Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

A2Z Insurance

Group DM

Fernando Cruz, number: 202220646

Inês Magessi, number: 20220590

Pedro Fernandes, number: 20220592

December, 2022

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

INDEX

1. Introduction 3
2. Data Exploration 3
    2.1. Types conversion 3
    2.2. Duplicates 3
    2.3. Coherence check 3
    2.4. Data visualization 4
3. Data Preprocessing 4
    3.1. Outliers using IQR 4
    3.2. Feature engineering 5
    3.3. Transforming skewed data 5
    3.4. Scaling 5
    3.5. Missing values 5
    3.6. Outliers using DBScan 6
4. Segmentation Perspectives 6
5. Assessment of the Discriminative Power of Features 6
    5.1. Numerical features 6
    5.2. Categorical features 7
6. Determining the Clustering Tendency of Data 7
7. Clustering 8
    7.1. Socio Demographic Segmentation 9
    7.2. Value Segmentation 10
    7.3. Product Segmentation 10
8. Clusters Profiling 11
9. Merging the Perspectives and Final Solution 11
10. References 13
11. Appendix 14

1. Introduction

Customer segmentation is a valuable tool for businesses looking to better understand and serve their customers. By identifying distinct groups within the customer base, businesses can tailor their marketing and sales efforts to meet the needs of specific customer segments.

In this project, we applied data mining techniques to customer data from A2Z Insurance in order to identify distinct segments within the customer base. Our analysis revealed valuable insights, which will be useful for A2Z Insurance, as it seeks to improve its marketing and sales efforts. In the following sections of this report, we will delve deeper into our research and findings.

2. Data Exploration

We began by conducting a general overview of the data [Table 2.1 and 2.2]. Upon this stage, we immediately identified a structural related problem that needed to be addressed. When the .sas file was imported, pandas created its own index. However, since we had the feature *CustID*, we decided to use it as the index instead. We then proceeded to a further analysis of the dataset.

2.1. Types Conversion

There were some data type conversions that needed to be made on the feature *EducDeg*, given that it was an ordinal feature saved as a byte string. Therefore, we converted the possible values *Basic, High School, BSc/MSc and PhD* to the values 1, 2, 3 and 4, respectively. This way, we were able to preserve the logical order of the options.

2.2. Duplicates

We searched for duplicates and we found 3 identical entries. Seeing that having clients with the exact same characteristics is redundant, we decided to remove them.

2.3. Coherence Check

At this stage, we noticed a few irregularities that needed to be resolved. The first thing we checked was the base year of the database - 2016. This meant that it would not be possible for a customer to have become a client after that year. We located one observation that did not meet this requirement, so we decided to simply drop it from the dataset.

After that, we checked if there were some individuals with a first year as a customer registered as earlier than the year of birth and, surprisingly, we found 1997 observations where that happened to be the case. Given the large amount of observations, we decided not to remove them and, instead, tried to find a plausible explanation for this presumed incoherence. The most reasonable cause we found was that, perhaps, those individuals were children of prior customers who subscribed plans for them when they were still underaged. As the premium was paid by the parent, but the insurance was actually for the child,it seemed possible that the *FirstPolYear* of the customer was recorded under the parent's information, while the *CustID* still referred to the insured person. Therefore, we decided that, following this explanation, it would be more logical to set the *FirstPolYear* to the *BirthYear* of those clients, as this would represent the customer's actual first year with the company.

2.4. Data Visualization

Before we get into the preprocessing stage we decided to do some visualizations in order to better understand the data and identify patterns and relevant correlations.

First, we visualized the distributions of the numerical features and immediately identified the presence of outliers [Fig 2.4.1]. After that, we checked the *Spearman Correlation* between numerical and ordinal features [Fig 2.4.2] and we concluded that most of them were moderately correlated. However both *CustMonVal* and *ClaimsRate* showed a strong negative correlation of -0.97, while *BirthYear* and *MonthSal* had a similar high correlation of -0.93. Therefore, we decided that using either *CustMonVal* and *ClaimsRate*, or *BirthYear* and *MonthSal* for clustering would be redundant.

Later, we further analyzed the relationships between all features and how they vary together. We found that unprofitable customers (*CMV* = -1) had higher motor insurance spending and lower spending in other areas compared to profitable customers [Fig 2.4.3]. This suggests that motor insurance may be riskier and lead to more losses for the company. We also concluded that there is a similar amount of total spendings per number of years as customer, despite seeing a slight increase with the most recent clients [Fig 2.4.4]. Finally, we were able to conclude that people from different ages spent more money in different insurances, with younger clients spending considerably less in motor and more in household [Fig 2.4.5]. We also observed that the average amount spent on premiums is similar across geographical locations, which suggests that this variable may not have strong discriminatory power. [Fig 2.4.6]. Lastly, we noticed that the majority of clients have a contract for all 5 insurance types [Fig 2.4.7].

3. Data Preprocessing

3.1. Outliers using IQR

Outliers are extreme values that are clearly different from the other observations, and they should never be removed without proper investigation because they may contain a lot of revealing information if they are not due to error. In order to study and control them, we decided to visualize boxplots for all the numerical features [Fig 3.1.1].

After that, we manually removed the observations we regarded as outliers, considering not only the plots but also domain knowledge. We only retained data that we believed to be representative of the general population, i.e. observations that met the following criteria:

- *BirthYear* > 1900
- *MonthSal* < 20000
- *CustMonVal > - 2000*
- *ClaimsRate* < 4
- *PremMotor* < 2000
- *PremHousehold* < 1500
- *PremHealth* < 400
- *PremLife* < 300
- *PremWork* < 300

Ultimately, we removed approximately 0.80% of the data, which we consider a very acceptable value. We decided not to remove all the observations that fell out of the IQR because it would be inappropriate and would lead us to an improper and not desired amount of lost data, which could potentially result in the loss of precious information. We then plotted the data again and we could then see boxplots with large boxes, as desired [Fig 3.1.2].

3.2. Feature engineering

Feature engineering is the process of designing new features from raw data that can help improve the quality and relevance of the features used to perform customer segmentation. After examining the original variables, we decided to generate additional ones that might provide more valuable insights. Seeing that we used information from the *Prem* columns to create the new features and those were missing some values, we first filled them, as it will be explained in Section 3.5. All the new variable names and corresponding descriptions can be found in Table 3.2.1.

3.3. Transforming skewed data

Skewed data may negatively impact the performance of some models and make it difficult to draw meaningful conclusions from the data. This is the case for the *K-Means* algorithm, given that it assumes that all variables have the same spherical variance, thus being roughly normally distributed.

Since we were planning on using this and other algorithms that make similar assumptions ahead, we tried to minimize the skewness in data. First, we examined the skewness measure of each feature: values greater than 1 or less than -1 mean that the distribution is highly right and left skewed, respectively. Some transformations that can be applied to make features more similar to a normal distribution include the square, root and log conversions. The skewed features, the respective applied transformations and post transformation skewness can be seen in Table 3.3.1.

3.4. Scaling

In the context of clustering, keeping all features in the same scale helps ensure that the distance calculations are more meaningful. Otherwise, if one feature is much larger than the others, it will dominate the distance calculations and the clusters may be primarily based on that feature.

We decided to use a *StandardScaler* in this stage for two main reasons. Firstly, the majority of the features now follow a gaussian distribution, thus being an appropriate scaler to use. Also, given we will be using algorithms that better work on spherical and similar variance clusters, transforming the data to have an approximate mean of 0 and a standard deviation of 1 may help improve their performance.

3.5. Missing values

As mentioned before, prior to engineering new features, we saw that almost all variables referring to premiums had missing values, so we decided to fill them with value 0. This decision was made under the assumption that the premiums were not recorded because they had not been contracted.

After filling the premiums' missing values and scaling the data, we still had 120 missing values to fill among the variables [Table 3.5.1]. First we started by dropping the record with a missing *GeoLivArea,* given it was only one. Then, we filled the gaps of *BirthYear* with the median, given that we considered that none of the other variables would give us useful information to fill them.

Following that, we used the *KNNImputer* to fill in the missing values of *MonthSal* and *FirstPolYear*. We considered that *BirthYear* was the only relevant feature to find the nearest neighbors

of the records with missing *FirstPolYear*, and to fill the missing values of *MonthSal* we only used information from *BirthYear* and *PremWork.* Finally, we did the same to fill the gaps of *Children* and *EducDeg,* both categorical variables, this time using the customized *KNNModeImputer* [Annex 11.2], that instead of returning the mean of the missing feature of the nearest neighbors, returns the mode. For that we only used *BirthYear* and *MonthSal* to find the K nearest neighbors.

3.6. Outliers using DBScan

After filling the missing values, we re-evaluated the presence of outliers, this time using DBScan. This algorithm works by identifying groups of closely-located points, the clusters, and labeling any points that are not part of them as outliers. Through this process, we were able to identify 94 observations that did not fit within the dense clusters of the data, potentially indicating unusual or unexpected values. We were left with a total of 1.72% of the data after discarding them.

4. Segmentation Perspectives

With the data cleaned and preprocessed we were ready to proceed with clustering. Knowing that the data includes a range of features related to various characteristics of the customers, we have decided to perform clustering from multiple perspectives to see if different groupings emerged when looking at the data from different angles. Given the nature of the data we had available, we decided to group our customers based on 3 types of features: their socio-demographic characteristics; variables that reflect their value to the company; and product preferences.

After defining the perspectives, we divided our variables into categories. This helped us define which features could be used when performing clustering under each perspective. For the socio-demographic view, the most suitable variables were *BirthYear, EducDeg, MonthSal, GeoLivArea* and *Children.* Next, we considered the following features as variables that reflect the value of a customer for the company: *FirstPolYear, MonthSal, CustMonVal, ClaimsRate, total_premiums* and *total_reversals.* Finally, to segment the clients based on their product preferences, we chose the *'PremX'* features as a possible set of features and also the engineered '*Premium'Percent* variables.

5. Assessment of the Discriminative Power of Features

Before using clustering algorithms on our data, we wanted to evaluate which of the previously grouped variables were the most effective in terms of their ability to distinguish customers between different groups. This would help us ensure that we would only be using the most reliable variables.

5.1. Numerical Variables

By analyzing the relationship between all pairs of numerical variables [Fig 5.1.1] and by observing the *Spearman Correlation* between metric and ordinal features [Fig 5.1.2], we were able to see, once more, that the high correlation between *CustMonVal* and *ClaimsRate (*-0.98) and *MonthSal* and *BirthYear* (-0.93) makes these variables redundant when used together. The same showed to be true for the '*PremX*' and '*Prem'Percent* features. Furthermore, it also became evident that neither *CustMonVal* nor *ClaimsRate* were significantly correlated with any other variable included in the *Value* perspective set. This meant that none of them would contribute to better grouping customers, thus we decided not to use them for clustering, but only for cluster profiling.

At last, we trained a SOM using all metric features and then observed the resulting component planes [Fig 5.1.3]. We realized that all features, except for *total_reversals,* were able to distinguish between different groups. Again, it was also clear the redundancy between *BirthYear* and *MonthSal* as well as *CustMonVal* and *ClaimsRate,* seeing they were inverse along SOM units.

5.2. Categorical Variables

Knowing that the previously mentioned methods were not suited for the categorical variables, and since these variables were selected as potential features for performing *Socio-Demographic* clustering along with the numerical features *BirthYear* and *MonthSal (*we excluded the last one given the mentioned redundancy), we decided to evaluate their relevance by plotting them against each other.

GeoLivArea seemed unchangeable along the values of the other features [Fig 5.2.1, 5.2.2 and 5.2.3] and *EducDeg* also looked the same for people with and without children [Fig 5.2.4]. However, we could see some differences in customers' age among different education degrees [Fig 5.2.5]. Next, we saw that *Children* was really discriminative when used together with *BirthYear* [Fig 5.2.6]. At this point *GeoLivArea* and *EducDeg* appeared to be the less informative features, so we decided to use a violin plot [Fig 5.2.7 and 5.2.8] to compare the distributions of the sets of 3 possible features we were then considering. Ultimately, we discarded *GeoLivArea* from our analysis, as it was the less discriminatory feature.

6. Determining the Clustering Tendency of Data

After selecting the final sets of features, we decided to also analyze the clustering tendency of the data using them. This could help us avoid the time and effort of performing unnecessary clustering steps without knowing more confidently that the results would be satisfactory.

Initially, visualized 3D plots of the combinations of numerical features chosen for *Value* [Fig 6.1] and *Product* [Fig 6.2 and 6.3] segmentations. Despite the limitation of only being able to visualize up to the third dimension, we were able to see non uniform distributions of points, which was a good prognosis regarding the clustering tendency of the data.

Next, we observed the pairwise distances between points using the chosen features for the *Socio-Demographic, Value* and *Product* perspectives [Fig 6.4, 6.5, 6.6 and 6.7]. Given that the *Socio-Demographic* features include both numerical and categorical data, we used the Gower distance instead of the usual Euclidean distance. More information about this distance can be found in the Appendix 11.1. Regardless of not being able to distinguish clear clusters, the presence of points closer and farther from each other suggests the potential for customer grouping.

To conclude our analysis, we applied the Hopkins test, a statistical tool that evaluates the distribution of data against the null hypothesis that the data is uniformly distributed, meaning that there are no discernible patterns or groups within the data. A value higher than 0.75 suggests clustering tendency at a 90% confidence level, which we were able to confirm for all sets of features. To perform this test we resorted to an online already implemented code [Github link].

Having all these results in mind, we became totally assured that we could go forward with clustering using the chosen sets of features.

7. Clustering

During this stage, we employed multiple clustering techniques following the different perspectives. The algorithms used for the *Socio-Demographic* segmentation were different from the ones applied for the *Value* and *Product* segmentations given the presence of categorical variables, and all the steps will be explained in detail in the next section. However, since only numerical variables were chosen for the latter segmentations, the exact same steps were followed for both *Value* and *Product* clustering perspectives. These steps are as follows:

In the first place, we employed *Hierarchical Clustering*. This algorithm starts by treating each data point as a separate cluster and then combines the closest two clusters based on some measure of similarity, in our case, the euclidean distance. There are some hyperparameters to be tuned, such as the linkage method and the final number of clusters. To choose the first one, we compared the *R2 scores* of clustering with *single, complete, average and ward* linkages, across a range of 1 to 5 clusters. After selecting the optimal linkage, we plotted a dendrogram without predefining a maximum number of clusters to visually assess the best number of groups. We did this by examining the size of the distances between observations, as the larger the more separation between clusters.

The next clustering algorithm applied was *K-Means*. This method works by randomly initializing K initial cluster centroids and then assigning each point to the nearest cluster based on the euclidean distance to the centroid. The algorithm then adjusts the centroids to the center of the points in each cluster and repeats the process until convergence. For this algorithm, it is also required to pre-define the number of clusters. To do that we relied on 3 methods: the *Elbow* method, the *Silhouette Score* and employing *K-Means* with a large number of clusters followed by *Hierarchical Clustering*. We chose the number of clusters showing best results on the majority of these methods.

Then, we used density based clustering techniques. The first to be employed was the *Mean-Shift*. This one works by treating each data point as a potential cluster center, and then adjusts the positions of the points towards the densest areas of the feature space until the points converge into final *attraction basins*. One problem with this algorithm is that it is sensitive to the choice of the bandwidth parameter, which determines the size of the region around each point that is considered in the density calculation. This hyperparameter was automatically chosen using a *sklearn* function.

The second density based clustering algorithm applied was *DBScan.* While this algorithm has several advantages such as being able to handle data with varying densities, shapes and not requiring the pre-specified number of clusters, it is also sensitive to the choice of the parameters *eps* (the size of the neighborhood around each point) and *min_samples* (the min number of points required on the neighborhood of a point to consider it core point). We used the recommended 2*dim value for the *min_samples* parameter and then plotted the frequency of the distances of each point to its *min_samples* closest neighbors to choose the ideal value for *eps.*

Lastly, we performed both *Hierarchical Clustering* and *K-Means* on top of *SOM units*. *SOM*s are a type of artificial neural network designed to represent high-dimensional data in a low-dimensional space, preserving the topological structure of the input data. We started by creating and training a 15 x 15 *SOM*. Then, we chose the best number of clusters for the *Hierarchical clustering* and *K-Means* with the same methods described earlier, this time clustering the *SOM units* instead of the original points. Finally, the BMUs clusters were merged with the original points.

In the end, each clustering algorithm solution was evaluated using the *R2* and *silhouette scores* in order to later choose the one with the best performance for the final clustering. These results, as well as the best parameters found for each method are detailed in sections 7.2 and 7.3.

7.1. Socio-Demographic Segmentation

As mentioned before, knowing that the *Socio-Demographic* variables (*BirthYear, EducDeg* and *Children)* are both categorical and numerical , we had to use algorithms that work with mixed data types. The employed methods were *K-Prototypes*, and then *Hierarchical Clustering* and *K-Medoids* using the *Gower* distance. The evaluation of the clustering solutions was made by examining the *silhouette score*, using the non euclidian distance measures employed by each algorithm.

K-Prototypes is a clustering algorithm similar to *K-means* that works by partitioning the data into *K* clusters, where each cluster centroid is represented by a prototype vector. This vector consists of both continuous and categorical variables. The algorithm assigns each data point to the cluster with the closest prototype vector, using a distance measure that takes into account both types of variables. Given the need to choose the number of clusters beforehand, we decided that the ideal number of clusters was 6 using the *Elbow method* [Fig 7.1.1].

After that, we applied *Hierarchical clustering* using the *Gower* distance. To choose the best linkage method we plotted the silhouette scores for each model (*single, complete* and *average*), across a range of 2 to 9 clusters [Fig 7.1.2]. *Ward* linkage cannot be used with Gower distance because it requires a measure based on the variance within clusters, while this distance is a measure of dissimilarity, not based on variance. The *average* linkage showed the best overall score, thus, we used it to plot a dendrogram [Fig 7.1.3] in which we observed that the ideal number of clusters is 3.

Finally, we applied *K-Medoids* with the *PAM (Partitioning Around Medoids)* algorithm, also using the *Gower* distance. Again, given the pre-defined number of clusters requisite, we used both *inertia* and *silhouette* scores [Fig 7.1.4] to decide that the ideal value was 5 clusters.

The silhouette scores of all clustering algorithms are displayed in Table 7.1.1.

7.2. Value Segmentation

In the beginning, we started by creating cell-based segments to better understand our clients at the outset. For that, we plotted the amount of observations falling inside the *total_premiums* and *ClaimRate* quartiles [Fig 7.2.1]. We considered that the customers were divided into 4 segments:

- Profitable: Low (q1 and q2) *ClaimsRate* and High (q3 and q4) *total_premiums*
- Risk-Averse: Low (q1 and q2) *ClaimsRate* and Low (q1 and q2) *total_premiums*
- Risk-Loving: High (q3 and q4) *ClaimsRate* and High (q3 and q4) *total_premiums*
- Harmful: High (q3 and q4) *ClaimsRate* and Low (q1 and q2) *total_premiums*

Then, we clustered the observations employing a variety of clustering algorithms using the variables *FirstPolYear*, *MonthSal* and *total_premiums*, as mentioned earlier in Section 7.

Firstly, *ward* was determined to be the most effective linkage for *Hierarchical clustering* [Fig 7.2.2] and the optimal number of clusters was found to be 4, as depicted in Fig 7.2.3. After that, using

the *Elbow method, Silhouette* and *Hierarchical clustering after K-Means* [Fig 7.2.4 and 7.2.5] showed that the best number of clusters to use in *K-Means* was 4. Following that, unfortunately, the *Mean-shift* clustering gave us a suboptimal result by finding only one cluster, thus becoming an irrelevant solution. Then, the DBScan with an *eps* value of 0.35 (chosen based on the plot on Fig 7.2.6) showed better but still bad results. Finally, a 15 x 15 *SOM* was trained. The resulting U-Matrix, a 2D grid with each cell representing the distance between a neuron and its neighbors, as well as the HitMap, in which we have the number of observations that found each neuron as their best matching unit, were analyzed and showed satisfactory results [Fig 7.2.7 and 7.2.8]. Lastly, to perform both *Hierarchical clustering* and *K-Means* on top of the *SOM units* we observed that the best parameters were the *ward* linkage and 4 clusters for the first algorithm [Fig 7.2.9 and 7.2.10] and, again, 4 clusters for the last one. The resulting clusters are plotted in Fig 7.2.11 and 7.2.12.

The *R2* and *Silhouette* scores of each clustering solution can be found in Table 7.2.1.

7.3. Product Segmentation

The same process used for *Value* was repeated for the *Product* segmentation, except for the cell-based segments. This time, all the above mentioned clustering algorithms were employed using one of the two chosen sets of features: *PremMotor, PremHousehold, PremHealth, PremLife, PremWork* or *MotorPercent, HouseholdPercent, HealthPercent, LifePercent*, *WorkPercent.*

Using the first set of features, the best *Hierarchical Clustering* parameters found were the *ward* linkage and 4 clusters [Fig 7.3.1 and 7.3.2]. The *K-Means* best performed using 3 clusters [Fig 7.3.3 and 7.3.4] and the *Mean-Shift* algorithm ended up, once again, only finding one cluster. The best *DBScan eps* value found was around 0.7 [Fig 7.3.5] and the solution given by this algorithm showed 4 clusters. Finally, *Hierarchical clustering* was performed on top of *SOM units* using *ward* linkage, as well as *K-Means* and both of them were set to find 3 clusters. All of these values were decided by analysis of the same type of plots used for regular *HC* and *K-Means*. The U-Matrix and HitMap can be found in Fig 7.3.6 and 7.3.7 and the units' clusters on Fig 7.3.8 and 7.3.9.

