Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

GROUP A – A2Z Insurance

Group A

Marta Dinis, number: 20220611

Patrícia Morais, number: 20220638

Samuel Santos, number: 20220609

December 2022

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

INDEX

1. Executive Summary

By analyzing the data available, we identified 5 large clusters of customers. While there is some overlapping, we characterized each cluster in terms of its members' sociodemographic and client value characteristics. For each cluster we created a persona that represents the cluster's members. We went on to describe these personas, identified some business opportunities and suggested a way to communicate with that segment. We also provided A2Z with valuable information regarding the customers that will have the most impact, either positive or negative, in their foreseeable lifetime.

We are confident that the information provided is both valuable and directly actionable by A2Z.

2. Introduction

2.1. Knowing our customer

A2Z Insurance is a top Portuguese insurance company that provides Motor, Household, Health, Life and Work Compensation insurances.

Its customers may be organized by demographic, geographic, psychographic and behavioral characteristics. These segments represent customers that are similar in some key features that can be actioned in future marketing efforts. For each segment it is also relevant to consider the customers' Lifetime Value.

2.2. Knowing our assignment

A2Z Insurance wants to find new customers and to keep current customers.

This team was assigned the task of extracting as much knowledge from existing data as possible. The company needs to have a better grasp of who its customers are, what main groups of customers can be found. This knowledge is crucial for maximizing the return on the marketing investment, by focusing on coherent customer groups and by better targeted campaigns that address both existing and prospective customers in a personalized manner, based on their characteristics and preferences. The company wants to move from a mass marketing approach to focused marketing programs, including cross selling and up selling of its products.

We are requested to deliver information on these customer groups, including its characteristics and the products that they might be more interested in buying.

The project is to be implemented in Python, without restrictions on libraries that can be used.

2.3. Knowing our data

A2Z Insurance has created several databases, and it has recently consolidated them in a single active Database. We expect that, because this database was created from poorly maintained smaller databases, there are bound to be several data quality problems that will need to be addressed.

We were provided a sample of 10.296 customers from the company's active Database. This data was exported in 2016.

3. Data Exploration

3.1. Input File Initial Handling

3.1.1. Importing Data

The sample provided was in SAS format. We used the pyreadstat library to read the file, after observing that the import directly from pandas was not perfect.

3.1.2. Setup

We started by setting the index on the dataframe as the costumer ID. Then, we defined lists for categorical and metric features. The variables *EducDeg, GeoLivArea* and *Children* are categorical features, and the remaining ones are metric features.

3.2. Preprocessing data

3.2.1. Duplicates

We found three pairs of records for which all its features were equal, except that they had different customer IDs. Knowing that we have several continuous variables in our dataset, including salary, year of birth, the premium for each insurance, we consider it unrealistic that two distinct customers would be exactly equal in each feature. Therefore, we dropped one observation out of each of these pairs.

3.2.2. Data Types

We checked the type of each feature and concluded that all of them were type 'float' (although some of them are categorical features), except for the variable *EducDeg*, which was type 'object'.

We changed the categorical variables' type to 'int'.

3.2.3. Handling Invalid Values

We checked the sample data for attributes whose value seems to be unplausible:

- First Policy's year cannot be higher than the date of the extraction (2016).
- Birth Year cannot be more than 100 years before the date of extraction.
- We consider that everybody has some degree of education; empty strings in feature Education Degree mean that we have missing data.

These features were considered to be invalid and were set to Null. They will be handled later on in the same way as with previously existing null values.

3.2.4. Handling Inconsistent Values

Some attributes have credible values, but they're not consistent with other values.

- Year of first policy cannot be before the birth year.

We assumed that the two fields were switched, so we switched them back.

3.2.5. Handling Null values

- Null values on insurance features:
  After counting Null values in each feature, we noticed that Household premium is sometimes 0, but it is never null. For the other 4 types of insurance, the opposite is observed: they are never 0, but sometimes they are null.
  Having missing information on an attribute as relevant for the company as the premium paid by the customer seems highly unlikely. We therefore decided that, in these 4 other insurance types, the null value is actually a zero value. We made the appropriate update.
- For categorical features, we replaced null values with the mode for that feature.
- For metric features, we replaced null values with the average from its 5 nearest neighbors.

3.2.6. Statistical Description

We then proceeded to use the function describe() to get an overview of our metric features' descriptive statistics.

- All the variables related to premiums have a very large range of values. The negative values on these features may represent settlement of payments.
- The same behavior seems to also apply to the features *CustMonVal* and *MonthSal.*

3.3. Feature Engineering

In order to make our data easier to interpret and also to find some other perspectives that could lead us to insightful information regarding our data, five new variables were created:

- *Years_as_Customer:* The difference between 2016 and *FirstPolYear*.
- *Age:* The difference between 2016 and *BirthYear*.
- *PremTotal:* The sum of all premiums for each customer.
- *PremHHL_propSal:* The proportion between the sum of all non-mandatory insurances and the customer's annual salary.
- *YearSal:* the customer's yearly salary.

Notice that some of these variables are built directly from another one, and therefore can be seen as their replacement. Because of this, some original features were removed from the dataset (*FirstPolYear*, *BirthYear* and *MonthSal*). Logically, we then updated our dataset accordingly.

Afterwards, we created three lists representing our main feature groups, namely:

- Sociodemographic features – features related to the customer's social and demographic characteristics (Education degree, age, if they have children, yearly salary, PremHHL_propSal and their geographical location).
- Client Value features – features pertaining to the customer's value to the company (motor, household, health, life and work premiums, total value of premiums, number of years as customer, customer monetary value and claims rate).

This segmentation is made in order to compare its performance with the clustering with all features, but mainly because it is useful for our client to have these two perspectives.

3.4. Visualizing Initial Data

After these first operations, we used visualizations to look at how data is distributed. We can see the numerical variables' histograms and boxplots in Figure 1 and Figure 2, respectively. A pairplot for the numerical variables was also plotted, using the binary feature as the color hue (Figure 3).

3.5. Removing Outliers

The boxplots and histograms plotted previously make it very clear that we have a very large number of outliers, and many of them are in fact extremely far from the mean for the feature. After some experimentation, we considered as outliers the observations that check the following conditions:

- The percentage of the customer's yearly salary that goes into paying insurance is smaller or equal to 20%. We consider that spending more than that on insurance is unrealistic and, therefore, an extreme occurrence.
- The customer's yearly salary is larger than 200 000€.
- The amount spent on motor insurance is larger or equal to 2000€.
- The amount spent on health insurance is larger or equal to 2500€.
- The amount spent on household insurance is larger or equal to 2500€.
- The amount spent on work insurance is larger or equal to 1000€.
- The customer's monetary value is smaller than -2500€.
- The customer's claims rate is larger than 4.

After removing outliers, we kept 99.14% of the initial data.

Throughout this initial data analysis, we realized that some of the values that would normally be considered outliers might actually be of great interest to the company, as some of them refer to clients that are extremely high spenders, while others make the company lose money. For this reason, and because we didn't want to lose too much data, we decided that these customers should be kept in the dataset. Instead of erasing these clients, we created three customer categories, separating 'average', 'bad' (8,5%) and 'good' (10,2%) clients based on how much we expect to profit in their lifetime.

After this step we replotted the numerical variables' histograms and boxplots, as seen in Figure 4 and Figure 5, as well as the respective pairplot in Figure 7. We also plotted a bar chart for the categorical variables, which we can see in Figure 6.

3.6. Feature Selection

As we know, the curse of dimensionality is a phenomenon that arises when analyzing data with too many variables. In order to avoid this issue, it's important to make some thoughtful selection in regard to which features to keep. There are several aspects to consider when choosing variables to remove. First, we must consider variable relevancy, that is, if a variable has no significant correlation to any of the other features, then it is considered 'irrelevant'. It's also crucial to avoid redundancy, in the sense that no two variables kept should have extremely high correlation. To fulfill these requisites, we looked at a heatmap of the metric variables' Kendall correlation (Figure 8) and then took the appropriate

conclusions, removing the following variables: *ClaimsRate* (extremely correlated with *CustMonVal* and no significant correlation with any of the other features) and *PremTotal* (high correlation with *PremHousehold*). We specifically used Kendall's correlation coefficient because, not only the underlying assumptions necessary to use Pearson's correlation are not satisfied, but also this coefficient is more robust than the Spearman's correlation coefficient.

After the feature selection, we replotted this heatmap and concluded that the correlations between the variables kept remained similar.

3.7. Data Encoding

The variable *EducDeg* has four unique values that come in a string format ('1 - Basic', '2 - High School', '3 - BSc/MSc', '4 - PhD'), which we transformed into the integer values corresponding to the first character in the string.

Since none of the clustering methods we used required one-hot-encoding for the categorical variables, we didn't perform this step.

3.8. Data Scaling

On a dataset with such different variables that have such varied ranges, it is important to scale our data in order to apply our clustering algorithms. With this in mind, and considering that our features don't seem to follow a gaussian distribution and that we have a large number of outliers, we applied a RobustScaler() to the metric features in our dataset.

4. Clustering

After this initial data exploration and analysis, we moved on to data clustering. We experimented with various clustering algorithms, and their combinations, in order to find the best solution possible. Furthermore, we also tried two different approaches: with and without data segmentation, using the feature categories described previously (sociodemographic, insurance and client value).

To compare our clustering methods, we implemented a function that calculates the R^2 for each clustering.

