# Data Mining - Project Report

## Customer segmentation for an insurance company

Course 2019-2020

---


Group Q

**Students:**

Lennart Dangers - m20190251@novaims.unl.pt

Michael Machatschek - m20190054@novaims.unl.pt

Pedro Santos - m20190420@novaims.unl.pt


---

Github: https://github.com/mrmachatschek/data_mining_project

# Index

# 1. Introduction

Nowadays getting data is not a big issue anymore. On the contrary, due to fewer costs in storage and professional software to collect data, companies and other organizations obtain a huge amount of data every day. That being said, it is vital to obtain the right insights out of the big data. Therefore, it is relevant to apply appropriate techniques to get as much knowledge as possible.

One possible technique to get more insights out of data is cluster analysis. In particular, when dealing with customer data, cluster analysis can help organizations to specialize in their strategies. For instance, marketing campaigns can be more customer-specific without doing individual campaigns. As a result, approaches like this can cause a reduction in (marketing) costs.

## 1.1. Project description

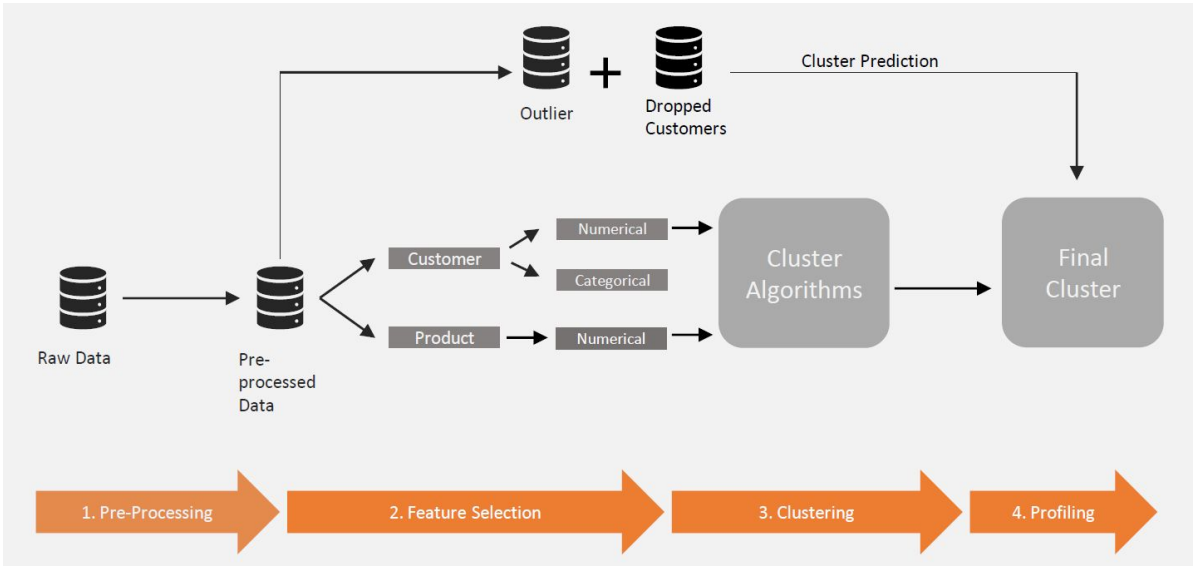
The given dataset contains data of 10,290 customers of an insurance company. Each variable is explained in the data dictionary, which can be found in the appendix. The goal of this project is to do a customer segmentation for the marketing department to get more insights about customer profiles and consequently create more specific marketing campaigns. To achieve this goal, cluster analysis with the appliance of different clustering algorithms followed by profiling is the approach. As a final result, the marketing department should get the profiles of four different customer clusters, which have interpretable characteristics. Furthermore, we give some ideas how these clusters can be targeted by marketing campaigns.

## 1.2. Methodology

Our cluster analysis is based on the typical data mining process, which starts from the raw data and ends with the knowledge gained from a given dataset. Applied on this project, the gained knowledge is the better interpretation and segmentation of the customers. All the necessary substeps are explained in chapter two.

The following graphic gives a brief overview of each process step of the cluster analysis, which contains four basic parts:

1. Data Preprocessing
2. Feature Selection
3. Cluster Algorithms
4. Profiling




Workflow “Cluster Analysis”

To find the best fitting clusters respectively clusters with a meaningful interpretation, we applied several combinations of cluster algorithms and decided for the algorithm, that fitted the best. A measurement for a good working cluster algorithm is its corresponding silhouette score. Even though the highest silhouette score means the best algorithm in terms of statistics, sometimes the algorithm with the second-highest silhouette score has more strength regarding the interpretation of one cluster. That is why we mainly focused on the marketing interpretation of the clusters, while choosing the best fitting algorithm.

Concerning the python code, project steps one to three are represented in the file “main.py” whereas the profiling of the customer is done in “profiling.py”. Furthermore, some side processes are depicted in other classes, which are defined in the table below. The goal of this separation is to have a clearer, more readable and maintainable code.

| File | Function |
|---|---|
| main.py | main file, cluster analysis steps 1-3 |
| profiling.py | cluster analysis step 4 (profiling); contains the final cross table between customer and product clusters |
| helperFunctions.py | contains functions to support the main steps of the cluster analysis, such as creating graphs |
| outlier_variations.py | contains the code to find appropriate thresholds for outliers |
| algorithm_selection.py | contains all approaches for algorithm selection, including the decision criteria |

# 2. Cluster Analysis

## 2.1. Pre-processing

In data mining and nearly every data analysis, data preprocessing is a vital previous step to obtain knowledge through machine learning algorithms. Raw data is usually not clean data, which means that it contains missing values, wrong formats or not understandable column names. Common steps within data preprocessing are data cleaning, the handling of missing data and the treatment of extreme values (outliers).

### 2.1.1. Data Cleaning

Since the customer identity is a unique number, the index of the given dataset is set to the customer identity. Secondly, the columns get more meaningful names to work with and the salary is changed to gross annually salary, which makes a comparison to other salary reports easier. Lastly, the column birth_year is dropped for the cluster analysis, because due to manual editing, it contains many mistakes which can skew the clusters. In terms of interpretation, it is possible to add the birth dates after clustering.

### 2.1.2. Dealing with missing values

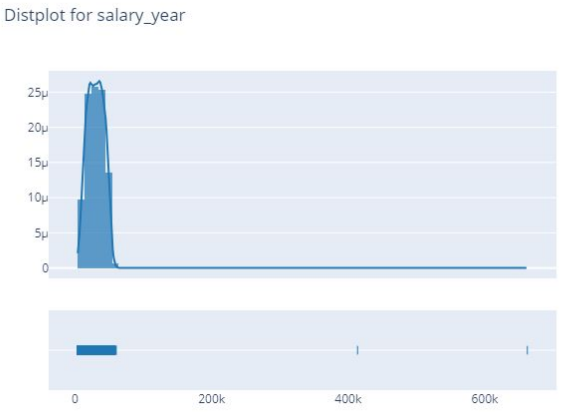
Almost every dataset contains missing values, which are illustrated as so-called “nan”-values in Python (numpy/pandas). In the given data nan-values exist in the columns salary, educ, has_children and in all of the premiums. Since the quota for missing values in salary, educ, and has_children is marginal, the rows can be dropped. However, after obtaining a classification tree for prediction, we will assign these dropped customers a cluster.

Under the assumption that a missing value in premiums means no contract, these values can be replaced by zero. Therefore, differentiation between no contract and a cancelled contract is still possible.

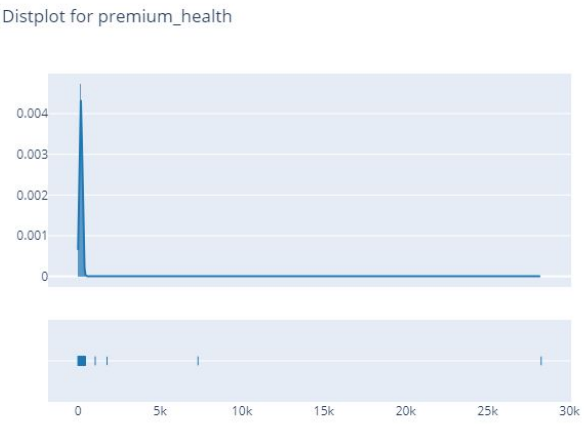
### 2.1.3. Outlier handling

Observations, which are far apart from other values in a random sample are called outliers. The reasons for extreme points are versatile. Firstly, measurements, as well as the data collection, can be incorrect. Furthermore, unusual but real situations can skew data (the Bill Gates effect). Lastly, it is also possible that the original population of these points is different. Outliers can skew the whole dataset and cause wrong insights. Therefore, it may be more appropriate to remove these values for analysis. Nevertheless, it is vital to treat them separately and include these values after doing the cluster analysis.

