



NOVA

IMS

Information
Management
School

Mestrado em Métodos Analíticos Avançados

Master Program in Advanced Analytics

Academic Year 2019/2020

Data Mining

Final Project

Group AW:

Maria Leonor Furtado M20190308

Giuliana Lopes M20190209

Ernesto Aguilar Madrid M20190559

Professors: Fernando Lucas Bação & Jorge Antunes

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

Universidade Nova de Lisboa

Index

1. Introduction	3
2. Data Mining Process	3
3. Insurance Market	3
4. Problem Description	4
5. Importing, Exploring and Preprocessing Data	5
5.1. Importing Data.....	5
5.2. Exploring and Transforming Data	5
5.3. Performing Feature Engineering to Calculate Additional Attributes	6
5.4. Detecting and Handling Outliers	7
6. Data Visualization	8
7. Input Space Reduction and Categorical / Numerical split.....	10
8. Clustering	12
8.1. Partitional Clustering	12
8.2. Density Based Clustering	17
8.3. Hierarchical Clustering.....	18
8.4. From Clustering to Classification	19
8.5. Clustering Assessment & Validation	20
8.6. Interpreting Clusters.....	21
9. Conclusion	21
10. References.....	21
11. Annex	22

All scripts for this project are available in the following private repository:

https://github.com/leolioness1/uni_projects

If you would like to be added, please send your GitHub username to M20190308@novaims.unl.pt

1. Introduction

Nowadays, it is key to understand customer behavior. This is an important aspect of customer segmentation that allows marketers to better focus their marketing efforts to various audience subsets in terms of promotional, marketing and product development strategies.

In this project we discuss the use of modern data mining (DM) methods to create cluster on the customers' profiles. Our objective is to facilitate the Marketing Department to better understand their customer base. The goal is to demonstrate the use of DM in understanding the customer profile. We demonstrate the ability to discover new underwriting parameters and to distinguish between distinct groups of policies. The study was performed on a data set from a fictional insurance company in Portugal. We demonstrate our ability to discover new underwriting parameters and to distinguish between distinct clusters of policies.

2. Data Mining Process

The process of building and implementing a DM solution is referred to as Knowledge Discovery in Databases [1]. The process contains three main components: pre-processing, modeling and analysis (data mining), and post-processing.

Pre-processing: is often the most time consuming of the process. It consists of several tasks to prepare the data for modeling, including defining the application problem, creating a target data set for the analysis, checking data integrity, cleaning the data, transforming and collapsing the data, among others. Each of the possible is being tested for its quality and the quantity of "knowledge" it contains. New predicting variables are generated from the original set using several transformations that may improve the amount of "knowledge".

Data mining (DM): consists in interrogating the data for patterns and trends for decision-making. DM is an interdisciplinary field involving query tools, regression models, association rules, decision trees, visualization, and others.

Classification and clustering are both fundamental steps in Data Mining. Classification is used mostly as a supervised learning method, clustering, however, for unsupervised learning (some models are for both). The objective of clustering is descriptive, that of classification is predictive (Veyssieres and Plant, 1998).

In data mining the data is mined using two learning approaches i.e. supervised learning or unsupervised clustering. In Supervised Learning training data includes both the input and the desired results. Supervised models are neural network, Multilayer Perceptron, Decision trees.

Unsupervised Learning is not provided with the correct results during the training. It can be used to cluster the input data in classes on the basis of their statistical properties only. Unsupervised models are different types of clustering, distances and normalization, k-means, self-organizing maps.

Post-processing: evaluating and interpreting the modeling results to make sure the models are adequate. Most important is validating the model results to guard against over-fitting. Often, several candidate models are used to study the data, requiring that one analyze the performance of the models and choose the final model for implementation.

3. Insurance Market

The insurance company has statistics on risks linked to all kinds of characteristics, such as gender, age, disabilities and behaviors. Typically, insurers collect every information available. They collect people into groups, or segment them, according to these risks. For each group, an average person and their likeliness (or riskiness) to become sick/have an accident etc. is defined.

Customer profiling is very important in determining premium rates and insurance companies consider several factors when calculating insurance premiums. For instance:

- **Age:** Insurance companies look at your age because that can predict the likelihood that you'll need to use the insurance. With health insurance, younger people are less likely to need medical care, so their premiums are generally cheaper. And teenage drivers are still working on building experience, so they're more expensive to insure. Likewise, older drivers—who tend to have slower reflexes—will also pay more.
- **The type of coverage:** In general, you have several options when you buy an insurance policy. The more comprehensive coverage you get, the more expensive it will be.
- **The amount of coverage:** The less coverage, the cheaper the premiums—no matter what you're insuring.
- **Personal information:** Depending on the type of insurance you're shopping for, the insurance company may take a close look at things like your claims history, driving record, credit history, gender, marital status, lifestyle, family medical history, health, smoking status, hobbies, job, and where you live.

Traditional methods to set up insurance rates are based on segmentation. The objective is to divide the population into “homogenous” segments. Then, the claim level for each customer is given by the average claim values of all policyholders belonging to the same segment. This method assumes that all individuals in the segment are “alike”. The segments are created to yield groups that are not-too-small and not-too-big. Small segments suffer from lack of statistical significance and poor prediction. Large segments may not be homogenous enough for decision-making. Sometimes, segmentation methods were replaced by the analysis of rate relativities, obtained from the analysis of the margins of the tables. Nevertheless, both segmentation and rate relativities methods may yield erroneous results because they usually do not consider interaction terms, which are very prevalent in the insurance industry.

4. Problem Description

A fictional insurance company in Portugal, has contracted our team to develop a consultancy about Customer Segmentation to help Marketing Department to understand all the different **Customers' Profiles**. To achieve this objective, the insurance company provided us a database file and the ABT (Analytic Based Table) with the information regarding 10 290 customers and 14 attributes per customer.

Table 1. Database attributes description.

Attribute	Description	Additional information
ID	Customer ID number	
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 database 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	(Yes = 1, No = 0)
CMV	Customer Monetary Value	CMV = (customer annual profit)(years as customer) - (acquisition cost)
Claims	Claims Rate	Claims Rate: Amount paid by the insurance company (€)/ Premiums (€) Note: in the last 2 years
Motor	Premiums in Motor LOB	Annual Premiums in (€) for year 2016. LOB: Line of Business Note: Negative Premiums may manifest reversals occurred in the current year, paid in previous one(s).
Household	Premiums in Household LOB	
Health	Premiums in Health LOB	
Life	Premiums in Life LOB	
Work_Compensation	Premiums in Work Compensations LOB	

5. Importing, Exploring and Preprocessing Data

5.1. Importing Data

First, we imported to the Python IDE (Integrated Development Environment) the data from both provided sources “insurance.db” and “A2Z Insurance.csv”. Using the **sqlite3** package we queried the “insurance.db” file and found the tables “Engage” and “LOB” (Line Of Business). The columns of these two tables, match the attributes described in Table 1. so we imported all the data through “SELECT * FROM table” SQL queries. We imported the static CSV data using pandas. We decided to work with the “insurance.db” file since both files contain the exact same information but since the database data is more likely to be kept up to date than a static data file, we proceeded with the database as our data source as a best practice.

5.2. Exploring and Transforming Data

To perform exploring and transforming tasks we used the **pandas** package. Both “Engage” and “LOB” tables contain the customer ID number, therefore we merged these two DataFrame into a single data frame to start working with a single object containing all the information. Due to the merging process, both the indexes of each DataFrame appear as independent columns, we drop those columns as our merged data frame has its own new index. After a quick exploration and panda’s description of the whole data frame, which shows that the “Education” attribute is the only one with categorical values. We manipulate this column in order to split the code from the description of the Education Level variable. For instance, “2 - High School” is now split into “2” in the edu_code column and “High School” in the edu_desc column. We decided to keep the edu_desc column only, as we intend to use K-Modes algorithm to analyze the categorical data in our DataFrame. We also considered and tried creating Dummy Variables for this feature but due to its increase in dimensionality of our input space and since this interferes with our goal was to use unsupervised learning algorithms involving clustering we decided to not do this.

Next, we explore the DataFrame to check for missing values row by row. As well as the percentage of null values by column. We decided to fill null values in the premium Line of Business (LBO) type columns with 0, as we interpreted this as meaning that the customer did not have this type of policy (it coincided with these columns not having any values equal to 0) and didn’t want to lose this precious information before we treated the null values. We found 92 customers had missing values, in at least one attribute. This correspond only to 0.89% of the rows. So, we decided to simply move these rows to a separate null values DataFrame, leaving us with 10 204 customers.

Continuing with the data exploration, we performed a business logic validation test to each column to check all values are within a logical frame and prevent potential outliers. For example: for the “customer ID”, there should not be repeated values, only positive values. For “birth year”, there should not exist any values larger than 2016, or negative values. After checking all columns, we just dropped 2 rows with inconsistencies, one of a customer with a value for birth year that was too low, which indicated that this customer was almost 100 years of age and another row for a customer with a “policy creation year” value larger than 2016 was found so we decided to drop this row as well having now 10 202 customers in our DataFrame.

In addition, in order to have more categorical features for our K-Modes analysis we converted the dummy variable of has_children from numerical to categorical by mapping 0 to “No” and 1 to “Yes”. We also converted our label encoded variable for geographical area from numeric to categorical for the same reason.

We moved onto another logical test involving the relationship between the “birth year” and “the creation year”, taking the assumption that only adults can take on insurance policy’s we asked the question of whether there were any occasions in which the “policy creation year” column was smaller than that of “birth_year” value + 18 years of age i.e that the customer created the policy before they were 18 years old. We uncovered that this was the case for 5031 customers, this represents almost half of our DataFrame after removing outliers and invalid data. This makes us believe that the “Birth Year” column is untrustworthy, very likely data collected manually via the insurance forms and then inputted into the systems leading to high probability of human record. Therefore, we decided to drop the “Birth Year” Column all together from the DataFrame. Another hypothesis would have been that the “policy creation

year” column is the one untrustworthy, especially since its value’s range indicates that the youngest customer tenure is 22 years and the oldest 46, indicating that there hasn’t been a new policy in the last 22 years. But since this attribute is more likely to be generated by the system we deem it trustworthy and therefore we keep it as it contains useful information for the feature engineering in the next step.

5.3. Performing Feature Engineering to Calculate Additional Attributes

Additional and valuable information can be extracted from the provided data frame, therefore we calculated and added the attributes shown in Table 2 to our Dataframe as new columns.

Table 2. Additional calculated attributes and their description.