4.1. Hierarchical + k-prototypes without segmentation

In this first approach, we applied a combination of Hierarchical and k-prototypes clustering.

Hierarchical clustering is a method that seeks to group data points based on a similarity metric, thus creating a hierarchy of clusters. Although it is a widely known clustering algorithm, it rarely provides the best solution, as it works badly with large datasets and mixed data types (which is the case). Additionally, the dendrogram – the visualization used for its interpretation – is easily misconstrued. With this in mind, we decided to use this algorithm with the sole purpose of choosing the number of clusters to consider for the k-prototypes clustering.

K-prototypes is an improved version of the k-means algorithm, that can work with both numerical and categorical data. The k-means algorithm starts by identifying k centroids, and then proceeds to associate every data point to the nearest cluster, all the while trying to minimize the intra-group variance of the clusters. K-modes works in an identical way but using the modes for centroid allocation. The k-prototypes method can be seen as a combination of the k-means and k-modes algorithms, as it uses the mean as the centroids for the numeric features and the mode for the categorical features. The k-means method is very sensitive to noise and outliers, as these can substantially influence the mean value. Furthermore, we have a high-dimensionality input space, which is also a weakness for algorithms that use distance as a metric, such as this one. Therefore, we expected this clustering method to produce poor clustering solutions; however, this wasn't the conclusion we took after applying this clustering method to our dataset. From the dendrogram, we concluded that we should apply k-means with 5 clusters, from which we achieved an R^2 of 0.54, which is unexpectedly high.

We also experimented with more and less clusters, but the clustering wasn't much better: less clusters caused a significant drop in the R^2 value and more clusters did not increase the R^2 value significantly.

4.2. Hierarchical + KMeans/KPrototypes with segmentation

In this second attempt, we performed a similar, but slightly different approach.

This time we executed the clustering for each feature category separately. Also, we started by using the k-means/k-prototypes (depending on whether the feature list had categorical variables or not) with a high number of clusters and proceeded to create a dendrogram through hierarchical clustering using the centroids of the clustering we first obtained. This allowed us to reduce the number of data points used to perform the dendrogram, thus creating a more perceptive and insightful visualization. From this dendrogram, we chose what seemed to be the appropriate number of clusters for each feature category, which we used to apply the k-means/k-prototypes clustering once more.

For the **Sociodemographic** features, we inferred from the dendrogram that 3 was the appropriate number of clusters to consider, which led to an R^2 score of 0.45. Although this isn't a very low value, methods applied later on produced significantly better results.

As for the **Client Value** category, we also chose to use 3 clusters (according to the dendrogram seen in Figure 9), leading to an R^2 value of 0.54. Surprisingly, this clustering method led to the biggest R^2 value we were able to obtain in this category. We can see the resulting cluster boxplots for this cluster solution in Figure 10.

4.3. DBSCAN

DBSCAN is a density-based clustering method that is designed to '*model clusters as dense regions in the data space, separated by sparse regions'*[1]. In this algorithm we are not required to choose a number of clusters up front, and it identifies outliers as noise, as well as being able to identify clusters of arbitrary shape, contrary to the previous algorithms we used. This method's principal disadvantage is that it is quite hard to implement, as it is highly sensitive to its two main parameters: ε, the radius

[1] Jiawei Han, J.H., Micheline Kamber, M.K., Jian Pei, J.P., Data Mining. Concepts and Techniques, 3rd Edition

defining the neighborhood of each data point, and *minPts*, the minimum of points in the ε-neighborhood.

After a lot of experimenting with the hyperparameters, the major finding is that DBSCAN is not adequate for our model. When we reduce ε we get a lot of clusters, but most of the observations are just in one cluster. When we increase *minPts*, we reduce clusters but significantly increase outliers (cluster=-1). This was the case both when we segmented our data by feature category and when we didn't.

4.4. Self-Organizing Maps

A Self-Organizing Map is an unsupervised neural network that is very well adapted for clustering, although it can also be used for other purposes, such as data visualization and outlier detection.

The self-organizing map is a grid of perceptrons, or units, that are spread through an n-dimensional space. This can be a 2 or more dimensional space but will not have higher dimensionality than the data in the input space.

During the training phase, units and input patterns are compared, establishing the closest unit as the winning unit. The closest unit is the one that is more similar to the data point, therefore, it's the best representation for that data.

The winning unit is then pulled towards the input data that it represents. The distance covered in this adjustment is higher (becoming closer) or lower (moving slower) depending on the learning rate for that epoch. The learning rate decreases along the epochs, allowing the map to keep adjusting, but making smaller adjustments as the SOM is fitting the data.

When a unit is moved, this also impacts the nearest neighbor units, because they're interconnected, but by a smaller degree. The impact on the neighborhood is also a hyperparameter that we adjust (reduce) during the training process.

We started by applying the SOM algorithm to the full dataset, that is, without segmentation, in order to get a reference result for the clustering.

After applying this clustering method to both categories from our variable segmentation, we obtained the following results:

- For the **Sociodemographic** features, the UMatrix (Figure 11) suggests the existence of 3 to 4 clusters. After experimenting with both, we opted for the best R^2 value – 0.80 – corresponding to 3 clusters. This was our best clustering solution for this category. In Figure 14 and Figure 15 are represented both the boxplots (for numerical variables) and bar charts (for categorical variables) associated with this clustering solution.
- For the **Client Value** features, the R^2 value achieved, also for 3 clusters, was of 0.31, which is considerably lower than the one obtained from the k-means clustering solution.

5. Final Cluster Solution

Following all the clustering attempts described previously, we decided that the best cluster solution for our data was to merge the clusters obtained using SOM for the Sociodemographic features and the ones derived from the k-means algorithm for the Client Value features.

The contingency table for these two cluster solutions showed that some of the 9 clusters obtained from the merge had a very small number of clients. It is quite unfeasible for this insurance company to be making specific marketing strategies for very small groups of customers; therefore, we filtered these clusters in order to only keep the ones that held 1000 clients or more. In the end, we were left with 5 clusters to explore.

We checked our final clusters for the presence of 'good' and 'bad' customers (Figure 23); while 'bad' customers are spread across all clusters, their presence is less significant (4/5%) in two of them and increased (16%) in another. Our model proved to be better at separating 'good' customers; they're almost absent from two clusters (<0,5%) and in another constitute 39% of the cluster.

5.1. Cluster Visualization (t-SNE, UMAP and PCA)

The t-distributed stochastic neighbor embedding (t-SNE) method is a manifold learning algorithm that constructs a probability distribution over the dataset, and another one in a lower dimensional dataspace, all the while trying to make them as 'close' as possible. We used this technique to represent our final cluster solution in a 2-dimensional space, as we can see in Figure 17.

UMAP, or Uniform Manifold Approximation and Projection, is a dimensionality reduction technique that, similarly to t-SNE, can be used to visualize high dimensional datasets in a 2- or 3-dimensional space. This procedure is composed of two main steps:

1. constructing a fuzzy topological representation in the original space.
2. searching - through stochastic gradient descent – for a low dimensional representation of the data that has the closest possible fuzzy topological representation, as measured by cross entropy.

This particular visualization is represented in Figure 18 (for 2 dimensions) and Figure 19 (for 3 dimensions).

PCA, or Principal Component Analysis, is a procedure that uses an orthogonal transformation to convert a dataset of possibly correlated variables into one of linearly uncorrelated features, that consist of linear combinations of the original variables, which we call principal components. Although this new space has the same number of components as the original one, most of the variance of the dataset is explained by the first few features, making it unnecessary to keep all of them. This means that we can use PCA as a dimensionality reduction technique and, if we can reduce our input space to 2 or 3 dimensions, we can also use PCA to visualize our clustering solutions. This is what is represented in Figure 16.

Using these visualization tools, we can see that we are clearly able to distinguish our clusters through a reduced dimension representation. Still, there is some overlapping, as expected.

5.2. Cluster Model Performance

In this phase, we are basically moving from unsupervised to supervised learning in order to assess our model's quality. Essentially, we are dividing our data into train and validation datasets, fitting a decision tree to our training data, and predicting the final clusters for our validation dataset. By doing this, we are able to evaluate our model's accuracy and, therefore, assess the validity of our clustering solution.

It is estimated that, in average, we can predict 87.37% of the customers' clusters correctly, making us quite confident in our clustering solution for this dataset.

5.3. Cluster Profiles and Business Strategies

Now that we have our final clustering solution for this dataset, we can start describing each cluster and discuss appropriate strategic measures the insurance company can take to improve its results based on this description. We can do this by looking at the general behavior of each feature in every cluster, that we can infer from the cluster's boxplots and bar charts, in Figure 21 and Figure 22, as well as the plots in Figure 20.

5.3.1. Fast & Furious Wanabees: too old to rock&roll, too young to die

Description: These clients are nearing retirement age and have a high income that they want to show off. After studying and investing in their career for many years, now they want to live recklessly and only care about their new expensive car, that their kids "stole" and scratched a few times.

Business Strategy Suggestions:

- Extensions to the existing auto insurance, including higher coverage and premium services, such as replacement car, towing, and so on.
- Cross-selling other insurances based on existing auto insurance.
- Create opportunities to develop a family motor insurance.
- Sell premium motor insurance through luxury car dealers
- Transmitting a sense of privilege in the insurance through marketing campaigns ("The privilege is ours").

