Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Segmentation of Customers for A2Z Insurance

Group AP

Adriana Monteiro, number: 20220604

Pedro Ferreira, number: 20220589

January, 2023

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

INDEX

1. Introduction

A2Z insurance is a Portuguese insurance company with a majority of national clients, but also with international clients who signed through the company's website. Because of the sign of the times, the enterprise decided it was time to update their process to a more data-driven methodology, so they started to try to use their data in a way that would help them increase profits.

We were presented with A2Z Insurance's database from 2016 which had some demographic and value characteristics of its customers with the purpose of trying to do some segmentation and develop more specific marketing campaigns for certain groups of customers. The dataset we were given had the features in table 1.

To reach the end goal of this project, we took special care at each stage of the process, it being the exploration, preprocessing, or clustering, so in the end we would get the best segmentation possible, so the marketing campaigns could the most personalized possible.

2. Exploration

The first thing we did, after assigning customer ID to be the index, was check the datatypes, where we saw that every feature had the same datatype – float – but "EducDeg", which was the object datatype. We also saw that almost every feature except for "CustMonVal", "ClaimsRate" and "PremHouseHold" had missing values and used a missing number matrix to showcase that fact. This matrix, in Figure 1, shows us that despite having a lot of features with missing values, there weren't a lot of observations where that happened in some column. Curiously, out of all premiums, only the household premium had missing values, which was later further investigated. We also saw that there were three duplicate entries in the dataset. For further analysis, we registered which were the categorical features – "GeoLivArea", "Children" and "EducDeg" – and metric features "FirstPolYear", "BirthYear", "MonthSal", "CustMonVal", "ClaimsRate", "PremMotor", "PremHousehold", "PremHealth", "PremLife" and "PremWork".

Box plotting and doing histograms of every numerical variable was where we first saw how many outliers there were on each feature, which were so many that the boxplots were barely perceptible, as we can see in Figures 2 and 3. This amount of outliers caught our attention, and when analyzing the features further we also had them in mind to see if we could understand why those were appearing.

So, we started going feature by feature to know a bit more of what was going on inside each one of them – and possible figure out what was happening with some outliers. In "FirstPolYear", as we can see in its boxplot, there was only one outlier. This outlier is one that is a plain wrong value, because it is not possible for a first policy year with the insurance company to be 53784. On "BirthYear", we saw that the only wrong value was that the year 1028 was on there, which, like the value we spoke about on "FirstPolYear", can not happen for a living person in 2016 or even a customer of an insurance policy which certainly was founded no more than 150 years ago. In "MonthSal", we realized that there were two customers which had a much higher salary than the rest of them, which was affecting the boxplots (and the mean, for example) – the so-called Bill Gates effect – with values that might not be wrong, but really higher than the insurance's "normal" range of customer salaries. For Claims Rate, we checked that 17% of customers had it above 1, which meant that, because it is the amount paid by the insurance company divided by the premium total the customer is paying, the company was losing money on 17% of its clients. The premiums seemed to be the features with the most outliers, where the one with the most was the Life insurance premium. Other features which were not mentioned had nothing out of the ordinary, just some outliers and missing values like the rest of the features – of course that the concept of outlier is not applicable to categorical features.

Then we started checking for incoherencies or inconsistencies within the data – like relationships of different features' values that did not make sense. We started by checking for people who had zero in all premiums, when we realized that no premium had zeroes except for the household premium. So, adding to our first analysis on missing values, we realized that household premium was the opposite from the others, i.e. the other premiums had missing values but no zeros, while household premium had zeros but no missing values. This was an important realization for our preprocessing. Therefore, instead of checking for people with no premiums, we checked for people with zero premium on household and missing values on all the other ones, where we found out that there were 12 individuals where this happened. Not only that, but all of them had -25 of customer monetary value and a claims rate of 0. This means that there was a cost of acquisition (bigger than the value the customers spent

on insurance the past 2 years) and that the insurance company either did not spend any money on these clients, or the clients simply had no premiums over the last 2 years, so the company did not spend money on them either – mathematically, claims rate for the latter would be a 0/0 indetermination, but for simplification one would just put 0 because what matters is the ratio of expenditure to money received. This was important information for when we later dealt with missing values on each feature.

The biggest incoherence we found was that there were exactly 1997 rows of data where a client's first policy year was before the year they were born on, which is not possible. So, we started to inspect the birth year feature, because it was the one where the logic for wrong combinations with other features would be easier. So, we saw that there were 87 people with less than 18 years old who had children – not impossible but unlikely -, that there were 104 minors which had a salary above minimum wage in Portugal (530€ in 2016) – also not impossible, but considering it is mandatory to be in school until 18 years old in Portugal, it would be highly unlikely for a child to be working full-time and studying full-time – and 116 with house insurance. All of this made us believe that this feature may not be trustworthy.

We also saw that there was one person that paid more insurance than their yearly salary, this just got our attention to a very high health insurance of 28272€, which was why this individual had this problem, so this was a value that could not be right. It was also checked if there were people who had work insurance but no monthly salary, but there was none.

To see the correlation between the numerical features, we also did a correlation matrix heatmap that's on Figure 4, where it popped out that birth year and monthly salary were very correlated, while the claims rate and customer monetary value were perfectly correlated.

3. Data Preparation and Pre-processing

3.1. Encoding

Because it did not depend on any other kind of preprocessing, we started by ordinal encoding the feature. We chose this kind of encoding because in spite of people's education not having a specific order, we thought that there was a clear less to more education which could be numbered, so we encoded the feature from 1 – basic education – to 4 – PhD.

3.2. Dealing with univariate outliers, incoherencies and missing values

The first decision we made was to delete the "BirthYear" feature. This decision was taken because of all the incoherencies that were pointed in the exploration part about this feature. Not only that, but because some of the incoherencies are technically possible, while highly unlikely, even if that analysis was totally wrong, which is not that probable, we would have the safety bag that the birth year feature is highly correlated (correlation coefficient of 0.8) with the monthly salary, so we indirectly kept information on that feature.

Then, we decided to change the wrong value we found in "FirstPolYear" to a missing value.

The other major inconsistency had to do with the missing values in all premiums but household we talked about earlier. For this, we decided we would impute 0 to all these missing premiums, on the basis that it did not make a lot of sense that every client had every premium except for some clients that just did not have household insurance. What added suspicion to our analysis was precisely the 12 cases we talked about – disclaimer: 12 instances are not enough to make assumptions of a whole dataset, they just provided some extra information that sparked our suspicion of what was going in this situation, while being good examples to justify our decision. If they had no claims rate and had a negative customer monetary value, that meant that it is highly probable that they did not pay any kind of premiums, while they had missing values on most of them. So, we thought that there was a mistake of consistency when defining how the company would register that a client did not have a premium, so the household premium got zeros while the others got missing values. Those 12 clients, just like the duplicate clients, were removed from the dataset, because they added no information about the current status of the insurance company, as having no premiums meant that realistically they were not this company's clients.

So, we started to check for outliers and how we could treat them. As we had already seen with the boxplots, there were a lot of outliers in many features. So, we did the Inter Quartile Range to check precisely how many there were on each feature. Furthermore, we inspected what percentage of data we would be keeping if we removed a feature's outliers and in the end what percentage we would keep if we deleted all the outliers. Soon enough, we figured the approach of just removing values outside the IQR was not suitable, because even though for some features we would keep 99% or more, for other features we would only have 93% left of the dataset left. To make it worse, if we removed them all, we would only keep 85% of the observations. All of these numbers can be seen in Table 2. So, this could not be the best way to treat the outliers.

With all this, we decided we would first do manual thresholds that would eliminate individuals with extreme or plain wrong values on some feature. By doing this, we kept 99.8% of the initial rows

of data. Our thought process for the values in the thresholds was the following: looking at the initial boxplot of Figure 3, we decided to exclude from the dataset the individuals which had abnormally high values, which were noticeably very far apart from the whole rest of the points. The decided thresholds are presented in Table 3. But this was not the last thing we did to outliers, as we knew this would not be enough to remove all the noise from the dataset and did this little thresholding – careful not to delete too much data – because we wanted to use DBSCAN to detect multidimensional noise (we will talk about that after the imputing).

As it was said earlier, we had decided to impute the missing values in all premiums with zeros, but there were other features with missing values other than those. For numerical features that required imputing, we decided to go with kNN imputer with the standard number of k (5), but had to use only k=1 for the imputing on categorical variables. This had to happen because kNN imputer uses the mean (and is restricted to the mean) of the neighbors to assign the missing value. All categorical variables were already encoded with numbers (either because we encoded, like education, or they were originally already binary or represented by numbers, like "Children" and "GeoLivArea" respectively). Well, the mean of categorical values, even if represented by numbers, has no meaning (e.g., k=2 and the neighbors to a missing value had values of 1 and 3, the value assigned would be 2, which is a category that has nothing to do with its neighbours'), so we just use the value of the closest neighbor. However, we also used this type of kNN for first policy year, the only numeric discrete variable. The reasoning to this was a bit different, and was because of the following: as we wanted every feature to have the same importance for the calculation of the closest neighbors, we used min max scaler to get everything numeric on the same scale. That meant that the mean of scaled discrete numbers would issue the problem of assigning a number that after reversing the scaling would not correspond to any year, so would not have meaning. For short, we used 5-NN to impute missing values on "MonthSal" and 1-NN to impute on "FirstPolYear", "GeoLivArea", "Children" and "EducDeg_enc", which is the education feature but ordinally encoded.

Having this done, we could finally start searching for multivariate outliers. So, we used DBSCAN, with min_samples being 18, which is the recommended amount because it's the doubled number of features we are using (remember that we are using only the 9 numeric features, DBSCAN uses Euclidean distance and that has no meaning for categorical variables, neither does the concept of outlier). So, we did the plot to check for the right epsilon, which is on Figure 5, by searching for the elbow, which was found at around 0.25. So, we performed the DBSCAN with those parameters and found 2 clusters, one of them being the outliers. So, there were 51 multivariate outliers, so around 0.5% of our data. As intended when we first came up with the strategy, we removed those outliers, so after all treatment we ended up with 99.1% of our data.

3.3. Feature Engineering

Moving on, we thought about some features which could help our process. First of all, we did a feature called "PremTotal", which simply added all premiums to get the amount paid on premiums by each customer in that year. Because Claims rate is defined as the amount paid by the insurance company divided by the premiums the customers pay and because this is defined as being the last 2 years, we got a feature called "LosingMoney2Year", which was 1 if it was – this happens then the claims rate is higher than 1 – and 0 if it was not. Similarly, we got a feature called "CompanySpend2Year" to see how much the company was spending on each customer. This was easily made by multiplying the

total premiums by 2 (because claims rate is of the last 2 years) and by the claims rate. By the same rational of "LosingMoney2Year", we saw which customers had a monetary value above 0, which meant the customer was, on all their stay with the company, profitable – customer monetary value is defined as annual profit from the customer times the number of years they are with the customer minus the acquisition cost of the customer. This formed the feature "ProfitableCust", a categorical feature which is 0 on 1 depending on whether or not the customer is not or is profitable, respectively. A ratio between a customer's total premiums and annual salary was also made, called "RatioPremSalary". Finally, we decided to do another binary feature regarding if a customer had a reversal on any of their premiums or not, called "Reversal".

By looking at this last one, "RatioPremSalary" grabbed our attention, because it was really skewed, in a way that did not look acceptable, so we decided to do a square root transformation which seemed suitable for distributions like this, with a skew on the right side, to make this variable have a more friendly distribution. We can see all engineered features' histograms in Figure 6, where we have the ratio before and after the transformation and we can observe that it helped this ratio be more similarly distributed to the premiums.

3.4. Feature Selection

3.4.1. Metric Features

First thing we did was drop the "normal" ratio of premiums to salary, keeping the square root transformed one. Then we used a correlation heatmap, scatterplots and and self-organizing maps' component planes to examine these features (Figures 7, 8 and 9).

We decided to remove was "CustMonVal", because on the correlation heatmap and the scatterplot we saw that it was very correlated with "ClaimsRate". Not only that, but when looking at the SOM component planes of those two features, they seem to give symmetric information about our data. For those reasons, having both of them seemed redundant and the decision to remove "CustMonVal" was because we had a feature ("ProfitableCust") which was created using exactly that one, so in a we would keep some information of a client's monetary value there. "CompanySpend2Year" was also very correlated with those two, as it was engineered from ClaimsRate, so did not seem to add anything new. It was deleted as well.

The other two features we decided to not use from then on were "MonthlySal" and "PremTotal", since information about those was on the new feature "sqrt_RatioPremSalary". But there was more to it, as "PremTotal was perfectly correlated with the household premium, having also almost identical component planes on the SOM. About the monthly salary, it was highly correlated with the premium/salary ratio, which we could see both on the correlation matrix and the scatterplot, and had a SOM component plane inverse to the ratio as well. Those were enough reasons to think that keeping those two would just be hold on to redundant information.

All the premiums also had very similar behaviors, but we decided to keep them all because we thought it would be important for the business case to keep all the products the company sells.