The given dataset includes points, which can be defined as outliers. A visual approach to detect outliers is a distribution plot (“distplot”) for each feature. The figures below show exemplary a distribution plot for the features annual salary and premium life.

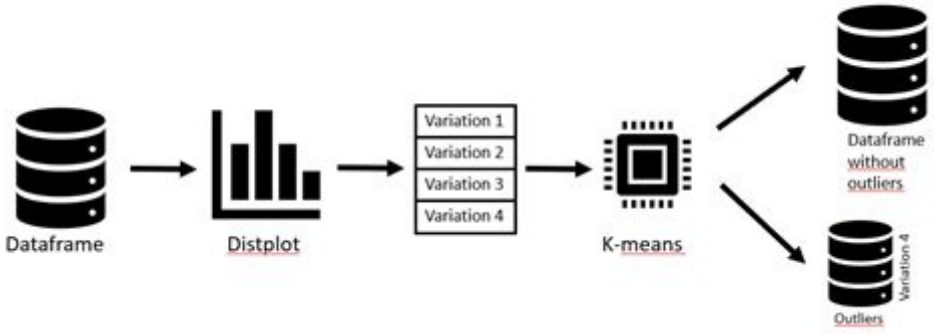



Distribution plot for the annual salary (with outliers)




Distribution plot for the premium in health (with outliers)

Because of these plots, outliers can be easily seen. For instance, the plot for the annual salary visualizes points above 400,000 Euro, which are far apart from the mean.




Process outlier treatment

The figure above visualizes the process of our outlier treatment (compare file outlier_variations.py). After plotting all feature distributions, four different variations were built. Each variation includes different thresholds for each of the variables, that cut off outliers (see table below).

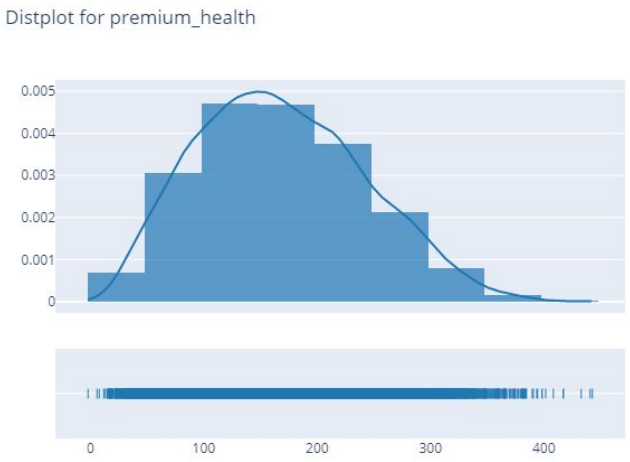
To find the best thresholds, we applied the new dataframe of each variation in a test scenario with k-means and compared the silhouette score, the interpretability and size of the clusters. The main goal is to choose a specific variation, which has a good silhouette score without skewing the interpretation of a cluster.

| | Variation 1 | Variation 2 | Variation 3 | Variation 4 |
|---|---|---|---|---|
| **salary_year** | 0 | 200,000 | 200,000 | 200,000 |
| **mon_value** | 0 | -50,000 | -1,000 | -1,000 |
| **claims_rate** | 0 | 20 | 8 | 3 |
| **premium_motor** | 0 | 2,000 | 600 | 600 |
| **premium_household** | 0 | 3,000 | 1,600 | 1,600 |
| **premium_health** | 0 | 5,000 | 400 | 400 |
| **premium_life** | 0 | 400 | 300 | 300 |
| **premium_work_comp** | 0 | 500 | 300 | 300 |
| **Silhouette Score** | 0.469 | 0.312 | 0.321 | 0.323 |
| **Interpretability** | **--** | **-** | **o** | **++** |
| **Outlier Size** | **0** | **21** | **74** | **159** |

Even though variation 1 has the highest silhouette score, we choose variation 4 for further analysis, because the focus of the analysis is on the interpretability of the cluster. Since variation 1 includes all extreme points each computed cluster has no interpretable centroids anymore. Moreover, the interpretability of variations 4 stands out from variations 2 and 3.

After extracting the outliers from the dataframe, the new distribution plots show a more normal distribution and a higher density. Further distribution plots (including and excluding outliers) can be found in the appendix.


Distribution plot for the annual salary (without outliers)




Distribution plot for the premium in health (without outliers)

## 2.2. Feature Selection

Feature selection is the way to choose a subset of variables of a given dataset to optimize the prediction or output of a specific analyze. Alongside choosing appropriate features can also mean to drop irrelevant ones in order to optimize the output. Further reasons for using feature selection are for example to avoid the curse of dimensionality or to make interpretations easier to understand.

### 2.2.1. Feature engineering

It can be rather difficult to train a model without having relevant features. Most likely, raw data does not show all the relevant information which can be extracted from the dataset. Therefore, it is vital to extract and create new features out of the raw dataset. This process is called feature engineering.

In the given dataset we created the following new features:

**Total amount of premiums**

The total amount of premiums is a new feature, that is the sum of all premiums for each customer. For interpretation purposes, this features gives us a sense of how risk-averse this customer is. A separation within premiums is only necessary when dealing with product clusters.

**Cancelled contracts**

This variable is a dummy variable which means that values can be either one or zero. If any of the contracts were cancelled recently (represented by negative values in the premiums), the value will be one. Otherwise, it contains zero. With this feature, the marketing department can get valuable insights about customers, who may have had a bad experience with our insurance, found a cheaper insurance or have other reasons for cancelling contracts with us. This information can be used by our marketing team to target customers, which are currently more prone to cancelling contracts, with advantages of our insurance or special offers.

**Has all contracts**

This dummy variable displays a one if a customer has all contracts possible and zero if not. Information like this can help to split customers into customers, who already have all possible contracts and those who still have the potential to sign more contracts.

**Is a profitable customer**

A profitable customer (equals one) has a positive monetary value. We choose this new feature to stand out these customers.

### 2.2.2. Feature splitting

Often the term of feature splitting is interchangeably used with the separation of the dataset into training and test sets. In our case, we use the term feature splitting at first to separate variables, which belongs to either the customer or the product. This process step aims to be able to apply different algorithms in a more specific way. For instance, the k-means algorithm is more useful when applying to numerical data.

Firstly, the features in this cluster analysis are split into customer-related and product-related features. Secondly, a further definition of numerical and categorical

variables is done. To have clearer clusters, the variable first policy is not used in the cluster analysis.

As you will see in the following parts of this report, we use the categorical features only for descriptive purposes. That is why we use in the beginning as many features as possible and decide in the profiling section, which of of them are useful for interpretation.

Regarding the selection of the numerical features for our cluster analysis, we had the following idea. We chose salary because it is very important to know whether our customers is able to pay for more contracts or not. Even though, monetary value, claims rate and total premium bear similar information, we decided to include all of them and not creating any ratios. The reason is the specific interpretation of these features, which can be very useful for marketing campaigns. The monetary value gives us information about the long term profitability of this customer. The claims rate gives us an indicator about whether the customer is careful or not in the recent years. Lastly, the total premium provides us an indicator about whether a customer is risk-averse or risk-friendly. As we will see in the profiling section, this separation was useful for interpreting our clusters.

| **Relation** | **Type** | **Features** |
|---|---|---|
| Customer | numerical | salary, mon_value, claims_rate, premium_total |
| Customer | categorical | location, has_children, educ, cancelled_contracts, has_all, is_profit |
| Product | numerical | premium_motor, premium_household, premium_health, premium_life, premium_work_comp |

### 2.2.3. Final Data dictionary

The following table visualizes and explains all features after preprocessing, data cleaning and feature selection. These are the final features, which are used to apply different algorithms.