5.3.2. Full Package, Old Money

Description: This is the oldest and better off group. On the other hand, they don't spend much on insurance. You are filthy rich, your kids are already off the nest, and you only have a few years left, so why worry about the future?

Business Strategy Suggestions:

- Create special packages for health, life and household insurance for elder people, with specific coverages and premium services such

as private hospital rooms, home medical consultations, home maintenance and repairs and access to luxury senior homes.

- Offer a one-off insurance for their children.
- Communication with the clients should create a sense of enjoyment of their golden years ("Your family tradition").

5.3.3. Geek-to-Rebel Mid-Life Crisis: lose the kids, get new convertible & partner

Description: These are parents that invested a lot in their education and now realize that it didn't *literally* pay off. They feel that, by opting to study, work hard and have kids, their youth was wasted. Now all they want is to leave their family, get a 20-something mistress, buy a convertible, and show both of them off in Vilamoura's marina. Not to be confused with the same guy who didn't take a master's degree →

Business Strategy Suggestions:

- Cross-selling of health and life insurance.
- Sell premium motor insurance through luxury car dealers.
- Marketing for these customers should enhance their sense of grandiosity and ego ("Make every car trip an ego trip").

5.3.4. Average Joe/Jane

Description: These are the perfectly average dependable clients. Never the highest spender, but never the lowest spender either. They earn a low to average yearly salary, are relatively young, have average studies and are most likely parents.

Business Strategy Suggestions:

- Create multiple insurance packages with special prices.
- Create a bonus for clients who stay with the company for x number of years.
- Transmit a sense of stability and promote a long-term relationship ("Life is a journey, let's take it together").

5.3.5. Conservative Young Parents

Description: These are the youngest and worst paid group of clients; still, they stand out in terms of spending in non-mandatory insurance. These are young new parents who fear the future.

Business Strategy Suggestions:

- Create multiple insurance packages, with increased benefits and coverages, to increase the share of wallet.
- Transmit confidence and dependability to these customers ("We are here for you!").