The best parameters found using the second set of features were almost identical to the ones found for the first set. The only difference was that *Hierarchical clustering* was set to find 3 instead of 4 clusters. All other parameters remained the same as the previously mentioned. The resulting plots were so similar to the ones mentioned above that we decided not to include them in the Appendix.

The scores of each clustering algorithm can be found in Table 7.3.1.

8. Clusters Profiling

After analyzing the silhouette scores in Table 7.1.1, we concluded that *K-Medoids* was the most promising algorithm regarding a *Socio-Demographic* perspective. Therefore, we once again run the algorithm using the selected parameters, which resulted in 5 clusters with somewhat distinct profiles. The distribution of observations per cluster can be found in Table 8.1. We can see the youngest and most recent clients in cluster 0, who have the lowest salary and spend the least on *Motor*, spending most on the other premiums. On the other hand, in cluster 1 we have the oldest and highest earners, who spend the most on *Health*. Regarding cluster 3 and 4, the age and salary ranges are quite average and similar. However, in cluster 3 we find the customers who spend the most on

Motor and the least on the other insurances, whereas cluster 4 groups customers with an average spending across all premiums. Finally, cluster 2 also has older customers who earn the most, but these spend less in motor and more in the majority of other insurances than clients in cluster 4 [Fig 8.1]. Regarding categorical variables [Fig 8.2], clusters 0, 2, and 4 have the largest number of the least educated people, most of whom have children, except in cluster 2 where no one has children. Clusters 1 and 3 contain more highly educated people. All clients in cluster 3 are parents, but none in cluster 1 are. The geographical living area seems uniformly distributed among clusters as expected given the conclusion of Section 5.2, and the same happened on the following cluster analysis.

Following a *Value* segmentation, the best performing algorithm was *K-Means*. The number of people in each cluster is in Table 8.2. Regarding the numerical features, 3 of the resulting clusters (0, 2 and 3) were very similar, with the major differences being on the *FirstPolYear, BirthYear, MonthSal*. Cluster 1 is the most contrasting one, showing a much lower value for *PremMotor* while having way bigger values for all the premiums except for *Health* [Fig 8.3]. Concerning categorical variables, clusters 0, 2 and 3 are, again, very similar, while cluster 1 presents considerable differences regarding the level of education: there are no customers with the highest level of education, showing a majority of people with a basic education level. Finally, cluster 0 is the only one where the ratio of people without children surpasses 0.5. Details can be found in Fig 8.4.

Finally, following a *Product* perspective, the first set of features showed better overall results than the second one, so we decided to only analyze the resulting clusters using the first set. The most favorable algorithm showed to be, once again, *K-Means* so we decided to proceed with it (clusters distributions on Table 8.3). The major differences among clusters were found on the amount spent in each premium. Cluster 0 compiles customers with the highest spending in *Household, Life and Work* and lowest spenders in *Motor*, whereas cluster 1 is the opposite, only counting with the highest spenders in *Motor* and the lowest spenders on all other premiums. Cluster 2, on the other hand, showed an average spending on all premiums, only having the highest spenders in *Health* . Clusters 1 and 2 gather the most highly educated customers, with both also having a majority of people with children. In cluster 0 are clients with a lower education level and it has a balanced amount of people with and without children. These cluster analysis plots are available in Fig 8.5 and 8.6.

9. Merging the Perspectives and Final Solution

In the end, after using a variety of clustering algorithms to create groups of customers from different perspectives, we merged these groups and ended up with 55 clusters [Table 9.1]. To arrive at a final solution, we used, once again, a clustering algorithm to group these clusters into a smaller set that would be more meaningful and useful for business purposes. This algorithm needed to be capable of handling mixed data types, as our cluster's centroids included both numerical and categorical features. Having that in mind, and knowing that *K-Medoids* was the most successful algorithm of this kind when clustering under a *Socio-Demographic* perspective, we decided to use it.

To determine the optimal number of final clusters, we applied the *K-medoids* algorithm with a range of 1 to 10 clusters and evaluated the results using the *elbow method*, *silhouette* and domain knowledge [Fig 9.1]. Based on these evaluations, we determined that a final solution with 4 clusters would be the most suitable. We were then able to assign the final labels to the original observations and analyze the results to gain further insights about the characteristics of each cluster [Fig 9.2 and

9.3]. Our cluster analysis identified four distinct groups of clients based on their characteristics and insurance spending patterns. The descriptions and respective marketing strategy for each segment of customers are described below.

- Cluster 0 - **Long-standing, wealthy:** This cluster includes 1757 observations and is composed by older, wealthier clients who have been with the company for a longer time and generally spend an average and similar amount on *Household, Health, Life,* and *Work*. This cluster gatters a balanced number of clients with and without children, all being less literate. In order to encourage these customers to continue doing business with the company, it might be a good strategy to offer them loyalty rewards and develop marketing campaigns that highlight the value and prestige of being a long-standing customer.

- Cluster 1 - **Young, profitable**: This cluster includes 1507 observations and is made up of younger, less affluent clients who spend a lot on *Household, Life*, and *Work* insurance and the least on *Motor*. They are the most recent and profitable clients, who have low levels of education and a majority has children. Given their age, these clients may be more likely to engage with the company on social media, so investing in social media marketing campaigns to reach this group and showcase the products and services might be a good strategy.

- Cluster 2 - **Risky**: This is the largest cluster counting 3713 observations. It includes middle-aged clients with an average salary who have the lowest *ClaimsRate* but also the lowest total spending on premiums. They spend a lot on *Motor* but very little on the other types of insurance and their contracts result in higher reversals than the other clusters. This cluster is made up of highly educated clients, so the company may consider offering educational resources such as webinars or workshops to help them make informed decisions about their insurance needs, perhaps, making them diversify their insurance investments.

- Cluster 3 - **Average**: This cluster includes 3142 observations and is composed of average clients who are middle-aged and earn an average salary. These clients are the highest spenders on *Health* insurance and have ordinary spending on the other types. It includes customers from all education levels and has a relatively balanced number of people with and without children. This cluster is composed of clients with diverse needs, so the company may consider offering special packages that allow them to purchase multiple services at once.

  All the clusters presented the same distribution of people from the different living areas.

A 2D visualization of the final clusters using t-SNE can be found in Fig 9.4, as well as a summarizing bubble chart in Fig 9.5. This solution presents a silhouette of 0.11 and the silhouettes of each cluster can be seen in Fig 9.6. The high prevalence of negative silhouettes in cluster 3 is expected, given it is the most average group and may include similar observations to other more distinct clusters. Finally, we briefly looked up the most relevant features when assigning a customer to a cluster using a *DecisionTree*, and those included *PremMotor, EducDeg* and *MonthSal* [Table 9.2].

Through this analysis, we were able to identify several meaningful patterns and insights in our data. However, it is important to note that our research had some limitations, such as the short variety of information provided by the company. It is our hope that our findings will benefit A2Z insurance in their pursuit of improving and expanding their customer services.

10. References

- *Bubble Charts in Python*. Bubble charts in Python. (n.d.). Retrieved January 2, 2023, from https://plotly.com/python/bubble-charts/
- Anand, D. (2022). Gower's Distance [web log]. Retrieved January 5, 2023, from https://medium.com/analytics-vidhya/gowers-distance-899f9c4bd553.
- Bücker, T. (2016). *Customer Clustering in the Insurance Sector by Means of Unsupervised Machine Learning* (dissertation).
- Deore, S. (2020, December 31). *Really, what is Hopkins statistic?* Medium. Retrieved December 14, 2022, from https://sushildeore99.medium.com/really-what-is-hopkins-statistic-bad1265df4b
- K-means clustering is not a free lunch was published on January 16, 2015. (n.d.). *K-means clustering is not a free lunch*. Variance Explained. Retrieved October 26, 2022, from http://varianceexplained.org/r/kmeans-free-lunch/
- Shendre, S. (2020, May 11). *Clustering datasets having both numerical and categorical variables*. Medium. Retrieved January 5, 2023, from https://towardsdatascience.com/clustering-datasets-having-both-numerical-and-categorical-variables-ed91cdca0677
- t-SNE and UMAP projections in Python . (n.d.). Retrieved December 27, 2022, from https://plotly.com/python/t-sne-and-umap-projections/
- Vadali, S. G. (2017, December 29). *Day 8: Data transformation - skewness, normalization and much more*. Medium. Retrieved December 4, 2022, from https://medium.com/@TheDataGyan/day-8-data-transformation-skewness-normalization-and-much-more-4c144d370e55
- Ziafat, H., & Shakeri, M. (2014). Using Data Mining Techniques in Customer Segmentation. Hasan Ziafat Int. Journal of Engineering Research and Applications.

11. Appendix

11.1. Gower Distance

The Gower distance (Gower 1971) is a similarity measure generally used in cases where the objects being compared are of mixed data types, such as a combination of continuous and categorical variables. For each feature $k = 1, \ldots, p$, we define a score $s_{ijk} \in [0, 1]$. If x_i and x_j are close to each other along k, then the score is close to 1. On the other hand, if they are far away, the score is close to 0. The metrics used for each data type are described below:

- Quantitative (interval): range-normalized Manhattan distance
- Ordinal: variable ranked and Manhattan distance is used with a special adjustment for ties
- Nominal: variables of *k* categories are first converted into *k* binary columns and then the Dice coefficient is used

Gower's distance is the average of partial dissimilarities across individuals and is defined as:

$$d(x_1, x_2) = 1 - \left(\frac{1}{p} * \sum_{j=1}^{p} s_{ij}(x_1, x_2)\right)$$

and for the numerical features:

$$s_{ij}(x_1, x_2) = |y_{1j} - y_{2j}| / R_j$$

For qualitative features, the Dice coefficient is calculated. More information can be found here.

11.2. KNNModeImputer

```
class KNNModeImputer(BaseEstimator, TransformerMixin):
   def __init__(self, n_neighbors = 5, metric='euclidean', feat_to_impute = None , feats_nn = None):
       self.n_neighbors = n_neighbors
       self.metric = metric
       self.feats_nn = feats_nn
       self.feat_to_impute = feat_to_impute

   def fit(self, X, y=None):
       self.X_ = X
       self.is_fitted_ = True
       return self

   def transform(self, X):
     check_is_fitted(self, 'is_fitted_')
     tree = KDTree(self.X_[self.feats_nn], metric=self.metric) # create KDTtree
     imputed_X = X.copy() # copy of the dataframe where the missing values will be filled
     missing_rows = X[X[self.feat_to_impute].isna()]
     # find K nearest neighbors
     distances, indices = tree.query(missing_rows[self.feats_nn].values, k= self.n_neighbors)
     missing_idx = np.array(missing_rows.index)
     for m_idx, n_idx in enumerate(indices):   # fill the missing values using the mode of the KNN
       k_neighbors = self.X_.loc[n_idx, self.feat_to_impute]
       mode = Counter(k_neighbors).most_common(1)[0][0] # mode
       imputed_X.loc[missing_idx[m_idx], self.feat_to_impute] = mode
     return imputed_X
```

Figures and Tables

	CustId	FirstPolYear	BirthYear	MonthSal	CustMonVal	ClaimsRate
min	1.0	1974.0	1028.0	333.0	-165680.42	0.0
std	2972.3	511.27	19.71	1157.45	1945.81	2.912
25%	2574.8	1980.0	1953.0	1706.0	-9.44	0.39
50%	5148.5	1986.0	1968.0	2501.5	186.87	0.72
mean	5148.5	1991.06	1968.01	2506.67	177.89	0.743
75%	7722.3	1992.0	1983.0	3290.25	399.78	0.98
count	10296.0	10266.0	10279.0	10260.0	10296.0	10296.0
max	12096.0	53784.0	2001.0	55215.0	11875.89	256.2

	PremMotor	PremHousehold	PremHealth	PremLife	PremWork
min	-4.11	-75.0	-2.11	-7.0	-12.0
std	211.91	352.60	296.41	47.48	51.51
25%	190.59	49.45	111.8	9.89	10.67
50%	298.61	132.8	162.81	25.56	25.67
mean	300.47	210.43	171.58	41.86	41.28
75%	408.3	290.05	219.82	57.79	56.79
count	10262.0	10296.0	10253.0	10192.0	10210.0
max	11604.42	25048.8	28272.0	398.3	1988.7

Table 2.1

EducDeg	Freq	GeoLivArea	Freq	Children	Freq
b'1 - Basic'	1272	1	3048	0	7262
b'2 - High School'	3510	2	1036	1	3013
b'3 - BSc/MSc'	4799	3	2066		
b'4 - PhD'	698	4	4145		