New attributes	Formula/ Logic
Cancelled Premiums Percentage	Count of columns with negative premiums (indicating a reversal compared to the previous year) from the 'motor_premiums', 'household_premiums', 'health_premiums', 'life_premiums', 'work_premiums' divided by the total active policies the previous year
Current Active Premiums	Count of columns with positive (non-zero) premiums from the 'motor_premiums', 'household_premiums', 'health_premiums', 'life_premiums', 'work_premiums'
Customer tenure	Customer tenure= today year - customer policy creation year
Customer acquisition cost	Customer acquisition cost = (customer annual profit)(customer policy age) – CMV
Total premiums	Total premiums = Premiums in Motor + Premiums in Household + + Premiums in Health + Premiums in Life + Premiums in Work Compensations
Amount paid by the company	Amount paid by the company = (Claims Rate)(Total Premiums)
Premium/ Wage Ratio	Premium/ Wage Ratio = Total Premiums / (Gross Monthly Salary*12)

Note: to calculate “today year” we used the **datetime** package

Finally, after performing exploring, cleaning on the data, we change the attributes names to shorter names. Thus, the data frame we are going to use to develop the Customer Segmentation analysis will have the attributes and corresponding data type shown in Table 3.

Table 3. Data frame attributes and their data type.

Attribute Name	Short Name	Data Type
Customer Identity	"DataFrame.index"	int32
First Policy's Year	policy_creation_year	int32
Birthday Year	birth_year	int32
Gross Monthly Salary	gross_monthly_salary	float64
Geographic Living Area	geographic_area	string
Has Children (Y=1)	has_children	string
Customer Monetary Value	customer_monetary_value	float64
Claims Rate	claims_rate	float64
Premiums in LOB: Motor	motor_premiums	float64
Premiums in LOB: Household	household_premiums	float64
Premiums in LOB: Health	health_premiums	float64
Premiums in LOB: Life	life_premiums	float64
Premiums in LOB: Work Compensations	work_premiums	float64
Cancelled Premiums Percentage	cancelled_premiums_pct	float64
Current Active Premiums	active_premiums	float64
Education Description	edu_desc	string
Customer Tenure	cust_tenure	int32
Premiums/Wage Ratio	premium_wage_ratio	float64
Customer Acquisition Cost	cust_acq_cost	float64
Total Premiums	total_premiums	float64
Amount paid by the company	amt_paidby_comp	float64

4.4. Detecting and Handling Outliers

For data preprocessing to be successful, it is essential to have an overall picture of the data. Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers [2]. For instance, we standardized the data frame *df* with **StandardScaler** from **scikit-learn**, just to visualize all variables in a single chart, and then plotted a Box-plot for each variable to notice the distributions and identify potential outliers. Results are shown in Figure 1, in which is obvious the presences of outliers in most of the variables.

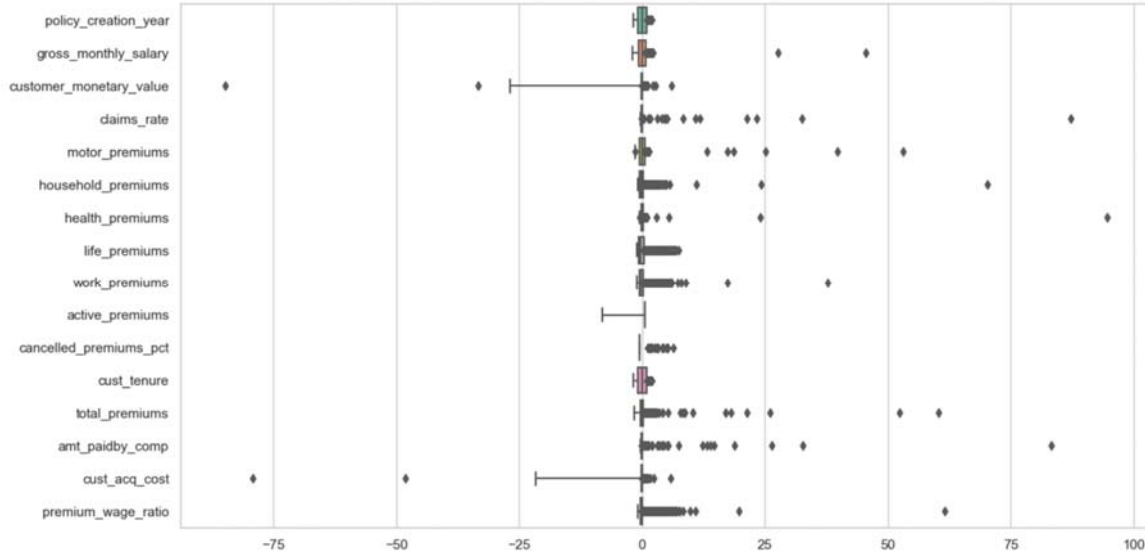


Figure 1. Box-plots for each standardized variable of *df*.

To determine which individuals are outliers we opted for applying the quantile criteria to determine outliers [2], choosing the 1st quantile as lower bound and the 99th quantile as upper bound. Figure 2 displays the Box-plot of each variable after excluding outliers. It is important to mention that the same quantile criteria were used to plot both charts and for determining which values are outliers i.e. the ends of the whiskers are placed at the 1st and 99th quantiles. The number of outliers were 1300, 12.7% of the 10202 customers mentioned on section 4.2, since this is a small amount, we excluded them from *df* and stored them in *df_outliers*, so finally, *df* will have information about 8899 customers.

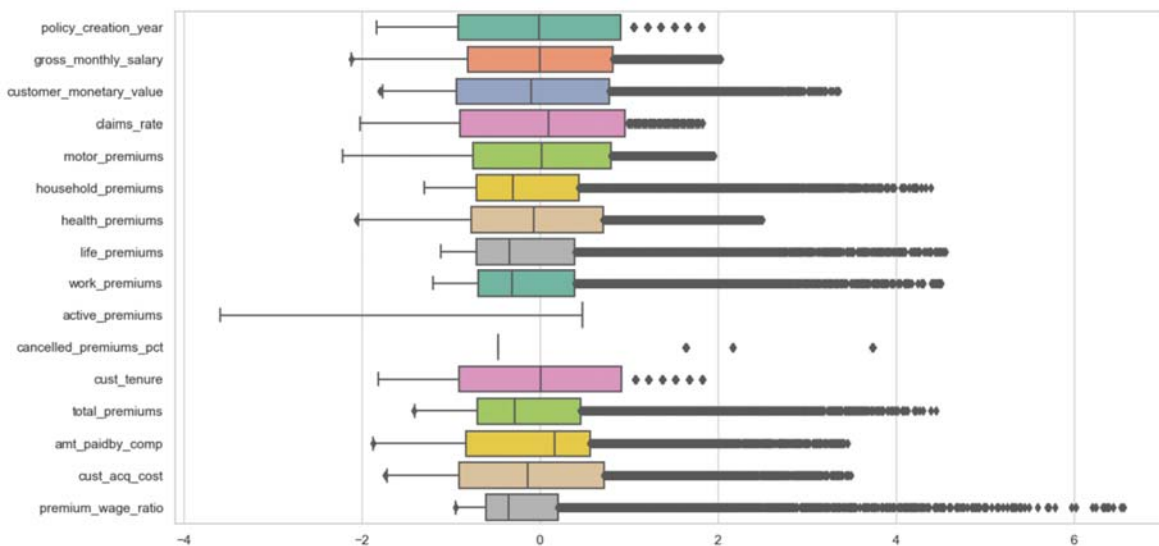


Figure 2. Box-plots for each standardized variable of *df* after excluding outliers.

6. Data Visualization

In order to get insights about the cleaned data, in this section we used **matplotlib** and **seaborn** visualization to ease the comprehension of distributions, proportions and relationships of variables. The first variable we addressed, due to its importance, was gross monthly salary. The following plots



Figure 3. Monthly salary histogram. 20 bins.

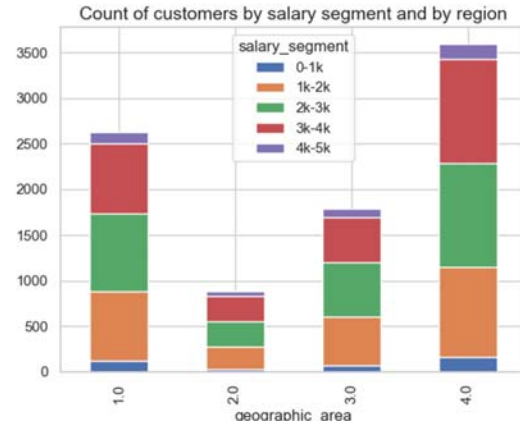


Figure 4. Count of salary segment by geographic area.

Figure 3 shows the frequency of the variable gross monthly salary along 20 defined bins. This histogram follows a regular bell shape distribution with a low Kurtosis, being the lowest salary EUR 654 and 4 430 the highest one. Figure 4 consider 6 segments of salary range, and aims to expose the proportions of salary ranges by region and the number of customers per region. There are more customers in region 4, nevertheless, the proportions of salary ranges are almost the same for each region, so geographic should not be a valuable variable to estimate customers with lower or higher salaries.

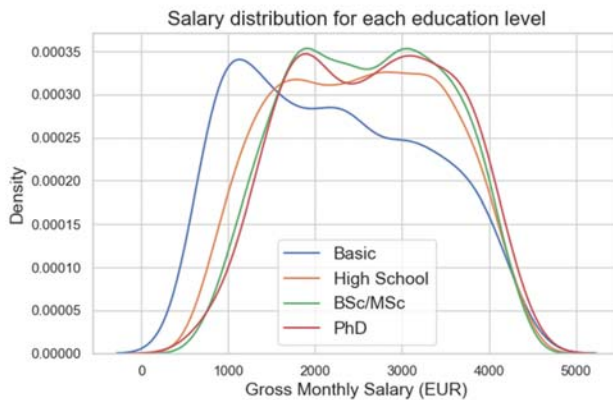


Figure 3. Distribution of salary by level of education.

Figure 5 exposes that only customers with Basic education has a lower, skewed salary density. The rest education level categories are homogeneously dense distributed between EUR 900 and 4 000. We can conclude that education level is not a valuable variable to determine a customer's salary, except for customers with Basic level of education. We have developed more graphic displays to describe statistics on data personal information of customers, but in order to save space, they will be just mentioned on Table 4.

