Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Data Mining Project

Group AE

Luís Santos, number: 20210694

Ana Luís, number: 20210671

Francisca, number:

January, 2022

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

INDEX

1. Introduction iii

1.1. Aim of this Project iii

1.2. Brief Dataset and Variables Description iii

2. Methodology iv

2.1. Imports iv

2.2. Data Preparation iv

2.2.1. Missing Values iv

2.2.2. Errors and outliers v

2.2.3. Incoherence values vi

2.2.4. Feature Engineering vi

2.2.5. Data Normalization vii

2.2.6. Encoding Categorical Variables vii

2.2.7. Principal Component Analysis vii

2.2.8. DB SCAN viii

3. Data Visualization viii

4. Cluster Analysis viii

4.1. Segmentation done with Metric Features viii

4.1.1. Hierarchical Clustering viii

4.1.2. K-Means Clustering viii

4.1.3. Profiling both Solutions ix

4.2. Different Segmentation ix

1. Introduction

1.1. Aim of this Project

Our goal is to conduct a Customer Segmentation in such a way that it will be possible for Marketing Department to better understand to better understand all the different Customers' Profiles.
Data is regarding a fictional insurance company in Portugal.

1.2. Brief Dataset and Variables Description

We were provided with an ABT (Analytic Based Table). In this table we have data regarding 10.290 customers, meaning we have 10.290 records in our dataset. Also, it stores 14 variables that describe each customer.
For each the following variables are available:

Variable	Description	Additional Information
ID	ID	
First Policy	Year of the customer's first policy	May be considered as the first year as a customer
Birthday	Customer's Birthday Year	The current year of the data-base is 2016
Education	Academic Degree	
Salary	Gross monthly salary (€)	
Area	Living area	No further information provided about the meaning of the area codes
Children	Binary variable (Y=1)	
CMV	Customer Monetary Value	Lifetime value = (annual profit from the customer) X (number of years that they are a customer) - (acquisition cost)
Claims	Claims Rate	Amount paid by the insurance company (€)/ Premiums (€) Note: in the last 2 years
Motor	Premiums (€) in LOB: Motor	Annual Premiums (2016) Negative premiums may manifest reversals occurred in the current year, paid in previous one(s)
Household	Premiums (€) in LOB: Household	
Health	Premiums (€) in LOB: Health	
Life	Premiums (€) in LOB: Life	
Work Compensation	Premiums (€) in LOB: Work Compensations	

Table 1 – Variables' Description

2. Methodology

2.1. Imports

In this project we manly used the following libraries:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Math
- Sklearn
- Scipy
- Umap

2.2. Data Preparation

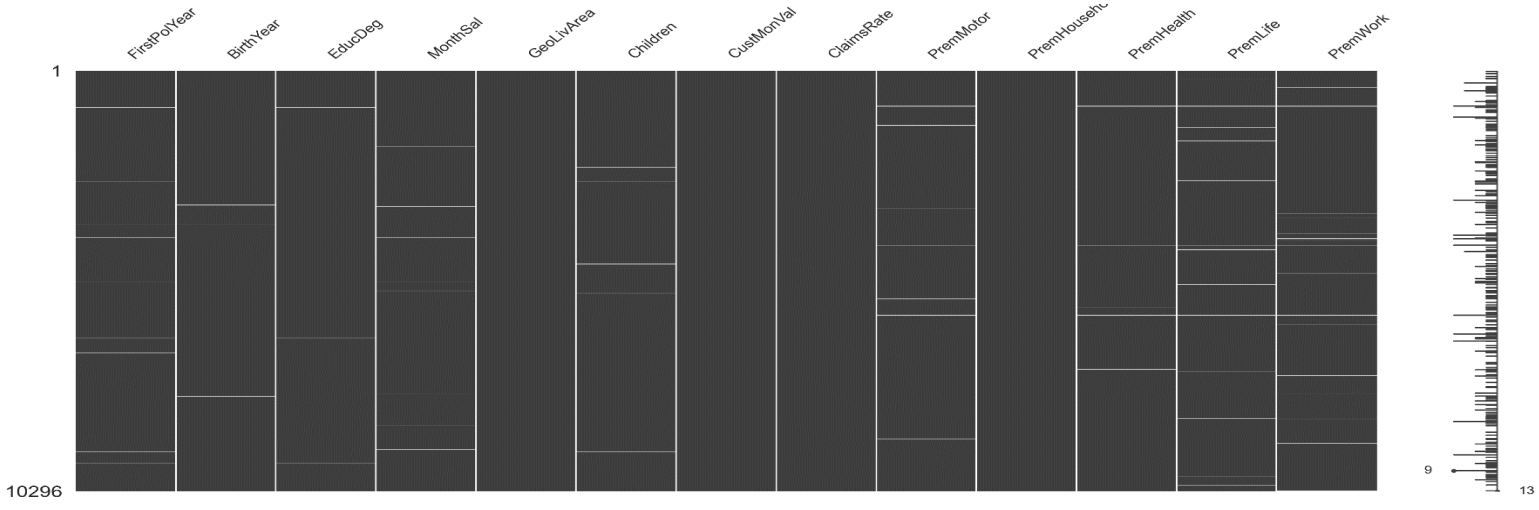
2.2.1. Missing values

Initially, we counted how many values were missing per column, finding that almost every column had missing values except for the variables CustID, CustMonVal, ClaimsRate and PremHousehold. Afterwards, we performed some visualization of the records and we verified that only 309 records i.e., 3% of our dataset, had in fact missing data.

Handling these values, we proceeded in two steps:

1. Since none of our categorical variables had significant numbers of missing values, we choose to replace those values with the most frequent class of each variable.
2. After step 1, we verified that only 2,87% of records in our data had missing values and many of them in more than one feature, so we decided to drop those records.

The following figure gives us a more tangible way of visualizing the missing values in our Data:

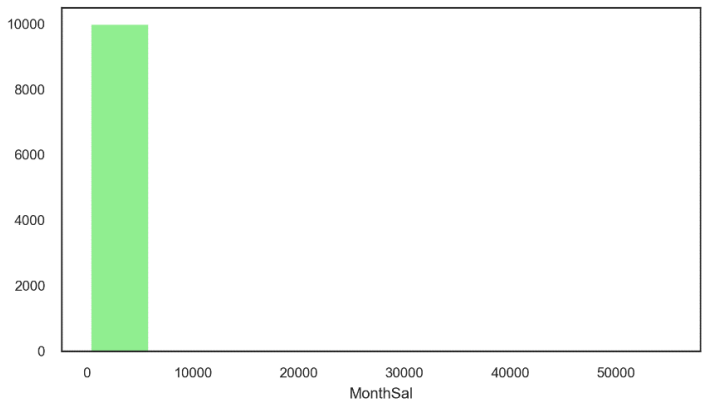


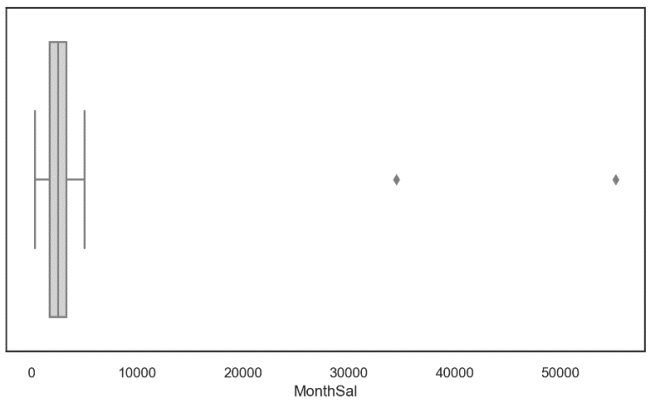
2.2.2. Errors and outliers

The importance of removing errors and outliers comes from the fact that if we perform data analysis on erroneous data, we will have erroneous conclusions. This task involves detecting data patterns that were corrupted by experimental errors, or that for some reason are not representative.

As a way of trying to assess what amount of outliers we could be dealing with in our Data, we proceeded to plot boxplots and histograms for all the metric features as well as frequency tables for all the categorical variables.

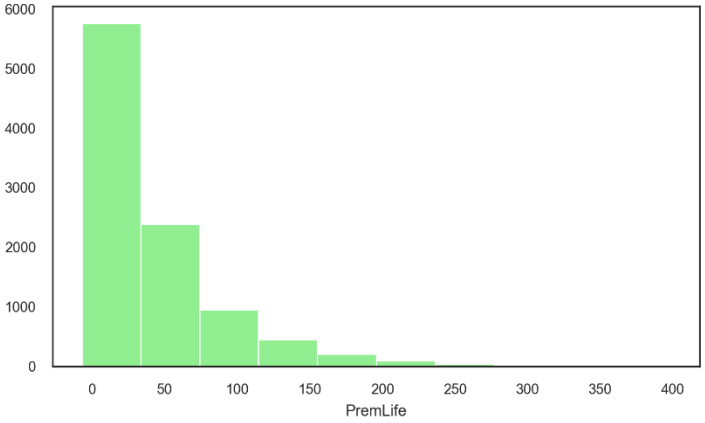
It became apparent, after examining each plot, that for most variables we had a very clear and obvious amount of points which were undoubtably anomalies, as shown in the boxplot of the variable MonthSal below:

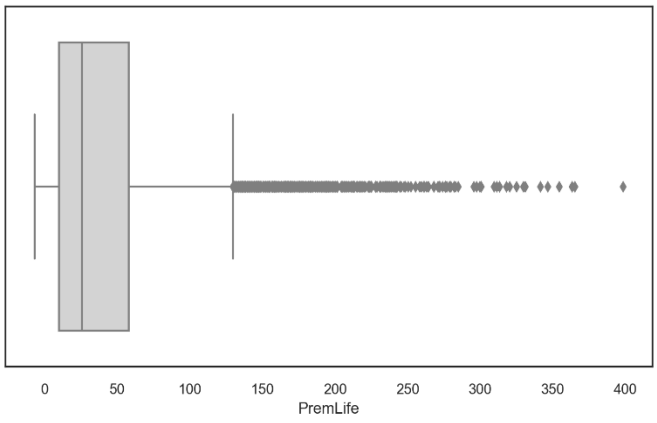




Fortunately, we easily dealt with these deviations with simple thresholds. The use of thresholds instead of resorting to more complex outlier removal methods like Inter quartile range (IQR), also allows us to better control the amount of information we lose.

1. The variable PremLife is a good example of how problematic the use of an outlier removal method like IQR could be, which in this case, would result in losing a considerable amount of information, imaginable with the boxplot bellow:





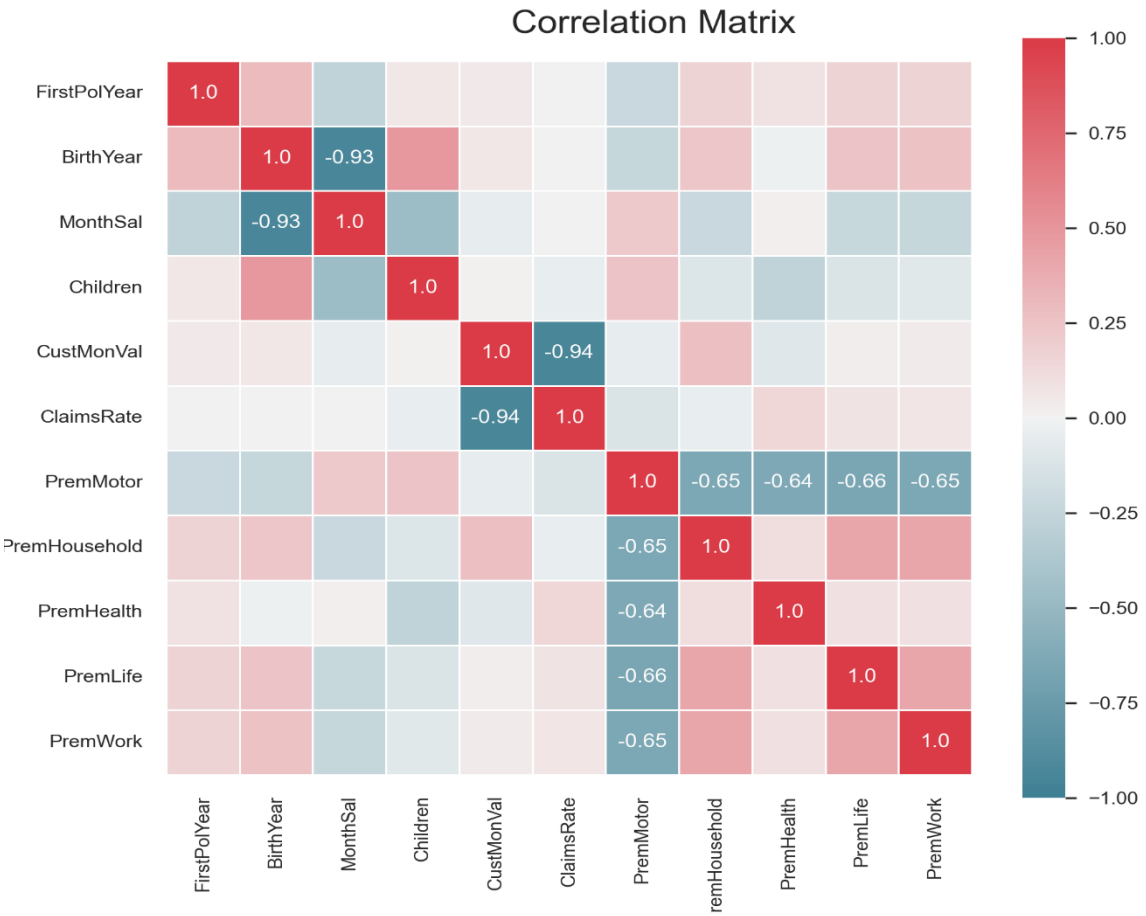
The outlier removal stage using thresholds, resulted in the removal of 1.08 % of our data.