6. Appendix

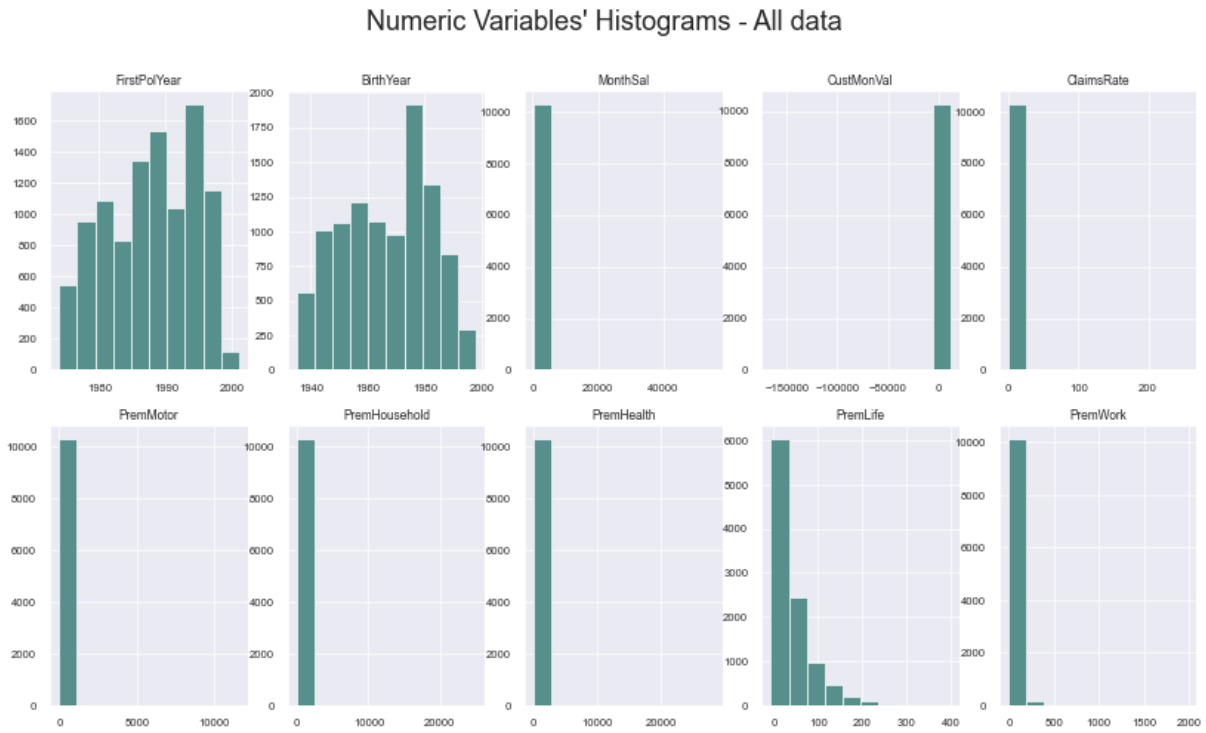

Figure 1 - Histograms for Initial Data

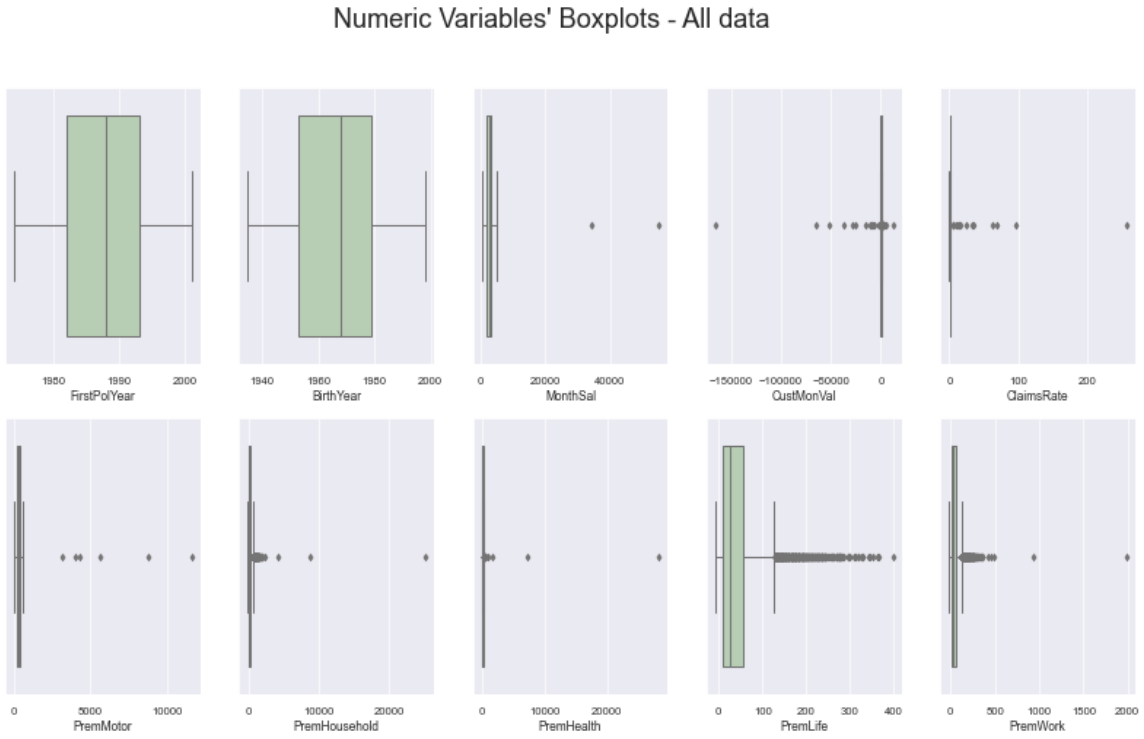

Figure 2 – Boxplots for Initial Data

Figure 3 - Pairplot for Initial Data

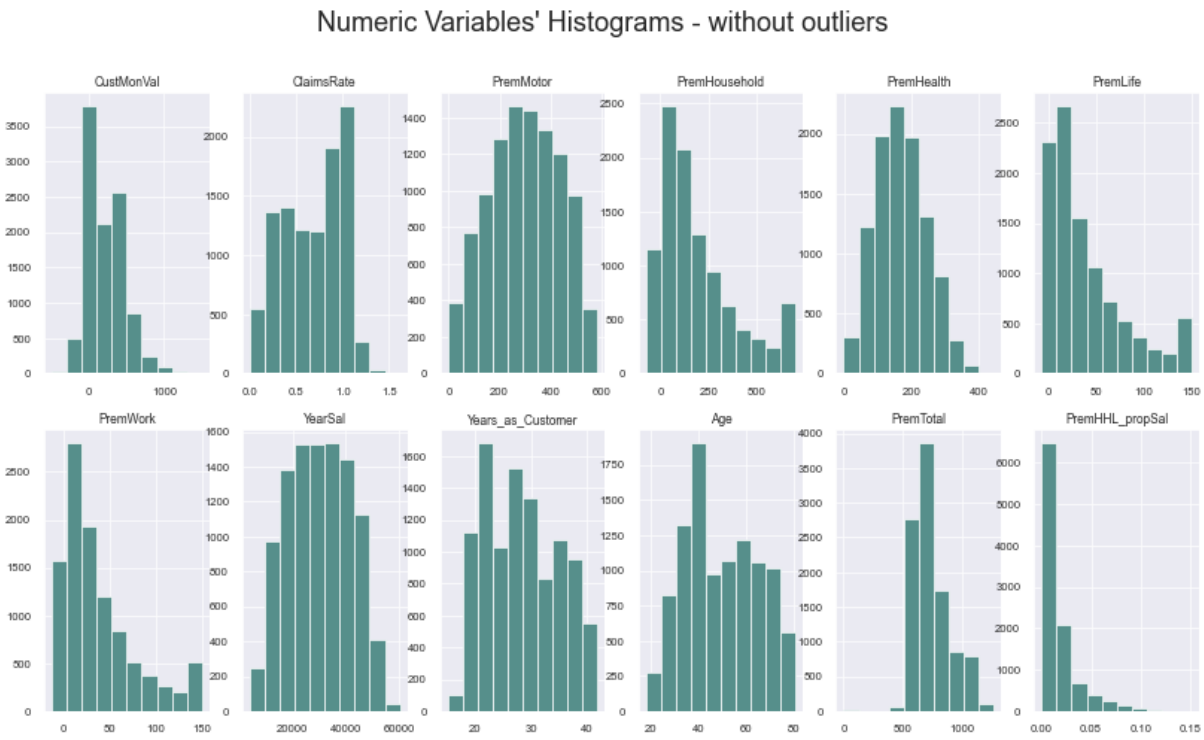

Figure 4 - Histograms After Outlier Removal

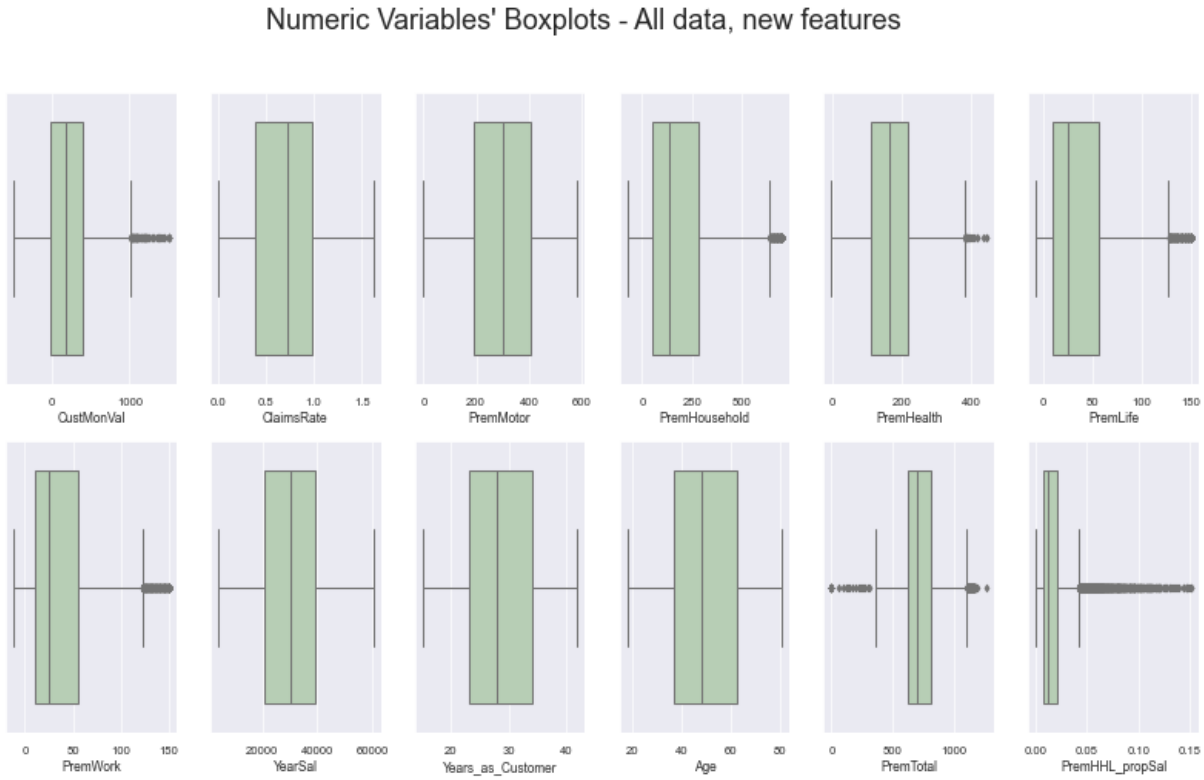

Figure 5 - Boxplots After Outlier Removal

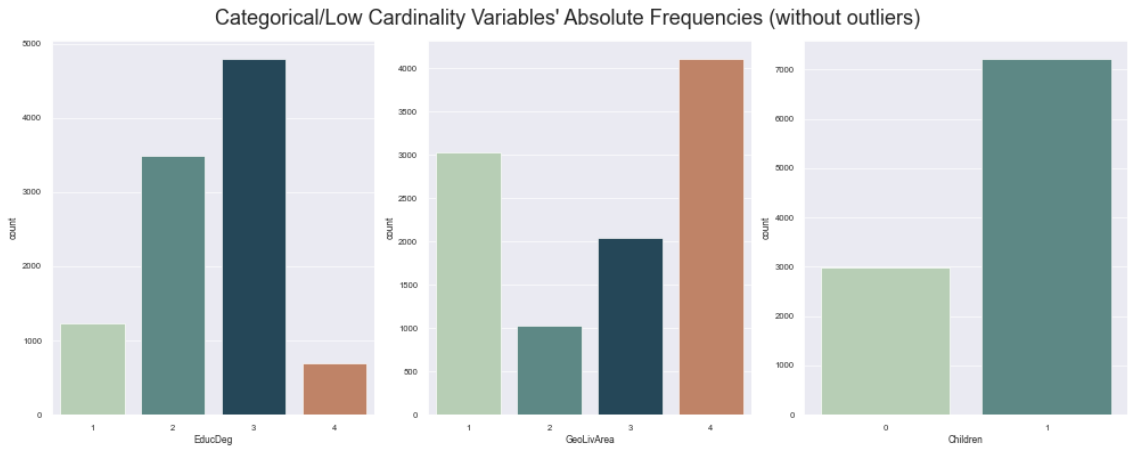