Table 4. Count of customers by categorical variable

Salary segment (EUR)	Count	Percentage
0 - 1k	380	4%
1k - 2k	2 542	29%
2k - 3k	2 859	32%
3k - 4k	2 685	30%
4k - 5k	433	5%
5k - 6k	0	0%
Geographic area		
1	2 622	30%
2	886	10%
3	1 791	20%
4	3 597	40%
Has children		
Yes	6 328	71%
No	2 571	29%
Education level		
Basic	910	10%
High School	3 113	35%
BSc/MSc	4 290	48%
PhD	585	7%

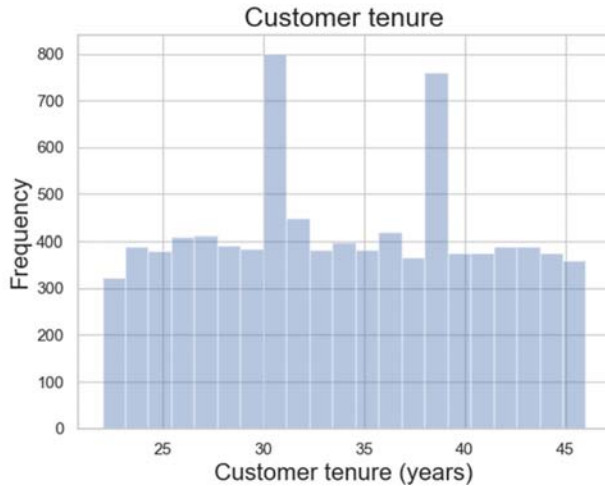


Figure 4. Customers tenure histogram.

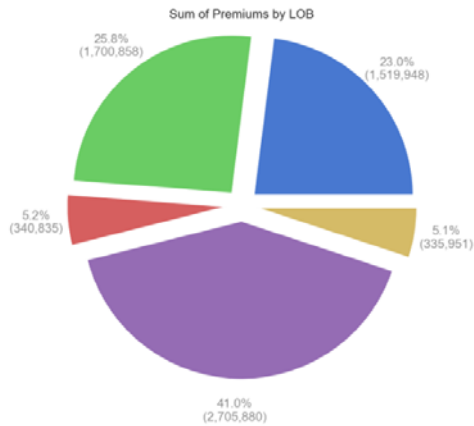


Figure 5. Distribution of Premium by line of business

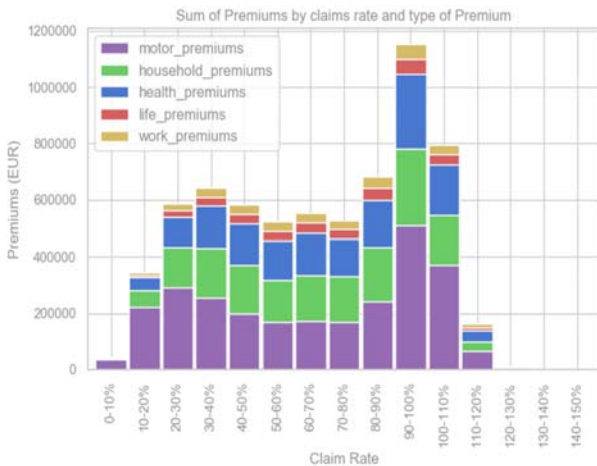


Figure 6. Sum of premiums by claim rate and LOB

Getting more into data related to the insurance business, Figure 6 shows a histogram for policy age. This histogram illustrates the behavior of customer acquisition by the insurance company over time. It explicitly shows a constant distribution on the policy age for customers, which means that the company acquired almost the same quantity of customers each year (around 380, excluding outliers). Except, for the years 1983 and 1978 when the company gained more than 700 customers. The strange fact about this variable is that the most recent customer acquisition has 22 years old, so the company did not have new customers since 1994.

The pie chart of Figure 5 shows the sum of premiums of all customers by LOB (Line Of Business). The sum of premiums is EUR 6 603 475.45, considering reversals. LOB can be ranked from higher to lower amount of premium in this order: Motor, Household, Health, Life and Work. Please notice that the legend displayed on Figure 10 applies for Figure 9 as well.

An important part of an insurance company situation can be exposed by taking a look to the claims rates of all customers. This is shown on Figure 6, where the claims rate is segmented by 10% bins and the sum of claimed premiums is stacked by LOB. By the claims rate definition exposed on Table 1, it is possible to determine an unfavorable situation for the insurance company, since most of the money has being returned to customers.

There are two conditions for a premium: reversal and active. Negative values on premiums represents a reversal condition and positive values active conditions. As there are 5 types of premiums, each customer will have a condition for each premium, meaning that a customer can have reversal condition in just one or many of the premiums. After the calculation, there are 1 956 reversal conditions on the premiums, without considering outliers.

The number of customers holding premiums by LOB is shown on Table 5. The percentage value of this table is calculated by the count divided by the sum of all customers, with or without reversals. From these facts it is possible to argue that reversals are occurring more on life, work and household LOB.

Table 5. Count of active premiums

Premium	Count	Percentage
health	8899	100%
motor	8073	90%
life	8358	93%
work	8899	100%
household	8166	91%

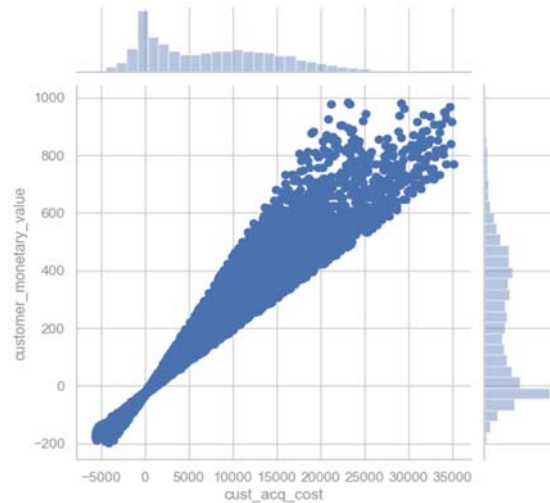


Figure 8. Scatterplot for CAC and CMV

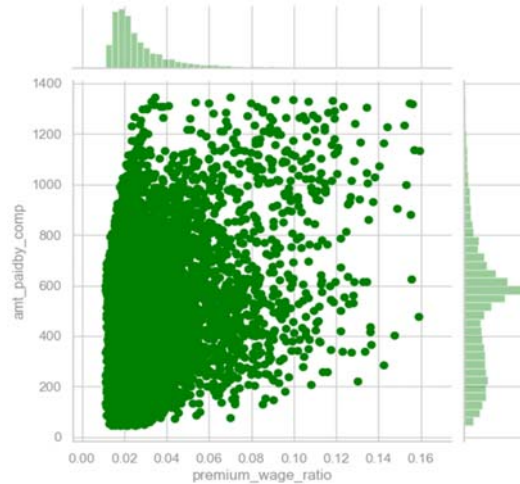


Figure 7. Scatterplot for PWR and Amount paid by company

Continuing with the variables visualization, the Customer Monetary Value (CMV) and the Customer Acquisition Cost (CAC) are critical to perform customer segmentation, due to they directly prompt the value of each customer for the company. Figure 8 shows a join-scatter plot in which both variables distribution and correlation are represented. Just looking at these 2 variables a positive trend between them. Which implies, that the company invest more on gaining or maintaining customers with higher salaries, according to CMV definition in Table 1. It's important to notice at the CMV and CAC histograms, it is possible to distinguish peaks near zero values.

Figure 7 displays the Premiums Wage Ratio (PWR) and the Amount paid by the Insurance Company for claims, during the last two years, 2015 and 2016. Just looking at PWR it is possible to analyze the amount of money that the premiums represents over each customer wage and determine if the premiums are a great load for their wages, 3% is the highest frequency shown for this variable. By other hand, the amount paid by the insurance company for claims exhibits a high frequency on payments between EUR 500 and 750, however the total amount paid by the company during those last two years was EUR 4 478 588, excluding the outliers.

7. Input Space Reduction and Categorical / Numerical split

After the feature engineering we have a much larger dimensional space now having a total of 19 variables. Many of these are new variables that were created from manipulating the old variables. In order to avoid the curse of dimensionality we are going to use correlation analysis together with hierarchical clustering to detect and remove redundant attributes. The aim of this is the reduction of our input space used for the unsupervised learning tasks i.e. to only take forward a subset of the dataset to perform clustering analysis upon. To help with this we use this hierarchically-clustered heatmap. (Figure 11)

As seen in Figure 12 there are many features that are highly correlated with each other. For example, Claims Rate is negatively correlated with Customer Monetary Value and Customer Acquisition Cost. Claims Rate is positively correlated with the amount paid by the company, which makes sense, because the amount paid by the company is a calculated value that depends of Claims Rate. K-means considers all variables regardless of their relationship with each other. If, for example, two variables are characteristics of the same underlying feature, both will be taken into consideration, and thus, this underlying feature will be weighted with two variables, creating bias in the clustering analysis.

Even though Claims Rate is correlated with Customer Monetary value we decided to keep both, as we believe that the underlying factor behind both these attributes that we know to be good discriminators for the insurance industry and therefore it is worth weighing twice into our cluster analysis.

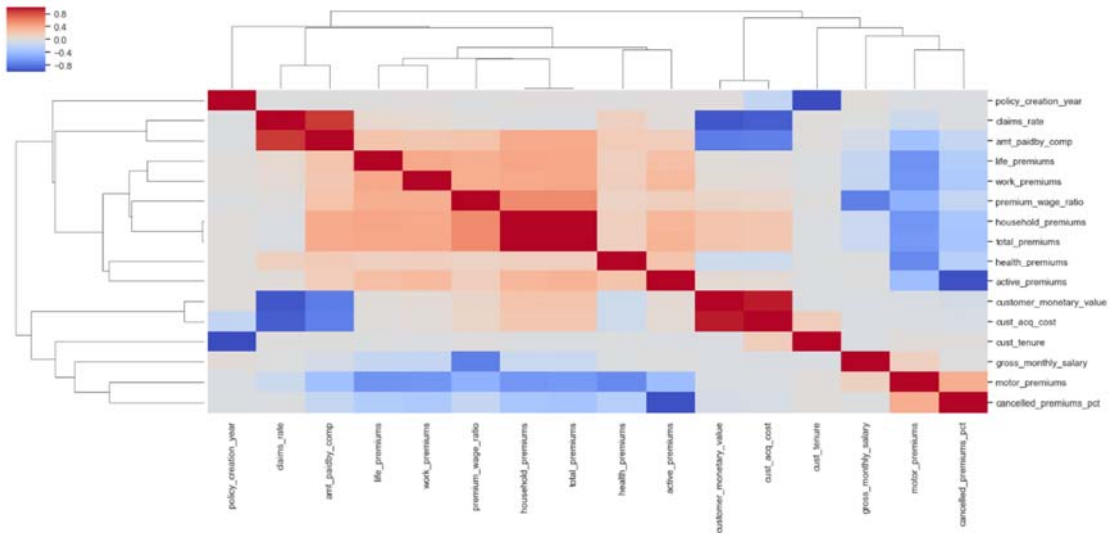


Figure 11. Correlation matrix for all numerical variables

In order to reduce the input space, we removed the following of these numeric variables because they are highly correlated with other variables: `policy_creation_year`, `gross_monthly_salary`, `active_premiums`, `total_premiums`, `amt_paidby_comp`, `cust_acq_cost`. We also remove `cust_tenure` as it seems to have irrelevant information, so we take it into account in our categorical variables analysis by creating segments of tenure years. Finally, for each of the LOB premiums variables we convert its value to percentage of each premium with the `total_premiums` variable such that $LOB_premium_pct = LOB_premium / total_premiums$. After all the above steps, our DataFrame before clustering `clust_df` looks like the following Figure 12.