2.2.3. Incoherence values

While looking for any type of incoherencies in our data, we spotted that our data had plenty of customers whom their birth year was most recent than their first policy year. Since there were 1881 customers in this situation, to decrease the number of affected records, we performed a quick prediction of what values FirstPolYear could take, for this we made use of a Linear Regression algorithm which allowed us to amend 906 records, leaving us with only 975 records still affected. We ended up not dealing with the left-over values right way since we would be removing the BirthYear variable all together at a later stage and relabeling the FirstPolYear variable as decades instead of year.

2.2.4. Feature Engineering

Regarding this process, we started by calculating the correlation matrix between features in order to identify any redundancy. For this purpose, we plotted an heatmap of the correlation matrix, displaying only the absolute values equal or bigger than 0,5.



After the plot analysis, we verified that two pairs of features had a high correlation between them: {BirthYear, MonthSal} and {CustMonVal, ClaimsRate}. For each pair we dropped one feature, BirthYear and CustMonVal, respectively. We chose to remove BirthYear instead of MonthSal due it's probably unreliable amount of values as seen in the coherence check phase. The removal of CustMonVal was simply because, besides the correlation with ClaimsRate, it had a higher correlation value with other variables than ClaimsRate had with other variables.

2.2.5. Data Normalization

Before initializing a scaling process, it's good practice to check if our data is normally distributed, since that could influence the choice of scaling method used later on. We proceeded with performance of normality tests (Shapiro-Wilk Test) for each feature, to confirm that the distribution of our variables wasn't, in fact, normal/gaussian.

By evaluating the p-value, we concluded that none of the features were normally distributed, as we already suspected after plotting the histograms and boxplots.

Since our data is not normally distributed, we used a Normalization scaling method like MinMaxScaler instead of a Standardization one, like StandardScaler.

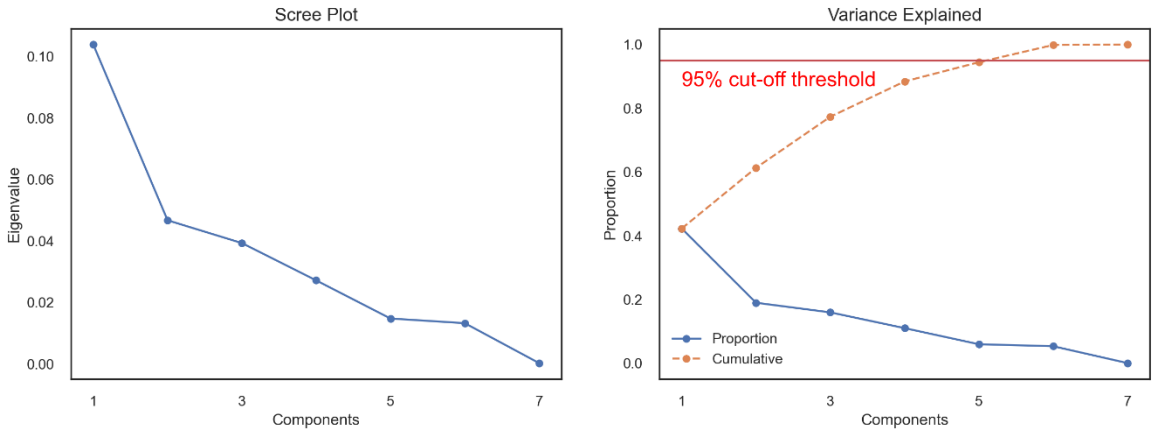
2.2.6. Enconding Categorical Variables

Before starting this process, we relabeled the variable FirstPolYear so we can make it nominal with decades instead of years, so we changed the type of this variable to *category*. Also, we removed the variable Children from this procedure since its already a Boolean variable.

We used OneHotEnconder method which converts each different label, for each categorical variable, into a binary vector.

2.2.7. Principal Component Analysis

Principal component analysis (PCA) is a technique for reducing the dimensionality of datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance.



Typically, we want the explained variance to be between 95 and 99%. To understand how many principal components we need, to reach at least 95% of variance explained, we plotted the chart above, and we concluded that around 5 components were needed. Next, we performed PCA but now with the number of principal components we wanted to retain.

We proceeded to generate a report using pandas profiling to do some extra data exploration after PCA. By analyzing this report, we could verify that a newly created PCA variable, PC0, was highly correlated with PremMotor, so we removed that variable.

2.2.8. DB SCAN

Before we actually moved on to clustering analysis we preformed some Density-Based Spatial Clustering, because it offers us a way to spot outliers when analyzing that data as a whole, instead of looking individually for them in every variable. We ended up removing an extra 197 records from our data.

3. Data Visualization

We manly used T-SNE and UMAP methods to plot a visualization, as a whole, of our data and continued to use those same methods in the cluster analysis section. All of the outputs will be available in the appendix at the end of the report.

4. Clustering Analysis

4.1. Segmentation done with Metric Features

4.1.1. Hierarchical Clustering

We started off, computing the R2 score for each type of linkage and ended up choosing the ward method.

Next, we plotted the dendrogram to get an idea of how clusters we should use. We reached the conclusion that 6 clusters would give us the best way to cluster our data

Finally, we performed the hierarchical clustering using the AgglomerativeClustering function and ended up with a prediction score in a Decision Tree Classifier of **77%**.

4.1.2. K-Means Clustering

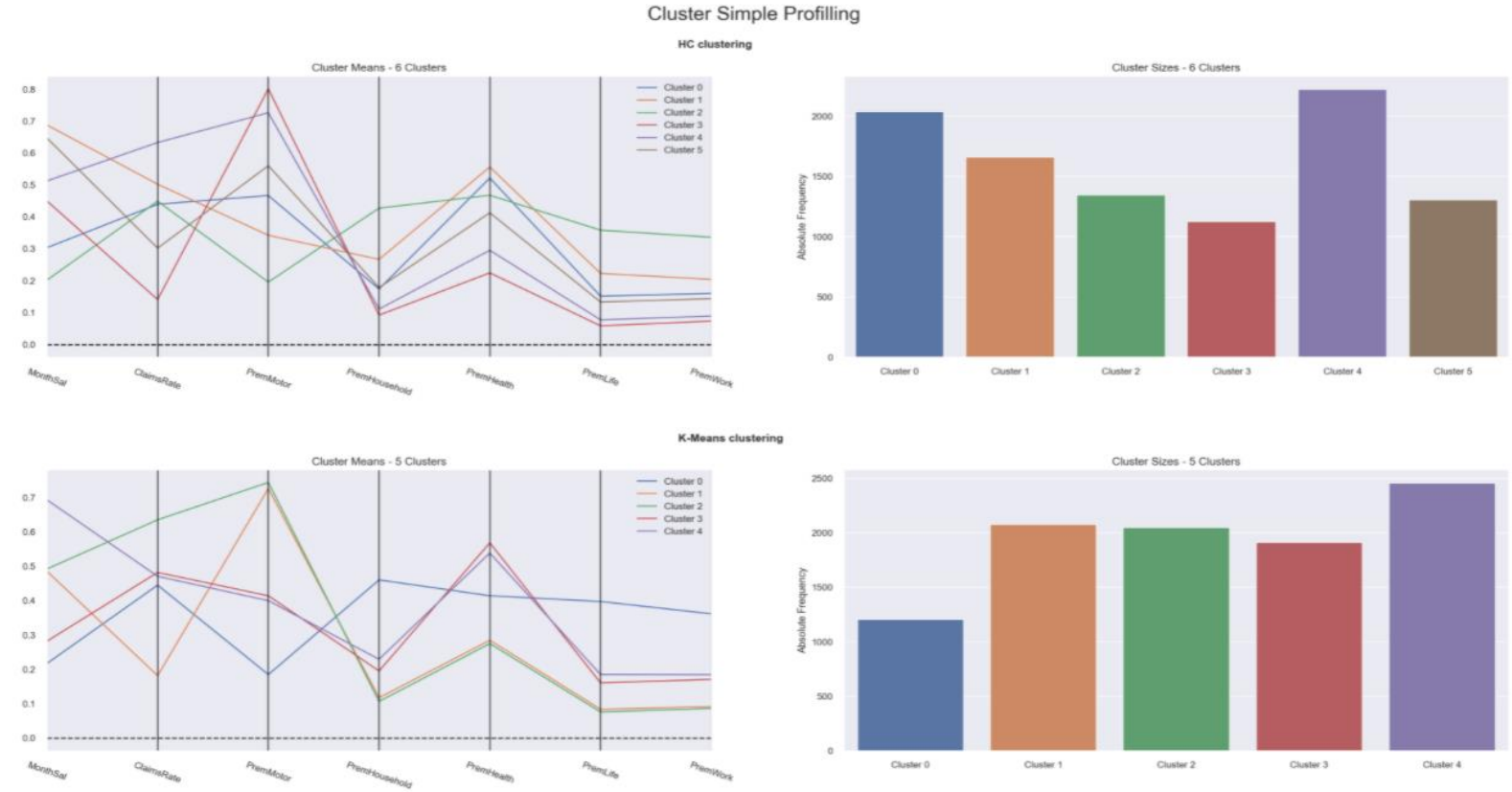
We started off, computing the inertia values and the gap statistics for a range of 20 clusters and using the elbow method reached the conclusion that 5 clusters look like the ideal number of clusters for our data using K-Means.

Finally, we performed after performing the K-Means clustering with end up with a prediction score of **88%**, once again done with a Decision Tree algorithm.

We still performed a Mean Shift clustering but the results were far from perfect so that solution wasn't be considered.

4.1.3. Profiling both Solutions

From the clustering methods used on the metric features we reached the two sets of clusters bellow, kmean's one having a better prediction score with 88% as well as a slightly more evenly distributed classes.



4.2. Different Segmentation

We also experiment this clustering methods with a different segmentation, using two sets of different features, one (premium_features) where you aggregated all the 'prem' variables plus the ClaimsRate, since it made contextual sense and the other (demographic_features) with the rest of the features which had GeoDemographics/Socio-economic characteristics.

We started off with, computing the R2 score for the best clustering method for each of one of the sets of variables. K-Means ended winning in both cases.

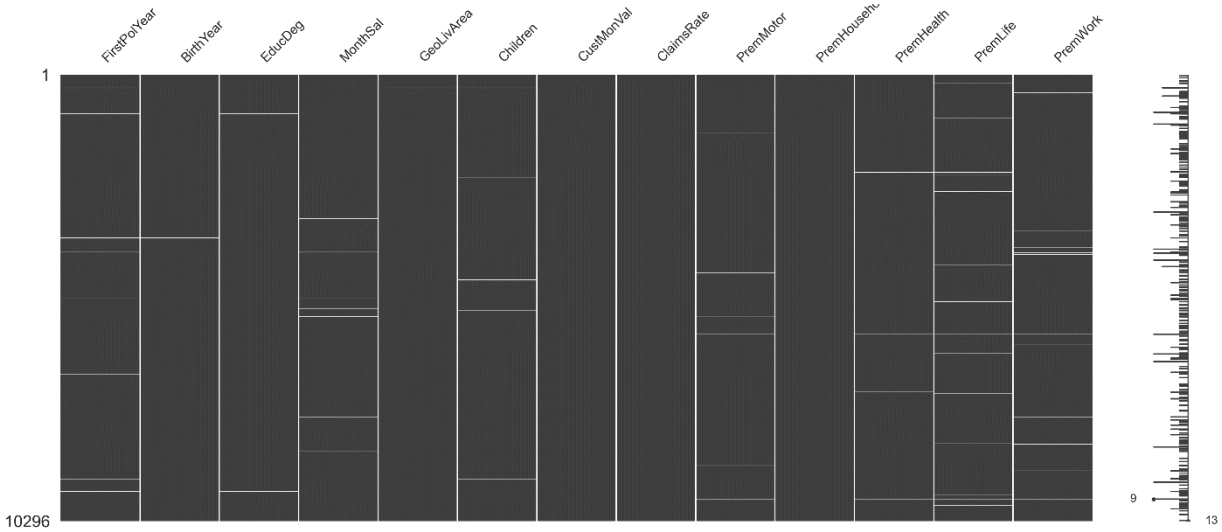
After clustering both segments we calculated how many points each label had and merged the low frequency ones with other labels.