Figure 6 - Bar Charts for Categorical Features

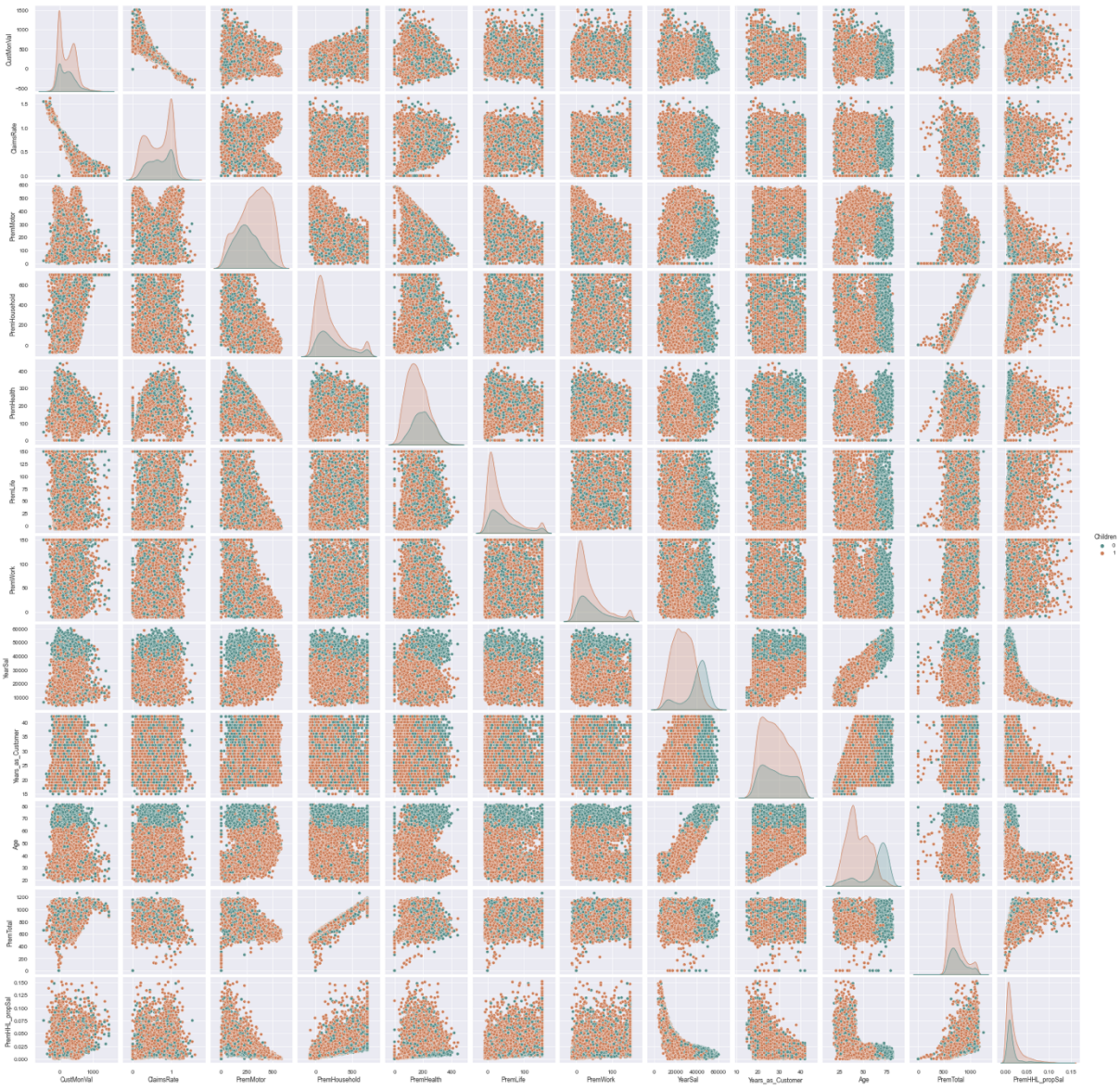

Figure 7 - Pairplot After Outlier Removal

Figure 8 - Correlation Matrix

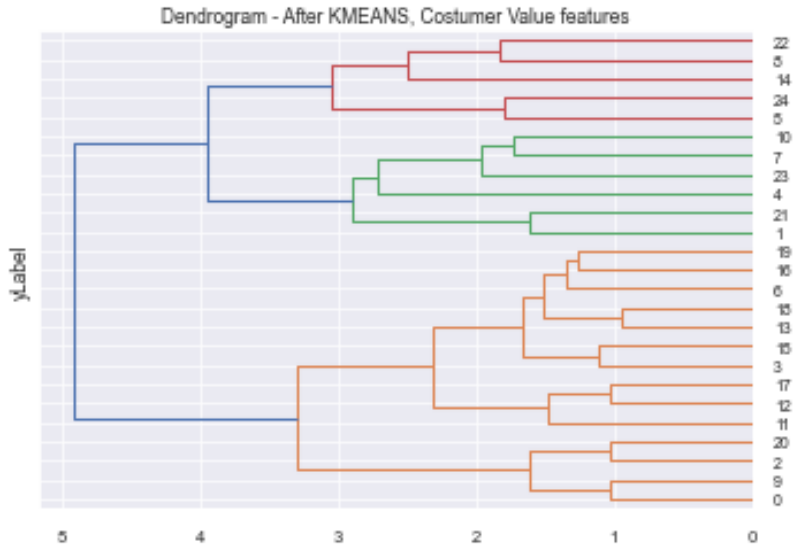

Figure 9 - Dendrogram for Client Value Features

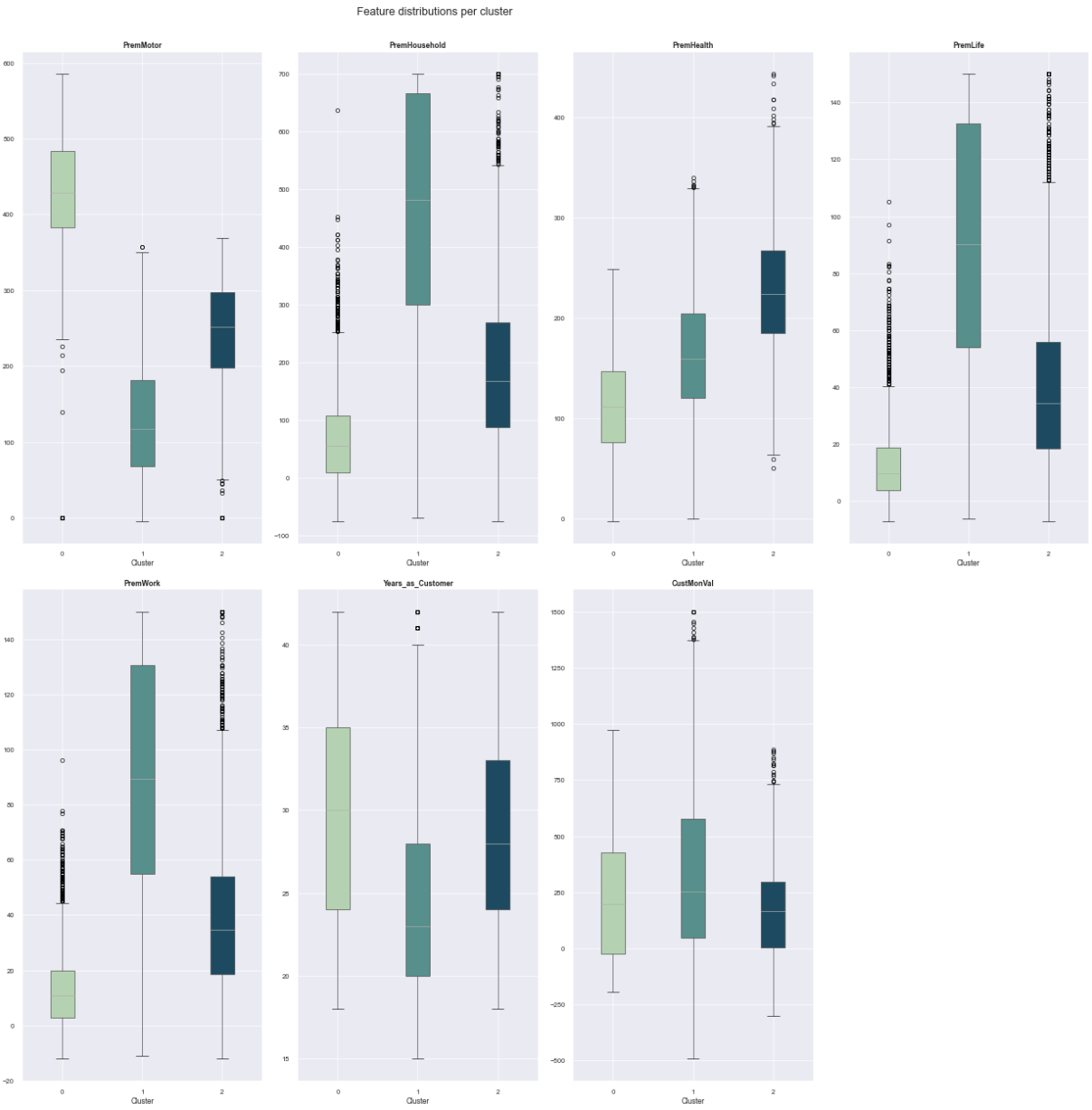

Figure 10 - Boxplots using k-means Clustering for Client Value Features

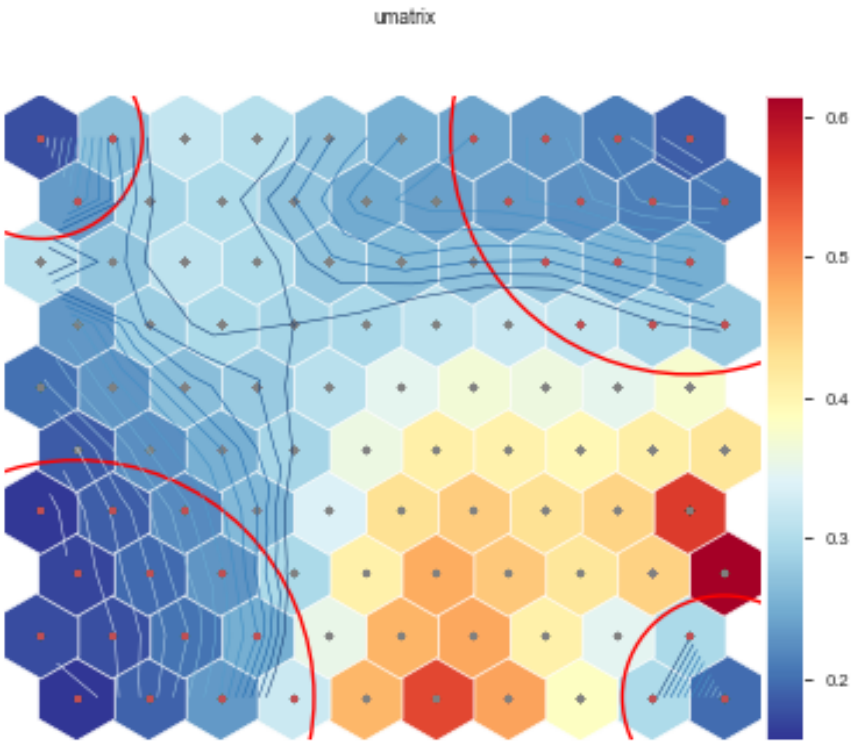

Figure 11 - UMatrix for Sociodemographic Features

