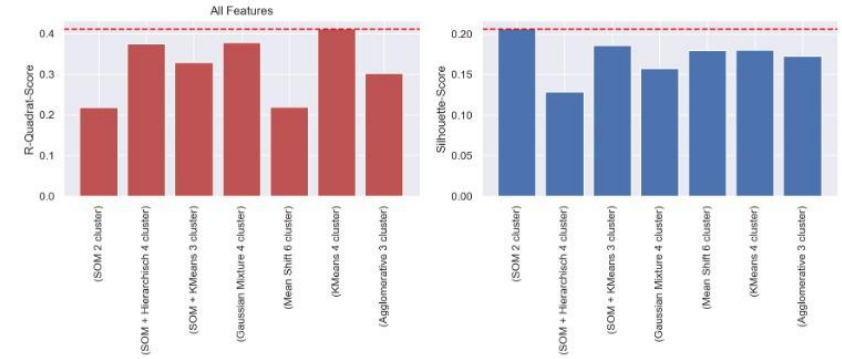
6.12. U-matrix small grid

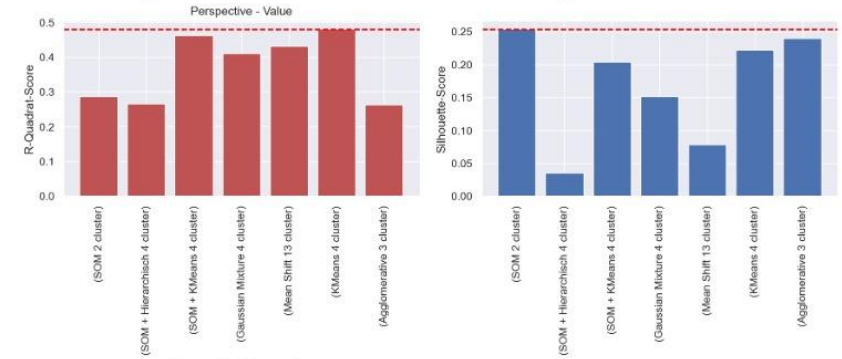


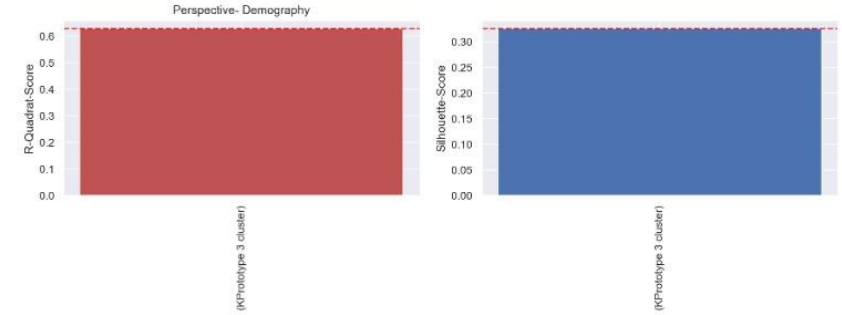
6.13. U-matrix – large grid

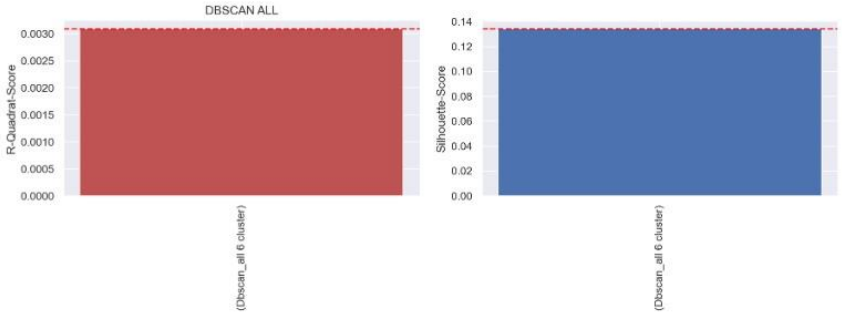


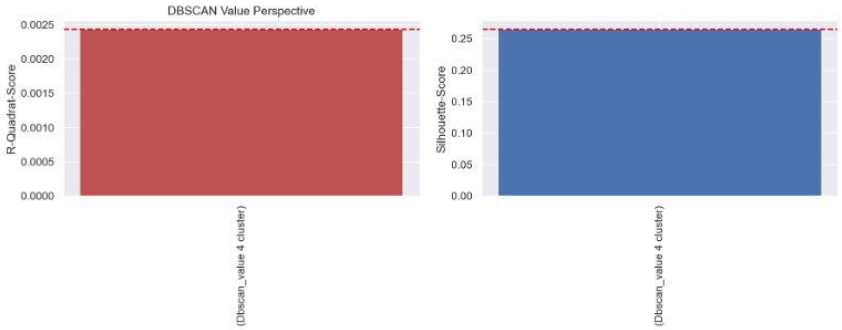
6.14. R2 and Average Silhouette Score for all used algorithms