3.4.2. Categorical Features

For this part, we wanted to see how well the categorical variables could discriminate data points or find patterns in our observations, so for each feature we did all the scatterplots of numerical data and we colored the observations by its value on the category it belong to of a specific feature (Figures 10, 11, 12, 13, 14 and 15). After that, we used t-SNE and UMAP (Figures 16 and 17) to reduce the dimensions of the numeric part of the dataset and did the same coloring as in the scatterplots.

For "Children", we could not see much segmentation on the scatterplots of figure 10, but it seemed like people with children had more tendency to have higher motor insurance and the inverse for health insurance. On the dimensionality reduction plots, UMAP or t-SNE, it looked like this feature had somewhat of a tiny capability of points from the categories being in different places, but not a lot. The results were a bit inconclusive; we kept the feature.

Either in the scatter plots (figure 11), or in UMAP or t-SNE, "GeoLivArea" seemed to have effect at all, its values appeared all over the place, no matter what the category was. So, we decided we should remove this feature, as we thought it would only add noise.

"Education_enc" – or our encoded level of education feature – on the other hand, had on its scatter plots (figure 12) a really distinguishable fade of categories, meaning different values for other features had different education levels associated with them. On the reduced dimensionalities, this separation was also noticeable. So, we kept this feature.

The feature "LosingMoney2Year" showed itself to behave identically to "ProfitableCust" when analysing their colored scatterplots (figures 13 and 14), so we discarded it right away, as we realized they were made in a very similar way and that the profitable customer feature had a bigger range of years represented. On the scatterplots, at first sight there might seem to be a big discrimination from this variable in "ClaimsRate", but that is just because of the way these features are constructed. On the other hand, in the UMAP and t-SNE the places of different categories look very different – but that might also be because ClaimsRate is a metric feature used in the dimensionality reduction methods. So, we kept "ProfitableCust", but kept an eye on it later on.

Finally, "Reversal" seemed to show itself fairly discriminative in its scatterplots (figure 15) and in the reduced dimensions plots – maybe not more because it is a fairly unbalanced feature. So, we also kept it.

In the end, with the metric and categorical features, we kept the following: "FirstPolYear", "ClaimsRate", "PremMotor", "PremHousehold", "PremHealth", "PremLife", "PremWork", "sqrt_RatioPremSalary", "Children", "EducDeg_enc", "ProfitableCust", "Reversal".

4. Clustering

4.1. Clustering all Features

After the feature selection, we moved on to clustering the data. Our first approach was to cluster all features together, having in mind that some algorithms – in our case: K-Means, Self-Organizing Map (SOM) and DBSCAN – only work with metric features.

We started by using DBSCAN with the same procedure as before, however, it did not seem to be able to identify any clusters, since the result was a single cluster with most of the data and 157 outliers. Having this in consideration, we decided that this algorithm was not suitable for the problem at hands.

Next, we decided to use K-Means and Hierarchical clustering, this last one with different types of linkage (complete, average, single and ward), and plotted the R^2 for a number of clusters (k) from 2 to 9 (figure 18). To find the optimal k, we should look for the elbow in the graph, but since it wasn't very clear, we calculated the silhouette score, that is, the average of the silhouette coefficient for each cluster, for the same range of k. The best value obtained was when k was equal to 2. This solution proved to not be good as well.

Then, we chose to perform an Emergent SOM with a K-Means clustering on top. Starting with a 50 by 50 net, we trained the SOM. After getting all of the nodes' positions, we clustered them over different values of k and calculated the sum of squares within groups (inertia) and plotted it (figure 19). With this, we found that the best number of clusters for this method was 4.

The last clustering technique we used was K-Prototypes, an algorithm that clusters numerical and categorical that by accommodating both K-Means and K-Modes, which uses the number of most common categories between data points as a similarity score. To get the optimal number of clusters, K-Prototypes has a built-in "cost" function, similar to the sum of squares within groups, but that accounts for categorical and metric features. By plotting the cost over the same range of k as before (figure 20), when looking for the elbow of the plot we determined that the best number of clusters to use was between 2 and 4 clusters. However, when looking at the means of the variables for each cluster for the three solutions, the only solution that differentiated costumers in "FirstPolYear" was the one with 4 clusters, so we determined this as the most fitting solution.

In the end, we thought that the best clustering solution for all variables was the one from K-Prototypes because, even though it had a worse R^2 than the solution provided by SOM, the number of individuals was more balanced for each class.

4.2. Separating by different perspectives

After dealing with all the dataset, we decided to separate the clustering between two perspectives: value for the company and demographic characteristics of the costumers. The first incorporated the features: "ClaimsRate", "PremMotor", "PremHousehold", "PremHealth", "PremLife", "PremWork", "ProfitableCust" and "Reversal" and the second one "FirstPolYear", "sqrt_RatioPremSalary", "Children" and "EducDeg_enc".

Regarding the demographic perspective, considering the data only had two numerical features, there was no need to for a clustering algorithm since it is possible to observe the feature space in two dimensions. We plotted the features against one another (figure 21) and did not encounter any evidence of clusters. So, we decided to only cluster with K-Prototypes, because it is able to use all four demographic features. By conducting the same analysis with the cost function as before (figure 22), we decided that the best number of clusters was either 2 or 4. However, the only significant change in the mean of the first solution (k = 2) was in "FirstPolYear" while the second (k = 4) distinguished the group means in "FirstPolYear" and "PremMotor", so we decided this one was the best.

On the other hand, for the features related with company value, we performed both K-Means and Hierarchical clustering on only the metric features, and K-Prototypes on all. For the first two algorithms, by comparing the R^2 value for multiple values of k like before (figure 23), we concluded that a 4 cluster K-Means solution was the best option. The silhouette score was very close with the one with 2 clusters, but still the highest overall. In the case of K-Prototypes, after plotting the cost function over the value of k (figure 24), we discovered that the most suited number of clusters was 4 as well. To find the best solution for the value features, considering that the means of each numerical feature, were very close, we first compared the R^2 for both solutions and the one for the K-Means was higher. However, we suspected that even though K-Prototypes had a worse performance in terms of R^2, its' clusters might show differences between themselves in terms of categorical features, considering it uses them for clustering. By comparing the frequency of each categorical label over each cluster, we accessed that the differences between the two methods were not enough to justify using K-Prototypes. Thus, we decided that the best clustering solution was the one provided by K-Means.

We wanted to conjugate these two clustering solutions, yet, 16 groups seemed too much for a marketing campaign (it wouldn't be practical), hence we decided to aggregate the clusters using Hierarchical clustering. We were left with 8 final clusters.

Comparing the means for each feature of this solution (figure 25) with the one obtained with just K-Prototypes (figure 26), we concluded that the one that contributed the most for our research was the one with the merged demographic and value clustering (8 clusters) since it provided more insights about the costumers' behaviors. A t-SNE and UMAP representations of the clustering solution can be observed in figure 27.

4.3. Cluster analysis

After choosing the final clustering solution, we got 8 clusters with different characteristics over all features we worked with. However, some clusters had similarities, which was good in a business point of view because we could join some of them on certain marketing campaigns, as 8 different campaigns would have been too many. All further analysis will be based on figure 25 and figure 28.

We saw that our older customers were part of clusters 0, 1, 4 and 6, but they did not stay together on the rest of the characteristics. To analyse the clusters, we decided we would do it in a 3-category way – higher class meaning that a customer had a higher value in a certain column, lower if the contrary happened and a middle class. The behaviors on the monetary features (premiums and the ratio of premiums and salary) separated very well on these three classes. So, clusters 1, 2, 6 and 7 were the ones who spent the most on Motor Premium, while being the opposite for Household, Health, Life, Work and the Ratio. On the other hand, clusters 0 and 5 were the ones where less money was being

spent on Motor premium, while being highest on the ratio, Household and Life and mid class for the rest of the insurances. For last, 3 and 4 were always mid class or better, for every metric feature that involved money. These are the most stable customers in terms of consistency of how much they spend on each premium.

For the categorical variables, cluster 3 seemed to have a lot of people without children, while 1 had a vast majority of people in the contrary situation. Clusters 0 and 5 had a bigger proportion of less formally educated people and 4 had curiously a lot of people who ended their academic career in high school. The binary variable which told us if a customer was profitable or not had only values of 1 (profitable) for clusters 2 and 6, while 1 and 7 had more non-profitable than the opposite. The rest of them had a profitable majority. Finally, in all clusters there was a majority of people that did not have Reversals, but 1, 2, 6 and 7 were the ones that seemed to have a higher rate of reversals.

4.4. Marketing Campaigns

We decided we would do four different marketing campaigns:

1- "Same car, new life?" for clients in clusters 1, 2, 6 and 7: "If you upgrade of sign to two or more insurance policies other than motor insurance, we offer a discount of 25% on the value of the upgrade (or the registration fee). If you have children, we give you 10% more discount if that insurance is Life insurance". This campaign was done because these people had a really higher motor premium than the rest, and it tries to incentive them to get more or upgrade other insurances. It has the add-on of life insurance if they have children because we know cluster 1 is the one with a bigger children rate.
2- "Motor +" for clients in clusters 0 and 5: "If you do not have motor insurance, we give you 15% off on the first premium, and if you already have it but upgrade it, we give you 20% of the value of the upgrade!". These are the groups with less motor insurance, so we want them to get some more of that.
3- "Stable Customer Package" for people in clusters 3 and 4: "You are a good customer, so we want to thank you for that by giving you a free 1-level upgrade of an insurance policy if you upgrade one of the following: household, life or work".
4- "Special one" for people in clusters 1 and 7: "We offer 15% in any insurance premium if you upgrade 2 others". These are the groups of people with the less ratio of profitable customers, so we want to have them sign or upgrade for two more policies.

5. References

Huang, Z., 1998. “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values”. Pages 285-302. huang98extensions.pdf (hkust.edu.hk)

Aprilliant, A., 2021. “The k-prototype as Clustering Algorithm for Mixed Data Type (Categorical and Numerical)”. https://towardsdatascience.com/the-k-prototype-as-clustering-algorithm-for-mixed-data-type-categorical-and-numerical-fe7c50538ebb

Coenen, A. & Pearce, A., year not disclosed. “Understanding UMAP”. https://pair-code.github.io/understanding-umap/

Blog post on Anatomise Biostats, 2017. “Transforming Skewed Data: How to choose the right transformation for your distribution”. https://anatomisebiostats.com/biostatistics-blog/transforming-skewed-data/

IBM Support, year not disclosed. “Clustering binary data with K-Means (should be avoided)”. https://www.ibm.com/support/pages/clustering-binary-data-k-means-should-be-avoided

Moosavi, V., Packmann, S., & Valles, I.. (2014). SOMPY: A Python Library for Self Organizing Map (SOM). GitHub - sevamoo/SOMPY: A Python Library for Self Organizing Map (SOM)

6. Appendix

6.1. Figures

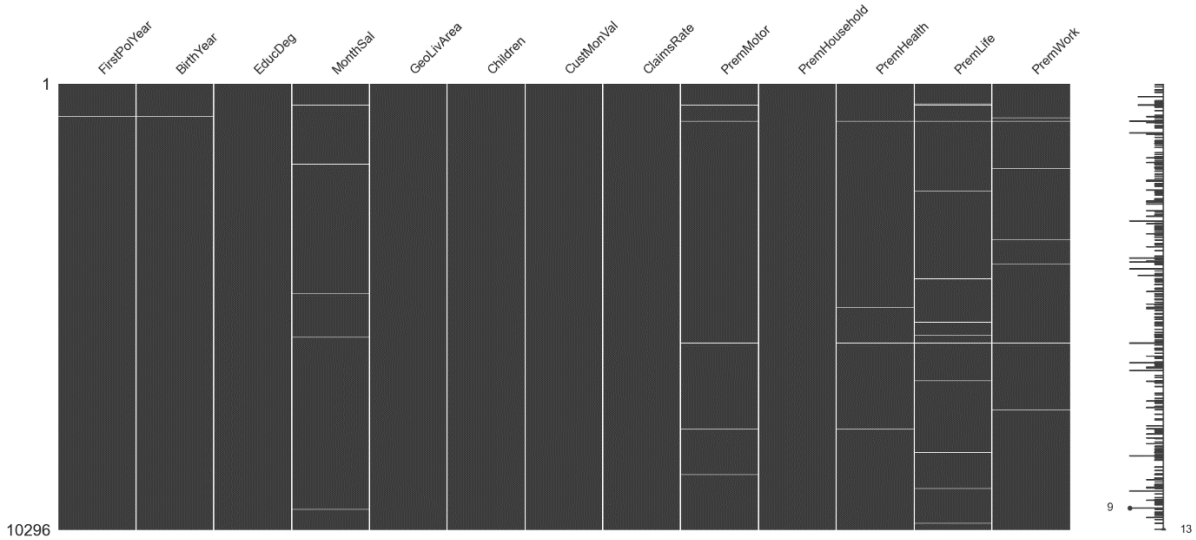

Figure 1 – Missing values map for our entire dataset.