Next, we proceeded calculate how many clusters we would need for a merged solution using hierarchical clustering and ended up choosing 4 clusters

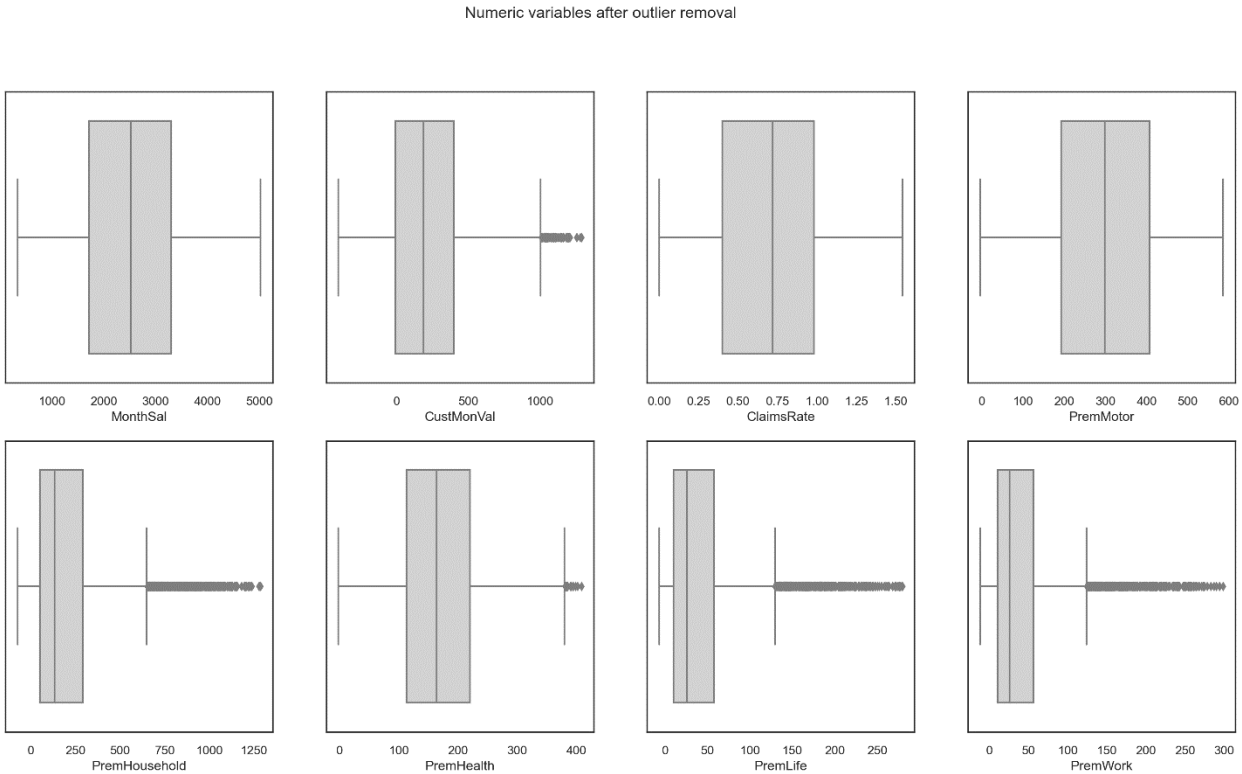
Finally the Decision Tree algorithm we got a prediction score of **89%**, just slightly better than the KMeans one, one metric features.