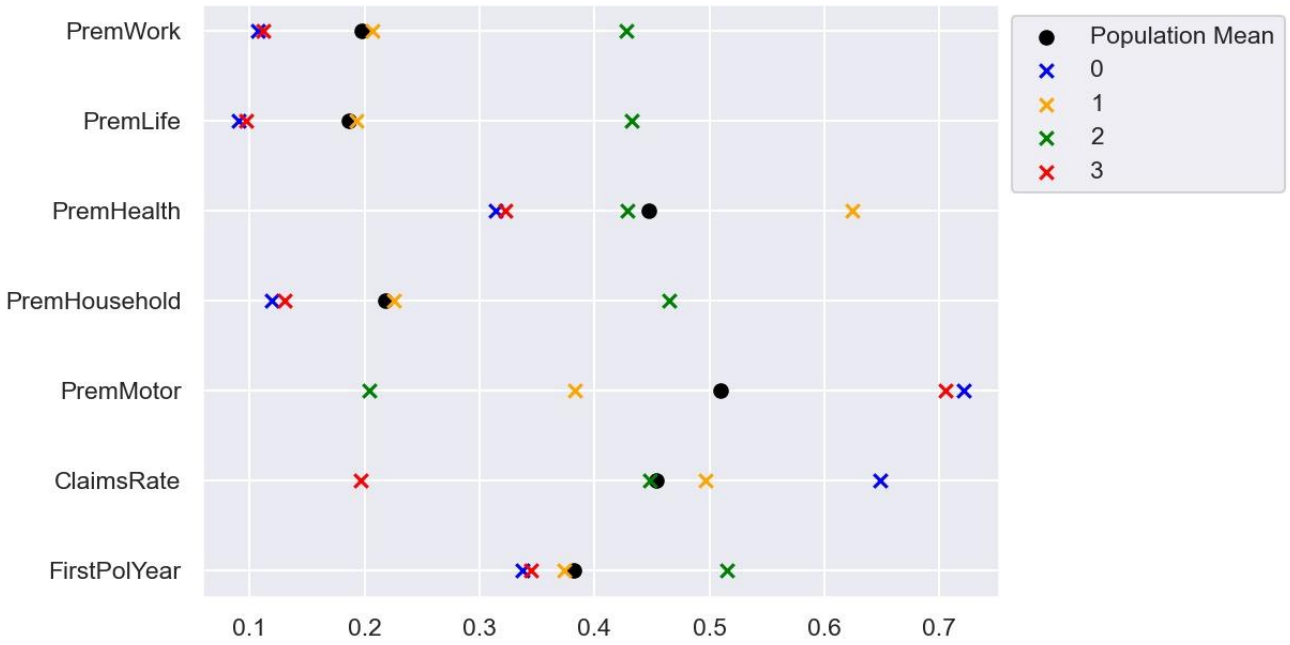




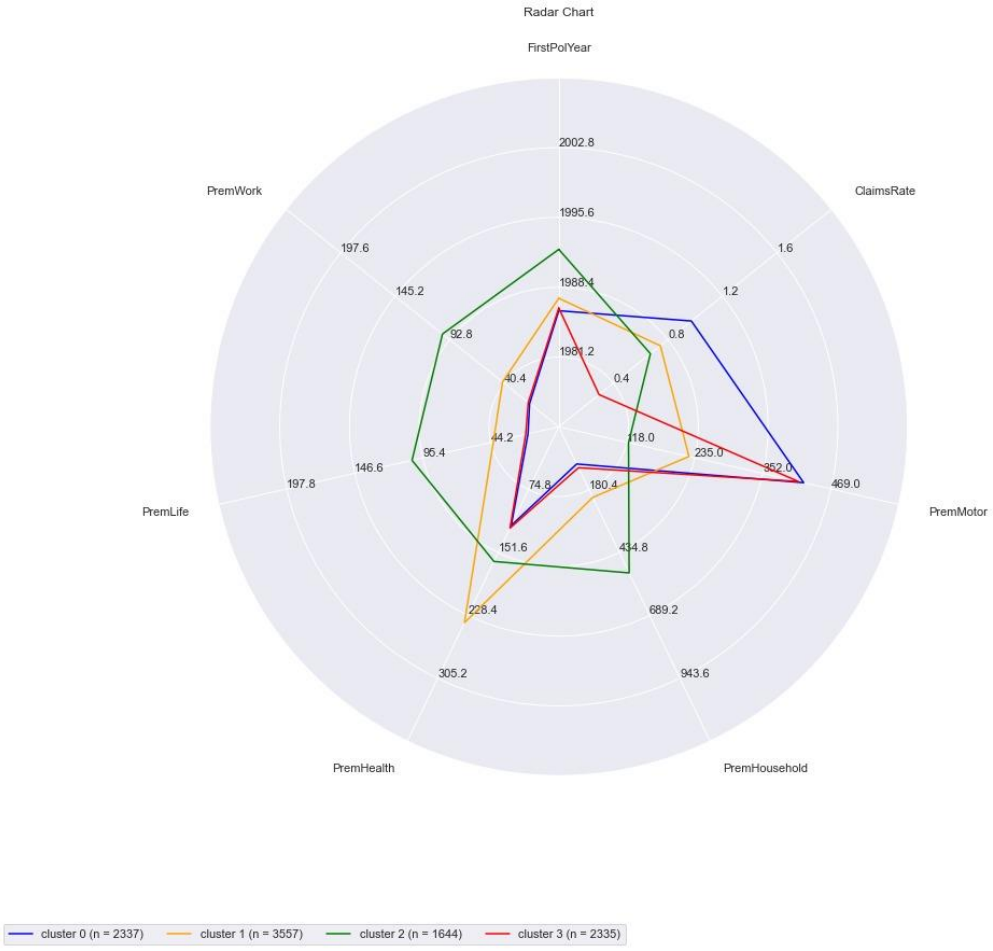