| **Variable** | **Description** |
|---|---|
| **Customer Identity(Index)** | The identity of each customer; unique number (only used for identification) |
| **salary_year** | Gross annual salary |
| **mon_value** | Customer monetary lifetime value; mon_value = annual profit x number of years – acquisition cost |
| **claims_rate** | contains data of the last two years; claims_rate = Amount paid by the insurance company / Premiums |
| **premium_total** | sum of all premiums of a customer |
| **educ** | highest academic degree |
| **location** | Living area codes; no further information provided |
| **has_children** | dummy variable; 1 = customer has children, 0 = no children |
| **cancelled_contracts** | dummy variable, if a customer has recently cancelled contracts |
| **has_all** | dummy variable, if a customer has all possible contracts |
| **is_profit** | dummy variable, if the monetary value is positive |
| **premium_motor** | Premiums paid by the customer for motor insurance |

| | |
|---|---|
| **premium_household** | Premiums paid by the customer for household insurance |
| **premium_health** | Premiums paid by the customer for health insurance |
| **premium_life** | Premiums paid by the customer for life insurance |
| **premium_work_comp** | Premiums paid by the customer for work compensation insurance |

# 2.3 Choosing the right algorithm

In this section of the report we describe the process of finding the right algorithm and the relevant corresponding parameters. We begin by giving a short theoretical introduction of the algorithms we used. The introduction is followed by an evaluation of the algorithms, which is split into customer clusters and product clusters. In the last part of this section we explain how we handled the customers with no assigned cluster.

## 2.3.1 Theoretical Introduction

To find the best clusters we applied several algorithms on our dataset. In the following section we provide a definition and short theoretical explanation of the used algorithms, to make it more comprehensible why we choose our final algorithms.

### K-means and modified versions

K-means is a prototype-based and partitional technique that tries to find a user-specified number of clusters (K) by minimizing the distance between the observations within a cluster and maximizing the distance between the clusters. Each cluster has a centroid, that is used to interpret the cluster. The algorithm requires all variables to be continuous.

K-modes is the equivalent to K-means for categorical variables. Instead of using the mean of the variables to measure the similarity of the observations, this algorithm uses the mode of the values. The K-prototype algorithm gives a solution for datasets with numerical and categorical variables by combining attributes of K-means and K-modes.

Important advantages of K-means (and mostly also for K-modes and K-prototypes) are that it is relatively easy to implement, it scales to large data sets, it guarantees convergence and it easily adapts to new examples. Disadvantages are that the user has to choose the number of cluster manually, the algorithm could have problems with clustering data of varying sizes and density and the technique is prone to outliers.

### Agglomerative Hierarchical Clustering

This clustering approach attempts to group observations by starting with each object as a single cluster and then repeatedly pairing the two closest clusters together until all clusters have been merged into one big cluster containing all objects. The result is a tree based representation of the cluster, also called Dendogram.

The main advantages of this technique are that we do not specify the number of clusters, it is easy to implement and easy to comprehend and with the help of the Dendogram we get

a tool to understand our data better. In contrast, the drawbacks of this approach are firstly that it is a greedy algorithm, which means that we can not undo a previous step even though it may not be the optimal decision. Secondly, the time complexity is relatively high with a best case scenario of $O(n^2)$.

We used this technique only after applying K-means with a large number of clusters. This combination allowed us to make us of some synergy effects. K-means is a quite efficient algorithm, whereas agglomerative hierarchical clustering (AHC) is not. If we use the cluster produced by K-means as the input for the AHC, we can reduce the computation time of AHC. Furthermore, K-means is designed to find homogeneous spherically-shaped clusters and the AHC identifies clusters in a tree-like structure. The hybrid of both can give us a combination of both structures.

#### Self organizing maps

A self organizing map (SOM) is a type of artificial neural network (ANN) that can be used for clustering. In contrast to other ANNs the SOM applies competitive learning. This means that this technique uses a neighborhood function to preserve the topology of the input space. As with the AHC we use SOM only in combination with other algorithms. One approach was to first run SOM with a fairly high number of nodes and then use K-means to find clusters in the output net. The second approach was to run AHC after SOM.

#### Evaluation of the results

We evaluated the results by statistical measures and business reasons. The latter was the more important factor in our analysis. As a statistical measure we used the average silhouette score, which shows how similar an object is to its own cluster (cohesion) and how different it is from the other cluster (separation). The value ranges from -1 to 1, where a high positive value shows similarity to its own cluster and dissimilarity to the other clusters. Regarding the business reasons, we mainly focused on the marketing interpretation of the resulting clusters. In the profiling section of this report, we explain in more detail how we carried out this decision process.

Before starting with the algorithm selection we had to choose a similarity measure. There are several measures of similarity available e.g. Euclidean distance, Manhatten distance or Cosine similarity. In our analysis we used the euclidean distance as a similarity measure, as this is usually the default method. Furthermore, we are dealing with quite dense data, which is a preferable condition for the euclidean distance.

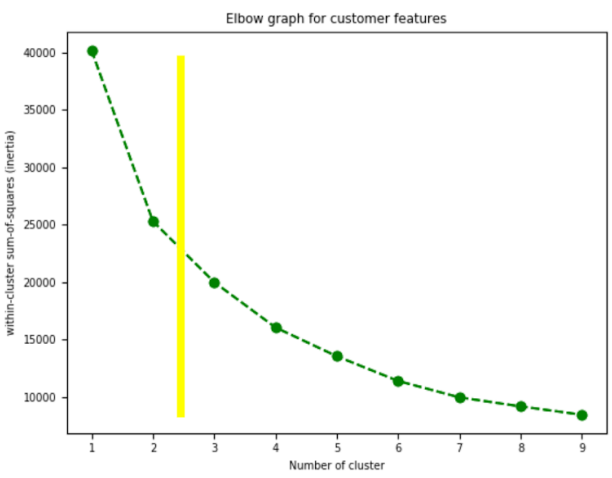
## 2.3.2 Applying the clustering algorithms

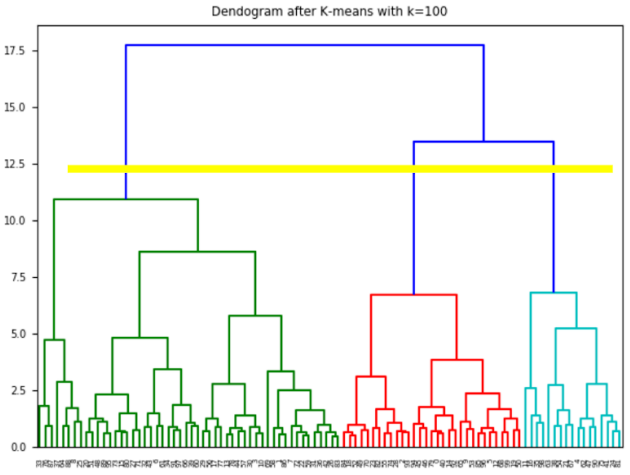
### 2.3.2.1 Customer related features

#### Numerical variables

In the beginning of this section, we want to illustrate how we chose the number of clusters.
To help us find a suitable number of clusters we created an Elbow graph with a simple K-means algorithm and a Dendogram with Agglomerative Clustering after K-means with

100 clusters. The Elbow graph shows the sum of squared errors for different number of clusters. It is good practice to choose the number of clusters right there where the “elbow” is. In our case this would be two clusters. However, for marketing reason we think it is more useful if we have a slightly higher number of clusters to not lose too much information about our customers. The Dendogram visualizes the clustering process carried out by Agglomerative Hierarchical algorithm. The height of the branches represents the distance between the two connected clusters. In this visual it is a good practice to choose the number of clusters by drawing a horizontal line which separates the tree with the biggest height. In our case this would result in two or three clusters.




After looking at these two graphs and experimenting with different number of clusters, we came to the decision that for the customer related numeric variables three clusters are our best solution.

