NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Data Mining
Final Project

Alex Anthony Panchot (M20190546)
Hugo Saisse Mentzingen da Silva (M20190215)
Rennan Valadares Ornelas Araújo (M20190146)

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

Table of Contents

1. Introduction

This report addresses the final project in the Data Mining course of Master Degree Program in Data Science and Advanced Analytics of Nova IMS. The groups of students received a fictional insurance company database, containing personal customers' data as well as insurance consumption information. The features contained in the dataset are described in the Project Description document.

While the team impersonated a data mining consultancy, the final objective of this work was to "develop a Customer Segmentation in such a way that it will be possible for the Marketing Department to better understand all the different Customers' Profiles".

Therefore, we present in the following pages the process applied for this dataset treatment, the questions, findings, discussion, and recommendations that arose within it.

This report is accompanied by the file Data_Mining_Project.ipynb which contains all the Python code used.

2. Data Preparation

The dataset was initially converted into a Pandas DataFrame and the existing year-based columns (`Brithday Year` and `First Policy´s Year`) were rescaled to facilitate an easier interpretation, being substituted by `Age` and `First Policy´s Age`. For further analysis, the `Premium: Sum` feature was also added to the `insurance_df` DataFrame, representing the sum of all premiums paid by the customer.

2.1. *NaNs* and outliers

The initial assessment revealed 309 rows (3% of the total) with at least one blank cell (*NaN*) as well as the existence of some outliers or even noisy data (Figure 1). Instead of merely discarding the rows with `NaNs`, which could result in a significative loss of information, the team decided to apply techniques to estimate numerical and categorical features.

These techniques, regression and classification, applied to numerical and categorical columns respectively, first required the removal of outliers, since this provide better estimations.

Outlier treatment: since the dataset is not normally distributed, the team used the *interquartile range* (IQR) as the method to remove outliers. The usage of 1.5 IQR as the initial criteria for all columns revealed that some features had a large amount of observations to be considered as outliers (Household, Life and Work Compensations with more than 600 outliers each). In these especific cases, only the most extreme outliers were removed (distance higher than 3 IQR).

The criterion above resulted in at most 162 outliers for the one feature and 439 outliers in total (4.3% of the dataset) that were temporarily removed from `insurance_df`. The resulting data is presented in Figure 2.

Rows with more than 3 NaNs were to be dropped considering they could also undermine the estimators. However, there were no rows suiting this criterion.

2.2. 'Age' column assessment

The visual analysis of the `Age` feature motivated a deeper assessment regarding its reliability, considering many customers have `Age` lower than `First Policy´s Age`. In fact, 1702 customers are in this situation and, additionally, 77.3% of customers below 19 years old already have children, which is quite uncommon.

For this reason, `Age` was not considered in subsequent analyses.

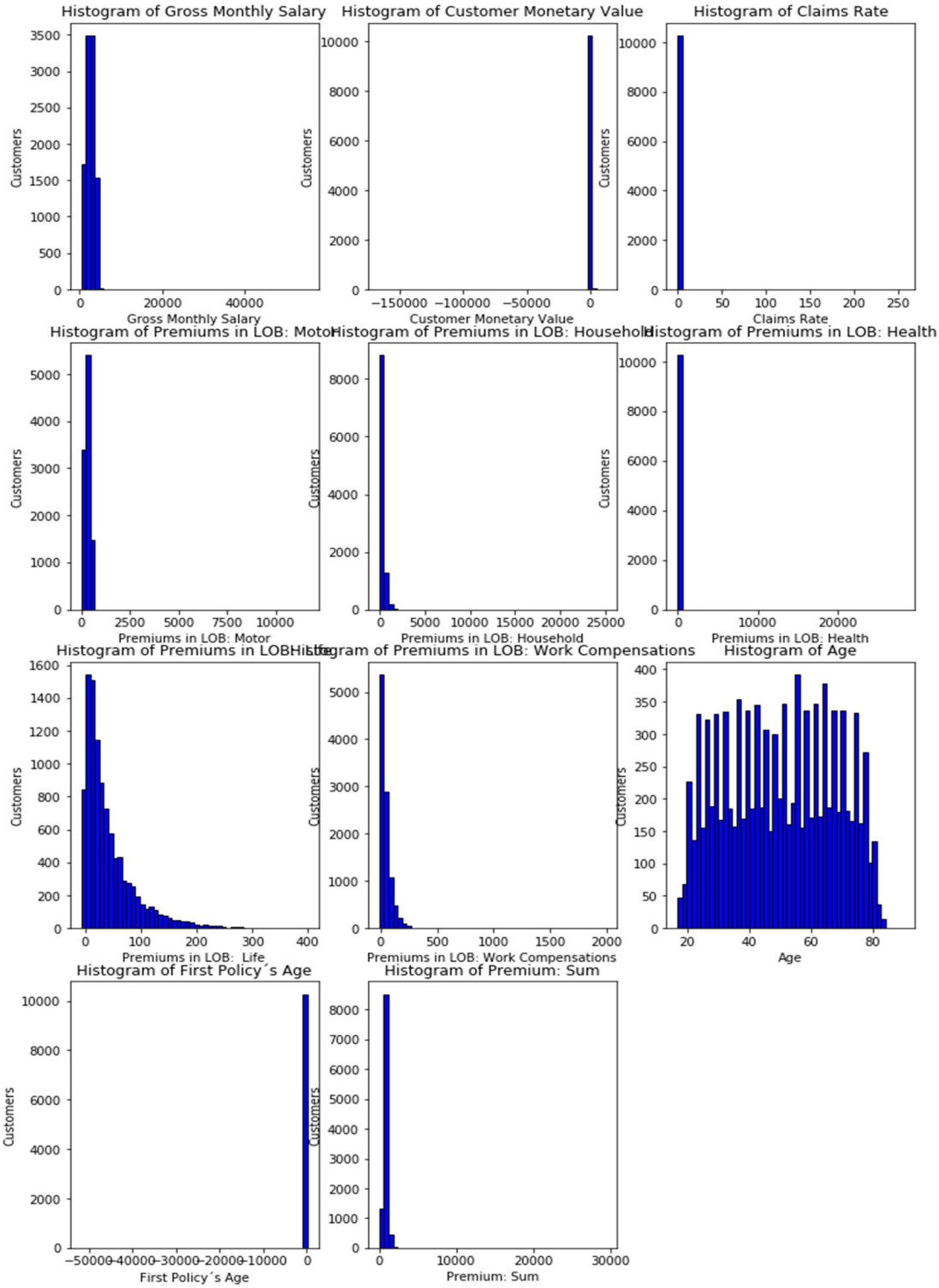


Figure 1 - Features histograms before removing outliers

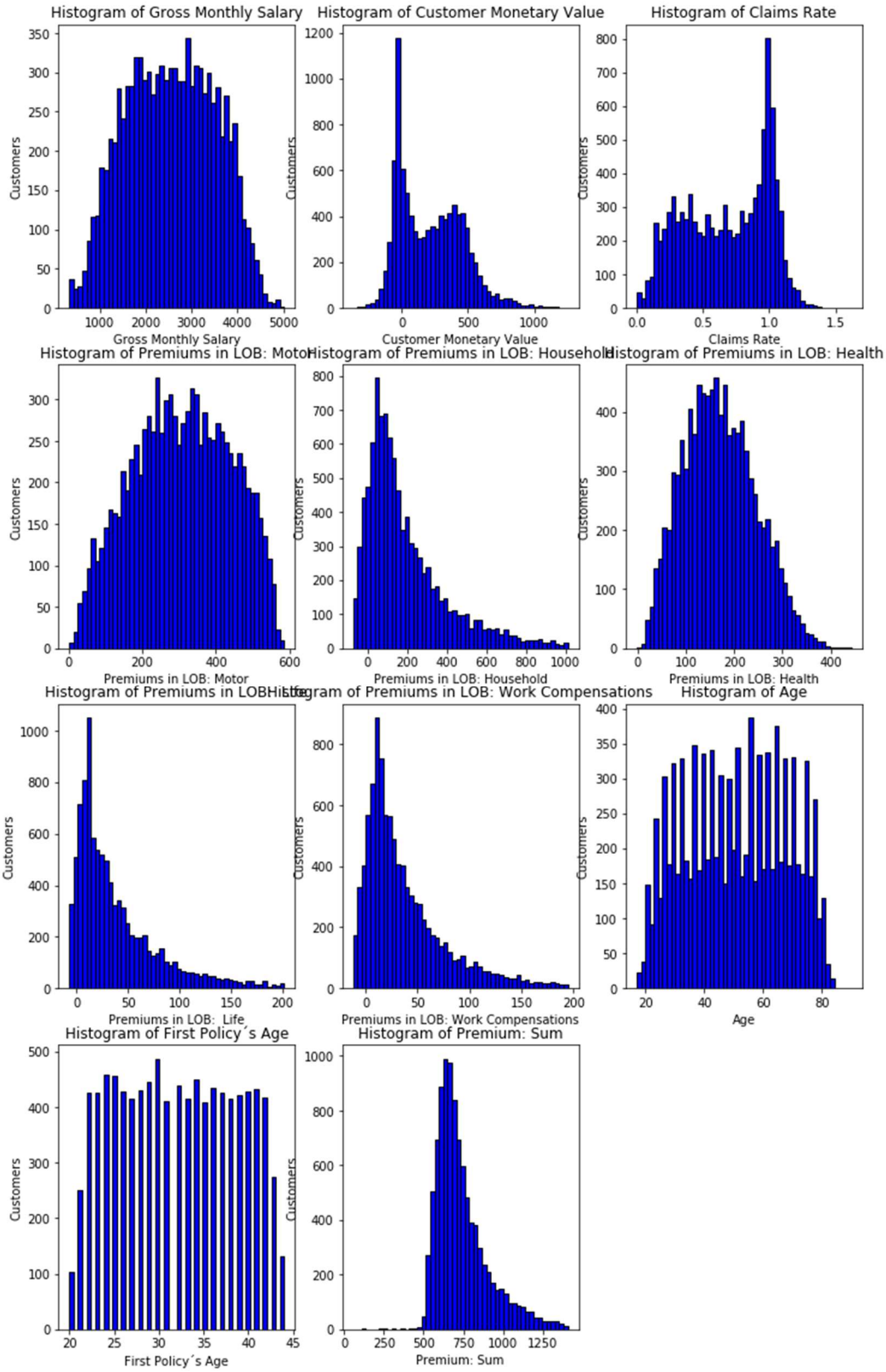


Figure 2 - Features histograms after removing outliers

2.3. Categorical features classification

Initially, the `Educational Degree` were encoded with numerical values by using the `LabelEnconder` from `scikit learn`. The numerical features were rescaled to be within the [0,1] range using the `MinMaxScaler`. The k-nearest neighbor algorithm `KneighborsClassifier` was then applied and its accuracy for each feature was evaluated with a custom function which computes a confusion matrix for each trained model. The results are shown in Figure 3.

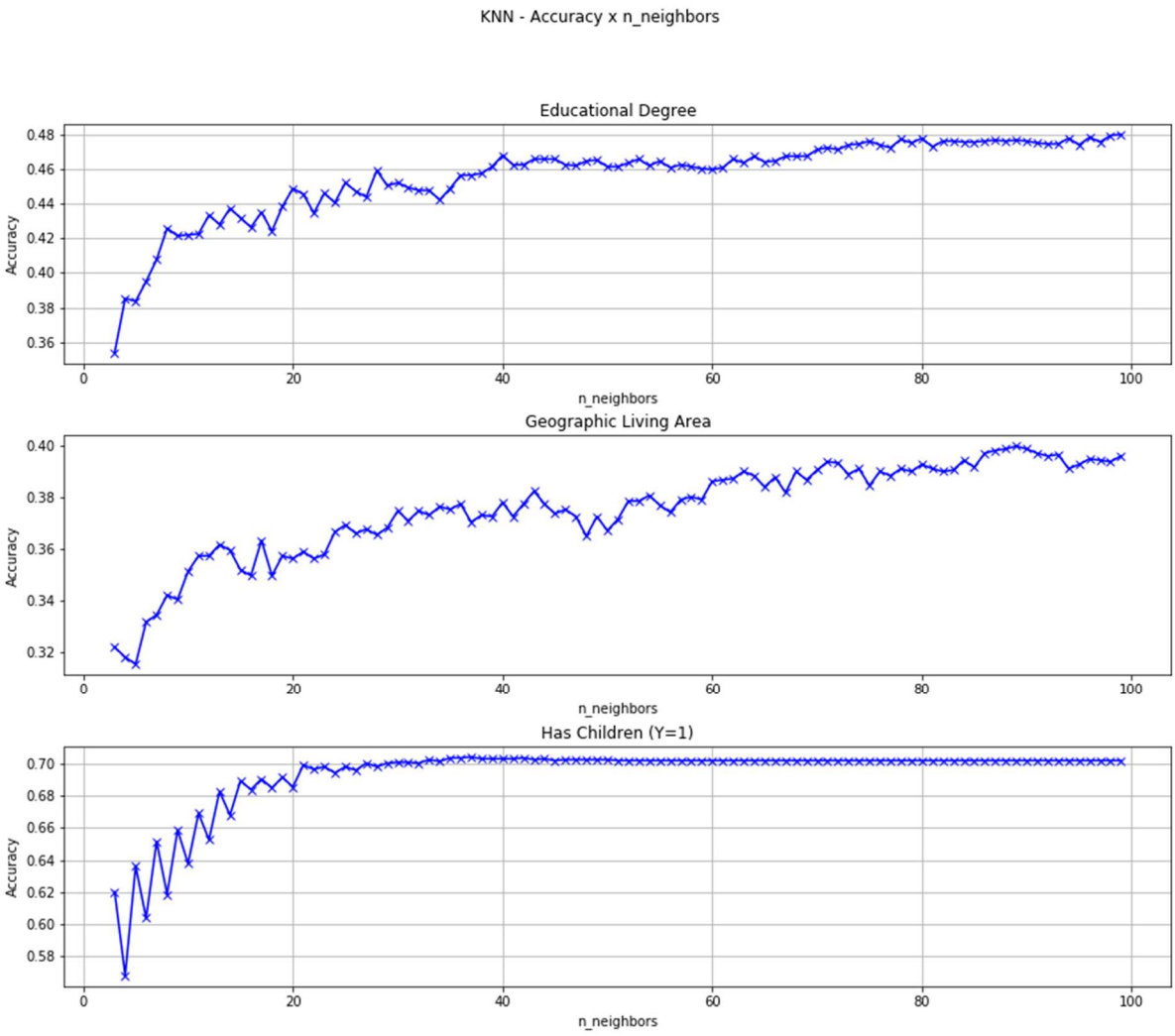


Figure 3 - Classifier accuracy for each categorical feature

Since we didn't achieve good estimations for the first two categorical features, the team decided to state the minimum accuracy at 2/3 and to drop the rows which contained `NaNs` on these two columns. The optimal value for the number of neighbor samples (`k`) to `Has Children (Y=1)` was set to 21.

`k` choice for the kNN algorithm: the 'rule of thumb' is to choose `k = sqrt(n)` where `n` stands for the number of samples in the training dataset and `k` is the number of instances that we take into account to determine affinity with classes. Since we chose 80% of the data to be the training dataset, `k` would be around 89. Figure 3 shows that there would be no improvement in selecting a number higher than 21 for `Has Children (Y=1)`.