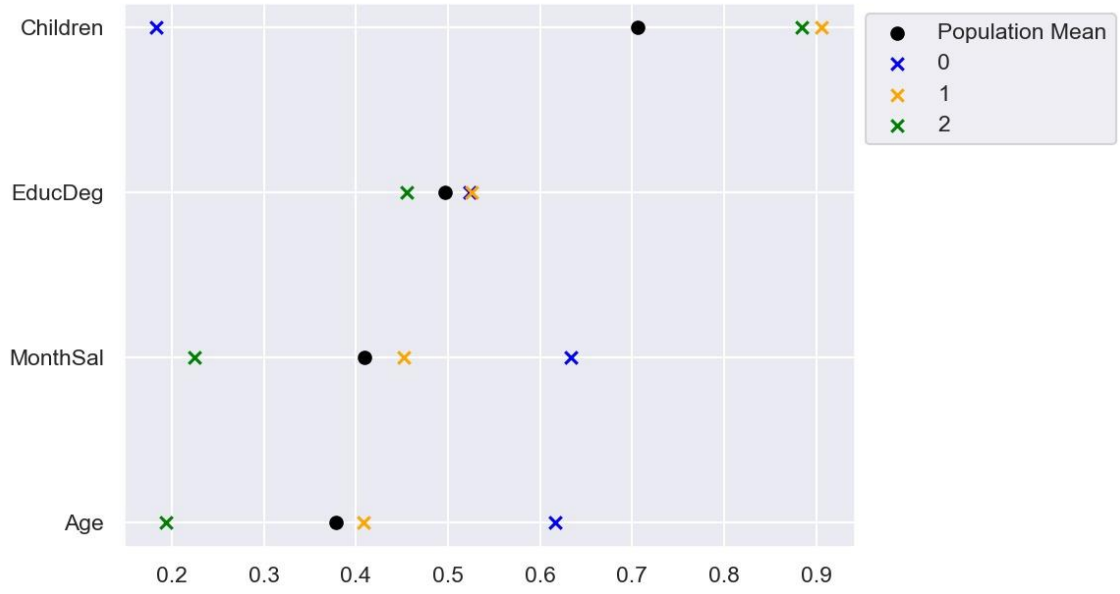
6.15. k-means (value) – cluster means vs population mean