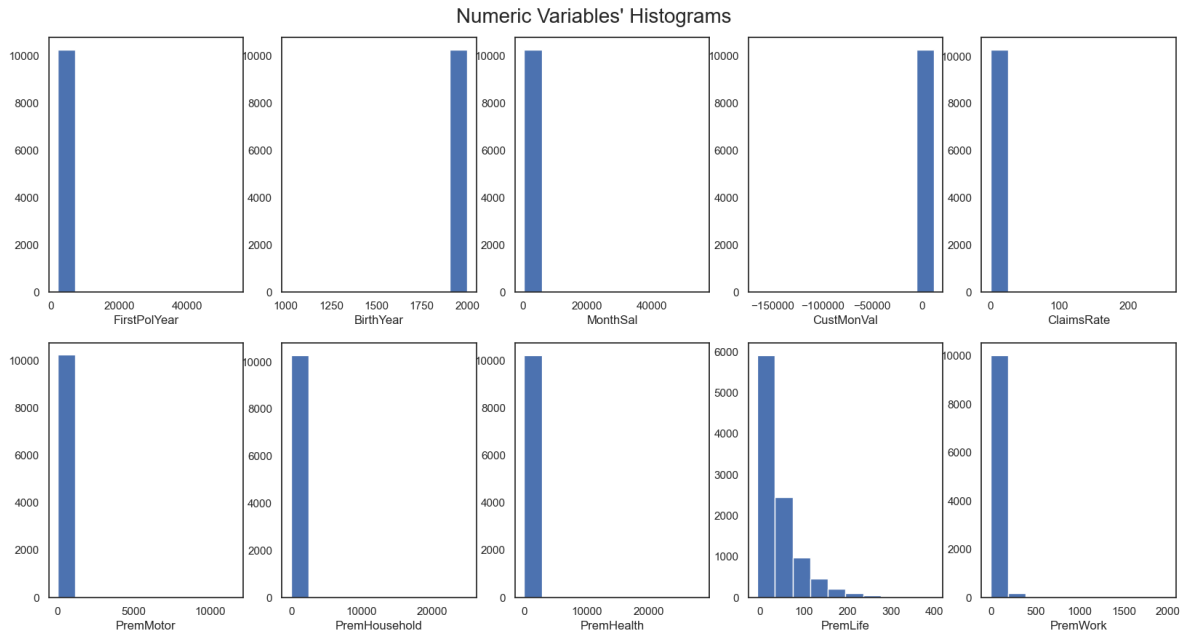

Figure 2 - All the dataset's numerical variables' initial histograms.

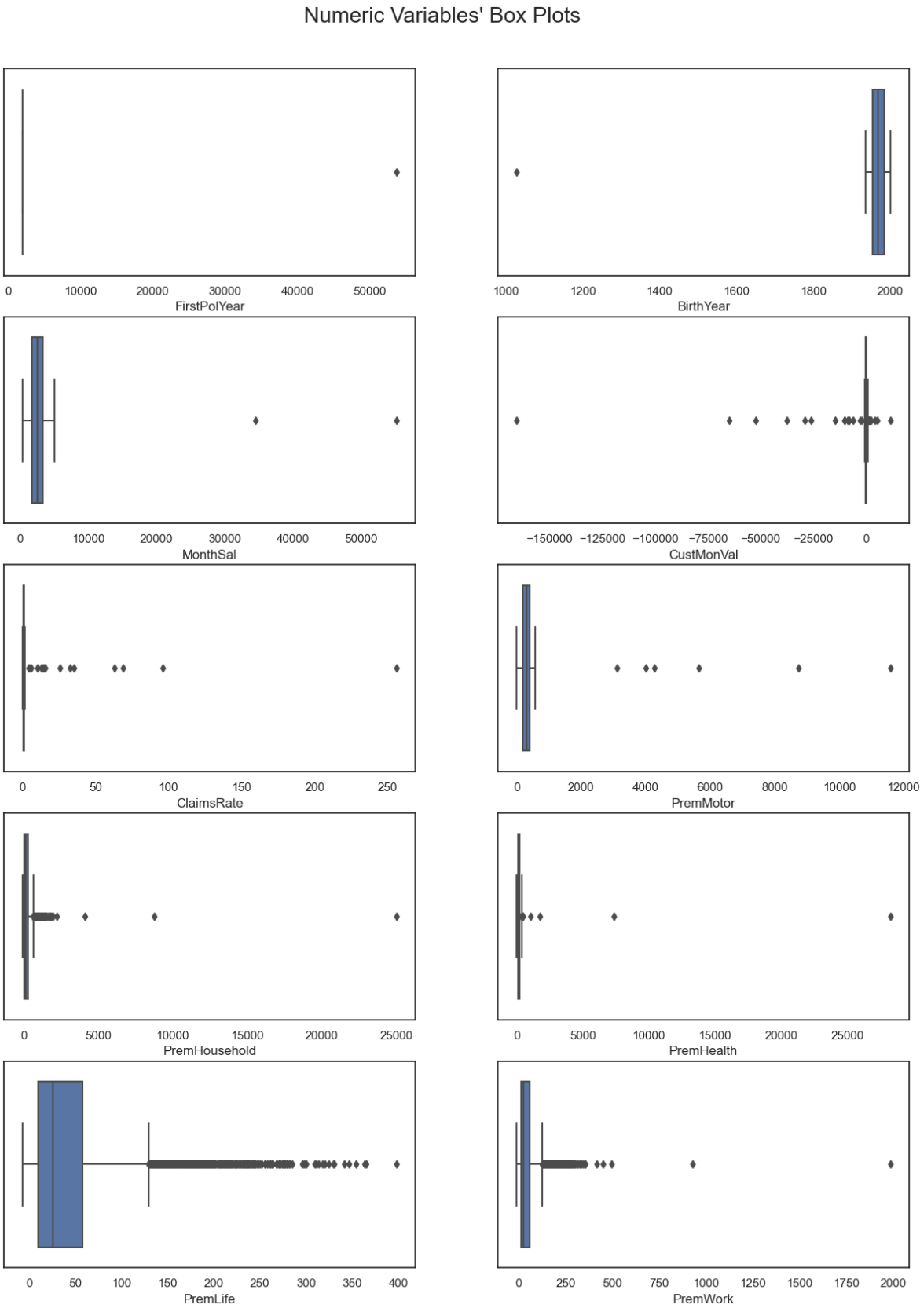

Figure 3 – All the dataset's numerical variables' initial boxplots.

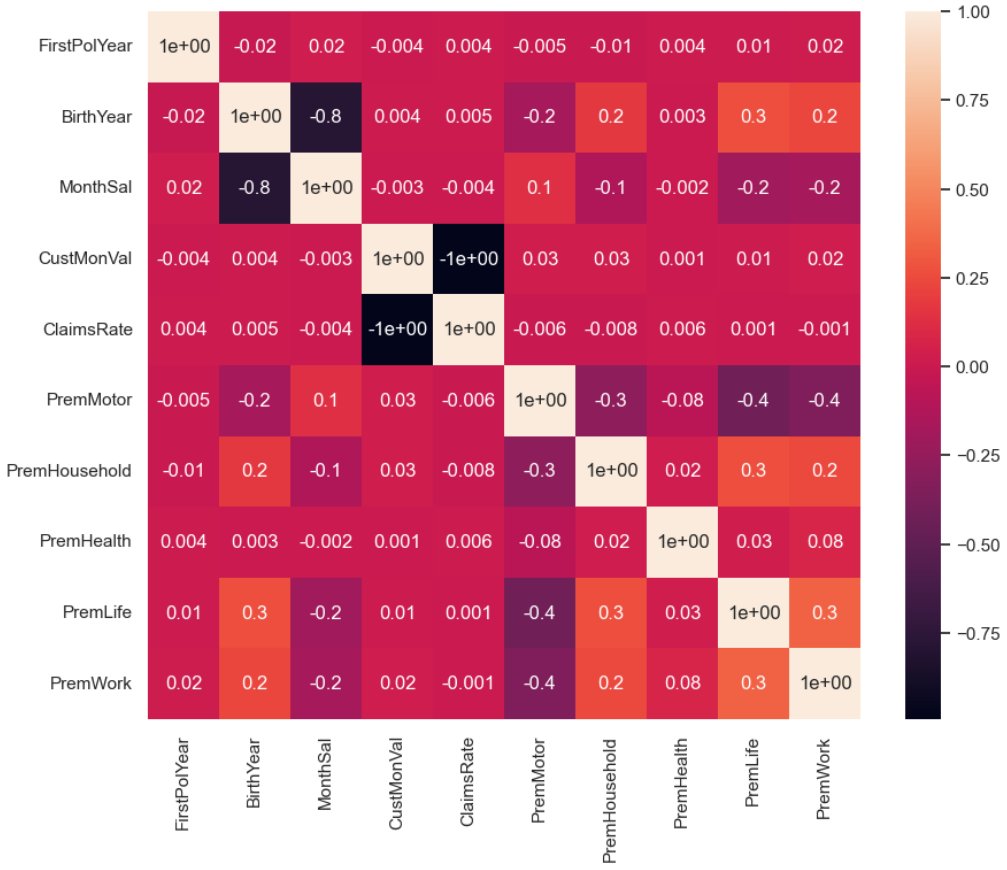

Figure 4 – Correlation matrix's heatmap for the initial numeric features.

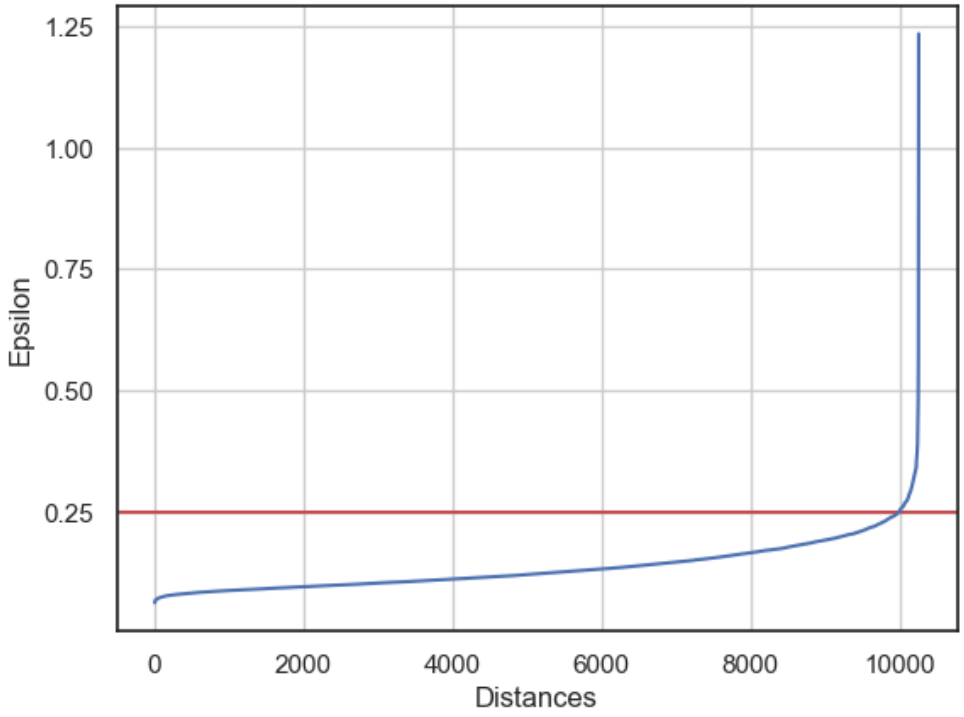

Figure 5 – Plot to use the elbow method on, to find the right parameter "epsilon" in DBSCAN.