2.4. Numerical features regression

The first approach to fill the Numerical NaNs was to create a function that trains three different regressors, split the dataset into complete (rows without NaNs) and incomplete (rows with at least 1 NaN) and then check the R Squared Error (R2E) of each algorithm. We're going to use the regressor with the best R2E. The algorithms tested are:

- ✓ DecisionTreeRegressor
- ✓ LinearRegressor
- ✓ LinearSVR

Using this technique only ensured that we would have the smallest MSE out of these 3 algorithms, but it was not enough to achieve good results. The parameterization of the correct algorithm would be crucial for the result to improve. For this, DecisionTreeRegressor was selected as an algorithm and a Genetic Algorithm (GA) implementation was made to try to achieve a good R2E. We will not delve into the explanation of the generic algorithm, but it explained in Appendix A along with the code being availible with the GitHub link that will be at the end of this report.

The group decided to use the DecisionTreeClassifier with the result of the GA only in the columns that it achieves more than 0.60 R2E and the rest will be dropped.

The result of the Genetic Algorithm showed that only in the `Premiums in LOB: Motor` and `Premiums in LOB: Life` columns the results were above 0.60, being 0.8157 and 0.6286 respectively. The NaN values of the cited columns were filled according to the parameterization selected by the GA.

2.5. Drop the rest of NaNs

After applying the regressor we still had 206 NaNs. We dropped the rest of them.

3. Data preprocessing

This chapter presents the techniques applied to analyze the data and the findings that emerged from them.

3.1. Features correlation

Plotting the correlation matrix for all features (Figure 4) we can see that `Age` is highly correlated to `Gross Monthly Salary`. Since `Age` is also an untrustworthy feature (see section 2.2) and the amount of information would remain in the correlated feature, the team decided to drop the `Age` column for the analysis.

The `Claims Rate` also has a high inverse correlation with `Customer Monetary Value` (CMV), which makes sense since the Claims Rate is related to the profit that the company makes with each customer. So, the team decided to again reduce the input space and later cluster the dataset only on the CMV perspective, since it carries more information than Claims Rate (includes customer retention and acquisition cost).

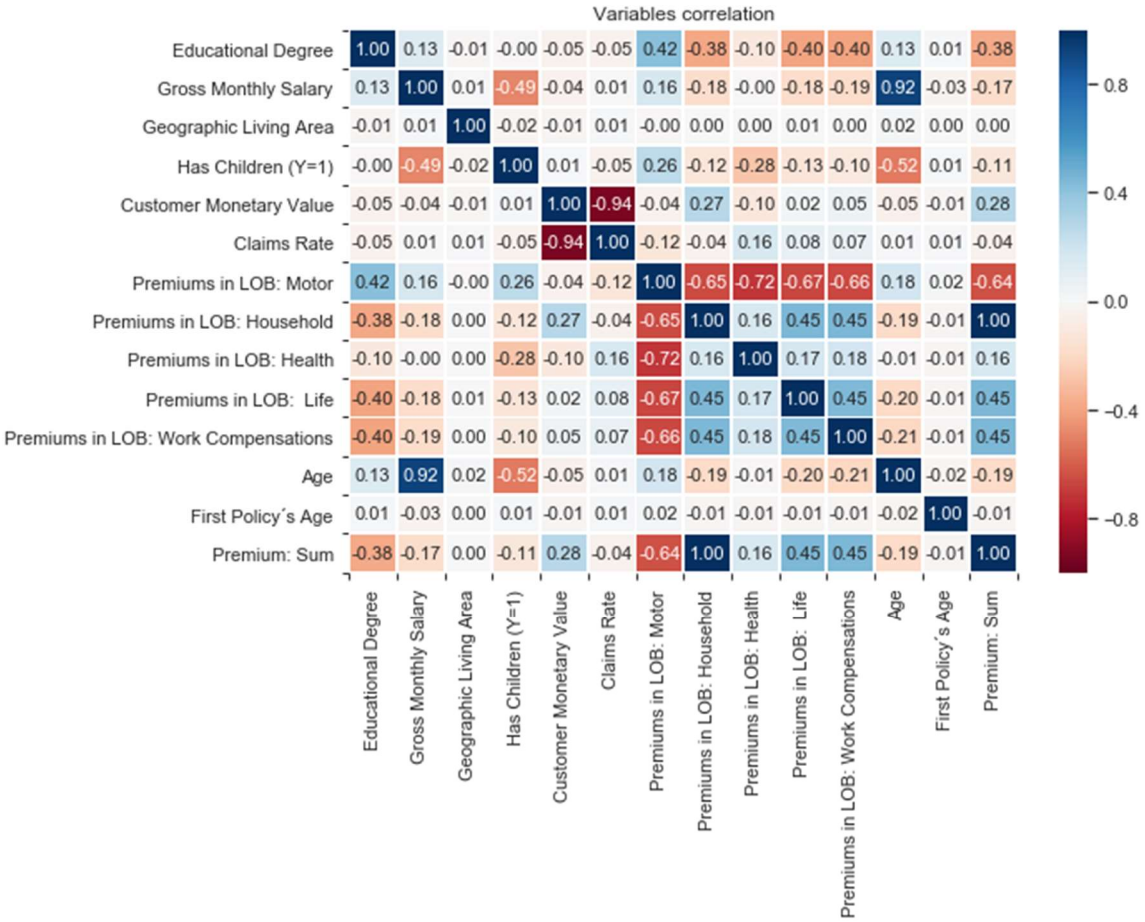


Figure 4 - Correlation matrix and heatmap for all features

What also calls our attention is that `Premiums in LOB: Motor` Insurance is considerably correlated (inversely) with all other insurance premium totals. Considering that signing a policy in any group could lead to increase proneness to contract other kinds of insurance (complementary products), this behavior is unexpected. Taking into account that the mean premium for Motor is substantially higher than for the other groups (Figure 5), it's contracting may be affecting the capacity or interest in paying for the other products. A possible insight would be to focus more on reverting this tendency.

	Sum	Mean	Median
Premiums in LOB: Motor	$ 2,945,215.76	$ 305.33	$ 307.28
Premiums in LOB: Household	$ 1,844,924.60	$ 191.26	$ 128.90
Premiums in LOB: Health	$ 1,641,113.59	$ 170.13	$ 165.03
Premiums in LOB: Life	$ 366,624.99	$ 38.01	$ 24.56
Premiums in LOB: Work Compensations	$ 357,539.00	$ 37.07	$ 24.67
Premium: Sum	$ 7,154,165.69	$ 741.67	$ 694.57

Figure 5 - Premiums measures

`Premiums in LOB: Household`, `Premiums in LOB: Life` and `Premiums in LOB: Work Compensations` correlation show that these groups are decently working as complementary products, and its correlation could also be improved.

Finally, it’s possible to note a perfect correlation between the amount in Household insurance and the sum of the premiums, a fact that can also be commercially explored by the company.

4. Analysis

4.1. Quartiles (a priori grouping)

Separating the premiums data into quartiles confirms the inverse correlation between Motor insurance consumption and the sum of premiums (Figure 6). One good strategy would be to focus on inverting the correlation. Regarding the other insurance groups, specifically Health, Life and Work Compensations, they show a scattered but still correlated distribution over the quartiles. Possible approaches for these findings would be to take actions in order to bring customers closer to the main diagonal, i.e., increase the premiums total for customers that already have high premiums in any of these groups, and increase consumption on specific groups when the customer has already a high premiums total.

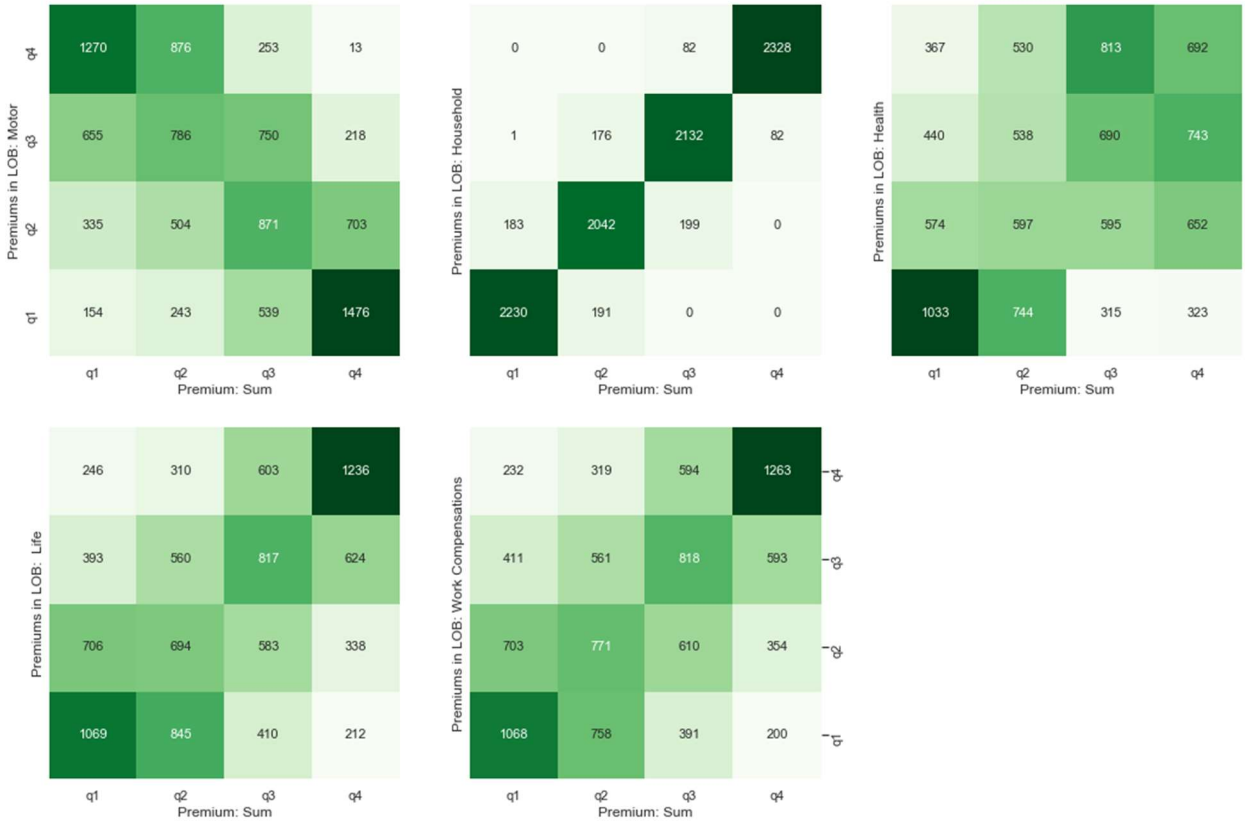


Figure 6 - Quartiles plot for insurance groups vs premium sum

Another perspective also analyzed by the team was the premium amount for each group versus the gross monthly salary (Figure 7), which, in principle, could reveal the customer potential for contracting insurance. Ther quartiles show that this potential is not being totally tapped. Stands out, for instance, the customers' group in Q3 for `Gross Monthly Salary` and Q1 for `Premiums in LOB: Health`.

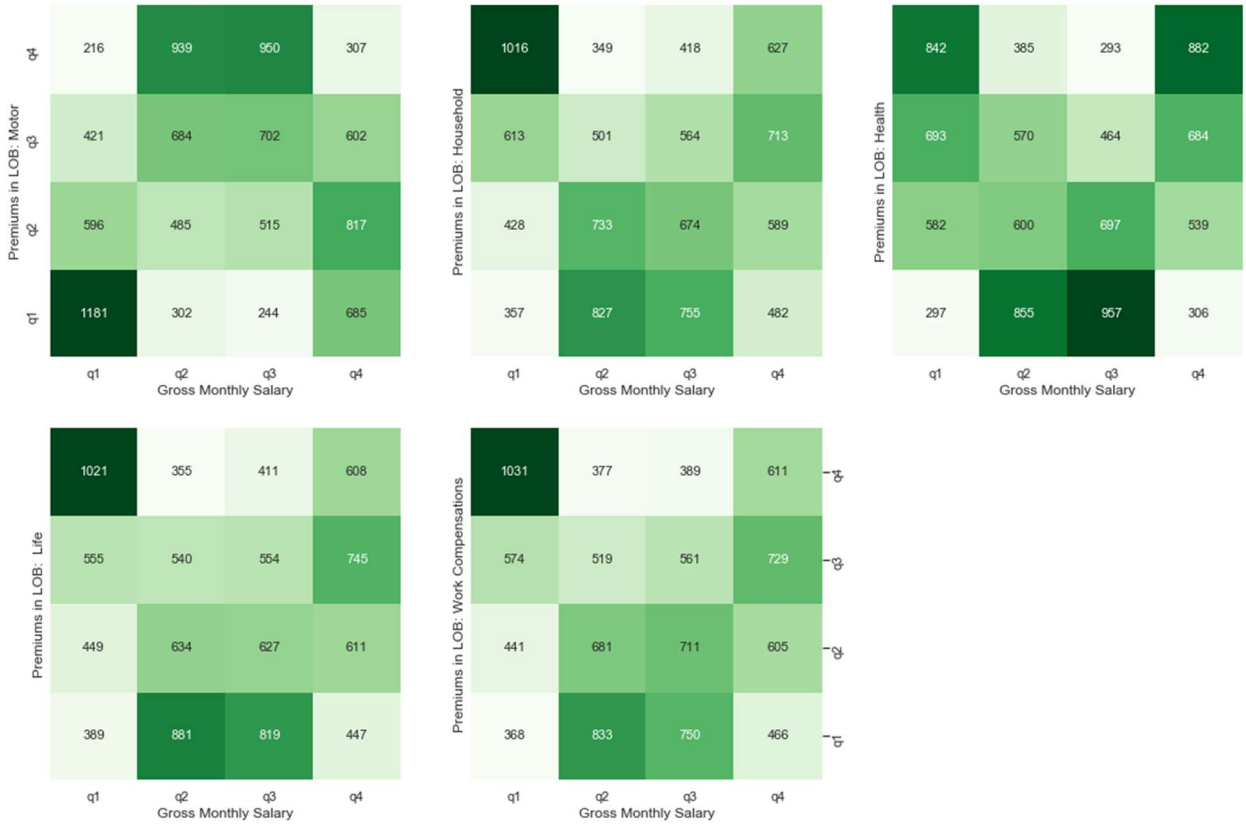


Figure 7 - Quartiles plot for insurance groups vs gross monthly salary

4.2. Boxplots

A further investigation continued with boxplots. The team was particularly interested in understanding the changes in the distribution of the insurance consumption features under the different customer categorical features.

Considering that people with children and higher education levels would be more prone to pay higher premiums in Health and Life since the premium is proportional to the insured sum and the coverages, there is an unexpected behavior on the plots of Figure 8 and Figure 9. The same occurs with Work Compensations and Household, and the opposite is observed for Motor insurance. It seems that clients with high insurance buying potential are not being accessed for events that can cause major changes in their lives.