6.16. k-means (value) – Radar Plot

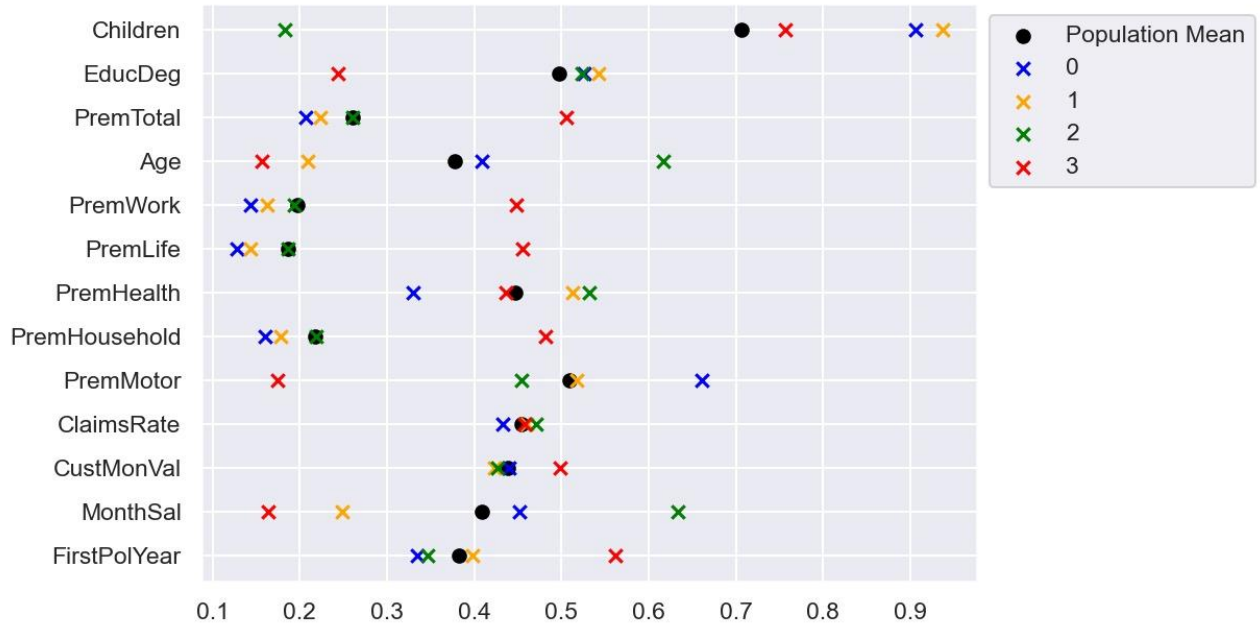


6.17. k-prototype – cluster means vs population mean

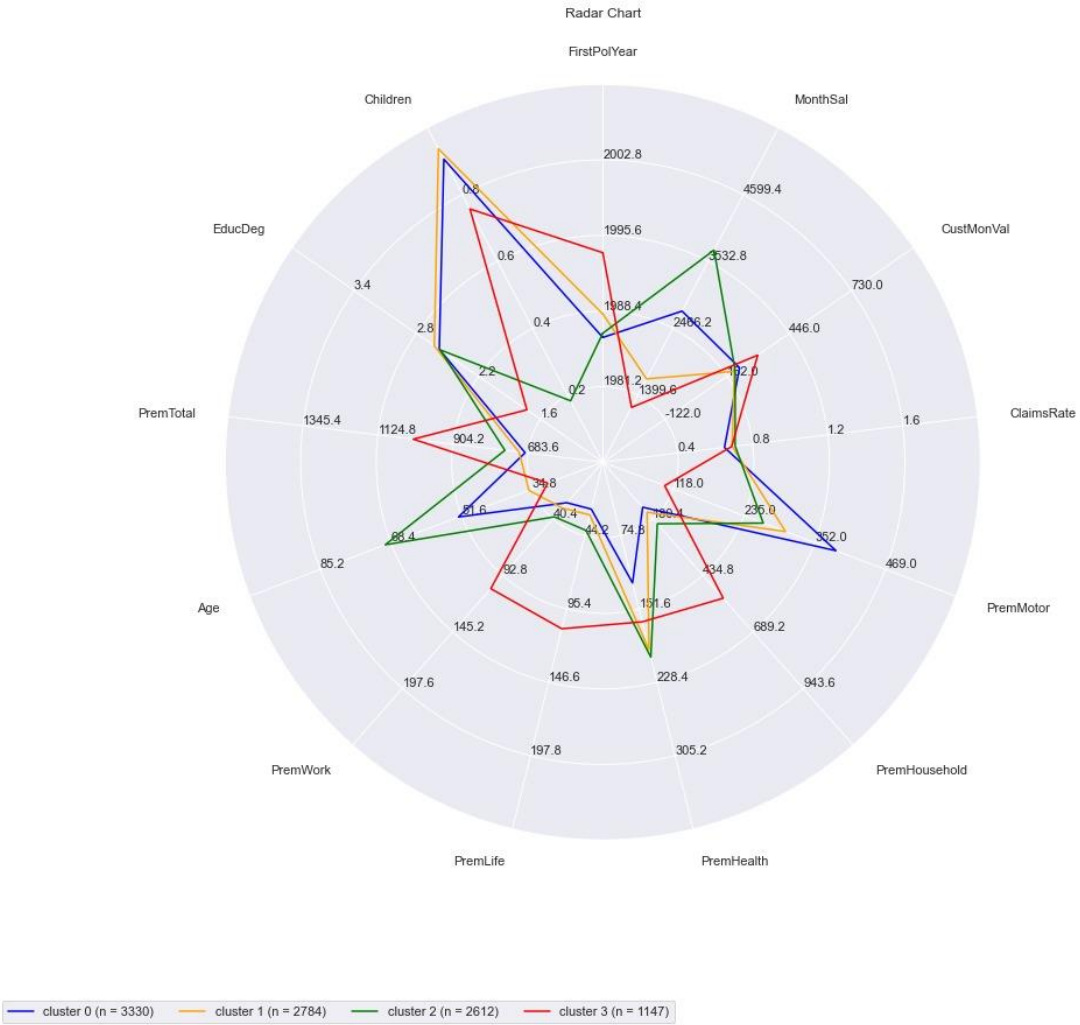