The next step was to standardize the data using **StandardScaler** function from `sklearn`, so that the data has mean 0 and standard deviation of 1 and load this standardized data into the `X_std_df`.

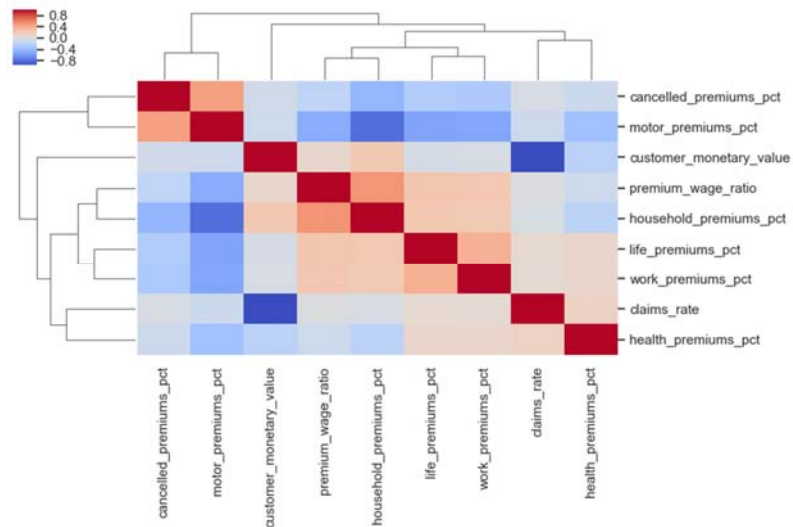


Figure 12. Correlation matrix after input space reduction

This new data frame `clust_df` is then divided into two dataframes with a subset of variables each. While the categorical variables are stored separately in their different dataframe:

1. **clust_df:** Contains all the variable shown in Figure 12
2. **X_value_df:** Contains the subset of “Value” variables: `cancelled_premiums_pct`, `claims_rate`, `customer_monetary_value`, `premium_wage_ratio`.
3. **X_prod_df:** Contains the subset of “Product” variables: `motor_premiums_pct`, `household_premiums_pct`, `health_premiums_pct`, `life_premiums_pct`, `work_premiums_pct`.
4. **Engage_df:** Contains all the “Engage” categorical variables: `geographic_area`, `has_children`, `edu_desc`, `salary_bin`, `tenure_bin`.

Note: In the code, we standardise and assign only one of the DataFrame we use for the rest of the analysis to `Std_clust_df`

8. Clustering

In these days, we are living in a world full of data. Every hour, people deal with a large amount of information and store or represent it as data, for further analysis. One of the crucial means in working with these data is to classify or group them into a set of categories or clusters. By definition, data clustering is the process of identifying natural groupings or clusters within multidimensional data based on some similarity measure (e.g. Euclidean distance) [3]. Clustering can be done by the different number of algorithms such as hierarchical, partitioning, grid and density based algorithms. In this report we are going to focus on the hierarchical and partitioning clustering. The partitioning is the centroid based clustering and the hierarchical clustering is the connectivity based clustering.

8.1. Partitional Clustering

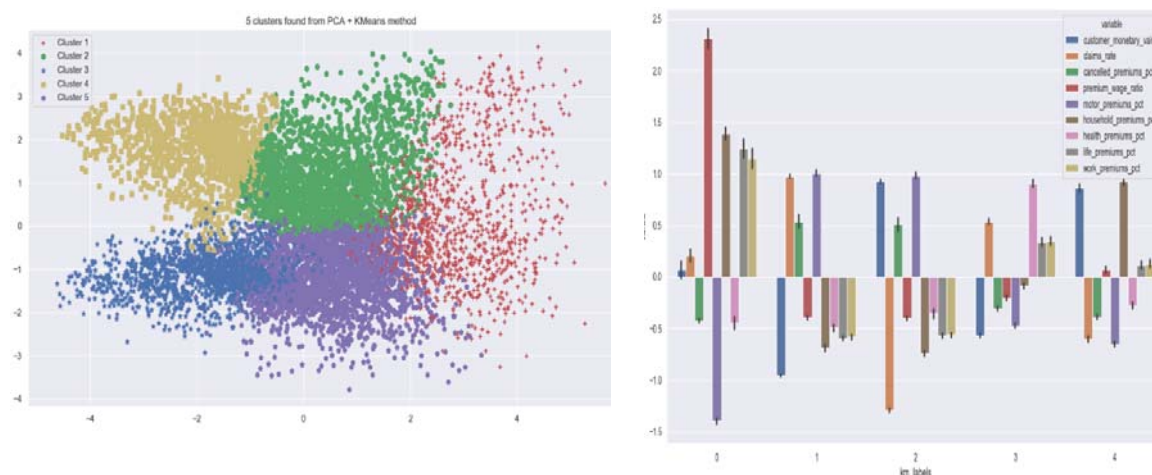
To make things clear, partitioning algorithms divide data into several subsets. They start with a big cluster and keep splitting it until they achieve the desirable number of clusters specified by the user. It tries to minimize certain criteria (e.g. a square error function) and can therefore be treated as optimization problems. There are many methods of partitioning clustering; they are K-Means, Medoids Method, PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), the Probabilistic Clustering, etc. In this report we cover the subset learned in class. These heuristic clustering methods work well for finding spherical-shaped clusters in small- to medium-size databases

8.1.1. K-Means Algorithm

K-means clustering is one of the simplest and most popular unsupervised machine learning algorithms. It divides a dataset into K clusters. This algorithm minimizes the intra-cluster distance. The K-means algorithm starts with K centroids (initial values for the centroids are randomly selected or derived from a priori information). Then, each pattern in the data set is assigned to the closest cluster. Finally, the centroids are recalculated according to the associated patterns. This process is repeated until convergence is achieved.

We tried running K-Means algorithm on the standardised full clustering Dataframe first, in order to mitigate the algorithm's sensitivity to noise we compare multiple forms of initialization and re-initialize several times, using the labels and centroids of the best run to perform our first K-Means attempts. We run K-Means on a range of K from 1 to 10, using the optimised parameters in order to be able to draw an elbow plot and understand the optimal level of K based on the marginal decrease in the sum of squared errors obtained when adding an extra cluster. Based on the elbow plot shown in Figure 14, we decided to run the final K-Means algorithm using K = 5 as our specified number of clusters.

We use the first 2 Principal Components (PCs) to be able to Visualize the clustering results in 2D through a scatter plot (we tried including the centroid centers, but due to the PCs these weren't in the correct place), the result is displayed on Figure 15, we repeat this process throughout this section of the report. Although PCA is successful reducing the dimensionality of the data, it does not seem to visualize the clusters very intuitively. This happens often with high dimensional data, as they are typically clustered around the same point and PCA extracts that information.



8.1.1.1. Principal Component Analysis with K-means

Continuing to try and fight the curse of dimensionality we use Principal Component Analysis to find the principal components of data. They are the directions where there is the most variance, the directions where the data is most spread out. We try to perform Factor Analysis in order to extract the underlying factors of all the variables in our clustering dataset, and then perform k-means on the obtained factors.

After performing PCA we find that most of the variance is captured by the first 4 Principal Components, these are the 4 variables we apply our K-Means algorithm on.

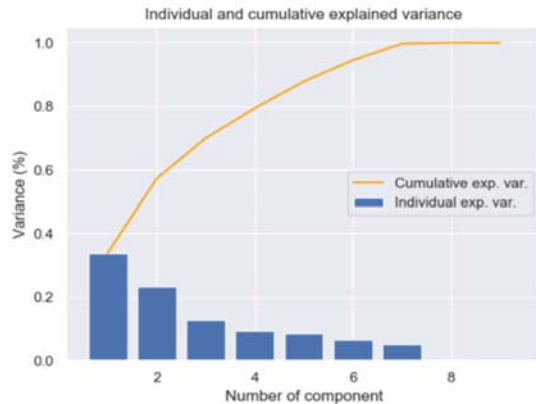


Figure 13. Explained variance of each principal component

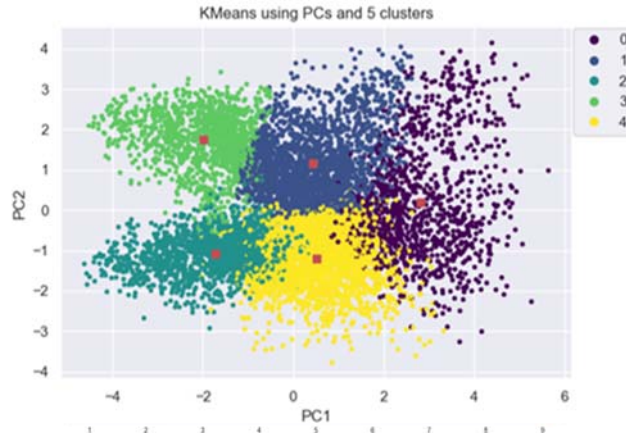


Figure 15. K-Means using PCs and 5 Clusters

Figure 14. Elbow graph indicates 5 clusters

We got the following results for the centroid centers for the K-Means with 5 clusters. In order to interpret these, we need to understand the relationship between the PCs and our original variables in the standardized clustering Dataframe for this we visualized the first 4 PCs in the colormap in Figure 16.

As you can see a disadvantage of this procedure might be the difficult interpretability of the factors, since not only we need to interpret the cluster center coordinates for each PC, but also we need then to map this relationship with that between the PCs and the original variables. For example, the cluster with Label 1, has the highest value in for PC2, which subsequently captures the most variance in the customer_monetary_value and claims_rate features, therefore we could interpret this cluster as representing high-value customers.

Table 6. Centroids and number of customers.