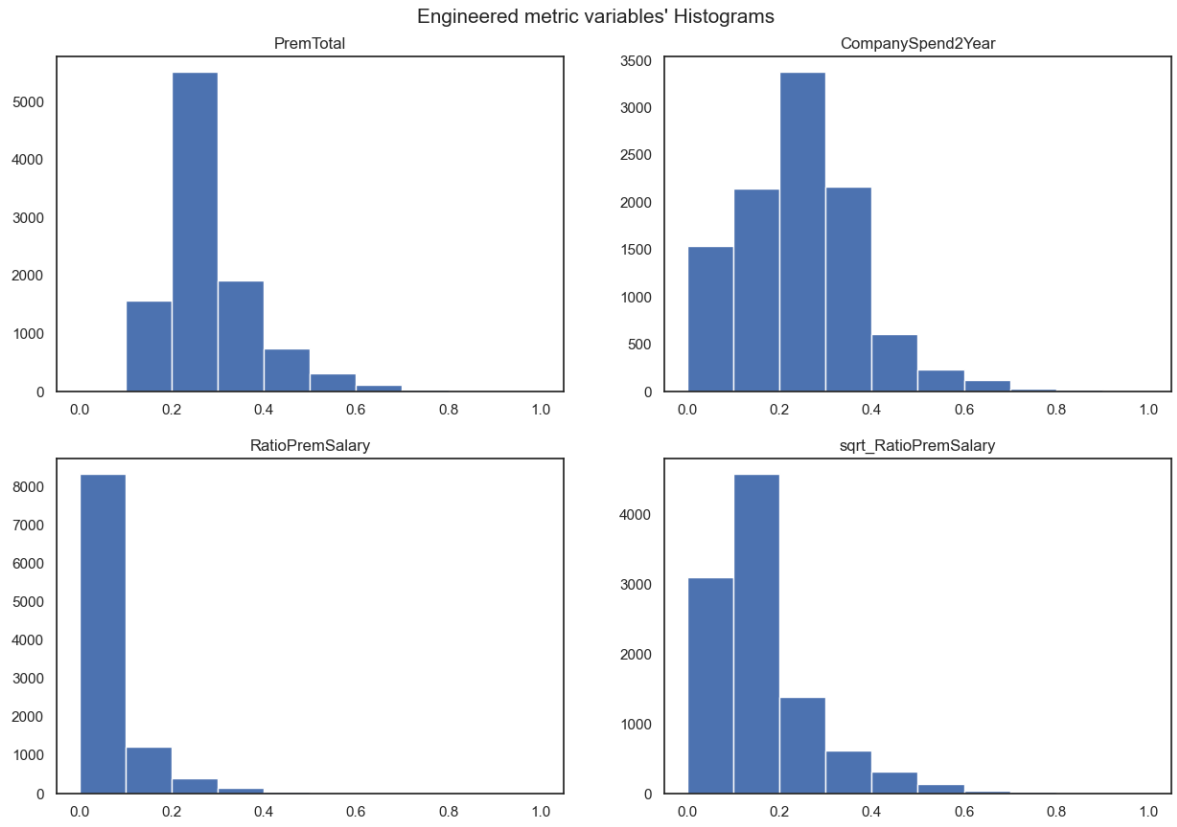

Figure 6 – Numeric created features' histograms.

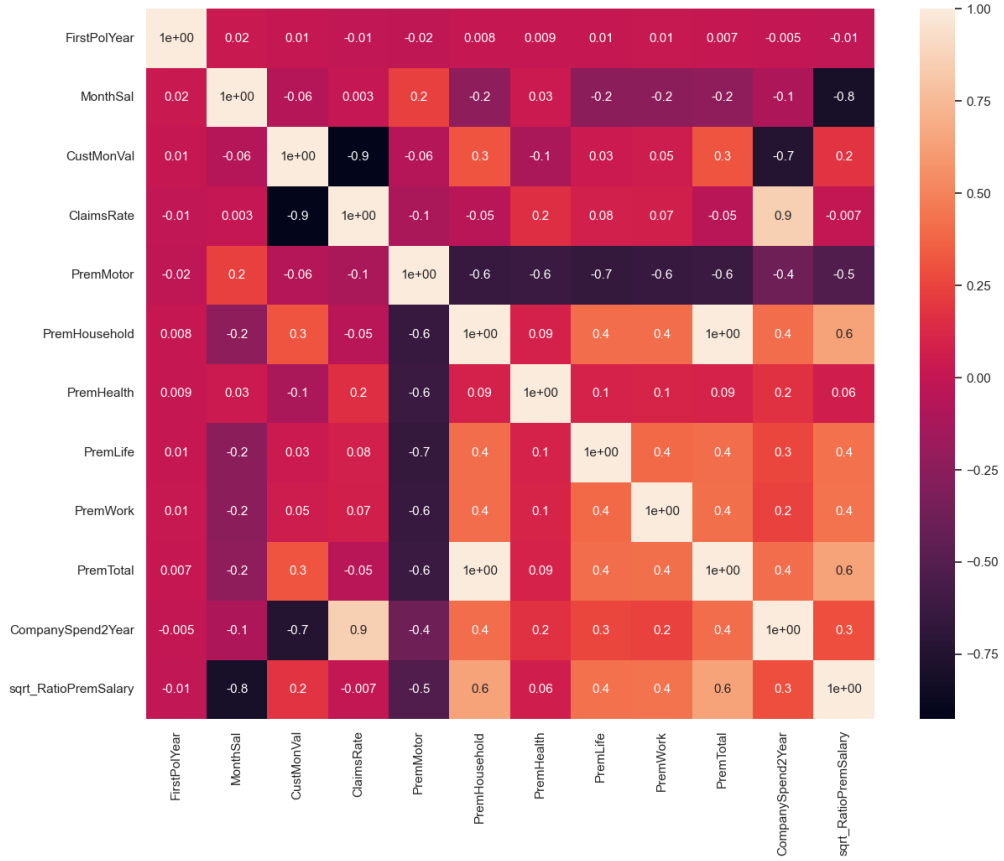

Figure 7 – All numeric features correlation matrix's heatmap, including engineered features

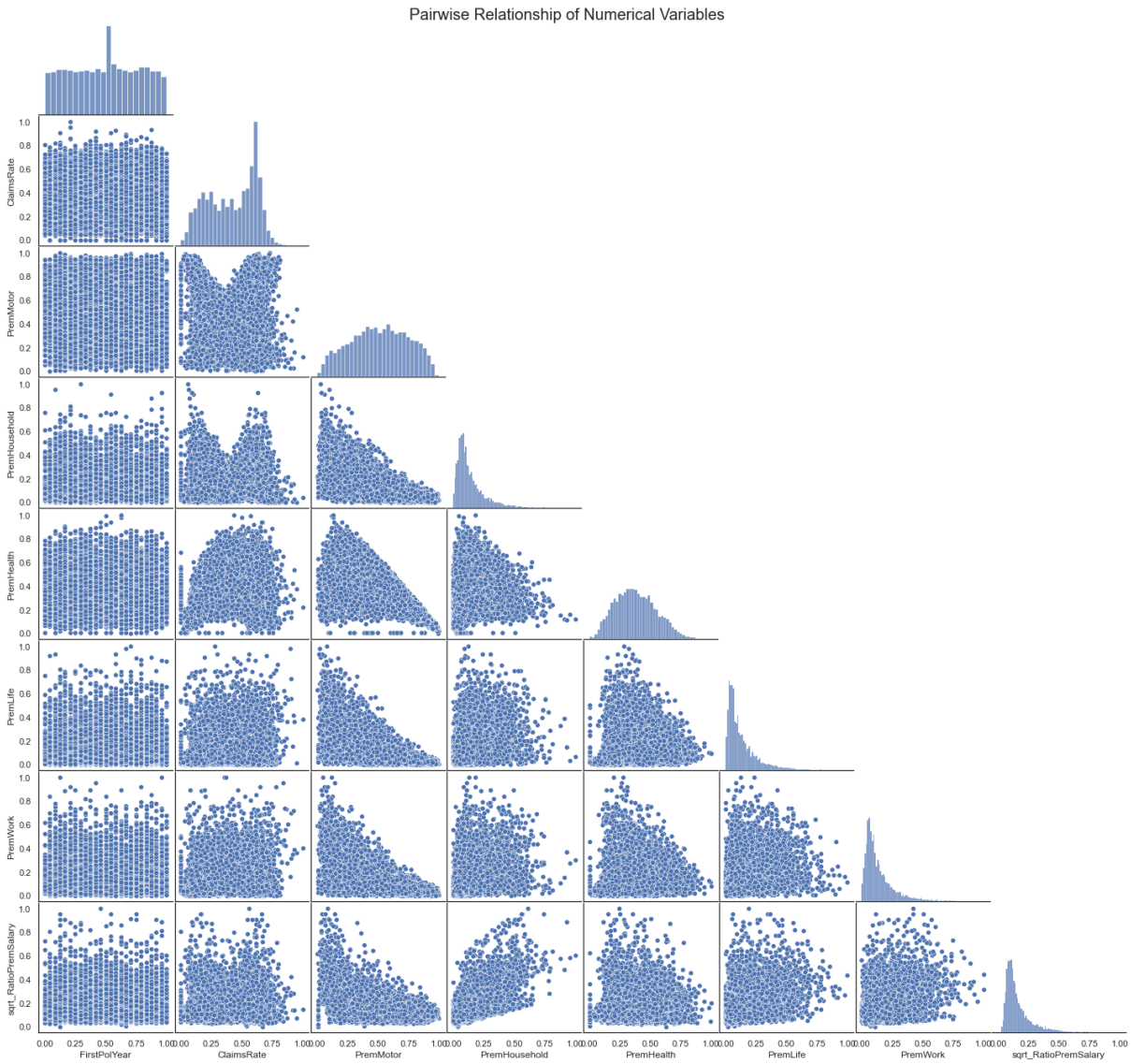

Figure 8 – Scatterplots of every metric feature vs every other metric feature. The plot is too big for the titles to be seen clearly, so the order of features is: FirstPolYear, MonthSal, CustMonVal, ClaimsRate, PremMotor, PremHousehold, PremHealth, PremLife, PremWork, PremTotal, CompanySpend2Year, sqrt_RatioPremSalary. Our focus is to see the shapes of the scattered data.

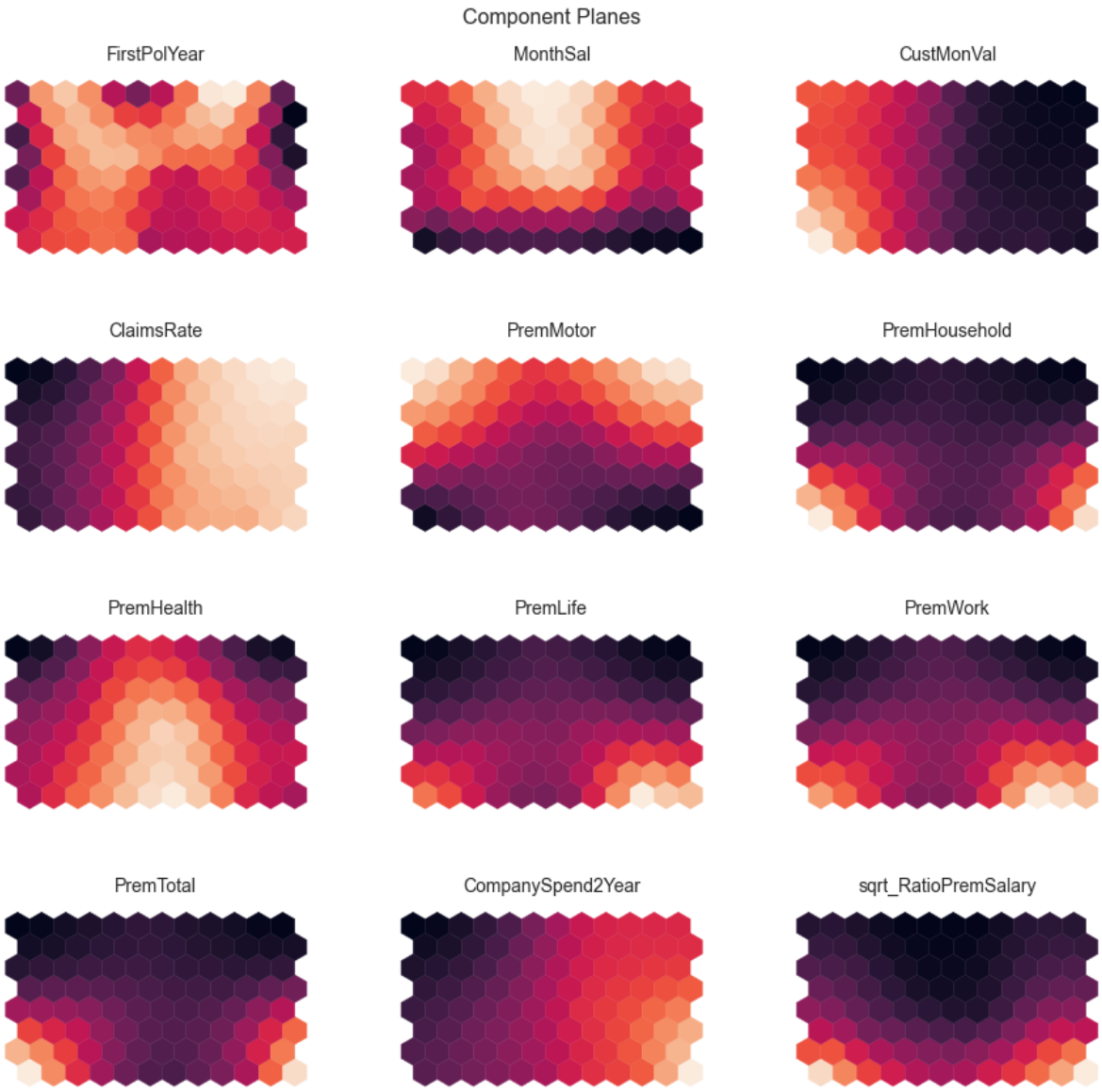

Figure 9 – Every feature's self-organizing map component plane.