6.18. k-prototype – Radar Plot

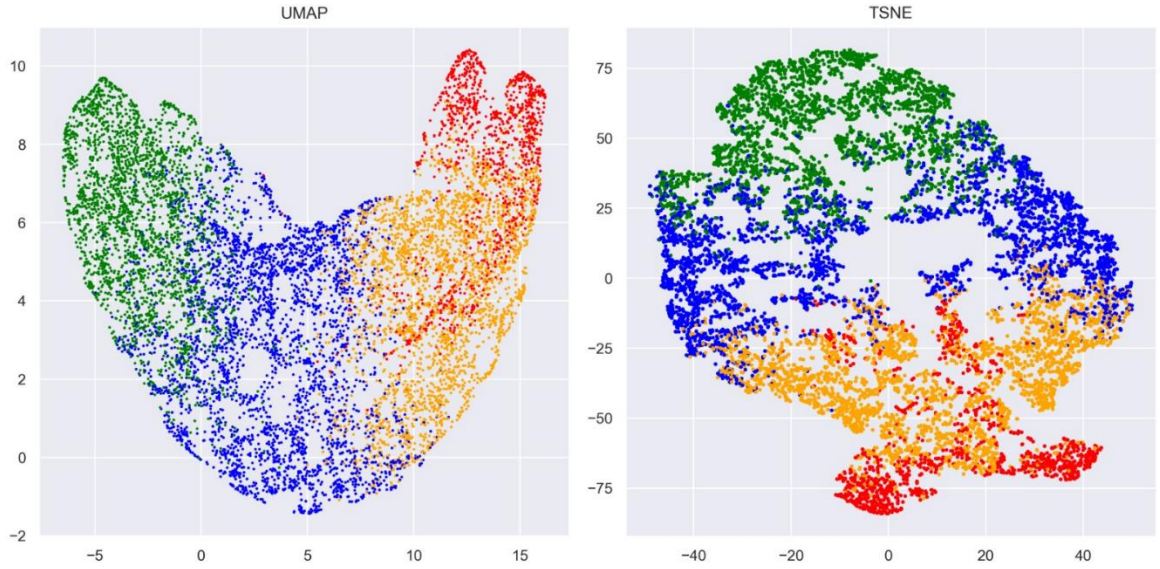
6.19. Final Merged Perspective – Population Mean vs Cluster Means