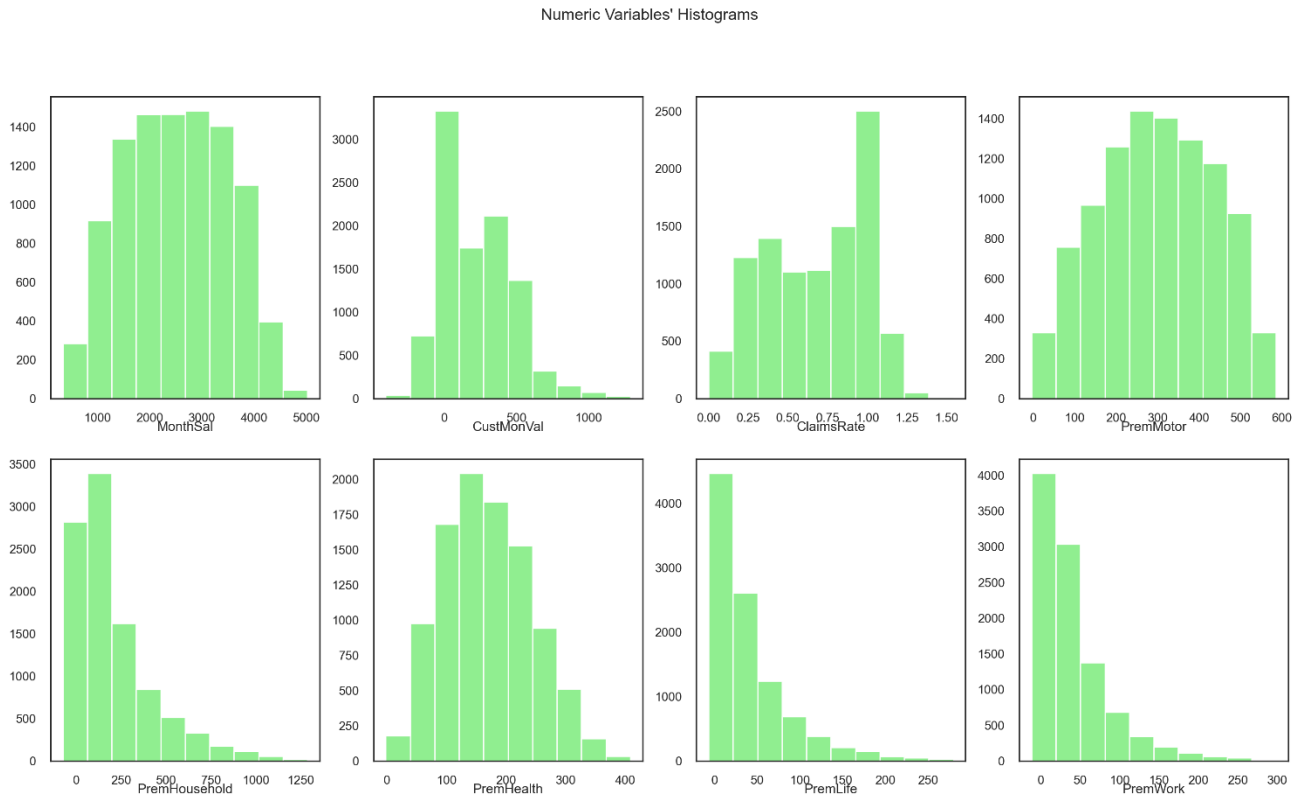
APPENDIX

Missing Values Matrix

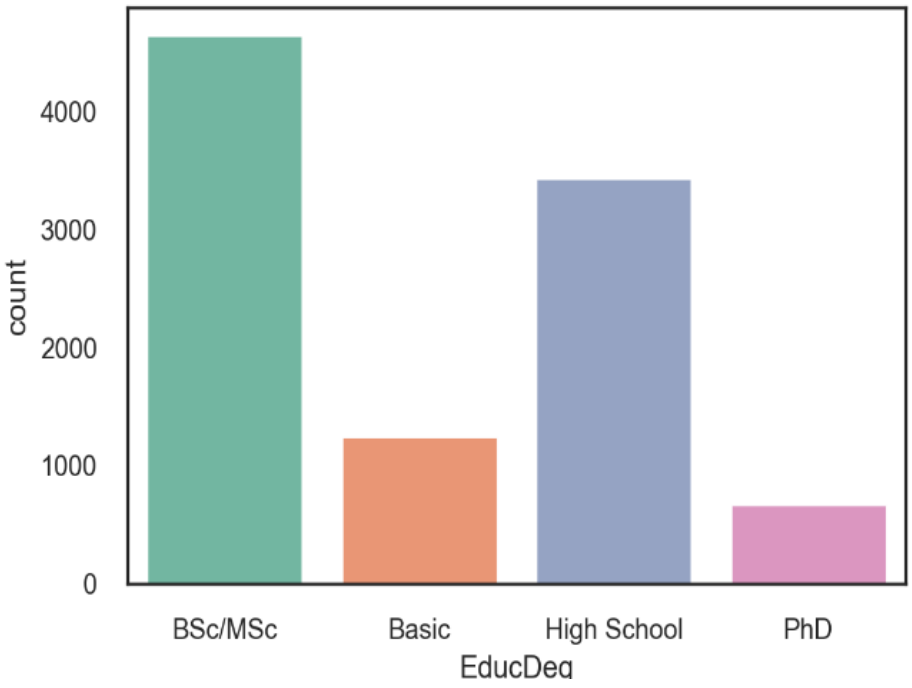


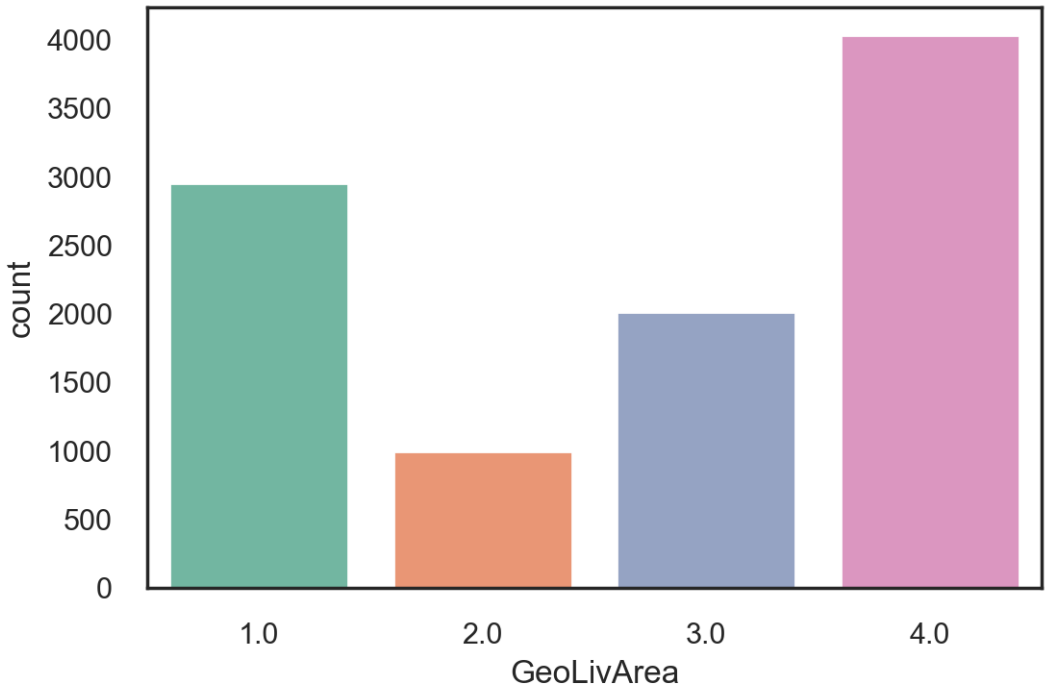
Boxplots and Histograms

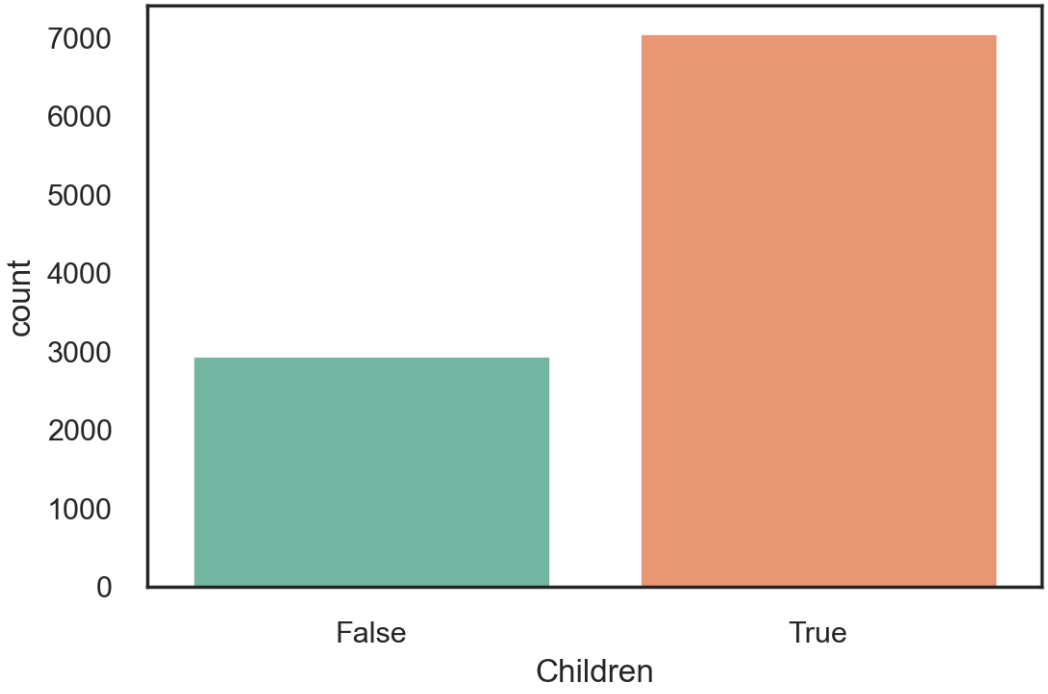




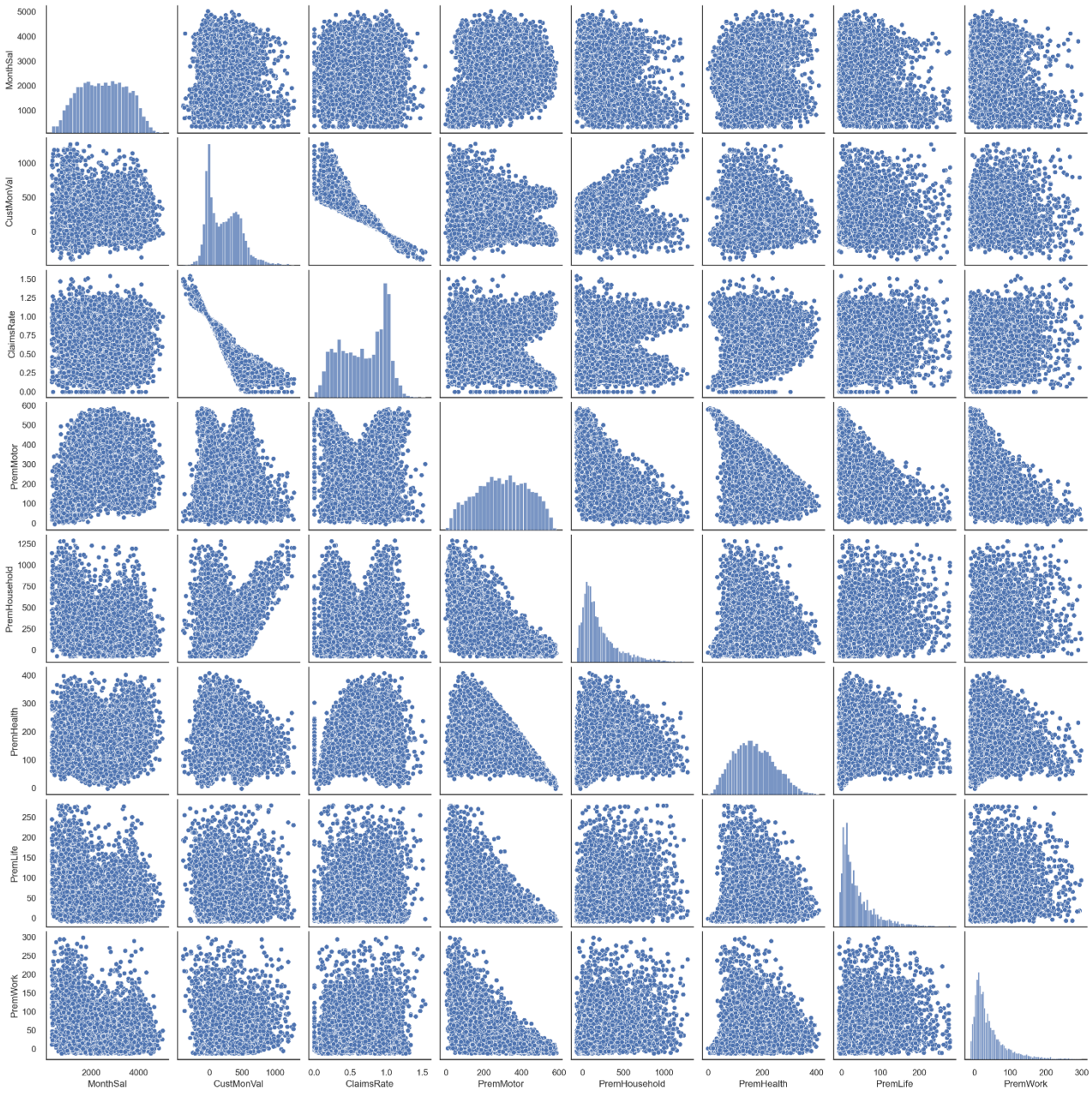
Frequency tables



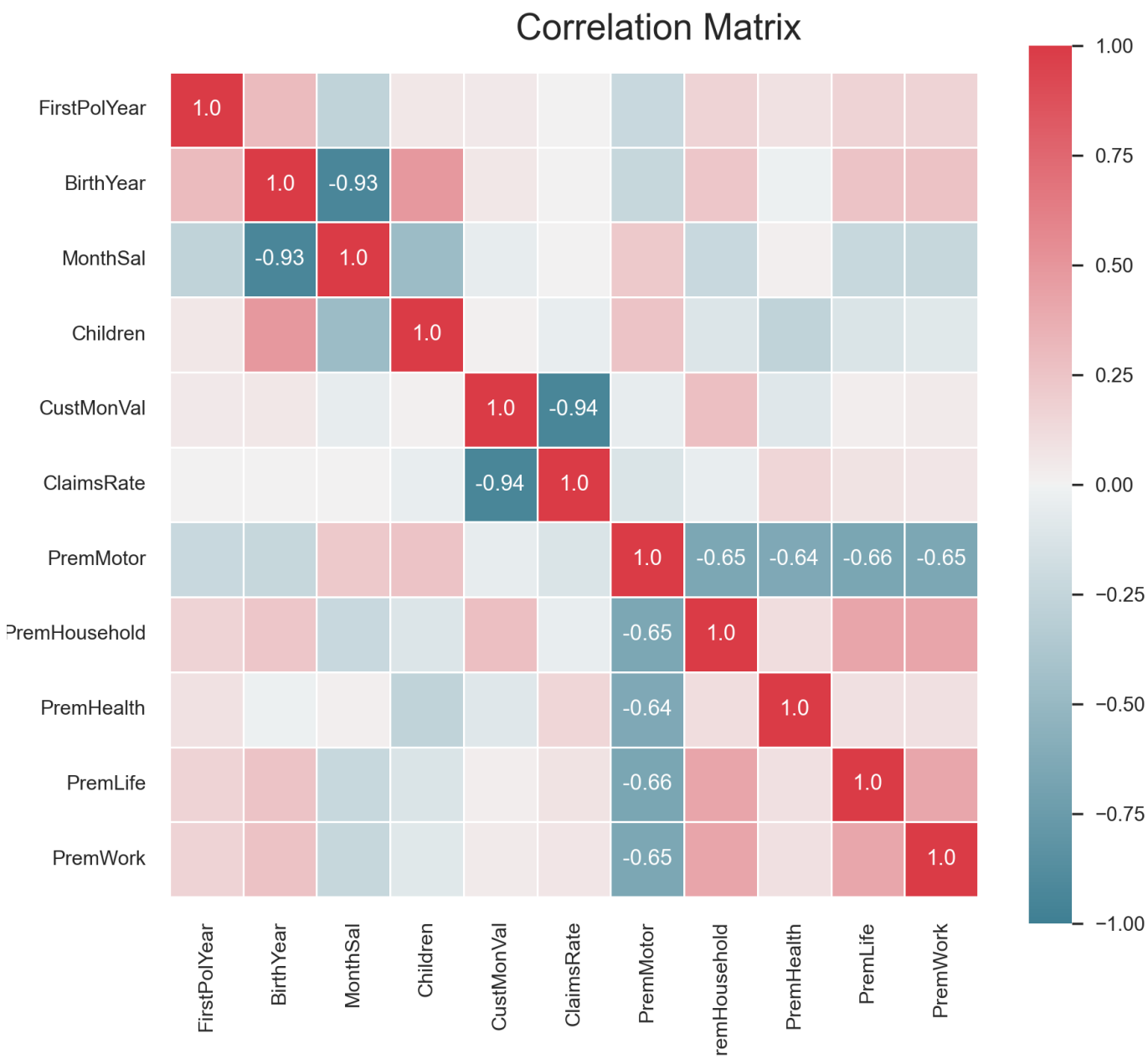

4000
3500
3000
2500
2000
1500
1000
500