Table 2.2

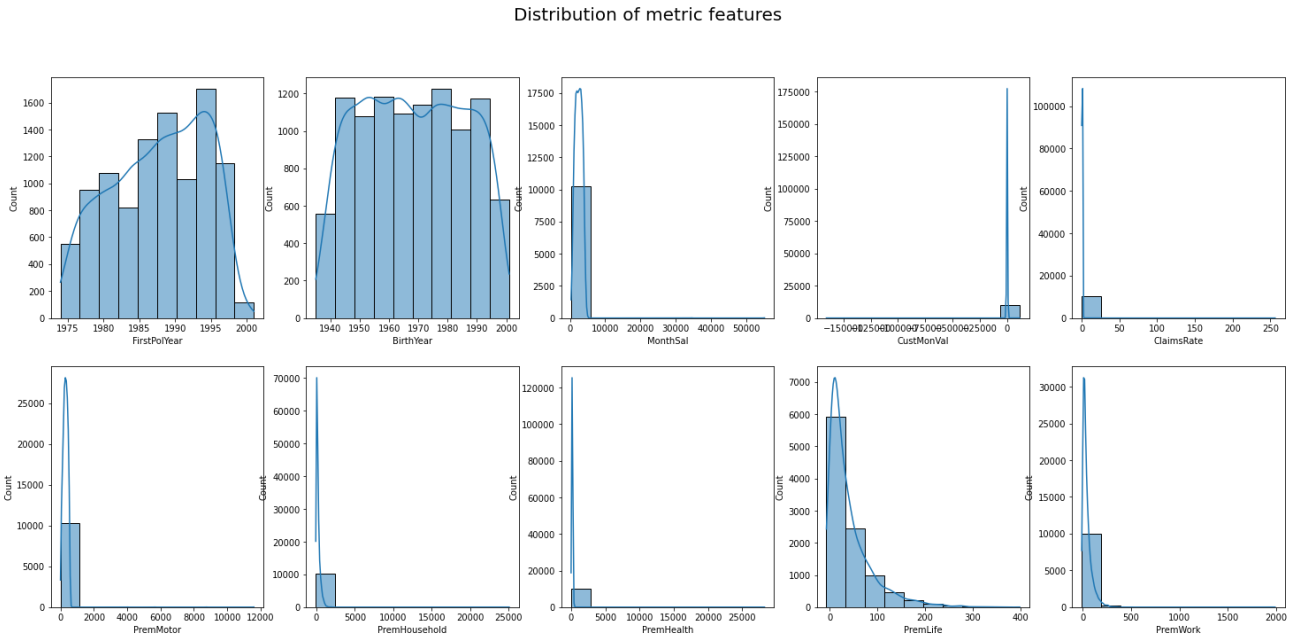


Fig 2.4.1

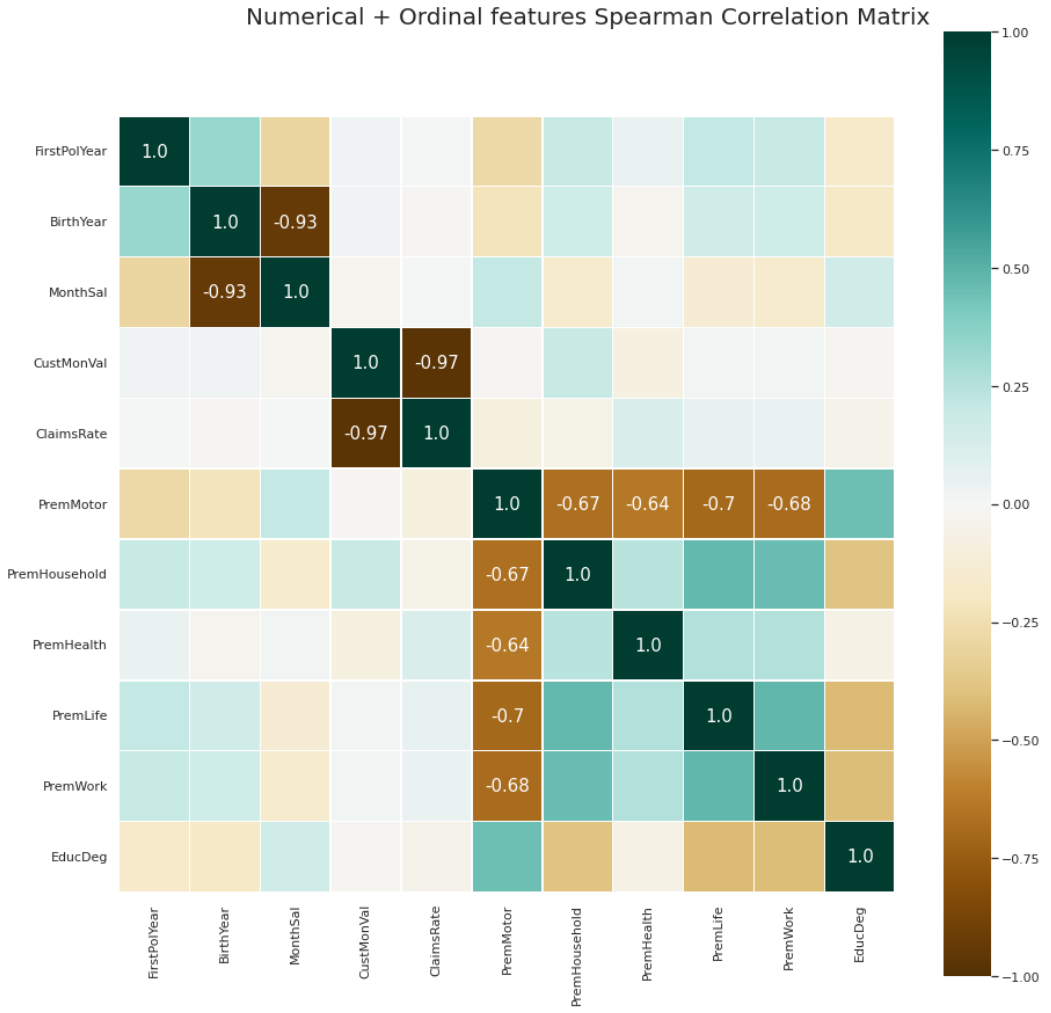


Fig 2.4.2

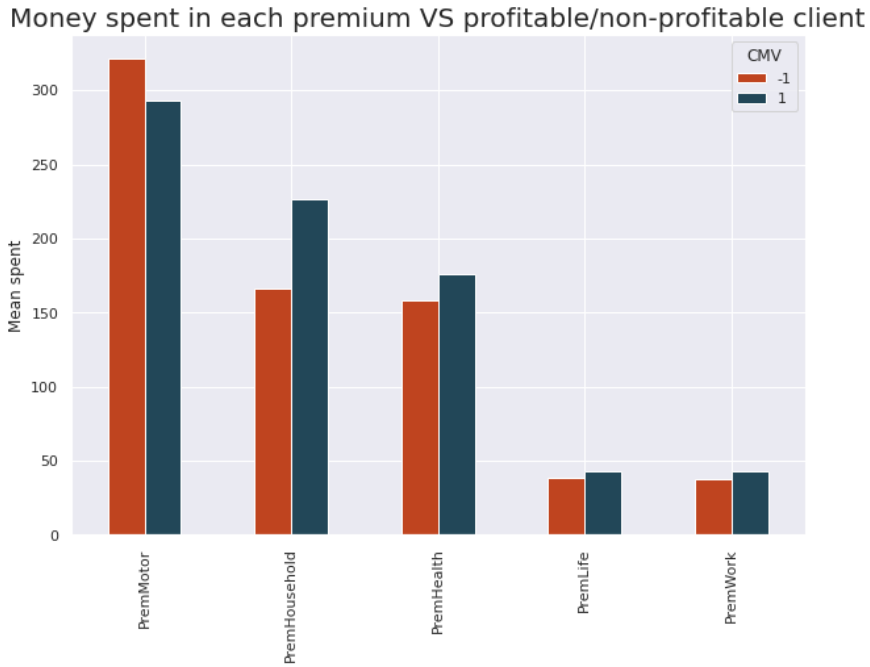


Fig 2.4.3

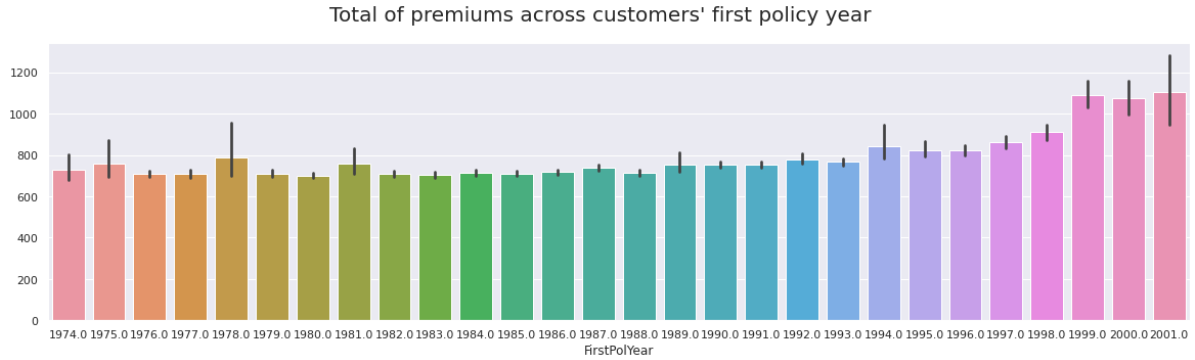


Fig 2.4.4

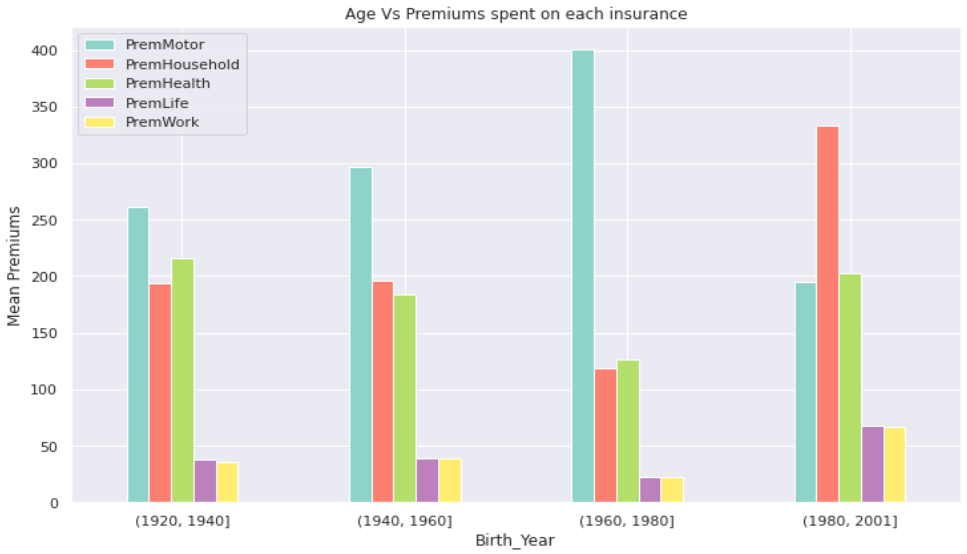


Fig 2.4.5

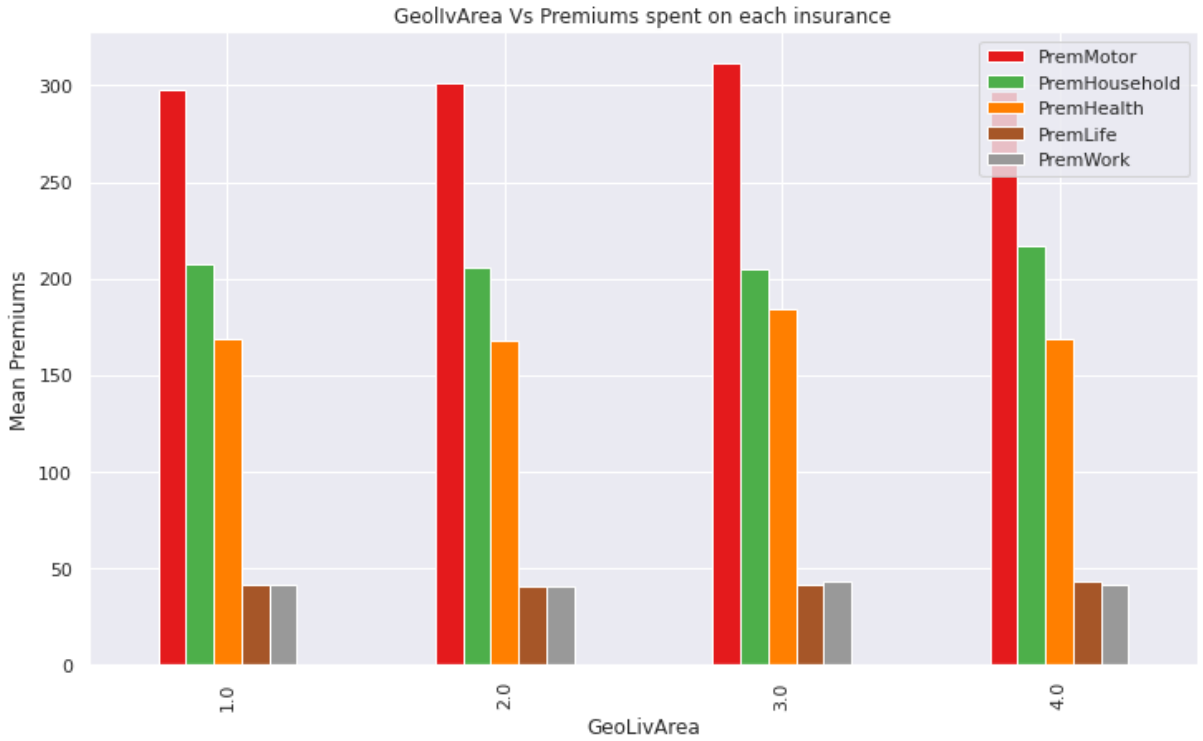


Fig 2.4.6

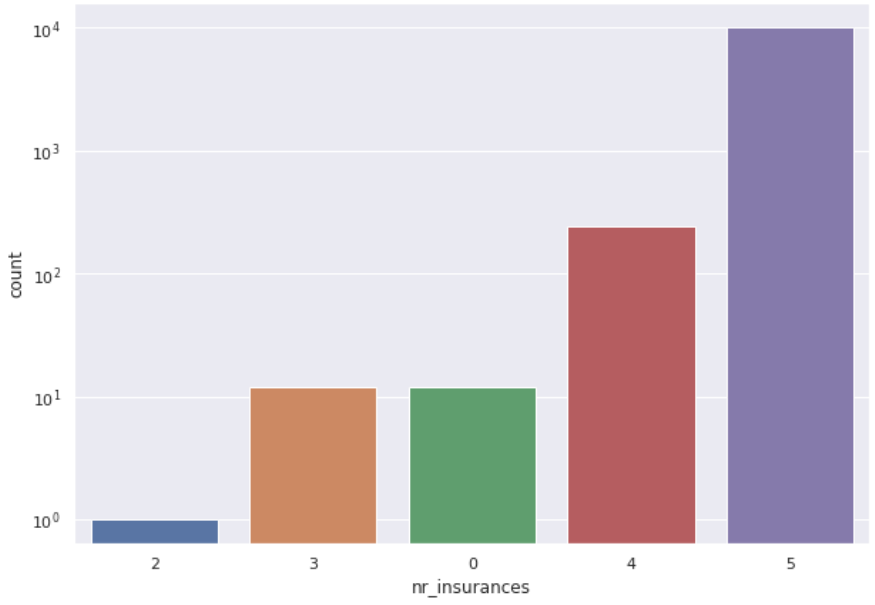


Fig 2.4.7

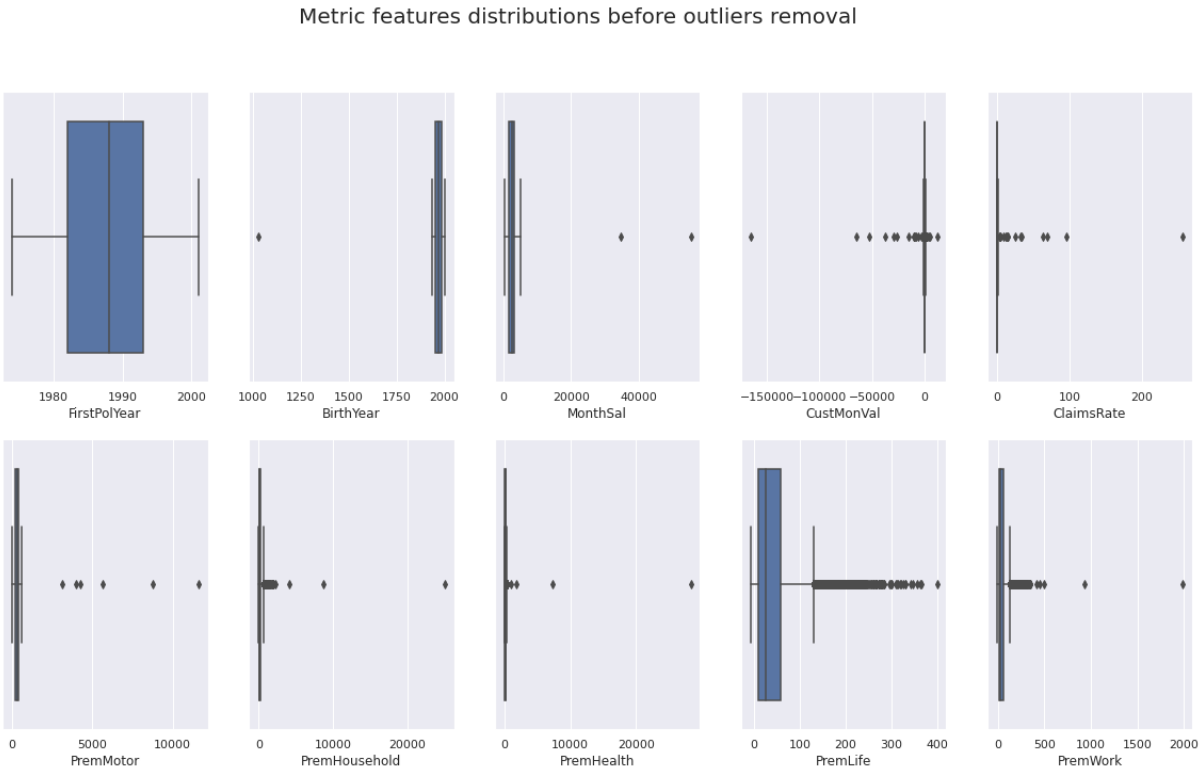


Fig 3.1.1

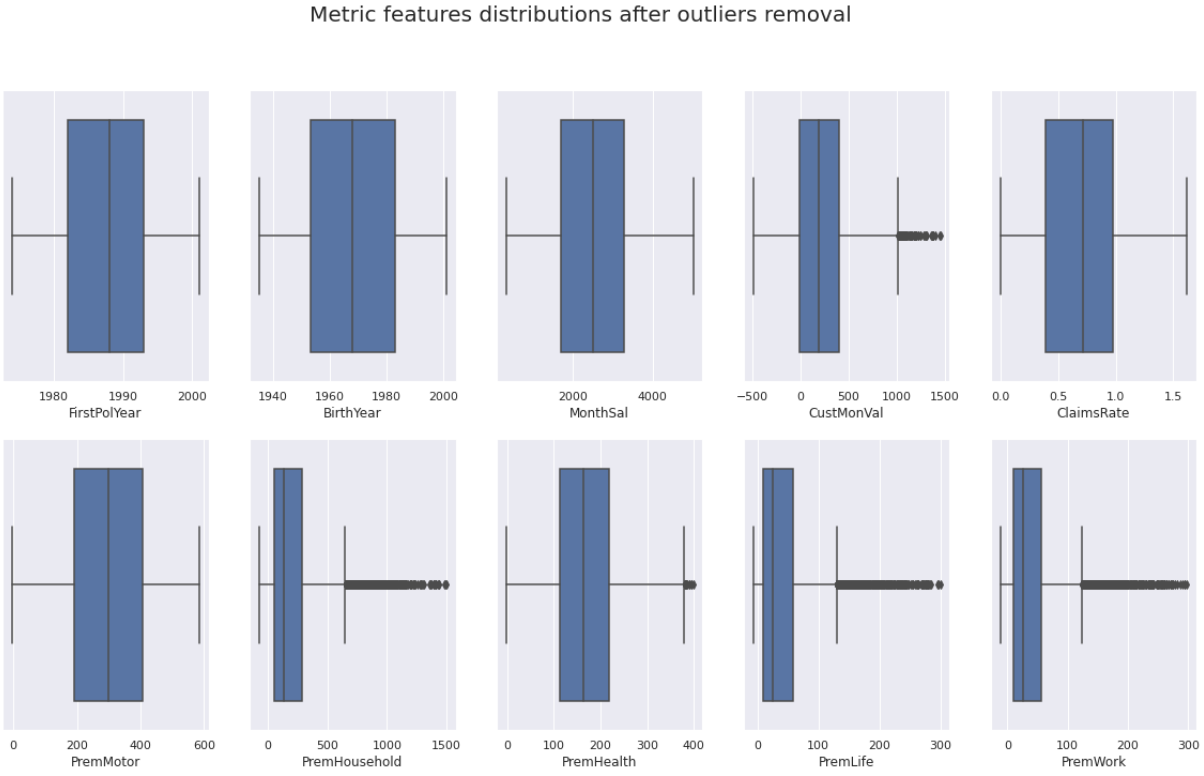


Fig 3.1.2

Feature	Description
total_reversals	Absolute value of the total amount of negative premiums Represents the value that the company owes the customer
total_premiums	Total amount of premiums
MotorPercent	Percentage of the total amount of premiums spent in *PremMotor*
HouseholdPercent	Percentage of the total amount of premiums spent in *PremHousehold*
HealthPercent	Percentage of the total amount of premiums spent in *PremHealth*
LifePercent	Percentage of the total amount of premiums spent in *PremLife*
WorkPercent	Percentage of the total amount of premiums spent in *PremWork*

Table 3.2.1

Feature	Original skewness value	Transformation	Resulting skewness value
total_premiums	1.5015	$x^{1/1.2}$	1.3270
PremHousehold	1.7516	$\sqrt[3]{x}$	0.3021
HouseholdPercent	0.7170	$\sqrt[3]{x}$	-0.4049
PremLife	1.9311	$\sqrt[3]{x}$	0.3254
LifePercent	2.0976	$\sqrt[3]{x}$	0.2317
PremWork	1.9291	$\sqrt[3]{x}$	0.3490
WorkPercent	2.1438	$\sqrt[2]{x}$	0.7957

Table 3.3.1

Feature	Number of missing values
GeoLivArea	1
BirthYear	17
FirstPolYear	29
MonthSal	36
Children	20
EducDeg	17

Table 3.5.1

Pairplot of metric features

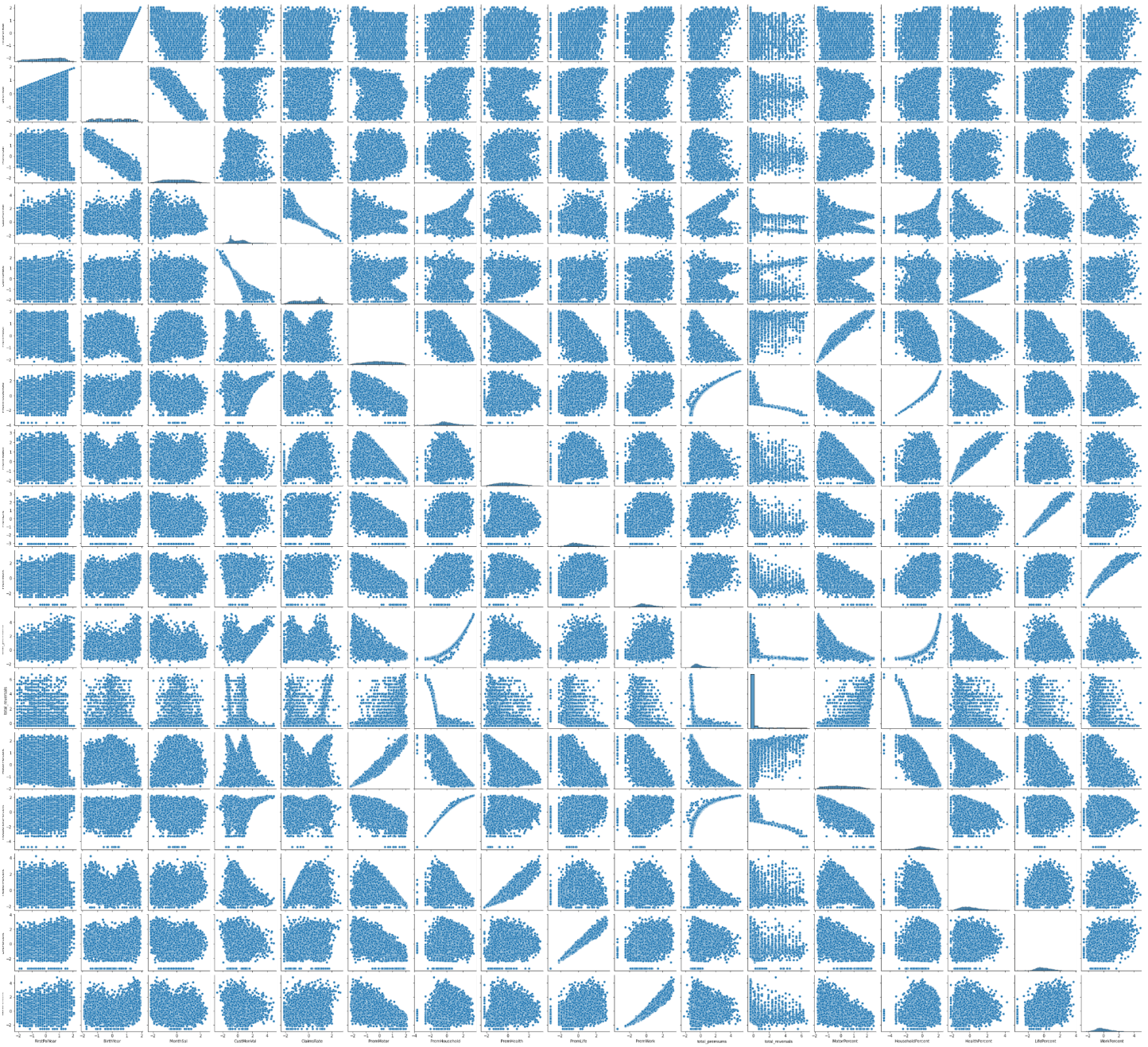


Fig 5.1.1

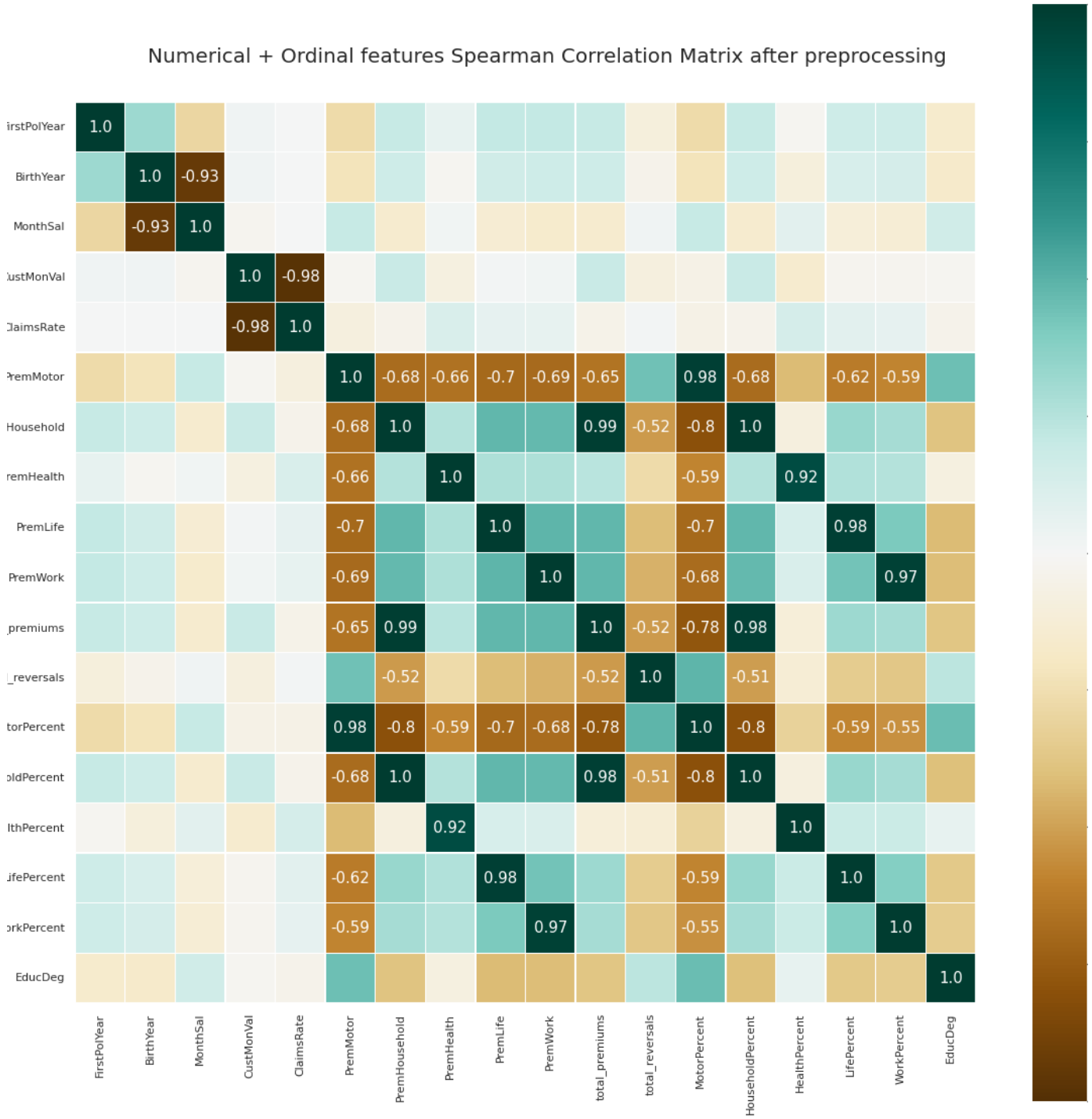

Numerical + Ordinal features Spearman Correlation Matrix after preprocessing
irstPolYear 1.0
BirthYear 1.0 -0.93
MonthSal -0.93 1.0
ustMonVal 1.0 -0.98
ClaimsRate -0.98 1.0
PremMotor 1.0 -0.68 -0.66 -0.7 -0.69 -0.65 0.98 -0.68 -0.62 -0.59
Household -0.68 1.0 0.99 -0.52 -0.8 1.0
remHealth -0.66 1.0 -0.59 0.92
PremLife -0.7 1.0 -0.7 0.98
PremWork -0.69 1.0 -0.68 0.97
premiums -0.65 0.99 1.0 -0.52 -0.78 0.98
_reversals -0.52 -0.52 1.0 -0.51
torPercent 0.98 -0.8 -0.59 -0.7 -0.68 -0.78 1.0 -0.8 -0.59 -0.55
oldPercent -0.68 1.0 0.98 -0.51 -0.8 1.0
lthPercent 0.92 1.0
ifePercent -0.62 0.98 -0.59 1.0
orkPercent -0.59 0.97 -0.55 1.0
EducDeg 1.0
FirstPolYear
BirthYear
MonthSal
CustMonVal
ClaimsRate
PremMotor
PremHousehold
PremHealth
PremLife
PremWork
total_premiums
total_reversals
MotorPercent
HouseholdPercent
HealthPercent
LifePercent
WorkPercent
EducDeg