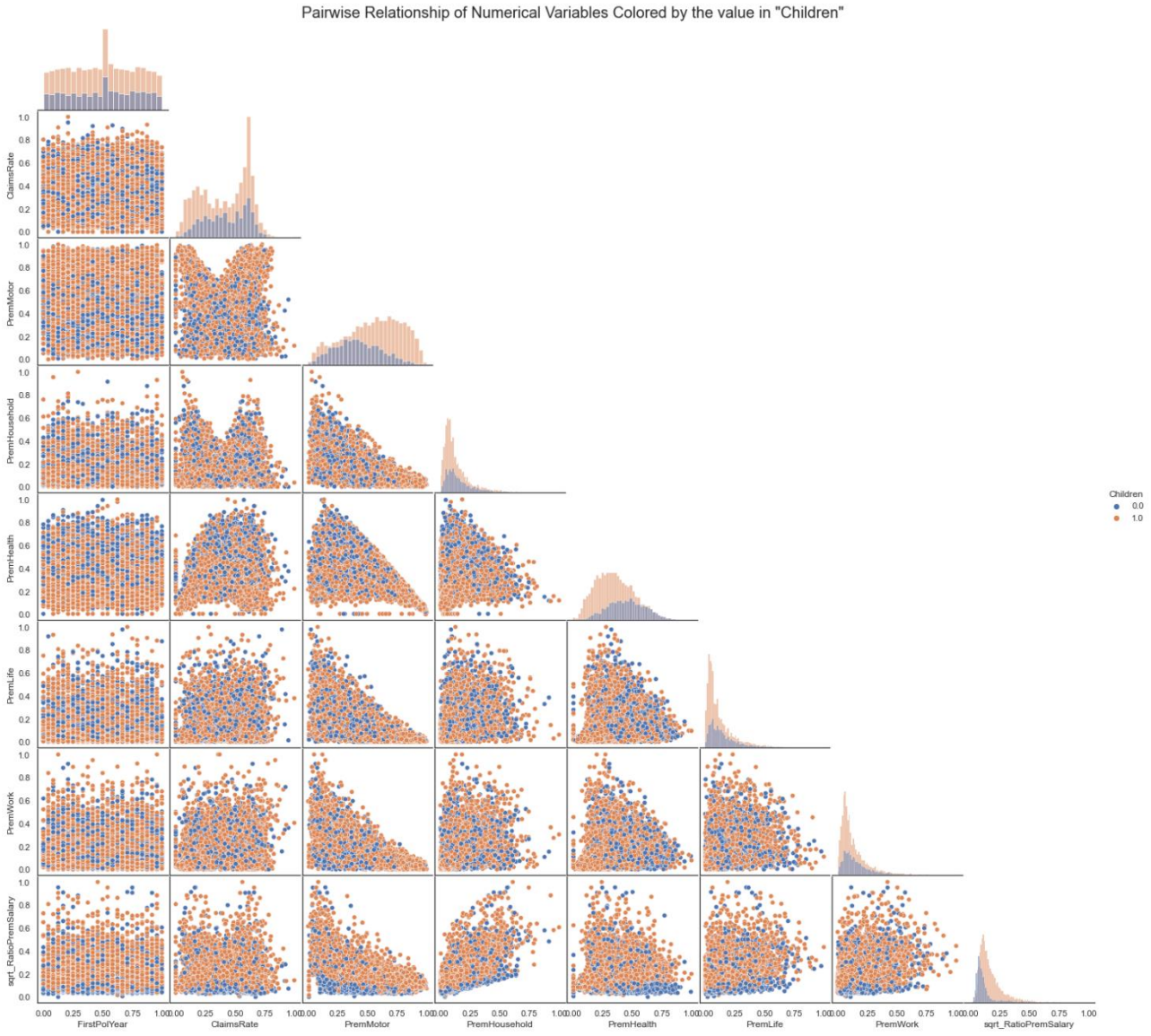

Figure 10 – Scattered points colored by their value in “Children”.

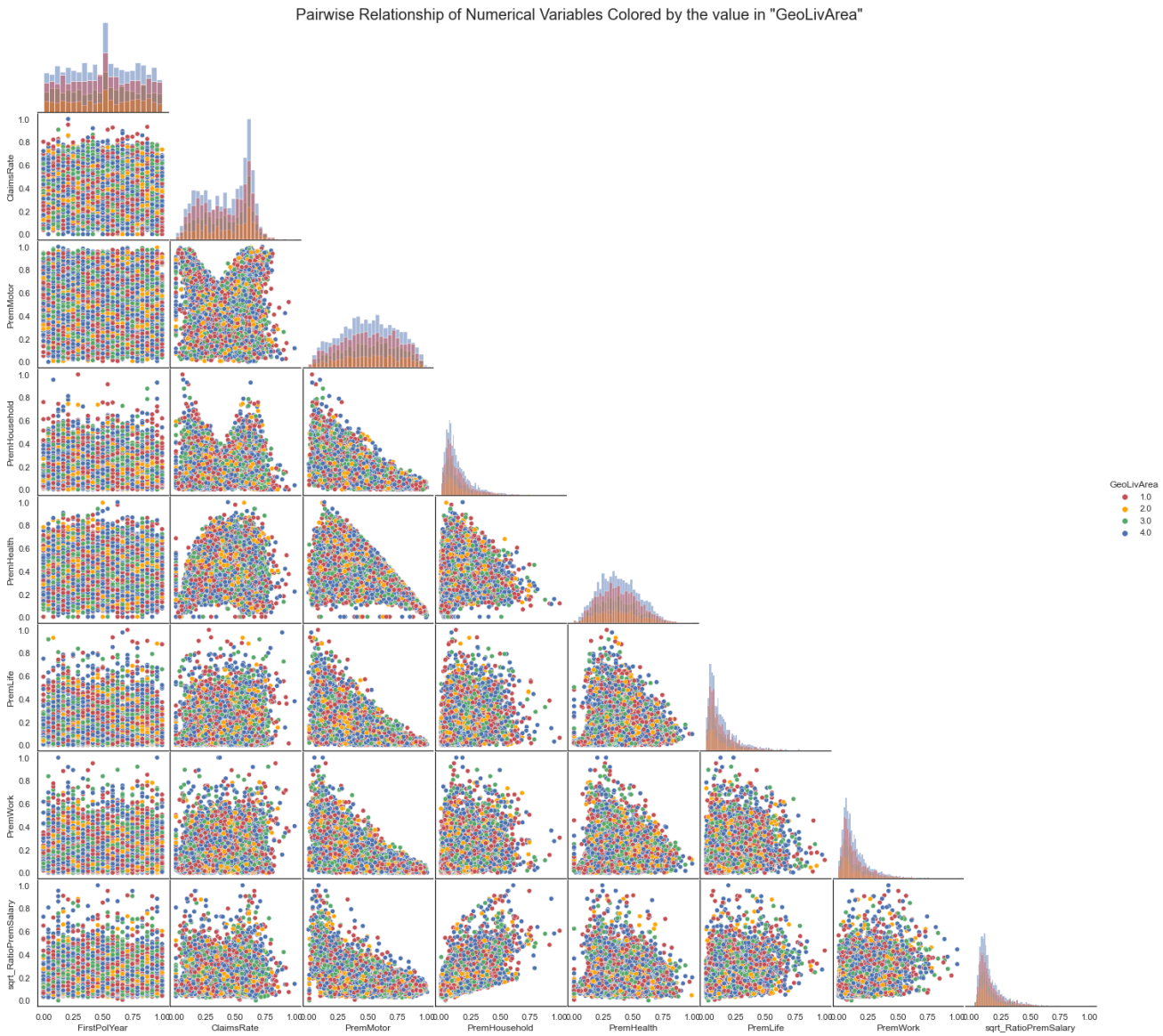

Figure 11 - Scattered points colored by their value in “GeoLivArea”.

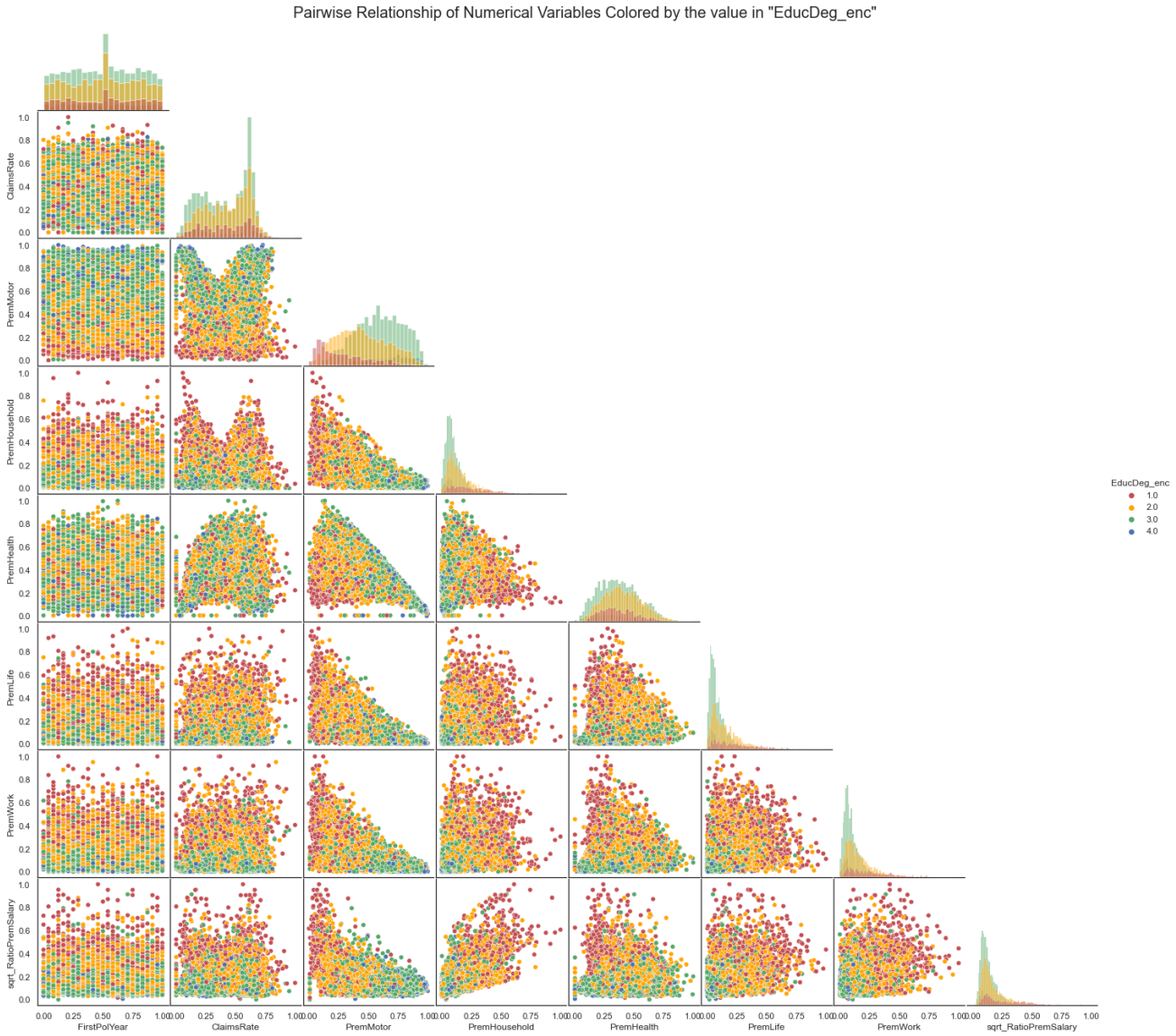

Figure 12 - Scattered points colored by their value in “EducDeg_enc”.