0
count
1.0
2.0
3.0
4.0
GeoLivArea


7000
6000
5000
4000
3000
2000
1000
0
count
False
True
Children

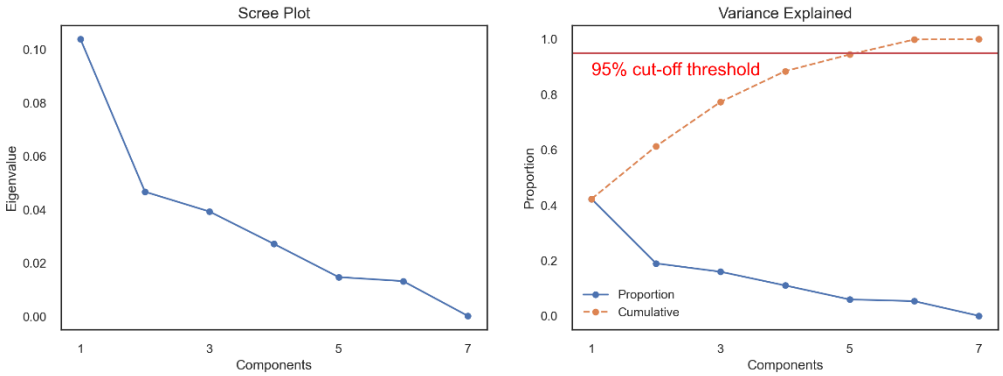
Correlogram



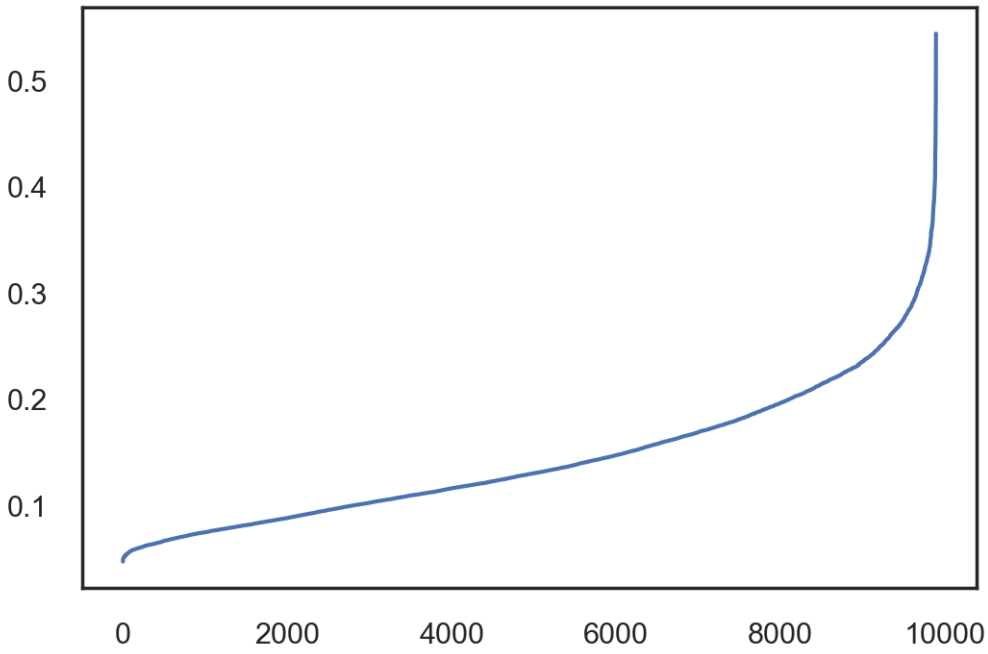
Correlation Matrix



Explain Variance

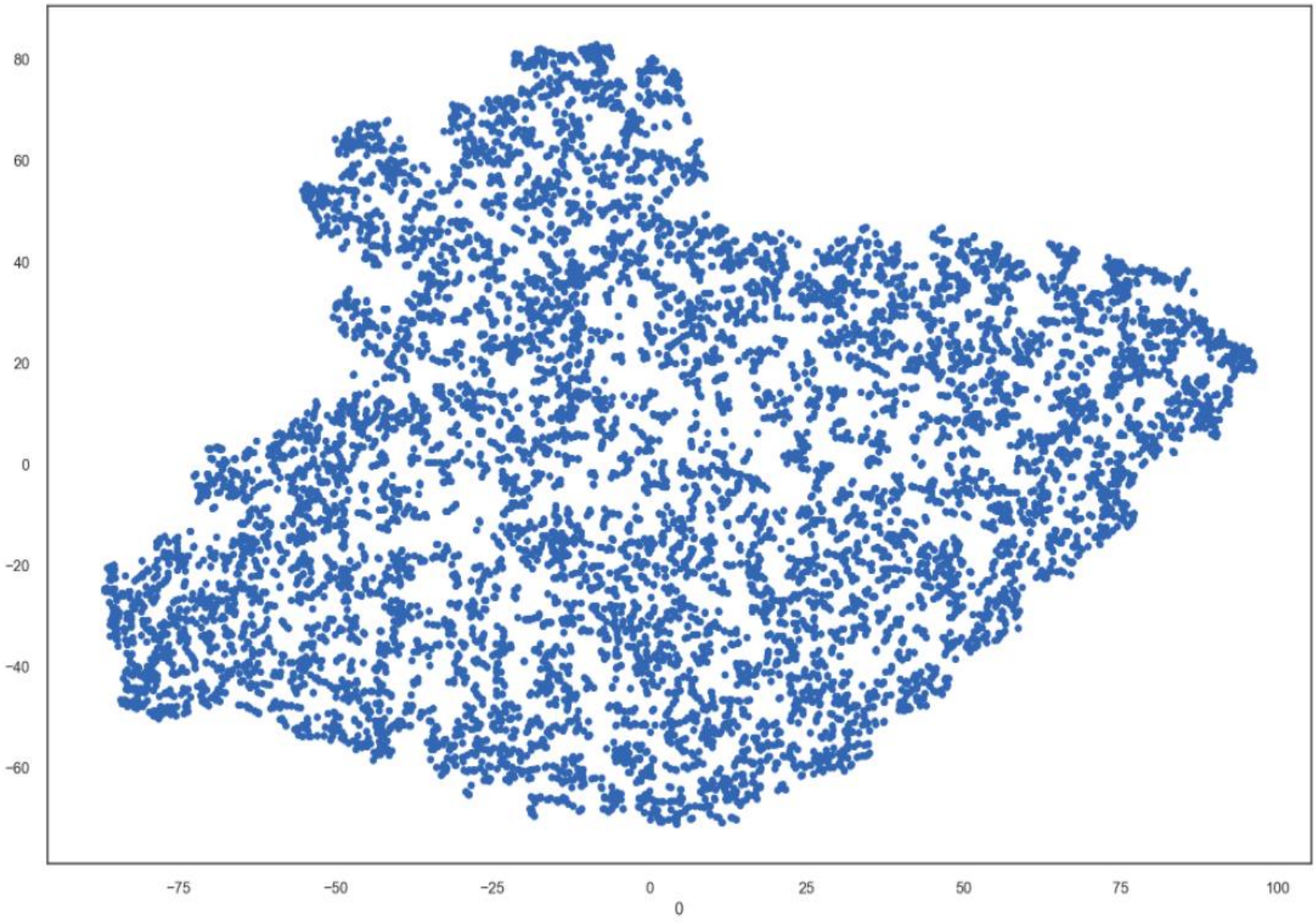


Right eps value for DBSCAN

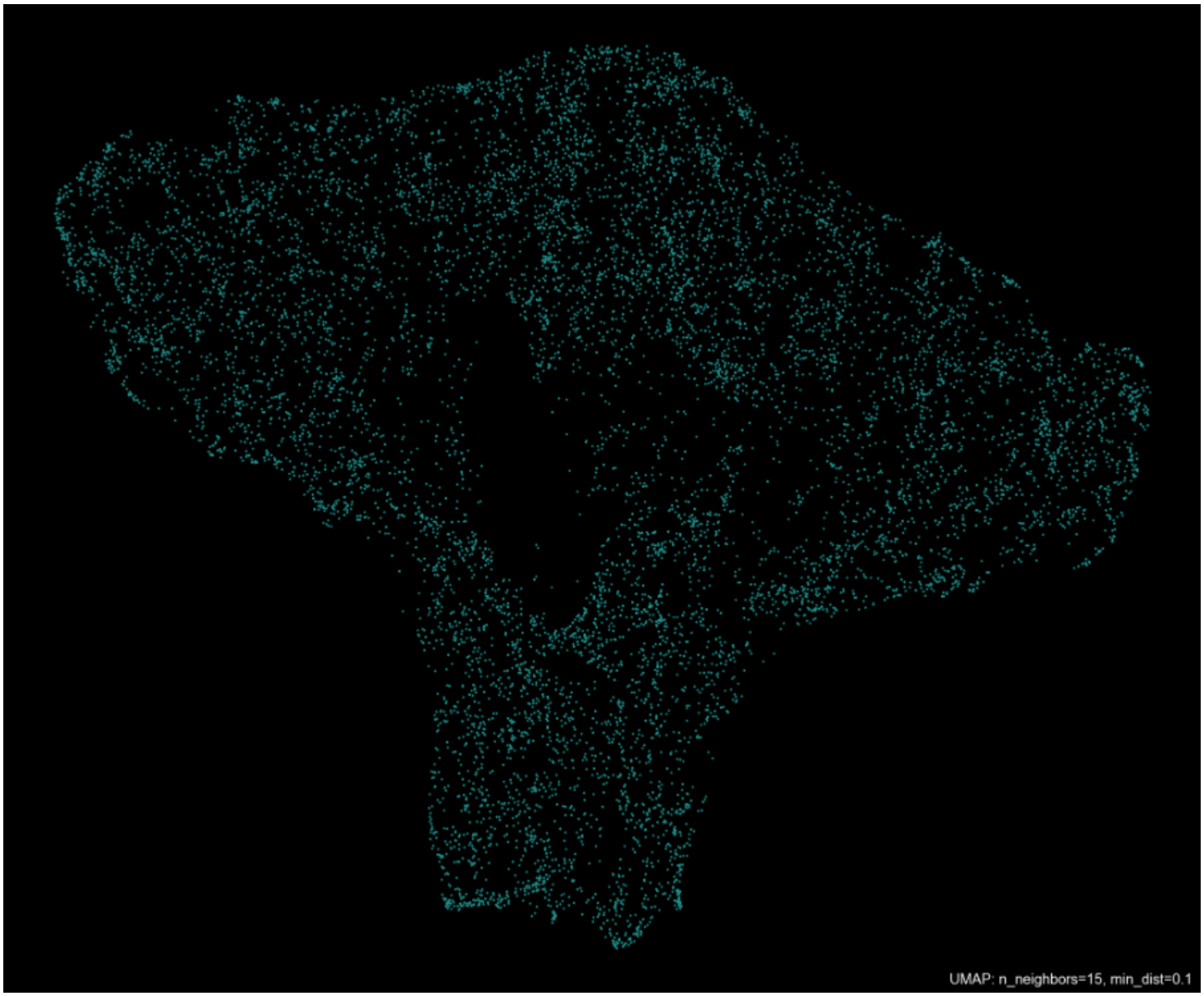


Pre-Processed Data Visualization