## Applying the algorithms

The following table shows the algorithms which we applied to our dataset. We included the centroids, the silhouette score and the size of the clusters for a first evaluation.

| **Algorithm** | **Centroids**<br>(salary, mon_value, claims_rate, premium_total) | **Silhouette Score** | **Cluster sizes** |
|---|---|---|---|
| K-means | 0: (32783; 399; 0.38; 700)<br>1: (18697; 504; 0.52; 1116)<br>2: (30704; 20; 0.93; 712) | 0.36 | 0: 0.38<br>1: 0.12<br>2: 0.5 |
| K-means → AHC | 0: (28306; 42; 0.91; 744)<br>1: (23623; 717; 0.30; 1067)<br>2: (34264; 374; 0.40; 697) | 0.30 | 0: 0.55<br>1: 0.08<br>2: 0.37 |
| SOM → K-means | 0: (30398; 51; 0.89; 700)<br>1: (34311; 347; 0.44; 685)<br>2: (18427; 341; 0.58; 879) | 0.34 | 0: 0.44<br>1: 0.35<br>2: 0.21 |
| SOM → AHC | 0: (29215; -71; 1.1; 763)<br>1: (34247; 378; 0.4; 698)<br>2: (19191; 645; 0.4; 1040) | 0.33 | 0: 0.50<br>1: 0.39<br>2: 0.11 |

AHC = Agglomerative Hierarchical Clustering | SOM = Self Organizing Map

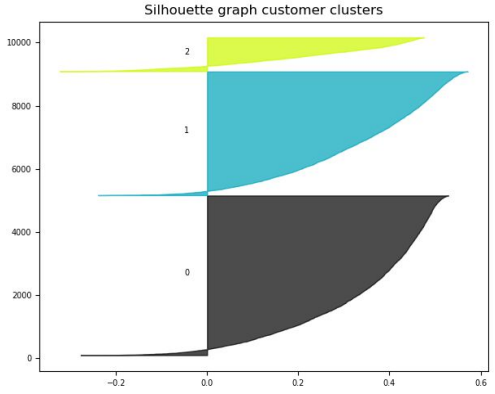
In addition to these algorithms we also applied the DBSCAN and Mean Shift technique. Both results were not nearly satisfying and we excluded them for further investigation. One reason for these unfavourable results can be that the DBSCAN and Mean Shift are good at separating clusters with high density from clusters with low density but not at separating clusters with similar densities.

While applying the algorithms displayed in the table we identified our first possible clusters. We inferred preferable customer profiles from these clusters to evaluate the applied techniques. This enabled us to give our marketing decision process a more objective foundation. In the following table you see the three customer profiles we created and the evaluation of the results of each algorithm. The evaluation focused on the characteristics of the customer, which were satisfied by the clusters or not.

| **Aimed Customer profile** | **K-means** | **K-means → AHC** | **SOM → K-means** | **SOM → AHC** |
|---|---|---|---|---|
| Careful and risk-averse: People with low salary, who spend a lot for their insurance (risk-averse) but still are careful (low claim rate) | *Focus on risk-averse aspect, but a higher claims rate.* | *Salary aspect not satisfied.* | *Careful aspect not satisfied.* | *All aspects satisfied, no special focus.* |
| Careless and quite wealthy: People with a medium to high salary, who spend an average amount for their insurance and are not careful (claim a lot) | *All aspects satisfied, no special focus.* | *All aspects satisfied, no special focus.* | *All aspects satisfied, no special focus.* | *All aspects satisfied, focus on careless.* |
| Careful and wealthy: People with high salary, which spend an average amount for their insurance and are careful (low claim rate).. | *All aspects satisfied, special focus on carefulness.* | *All aspects satisfied, special focus on salary.* | *All aspects satisfied, special focus on salary.* | *All aspects satisfied, special focus on salary.* |

The final approach we chose for the numerical customer variables was the **Self Organizing Map followed by Agglomerative Hierarchical Clustering**. This is mainly because the resulting clusters of this technique were in our opinion best suited for an effective marketing campaign. It satisfied all aspects of our predefined customer profiles and it positioned our most favourable customer cluster, the “careful and risk-averse”, quite clearly.

Even though the silhouette score was only on the third rank, the differences to the better options are still relatively marginal. The following graphic on the left shows the silhouette graph that plots the silhouette score of each observation within its corresponding cluster. As you can see only a few observations have a negative silhouette score.



### Categorical variables

We decided to not use any clustering technique on the categorical variables but instead use the individual variables to describe our final clusters. This is because, first of all, we did not get satisfying results from our two clustering approaches, i.e. there was no meaningful interpretation in the clusters. Secondly, we did not want to add misleading information to our final cluster, e.g. only because most of the customers within a cluster have children, we should not assume in our marketing campaign that all customers of this clusters have children.

We applied the K-modes and K-prototype technique for these variables. The k-modes approach lead us to the following clusters:

| Clusters | location | has_children | education | cancelled_contracts | has_all | is_profit |
|---|---|---|---|---|---|---|
| 0 | 4 | 1 | BSc/MSc | 0 | 1 | 1 |
| 1 | 1 | 0 | High School | 0 | 1 | 0 |
| 2 | 4 | 1 | Bsc/MSc | 1 | 0 | 1 |
| 3 | 3 | 0 | High School | 0 | 1 | 1 |

As you can see in the table, the clusters are not easy to interpret and sometimes the information is not very useful, like described above with the feature “has_children”.

Another algorithm we used was K-prototype. This technique allowed us to create clusters with our numerical and categorical clusters together. Even though the numerical features were good to interpret, the categorical clusters were not useful.

**Categorical cluster from K-prototype**

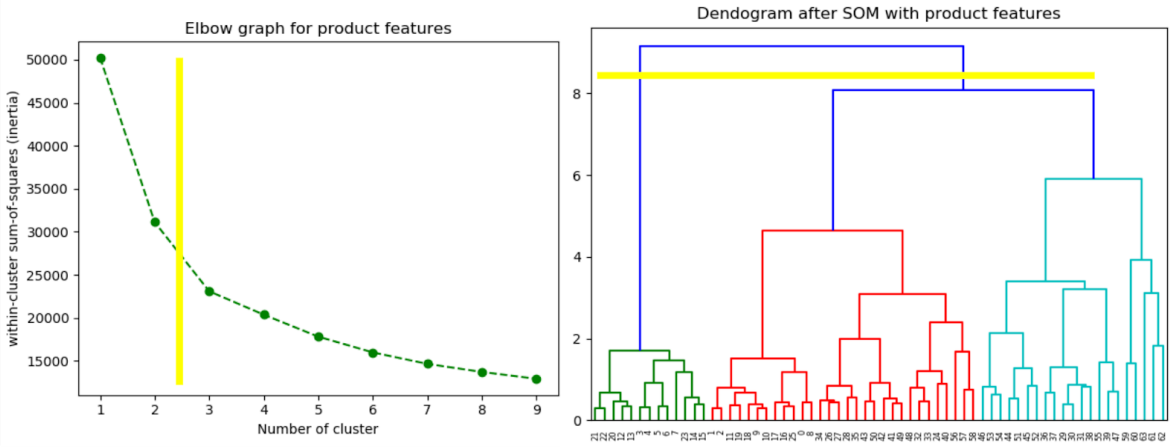
| Clusters | location | has_children | education | cancelled_contracts | has_all | is_profit |
|---|---|---|---|---|---|---|
| 0 | 4 | 1 | BSc/MSc | 0 | 1 | 1 |
| 1 | 4 | 1 | High School | 0 | 1 | 1 |
| 2 | 4 | 1 | Bsc/MSc | 0 | 1 | 0 |

**Numerical cluster from K-prototype**

| Clusters | salary_year | mon_value | claims_rate | premium_total |
|---|---|---|---|---|
| 0 | 32902 | 386 | 0.4 | 701 |
| 1 | 18658 | 467 | 0.55 | 1093 |
| 2 | 30908 | 7 | 0.96 | 704 |

### 2.3.2.2 Product related features

For the product related features we decided us for two clusters. On the one hand, we got this insight from the Elbowgraph and the Dendogram, which suggests two to three clusters. On the other hand, we wanted to keep the number of dimensions as low as possible for our subsequent merge with the customer clusters.




Like for our customer features, we applied K-means, K-means followed by AHC, SOM followed by K-means, SOM followed by AHC, DBSCAN and Mean Shift. The following table shows the two techniques we investigated further. As the clusters seem to be very well separated with our first applied algorithms, we only tried two different approaches.

| **Algorithm** | **Centroids**<br>(motor, household, health, life, work_compensation) | **Silhouette Score** | **Cluster sizes** |
|---|---|---|---|
| K-means | 0: (400; 87; 133; 17, 17)<br>1: (172; 349; 210; 68; 67) | 0.36 | 0: 0.55<br>1: 0.45 |
| SOM → AHC | 0: (248; 257; 191; 51; 45)<br>1: (419; 98; 116; 18, 15) | 0.28 | 0: 0.67<br>1: 0.33 |

We stuck to our first approach, the K-means version. Firstly, the silhouette score was better than the one from SOM and AHC. Secondly, the cluster sizes were more equally distributed. Lastly, the cluster were clearer to interpret. With K-means we have one cluster with a strong focus on motor insurances and one cluster with a quite strong focus on household. This was more preferable than two cluster with a high focus on motor.