K-means + PCA with 5 clusters. Silhouette score: 0.1903					
PC1	PC2	PC3	PC4	Cluster Label	Customers Count
2.82	0.20	0.90	0.71	0	1156
0.43	1.15	-0.29	-0.41	1	2216
-1.73	-1.07	0.87	0.04	2	1766
-1.99	1.74	-0.29	0.44	3	1229
0.51	-1.20	-0.63	-0.21	4	2532

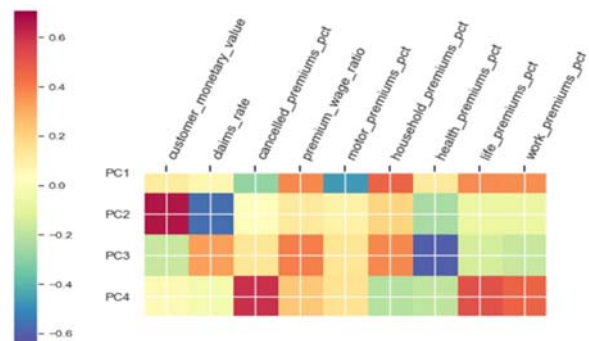


Figure 16. Colormap First 4 PCs

We found that a more intuitive way of doing this was to take the labels produced by the K-means after PCA and add them as a column to the DataFrame. This way we could group the DataFrame by label and visualize the mean of each feature for each cluster directly (we repeat this process throughout this section for analysis purposes). As you can see for the Label 1, it is easy to identify the high customer monetary value by just glancing at Figure 17. We can also make up the household_premium_pct as the most valuable LOB for this segment of customers, this was “hidden” in

the first Principal Component and couldn't be identified as easily unless we had produced this method for interpretation.

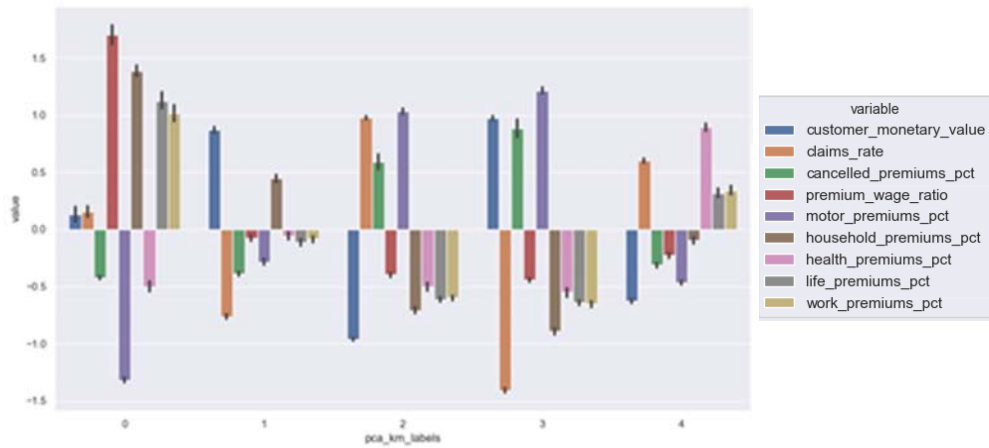


Figure 17. Mean of customers' feature by cluster. PCA + K-means

8.1.1.2. Neural Networks: Self-Organizing Map (SOM)

This type of algorithm represents each cluster by a neuron or “prototype”. The input data is also represented by neurons, which are connected to the prototype neurons. Each such connection has a weight, which is learned adaptively during learning.

A SOM is an artificial neural network composed by a grid of output neurons connected to an input layer. This type of neural network uses an unsupervised learning algorithm to find clusters in data without any privileged knowledge a priori. The algorithm maps a multidimensional training set in a 2D grid of neurons in a way that preserves the original topological relationships. SOMs are mainly a dimensionality reduction algorithm, with the capacity to preserve a given topology of output representation. In that frame, we used SOM to discover patterns throw all the dimensions of our data frame. To perform this technique, we set typical hexagon shapes on the lattice parameter and a 20, 20 map-size of 400 neurons in total to have a better capture of the original topology of data. It took 50 iterations and 9.9 seconds for neurons in SOM grid to stabilize and get a final quantization error of 1.20. Figure 18 give us an overview of how many data points each neuron corresponded to. Each neuron is represented by a hexagon, and the color scale represents the relative number of data points that neuron is positioned closest to. Until that point, cannot be distinguished clear clusters yet.

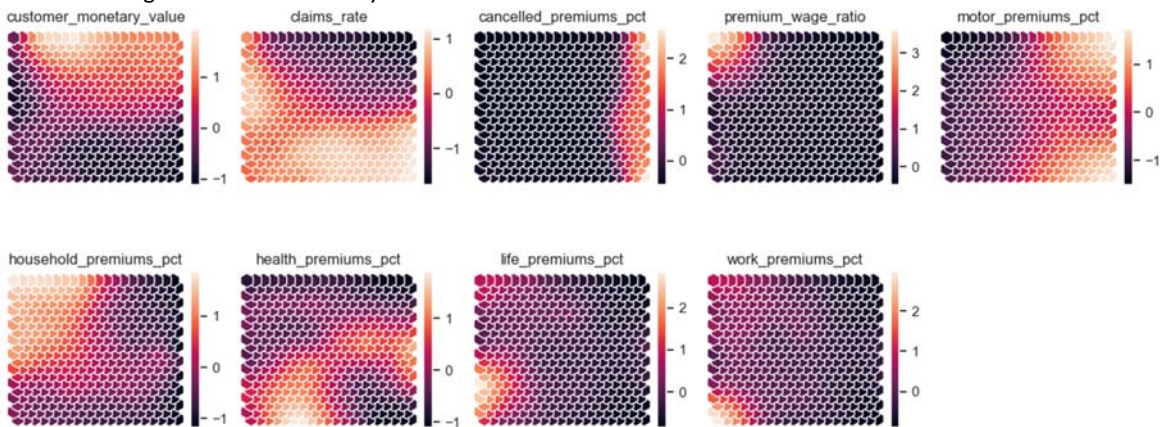


Figure 18. Plane Plotting for each training variable.

To verify that there is indeed a divide, we can plot what’s called a U-matrix shown in Figure 19, gives us a sense of the landscape of the SOM grid, and what the neighborhoods of the neurons are. When two neurons correspond to vastly different sets of data points, they would be separated by a larger distance, denoted by a blue color. On the other hand, neurons representing similar data points are separated by shorter distances, denoted by a red color. So we can observe two clusters, one on the top right corner and other on the left bottom corner of the U-matrix. Nevertheless, this can be only our personal perception, that is why used the elbow graph conclusion from the section 9.1.2 to find the optimal number of clusters. The results of 5 clusters are shown on the Figure 20.

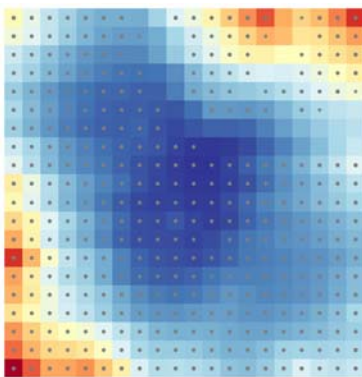


Figure 19. U-matrix showing similarity and dissimilarity between neurons.

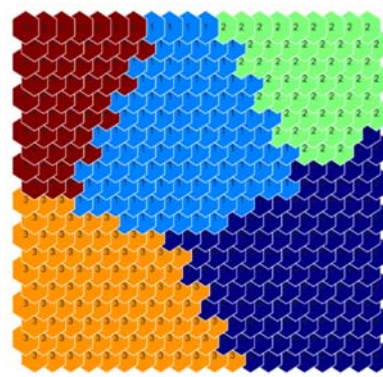


Figure 20. Applying K-means for K=5, over SOM results.

8.1.2. K-modes Algorithm

K-modes algorithm defines clusters based on the number of matching categories between data points. We performed K-modes clustering over the Engage data frame, which mostly has categorical variables and two numerical variables, `gross_monthly_salary` and `cust_tenure`, which were transformed into categorical by using `pandas.cut` function to bin their values into segment. The Engage data frame, with all its possible values is presented on Table 7. Where the salary bin represents salaries in thousands scales and tenure bin segments are expressed in years.

Firstly, we ran the K-Modes algorithm to compare the cost against each K cluster, in order to decide a reasonable number for K, where K is the parameter that controls the number of clusters. Figure 21 shows the results of this execution and our choice of K=3 clusters.

The resulting 3 clusters for the K-mode algorithm are shown on Table 8. Those clusters represent the most common values that describe the customer's profiles i.e. the mode of each cluster, but not much concluding information can be extracted from this algorithm results. Just the number of customers allocated into each cluster can show some relationship with the data exposed on Data Visualization section. For this reason, we decided to extract the two principal components from the numerical standardized data frame `X_std_df` shown on Figure 12 and label the scatter plot of these principal components with the K-modes results to see if an important pattern is displayed. The result of these procedure is shown on Figure 22. Unfortunately, no clear pattern can be observed.

Table 7. Input variables for K-mode

geo_area	has_children	edu_desc	salary_bin	tenure_bin
1	Yes	Basic	0-1k	20-25
2	No	H School	1k-2k	25-30
3		BSc/MSc	1k-2k	30-35
4		PhD	2k-3k	35-40
			4k-5k	40-45
				45-50

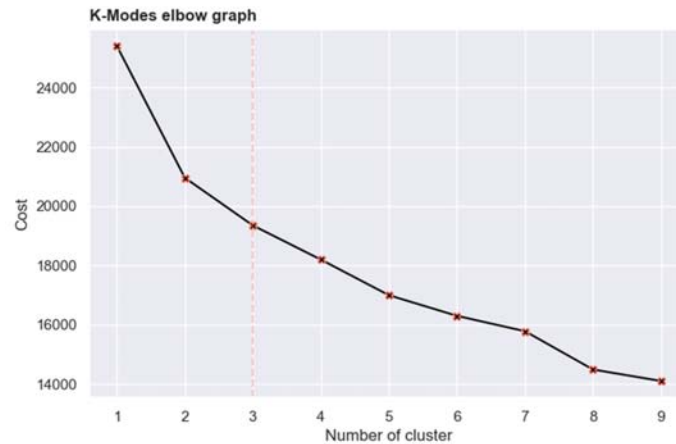


Figure 21. K-modes cost vs. number of cluster elbow graph

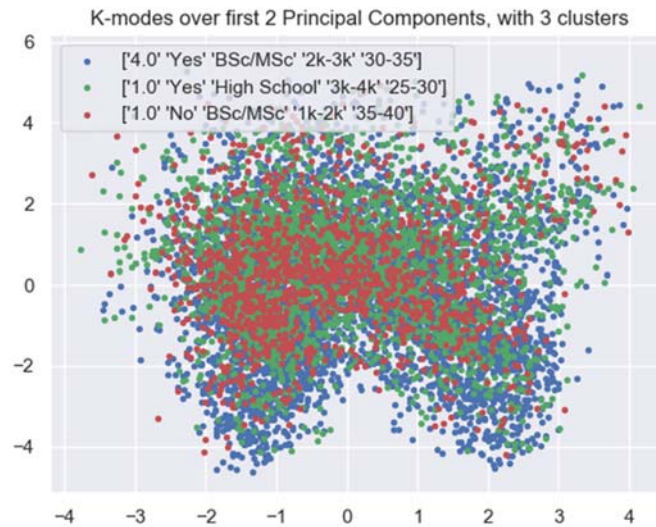


Figure 22. K-modes labels over 2 principal components.