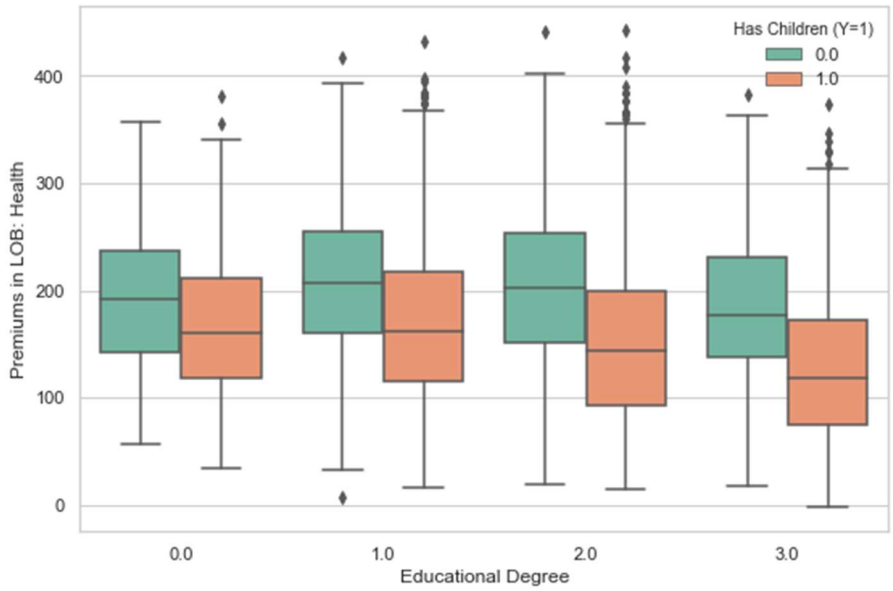


Figure 8 – Health premiums distribution on the educational degree and children in the family.

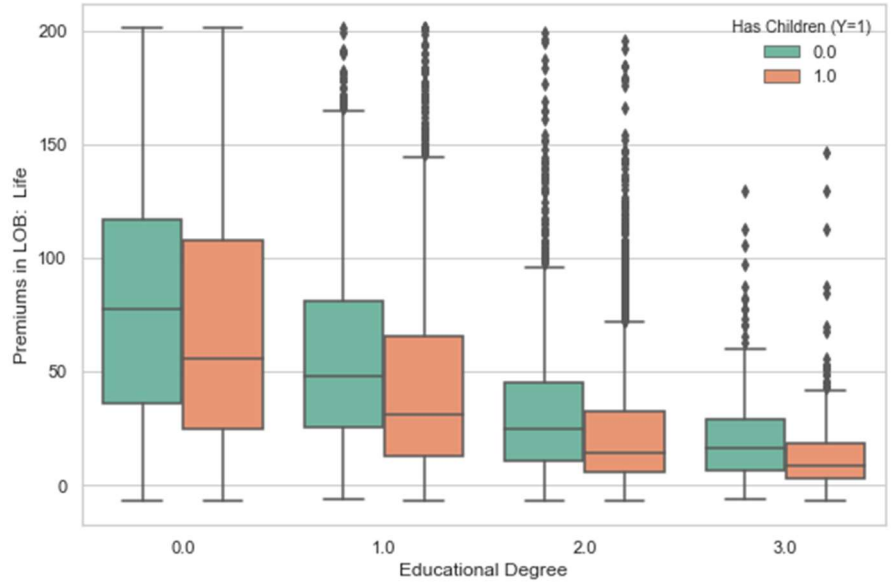


Figure 9 – Life premiums distribution on the educational degree and children in the family.

4.3. Decision tree

Trying to predict the `Premium: Sum` or the `Customer Monetary Value` quartiles from a customer profile (Educational Degree, Geography, Children, Salary and First Policy´s Age), resulted in a maximum accuracy of only 0.357. On the other hand, the graph confirms the behavior observed so far and reveals that geography is not a relevant variable for customer potential measurement (Figure 10).

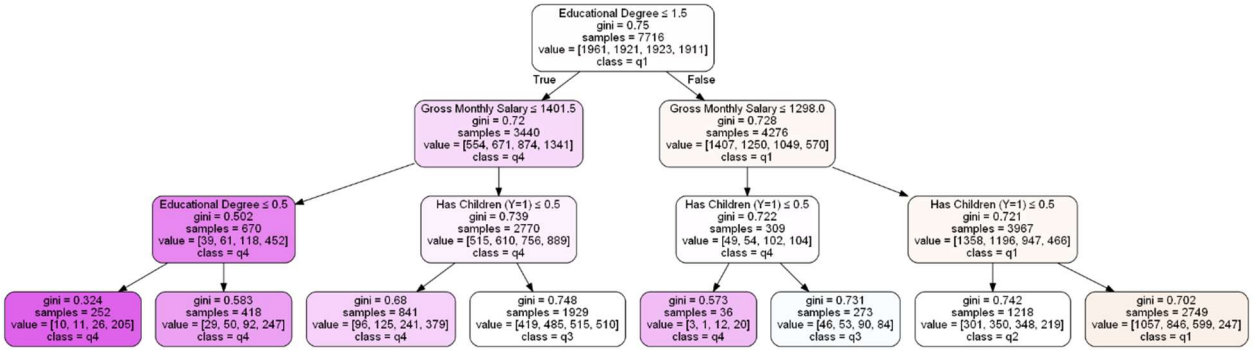


Figure 10 - Decision tree for quartiles of Premium: Sum

5. Clustering

5.1. Groupby Table

After removing all of the outlying data, we can try to build clusters in order to better understand our customers. As categorical and continuous data cannot be easily used together in a clustering algorithm, we can begin by looking at just the categorical data. We can use a groupby to separate the data into unique groups (similar to the RFM method).

As `Customer Monetary Value` is a good target variable (it shows a customer's value to the company), we can aggregate it with either the mean or median and then group it by each categorical data. We can then sort each group by its `Customer Monetary Value` average such that we see which set of categorical parameters are on average better or worse for the company

As the dataset no longer includes any significant outliers, we should expect that the median and mean should be nearly the same value. However, we see that only the mean returns a meaningful result for `Has Children (Y=1)` and `Educational Degree`. Customers with no children have a slightly higher average claims rate than customers with children. The six groups with the highest "CMV" average all have a basic education. The rest of the groups are randomly spread across the different education levels. This suggests that people with the lowest level of education have the highest value to the company, while other levels of education have not only a lower level of value to the company, but also cannot be differentiated from each other.

Educational Degree	Geographic Living Area	Has Children (Y=1)	
0.00	2.00	0.00	332.48
	3.00	0.00	271.75
	1.00	0.00	268.41
	3.00	1.00	261.13
	4.00	1.00	259.15
		0.00	247.94
3.00	3.00	0.00	243.83
0.00	1.00	1.00	238.81
	2.00	1.00	235.20
3.00	1.00	1.00	228.13
1.00	2.00	1.00	223.84
	1.00	0.00	222.21
	3.00	0.00	220.98
	4.00	1.00	220.89
2.00	3.00	1.00	219.96
	1.00	1.00	218.65
3.00	4.00	1.00	213.58
2.00	1.00	0.00	211.72
1.00	1.00	1.00	209.74
2.00	2.00	0.00	208.34
	4.00	1.00	206.36
1.00	4.00	0.00	203.95
	3.00	1.00	203.51
3.00	2.00	1.00	203.45
1.00	2.00	0.00	197.96
2.00	2.00	1.00	195.13
	4.00	0.00	191.54
	3.00	0.00	189.52
3.00	3.00	1.00	187.27
	1.00	0.00	182.17
	4.00	0.00	181.19
	2.00	0.00	154.07

Name: Customer Monetary Value, dtype: float64

Figure 11 – Grouped categorical features (CMV as target variable)

While these observations disappear when using the median as the average, we see that the averages have a larger range. This is most likely a result of the mean averaging the people with a ratio of 0 and the median taking the 50th percentile of customers in that group.

Using count instead of the average also reveals that half of our customers have children, live in areas 1 or 4 and have an education of 1 or 2. Using this data can help to target advertising towards potential customers. This gives us 4 groups to target instead of the 32 possible groups.

5.2. K-Means

We can use the k-means algorithm to cluster the numerical data. In order to do this, we must define the number of clusters. For this we can run the k-means algorithm several times and build an elbow plot. We can also build a dendrogram to confirm the results in the elbow plot.

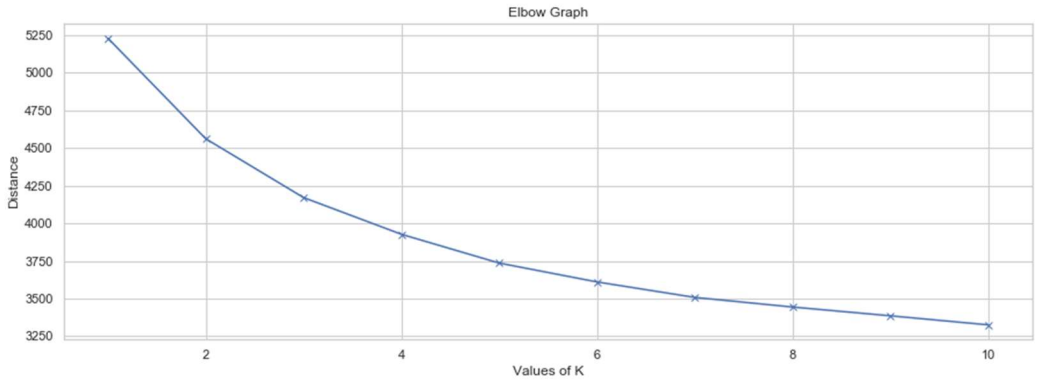


Figure 12 - Elbow graph for clusters with the numerical features

From the elbow plot in Figure 12, we can see that a significant elbow does not really appear, but a slight one appears at either 2, 3 or 4 clusters. We can create a dendrogram to compare (Figure 13).

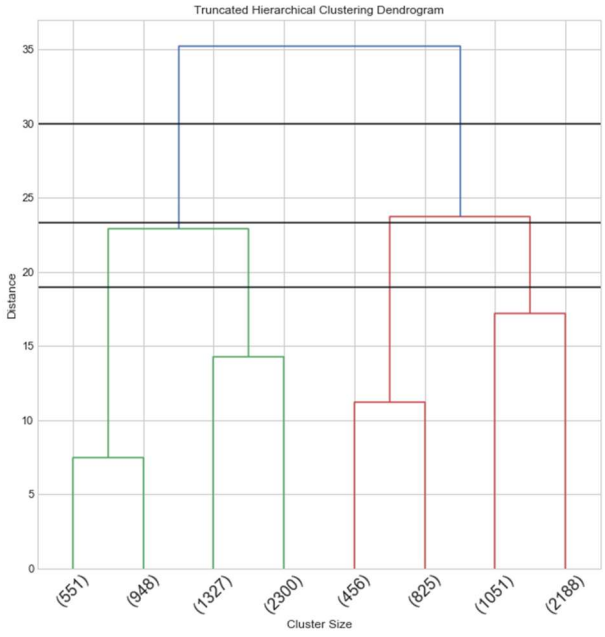


Figure 13 - Dendrogram for clusters with the numerical features

The black horizontal lines are drawn such that they cut the dendrogram into 2, 3 and 4 clusters. Looking at the number of customers in each branch (indicated by the number at the bottom), we can see that using 3 would create a cluster of around 1000 customers while the other clusters are much larger. 4 clusters seem appropriate as

using 4 reduces the distance significantly over 2 while also keeping the number of clusters similarly sized.

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	3,057.94	228.44	238.10	248.88	203.09	50.84	47.56	34.55
1	2,549.88	182.82	368.54	98.82	161.26	19.63	19.28	27.47
2	1,364.08	266.28	130.04	458.81	187.72	88.48	90.96	30.72
3	2,416.87	217.92	447.46	61.81	105.35	11.62	11.37	38.43

Figure 14 - Cluster means for dendrogram

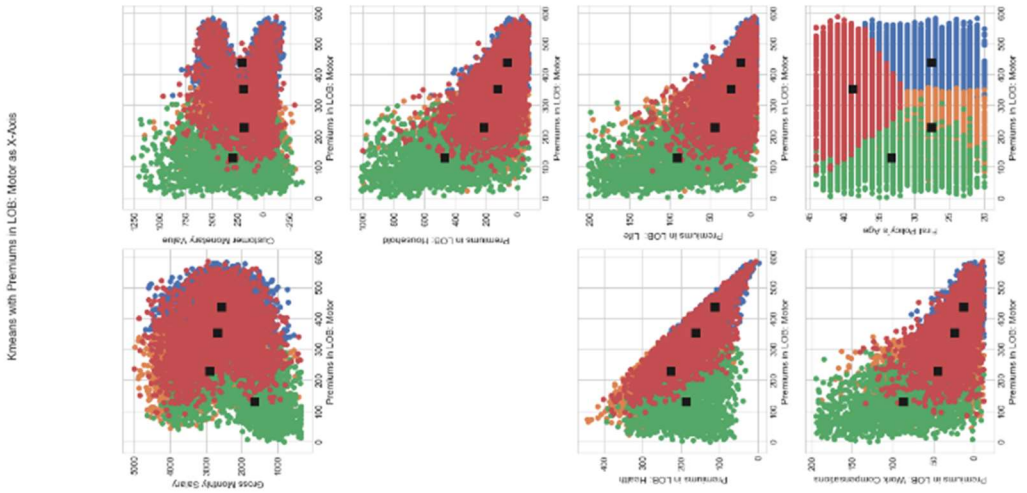


Figure 15 - K-means with 4 clusters (Motor premiums in X axis)

Running the k-means with 4 clusters does not give a satisfactory result when plotting any two random variables. Each graph is plotted with the `Gross Monthly Salary` as the x-axis. If we look at Figure 15, we can see that the clusters in the first section have no meaning as they simply break the data into two equal parts (all of the data in the section is plotted in a blob). Plotting with a different x-axis (`Premiums in LOB: Motor`) in Figure 15 returns more interesting shapes (triangles rather than the circles or squares here). As the triangles show a negative correlation, we can predict that customers who spend a lot on motor insurance spend less on other insurances. All of the 2 variable plots are presented in the appendix.

5.3. Mean Shift

Using mean shift is in theory a good choice as it will allow us to find clusters without specifying the number of clusters beforehand. However, there can be a problem as we need to specify the bandwidth used in the mean shift algorithm. If we do not specify it, the algorithm can estimate it. For this dataset the algorithm finds 3 clusters. If we change the bandwidth manually we can “force” it to have as many clusters as we want (by lowering the value of bandwidth), but this of course defeats the point of this algorithm. The bandwidth parameter is also very sensitive as lowering it too much means that many of the further points become outliers and the number of clusters increases unreasonably.

Unlike k-means, the clusters are more or less clearly defined for most variables when plotted against each other. The algorithm does suffer from lots of noisy points that make defining the border between clusters difficult. The plots are presented in the appendix.

5.4. Gaussian Mix

Running the gaussian mix algorithm requires the number of clusters to be specified. If we specify 4 clusters, we do not get a significant fourth cluster. This means that the algorithm is finding 3 real clusters. Gaussian Mix is better than k-means and has more defined borders than Mean Shift but one of the clusters is significantly larger than the other two. Mean shift gives more equally sized clusters.

5.5. DBSCAN

While DBSCAN does not require the number of clusters to be set before running, it does require setting two parameters: the radius of the selection circle and the number of points in the aforementioned circle for that point to be used to select new points for the cluster. As the data in this set are very close together, DBSCAN will not work very well and the results confirm this. With most values for the parameters for this dataset, the algorithm will return one cluster and if the parameters are changed such that a second or third cluster is produced they are very small and are not of any significant size. With more clusters, the algorithm breaks down and just returns lots of noisy points while maintaining the larger central cluster.