## 2.3.3 Predict clusters of dropped customers

In the data preprocessing part of our analysis we dropped a few outliers and rows with missing values from our main dataframe. To make use of all of our customers in the marketing campaign, we need to assign the dropped customers a cluster. We can do that by training a classifier with our detected clusters and the corresponding features. This part of machine learning is called supervised learning as we give the algorithm a label that should be predicted by the given input features. In our case the cluster is the label and the variables are the input features. We decided to use a classification tree for predicting the cluster of a customer. As some of the customers had missing values, we trained a model

for each combination of input features. Another useful insight that can be obtained by training a classification tree with the clusters, is the visualization of the importance of our features. In the appendix of this report, you will find a graphic with the visualization of our classification tree.

## 2.4. Profiling

In this section, we explain how we worked with both product and customer clusters and joined them in order to summarize the maximum amount of information in a few and clear amount of groups of customers.

### 2.4.1 Customer Cluster

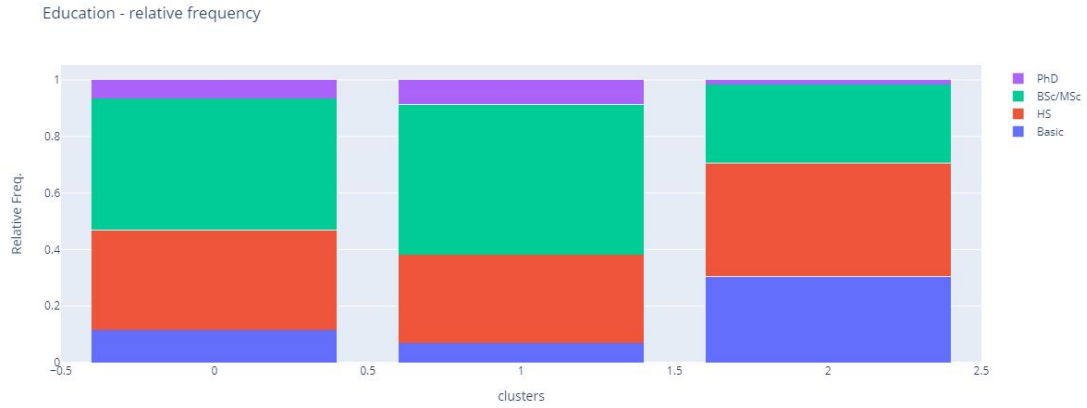
#### Categorical features

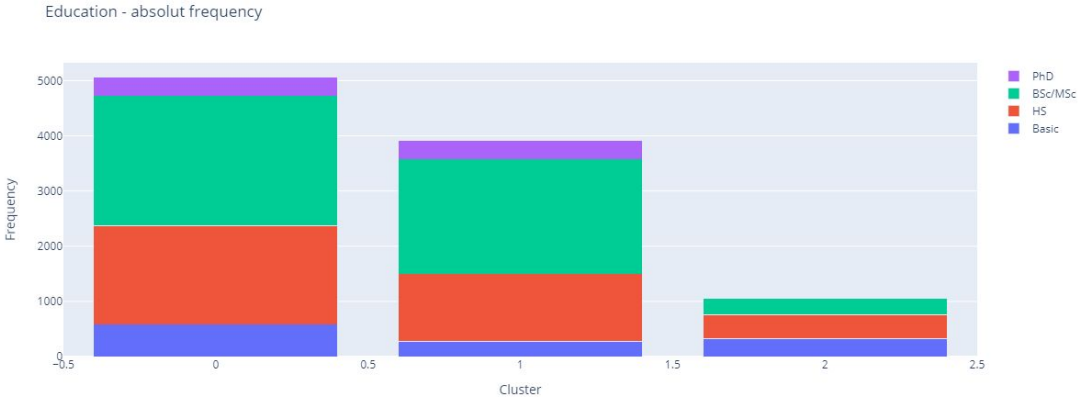
Through a business perspective, the AHC over the SOM produced the better clusters results, as written previously. This is because we were able to obtain three clearly different clusters.
To further distinguish our customers, we tested the different categorical features individually and see how the new clusters would perform. Surprisingly, most features were not helpful to improve the interpretation of our clusters. If we plot the frequency distribution of our clusters against each categorical feature, we observe that:

**- Location** is a variable numbered from 1 to 4. It has an even distribution within all clusters, with similar proportions for each location. This doesn't allow us to take any particular conclusion. Therefore, we deemed this variable as not relevant.

**- Has_Children** is a dummy variable where customers that have children are represented by a 1. On our clusters, the majority of clients (70% or higher) have children, therefore, there is no key takeaway from this variable.

**- Education**, which is split between 0) Basic, 1) High School, 2) Bachelors and Masters and 3) PhD. Our first two clusters have a similar distribution, which, again, means that these two types of customers are not so different in terms of education. The third cluster has a slightly different distribution, but it is also the smallest cluster. Therefore, splitting our clusters based on education would increase the levels of complexity in our analysis instead of simplifying it, which is our goal.

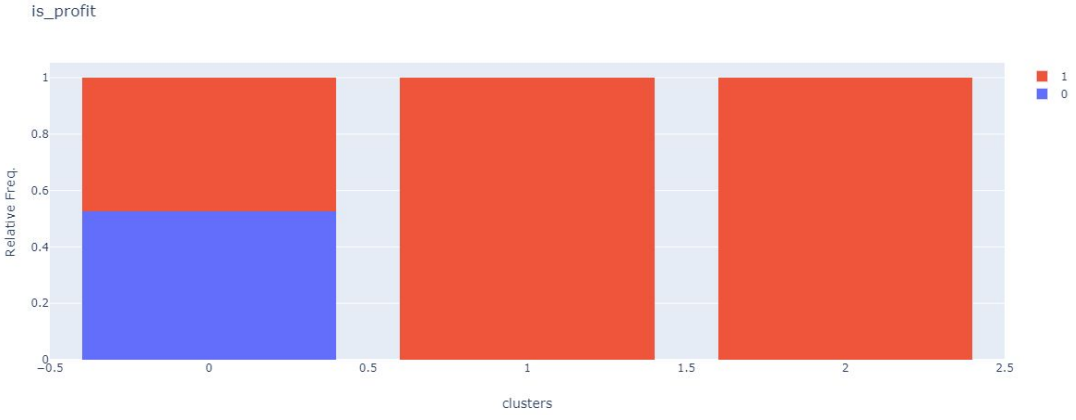



**- Has_all:** The number of contracts each customer has, ie, if the customer has all contracts then 1, else 0. Most of our customers (around 80%) have all contracts as well, therefore, this variable is not useful for our clustering solution.

**- Cancelled_Contracts:** a dummy that states if a client has a cancelled contract or not. If so, then 1. We classify a cancelled contract whenever the value in a premium is negative. This variable has a similar behavior as Has_all and, therefore, we did not find the results interesting enough for the final clustering solution.

**- Is_Profit:** a dummy that returns 1 if the client is profitable and 0 if not. This variable has an interesting distribution as it splits our biggest cluster in two while keeping the other clusters the same. This variable can lead to interesting conclusions and we will use it to further split our clusters.



Our numerical features will be only affected by this categorical feature. We will talk about this change soon.
Note that even though most categorical features didn't serve a clear purpose for our clustering solution, it does not mean that the variables are not useful. One can perform a specific marketing campaign for customers with specific characteristics. As an example, if the insurance company wants to increase the number of contracts, they can specifically look for people who do not have all contracts and try to market them. Another example could be targeting the customers who have cancelled contracts and try to understand what lead them to cancel them.