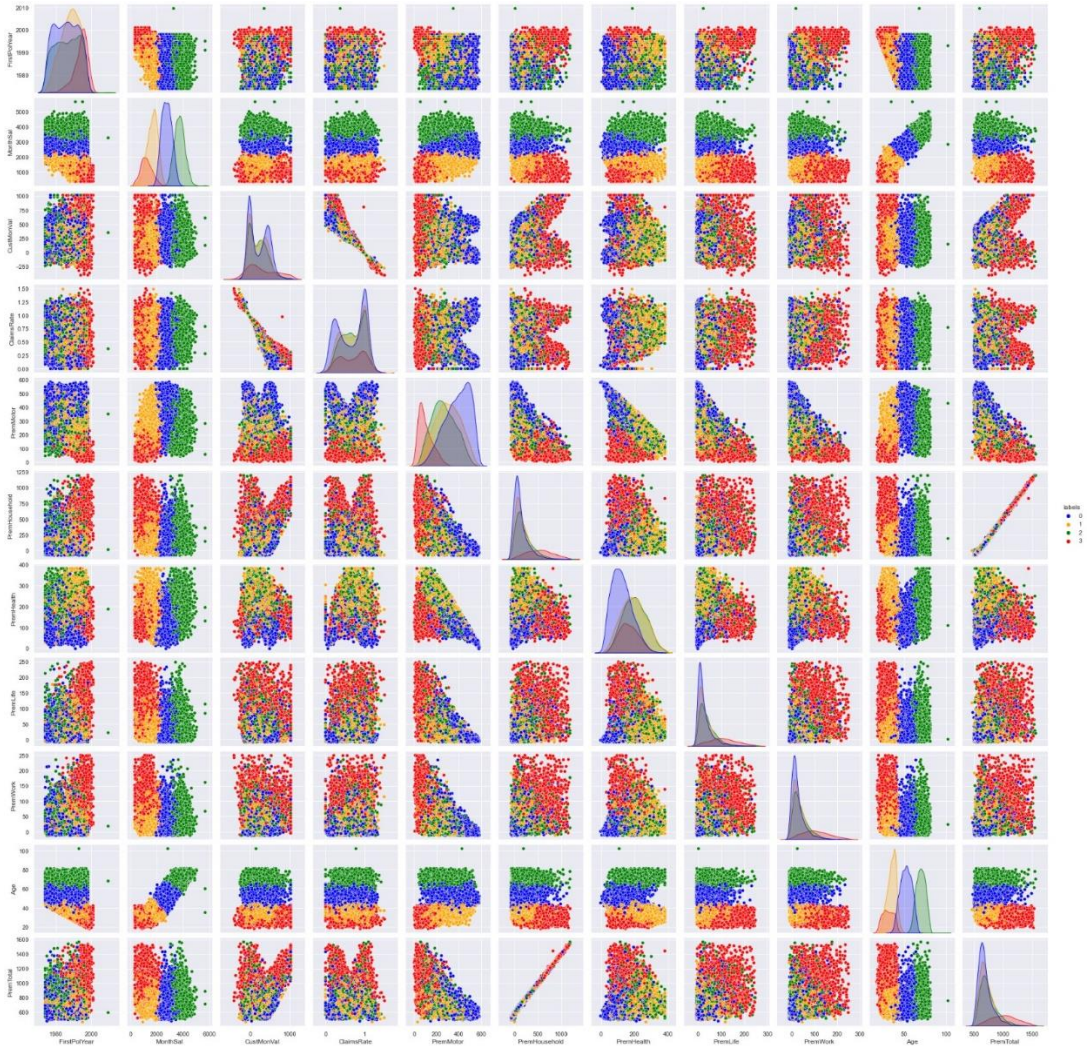
6.20. Final Merged Perspective – Radar Plot



6.21. Final merged perspective – UMAP and T-SNE (Color Coding: Labels)



6.22. Final merged perspective – Pairplot (color Coding: Labels)



6.23. Final merged perspective – Decision Tree