5.6. Kmodes

Using k-modes is a good alternative to k-means for categorical data but only when the categorical data contains lots of different values. In this case, we only have two or four unique values for each attribute. Because of this as well as needing to pick the number of clusters does not make k-modes a very useful algorithm for this dataset. Even using two clusters is worse than the groupby tables used previously.

6. Outliers classification

The outliers were finally classified using a k-nearest neighbor algorithm (with neighbors equal to 7), in order to allow their inclusion in marketing campaigns or other directed business strategies.

	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age	Labels
13	1,043.00	-75.12	44.34	342.85	127.69	267.94	94.46	35.00	1
44	1,065.00	-128.68	111.80	-35.00	208.26	224.71	44.23	33.00	0
51	3,234.00	-14,714.08	557.44	20.00	29.56	5.00	-9.00	36.00	1
108	764.00	71.24	79.68	912.95	97.24	213.04	16.56	27.00	2
112	2,354.00	-8,719.04	518.32	4.45	55.90	3.89	10.89	21.00	1
135	2,176.00	-10,198.91	297.61	162.80	143.36	136.47	-3.00	21.00	1
149	984.00	255.71	64.90	197.25	29.56	18.56	451.53	29.00	1
171	1,086.00	-165,680.42	378.07	78.90	166.81	6.89	18.45	28.00	1
179	1,739.00	284.07	99.02	263.95	158.03	45.12	209.04	34.00	1
186	1,247.00	-128.24	33.23	1,026.30	82.57	75.68	212.15	25.00	1
191	1,486.00	872.07	106.02	587.90	76.46	22.45	225.60	22.00	1
257	380.00	1,105.42	50.90	1,012.40	221.93	10.78	128.80	27.00	2
301	1,497.00	1,356.71	41.23	1,089.10	69.68	101.24	150.14	42.00	2
316	1,569.00	14.23	92.35	1,025.75	195.26	54.12	53.12	38.00	0
339	1,288.00	455.43	11.67	245.05	183.70	223.71	104.91	39.00	1

Figure 16 - Outliers after classifying with 7 nearest neighbors

Summary

1. Outliers were removed using IQR and needed specific treatment for some features.
2. The AGE feature revealed not to be trustworthy. The team decided not to use it in the subsequent analysis, preferring the correlated feature SALARY.
3. The HAS CHILDREN categorical feature was the only where using k-nearest neighbor algorithm resulted in accurate classification.
4. Numerical features regression was tested but didn't result in good accuracy. A genetic algorithm was then used to best choose the parameters for a decision tree regressor that estimated the NaNs in the numerical columns.
5. Some pairs of features revealed to have high correlation values, what can be explored to improve the customer portfolio: `Claims Rate` and `Customer Monetary Value` (inversely), `Premiums in LOB: Motor` and all other insurance premium totals (inversely), `Premiums in LOB: Household`, `Premiums in LOB: Life` and `Premiums in LOB: Work Compensations`, Household insurance and the sum of the premiums.
6. The quartiles showed opportunities to increase insurance consumption in some areas and an unassessed potential on clients with high income.
7. Boxplots revealed that the highest insurance premiums in Health and Life come from people with no children and lower education levels. It seems that clients with high insurance buying potential are not being accessed for events that can cause major changes in their lives.
8. Using a decision tree it was possible to confirm that the higher premium sums are concentrated on people with lower educational levels and that geography is not a relevant variable for customer potential measurement.
9. Using a group-by table, the effects of categorical data on the `Customer Monetary Value` can be calculated. Using the count and mean as the aggregation, the number of customers and the value of these customers in each group is easily seen. The mean shows that people with less education and no children have a higher value to the company.
10. Using a dendrogram and the K-Means algorithm, the number of clusters and then the clusters themselves can be calculated. Looking at the graphs produced by the K-Means as well as the other clustering techniques shows does not reveal any particularly useful information. If a group-by table is used on the results of these clustering algorithms, the information can be more easily analyzed.

11. For the group-by tables, the dendrogram and Mean Shift algorithm were selected with the clusters aggregated with the mean average. This is because the dendrogram gave four equally large clusters and the Mean Shift algorithm gave the cleanest looking clusters when plotting. Figure 17 shows that the clusters are not as well defined at Figure 18. This is because the averages of each cluster are more like each other. Using the Mean Shift algorithm gives an average for each feature that is significantly larger for one cluster than the other clusters. This allows us to clearly define what each cluster represents.

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	3,057.94	228.44	238.10	248.88	203.09	50.84	47.56	34.55
1	2,549.88	182.82	368.54	98.82	161.26	19.63	19.28	27.47
2	1,364.08	266.28	130.04	458.81	187.72	88.48	90.96	30.72
3	2,416.87	217.92	447.46	61.81	105.35	11.62	11.37	38.43

Figure 17 - Dendrogram Group-By Table with Mean Aggregate

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	2,609.18	204.18	328.70	153.50	170.24	29.71	29.32	32.00
1	1,990.55	189.22	105.01	434.70	171.54	103.10	123.12	30.93
2	1,941.77	582.91	113.56	697.15	163.63	120.86	52.66	33.99

Figure 18 – Mean Shift Group-By Table with Mean Aggregate

Appendix A

Introduction

The purpose of this appendix is to explain the genetic algorithm created to improve the parameterization of the DecisionTreeClassifier.

After not getting good results when filling the NaN values with basic machine learning algorithms, the team decided to create a genetic algorithm to tune the decision three algorithm for better results.

Tests were made with all numerical columns that have NaNs, which are:

- `Gross Monthly Salary` (Index 10);
- `Premiums in LOB: Motor` (Index 13);
- `Premiums in LOB: Health` (Index 15);
- `Premiums in LOB:  Life` (Index 16);
- `Premiums in LOB: Work Compensations` (Index 17);
- `Age` (Index 18);
- `First Policy´s Age` (Index 19).

Rules

- Each solution was represented by an array with length 5 in which:
    - The first element represents the criterion parameter;
        - 0 == ‘mse’;
        - 1 == ‘friedman_mse';
        - 2 == ‘mae’.
    - The second element represents the min_sample_split parameter;
    - The third element represents the min_samples_leaf parameter;
    - The fourth element represents the max_features parameter;
    - The fifth element represents the max_depth parameter.
    - Example: [1, 23, 10, 9, 8] = [criterion=’friedman_mse’, min_sample_split=23, min_samples_leaf=10, max_features=9, max_depth=8]
- The algorithm was run 10 times for each column;
- Only regressions (solutions) with R-Squared above 0.60 were used in the columns;

- To measure the fitness of each solution the dataset was split into train and test data (65% and 35%, respectively), and after applying the regressor the R-Squared was measured using predictions versus true values from the test set;
- The population size was set 30 and the number of generations for each run was set to 100;
- The mutation probability and the crossover probability were set to 0.5 and 0.8, respectively;
- Single-point crossover, single-point mutation and roulette wheel selection were used as crossover, mutation and selection algorithms, respectively;
- The following constraints were set:

```
DT_constraints = {
    "min_sample_split" : [0,301],
    "min_samples_leaf"“:[0,301],
    "max_features": [0,10],
    "max_depth" : [0,30]
}
```

- The code for the GA algorithm is available on GitHub.

Tests

Column: `Gross Monthly Salary` (index 10)

Column index 10 - Gross Monthly Salary		
ID	REP	FITNESS
TEST 1	[0, 243, 32, 9, 23]	0,477276686
TEST 2	[1, 6, 115, 9, 11]	0,477709166
TEST 3	[0, 230, 84, 7, 26]	0,473805739
TEST 4	[0, 219, 106, 9, 28]	0,478128313
TEST 5	[0, 246, 55, 7, 20]	0,475999419
TEST 6	[1, 60, 133, 9, 16]	0,478825235
TEST 7	[1, 17, 177, 7, 6]	0,473022784
TEST 8	[0, 91, 126, 9, 16]	0,478801144
TEST 9	[1, 99, 60, 9, 27]	0,478328645
TEST 10	[1, 218, 49, 8, 15]	0,475193149

The best solution occurred in Test 6 - [1, 60, 133, 9, 16] - fitness: 0.4788 – R^2 = 47.88%

[criterion= ‘friedman_mse’, min_sample_split=60, min_samples_leaf=133, max_features=9, max_depth=16]

Since the best solution has the fitness lower than 0.6, it has been **discarded** and DecisionTree was not used to regress the null values in the column.

Column: `Premiums in LOB: Motor` (index 13)

Column index 13 - Premiums in LOB: Motor		
ID	**REP**	**FITNESS**
TEST 1	[1, 81, 7, 9, 26]	0,531480899
TEST 2	[0, 10, 9, 7, 9]	0,476222785
TEST 3	[0, 142, 7, 8, 20]	0,523394465
TEST 4	[1, 256, 7, 9, 23]	0,518537997
TEST 5	[1, 122, 1, 3, 28]	0,815182243
TEST 6	[1, 156, 22, 9, 23]	0,289806812
TEST 7	[1, 144, 20, 9, 9]	0,293341778
TEST 8	[1, 183, 7, 8, 19]	0,522199203
TEST 9	[0, 87, 44, 8, 17]	0,271270284
TEST 10	[1, 220, 4, 8, 12]	0,515576511

The best solution occurred in Test 5 - [1, 122, 1,3, 28] - fitness: 0.8157 – R^2 = 81.57%

[criterion= 'friedman_mse', min_sample_split=81, min_samples_leaf=7, max_features=9, max_depth=26]

Since the best solution has a fitness higher than 0.6, **this solution was used** for regressing null values in this column.

Column: `Premiums in LOB: Health` (index 15)

Column index 15 - Premiums in LOB: Health		
ID	**REP**	**FITNESS**
TEST 1	[0, 46, 13, 9, 25]	0,056393847
TEST 2	[0, 31, 7, 4, 15]	0,094696699
TEST 3	[1, 6, 8, 9, 20]	0,086093738
TEST 4	[1, 121, 56, 5, 12]	0,018144267
TEST 5	[0, 126, 12, 7, 17]	0,057738533
TEST 6	[0, 42, 5, 9, 14]	0,131639069
TEST 7	[1, 164, 21, 4, 8]	0,037145065
TEST 8	[0, 19, 29, 6, 14]	0,016580034
TEST 9	[0, 248, 22, 5, 19]	0,035134211
TEST 10	[1, 191, 4, 4, 15]	0,155269953

The best solution occurred in Test 6 - [0, 42, 5, 9, 14] - fitness: 0.1316 – R^2 = 13.16%

[criterion= ‘mse’, min_sample_split=42, min_samples_leaf=5, max_features=9, max_depth=14]

Since the best solution has the fitness lower than 0.6, it has been **discarded** and DecisionTree was not used to regress the null values in the column.

Column: `Premiums in LOB: Life` (index 16)

Column 16 - Premiums in LOB: Life		
ID	REPRESENTATION	FITNESS
TEST 1	[1, 46 ,10, 9, 28]	0.6201194499604908
TEST 2	[0, 82, 32, 9, 26]	0.5921531472303051
TEST 3	[0, 79, 67, 9, 11]	0.5645884754229019
TEST 4	[1, 2, 3, 9, 9]	0.6286779273936715
TEST 5	[1, 29, 24, 9, 13]	0.5845829375832305
TEST 6	[0, 44, 56, 9, 20]	0.5722083169781447
TEST 7	[1, 5, 21, 8, 19]	0.5766313476368804
TEST 8	[0, 121, 26, 9, 23]	0.5768651510799641
TEST 9	[1, 41, 15, 9, 22]	0.6079690901162504
TEST 10	[1, 14, 12, 9, 9]	0.6020106651679107

The best solution occurred in Test 4 - [1, 2, 3, 9, 9] - fitness: 0.6286 – R^2 = 62.86%

[criterion= ‘friedman_mse’, min_sample_split=2, min_samples_leaf=3, max_features=9, max_depth=9]

Since the best solution has a fitness higher than 0.6, **this solution was used** for regressing null values in this column.

Column: `Premiums in LOB: Work Compensations` (index 17)

Column index 17 - Premiums in LOB: Work Compensations		
ID	REP	FITNESS
TEST 1	[0, 5, 96, 8, 9]	0,331477986
TEST 2	[0, 33, 14, 9, 7]	0,405035868
TEST 3	[1, 74, 2, 5, 28]	0,408036522
TEST 4	[1, 88, 32, 9, 17]	0,37940321
TEST 5	[1, 9, 5, 8, 17]	0,394545105
TEST 6	[0, 68, 6, 8, 18]	0,384649996
TEST 7	[1, 86, 53, 9, 22]	0,377327598
TEST 8	[0, 146, 25, 9, 10]	0,368992978
TEST 9	[1, 52, 8, 7, 25]	0,408972296
TEST 10	[0, 53, 74, 9, 27]	0,353070222

The best solution occurred in Test 9 - [1, 52, 8, 7, 25] - fitness: 0.4089 – R^2 = 40.89%

[criterion= ‘friedman_mse’, min_sample_split=52, min_samples_leaf=8, max_features=7, max_depth=25]

Since the best solution has the fitness lower than 0.6, it has been **discarded** and DecisionTree was not used to regress the null values in the column.

Column: `Age` (index 18)

Column index 18 – Age		
ID	REP	FITNESS
TEST 1	[0, 8, 36, 9, 13]	0,881717191
TEST 2	[1, 189, 76, 8, 29]	0,880112799
TEST 3	[1, 236, 87, 8, 16]	0,88034088
TEST 4	[1, 5, 66, 8, 7]	0,881111537
TEST 5	[0, 266, 11, 8, 27]	0,876731747
TEST 6	[1, 242, 34, 8, 12]	0,880314792
TEST 7	[0, 210, 120, 8, 17]	0,875409194
TEST 8	[1, 77, 113, 5, 20]	0,865051458
TEST 9	[0, 143, 94, 8, 29]	0,879270313
TEST 10	[1, 58, 120, 8, 29]	0,875409194

The best solution occurred in Test 1 - [0, 8, 36, 9, 13] - fitness: 0.8817 – R^2 = 88.17%

[criterion= ‘mse’, min_sample_split=8, min_samples_leaf=36, max_features=9, max_depth=13]

Since the best solution has a fitness higher than 0.6, **this solution was used** for regressing null values in this column. **Following the reasons presented in the report, this column was later discarded for further analysis.**

Column: `First Policy´s Age` (index 19)

Column index 19 – First Policy´s Age		
ID	REP	FITNESS
TEST 1	[0, 22, 5, 1, 1]	-1,12523E-05
TEST 2	[1, 34, 226, 7, 2]	-0,000230829
TEST 3	[0, 254, 190, 1, 20]	-0,004908259
TEST 4	[1, 290, 162, 5, 1]	-0,000407931
TEST 5	[1, 146, 178, 5, 1]	0,000198636
TEST 6	[1, 106, 38, 2, 1]	0,000625661
TEST 7	[0, 198, 56, 2, 1]	0,000625661
TEST 8	[1, 225, 150, 4, 4]	-0,000737923
TEST 9	[1, 225, 251, 3, 2]	-0,000218549
TEST 10	[1, 99, 52, 1, 1]	-1,12523E-05