Fig 5.1.2

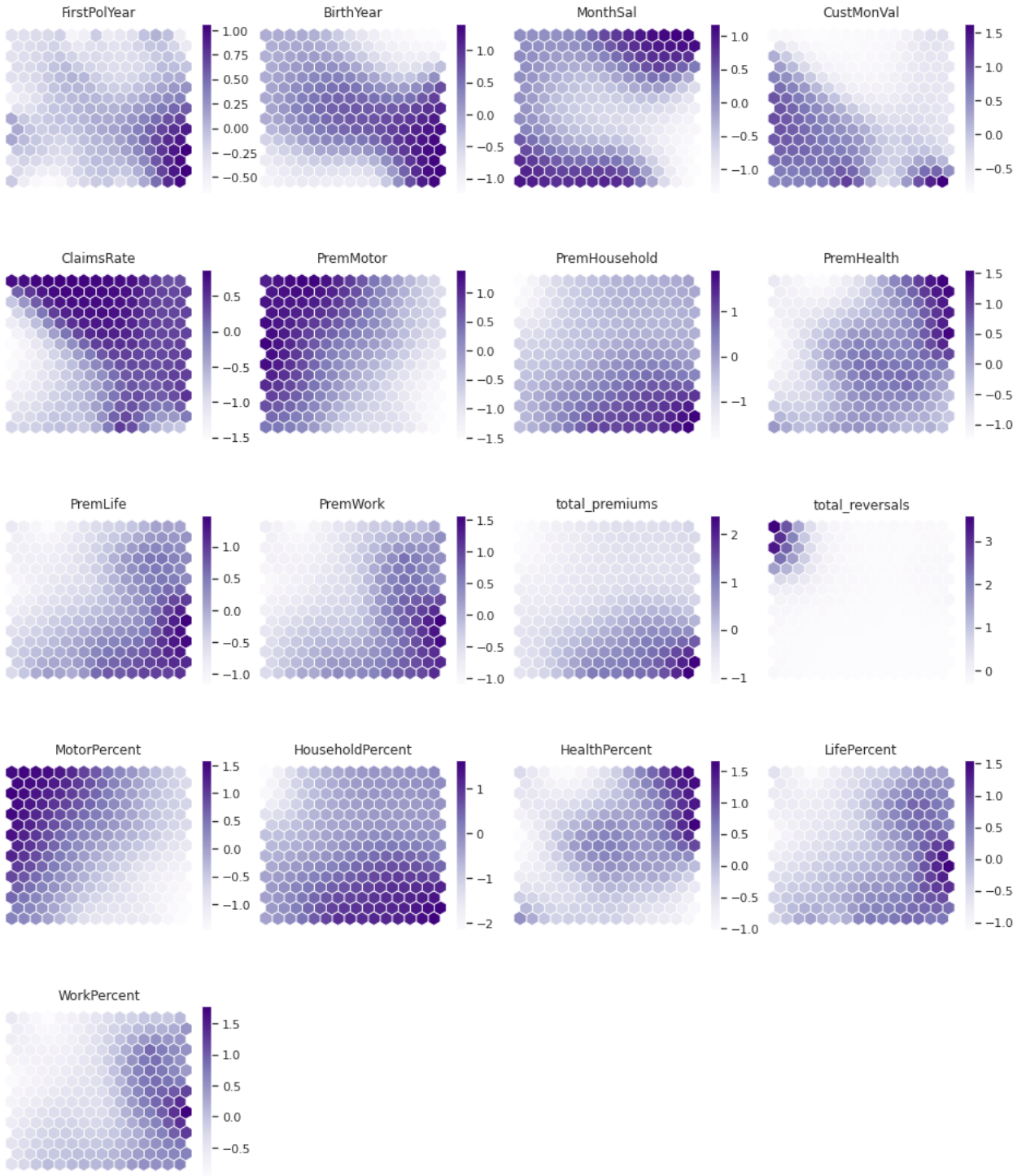

Component Planes
FirstPolYear
BirthYear
MonthSal
CustMonVal
ClaimsRate
PremMotor
PremHousehold
PremHealth
PremLife
PremWork
total_premiums
total_reversals
MotorPercent
HouseholdPercent
HealthPercent
LifePercent
WorkPercent

Fig 5.1.3

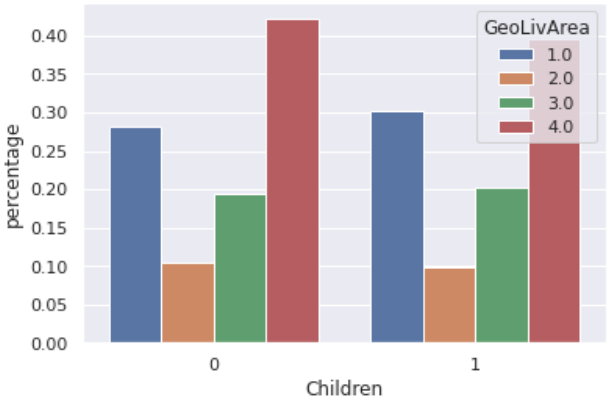


Fig 5.2.1

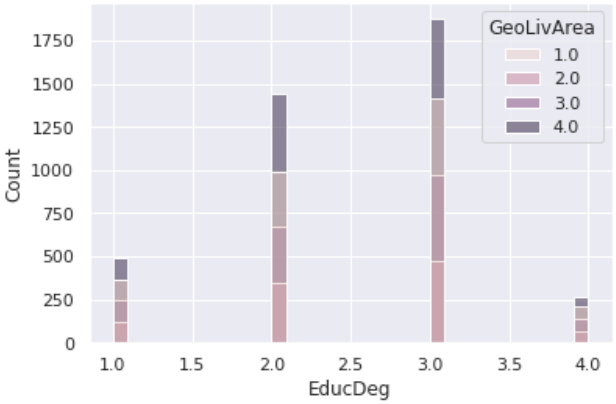


Fig 5.2.2

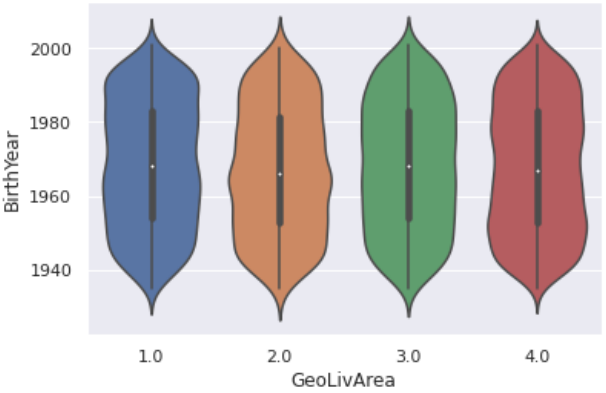


Fig 5.2.3

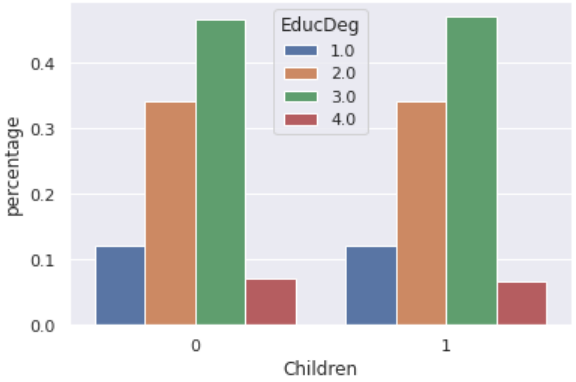


Fig 5.2.4

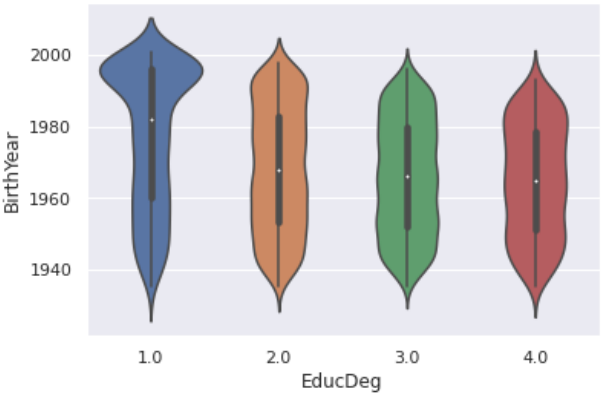


Fig 5.2.5

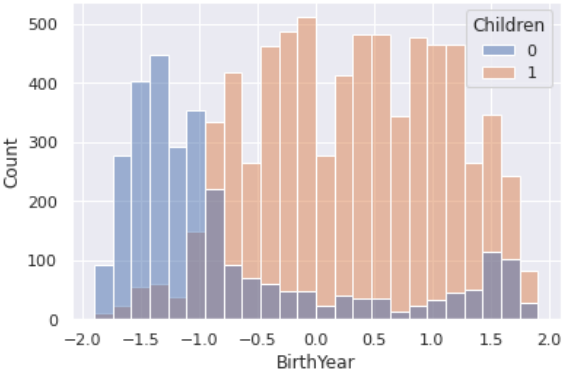


Fig 5.2.6

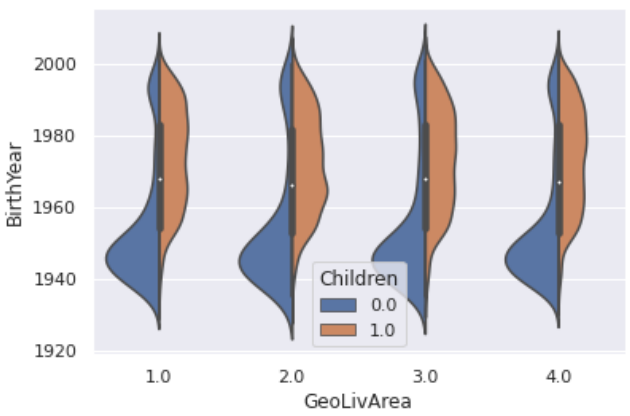


Fig 5.2.7

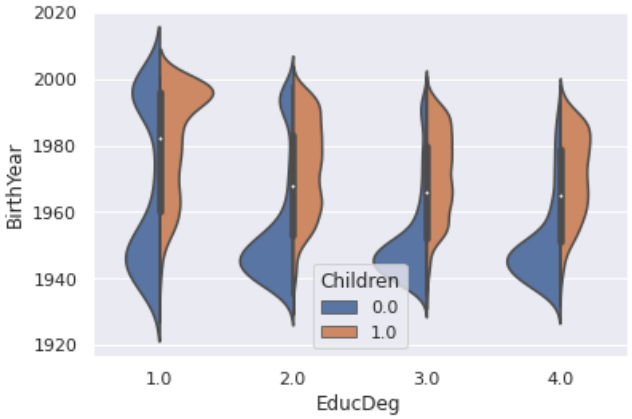


Fig 5.2.8.

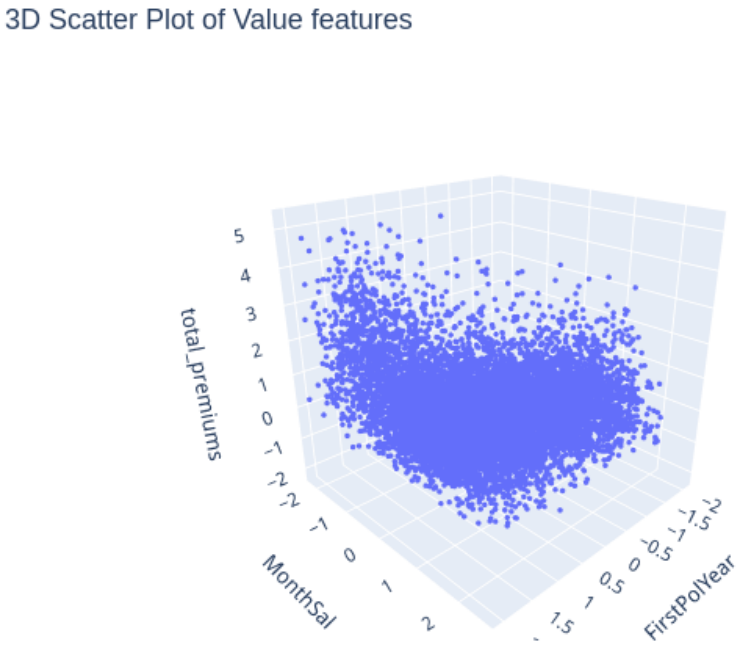


Fig 6.1

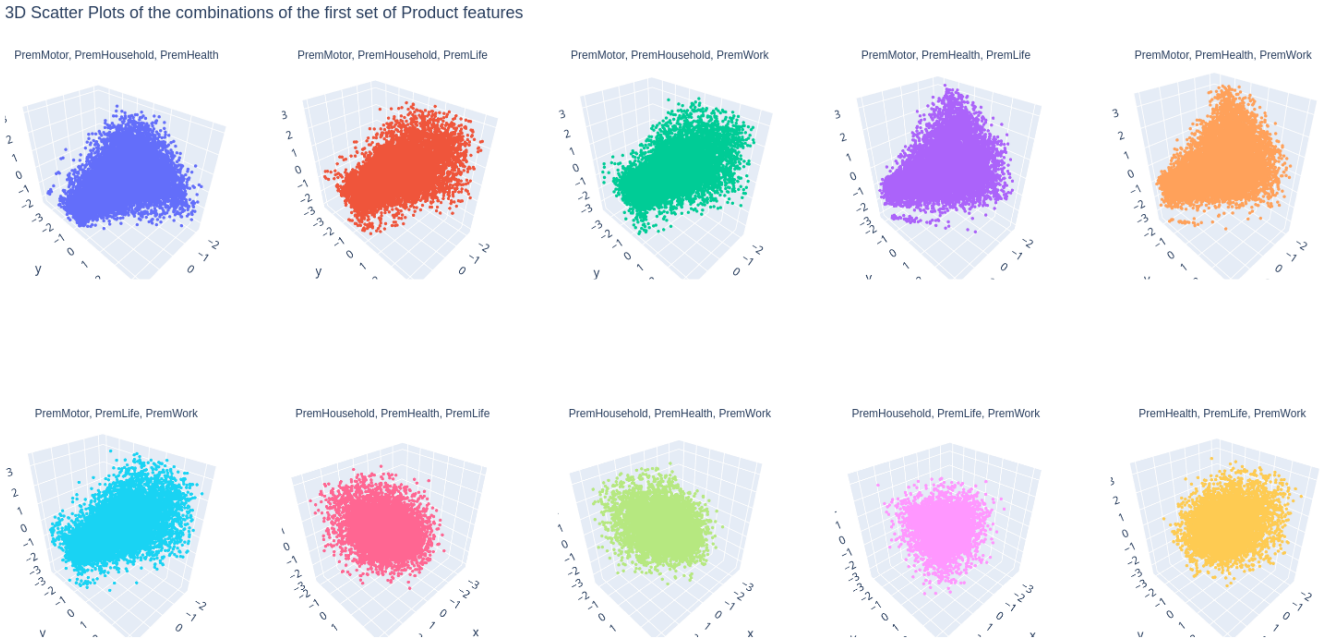

3D Scatter Plots of the combinations of the first set of Product features
PremMotor, PremHousehold, PremHealth
PremMotor, PremHousehold, PremLife
PremMotor, PremHousehold, PremWork
PremMotor, PremHealth, PremLife
PremMotor, PremHealth, PremWork
PremMotor, PremLife, PremWork
PremHousehold, PremHealth, PremLife
PremHousehold, PremHealth, PremWork
PremHousehold, PremLife, PremWork
PremHealth, PremLife, PremWork

Fig 6.2


3D Scatter Plots of the combinations of the second set of Product features
MotorPercent, HouseholdPercent, HealthPercent
MotorPercent, HouseholdPercent, LifePercent
MotorPercent, HouseholdPercent, WorkPercent
MotorPercent, HealthPercent, LifePercent
MotorPercent, HealthPercent, WorkPercent
MotorPercent, LifePercent, WorkPercent
HouseholdPercent, HealthPercent, LifePercent
HouseholdPercent, HealthPercent, WorkPercent
HouseholdPercent, LifePercent, WorkPercent
HealthPercent, LifePercent, WorkPercent

Fig 6.3

Pairwise distance matrix under a Socio-Demographic clustering perspective

Fig 6.4

Pairwise distance matrix under a Value clustering perspective

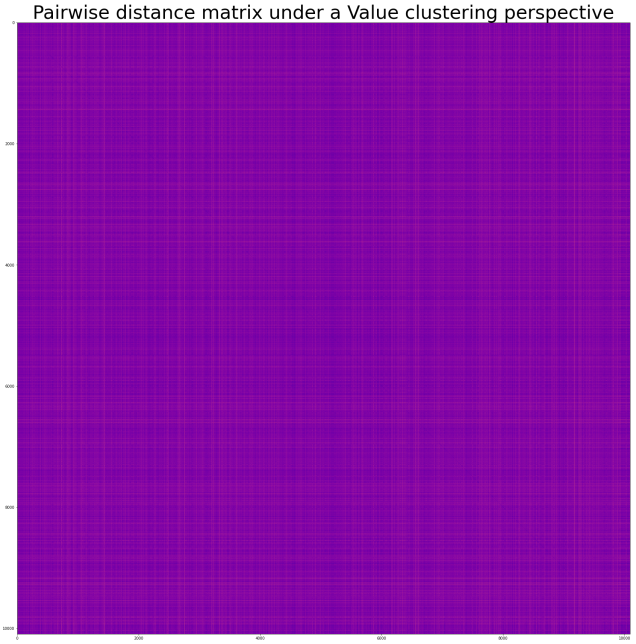


Fig 6.5

Pairwise distance matrix under a Product (total premiums) clustering perspective

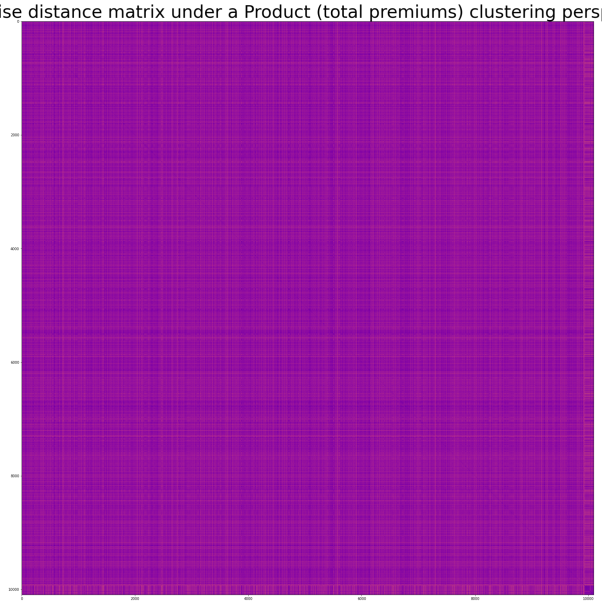


Fig 6.6

Pairwise distance matrix under a Product (premiums percentage) clustering perspective