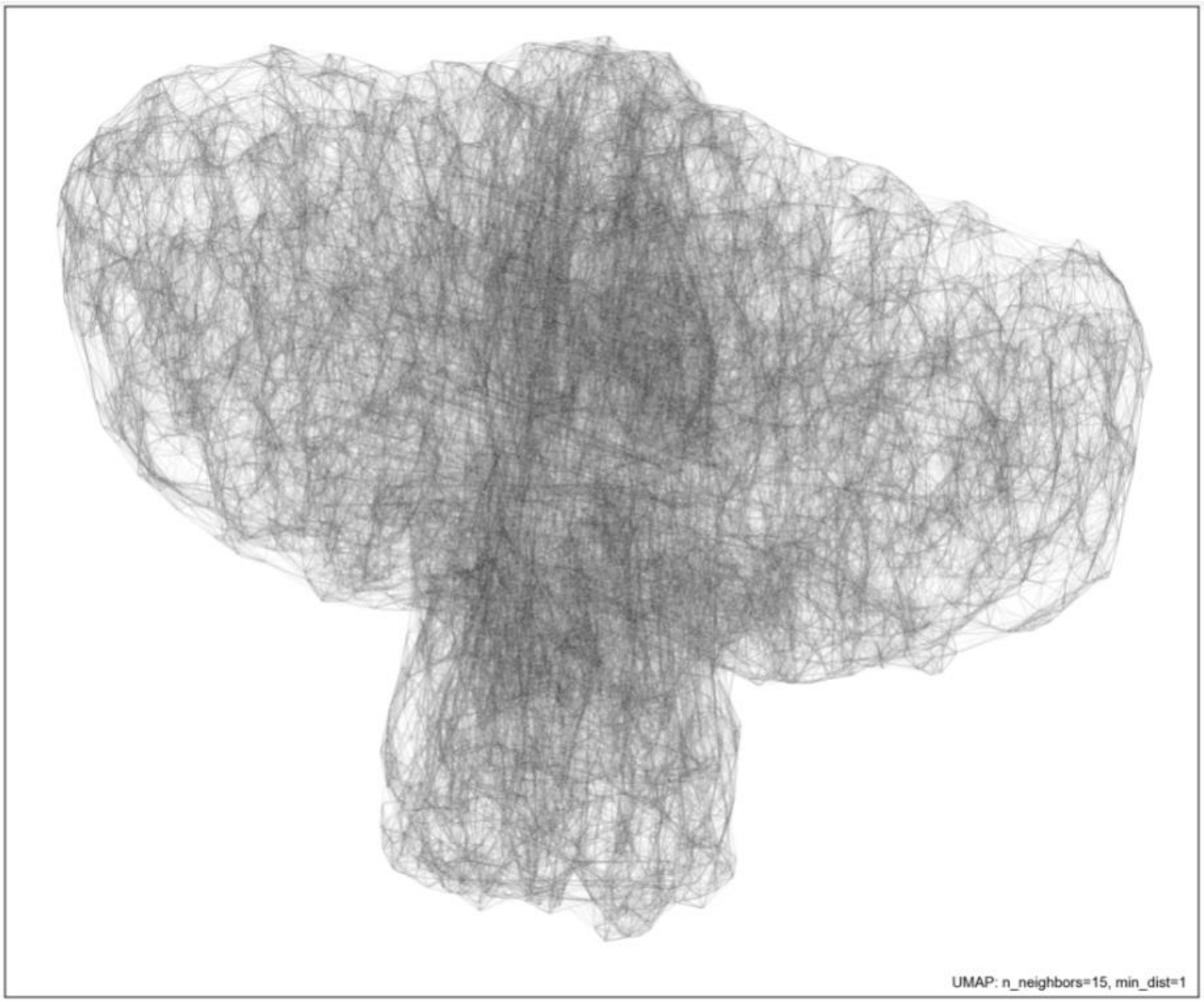
T-SNE



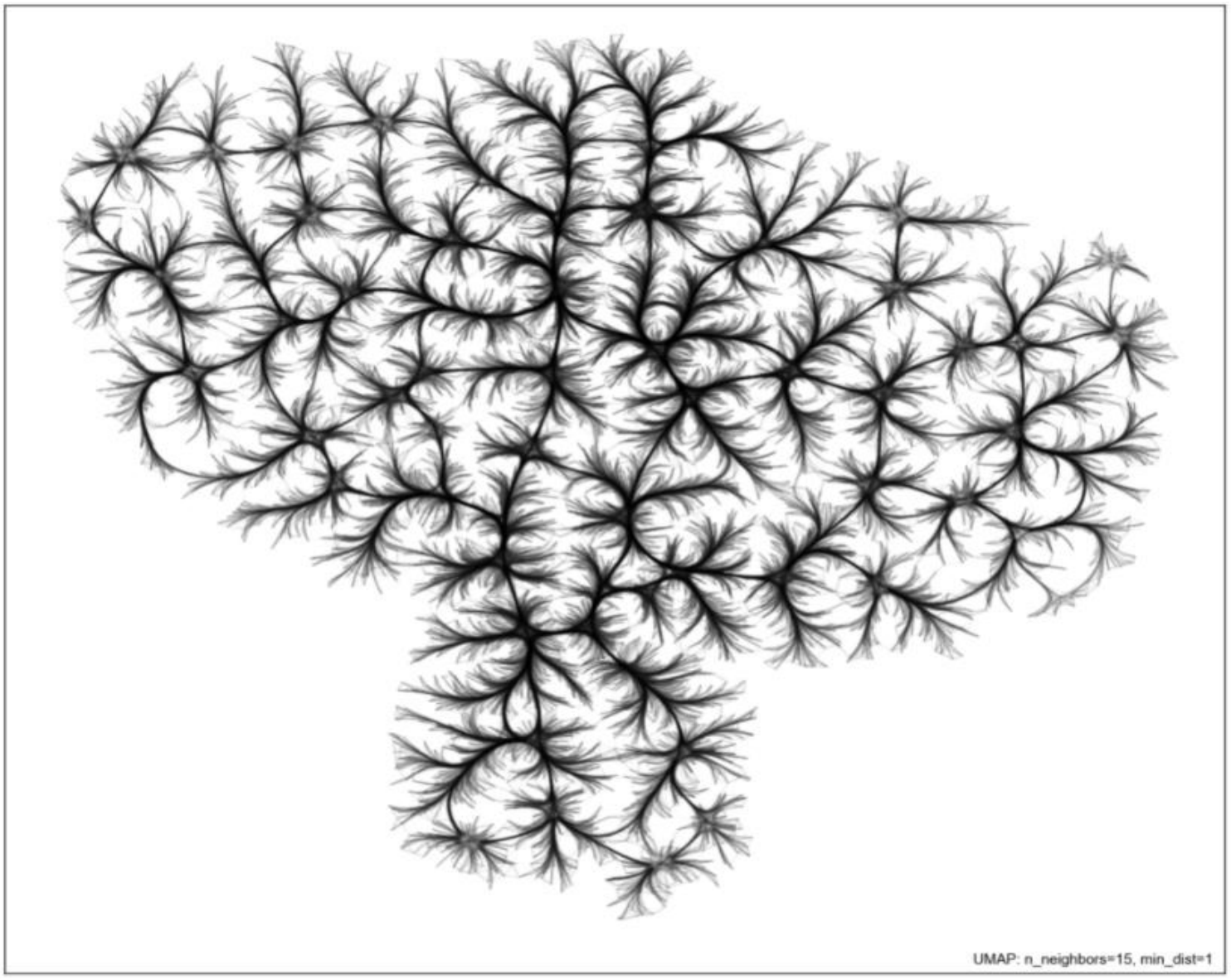
UMAP



UMAP connectivity

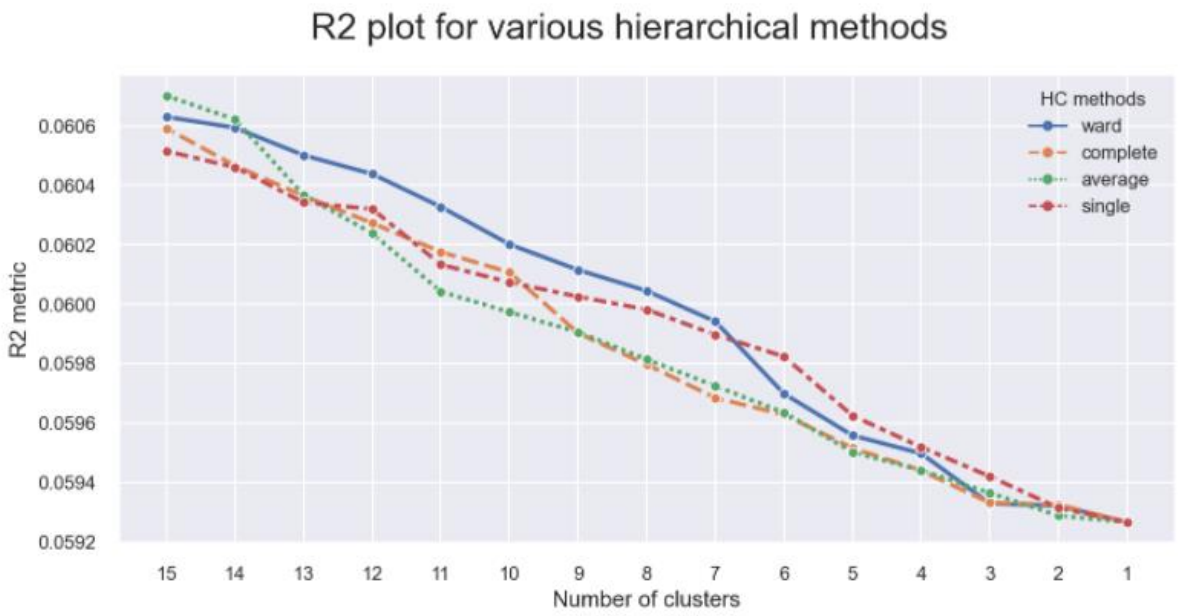


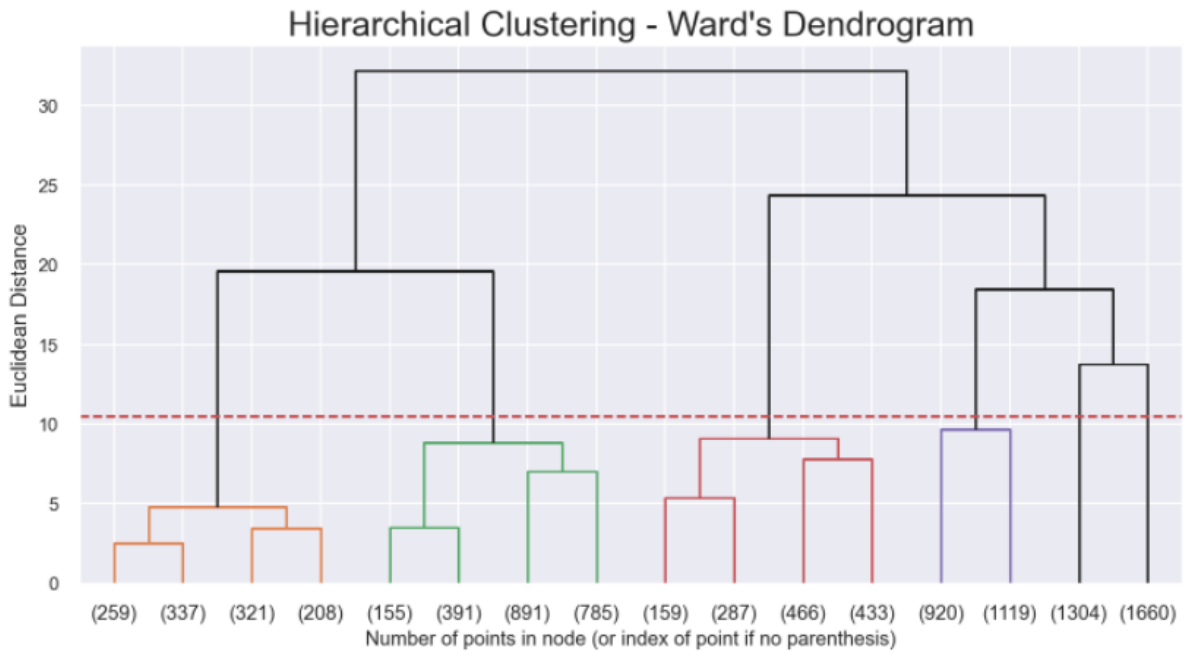
UMAP connectivity

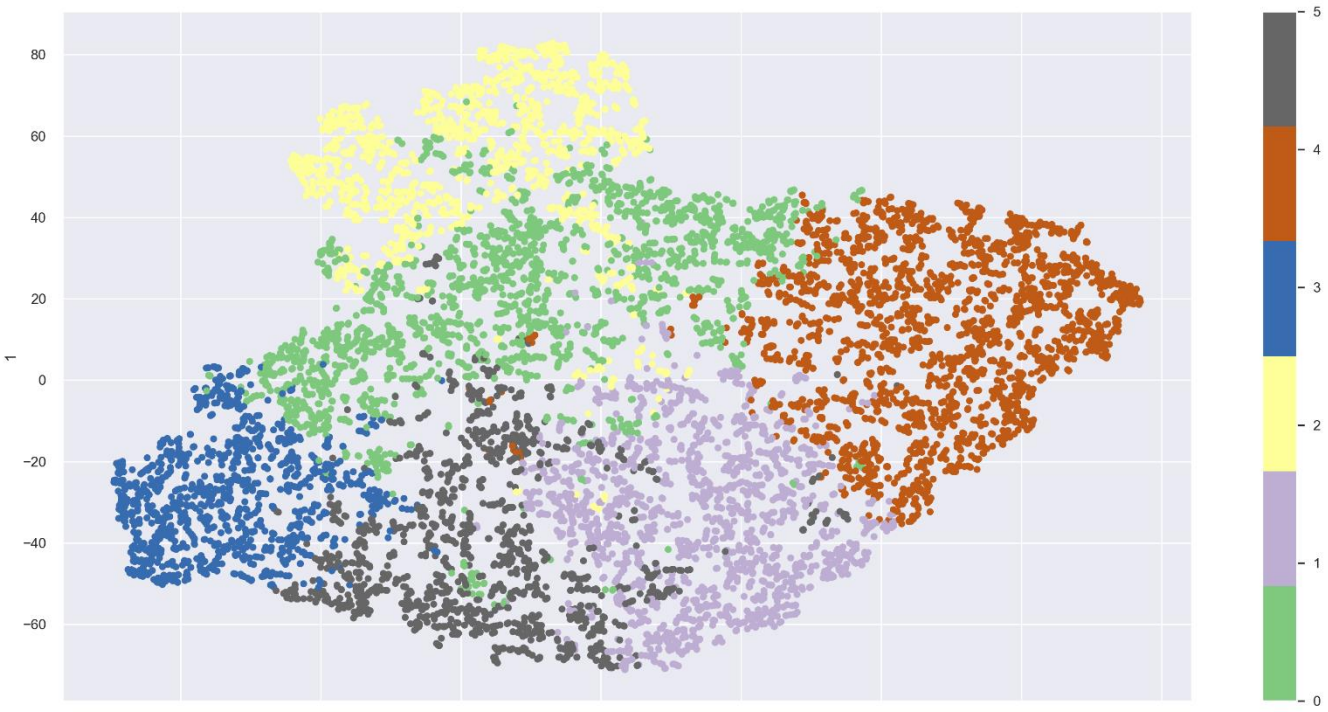


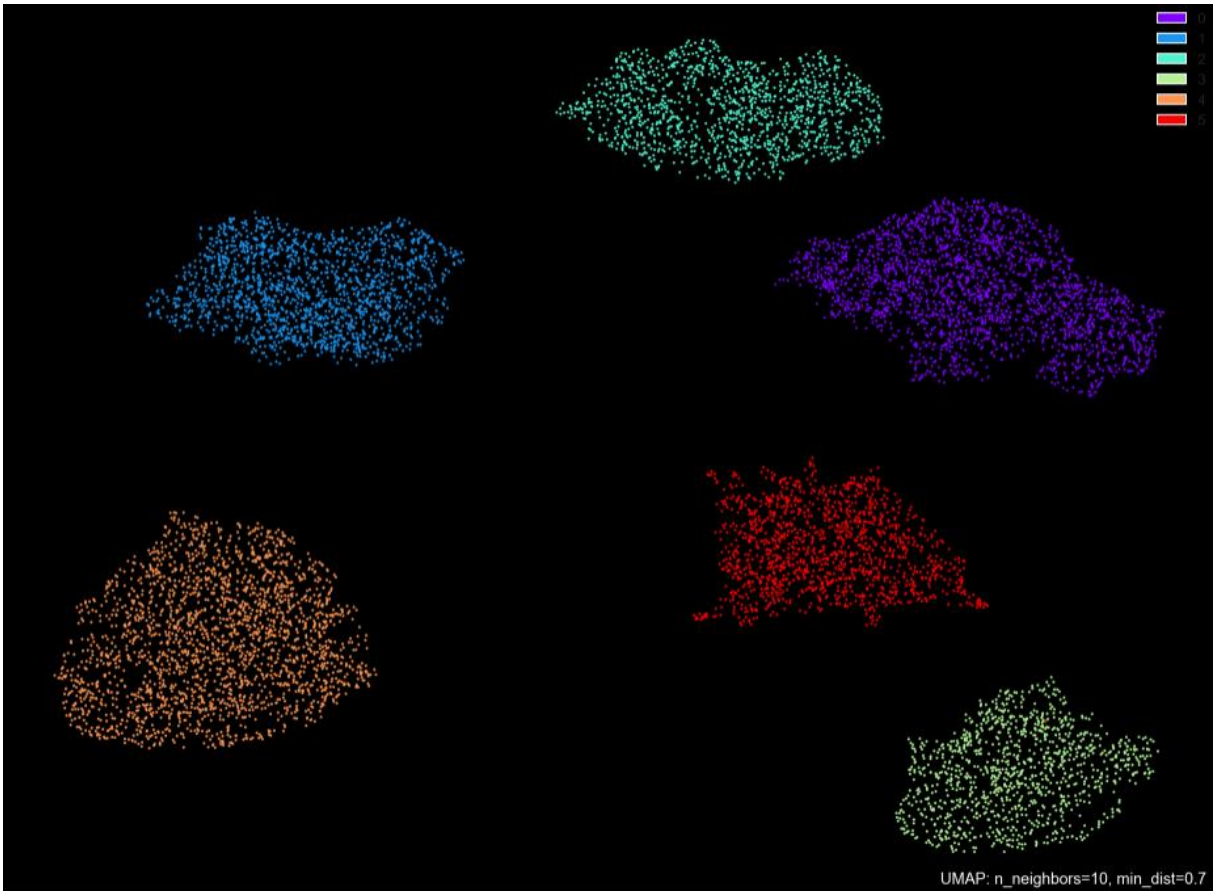
Hierarchical clustering – Metric features plots