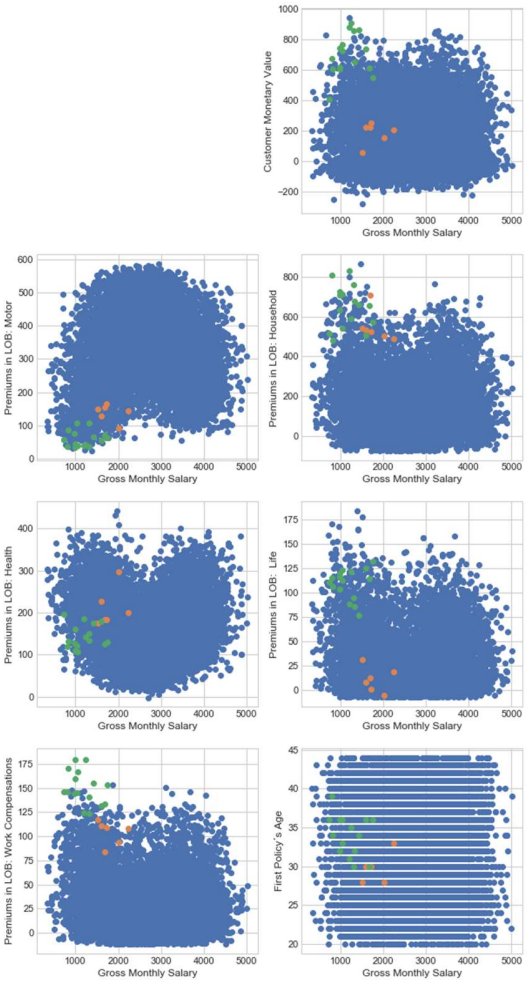
The best solution occurred in Test 6 - [1, 106, 38, 2, 1] - fitness: 0.0006 – R^2 = 00.06%

[criterion= ‘friedman_mse’, min_sample_split=106, min_samples_leaf=38, max_features=2, max_depth=1]

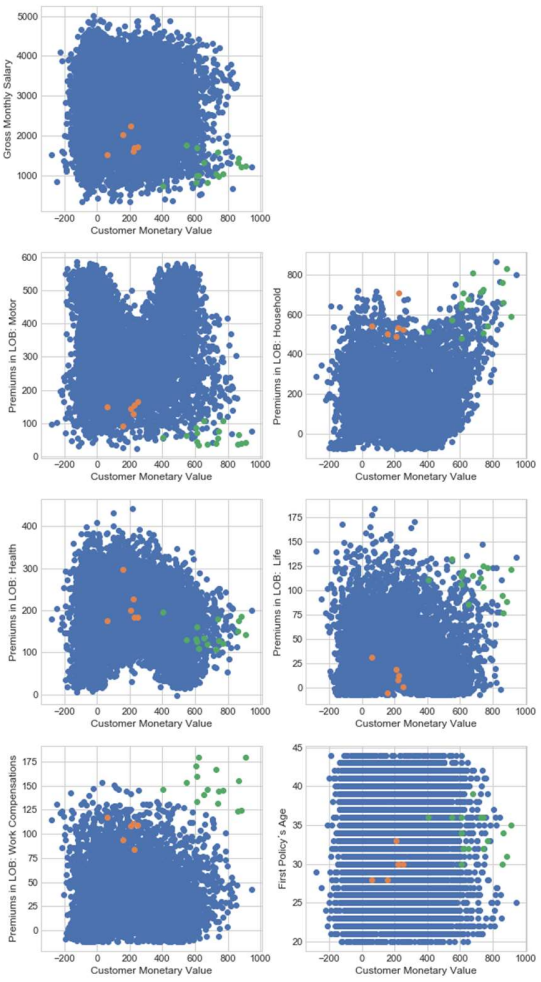
Since the best solution has the fitness lower than 0.6, it has been **discarded** and DecisionTree was not used to regress the null values in the column.

Appendix B

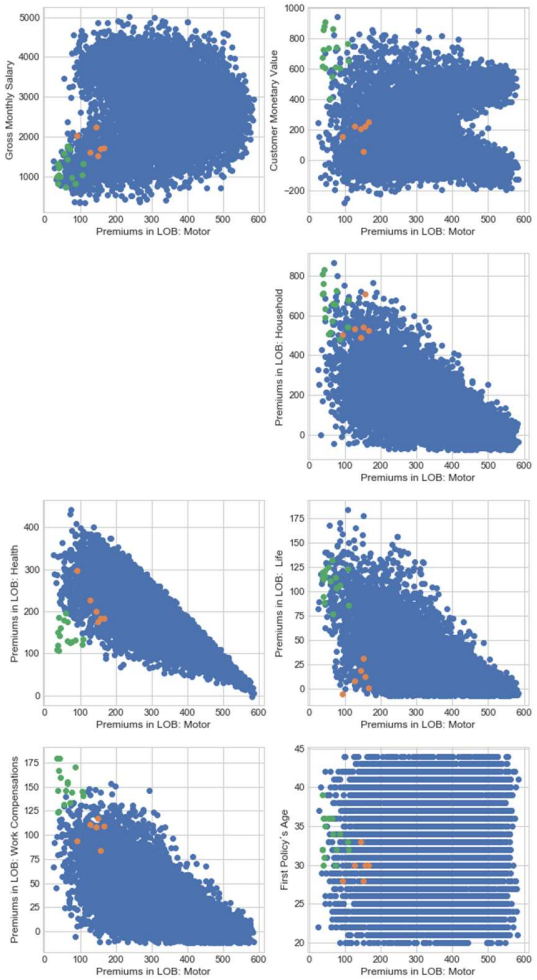
DBSCAN with Gross Monthly Salary as X-Axis



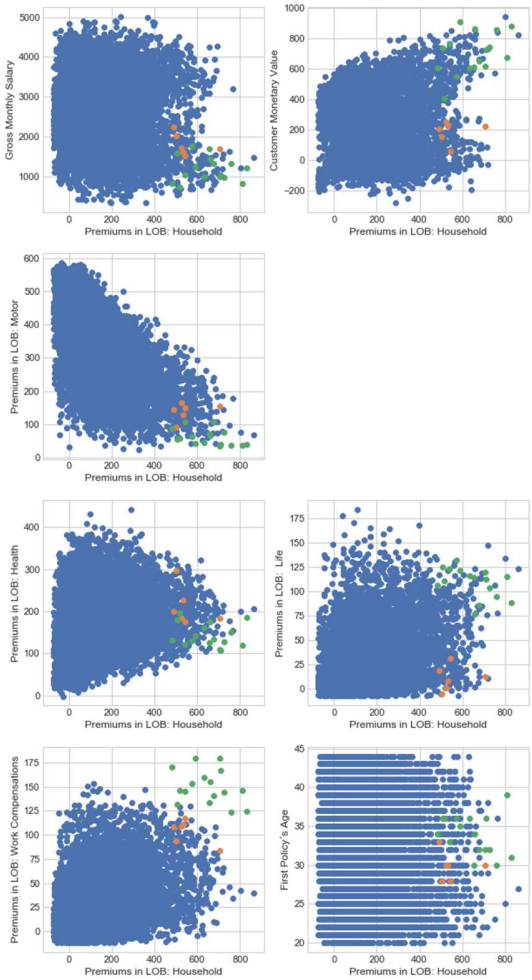
DBSCAN with Customer Monetary Value as X-Axis



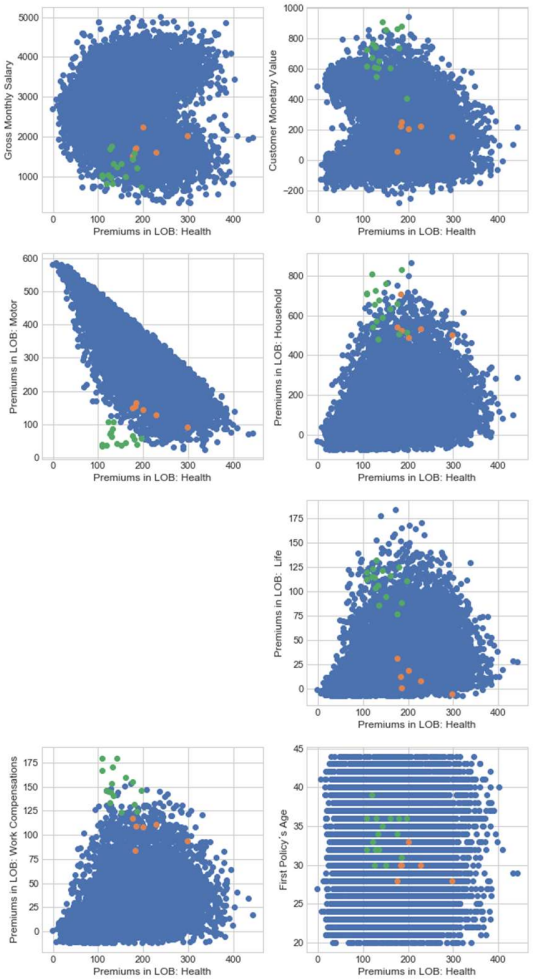
DBSCAN with Premiums in LOB: Motor as X-Axis



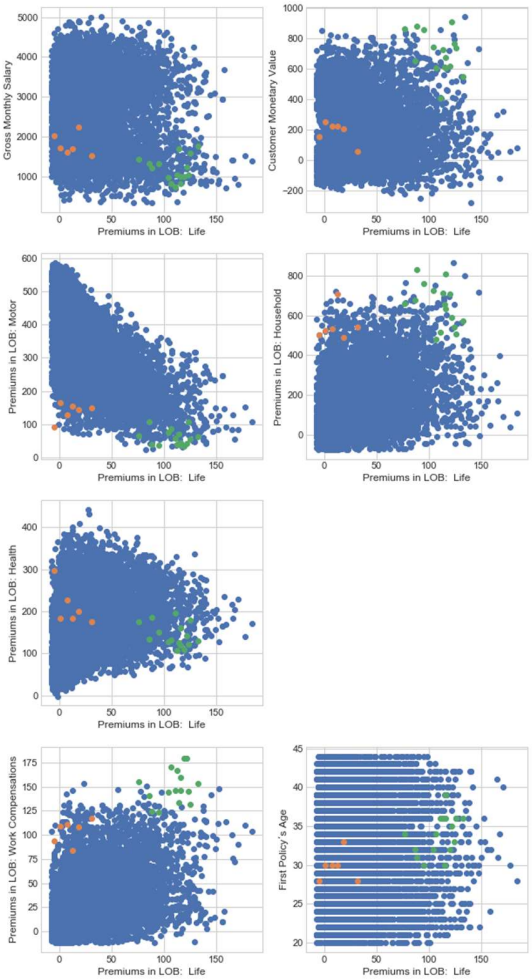
DBSCAN with Premiums in LOB: Household as X-Axis



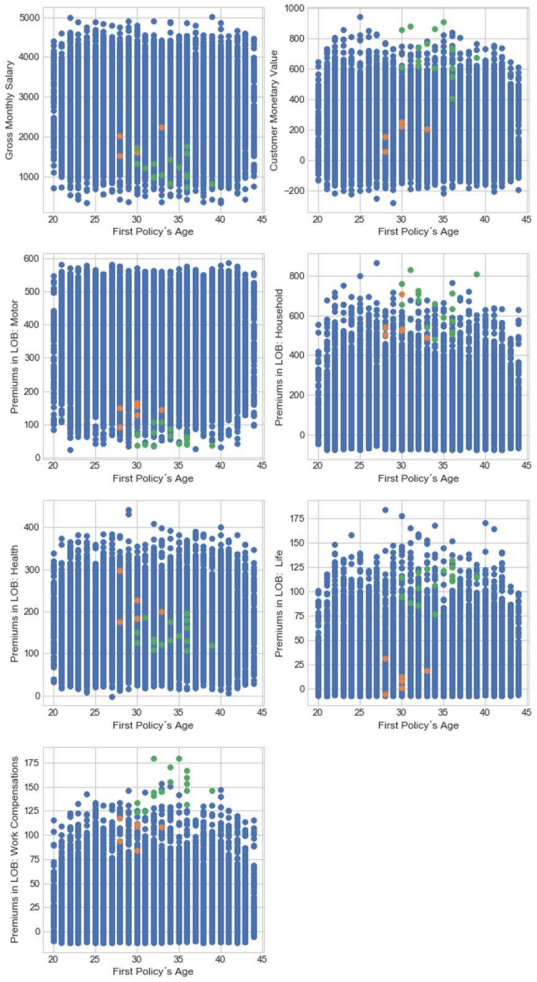
DBSCAN with Premiums in LOB: Health as X-Axis



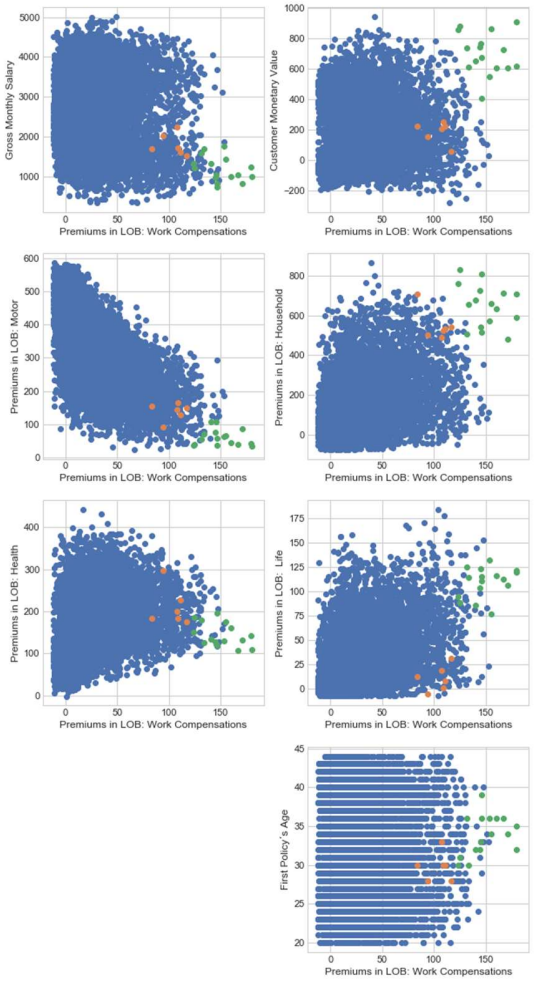
DBSCAN with Premiums in LOB: Life as X-Axis



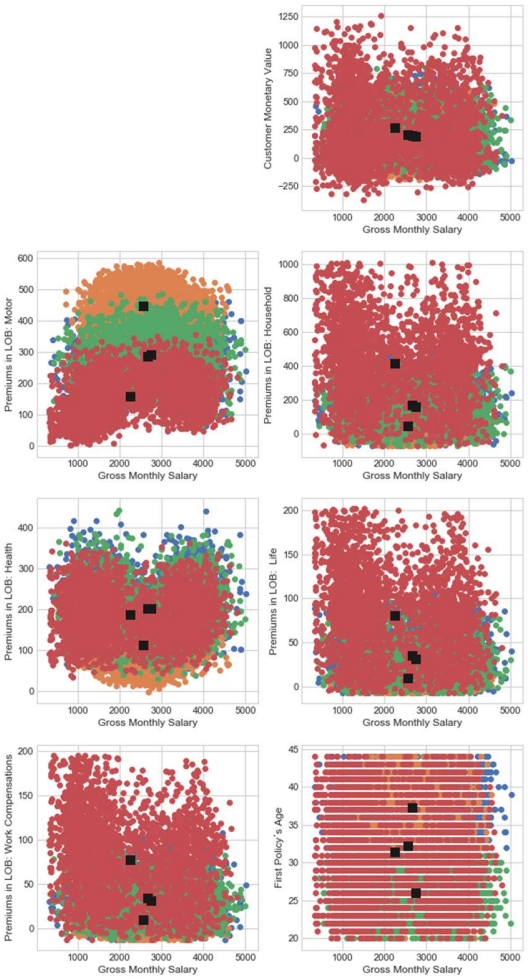
DBSCAN with First Policy´s Age as X-Axis