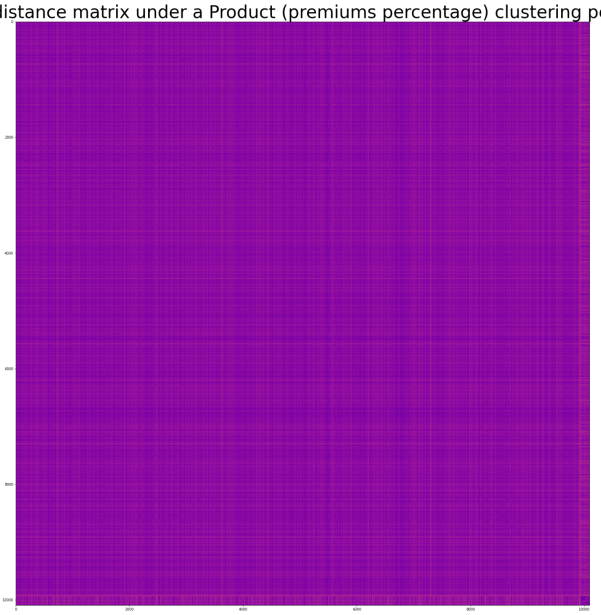


Fig 6.7

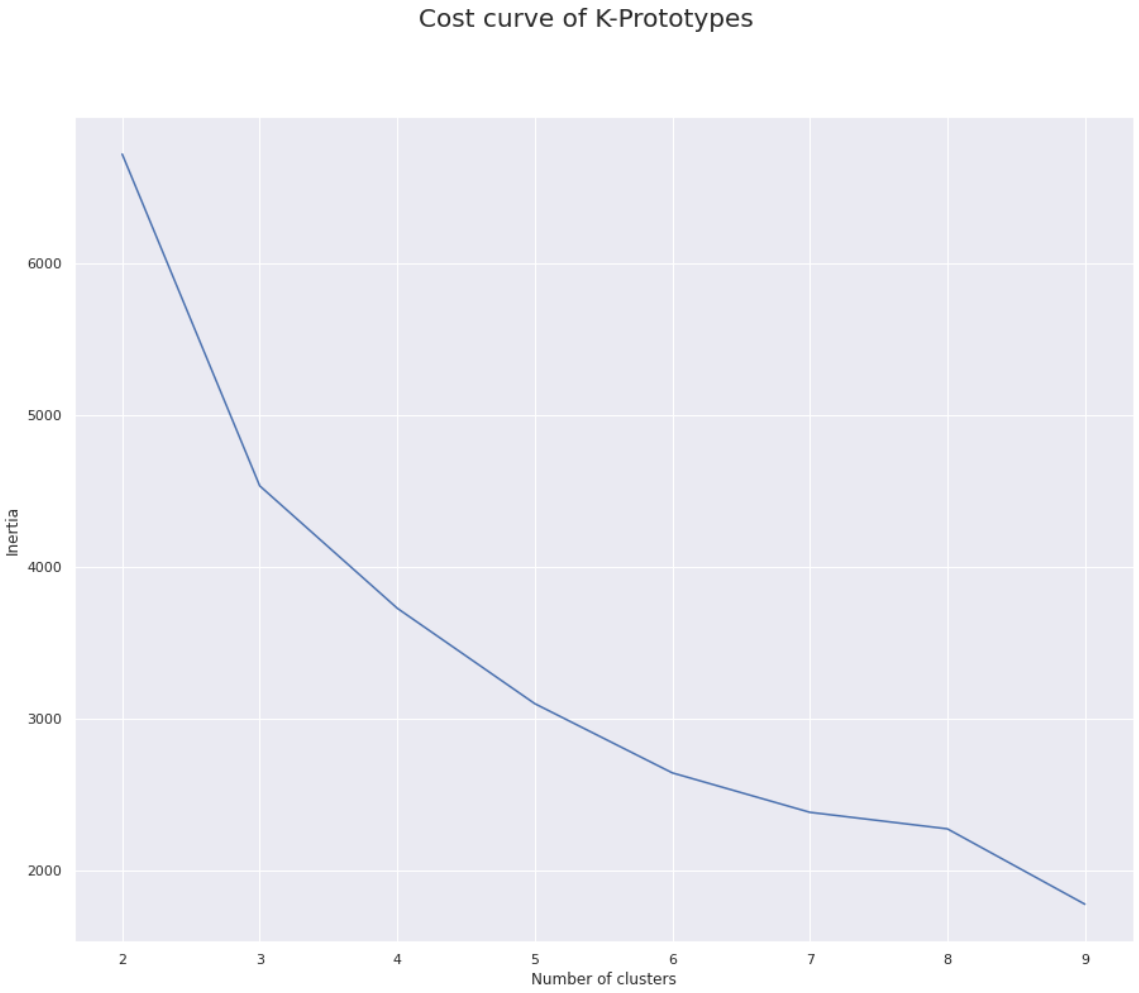


Fig 7.1.1

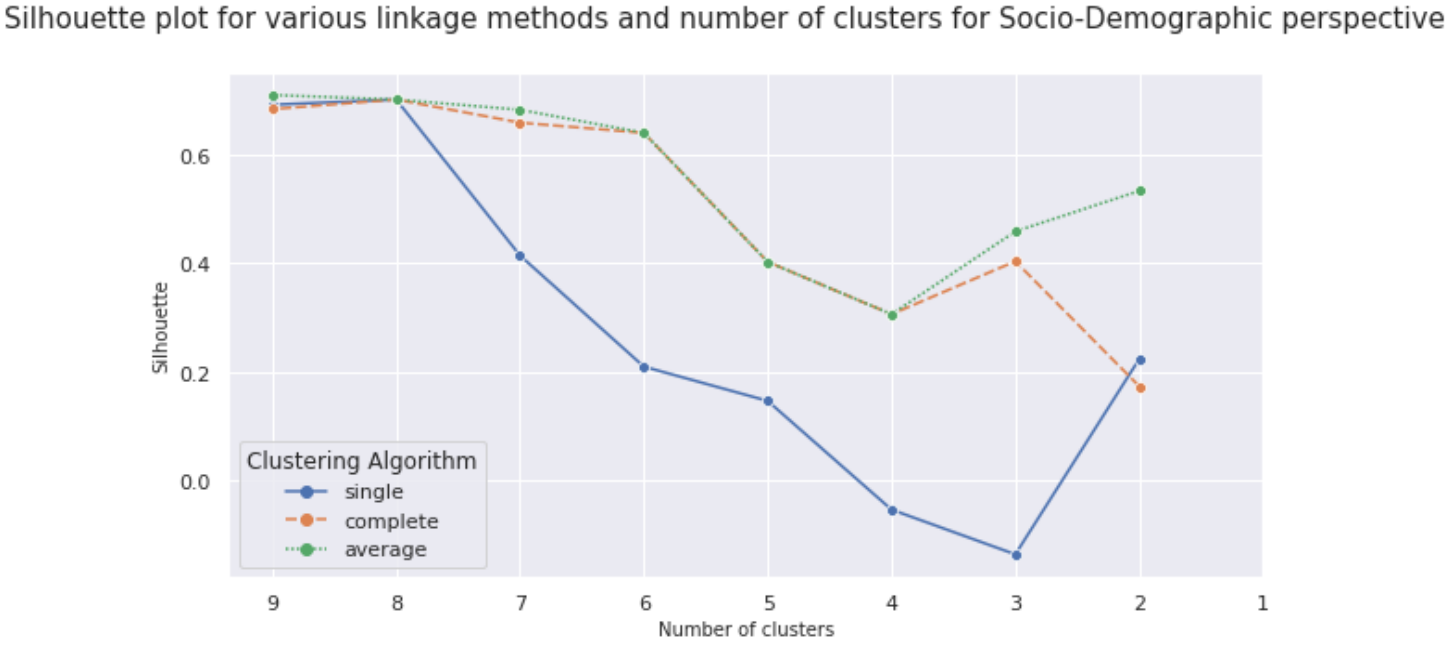


Fig 7.1.2

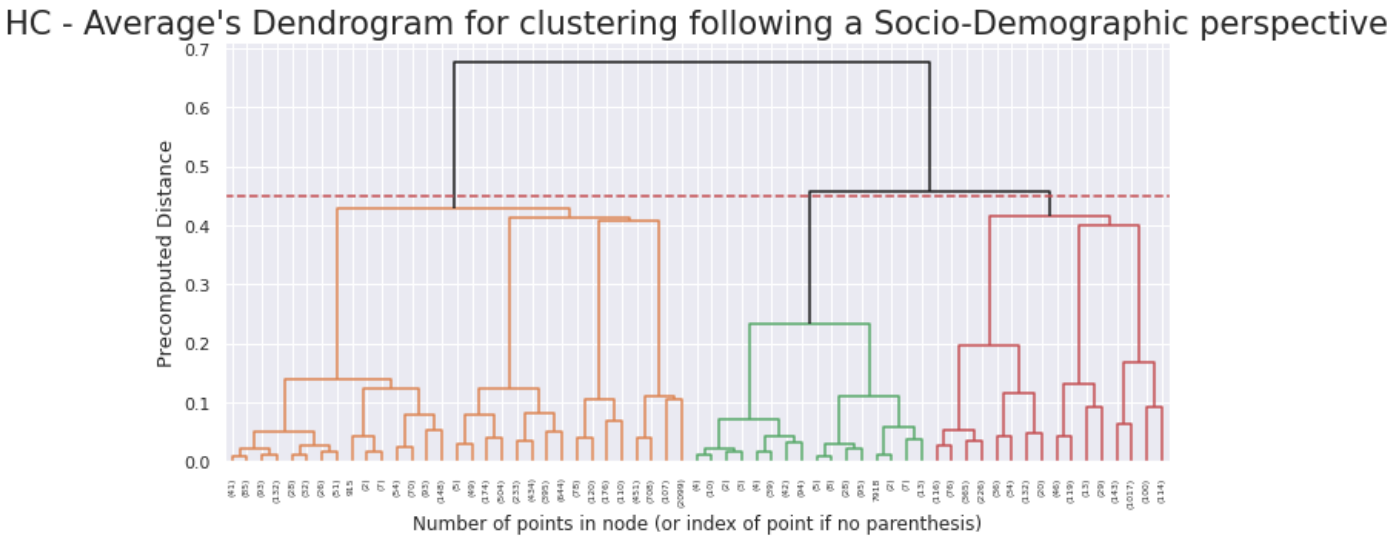


Fig 7.1.3

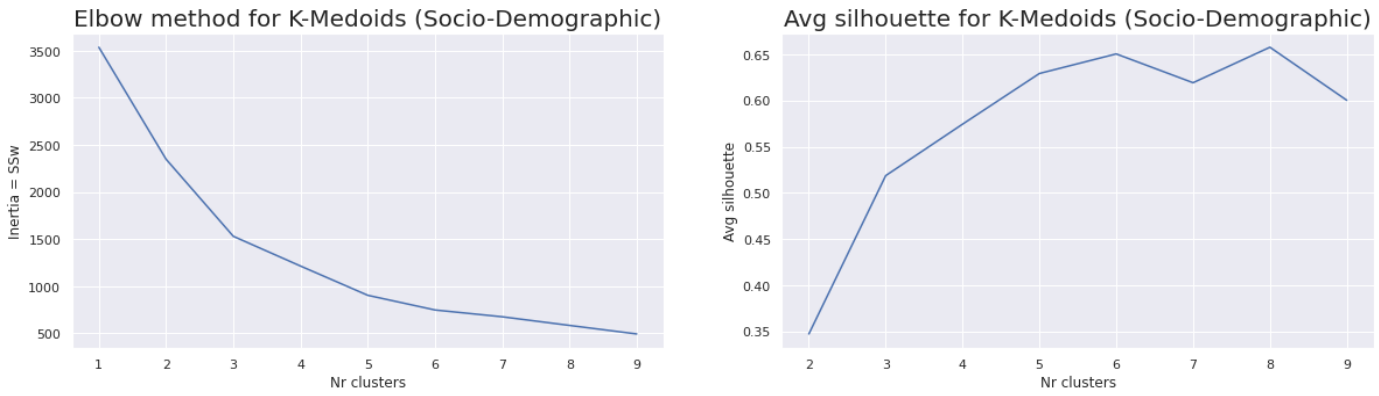


Fig 7.1.4

Algorithm	*Silhouette*
K-Prototypes	0.2844
Hierarchical clustering	0.4587
K-Medoids	0.6292