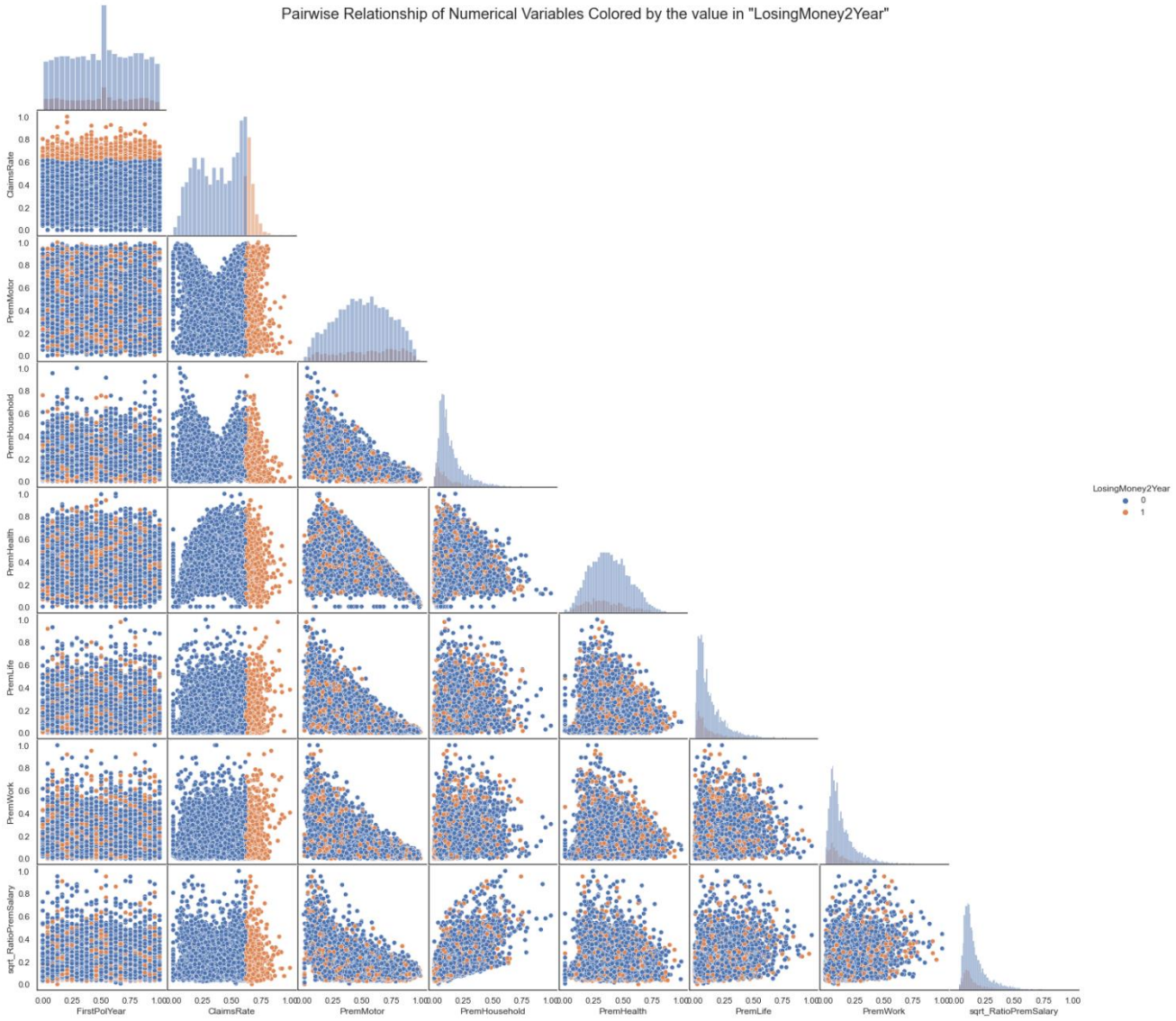

Figure 13 - Scattered points colored by their value in “LosingMoney2Year”.

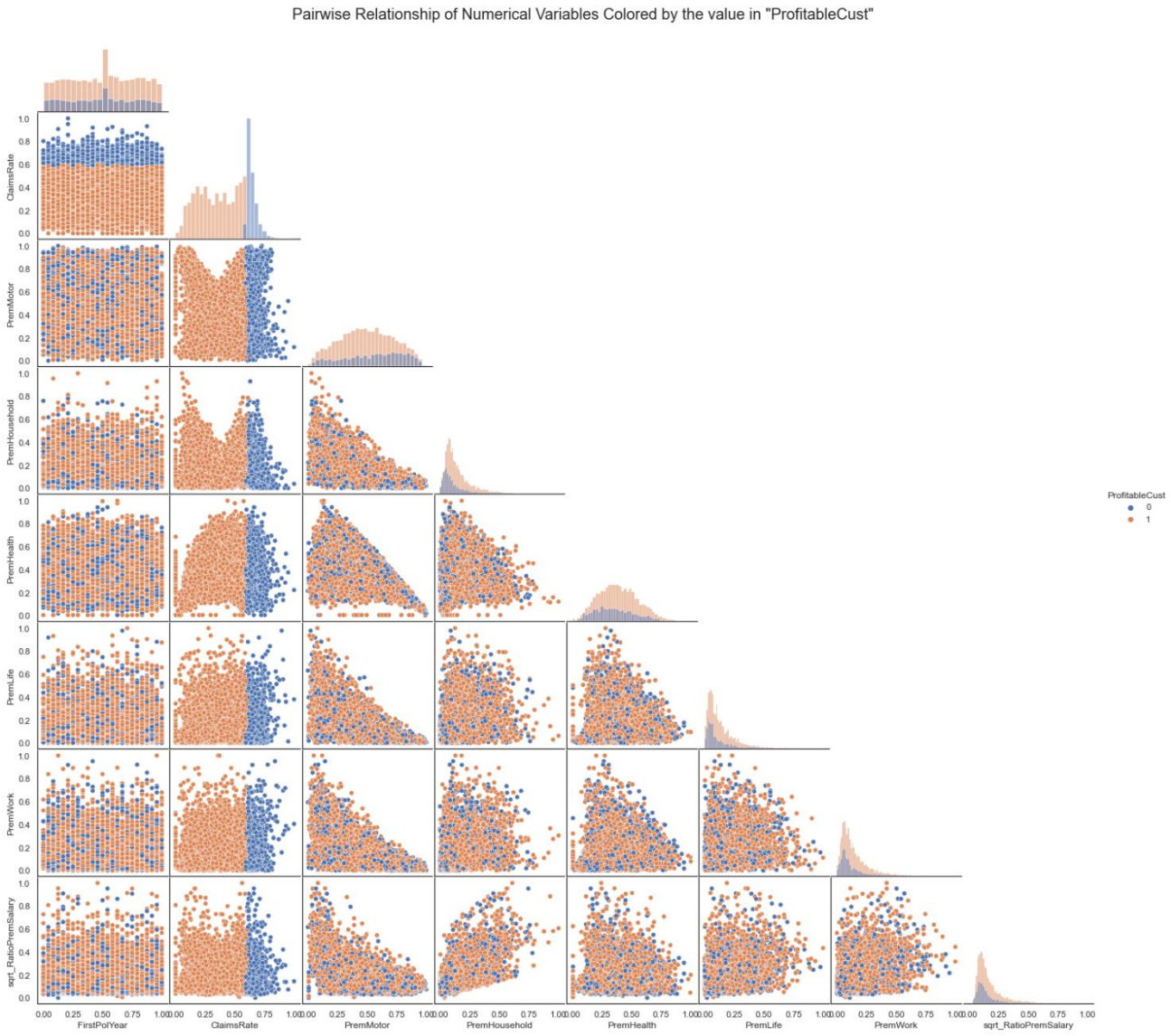

Figure 14 - Scattered points colored by their value in “ProfitableCust”.

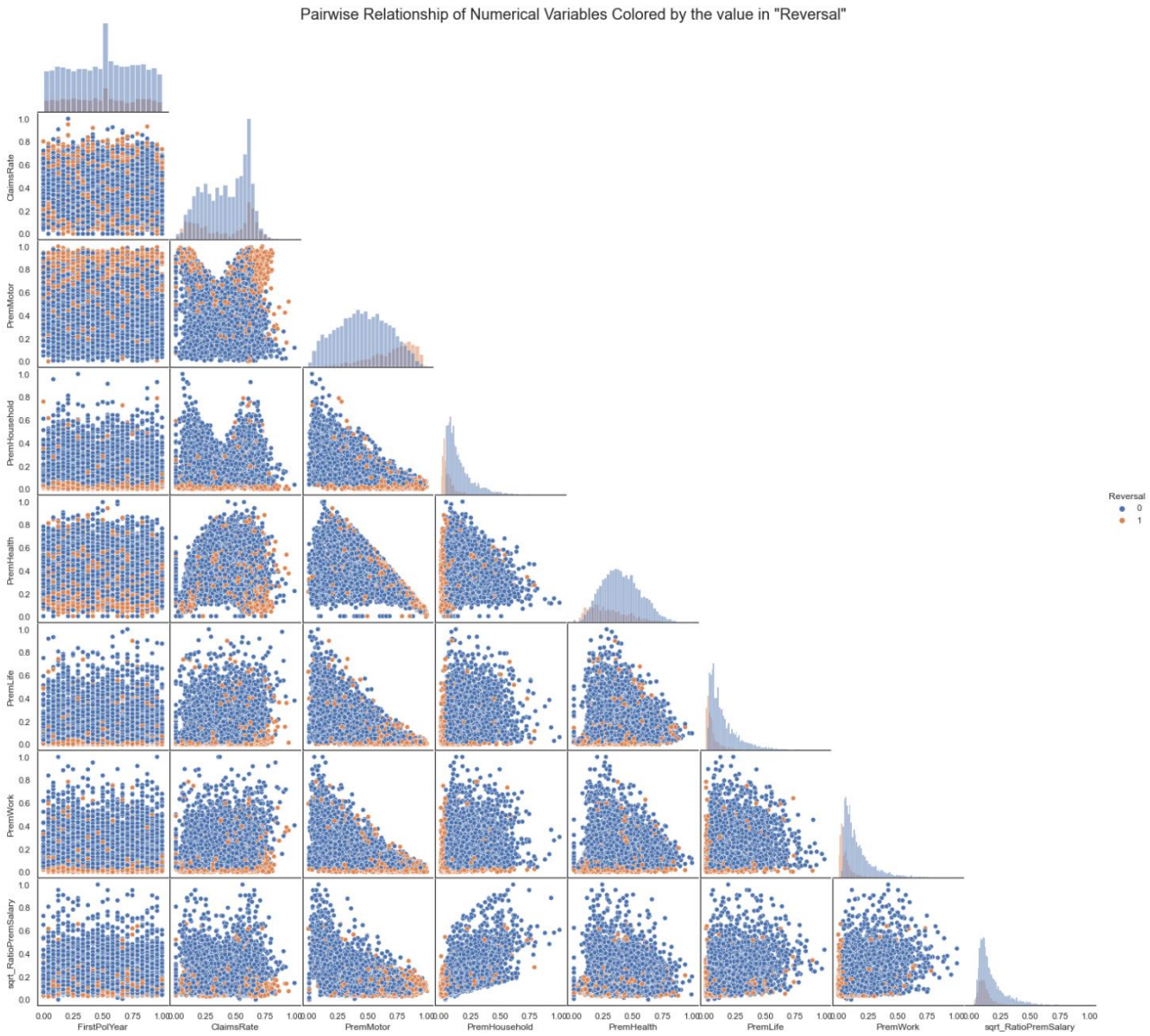

Figure 15 - Scattered points colored by their value in “Reversal”.

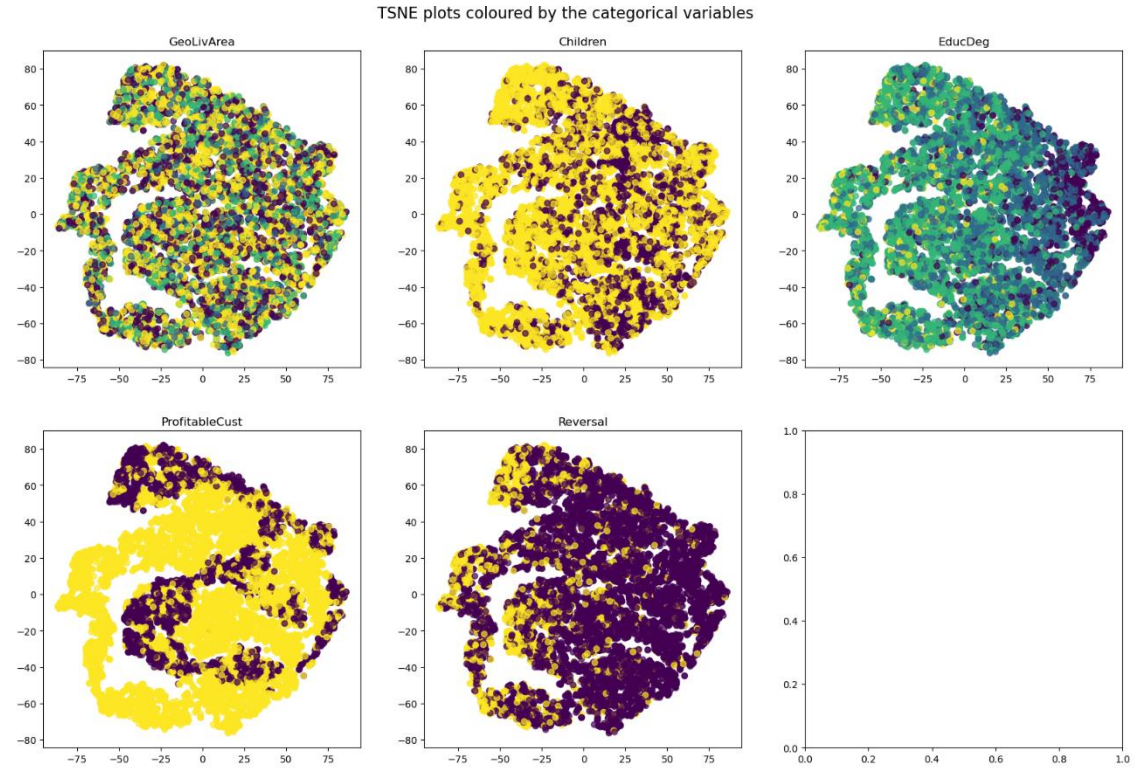

Figure 16 – t-SNE dimensionality reduction of the metric features after selection colored by corresponding category on every categorical feature left. Perplexity parameter set to 30.