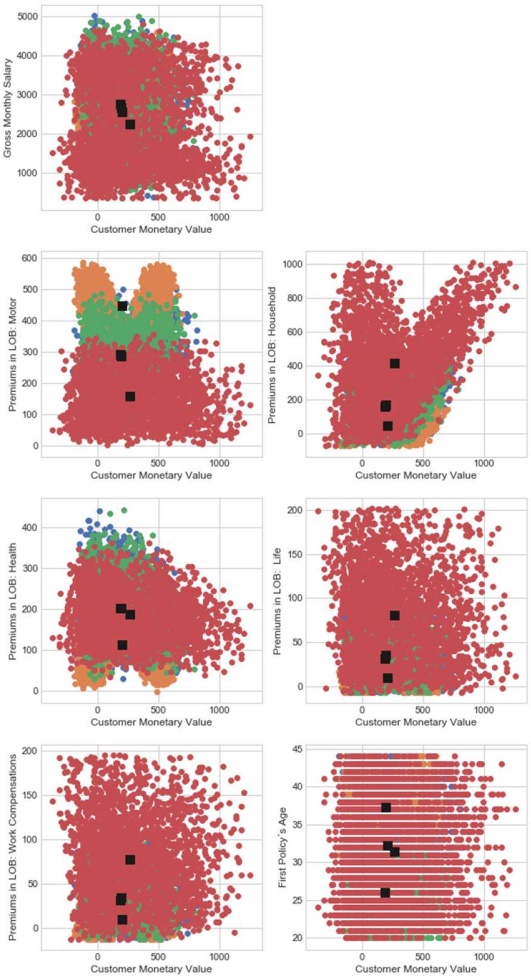
DBSCAN with Premiums in LOB: Work Compensations as X-Axis



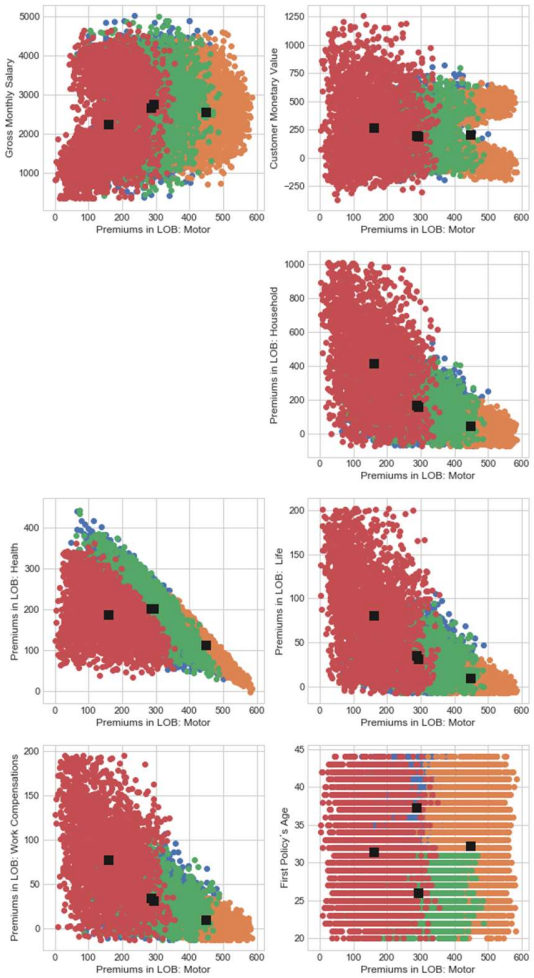
Gaussian Mixture with Premiums in LOB: Household as X-Axis



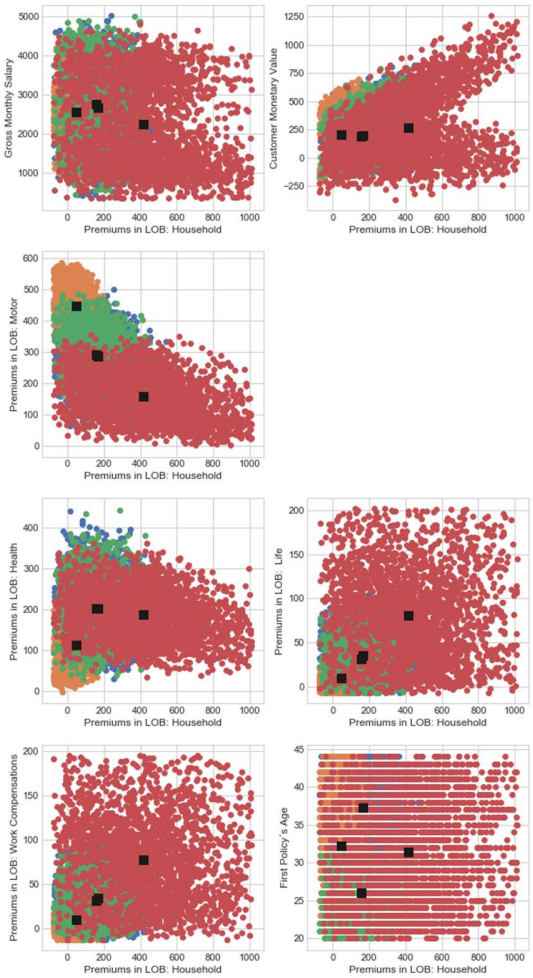
Gaussian Mixture with Gross Monthly Salary as X-Axis



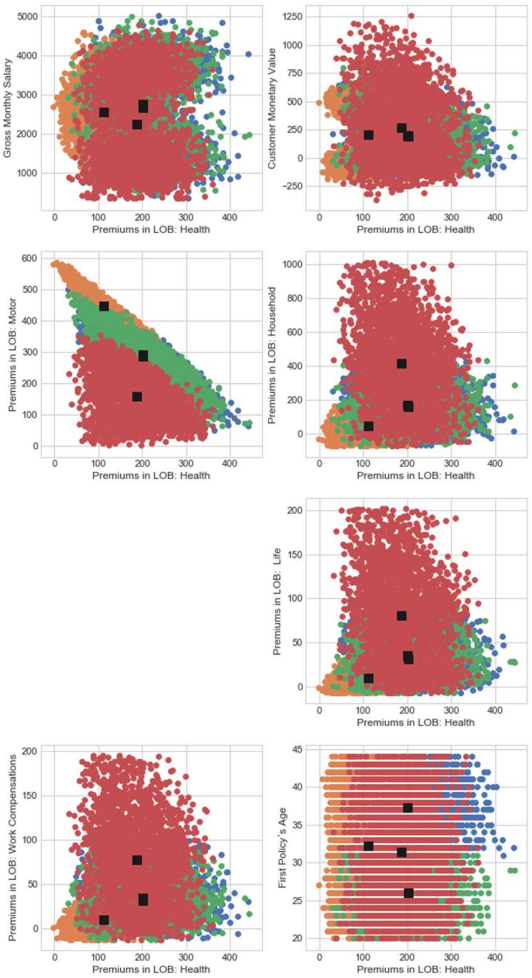
Gaussian Mixture with Customer Monetary Value as X-Axis



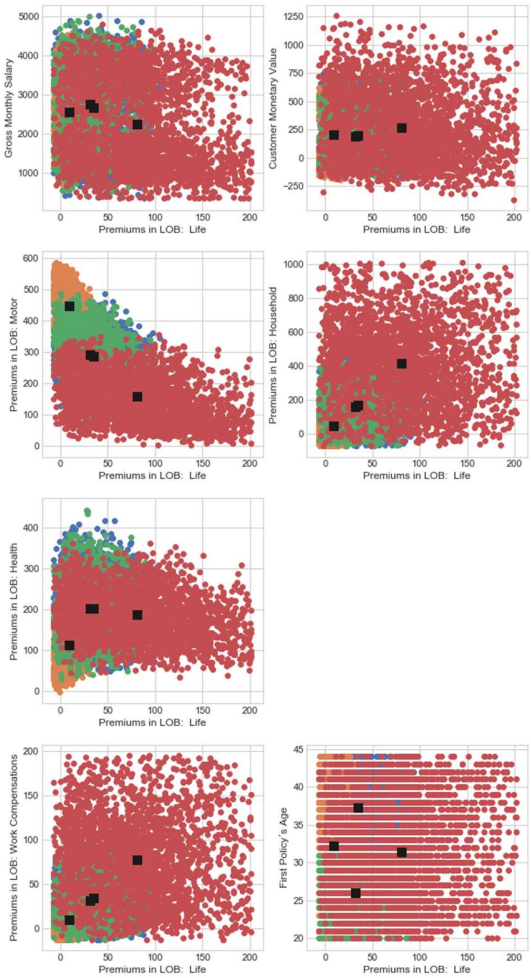
Gaussian Mixture with Premiums in LOB: Motor as X-Axis



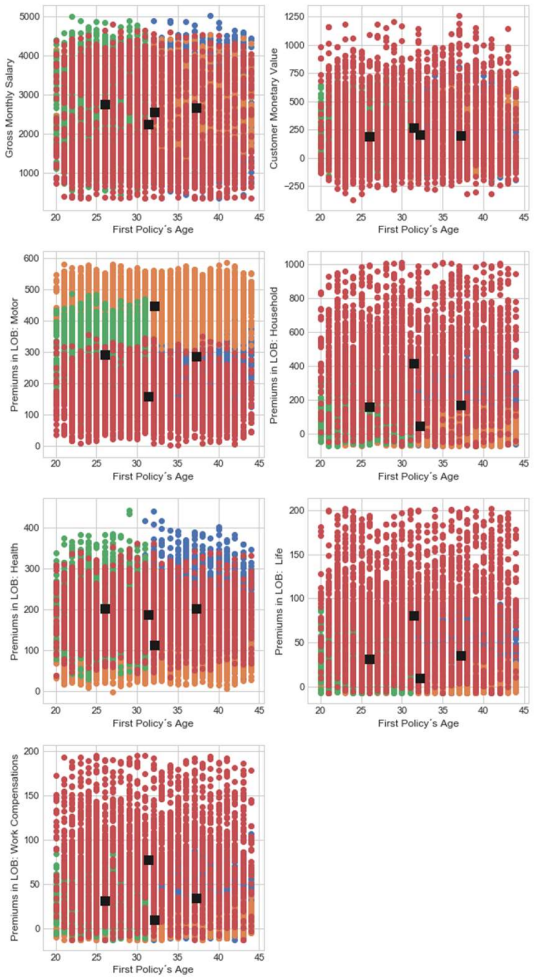
Gaussian Mixture with Premiums in LOB: Household as X-Axis



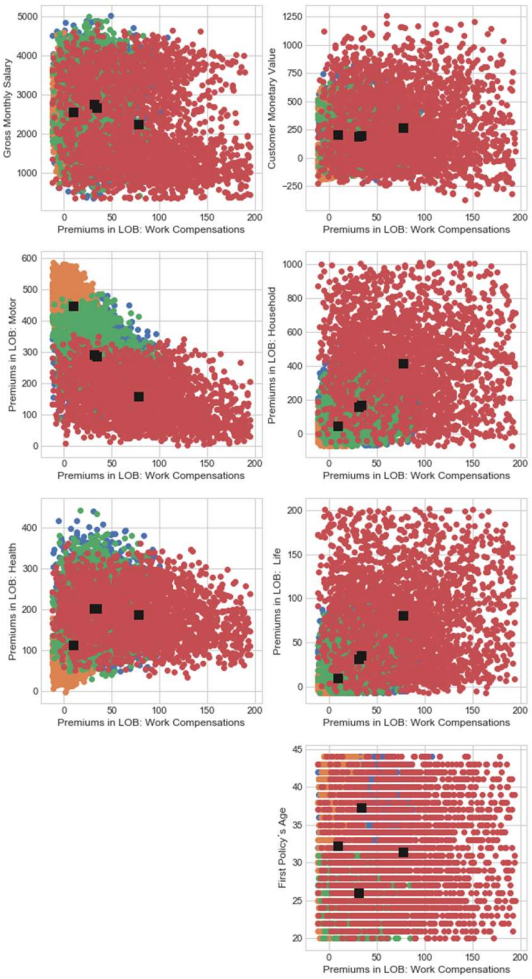
Gaussian Mixture with Premiums in LOB: Health as X-Axis



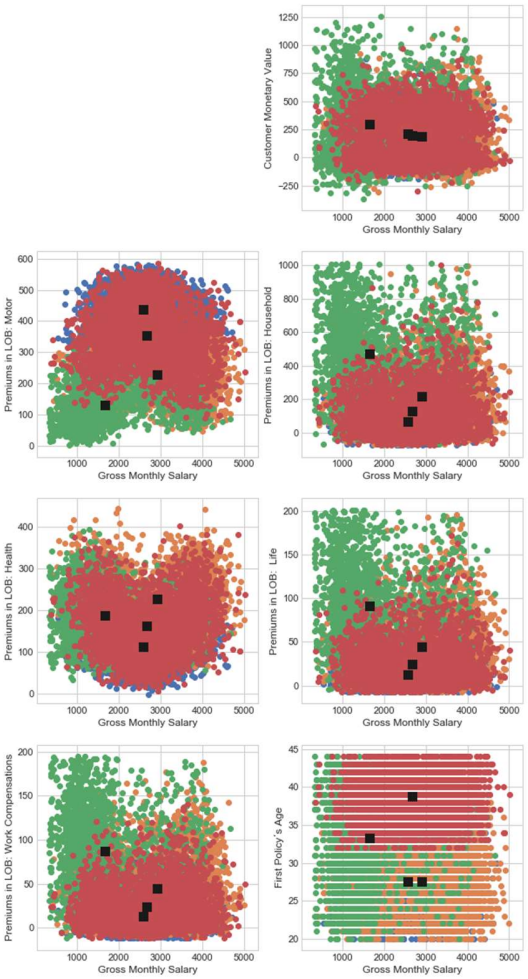
Gaussian Mixture with Premiums in LOB: Work Compensations as X-Axis



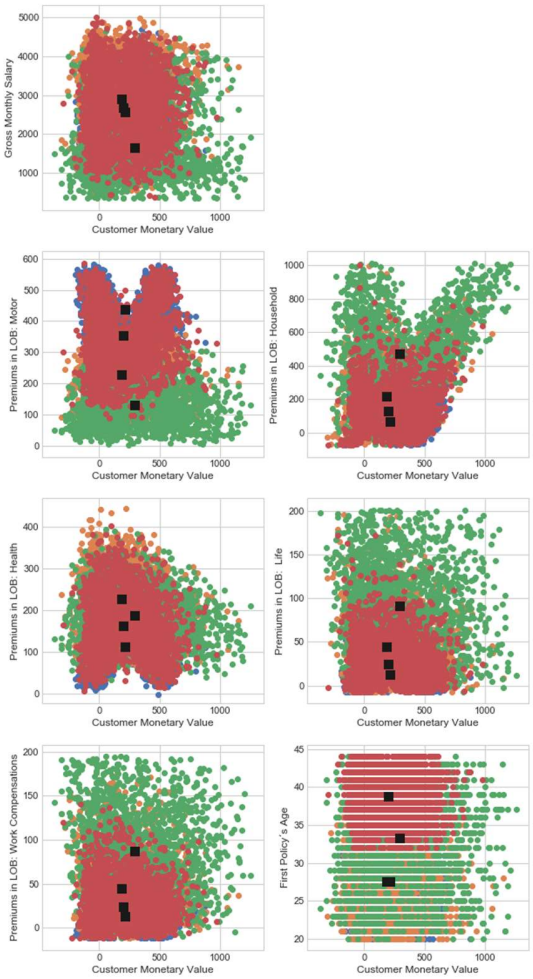
Gaussian Mixture with Premiums in LOB: Life as X-Axis