R2 plot, dendrogram, T-sne and UMAP



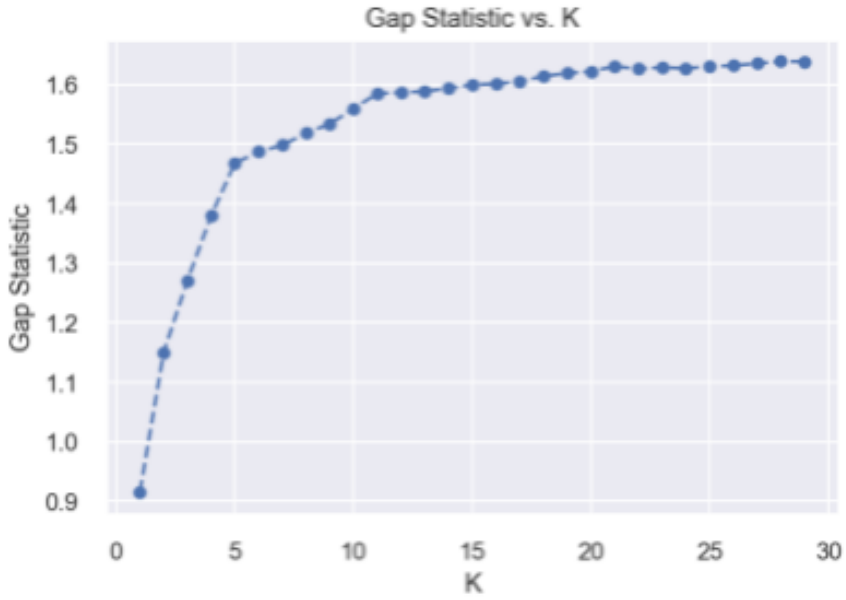


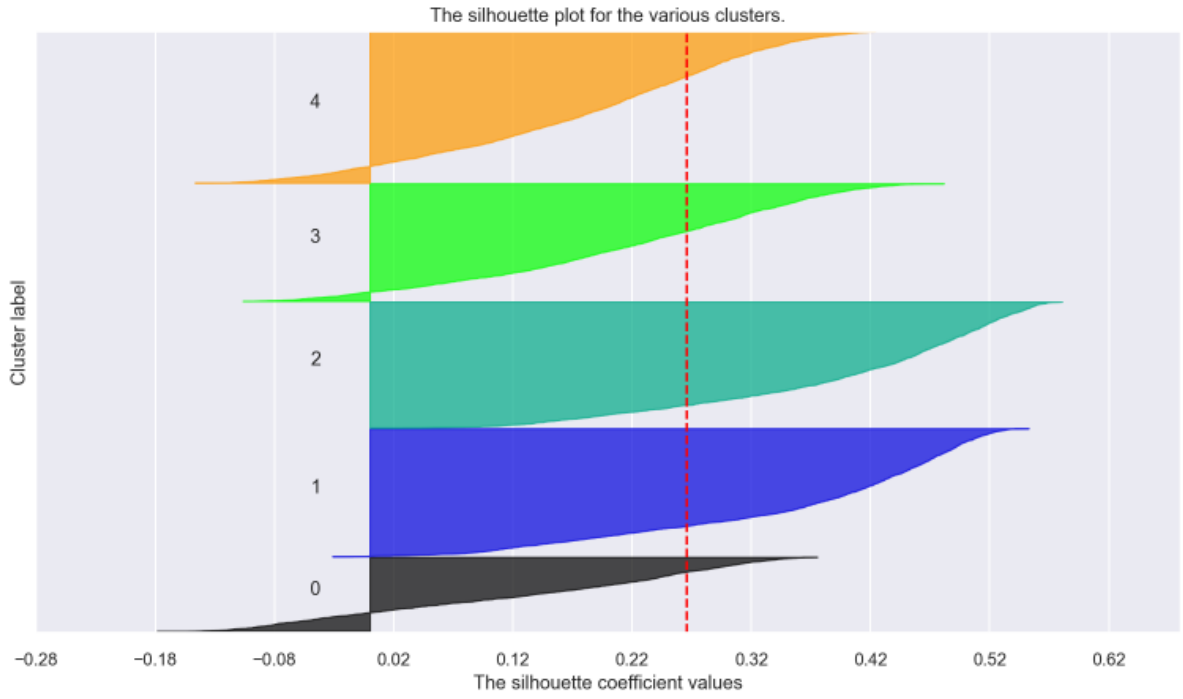

80
60
40
20
0
−20
−40
−60
1
5
4
3
2
1
0

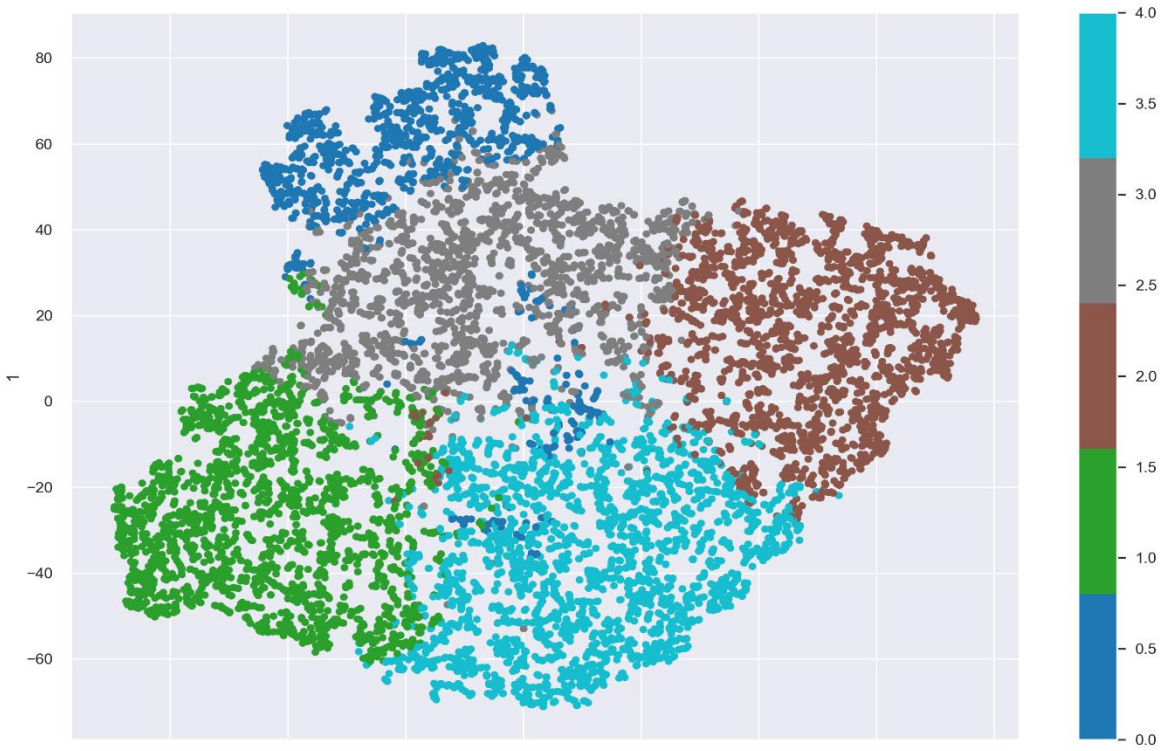

UMAP: n_neighbors=10, min_dist=0.7

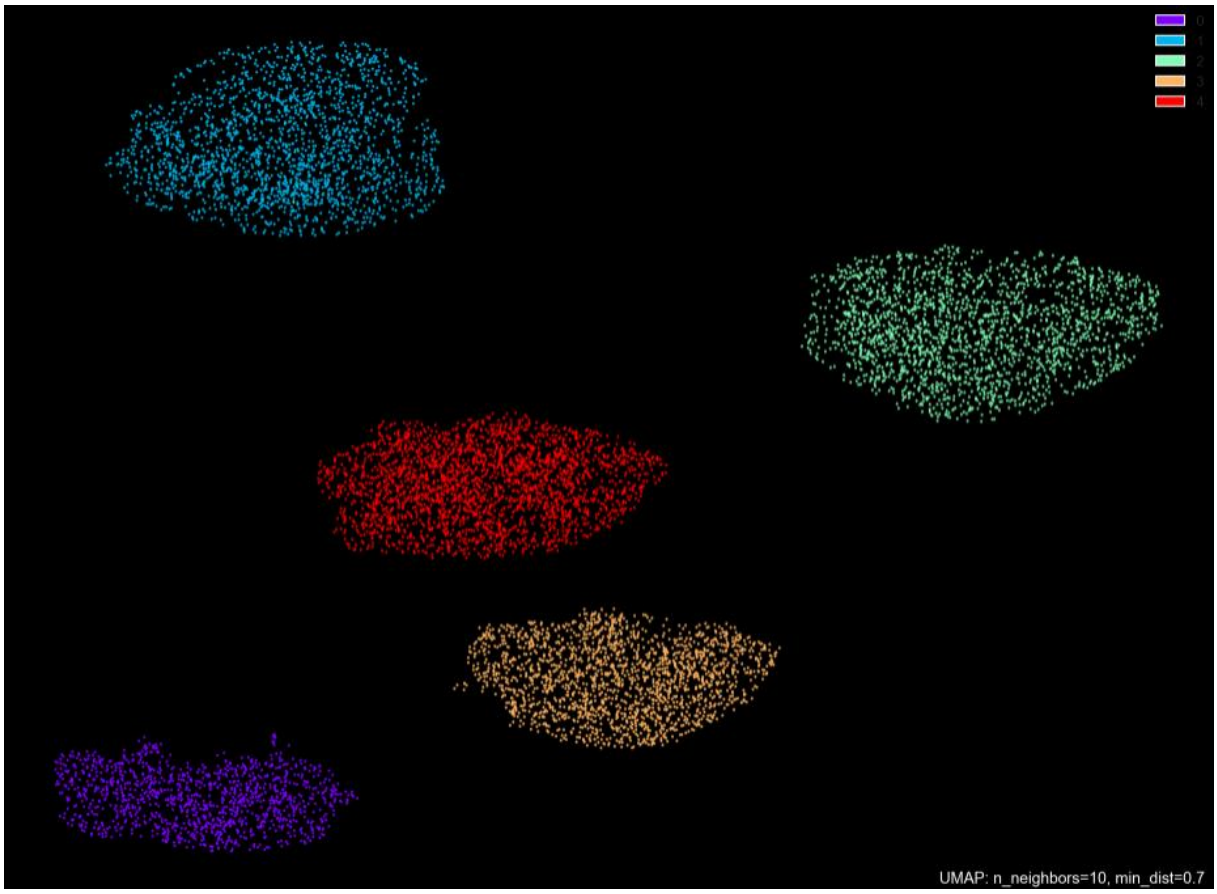
K-Means clustering – Metric features plots

Distortion Score, Gap Statistic, Silhouette score, T-sne and UMAP

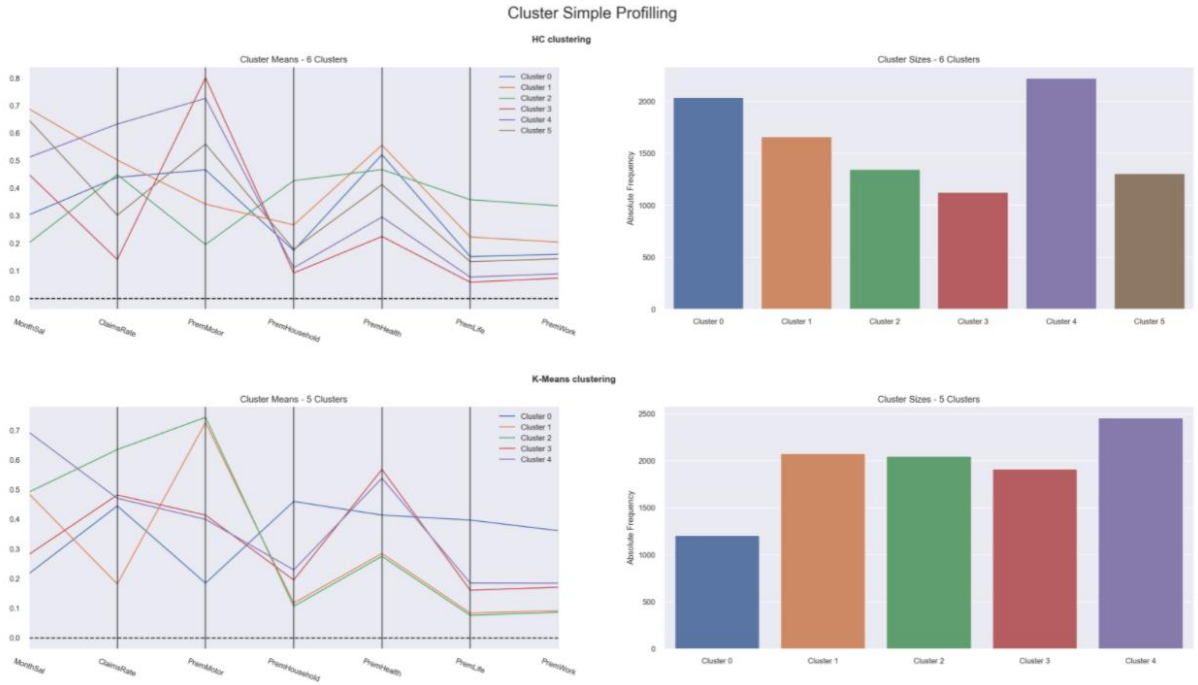



The silhouette plot for the various clusters.
Cluster label
4
3
2
1
0
−0.28
−0.18
−0.08
0.02
0.12
0.22
0.32
0.42
0.52
0.62
The silhouette coefficient values


80
60
40
20
0
−20
−40
−60
1
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0