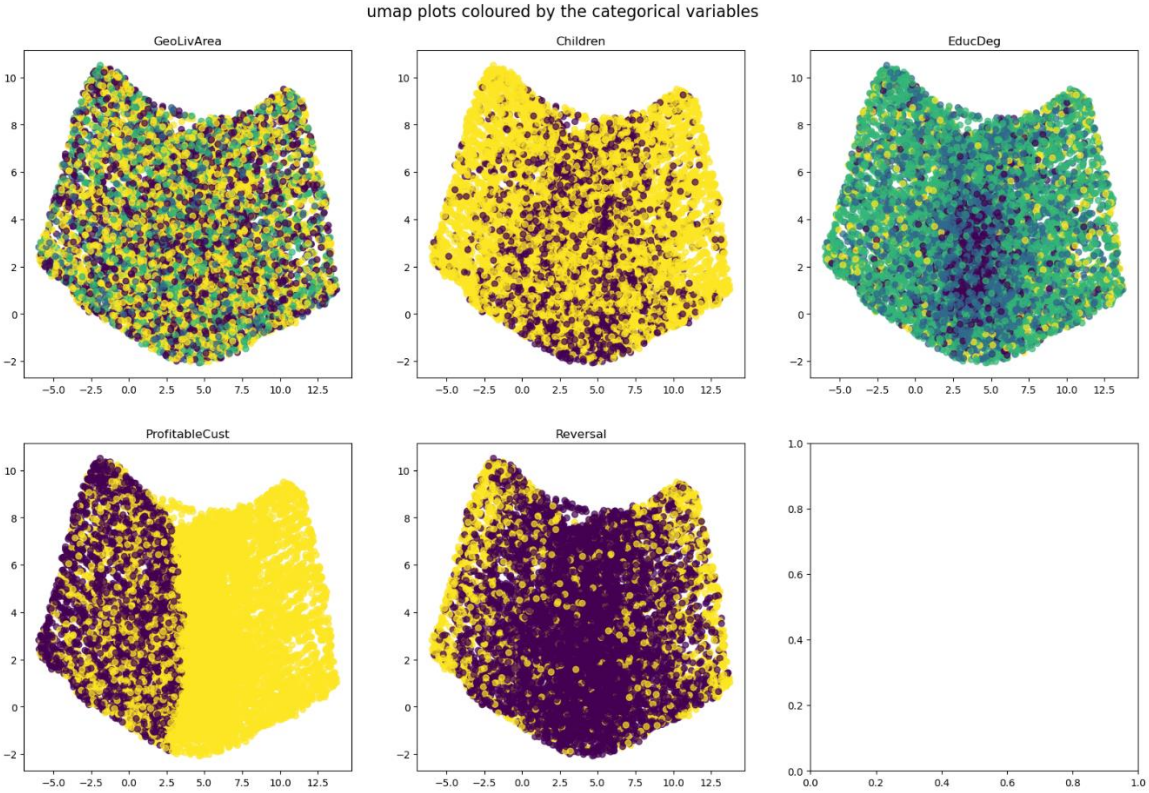

Figure 17 - t-SNE dimensionality reduction of the metric features after selection colored by corresponding category on every categorical feature left. Min_dist and n_neighbors parameters set to 0.5 and 30.

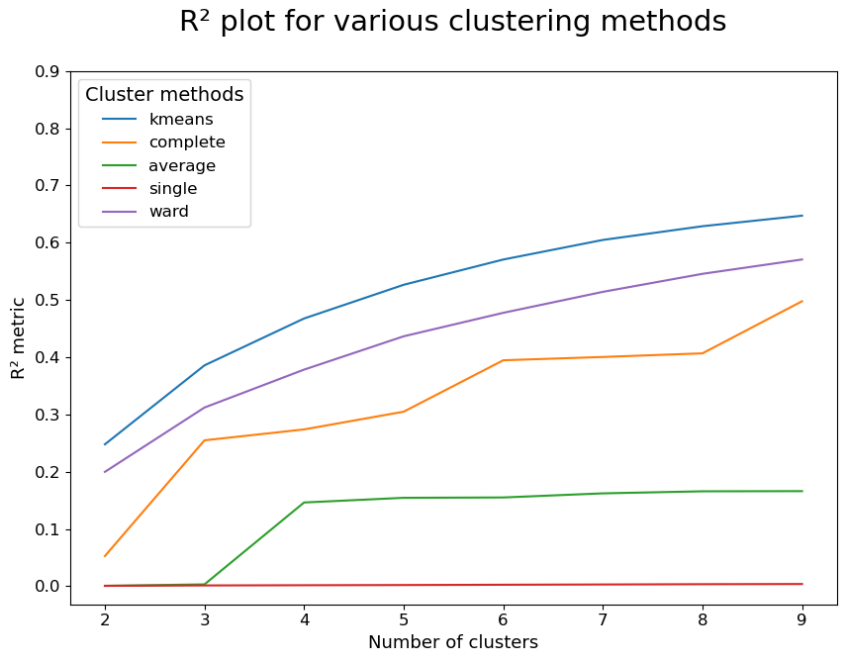

Figure 18 – R^2 plot for K-Means and Hierarchical Clustering with different linkage types against the numbers of clusters for all metric features.

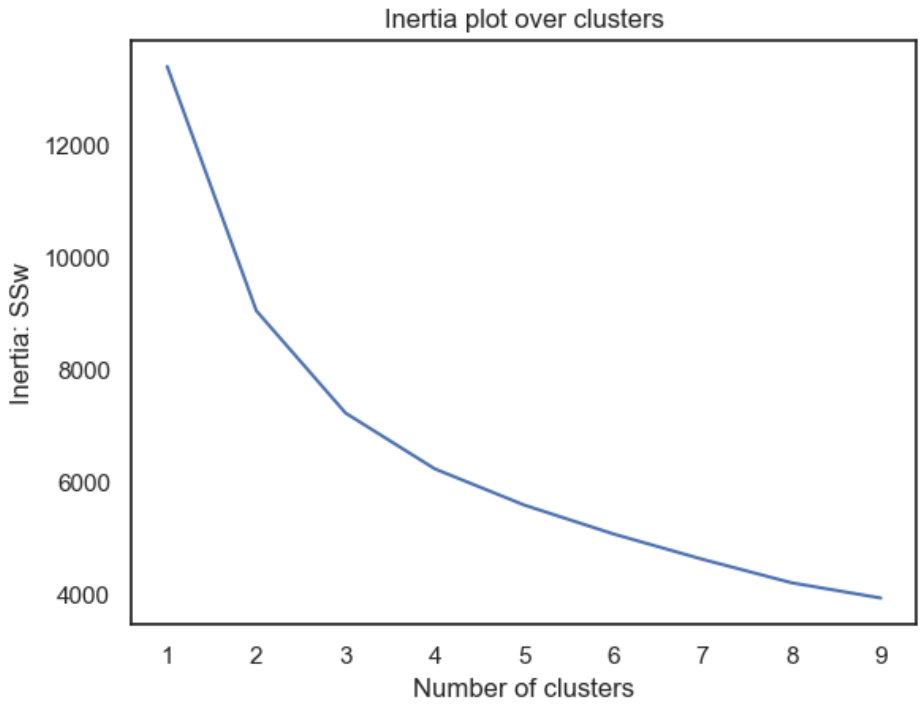

Figure 19 – Cluster inertia over the number of cluster when performing SOM.

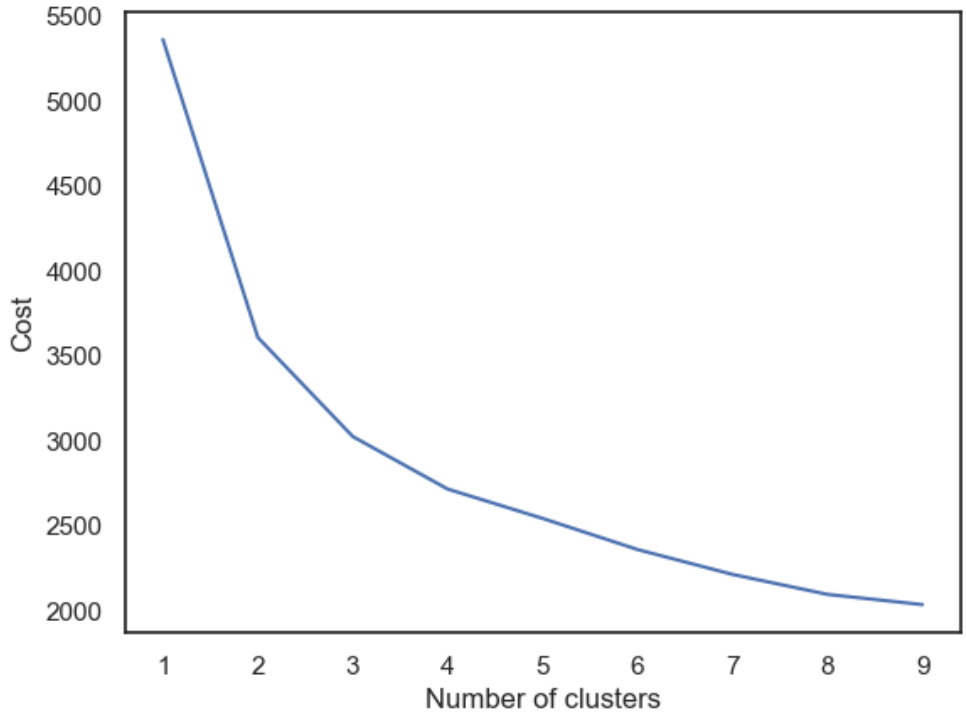

Figure 20 – Cost function over the number of clusters of K-Prototypes for all features.

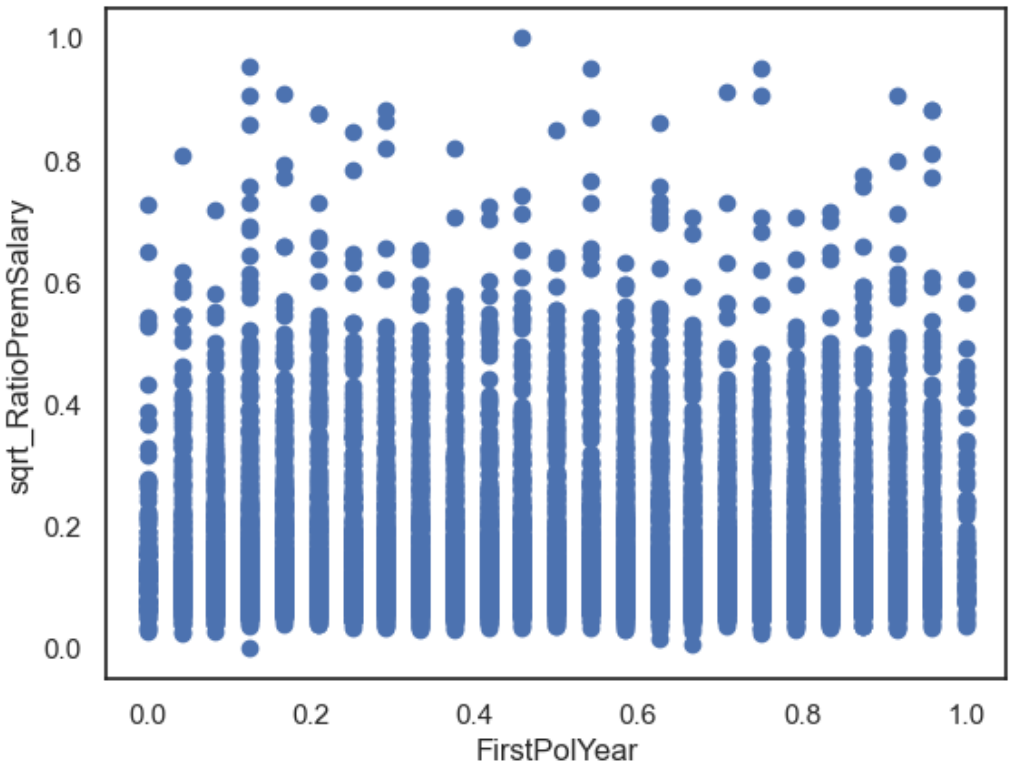

Figure 21 – “FirstPolYear” vs “sqrt_RatioPremSalary”