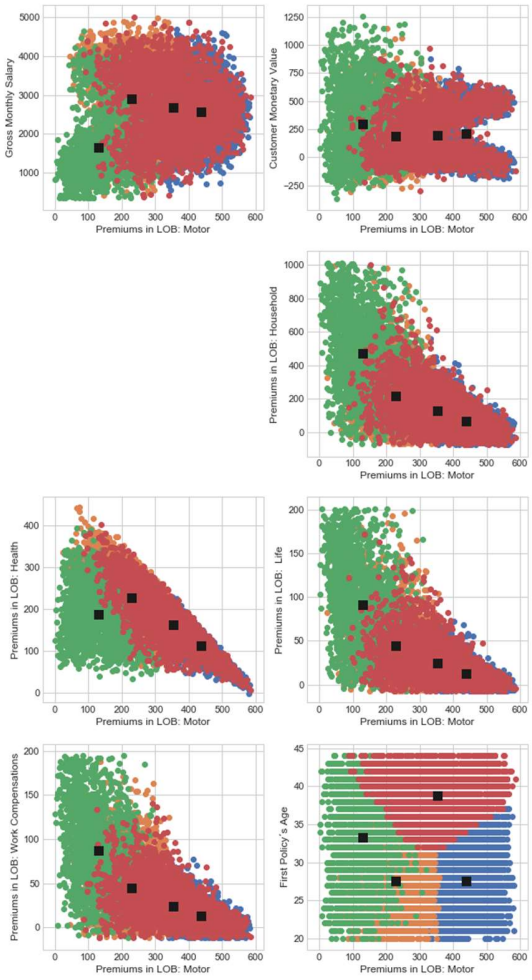
Kmeans with Gross Monthly Salary as X-Axis



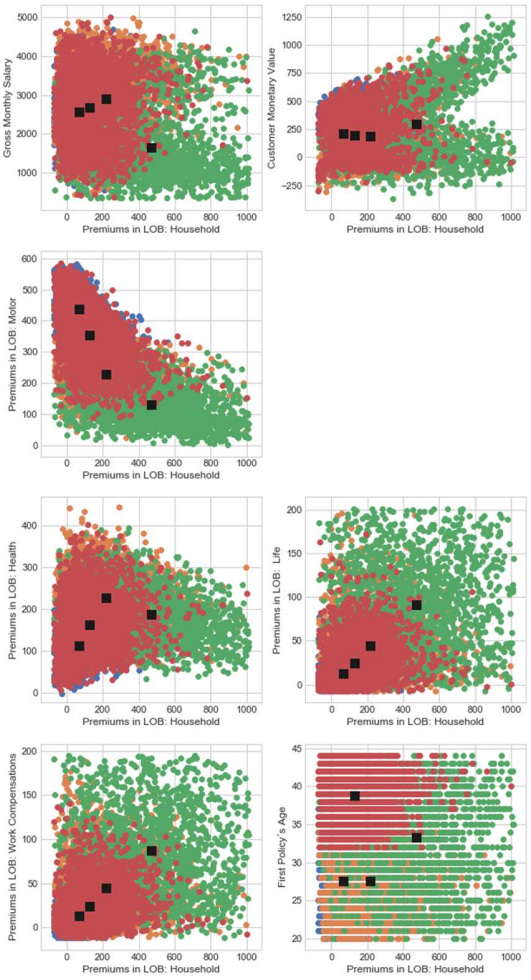
Kmeans with Customer Monetary Value as X-Axis



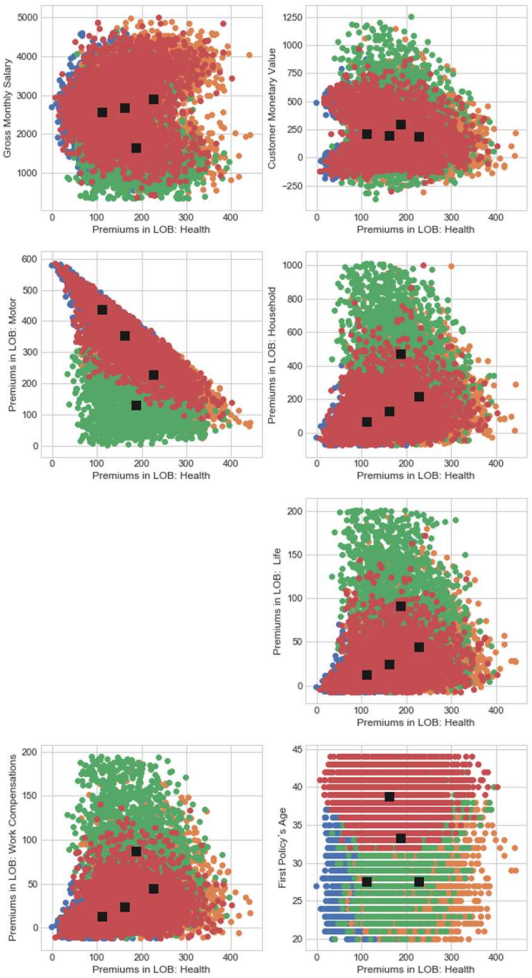
Kmeans with Premiums in LOB: Motor as X-Axis



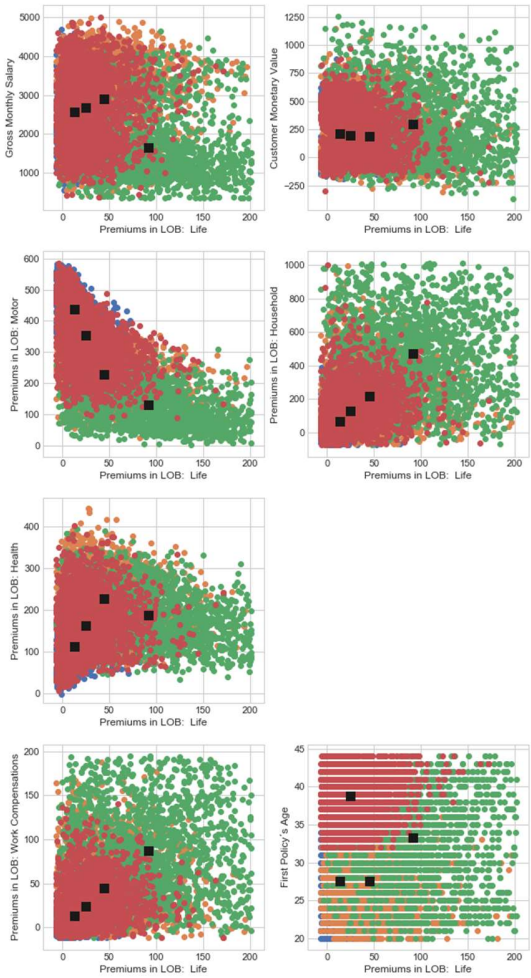
Kmeans with Premiums in LOB: Household as X-Axis



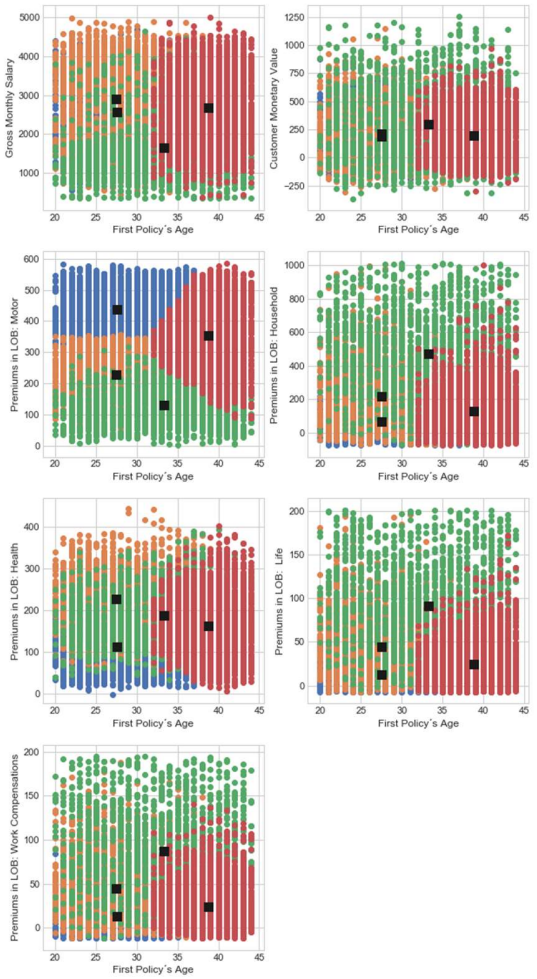
Kmeans with Premiums in LOB: Health as X-Axis



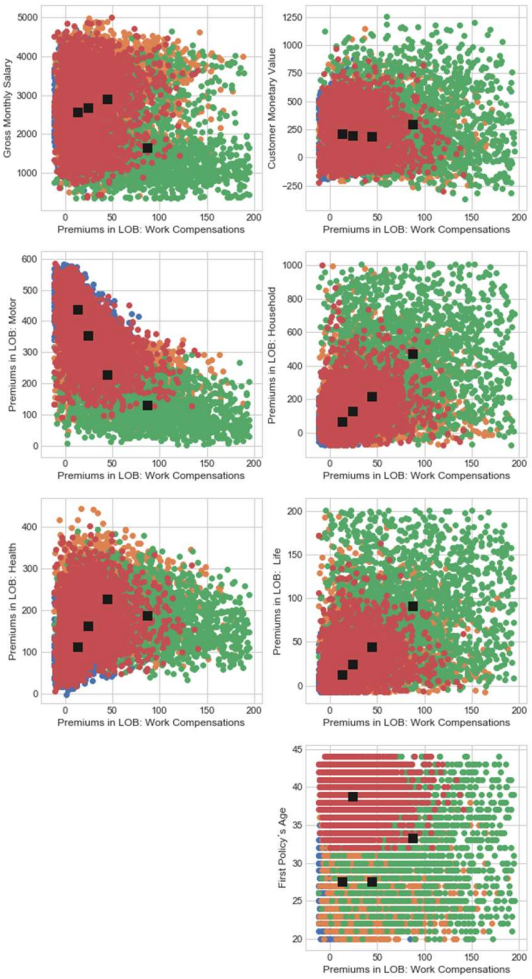
Kmeans with Premiums in LOB: Life as X-Axis



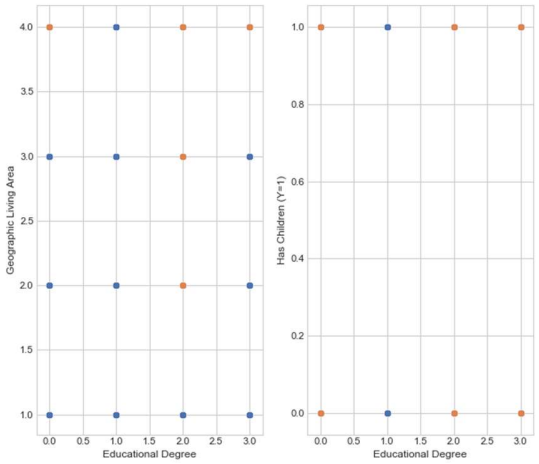
Kmeans with First Policy´s Age as X-Axis



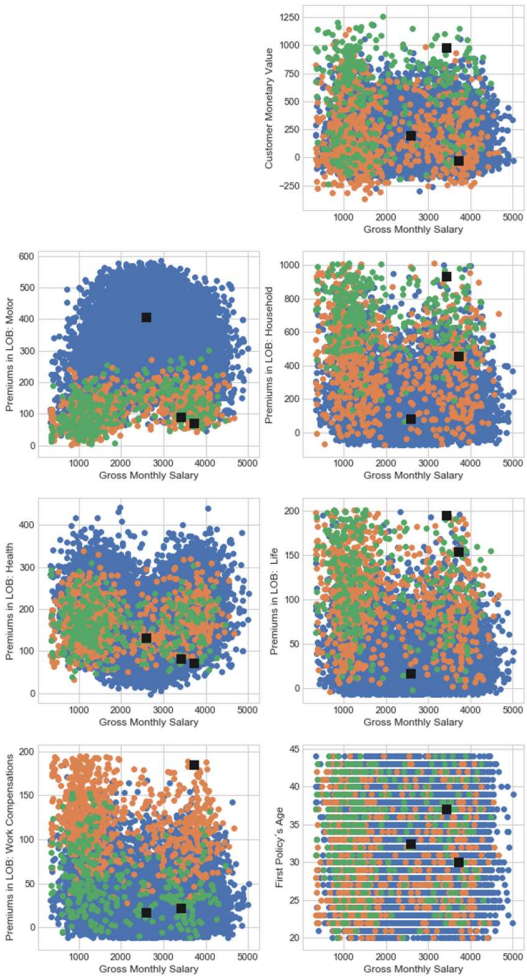
Kmeans with Premiums in LOB: Work Compensations as X-Axis



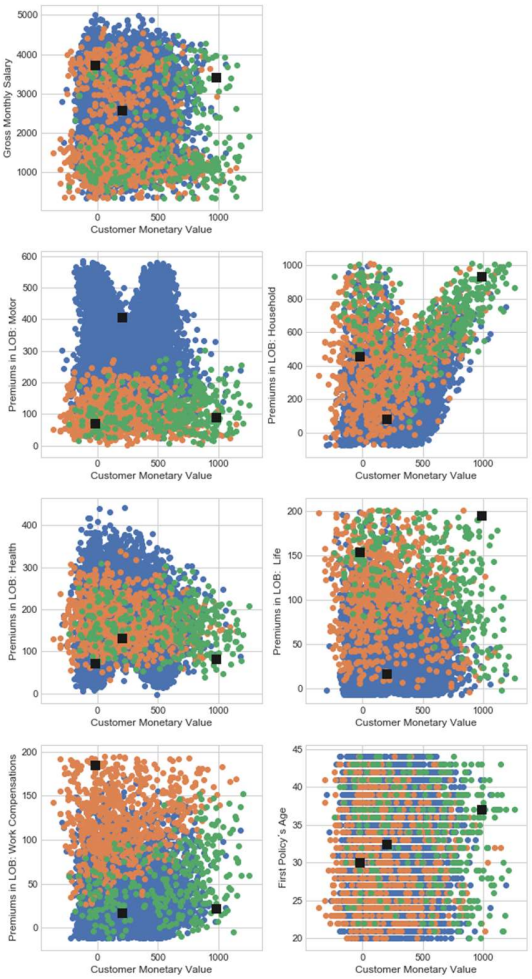
KModes with Educational Degree as X-Axis