Table 7.1.1

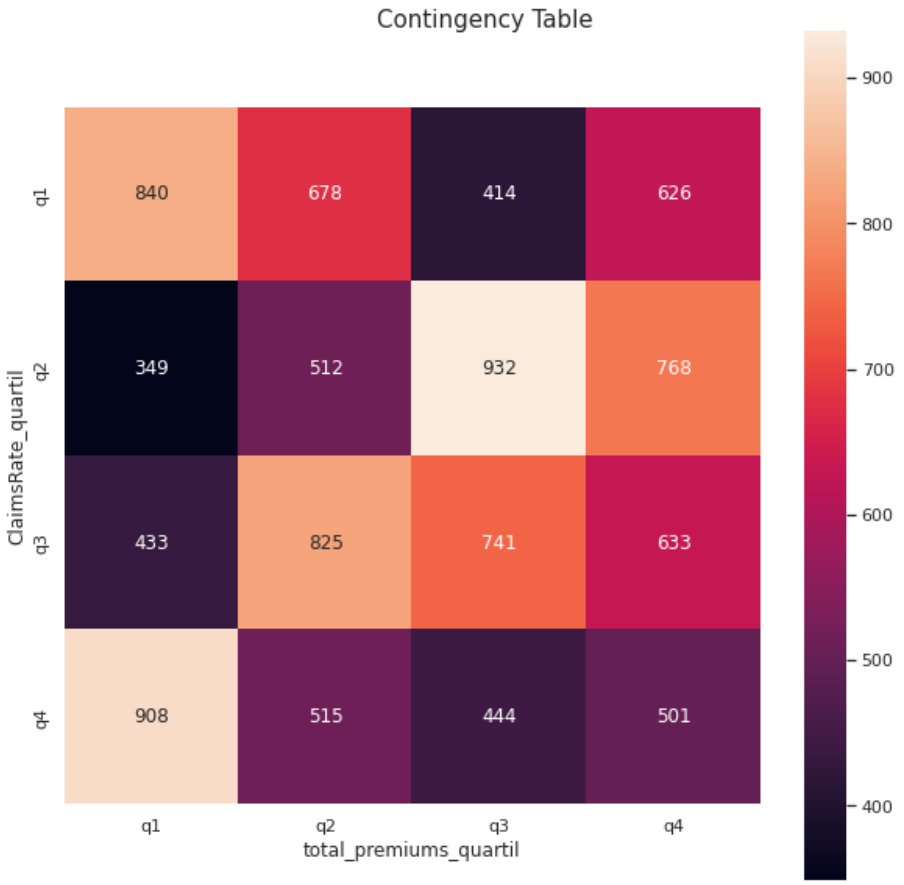


Fig 7.2.1

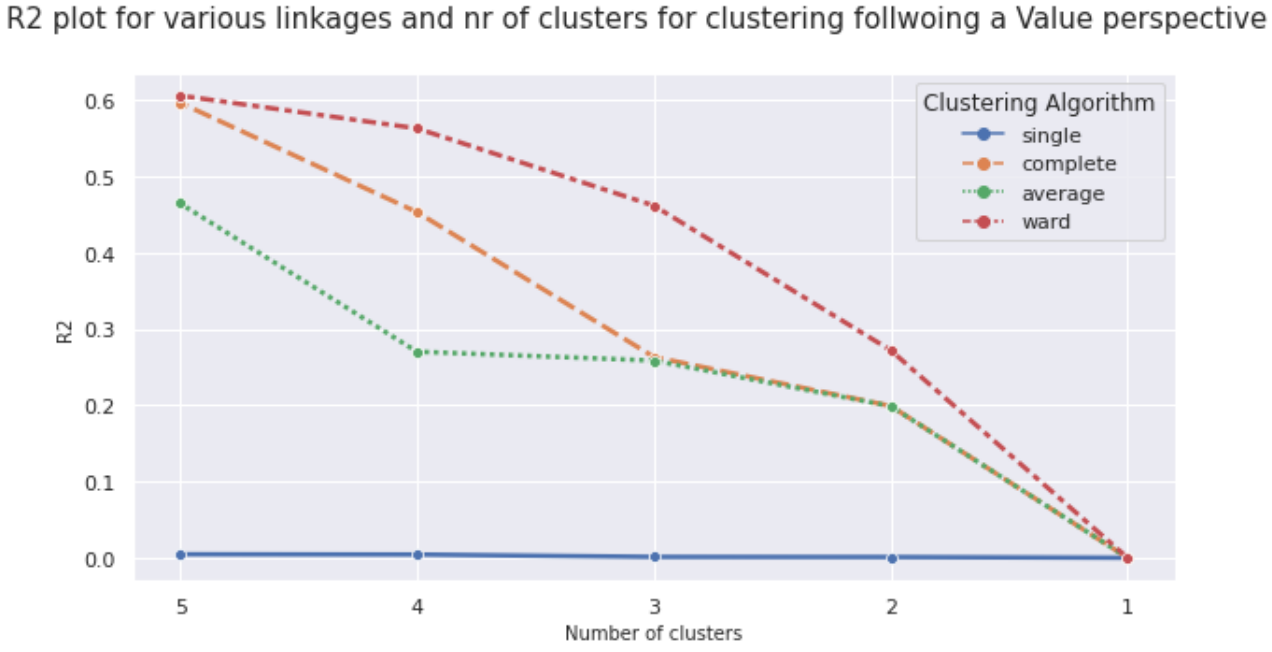


Fig 7.2.2

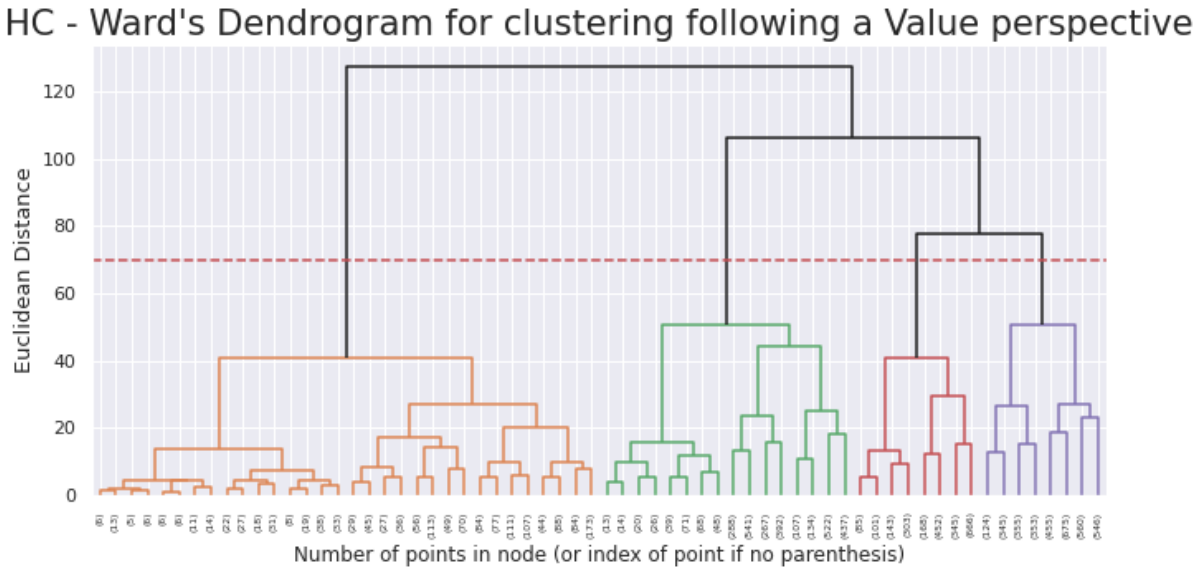


Fig 7.2.3

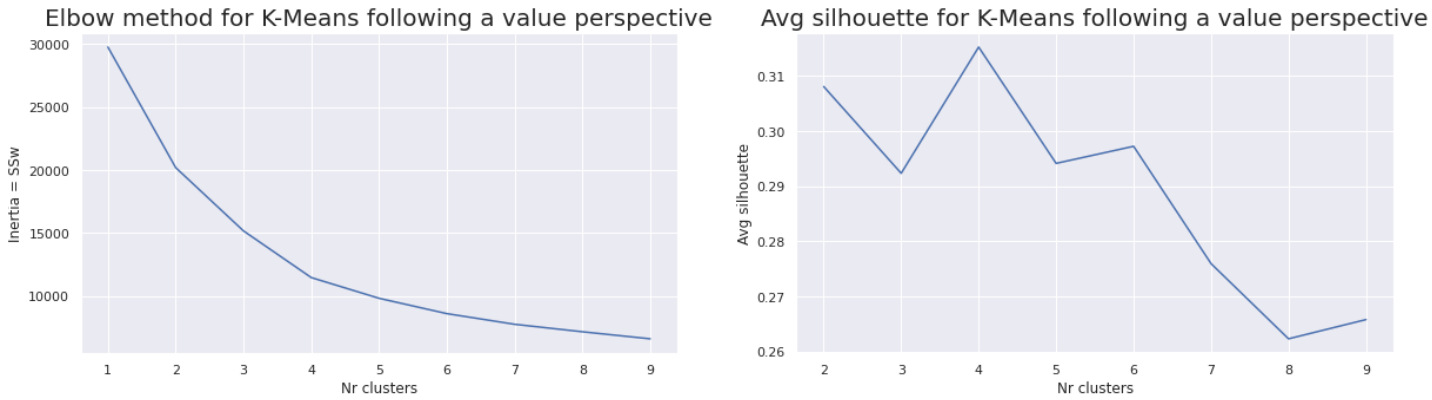


Fig 7.2.4

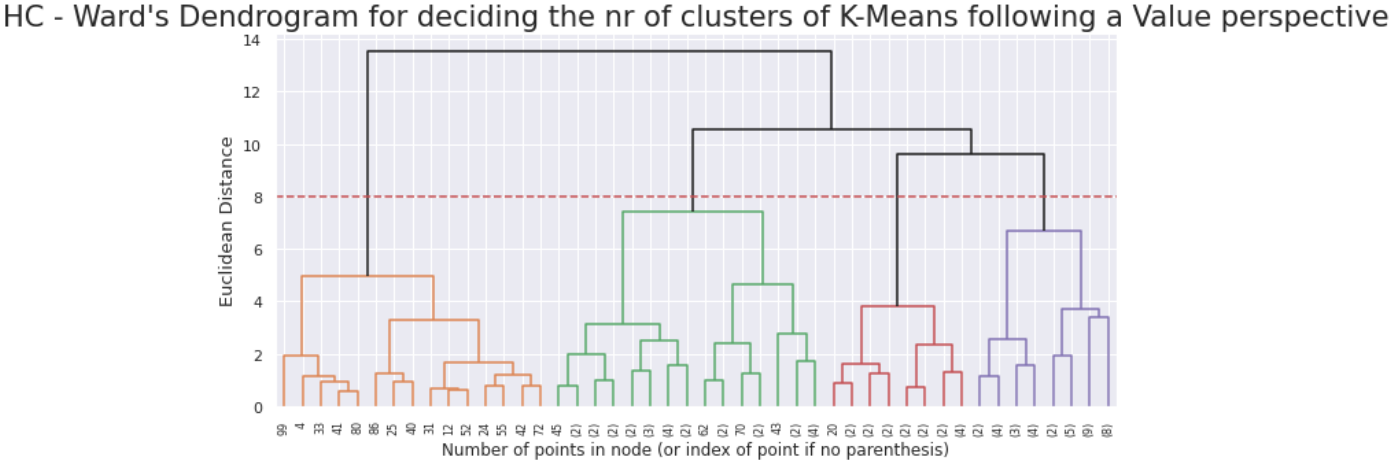


Fig 7.2.5

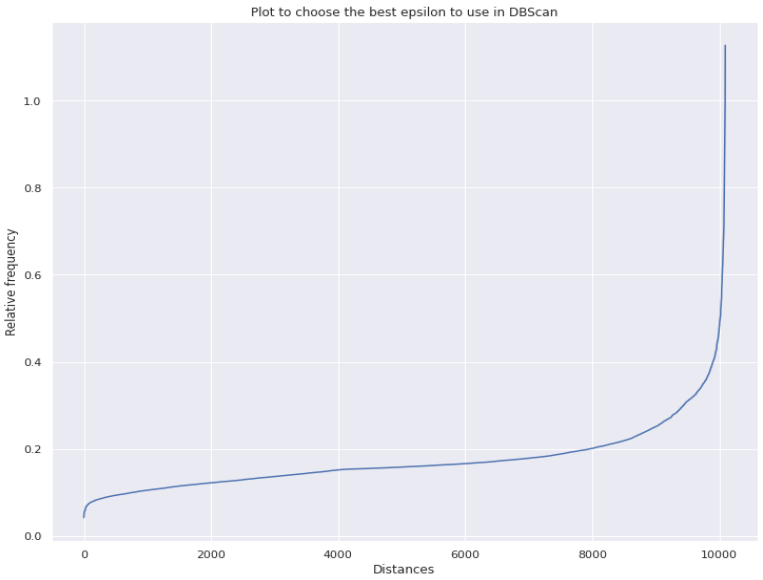


Fig 7.2.6

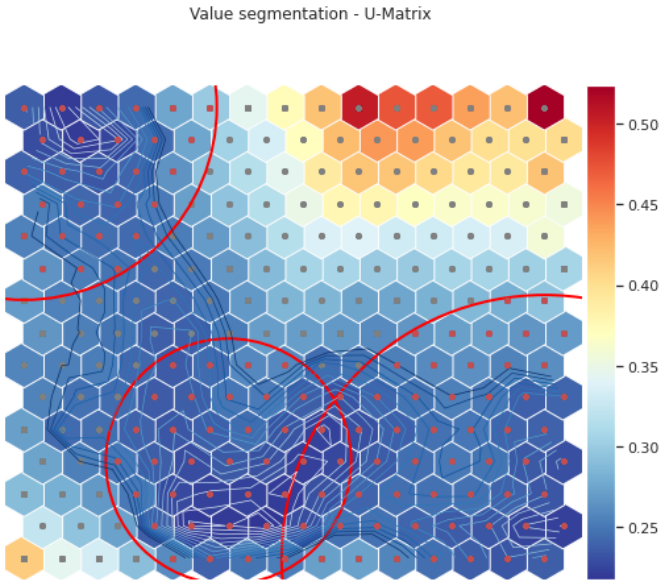


Fig 7.2.7

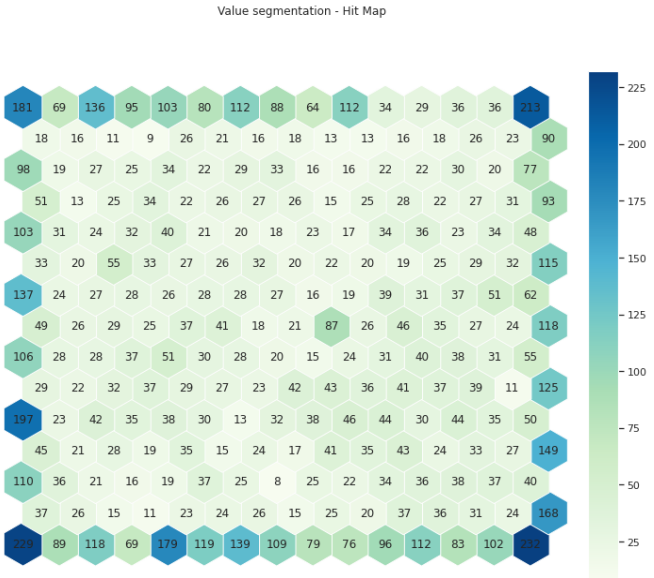


Fig 7.2.8

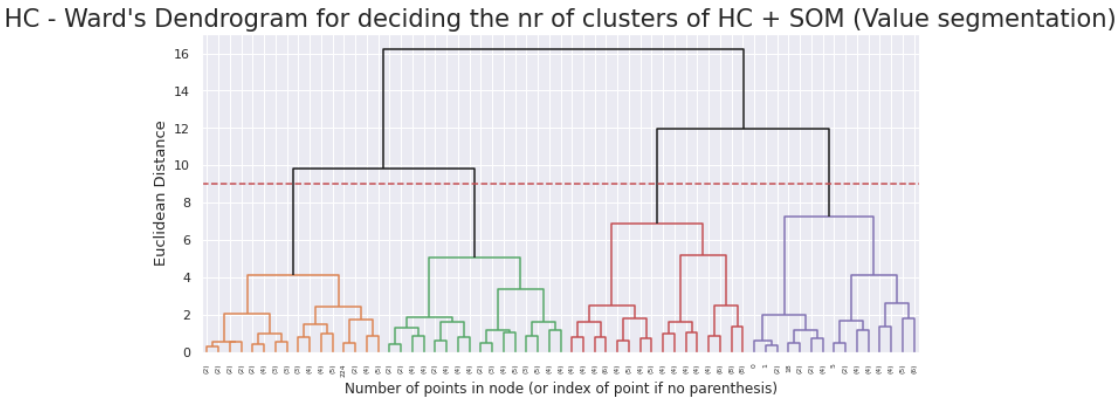


Fig 7.2.9

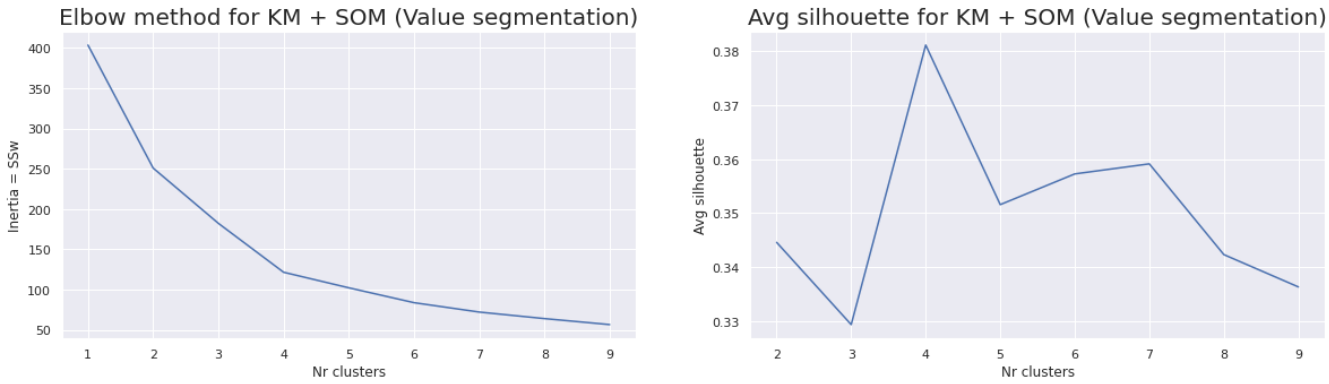


Fig 7.2.10

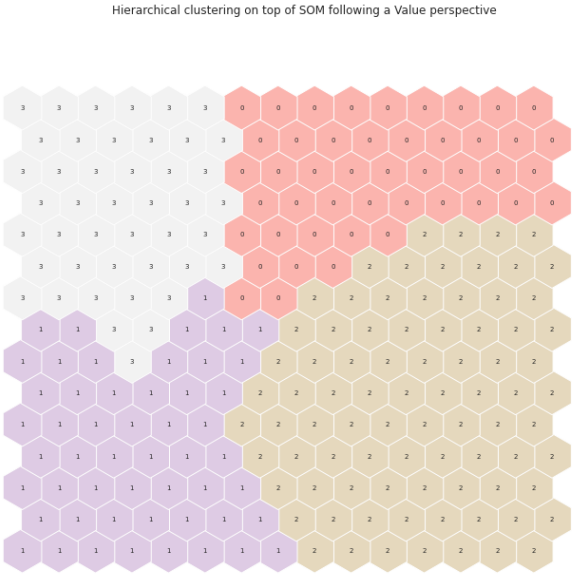


Fig 7.2.11

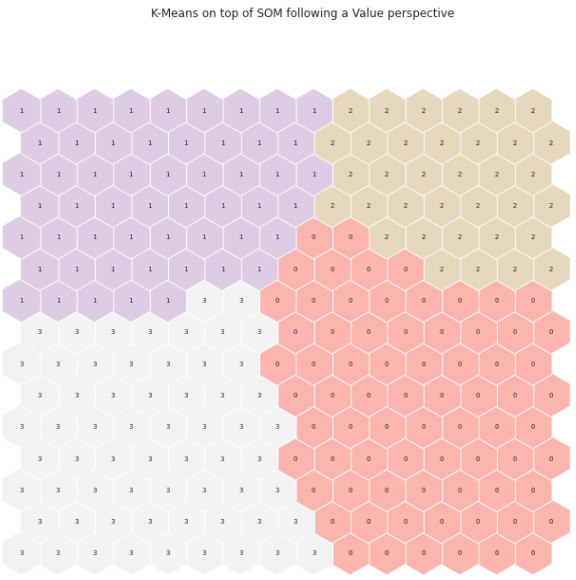


Fig 7.2.12

Algorithms for Value segmentation	*R2*	*Silhouette*
Hierarchical clustering	0.5630	0.25715
K-Means	0.6152	0.3152
DBScan	0.0446	0.1173
SOM + Hierarchical clustering	0.5410	0.2522
SOM + K-Means	0.6005	0.2968