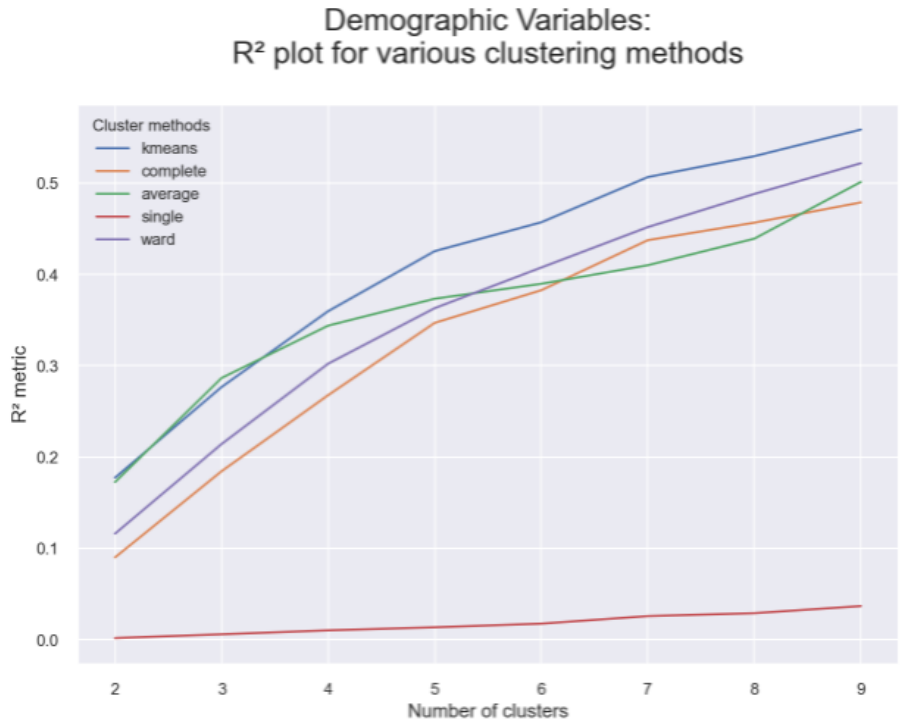


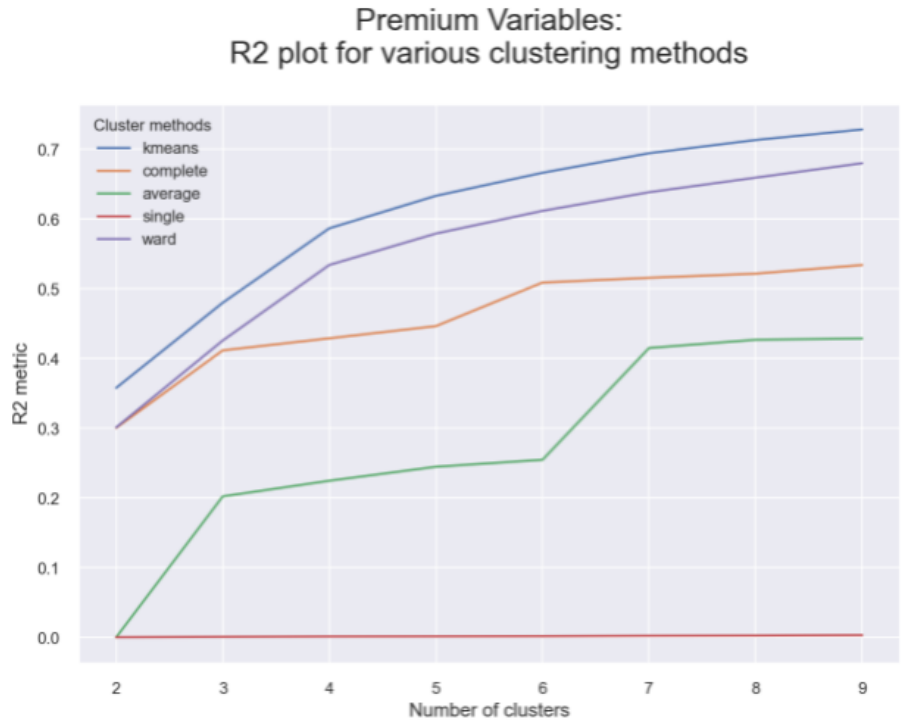
Profiling HC and K-means solutions

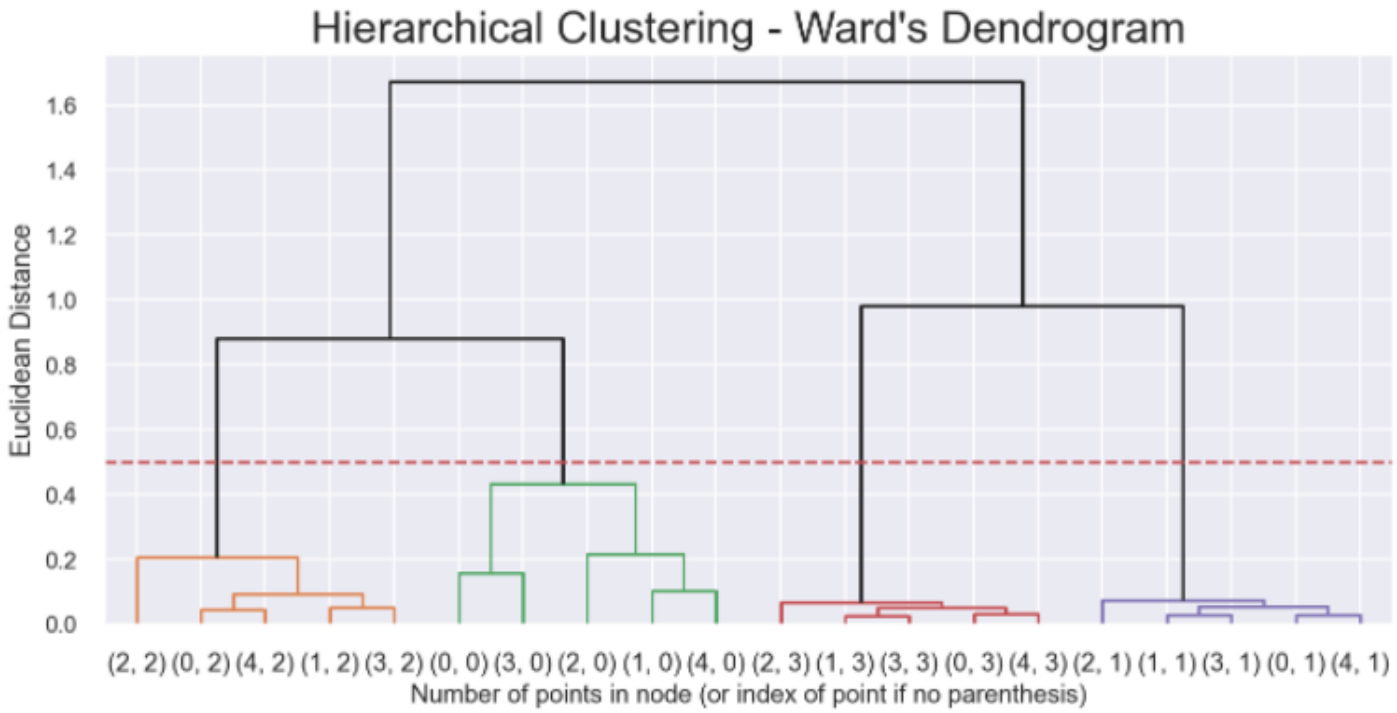


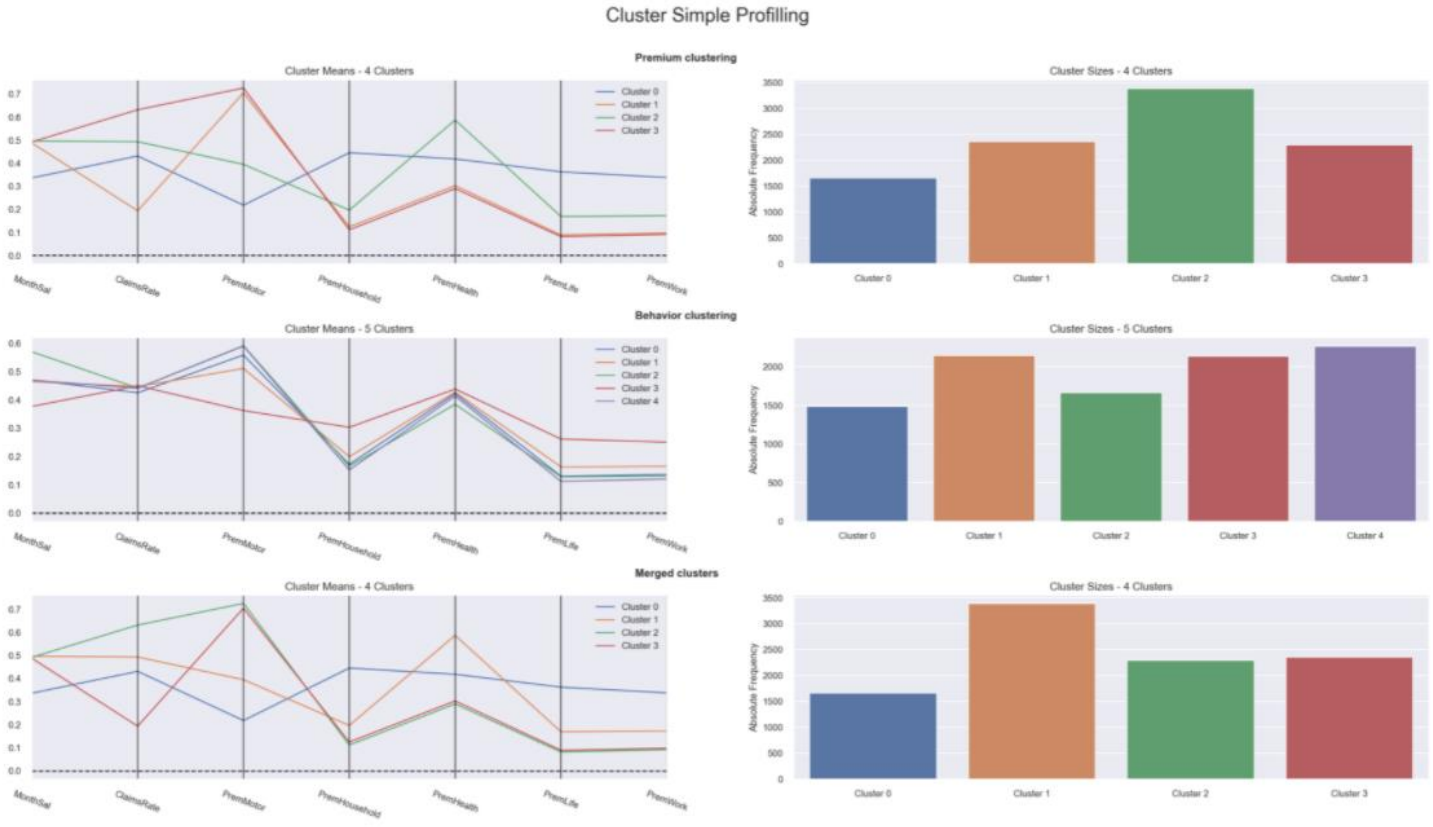
Different Segmentation

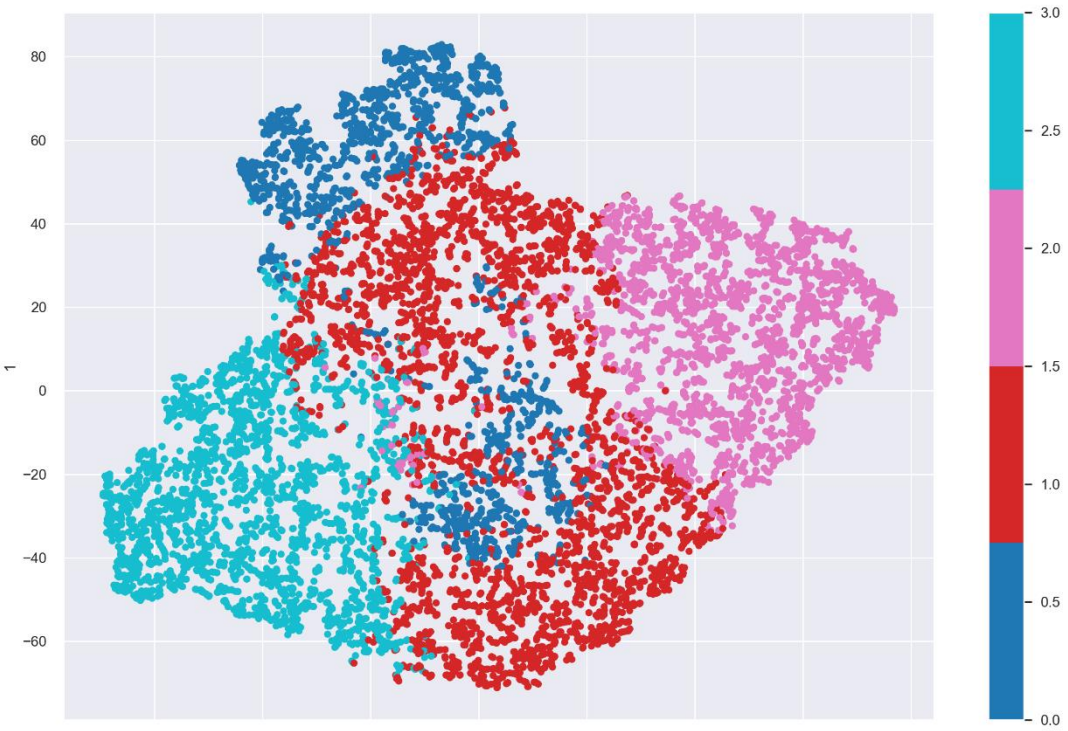
R2 plots, dendrogram, profiling, T-sne and UMAP

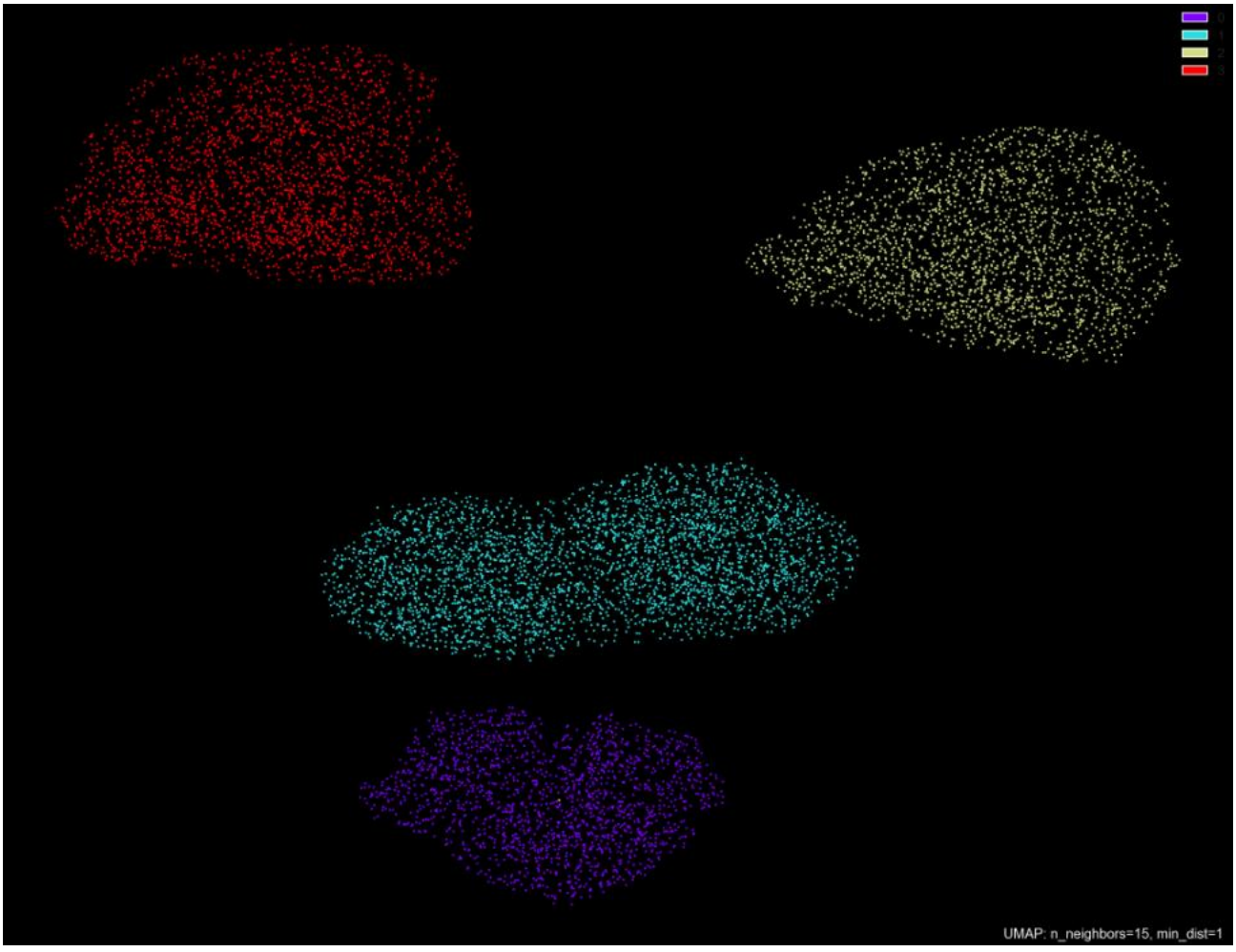





Hierarchical Clustering - Ward's Dendrogram
Euclidean Distance
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
(2, 2) (0, 2) (4, 2) (1, 2) (3, 2) (0, 0) (3, 0) (2, 0) (1, 0) (4, 0) (2, 3) (1, 3) (3, 3) (0, 3) (4, 3) (2, 1) (1, 1) (3, 1) (0, 1) (4, 1)
Number of points in node (or index of point if no parenthesis)


Cluster Simple Profilling
Premium clustering
Cluster Means - 4 Clusters
Cluster 0
Cluster 1
Cluster 2
Cluster 3
MonthSal
ClaimsRate
PremMotor
PremHousehold
PremHealth
PremLife
PremWork
Cluster Sizes - 4 Clusters
Absolute Frequency
Behavior clustering
Cluster Means - 5 Clusters
Cluster 4
Cluster Sizes - 5 Clusters
Merged clusters
Cluster Means - 4 Clusters
Cluster Sizes - 4 Clusters


80
60
40
20
0
−20
−40
−60
1
3.0
2.5
2.0
1.5
1.0
0.5
0.0


UMAP: n_neighbors=15, min_dist=1