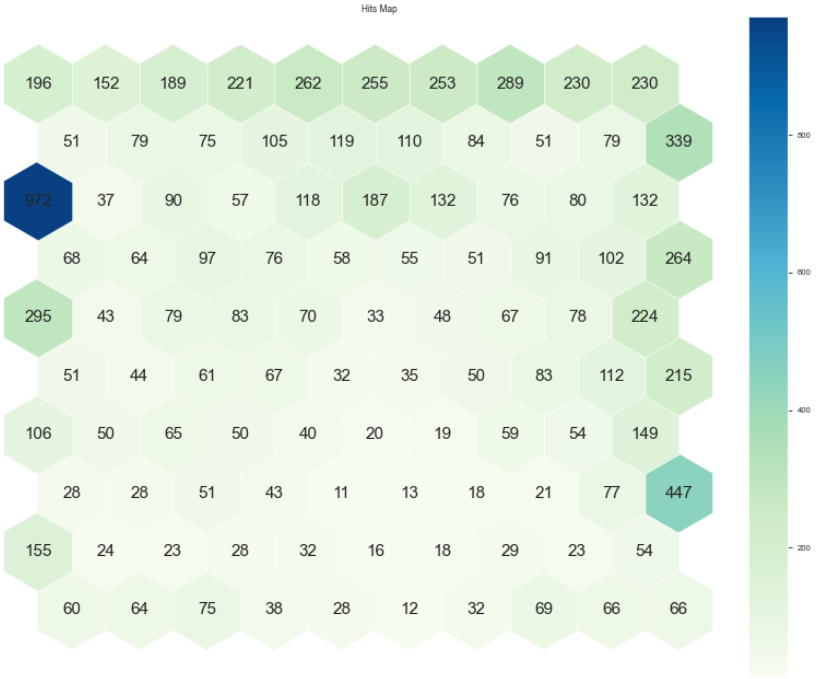

Figure 12 - Hit Map for Sociodemographic Features

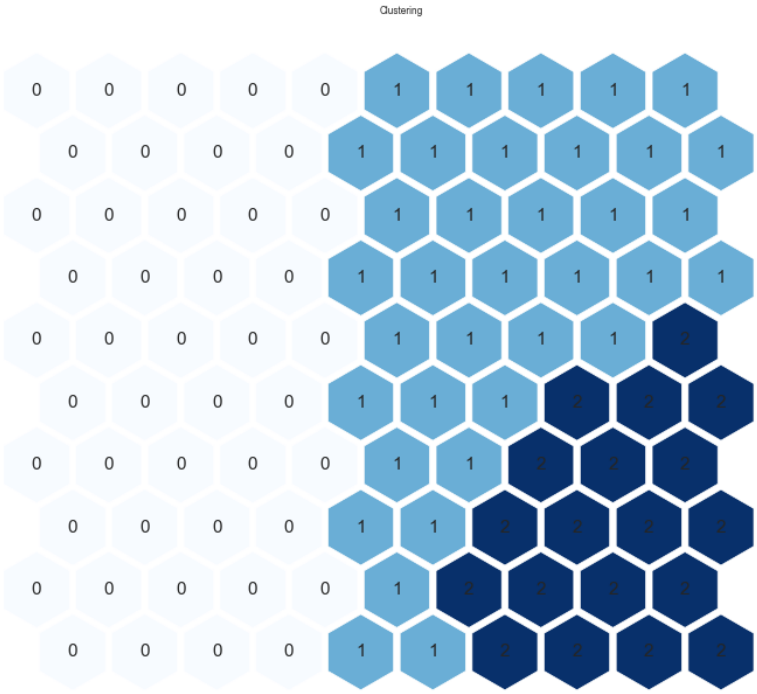

Figure 13 – Hit Map for Cluster Labels

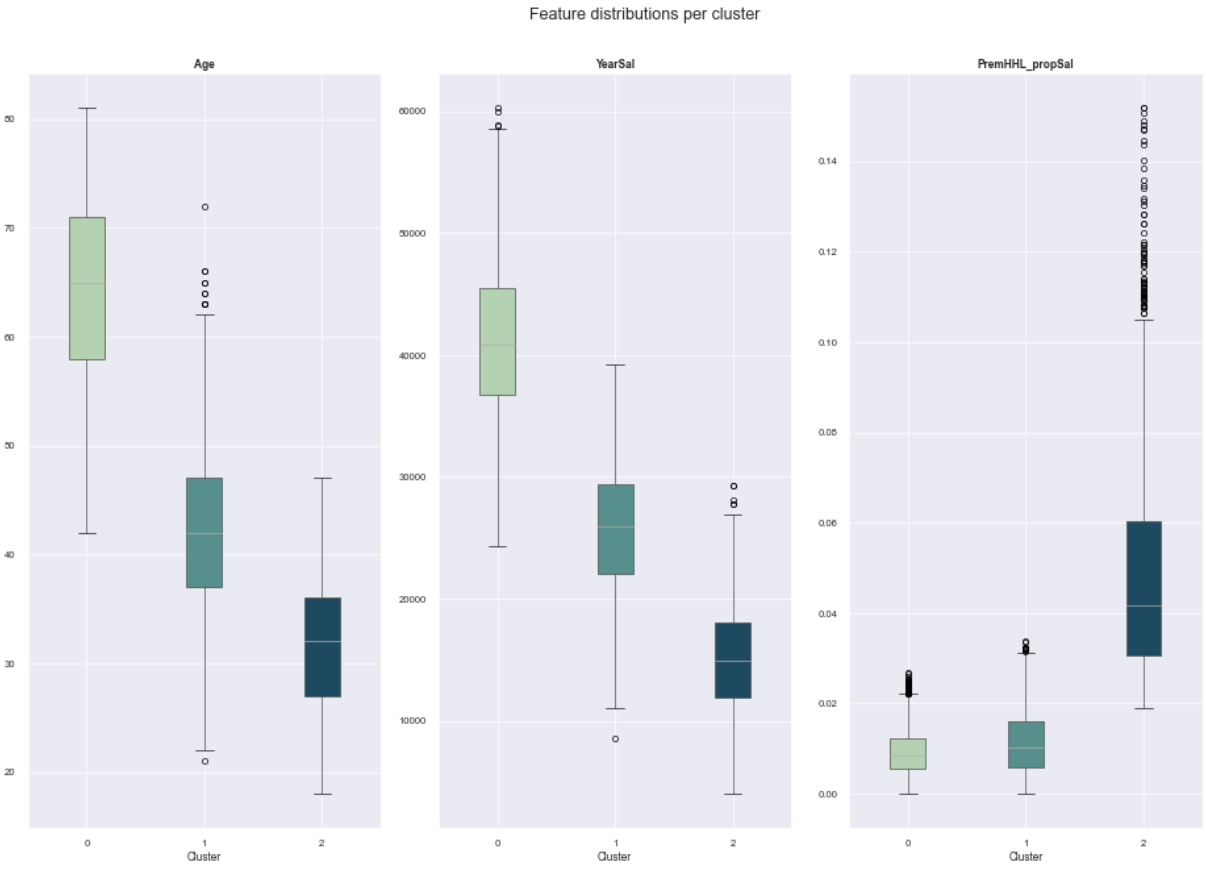

Figure 14 - Boxplots using SOM Clustering for Sociodemographic Features

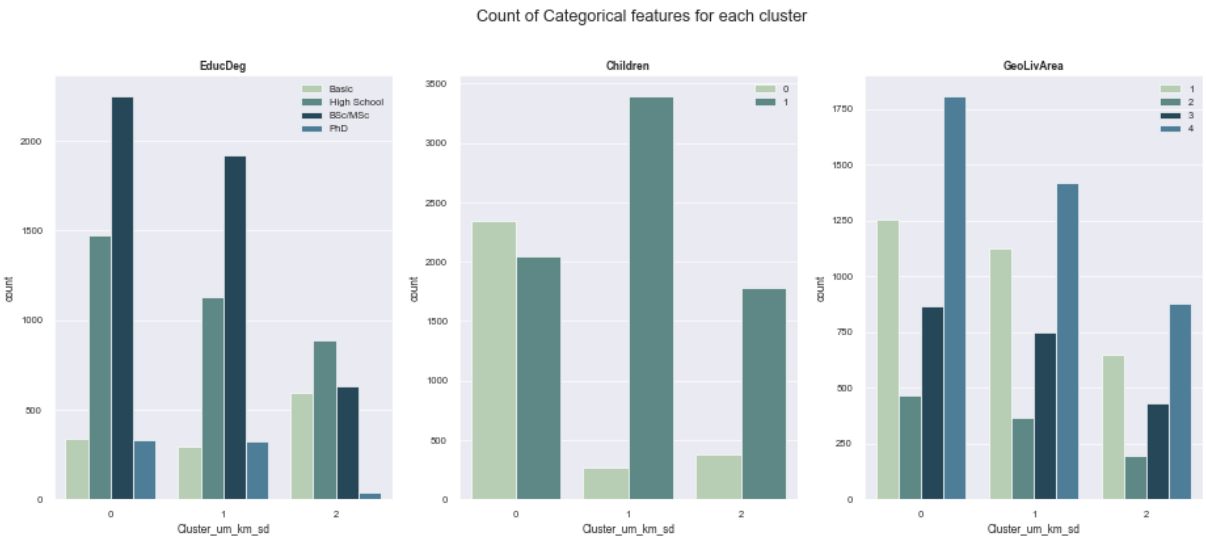

Figure 15 - Bar Charts using SOM Clustering for Sociodemographic Features

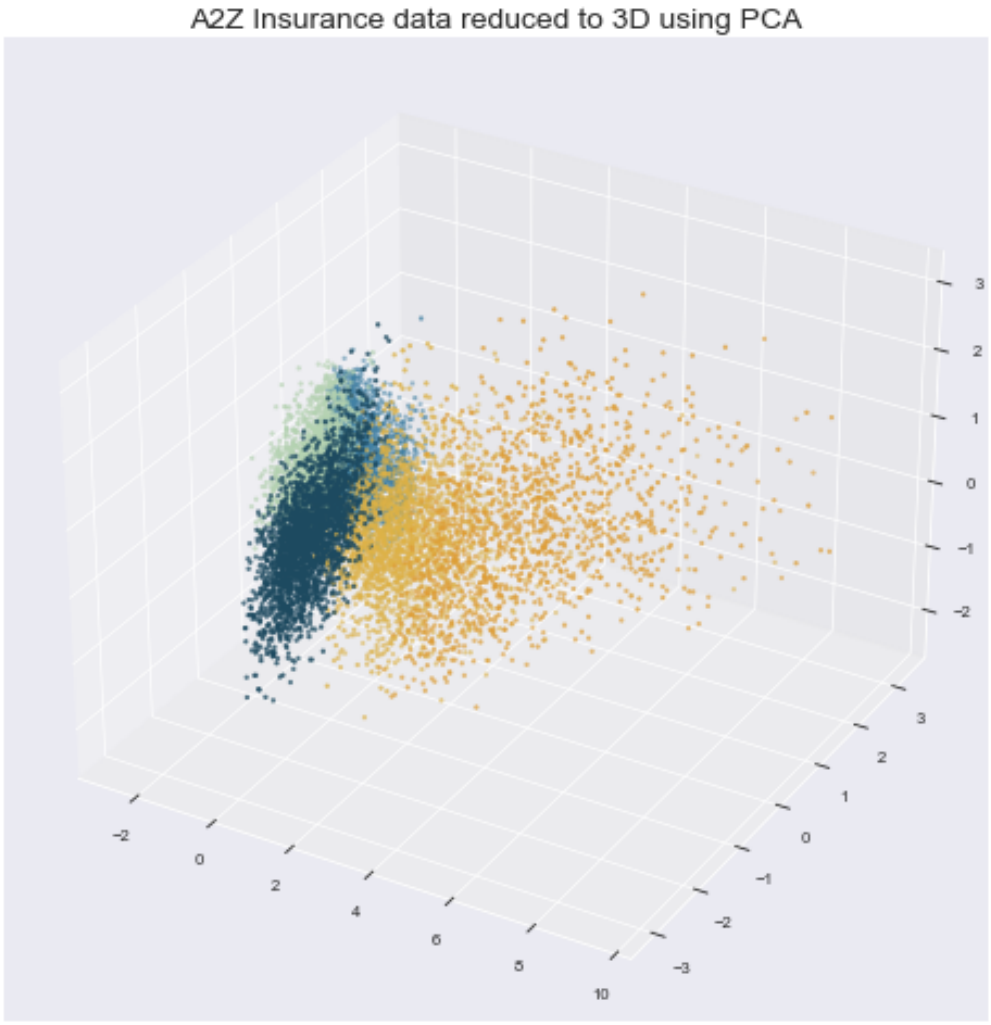