Table 8. Results of K-modes clustering for each categorical variable.

	Customers count	geo_area	Has_children	Educ_desc	Salary_bin	Tenure_bin
Cluster 1	5 128	4	Yes	BSc/MSc	2k-3k	30-35
Cluster 2	2 541	1	Yes	H School	3k-4k	25-30
Cluster 3	1 230	1	No	BSc/MSc	1k-2k	35-40

8.1.1. The Gaussian Expectation-Maximization Algorithm

Another popular clustering algorithm is the Expectation-Maximization (EM) algorithm. This is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing K-Means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. For that reason, we re-use the elbow graph conclusion from previous step to determine the number of cluster $K = 5$ to apply this technique.

Results of this algorithm are shown in Figure 23 and Figure 24 which are not similar to the PCA + K-Means results, in other words, the results of these two algorithm point to completely different customer's profiles, which leads to completely different marketing strategies.

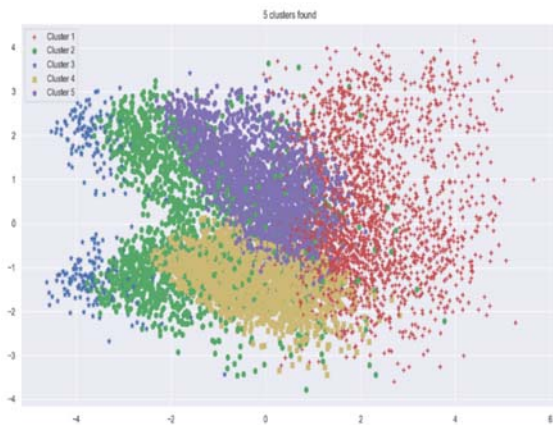


Figure 23. Gaussian Mixture scatter plot for 5 Clusters.

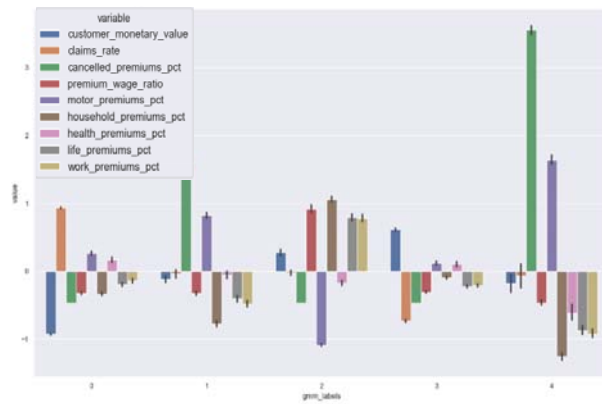


Figure 24. Mean of customers' feature by cluster. GMM.

8.2. Density Based Clustering

To execute density-based clustering, it is not needed to specify a number of clusters we're interested in— it will automatically discover some number of clusters based on the specified parameters. This is especially useful when you expect all of your clusters to have a similar density.

8.2.1. The Mean Shift Algorithm

The mean shift algorithm automatically finds the number of clusters in a data set and can work with arbitrary shaped clusters. The mean shift algorithm starts with a number of K estimators in the input space. These estimators are then repeatedly moved towards areas of higher density. When they all reached stability, all the k that are near to each other are grouped together. For this clustering approach we tried many quantile distances on the estimate_bandwidth parameter of scikit-learn, in order to find a consistent number of clusters. The main problem with this technic was the sensitivity of the quartile parameter value and the inconsistency of the results in term of number of clusters and similarity of the clusters results on each variable, that is, for small changes on this parameter, from 0.05 to 0.30, the number of clusters changes randomly from 3 to 9 with no consistent behavior on the variables. Same behavior occurred for the three data frames: products, values and the data frame with all variables, we even tried to perform this algorithm with the SOM results, but since the SOM algorithm summarizes the data no clear density

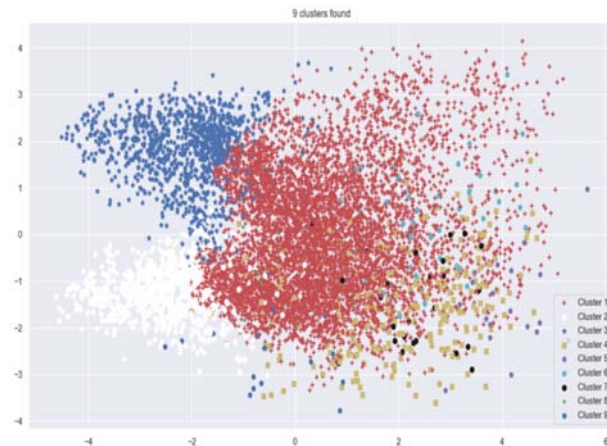


Figure 259. Mean shift clustering using quartile distance = 0.11

pattern was founded by Mean Shift Algorithm. For this random behavior, we decided not to base our conclusion using this algorithm.

8.2.2. DBSCAN

DBSCAN works by running a connected components algorithm across the different core points. If two core points share border points, or a core point is a border point in another core point's neighborhood, then they're part of the same connected component, which forms a cluster. All the points that don't satisfy this get assigned a label of -1, which is often interpreted as Noise. The sensitivity of this algorithm can be adjusted by changing two main parameters for DBSCAN are the minimum number of points that constitute a cluster (minPts) and the size of the neighborhood (eps). We do not want minPts to be very small as clusters from noise will be generated more often, it is better to set minPts to at least the number of columns in your input space, we set this parameter at 50. eps is a bit more difficult to optimize, but should be set to a small value, we chose 1.

Results of DB-Scan execution are shown on Figure 26 and 27, from those, we observe that DBSCAN generated clusters are more heavily impacted by the cancelled_premiums_pct and we can see more clearly its relationship with the cancelled premiums variables. This makes sense as because of the all the points in black that weren't considered in the clustering profiling as they were market by DBSCAN as

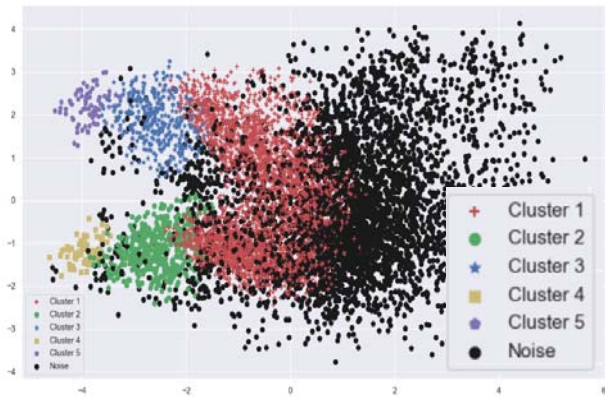


Figure 26. DB Scan clustering, recognizes too much noise.

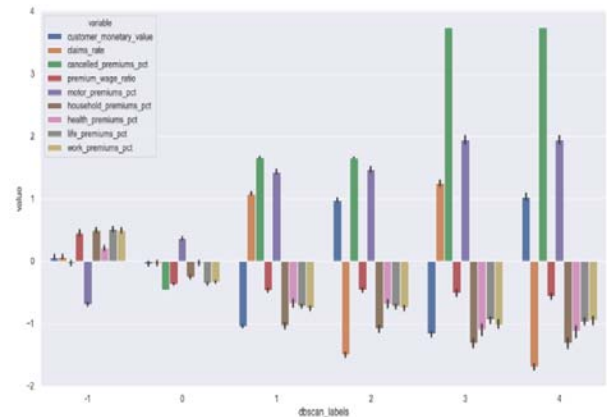


Figure 27. Mean of customers' feature by cluster. DB-

8.3. Hierarchical Clustering

Hierarchical clustering is an algorithm that builds hierarchy of clusters. Hierarchical clustering, does not require the user to specify the number of clusters. Initially, each point is considered as a separate cluster, then it recursively clusters the points together depending upon the distance between them. The points are clustered in such a way that the distance between points within a cluster is minimum

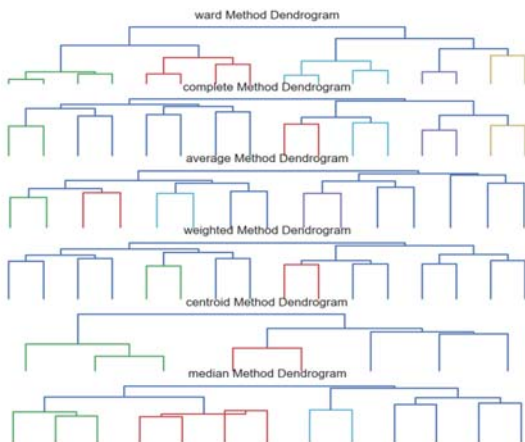


Figure 28. Dendrograms for 6 different Hierarchical clustering techniques.

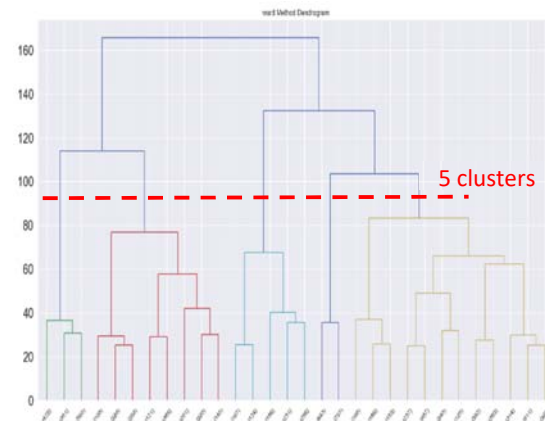


Figure 29. Ward Method Dendrogram for 5 clusters.

and distance between the cluster is maximum. Commonly used distance measures are Euclidean distance, Manhattan distance or Mahalanobis distance. Unlike k-means clustering, it is "bottom-up" approach. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left. The decision of the number of clusters that can best depict different groups can be chosen by observing the dendrogram shown in Figure 28 and Figure 29. We performed 4 Agglomerative Clustering techniques, each using a different type of linkage (average, complete, ward, and single), only the first 3 are visualised below as the dendrogram for the single linkage showed it's ineffective clustering ability, after comparing them, we decided to select ward method results due to its evenly distribution and the fact that it uses the Euclidean distance.