### Numerical features

Our numerical features describe better the different types of customers. As written previously, our centroids describe three different customers: careless and wealthy (0), careful and wealthy (1), and careful and risk-averse (2).

| Cluster | Salary Year | Mon. Value | Claims Rate | Premium Total | Frequency |
|---|---|---|---|---|---|
| **0** | 29.214,6 | -71,3 | 1,1 | 762,9 | 5218 |
| **1** | 34.246,7 | 378,4 | 0,4 | 698,7 | 3940 |
| **2** | 19.191,1 | 645,5 | 0,4 | 1.040,7 | 1095 |

We can also observe that the cluster 0 is the largest while having a non-profitable centroid, which is an undesirable result. Furthermore, from what we have seen above, almost 50% of the customers of this cluster are profitable. Clients with different profitability should be handled differently, as we can target them with different goals. It could be a business strategy to minimize the non-profitable clients in order to maximize profits, and, this way, it is important to have a distinction between them. It could also be a strategy to focus on improving the low-profit clients and turn them more profitable to balance with the non-profitable clients, instead of risking further marketing investment on a group of clients that may not pay off. Therefore, we split the clusters bearing in mind their profitability and recalculated the centroids.

| Cluster | Salary Year | Mon. Value | Claims Rate | Premium Total | Is Profitable | Frequency |
|---|---|---|---|---|---|---|
| **0** | 28.351,4 | 92,1 | 0,84 | 803,4 | 1 | 2471 |
| **1** | 34.246,7 | 378,4 | 0,4 | 698,7 | 1 | 3940 |
| **2** | 19.191,1 | 645,5 | 0,4 | 1.040,7 | 1 | 1095 |
| **3** | 29.991,2 | -218,3 | 1,3 | 726,3 | 0 | 2747 |

The split results in more balanced clusters and improved readability and analysis capability. The new clusters (0 and 3) represent customers that are wealthy, with a median premium total but have different financial impacts. These centroids represent our final customer cluster.

### 2.4.2 Product Cluster

Our product cluster solution is only composed of two clusters. Not only this helps us create the final clustering solution, but also maximizes the readability of the centroids.

| Cluster | Motor | Household | Health | Life | Work | Frequency |
|---|---|---|---|---|---|---|
| **0** | 407,1 | 93,7 | 135,4 | 17,5 | 17,9 | 5618 |
| **1** | 169,6 | 352,8 | 214,3 | 70,6 | 69,0 | 4635 |

As of now, we have two clear types of customers, those who spend a lot in Motor insurance and moderate to low on everything else and those who spend a lot on Household and Health and moderate to high on everything else, whereas with 3 clusters we would have an intermediate group, which is not helpful in our analysis as it would eventually be merged into the other clusters. It is also important to note that the larger the amount of clusters, the more difficult and expensive it is to target and segment them for marketing purposes. Distribution-wise, these clusters also make sense since the two groups are similar sized.

### 2.4.3 Final Cluster

It is through the combination of the product and customer clusters that we can obtain the final solution.

| Customer/Product Cluster | 0 | 1 |
|---|---|---|
| **0** | 1053 | 1418 |
| **1** | 2710 | 1230 |
| **2** | 141 | 954 |
| **3** | 1714 | 1033 |

By observing the table above, one can see that there is no clear match between the clusters, with the exception of customer cluster 2, which is composed by risk-averse clients with low income but high profitability, where about 90% of the distribution belong to the product cluster 1, represented by high household and health insurances.
Although this solution provides an accurate representation of the database, a high number of clusters is not ideal as it is necessary for several different marketing campaigns to reach all groups so, in order to minimize the amount of clusters, we are moving the customers within the less relevant clusters according to:
1) their probability of being in another cluster or
2) assigning them to the cluster which makes the most sense.

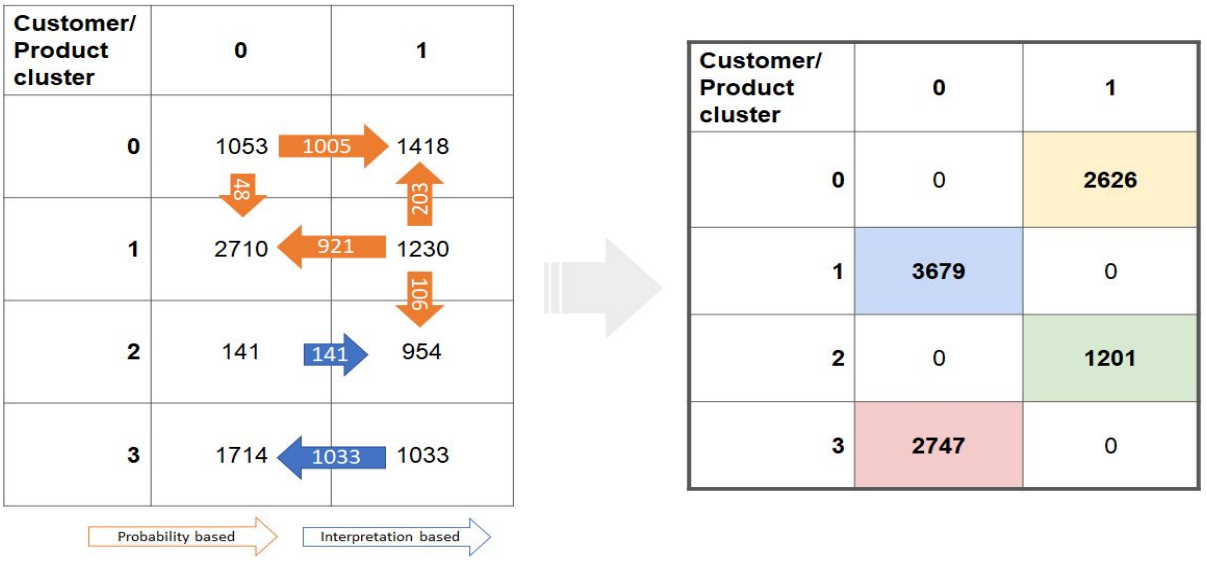
Our goal is to make the moves that allow us to keep the maximum business interpretation possible. With this in mind, we have decided to move the clusters with less frequency:

- Cluster Combination [0,0] and [1,1] based on their probability of belonging to another cluster.

- Cluster Combination [2,0] and [3,1] to their opposite product cluster. We decided this way, because the customers in customer cluster 2 are the ones with the highest profitability and we want to keep them together. With the same concept in mind, we also want to keep the non-profitable customers together.

In order to distribute them according to probability, we plotted the normal distributions based on the centroids we want to keep and calculated the probability of each customer to belong to those centroids.

The following graphic shows how many customers were switched to each of the final cluster.




| Customer/ Product cluster | 0 | 1 |
|---|---|---|
| 0 | 0 | **2626** |
| 1 | **3679** | 0 |
| 2 | 0 | **1201** |
| 3 | **2747** | 0 |

**Finally, we obtain the following table, with the recalculated centroids:**

| Cluster | Salary Year | Mon Value | Claims Rate | Premium Total | Health | Household | Life | Motor | Work | Frequency |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 29328 | 96 | 0.84 | 790 | 204 | 227 | 49 | 259 | 48 | 2626 |
| **1** | 33421 | 379 | 0.40 | 695 | 155 | 128 | 28 | 351 | 27 | 3679 |
| **2** | 21674 | 646 | 0.36 | 1033 | 174 | 527 | 71 | 184 | 74 | 1201 |
| **3** | 29991 | -218 | 1.28 | 726 | 158 | 167 | 37 | 319 | 37 | 2747 |

Red: Low value ; Blue: Medium value; Green: High value

## Description of final cluster and their marketing approach

This final description summarizes how our database looks like and what kind of customers we have, bearing in mind the characteristics specified previously.

0: Careless and quite Wealthy. Balanced approach to insurance types

2626 customers are characterized by being wealthy, spending averagely on premiums and having a high claims rate. Nevertheless, these clients are profitable. They also consume the most of Health insurances, while being moderate spenders on other insurances. While these clients represent an important group of our database, they are not very interesting in a financial point of view, as their profits are low. A possible marketing approach would be to increase their premiums' costs in order to boost profits as these clients seem careless, i.e., they claim a lot, therefore, they will always try to be insured, which means that they should have low risk of leaving the company.

1: Careful and Wealthy. Motor enthusiasts

3679 customers belong to the wealthy and careful group, as their claims rate are moderate, and they do not spend a lot in premiums. This type of customers is profitable and have a preference on Motor insurance. These customers tend to be the richest in our database and seem to have great potential to be explored. On the other hand, they tend to not value insurances as much, since most of their insurances, apart from Motor, tend to have low values. Since this is the biggest group, it would be interesting to promote the general importance of insurances, in order to try to gain more valuable contracts.