Figure 16 - Cluster Visualization using PCA

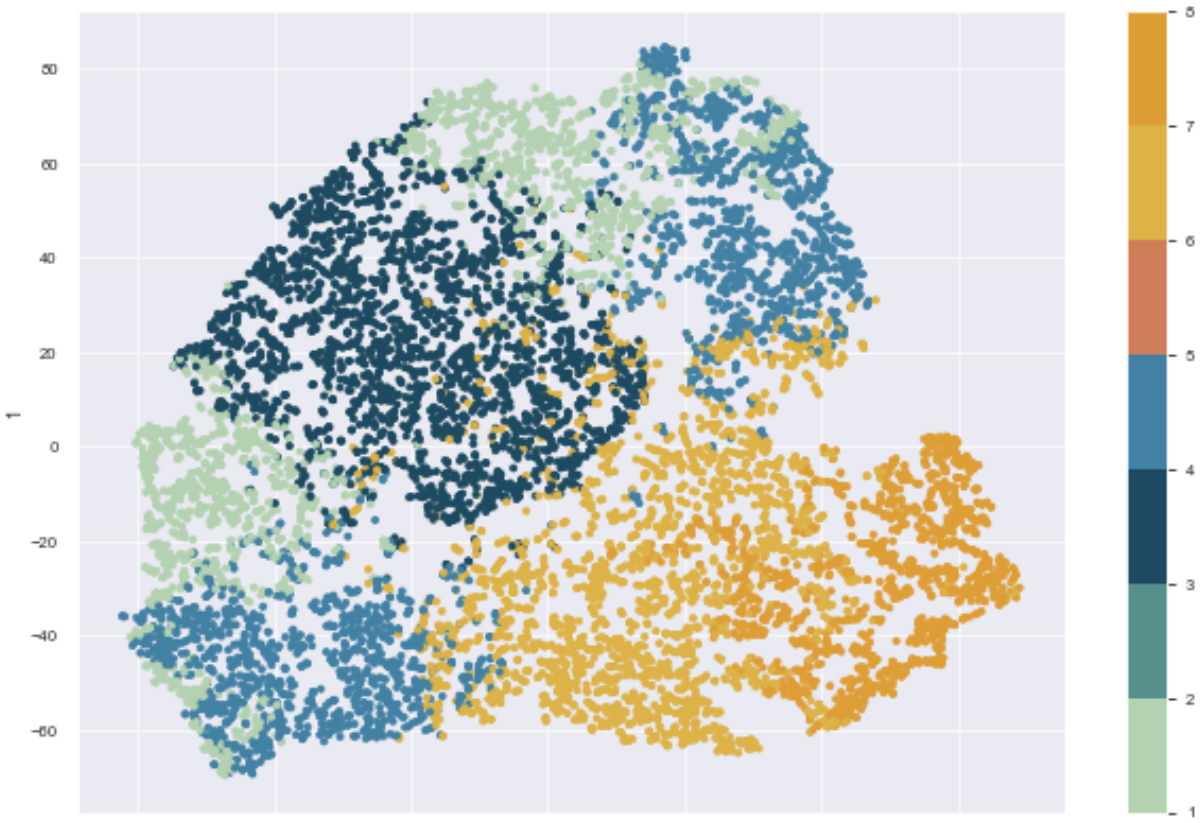

Figure 17 - Cluster Visualization using t-SNE

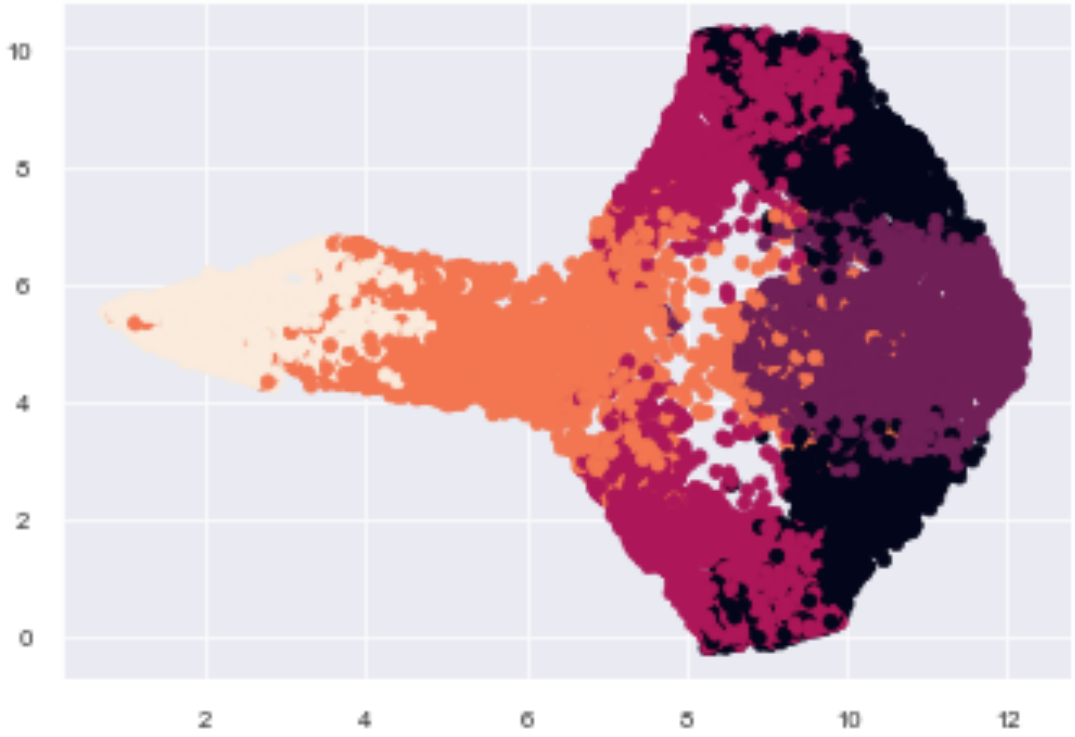

Figure 18 - Cluster Visualization in 2D using UMAP

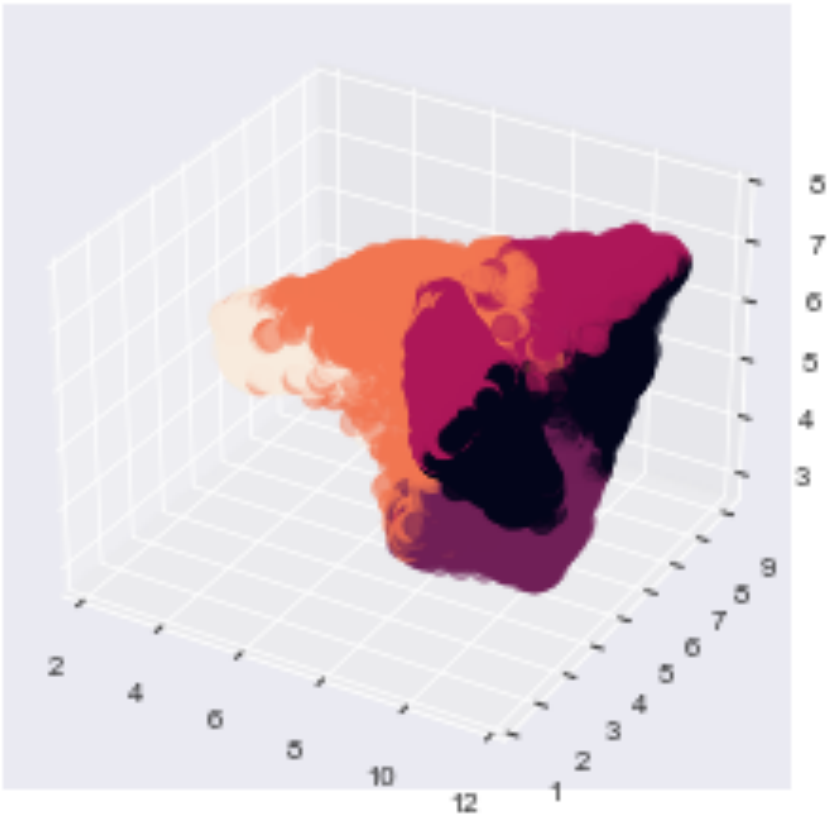

Figure 19 - Cluster Visualization in 3D using UMAP

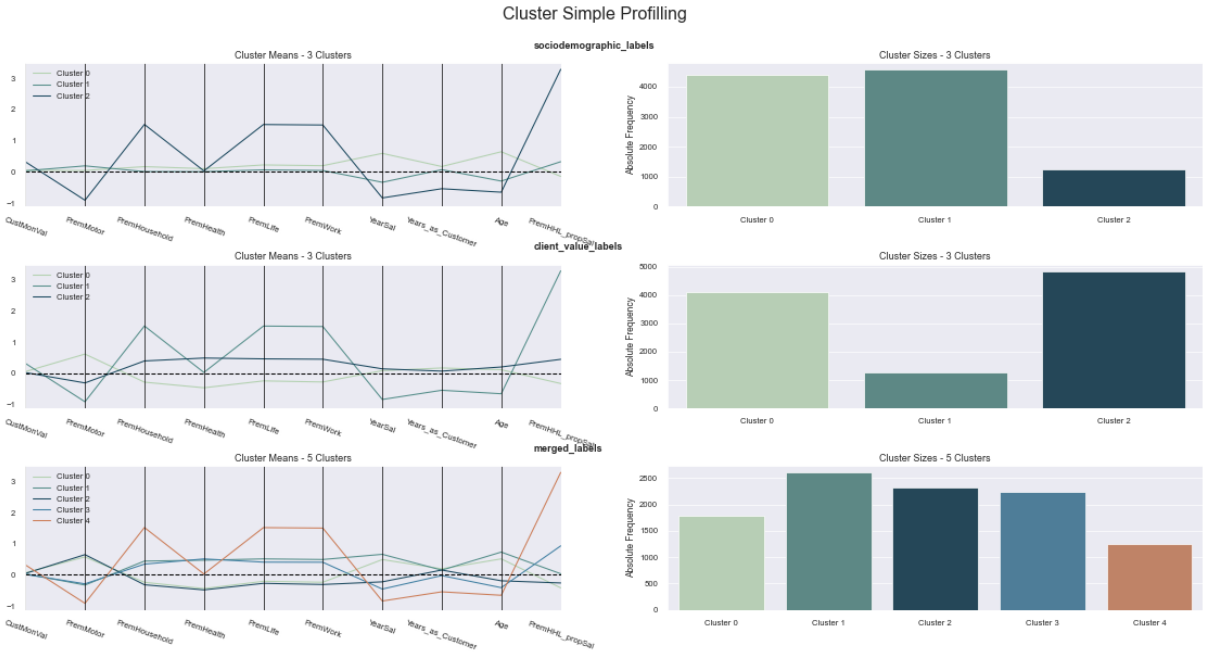

Figure 20 - Cluster Simple Profilling

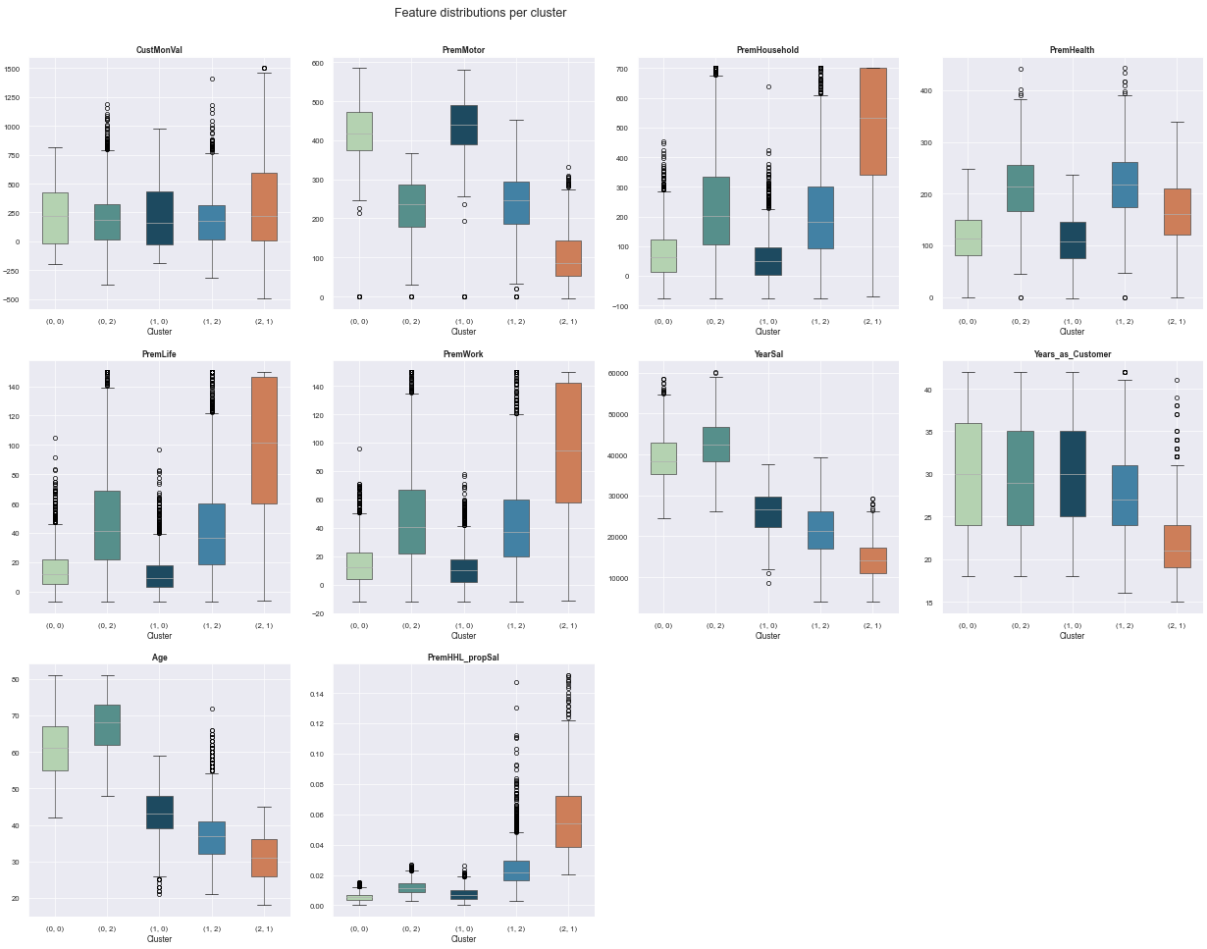