Table 7.2.1

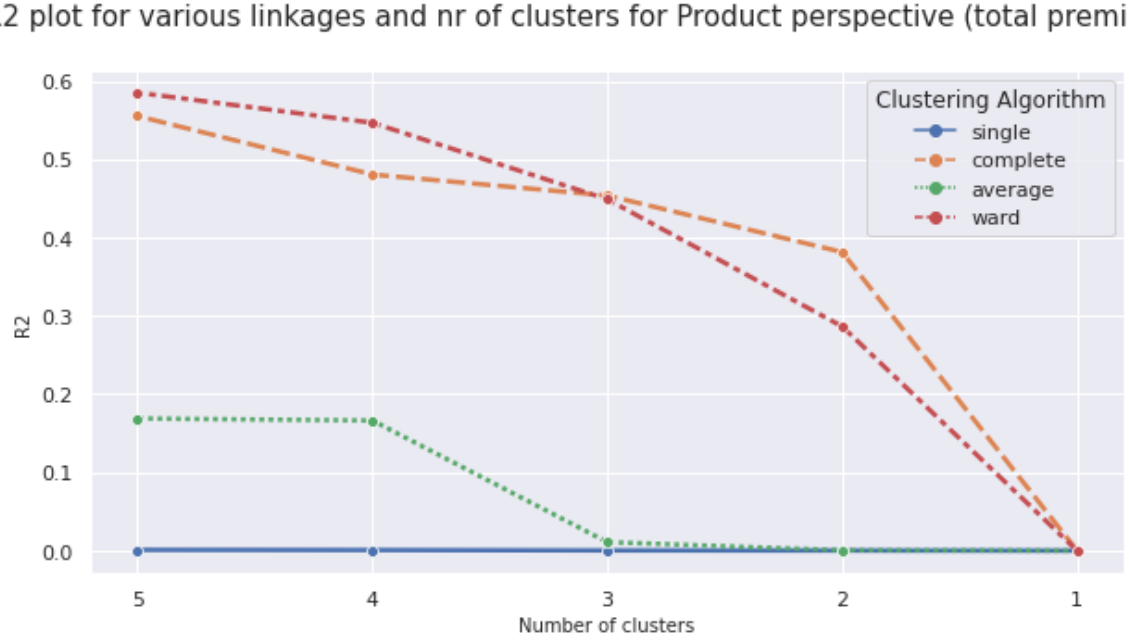


Fig 7.3.1

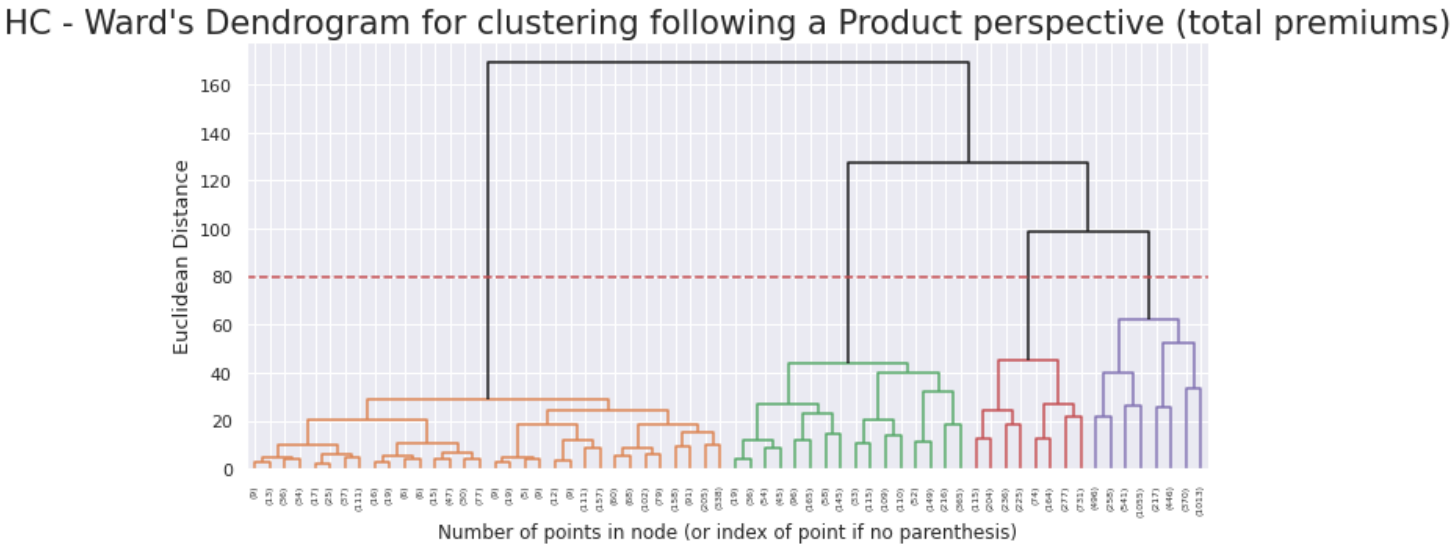


Fig 7.3.2

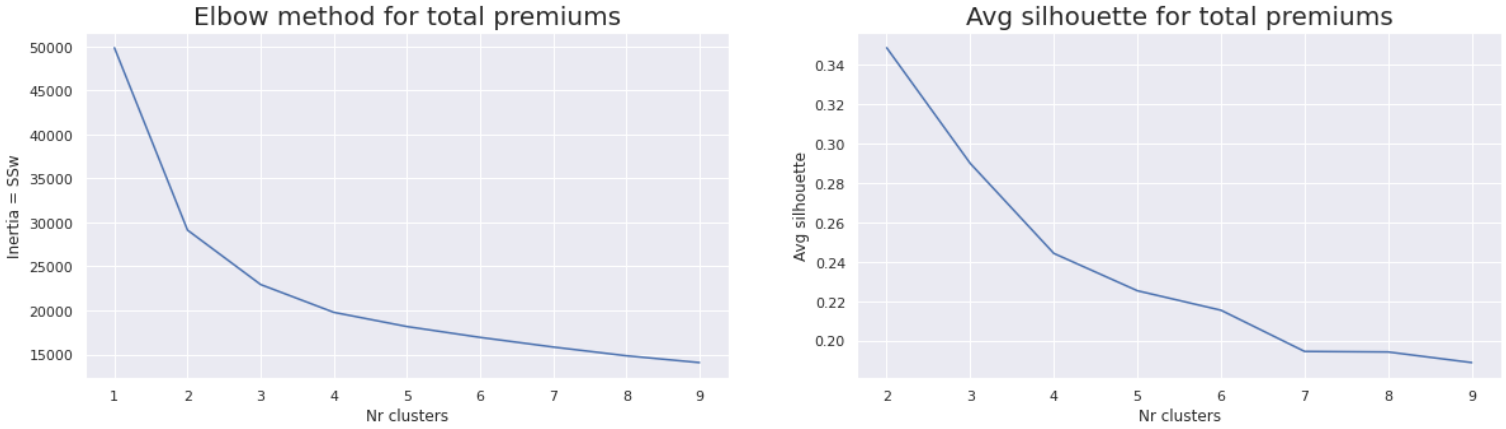


Fig 7.3.3

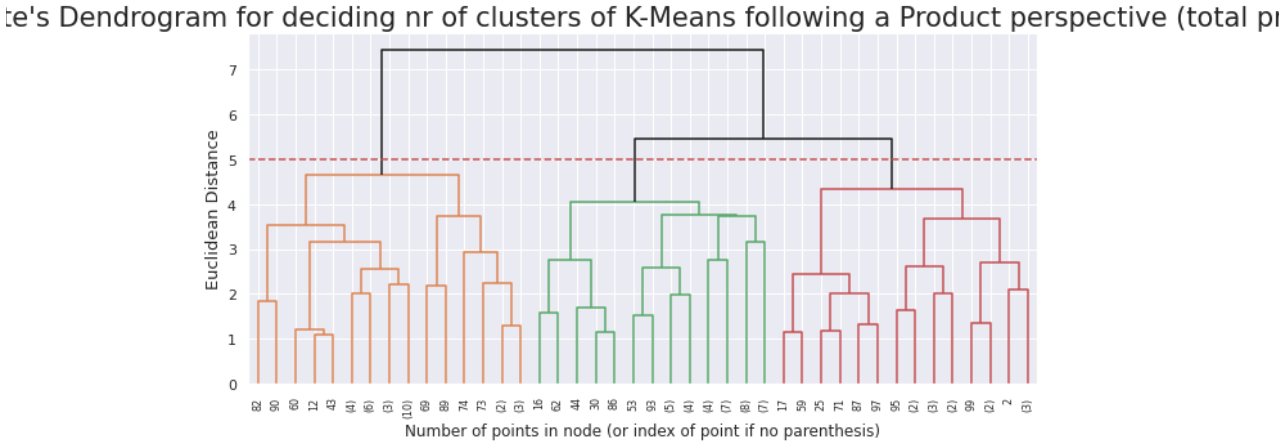


Fig 7.3.4

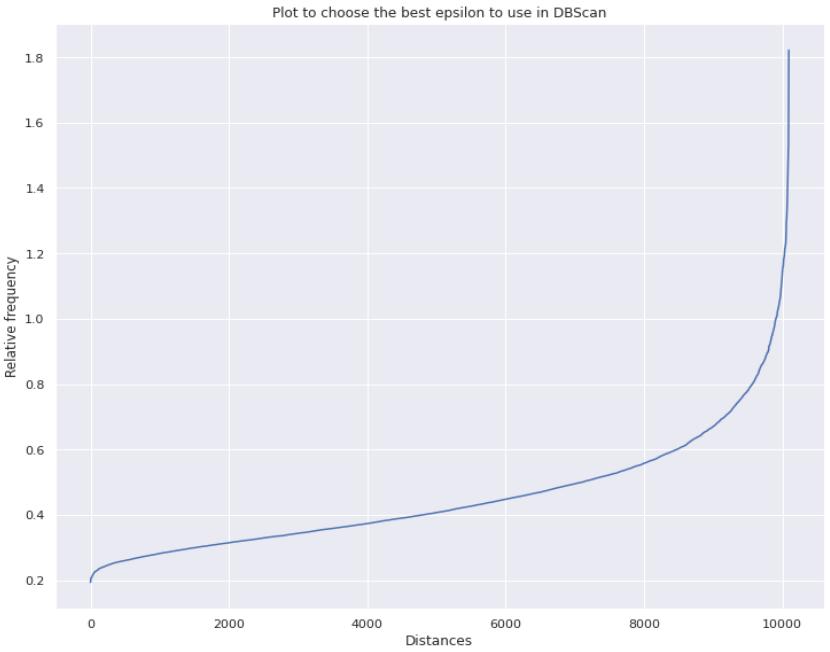
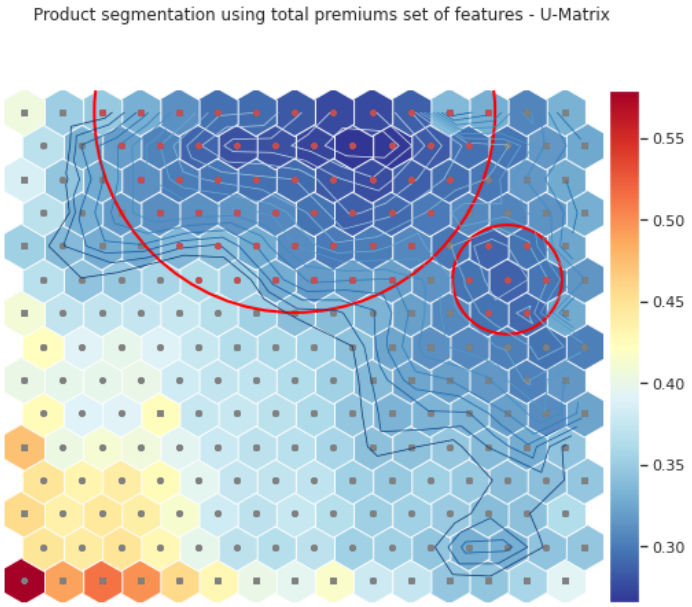


Fig 7.3.5



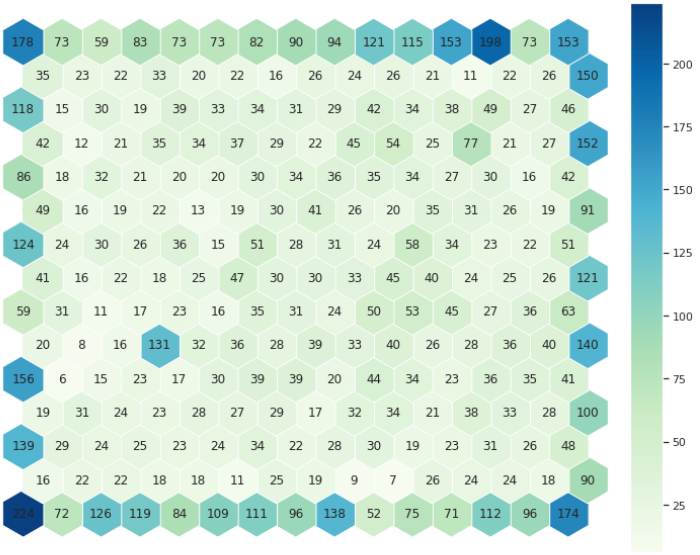
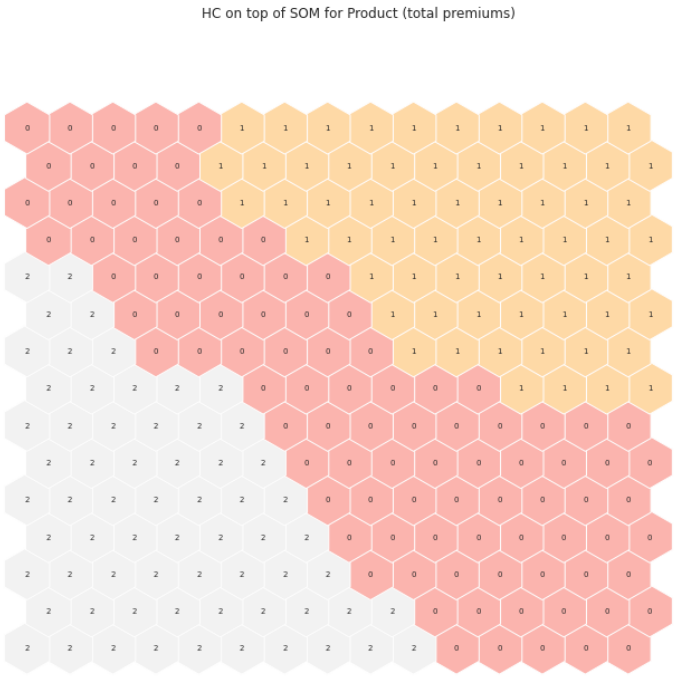


Fig 7.3.6

Fig 7.3.7



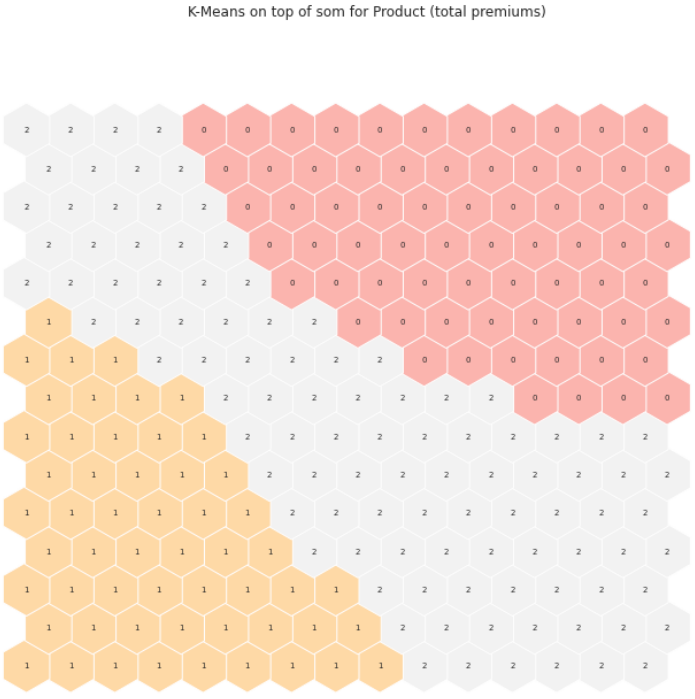


Fig 7.3.8

Fig 7.3.9

Algorithms for Product Segmentation	*Set of Prem features*		*Set of Percent features*	
	R2	*Silhouette*	*R2*	*Silhouette*
Hierarchical clustering	0.5465	0.1916	0.4277	0.2041
K-Means	0.5397	0.2900	0.4801	0.2723
DBScan	0.0175	0.1358	0.0331	0.1428
SOM + Hierarchical clustering	0.5407	0.3150	0.4383	007950861
SOM + K-Means	0.5264	0.2654	0.4544	0.2533

Table 7.3.1

Cluster (Socio-Demo Segmentation)	Number of observations
0	1138
1	1601
2	1193
3	3625
4	1601