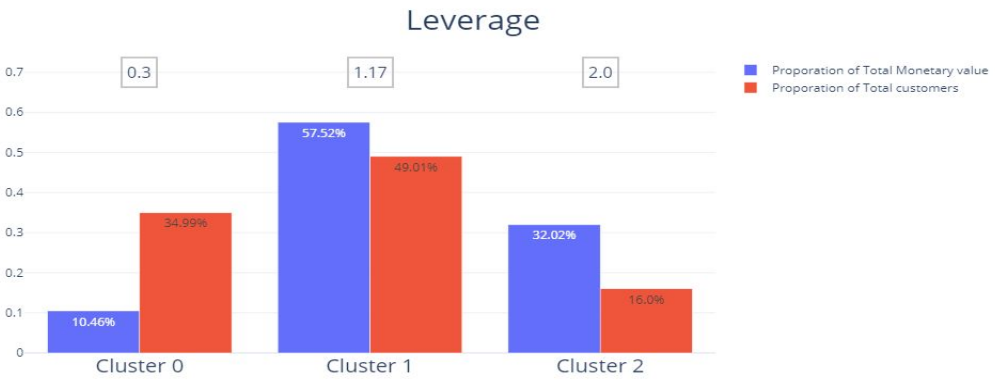
2: Careful and Risk-Averse. Household

1201 customers are defined by being careful and risk-averse as well as having low income. These customers are the most profitable type in our database and they have a clear preference on Household insurances. They are also the highest spenders on Life and Work and have the lowest interest in Motor insurance, which supports their risk-averse characteristic. This group represents the best customers for the company, as they tend to have high premiums paired with a low claims rate. Business wise, keeping these customers is the most important strategy, as it is not easy to further improve this group. One marketing approach to reach this goal would be to reward these customers with special offers or bonuses.

3: Careless and quite Wealthy. Non-profitable customers

Finally, 2747 customers belong to the last group, wealthy and careless, with negative profitability. These customers tend to prefer Motor insurance, although they consume the remaining moderately. This group is rather problematic and represents a challenge to the company due to their high claims rate. An increase in premium could work, since most of these clients seem dependent on insurances (due to their high claim rate). It could also be a strategy to minimize the investment on these customers and either phase them out of the company or wait for them to turn profitable and reclassify them. The trade-off here would be investing on a group that represents a loss of money *vs* accepting the losses and focusing on clients who are profitable.

**Leverage of clusters**




Last but not least, we want to add some information about the leverage of each of the final cluster. On the graphic above, one can observe the relative importance of each cluster to the profitability of the company. Only the clusters whose monetary value is positive are represented. We can see that, while the cluster 0 is a big part of our database, they barely represent any income to the company (leverage of 0.3), while the opposite happens in cluster 2, with a leverage ratio of 2. Our biggest cluster also represents more than 50% of profitability, with a leverage of 1.2. It is also important to note that the missing cluster has a similar size to the cluster 0 while having losses, which, as written before, is a problem that must be addressed.

## 3. Conclusion

In a cluster analysis with a business context, it is usually challenging to find the right balance between statistical accuracy and a meaningful interpretation of the solution. That is why it can be more appropriate to decide against accuracy to obtain a more interpretable result. The goal of this analysis was to cluster the customer in the database to support the marketing department regarding more custom-built strategies.

To get there, after data preprocessing, we separated the features into product and customer-related features and applied several clustering (unsupervised) algorithms. Furthermore, we measured and evaluated the corresponding outcome and their usability for marketing purposes. While taking also the categorical features into account, we segmented the customers into four final clusters:

Cluster 0: Careless and quite wealthy customers
Cluster 1: Careful and wealthy as well as motor enthusiasts
Cluster 2: Careful and risk-averse especially in household
Cluster 3: Careless and quite wealthy, but not profitable

Due to these clearly separated customer clusters, the marketing department should now be able to roll out more specific marketing campaigns.