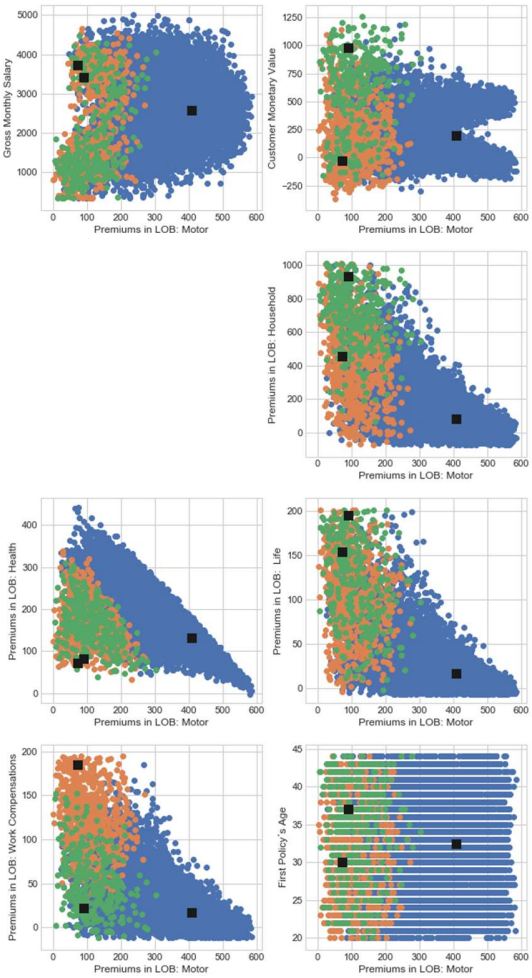
Mean Shift with Gross Monthly Salary as X-Axis



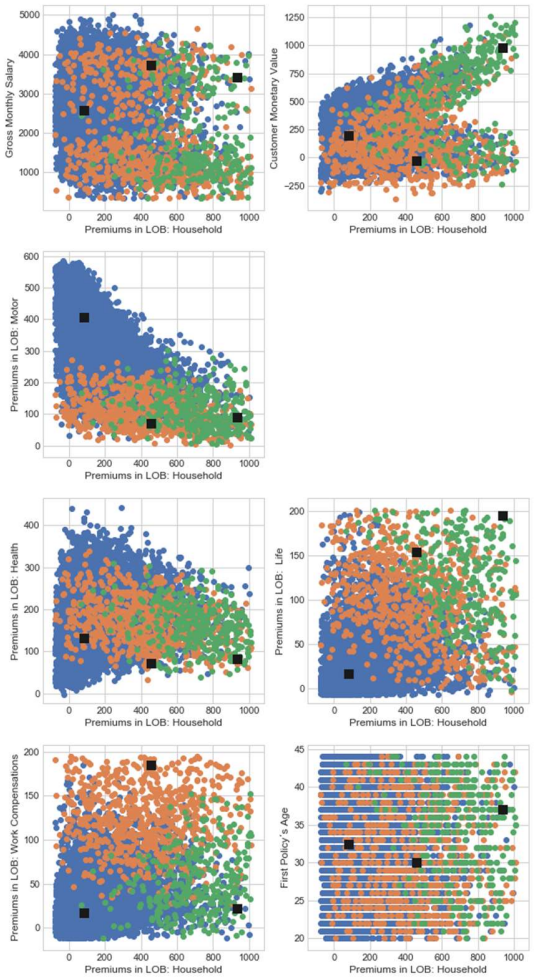
Mean Shift with Customer Monetary Value as X-Axis



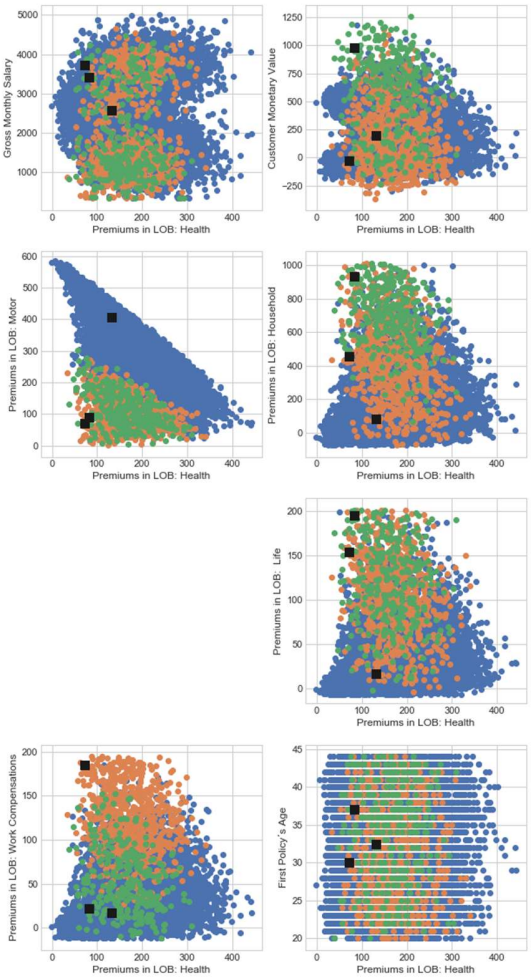
Mean Shift with Premiums in LOB: Motor as X-Axis



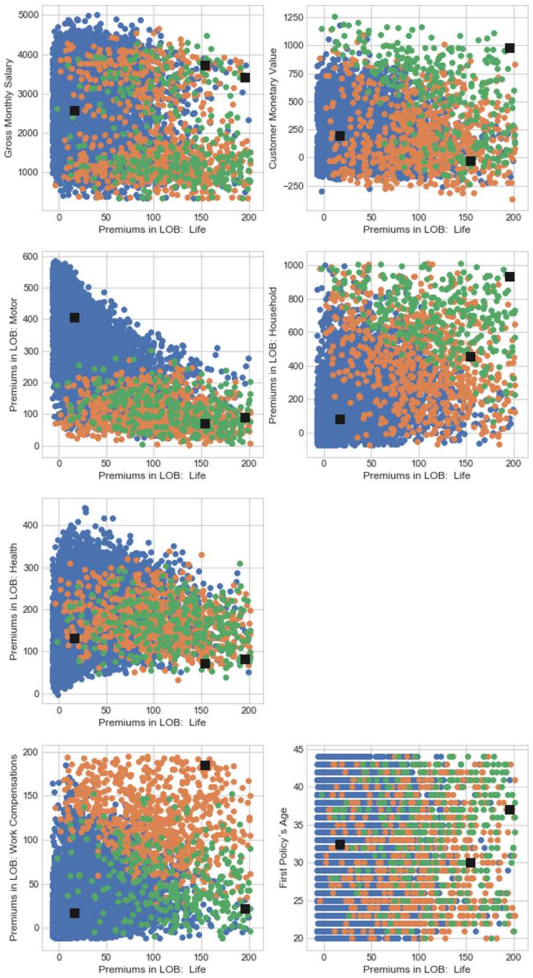
Mean Shift with Premiums in LOB: Household as X-Axis



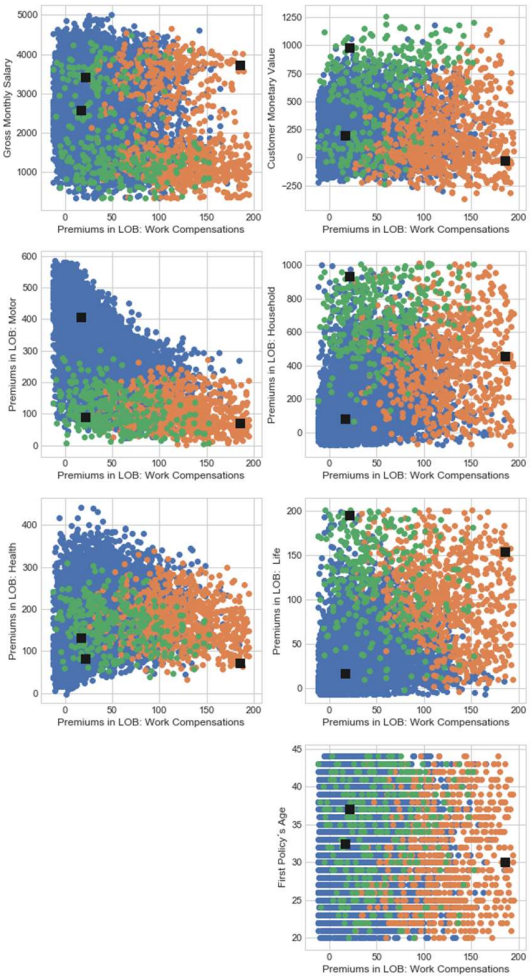
Mean Shift with Premiums in LOB: Health as X-Axis



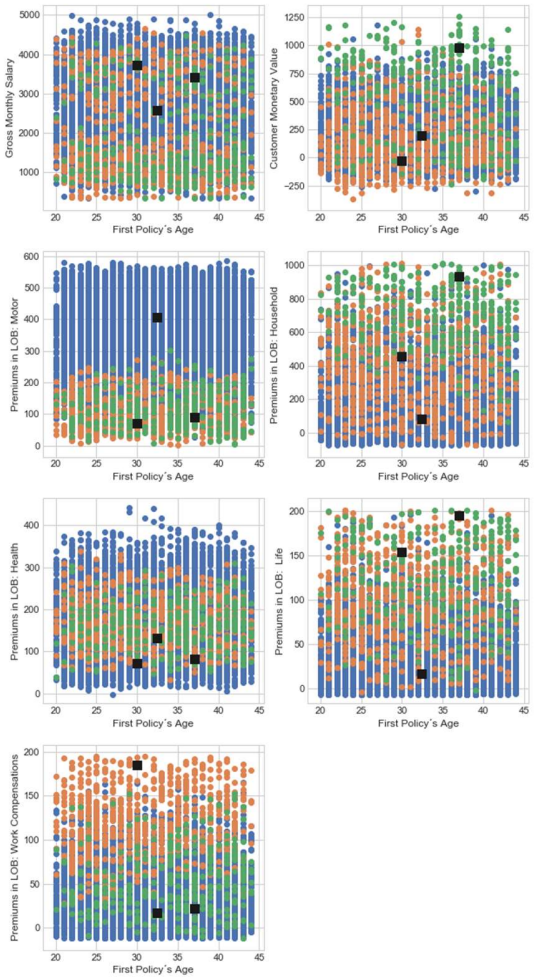
Mean Shift with Premiums in LOB: Life as X-Axis



Mean Shift with Premiums in LOB: Work Compensations as X-Axis



Mean Shift with First Policy´s Age as X-Axis



Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	3239	3239	3239	3239	3239	3239	3239	3239
1	3627	3627	3627	3627	3627	3627	3627	3627
2	1281	1281	1281	1281	1281	1281	1281	1281
3	1499	1499	1499	1499	1499	1499	1499	1499

Figure 19 - Customers count for dendrogram with 4 clusters

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	3,057.94	228.44	238.10	248.88	203.09	50.84	47.56	34.55
1	2,549.88	182.82	368.54	98.82	161.26	19.63	19.28	27.47
2	1,364.08	266.28	130.04	458.81	187.72	88.48	90.96	30.72
3	2,416.87	217.92	447.46	61.81	105.35	11.62	11.37	38.43

Figure 20 - Means for dendrogram with 4 clusters

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	8627	8627	8627	8627	8627	8627	8627	8627
1	723	723	723	723	723	723	723	723
2	296	296	296	296	296	296	296	296

Figure 21 - Customers count for mean shift

Labels	Gross Monthly Salary	Customer Monetary Value	Premiums in LOB: Motor	Premiums in LOB: Household	Premiums in LOB: Health	Premiums in LOB: Life	Premiums in LOB: Work Compensations	First Policy´s Age
0	2,609.18	204.18	328.70	153.50	170.24	29.71	29.32	32.00
1	1,990.55	189.22	105.01	434.70	171.54	103.10	123.12	30.93
2	1,941.77	582.91	113.56	697.15	163.63	120.86	52.66	33.99

Figure 22 - Means for mean shift

Educational Degree	Geographic Living Area	Has Children (Y=1)	
2.00	4.00	1.00	1282
	1.00	1.00	1002
1.00	4.00	1.00	972
2.00	3.00	1.00	689
1.00	1.00	1.00	685
2.00	4.00	0.00	566
1.00	3.00	1.00	471
	4.00	0.00	413
2.00	1.00	0.00	385
	2.00	1.00	326
0.00	4.00	1.00	273
2.00	3.00	0.00	271
1.00	1.00	0.00	267
	2.00	1.00	232
0.00	1.00	1.00	220
3.00	4.00	1.00	181
1.00	3.00	0.00	180
3.00	1.00	1.00	149
0.00	3.00	1.00	139
2.00	2.00	0.00	132
0.00	4.00	0.00	127
1.00	2.00	0.00	101
3.00	3.00	1.00	94
	4.00	0.00	82
0.00	1.00	0.00	74
	2.00	1.00	71
3.00	1.00	0.00	63
0.00	3.00	0.00	60
3.00	2.00	1.00	48
	3.00	0.00	41
0.00	2.00	0.00	32
3.00	2.00	0.00	18

Name: Customer Monetary Value, dtype: int64

Figure 23 - Categorical features grouping (customer count)

Educational Degree	Geographic Living Area	Has Children (Y=1)	
0.00	2.00	0.00	332.48
	3.00	0.00	271.75
	1.00	0.00	268.41
	3.00	1.00	261.13
	4.00	1.00	259.15
		0.00	247.94
3.00	3.00	0.00	243.83
0.00	1.00	1.00	238.81
	2.00	1.00	235.20
3.00	1.00	1.00	228.13
1.00	2.00	1.00	223.84
	1.00	0.00	222.21
	3.00	0.00	220.98
	4.00	1.00	220.89
2.00	3.00	1.00	219.96
	1.00	1.00	218.65
3.00	4.00	1.00	213.58
2.00	1.00	0.00	211.72
1.00	1.00	1.00	209.74
2.00	2.00	0.00	208.34
	4.00	1.00	206.36
1.00	4.00	0.00	203.95
	3.00	1.00	203.51
3.00	2.00	1.00	203.45
1.00	2.00	0.00	197.96
2.00	2.00	1.00	195.13
	4.00	0.00	191.54
	3.00	0.00	189.52
3.00	3.00	1.00	187.27
	1.00	0.00	182.17
	4.00	0.00	181.19
	2.00	0.00	154.07

Name: Customer Monetary Value, dtype: float64

Figure 24 - Categorical features grouping (mean)

Educational Degree	Geographic Living Area	Has Children (Y=1)	
0.00	2.00	0.00	337.07
3.00	1.00	1.00	288.17
	3.00	0.00	246.05
0.00	4.00	1.00	237.50
	1.00	0.00	232.66
2.00	3.00	1.00	229.93
0.00	3.00	0.00	224.50
3.00	4.00	1.00	222.39
2.00	1.00	1.00	221.26
		0.00	216.28
0.00	1.00	1.00	200.99
2.00	3.00	0.00	198.29
0.00	3.00	1.00	196.50
	2.00	1.00	194.93
1.00	3.00	0.00	194.87
2.00	2.00	0.00	192.31
	4.00	1.00	183.20
1.00	4.00	1.00	182.77
	2.00	0.00	182.71
3.00	2.00	1.00	179.15
1.00	1.00	1.00	177.70
		0.00	175.05
	2.00	1.00	173.87
0.00	4.00	0.00	173.26
1.00	3.00	1.00	170.36
2.00	4.00	0.00	167.03
3.00	4.00	0.00	161.75
1.00	4.00	0.00	159.58
3.00	1.00	0.00	151.70
2.00	2.00	1.00	137.37
3.00	3.00	1.00	68.73
	2.00	0.00	20.45

Name: Customer Monetary Value, dtype: float64

Figure 25 - Categorical features grouping (median)