Figure 21 - Cluster Boxplots by Feature

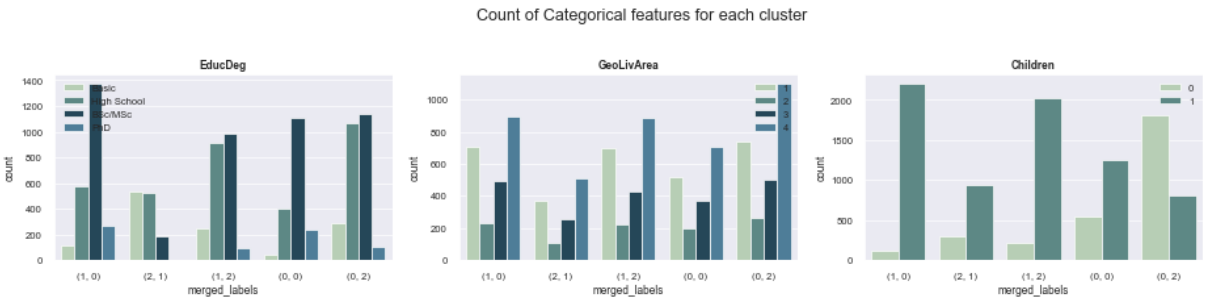

Figure 22 - Cluster Bar Charts by Feature

merged_labels	(0, 0)	(0, 2)	(1, 0)	(1, 2)	(2, 1)
Customer Score					
-1	72	125	236	230	206
0	1711	2469	1844	1715	557
1	7	8	246	294	487

merged_labels	(0, 0)	(0, 2)	(1, 0)	(1, 2)	(2, 1)
Customer Score					
-1	4%	5%	10%	10%	16%
0	96%	95%	79%	77%	45%
1	0%	0%	11%	13%	39%

Figure 23 - 'Good' (1) and 'Bad' (-1) Customer Distribution by Cluster