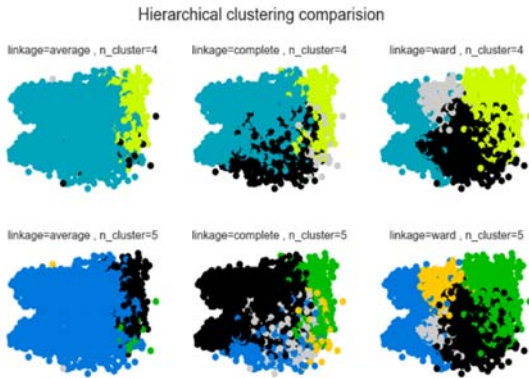


Figure 30. Scatter plots based on Hierarchical

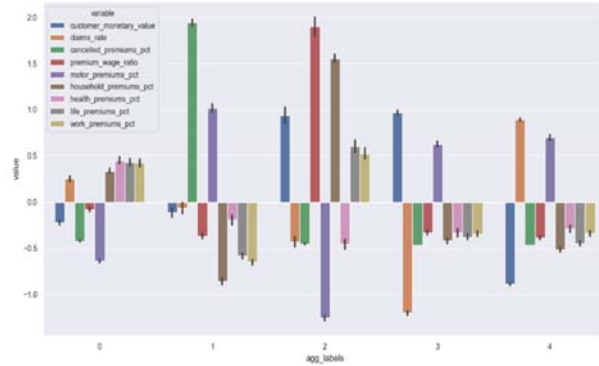


Figure 31. Mean of customers' feature by cluster. Ward method.

The best choice of the number of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster, like presented on Figure 30.

Based on Ward Method Hierarchical Clustering, we constructed the bar chart shown on Figure 31 to define and compare each customers' profiles by cluster.

8.4. From Clustering to Classification

In this section, we use DecisionTreeClassifier from sklearn to create a model that predicts the clusters labels for the outlier's customers by learning simple decision rules inferred from the complete data frame which data features. So the Decision Tree model was trained using the 9 variables chosen after input space reduction and with the clusters labels obtained by the K-means. The model was set with gini criterion, best splitter, a min sample leaf value of 3 and no constraint on max depth. The accuracy of this predictive model is 80%. Since the depth was not constrained to improve the accuracy, then the complexity of the model increases, but in order to present an illustration, we executed a model with max depth set to 3, this is shown in Figure 32.

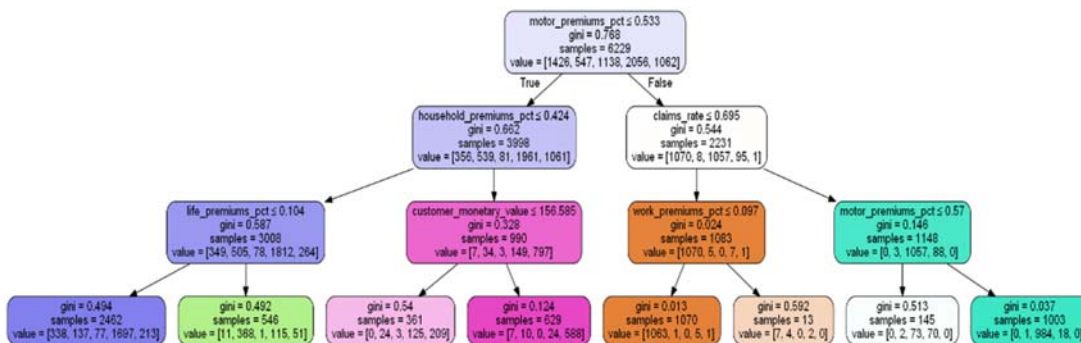


Figure 32. Decision Tree Classifier visualization, constrained with max depth to 3.

After predicting the labels for the outliers we concatenated the outlier DataFrame with the clustering DataFrame in order to recalculate the cluster centroids and see how the reintroduction of the outliers would affect the characteristics of our clusters previously identified. In Table 9 are specified the standardized values for the 5 recalculated clusters' centroids, there are presented the count of customers per cluster as well, which sum up all the customers considered after logical validation.

Table 9. Standardized recalculation of clusters' centroids after including the outliers.

CMV	claims rate %	cancelled prem. %	Prem. wage ratio	motor prem. %	house prem. %	health prem. %	life prem. %	work prem. %	Cluster	Customers count
162.73	75.5%	0.4%	2.8%	29.9%	28.1%	28.1%	7.0%	6.9%	0	3539
188.45	66.0%	22.9%	2.3%	67.9%	6.6%	21.6%	2.1%	1.7%	1	1648
429.25	54.9%	0.1%	6.7%	15.7%	50.1%	18.9%	7.9%	7.4%	2	974
436.7	31.7%	0.0%	2.3%	59.0%	14.5%	20.2%	3.1%	3.2%	3	1338
9.56	94.9%	0.0%	2.2%	60.6%	12.8%	20.6%	2.8%	3.3%	4	1400

A graphical representation of how variance of each cluster increases with the incorporation of the outliers customers is displayed on Figure 33, this is an expected behavior, since outliers are in essence non-typical values. Nevertheless, this is an important step to consider all the customers on the segmentation analysis.

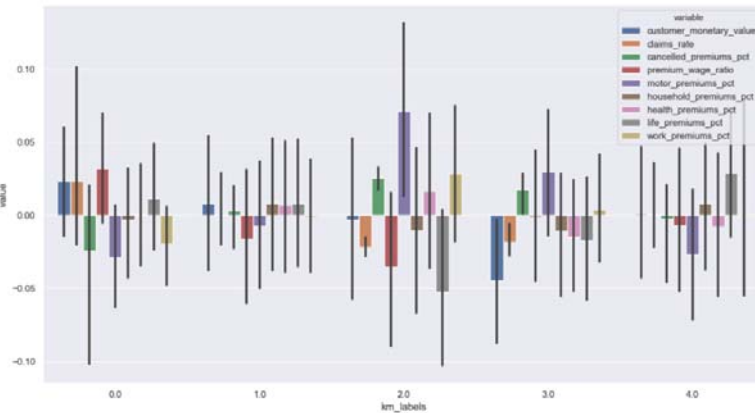


Figure 33. Mean of customers' feature by cluster. After adding outliers with Decision Tree

8.5. Clustering Assessment & Validation

The cluster validation problem is defining the number of clusters and the best clustering technic. For this reason, we assessed each clustering technic with the similarity score to decide which one is the best, a second, but not less important criterion was the interpretability of the variables for each cluster, since customer segmentation should follow a logic according to difference on customers. The last criterion was the importance of the data frame for marketing purposes. Being said that, the value data frame contains the most relevant information for marketing purposes. The execution for this data frame also exhibit the highest scores along the three data frames.

Finally, we decided to choose the clustering results from Kmeans with K=4 instead of Mean-Shift, like is highlighted on Table N. Despite the higher score of Mean-Shift, mainly because of interpretability, since Mean-Shift Clustering converged into 7 clusters and some of them are very similar.

Figure 10. Score comparison for Clustering techniques.

Data frame	Clustering Technique	Similarity score
X_std_df	PCA + K-means. K=4	0.208
	K-means. K=5	0.207
	PCA + K-means. K=5	0.190
	Gaussian-Mixture	0.152
	Mean-Shift	0.143
	Hier. clust., ward method	0.129
	DB-Scan	0.067
X_prod_df	Mean-Shift	0.335
	K-means. K=4	0.300
	Hier. clust., ward method	0.246
	Gaussian-Mixture	0.227
	PCA + K-means. K=4	0.208
	PCA + K-means. K=5	0.190
	DB-Scan	-0.075
X_value_df	Mean-Shift	0.462
	K-means. K=4	0.437
	Gaussian-Mixture	0.348
	DB-Scan	0.304
	Hier. clust., ward method	0.272
	PCA + K-means. K=4	0.208
	PCA + K-means. K=5	0.190

8.6. Interpreting Clusters

Because clustering is unsupervised, no “truth” is available to verify results. The absence of truth complicates assessing quality. Further, real-world datasets typically do not fall into obvious clusters. After assessing all the clustering techniques results, the best solution in terms of customer’s segmentation was found by executing K-means with K=4 over the “Value” attributes, which are the most significant for marketing purposes.

The interpretation of results leads us to the following deductions: Customers with Label 0 has a high Customer Monetary Value, moderate claims rate and almost no cancellations, so they are good Customers. Customers with Label 1 has a medium value for the company and present a moderate claim rate, but no cancelation on policies. A marketing solution for the claims rate can be applied to reduce claims. Customers with Label 2 should have a special treatment in terms of marketing campaign because they are customers that are cancelling policies, so discounts or promotions can be offered to retain them, since they are in the Medium monetary value range. Customers with Label 3 are the biggest cluster are Low Monetary Value customers with low cancellation rate, but with a high claim rate value, which allow us to suggest to the whole company to seek for fraud on those customers or raise the policies prevent revenue loss on by claims.

9. Conclusion

This project was based around a fictional insurance company in Portugal. The ultimate goal of this project was to segment clients into smaller groups which can hereafter be used by the Marketing Department to enhance Customers’ Profiles based on a data driven process.

After applying the techniques, we learned in this Data Mining course, we found business strategies can now be aligned with the special needs of a specific customer segments. The outcome of the segmentation, however, could be improved by refining the quality of data or by pre-selecting a groups of customers with special situation.

The data exploration, validation, cleaning and splitting was initially performed, then we focused on executing Clustering techniques and analyzing their results. Our customer segmentation approach was presented and evaluated over the report. We focused on the clustering analysis. Clustering techniques were tested and compared in order to achieve the best result possible. We did an extensive analysis of how the data should be preprocessed and weighted in order to get a better outcome from the clustering algorithm.

Our customer segmentation approach was presented and evaluated over the report. We focused on the clustering analysis. Clustering techniques were tested and compared in order to achieve the best result possible. We did an extensive analysis of how the data should be preprocessed and weighted in order to get a better outcome from the clustering algorithm.

Our main suggestion for the insurance company is to offer differentiated experience across all types of customers based on the interpretation described on previous section, nevertheless, these descriptions are general and further analysis can be done to find more detailed marketing plans.

Consumers believe organizations that provide custom content are interested in building good relationships. For instance, longer-tenured customers that spend more than the average consumer have come to expect a higher-degree of service in return for their loyalty. High-value customers (Label 0), are a segment of the marketplace that represents preferred risk profiles, better credit scores, fewer traffic violations, more insurance products and higher premiums. These less price-sensitive consumers can be attracted through a combination of environmental conditions and capabilities designed.

10. References

- [1] P. S. Bradley and U. M. Fayyad, “Refining Initial Points for K-Means Clustering Bradley and Fayyad Refining Initial Points for K-Means Clustering,” Morgan Kaufmann, 1998.
- [2] J. Han, M. Kamber, and J. Pei, “Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems),” 2011.
- [3] “Clustering by Jain_Dubes.” [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/BOOKS/JAIN/Clustering_Jain_Dubes.pdf. [Accessed: 08-Jan-2020].