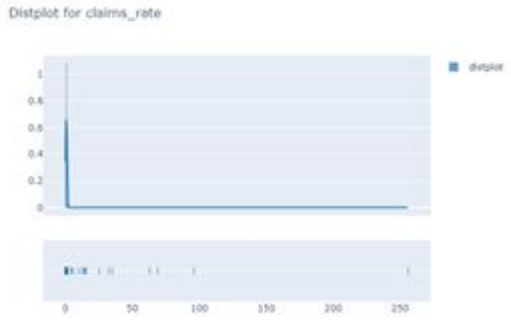
# Appendix

## List of annexes

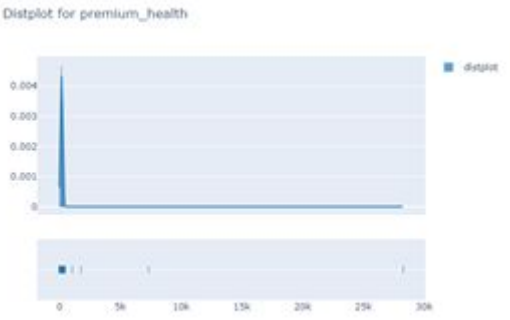


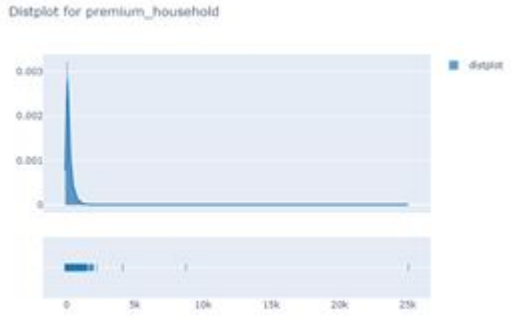
## 1. Distribution Plots

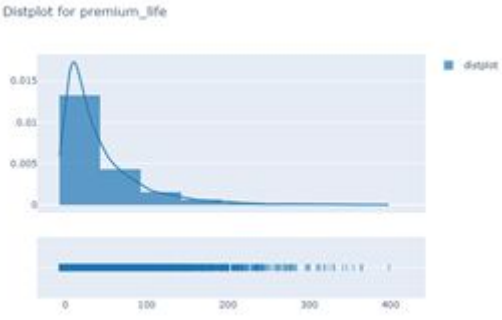
### 1.1. with outliers

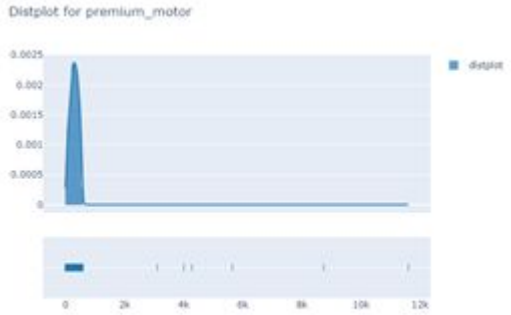


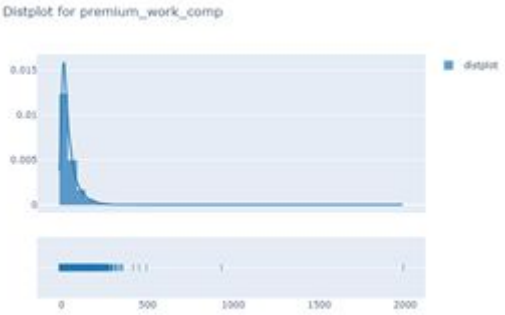


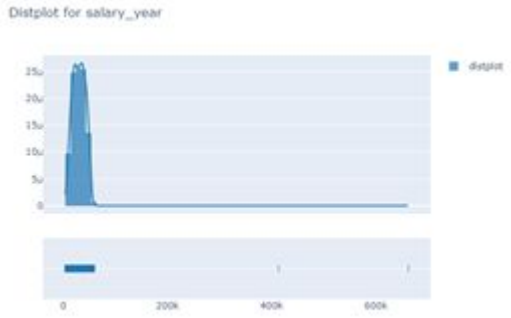



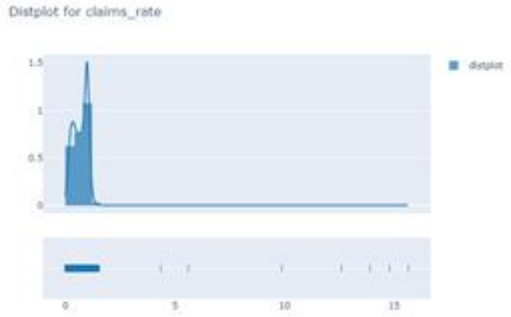





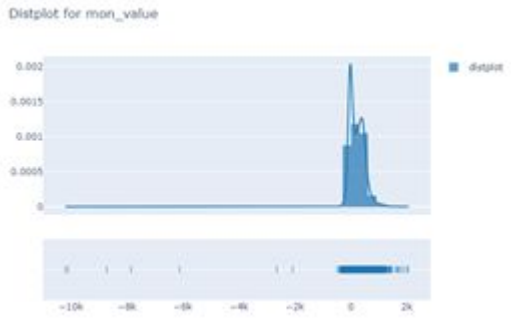



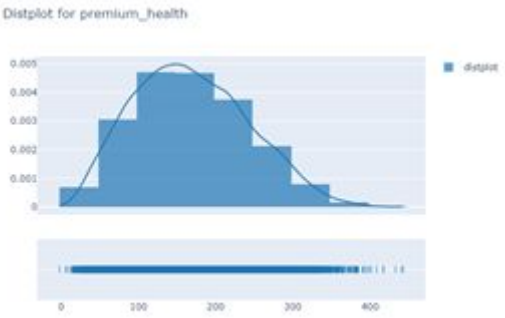




## 1.2. without outliers

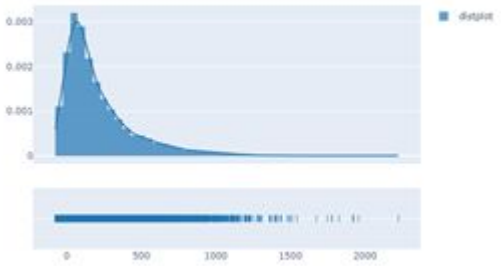




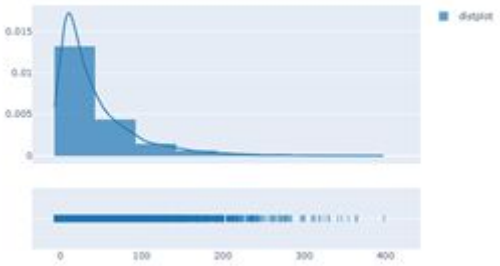


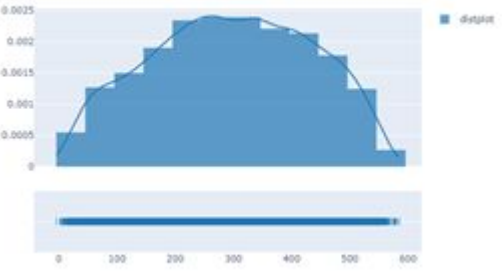
Distplot for premium_household



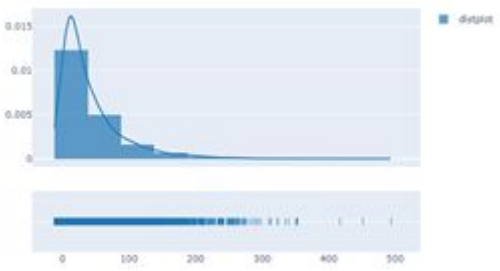

Distplot for premium_life



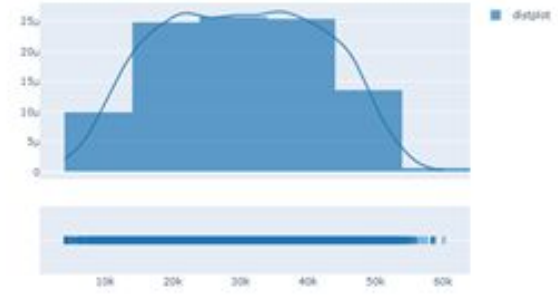

Distplot for premium_motor



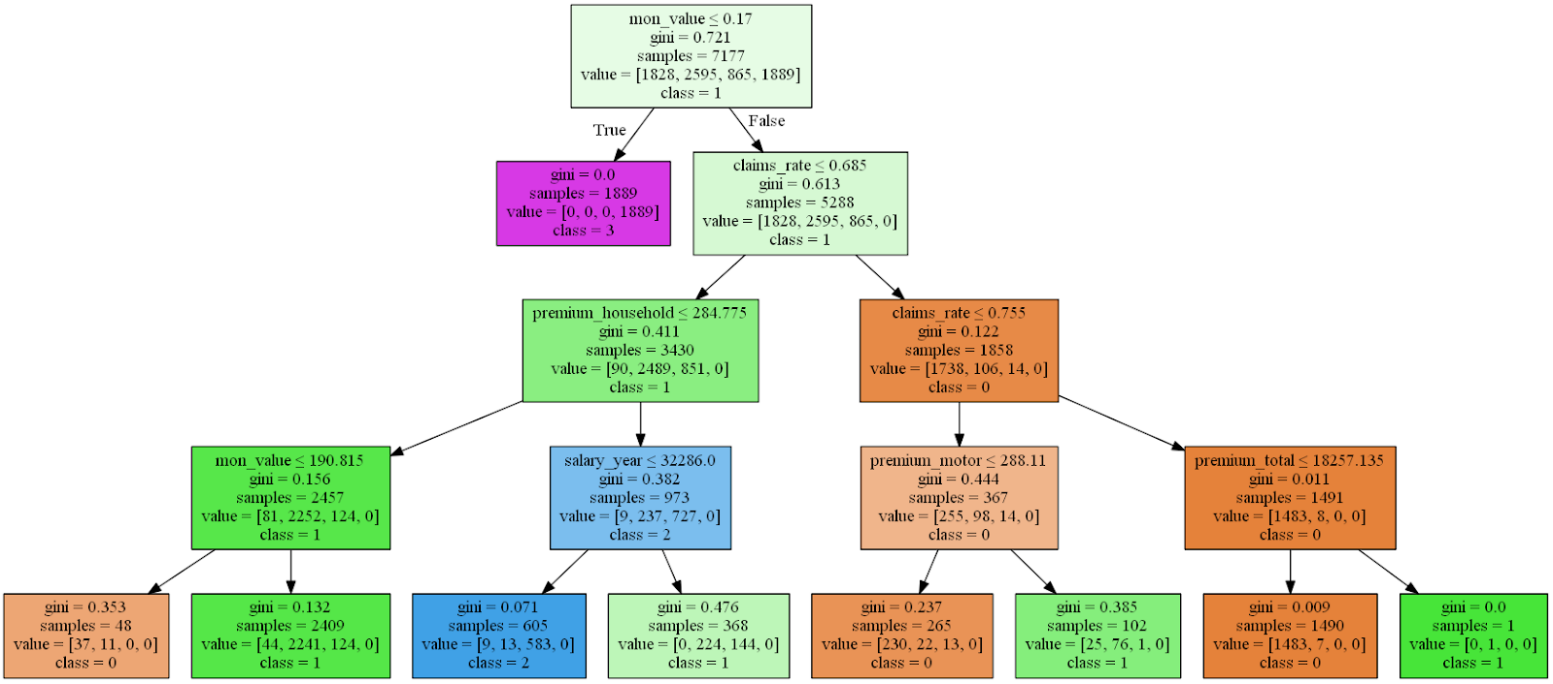

Distplot for premium_work_comp




Distplot for salary_year



## 2. Classification Tree




## 3. Raw Data Dictionary

The following table shows each variable of the dataset before data cleaning and feature selection.

| **Variable** | **Description** |
|---|---|
| Customer Identity | The identity of each customer; unique number |
| first_policy | date when the first contract was signed |
| birth_year | the birth year of each customer; manually inserted with several manual mistakes |
| educ | highest academic degree |
| gross monthly salary (salary_year) | Gross monthly salary (later gross annual salary) |
| location | Living area codes; no further information provided |
| has_children | dummy variable; 1 = customer has children, 0 = no children |
| mon_value | Customer monetary lifetime value; mon_value = annual profit x number of years – acquisition cost |
| claims_rate | contains data of the last two years; claims_rate = Amount paid by the insurance company / Premiums |

| | |
|---|---|
| premium_motor | Premiums paid by the customer for motor insurance |
| premium_household | Premiums paid by the customer for household insurance |
| premium_health | Premiums paid by the customer for health insurance |
| premium_life | Premiums paid by the customer for life insurance |
| premium_work_comp | Premiums paid by the customer for work compensation insurance |