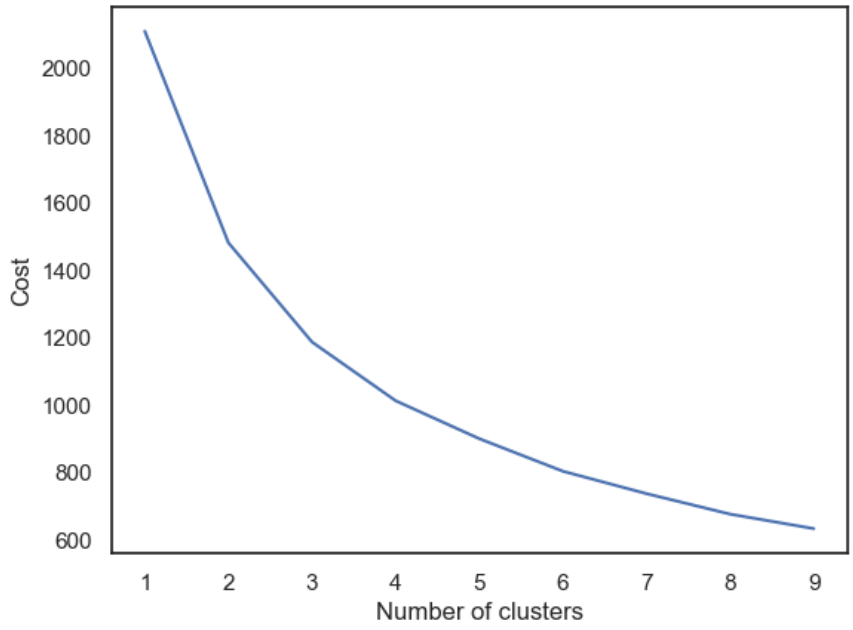

Figure 22 – Cost function over the number of clusters for all demographic variables

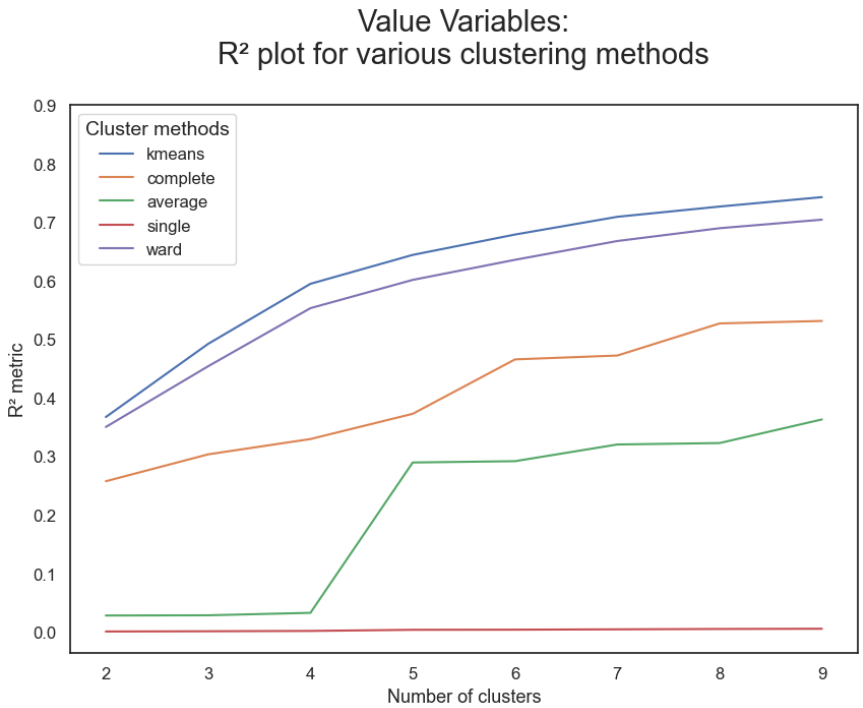

Figure 23 – R^2 plot for K-Means and Hierarchical Clustering with different linkage types against the numbers of clusters for all metric value features.

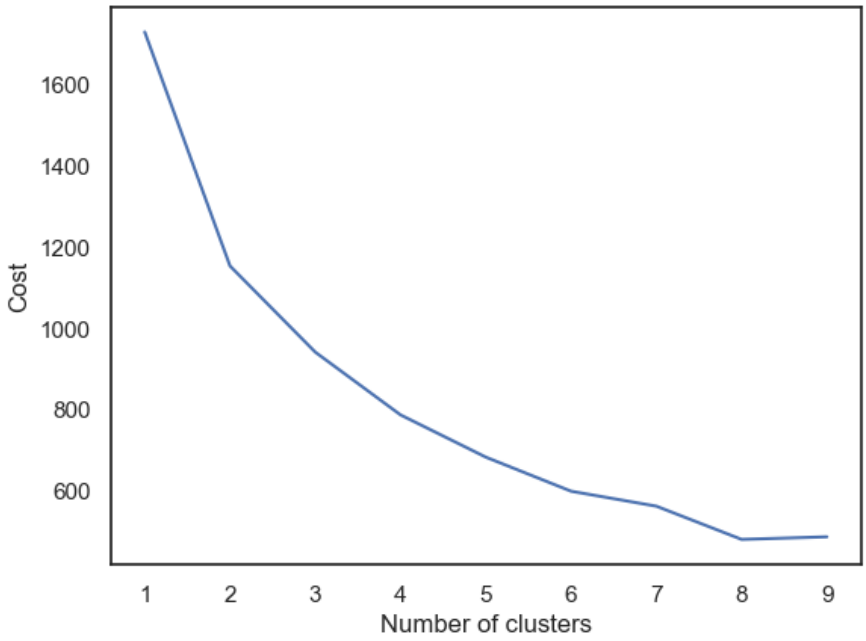

Figure 24 – Cost function over the number of clusters of K-Prototypes for all value features.

Figure 25 – Cluster profile for the solution with 8 clusters. On the left: Mean of all variables for each cluster. On the right: Size of each cluster

Figure 26 – Cluster profile for the solution with 4 clusters. On the left: Mean of all variables for each cluster. On the right: Size of each cluster

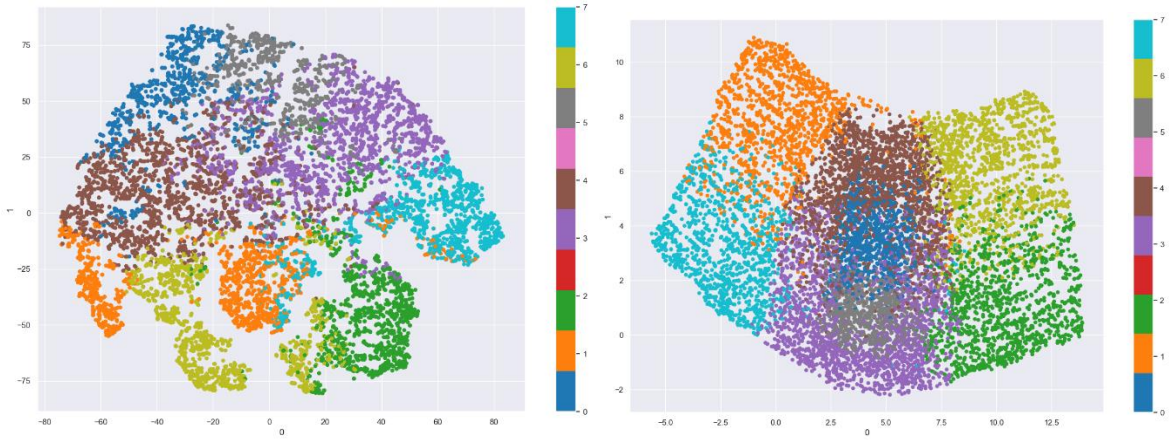

Figure 27 – t-SNE and UMAP representations of the final clustering solution.

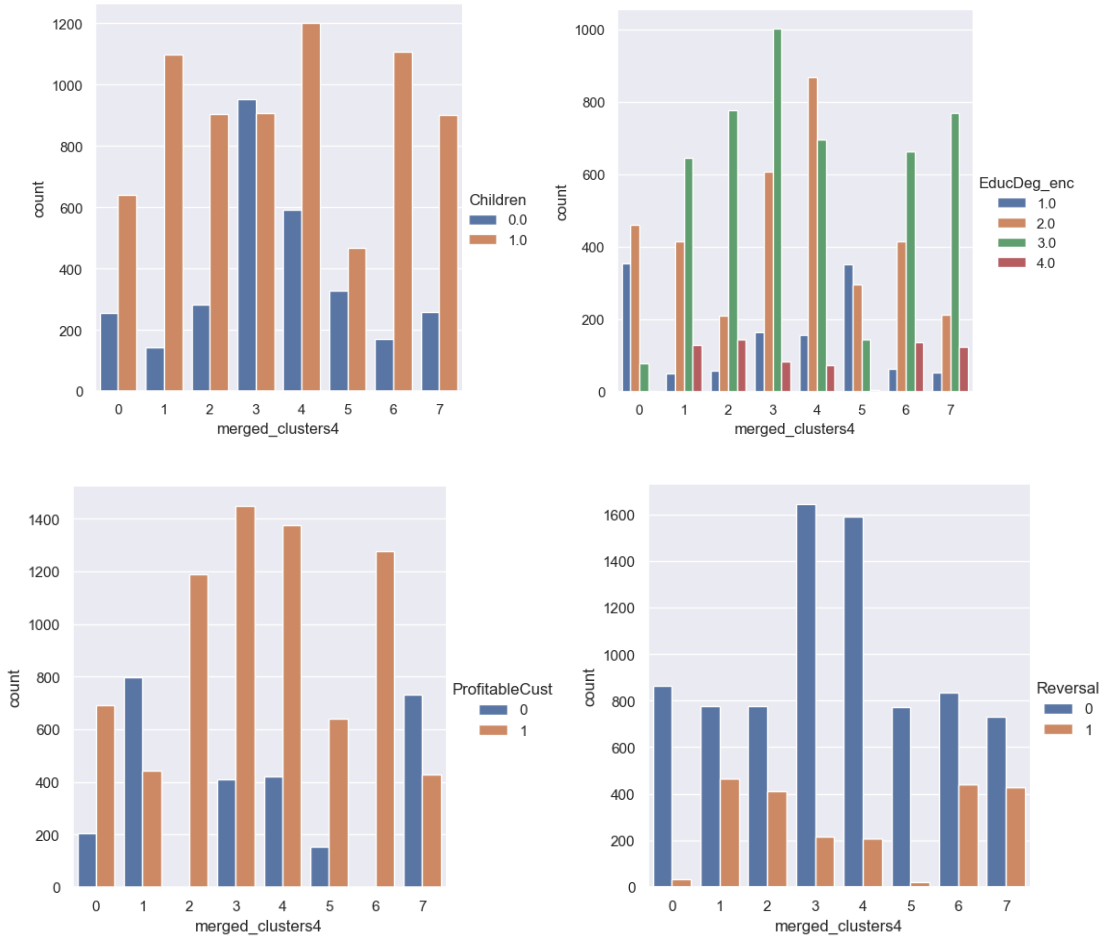

Figure 28 – Categorical features for the final clustering solution.

6.2. Tables

Variable	Variable Name	Meaning
CustID	Customer ID	Unique identifier of a customer
FirstPolYear	First Policy Year	The year the customer made their first insurance policy
BirthYear	Birth Year	Birth Year of the customer
EducDeg	Education Degree	The customer's level of formal education
MonthSal	Monthly Salary	Customers' monthly salary
GeoLivArea	Area	A number identifying a customer's living area going from 1 to 4
Children	Children	Binary variable which is 1

		if the customer has children (one or more)
CustMonVal	Customer Monetary Value or Lifetime Value	This variable has an amount which is calculated by subtracting a customer's acquisition cost to the multiplication of the annual profit from them and the number of years they are a customer
ClaimsRate	Claims Rate	Amount paid by the insurance divided by the premiums a customer pays, in the last 2 years
PremMotor	Motor Premium	Motor Premium paid in 2016 in euros
PremHousehold	Household Premium	Household Premium paid in 2016 in euros
PremHealth	Health Premium	Health Premium paid in 2016 in euros
PremLife	Life Premium	Life Premium paid in 2016 in euros
PremWork	Work Compensation Premium	Work Premium paid in 2016 in euros

Table 1 – Table with the original features and their meaning

Variable	**Number of outliers**	**Percentage of the observations kept after hypothetic removal**
FirstPolYear	31	99.7%
MonthSal	38	99.63%
CustMonVal	110	98.93%
ClaimsRate	15	99.85%
PremMotor	6	99.94%
PremHousehold	632	93.85%
PremHealth	26	99.75%
PremLife	650	93.68%
PremWork	632	93.85%
All of them	2140	84.96%

Table 2 – Table with the number of outliers by feature and what percentage of the dataset we would keep if we simply removed them.

Variable	Threshold	Reasoning
FirstPolYear	< 2017	The dataset is from 2016, there can not be customers with their first policy after that
MonthSal	< 30000	Huge gap between the customers who earned around 5000 and the very few ones earning more than 30000
CustMonVal	< 10000	*
ClaimsRate	< 200	*
PremMotor	< 2000	*
PremHousehold	< 10000	*
PremHealth	< 10000	*
PremWork	< 750	*

Table 3 – Table with the manual thresholds for every feature we decided to do one on to remove the most extreme outliers, with the reasoning for each one. Reasoning for "*" is the same for every feature and is the following: *Above this threshold observations seemed very separate from the rest of the dataset