11. Annex

A summary of all the applied techniques are displayed on the following tables.

Figure A1. Results comparison for clustering techniques applied to the X_std_df data frame shown in Figure 12, which includes 9 variables. Note: Results for principal components were no re-scaled to avoid misinterpretations of the components.

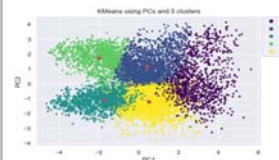
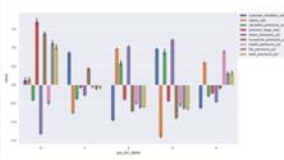
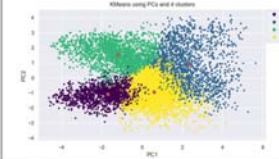
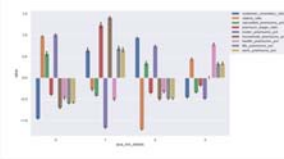
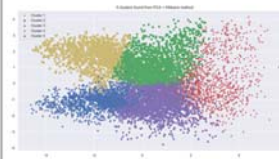
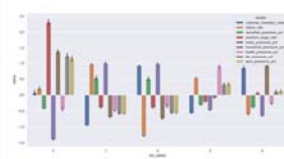
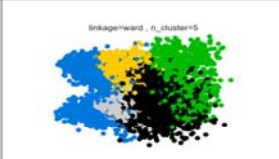
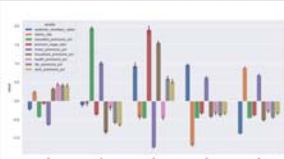
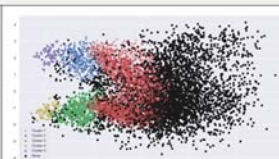
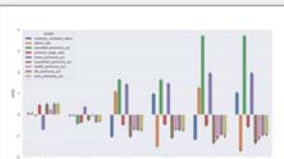
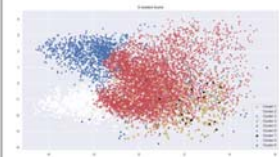
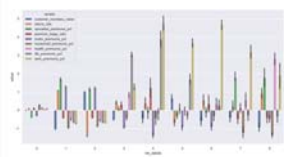
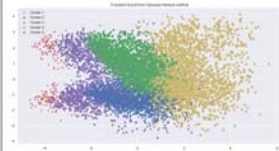
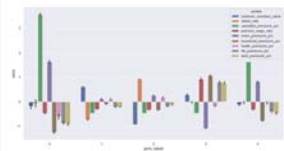
Clustering Technique	Similarity scores	Centroids				Cluster label	Number of customer	Scatter-Plot	Mean values of each variable					
		PC1	PC2	PC3	PC4									
PCA + K-means K=5	0.190	2.817	0.195	0.895	0.712	0	1156							
		0.427	1.155	-0.290	-0.409	1	2216							
		-1.728	-1.071	0.873	0.041	2	1766							
		-1.990	1.742	-0.287	0.445	3	1229							
		0.512	-1.201	-0.627	-0.212	4	2532							
PC1 PC2 PC3 PC4														
PCA + K-means K=4	0.208	-1.690	-1.080	0.843	0.018	0	1831							
		2.322	0.870	0.656	0.283	1	1690							
		-1.206	1.533	-0.364	0.060	2	2246							
		0.604	-0.937	-0.587	-0.207	3	3132							
CMV CR Cancel PWR Motor House Health Life Work														
K-means K=5	0.207	120.04	80%	1%	3%	36%	24%	31%	5%	5%	0	2936		
		436.13	29%	8%	2%	64%	11%	20%	2%	2%	1	2021		
		122.58	79%	2%	3%	25%	20%	28%	14%	13%	2	801		
		-9.78	98%	10%	2%	70%	8%	18%	2%	2%	3	1665		
		397.37	57%	1%	6%	18%	50%	18%	7%	7%	4	1476		
CMV CR Cancel PWR Motor House Health Life Work														
Hierarchical clustering, ward method	0.129	162.73	76%	0%	3%	30%	28%	28%	7%	7%	0	3539		
		188.45	66%	23%	2%	68%	7%	22%	2%	2%	1	1648		
		429.26	55%	0%	7%	16%	50%	19%	8%	7%	2	974		
		436.71	32%	0%	2%	59%	15%	20%	3%	3%	3	1338		
		9.56	95%	0%	2%	61%	13%	21%	3%	3%	4	1400		
CMV CR Cancel PWR Motor House Health Life Work														
DB-Scan	0.067	203.93	67%	0%	2%	53%	17%	23%	3%	3%	0	3779		
		-25.45	101%	20%	2%	77%	3%	17%	1%	1%	1	455		
		438.90	23%	20%	2%	78%	3%	17%	1%	1%	2	382		
		-55.50	106%	40%	2%	89%	-2%	12%	0%	0%	3	65		
		450.55	17%	40%	2%	89%	-2%	12%	0%	0%	4	77		
CMV CR Cancel PWR Motor House Health Life Work														
Mean-Shit	0.143	163.78	74%	0%	2%	46%	20%	26%	4%	4%	0	6619		
		-24.92	101%	23%	2%	75%	4%	18%	2%	1%	1	838		
		421.13	26%	23%	2%	76%	3%	18%	2%	1%	2	1057		
		97.81	78%	20%	3%	31%	-5%	37%	24%	13%	3	256		
		102.64	78%	20%	4%	20%	-3%	19%	29%	35%	4	20		
		426.65	23%	20%	1%	35%	-2%	23%	15%	29%	5	51		
		143.82	70%	20%	3%	30%	0%	28%	5%	37%	6	27		
		88.59	79%	40%	1%	56%	-10%	29%	0%	25%	7	14		
-157.69	123%	40%	3%	34%	-2%	49%	-1%	20%	8	17				
CMV CR Cancel PWR Motor House Health Life Work														
Gaussian-Mixture	0.152	355.26	46%	0%	2%	47%	20%	25%	4%	4%	0	2646		
		0.78	96%	0%	2%	51%	16%	25%	4%	4%	1	2214		
		277.53	67%	0%	5%	20%	41%	22%	9%	9%	2	2319		
		186.49	67%	20%	2%	63%	8%	23%	3%	3%	3	1452		
		172.19	66%	38%	2%	82%	-1%	17%	1%	0%	4	268		

Figure A2. Results comparison for clustering techniques applied to the X_prod_std_df data frame, which contains the subset of “Product” variables: motor_premiums_pct, household_premiums_pct, health_premiums_pct, life_premiums_pct, work_premiums_pct. Note: Results for principal components were not re-scaled to avoid misinterpretations of the components.

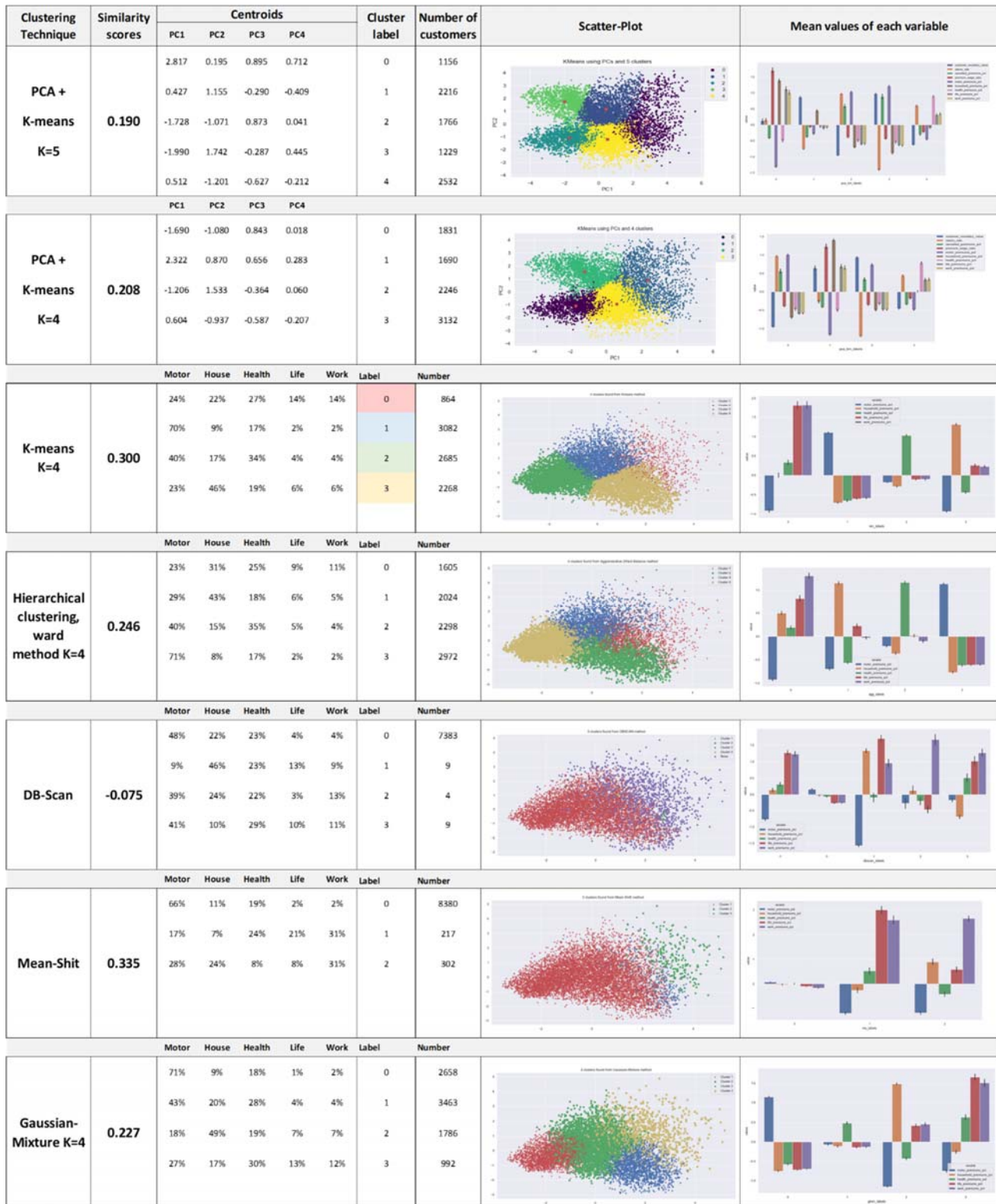


Figure A3. Results comparison for clustering techniques applied to the X_value_std_df data frame, which contains the subset of “Value” variables: cancelled_premiums_pct, claims_rate, customer_monetary_value, premium_wage_ratio. Note: Results for principal components were not re-scaled to avoid misinterpretations of the components.