Table 8.1

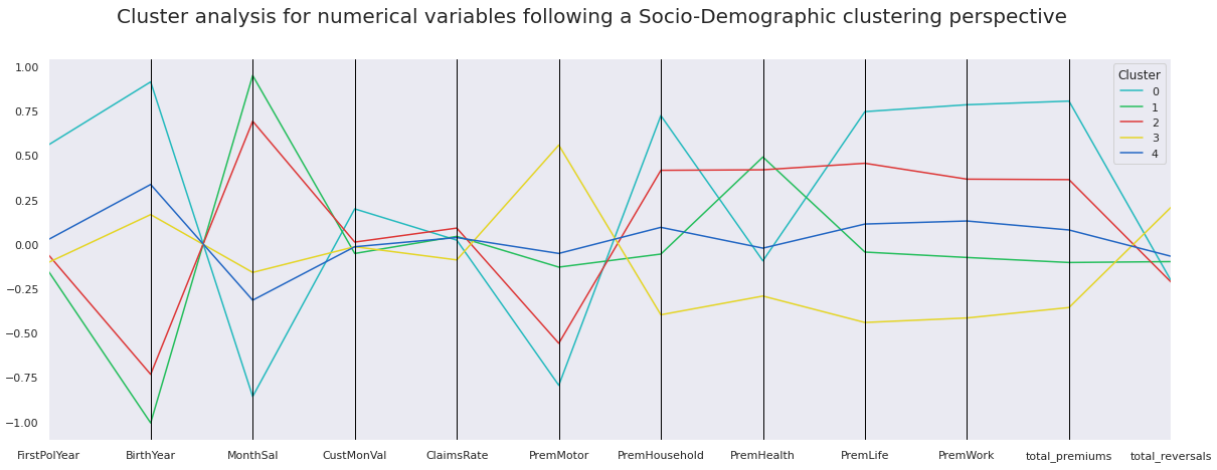


Fig 8.1

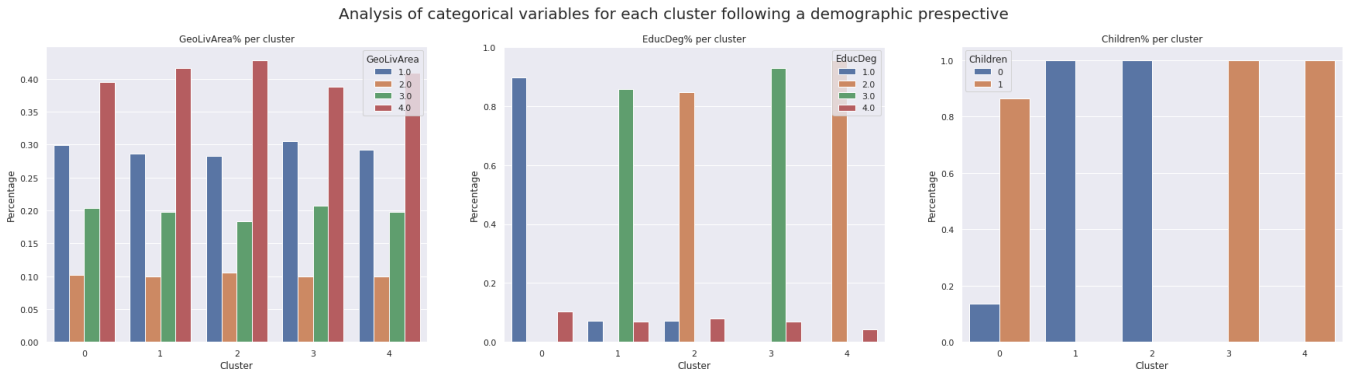


Fig 8.2

Cluster (Value Segmentation)	Number of observations
0	2468
1	1161
2	3294
3	1161

Table 8.2

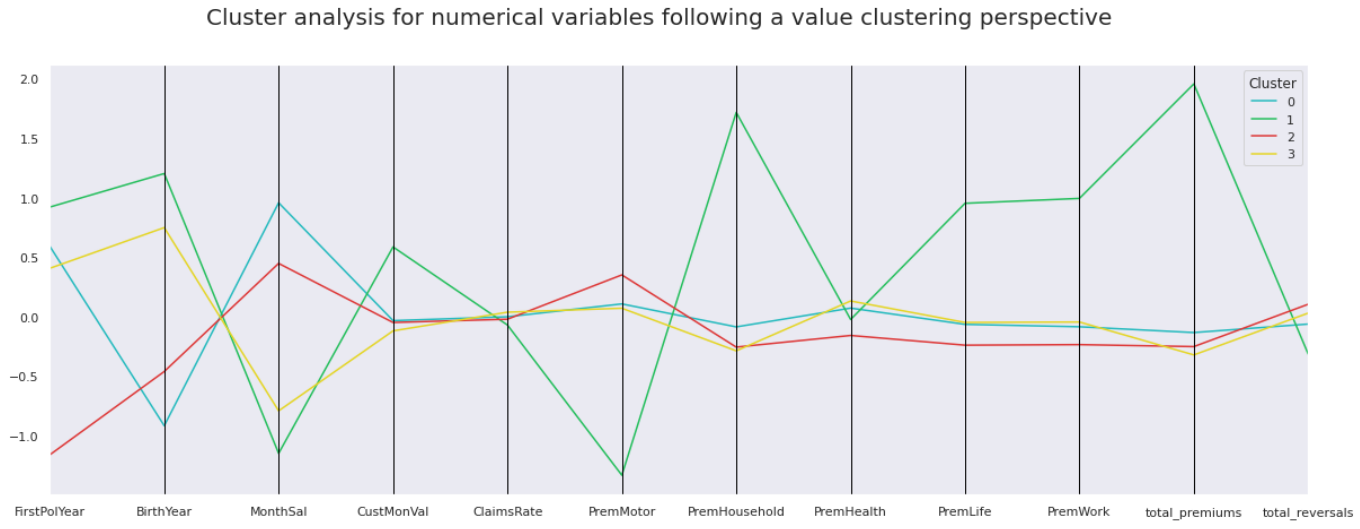


Fig 8.3

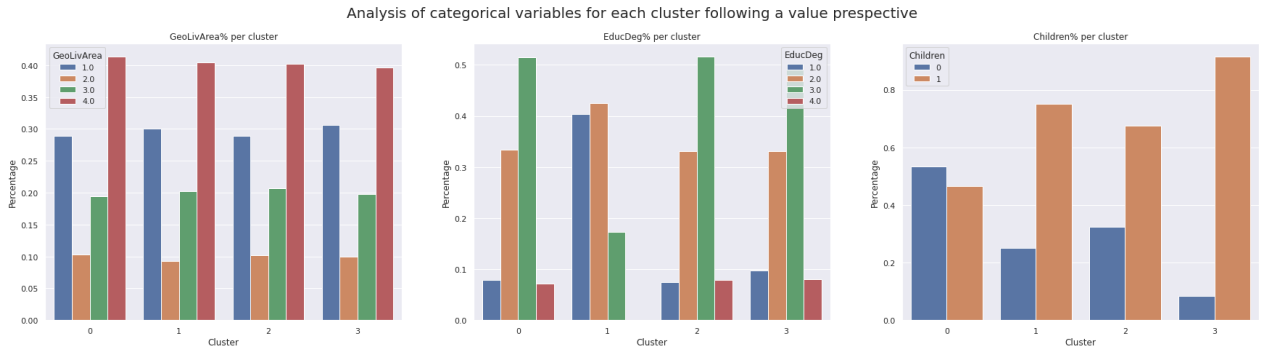


Fig 8.4

Cluster (Product Segmentation)	Number of observations
0	2309
1	3713
2	4097

Table 8.3

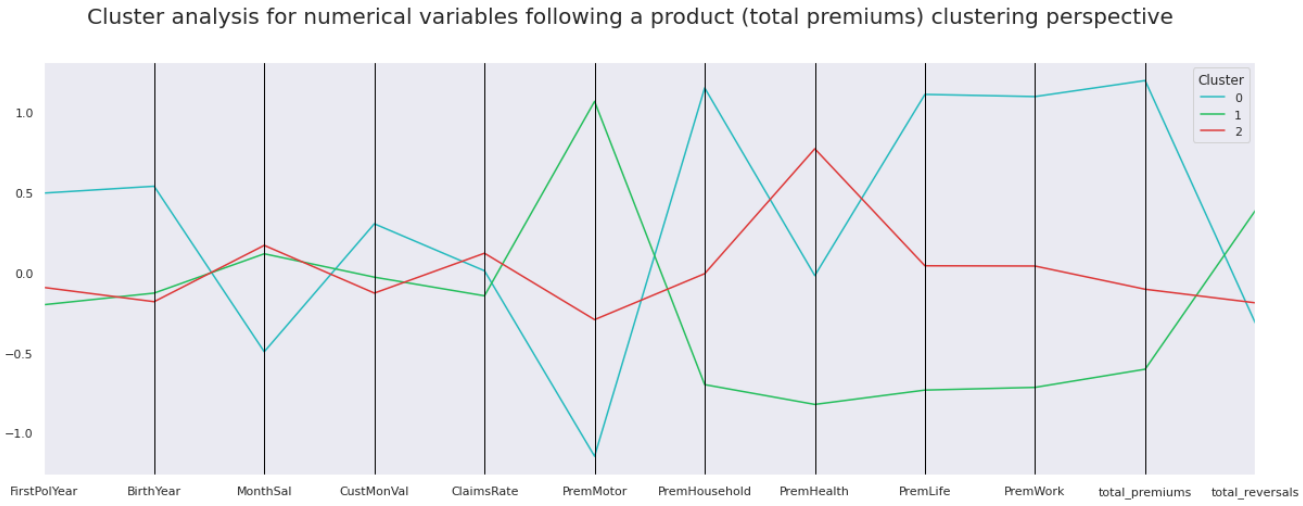


Fig 8.5

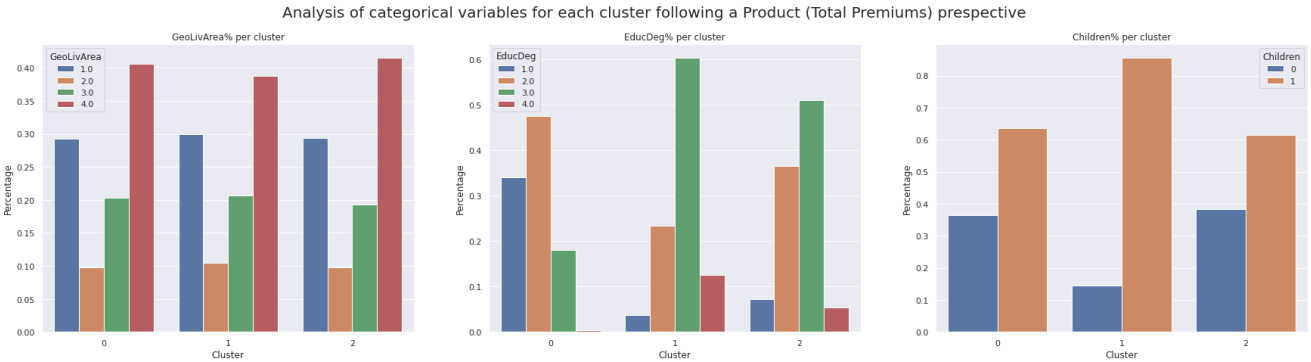


Fig 8.6

0	0	0	0.31	0.83	-0.89	-0.42	-0.07	0.92	0.12	0.16	-0.18	4	2	1
0	0	1	0.62	1.13	-1.08	-1.02	0.33	0.16	1.27	01.24	0.14	4	2	1
0	0	2	0.37	0.35	-0.46	1	-0.7	-0.73	-0.65	-0.65	-0.6	4	2	1
0	1	0	-1.11	-0.13	0.17	-0.12	0	0.4	0.13	0.1	-0.09	4	2	1
0	1	1	-1.24	-0.34	0.37	-0.67	0.88	-0.3	1,00	0.88	0.79	4	2	1
0	1	2	-1.1	-0.1	0.15	01.03	-0.61	-0.88	-0.63	-0.56	-0.55	4	2	1
0	2	0	0.66	-0.67	0.72	-0.16	0.12	0.29	0.19	0.21	-0.01	4	2	1
0	2	1	0.55	-0.65	0.75	-0.53	0.69	-0.3	0.86	0.98	0.55	4	2	1
0	2	2	0.65	-0.42	0.65	01.02	-0.49	-0.92	-0.59	-0.57	-0.48	4	2	1
0	3	0	0.63	0.89	-0.86	-0.76	1.41	0.85	-0.2	-0.01	1.43	4	2	1
0	3	1	0.8	1.16	-1.1	-1.28	1.67	0.04	0.98	0.98	1.86	4	2	1
1	0	0	0.31	0.86	-0.87	-0.33	-0.17	0.85	0.05	0.18	-0.24	4	1	1
1	0	1	0.96	1.36	-1.23	-1.4	0.29	0.08	1.68	1.56	0.09	4	1	1
1	0	2	0.37	0.64	-0.66	0.94	-0.73	-0.63	-0.65	-0.69	-0.64	1	4	1
1	1	0	-1.03	0.07	0.08	-0.15	-0.07	0.41	0.13	0.17	-0.17	1	1	1
1	1	1	-1.17	-0.19	0.2	-0.79	0.95	-0.21	0.97	01.03	0.84	4	1	1
1	1	2	-1.15	-0.04	0.07	0.92	-0.36	-0.84	-0.47	-0.53	-0.4	4	1	1
1	2	0	0.66	-0.76	0.72	-0.07	0.03	0.07	0.39	0.09	-0.11	1	1	1
1	2	1	0.6	-0.61	0.75	-0.53	0.64	-0.48	01.04	01.03	0.54	4	1	1
1	2	2	0.72	-0.44	0.55	0.71	-0.17	-0.65	-0.36	-0.36	-0.25	4	1	1
1	3	0	0.67	1.1	-1.05	-0.75	1.25	0.73	-0.13	0.12	1.17	1	1	1
1	3	1	1.23	1.5	-1.44	-1.62	1.85	-0.34	1.32	1.42	2.2	4	1	1
2	0	0	0.55	0.7	-0.77	-0.44	-0.21	0.75	0.47	0.25	-0.31	4	2	0
2	0	1	0.85	1.3	-1.24	-1.18	0.13	0.23	1.54	1.37	-0.03	1	2	0
2	0	2	0.25	0.36	-0.51	01.07	-0.49	-1.01	-0.44	-0.79	-0.48	4	2	0
2	1	0	-1.23	-1.16	01.03	-0.46	0.07	0.84	0.24	0.14	-0.04	4	2	0
2	1	1	-1.28	-1.13	01.06	-0.98	01.06	0.16	0.91	0.94	0.99	4	2	0
2	1	2	-1.15	-0.67	0.61	0.95	-0.5	-0.75	-0.64	-0.59	-0.48	4	2	0
2	2	0	0.53	-1.25	1.19	-0.51	0.13	0.99	0.2	0.08	0.01	4	2	0
2	2	1	0.53	-1.17	1.21	-1	0.88	0.28	0.99	0.95	0.78	4	2	0
2	2	2	0.58	-0.8	0.92	0.87	-0.51	-0.6	-0.55	-0.6	-0.48	4	2	0
2	3	0	0.21	0.87	-0.87	-0.85	1.48	0.98	-0.58	0.3	1.51	4	2	0
2	3	1	0.81	0.78	-0.77	-1.33	1.81	-0.03	01.06	0.86	02.09	4	2	0
3	0	0	0.31	0.95	-0.91	-0.26	-0.13	0.89	-0.04	-0.05	-0.23	4	3	1
3	0	1	0.57	01.07	-1.04	-0.88	0.43	0.21	1.11	01.03	0.2	4	3	1
3	0	2	0.39	0.4	-0.54	1.1	-0.79	-0.79	-0.85	-0.8	-0.66	4	3	1
3	1	0	-1.14	-0.2	0.2	0.01	-0.07	0.38	-0.08	-0.07	-0.15	4	3	1
3	1	1	-1.02	-0.38	0.45	-0.32	0.8	-0.37	0.58	0.87	0.69	4	3	1
3	1	2	-1.16	-0.17	0.17	1.18	-0.79	-0.95	-0.83	-0.8	-0.65	4	3	1
3	2	0	0.61	-0.62	0.72	0.02	-0.02	0.3	0,00	0.01	-0.12	4	3	1
3	2	1	0.47	-0.62	0.73	-0.27	0.6	-0.43	0.67	0.9	0.44	4	3	1
3	2	2	0.7	-0.5	0.62	1.14	-0.69	-0.96	-0.75	-0.74	-0.6	4	3	1
3	3	0	0.6	01.08	-0.91	-0.68	1.32	0.86	-0.31	-0.13	1.33	4	3	1
3	3	1	0.68	1.15	-1.07	-1.06	1.58	0.14	0.73	0.71	1.71	1	3	1
4	0	0	0.38	0.8	-0.8	-0.23	0.03	0.48	0.03	0.25	-0.09	4	3	0
4	0	1	0.51	0.99	-1.06	-0.93	0.25	0.14	1.22	1.12	0.09	4	3	0
4	0	2	0.41	0.2	-0.47	01.02	-0.56	-0.84	-0.58	-0.85	-0.53	1	3	0
4	1	0	-1.2	-1.23	1.15	-0.33	-0.1	1	-0.05	-0.07	-0.2	4	3	0
4	1	1	-1.19	-1.01	0.88	-0.89	0.87	0.29	0.96	0.85	0.81	4	3	0
4	1	2	-1.15	-0.94	0.87	0.89	-0.74	-0.52	-0.65	-0.74	-0.64	4	3	0
4	2	0	0.53	-1.32	1.29	-0.33	-0.09	0.98	-0.04	-0.06	-0.18	4	3	0
4	2	1	0.54	-1.3	1.26	-0.94	0.95	0.26	0.92	0.85	0.86	1	3	0
4	2	2	0.62	-1.13	1.12	0.84	-0.58	-0.45	-0.63	-0.66	-0.54	4	3	0
4	3	0	01.01	0.85	-0.73	-0.65	1.46	0.7	-0.5	-0.15	1.47	4	3	0
4	3	1	0.45	0.52	-0.59	-1.17	1.77	0.01	0.74	0.76	02.01	4	3	0

Table 9.1

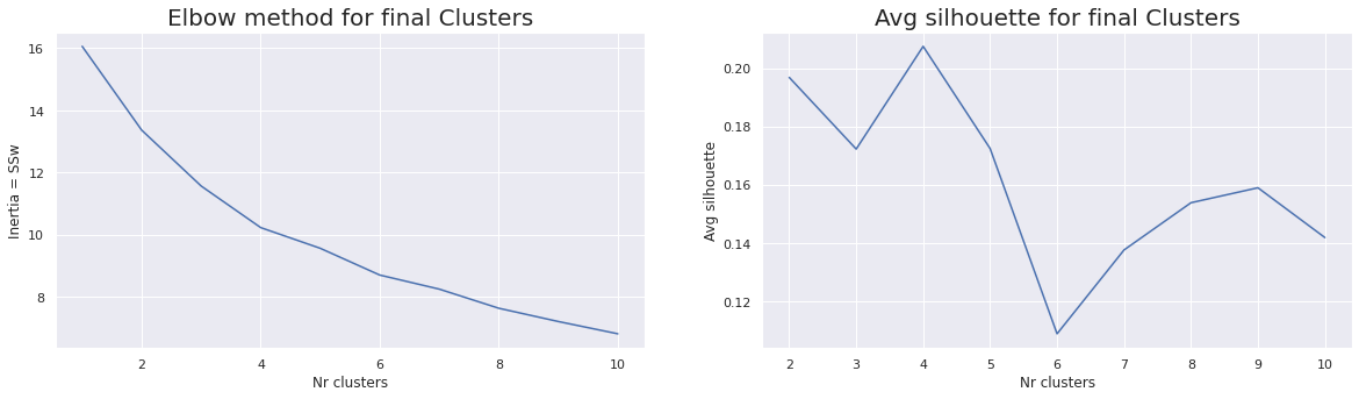


Fig 9.1

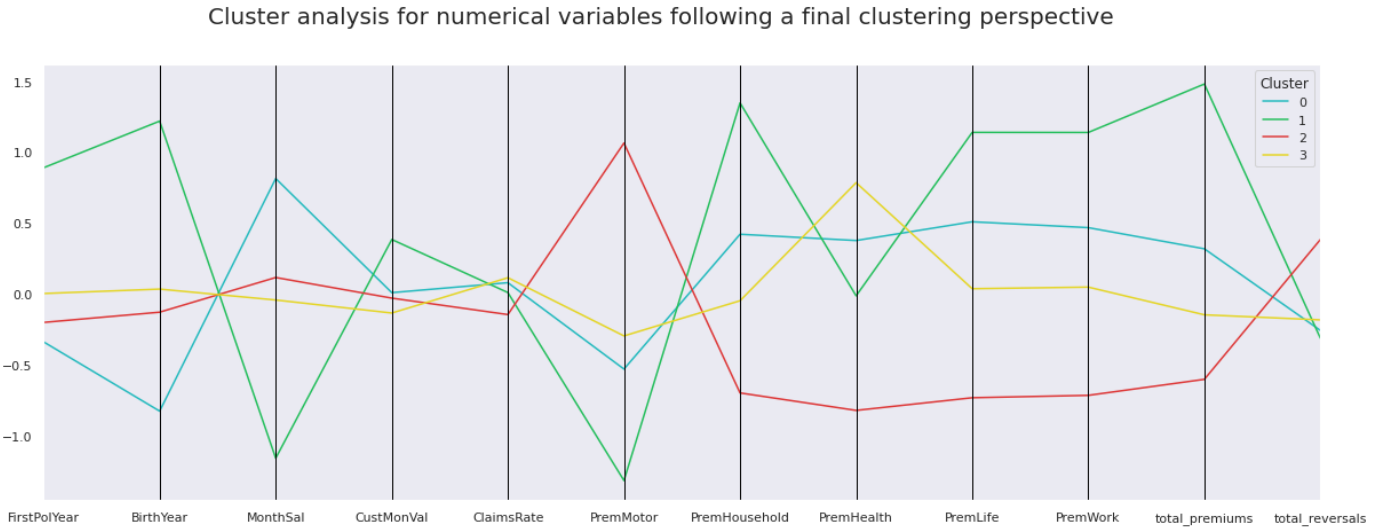


Fig 9.2

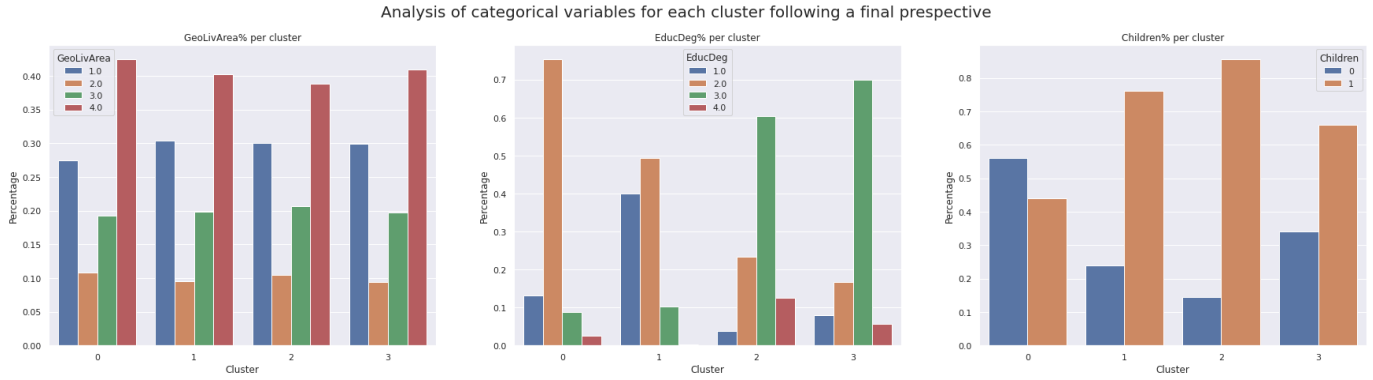


Fig 9.3

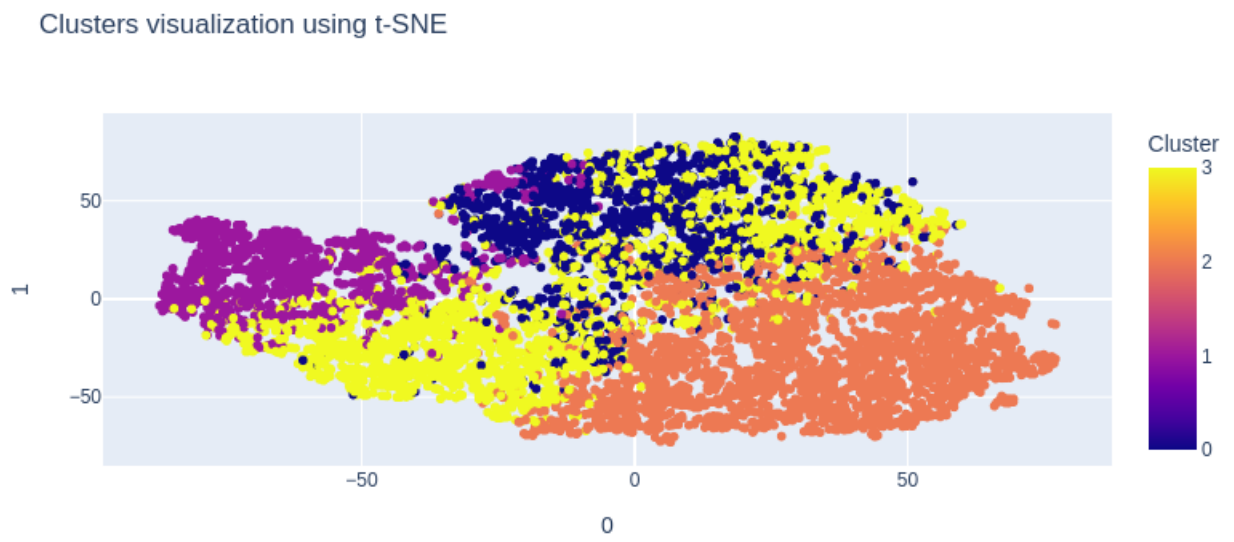

Clusters visualization using t-SNE
Cluster
3
2
1
0
50
0
−50
1
−50
0
50
0

Fig 9.4

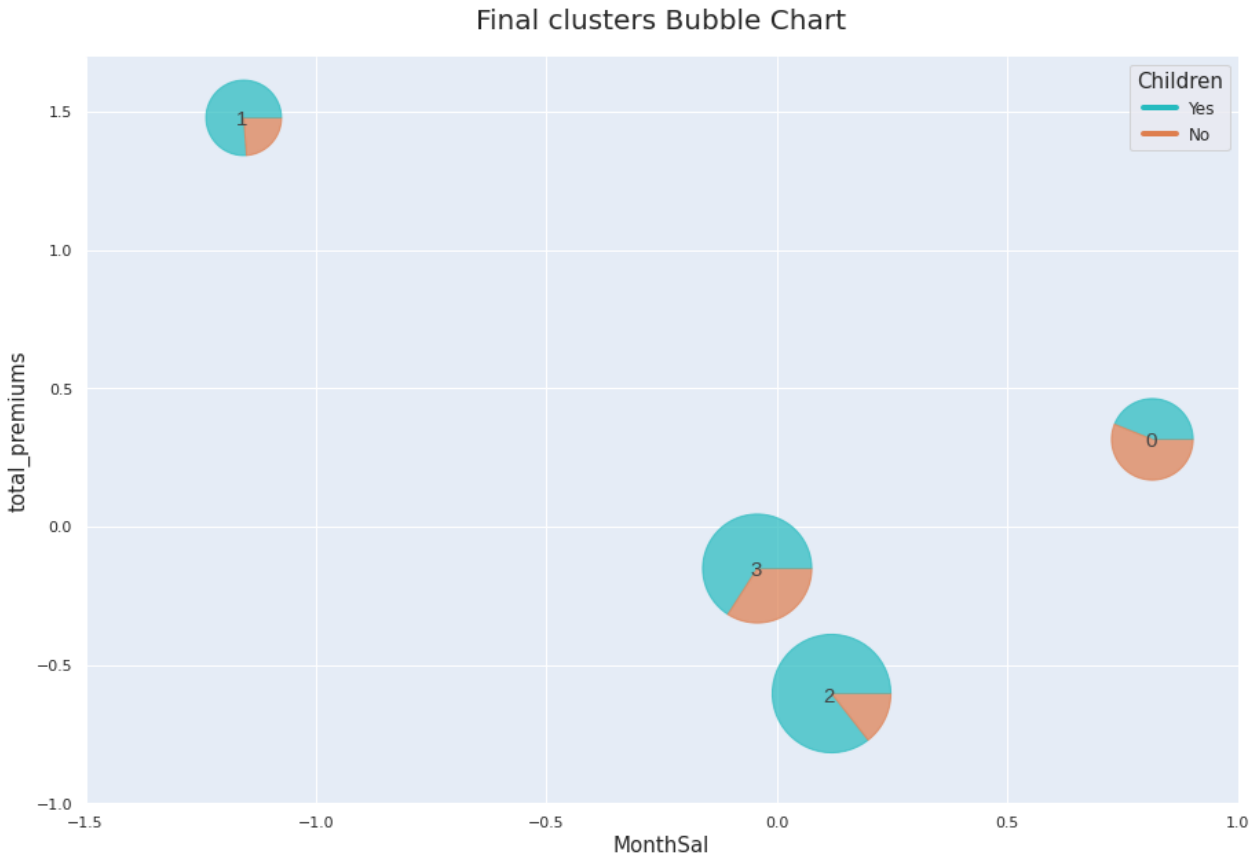

Final clusters Bubble Chart
Children
Yes
No
1
0
3
2
1.5
1.0
0.5
0.0
−0.5
−1.0
total_premiums
−1.5
−1.0
−0.5
0.0
0.5
1.0
MonthSal

Fig 9.5

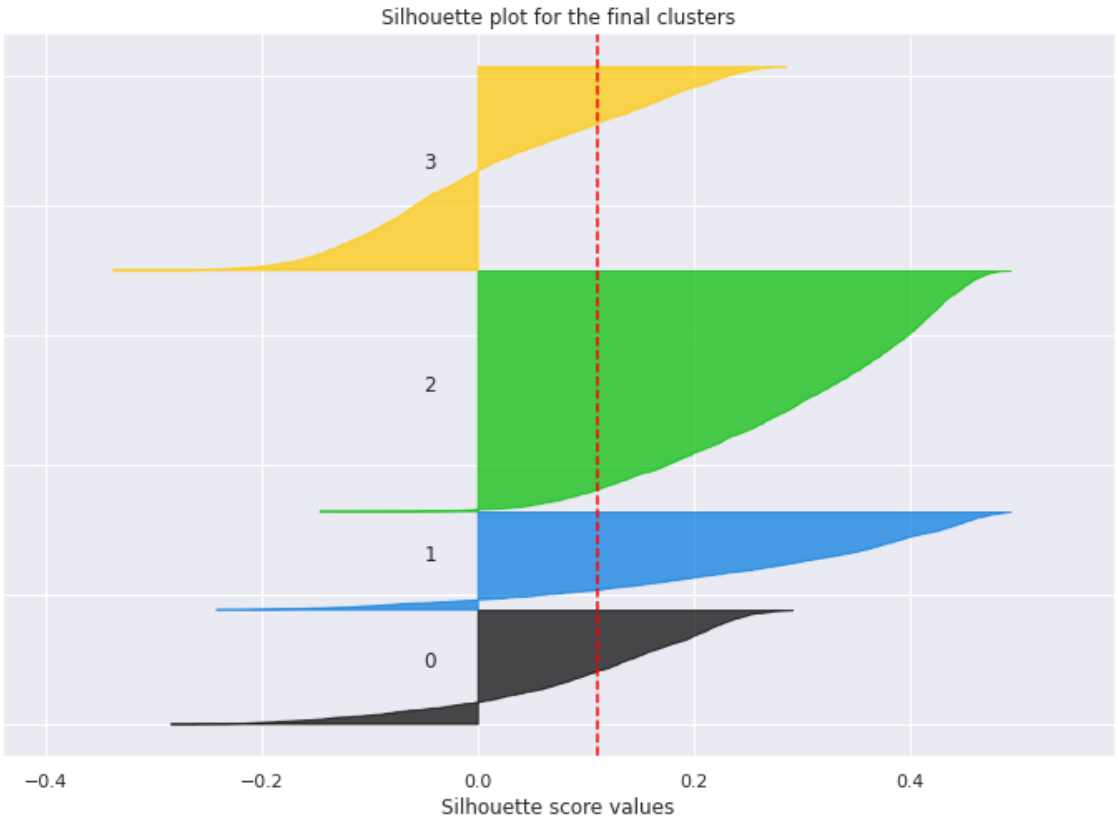


Fig 9.6

Feature	Importance
FirstPolYear	0.0038
BirthYear	0.0070
MonthSal	0.1603
PremMotor	0.5099
PremHousehold	0.0377
PremHealth	0

Feature	Importance
PremLife	0.0435
PremWork	0.0034
total_premiums	0.0587
EducDeg	0.1757
GeoLivArea	0
Children	0

Table 9.2