Data Mining Project

# Table of Contents

[**Table of Contents**](#_y830ah3xd7g) **2**

[**Introduction**](#_pdmu6uxokyvd) **3**

[**Pre-Processing**](#_arg434sdo8op) **4**

[**Analysis**](#_sh9z7bxv1g7d) **5**

[Quantile Matrix](#_2w44fetnhtgg) 5

[Boxplots](#_4eb0bd3eyst5) 5

[**Predictive Modeling**](#_j2wsn2ydo4zu) **6**

[**Summary**](#_3i0s27p2i0mr) **7**

# Introduction

This report addresses the final project in the Data Mining course of Master Degree Program in Data Science and Advanced Analytics of Nova IMS. The groups of students received a fictional insurance company database, containing personal customers’ data as well as insurance consumption information. The features contained in the dataset are described in the [Project Description](http://elearning.novaims.unl.pt/moodle/pluginfile.php?file=%2F53291%2Fmod_folder%2Fcontent%2F0%2FProject%20Description.pdf&forcedownload=1) document.

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

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

This report is accompanied by the file [Data\_Mining\_Project.ipynb](https://1drv.ms/u/s!AvRbdQNKauSThrNiHocr8C_nsKQyfg?e=fDvik3) which contains all the Python code used.

# Preprocessing

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

# *NaNs* and outliers

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

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

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

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

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

# ‘Age’ column assessment

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

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

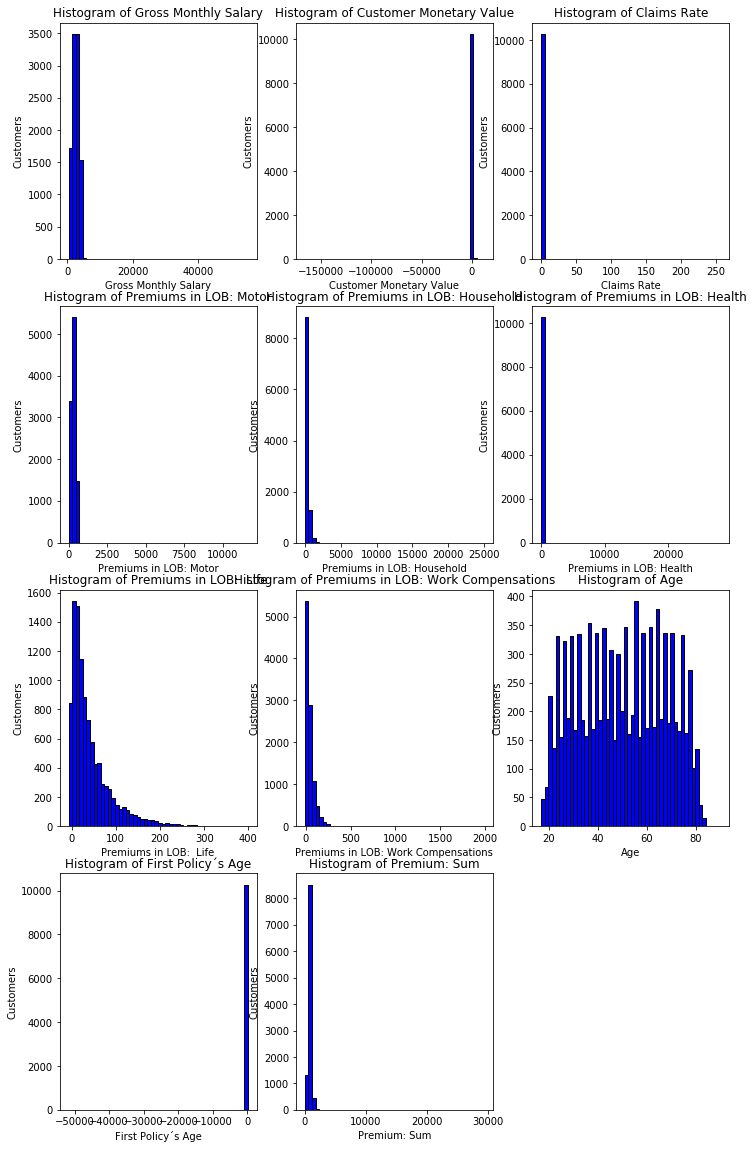


Figure 1 - Features histograms before removing outliers

A picture containing indoor

Description automatically generated

Figure 2 - Features histograms after removing outliers

# Categorical features classification

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

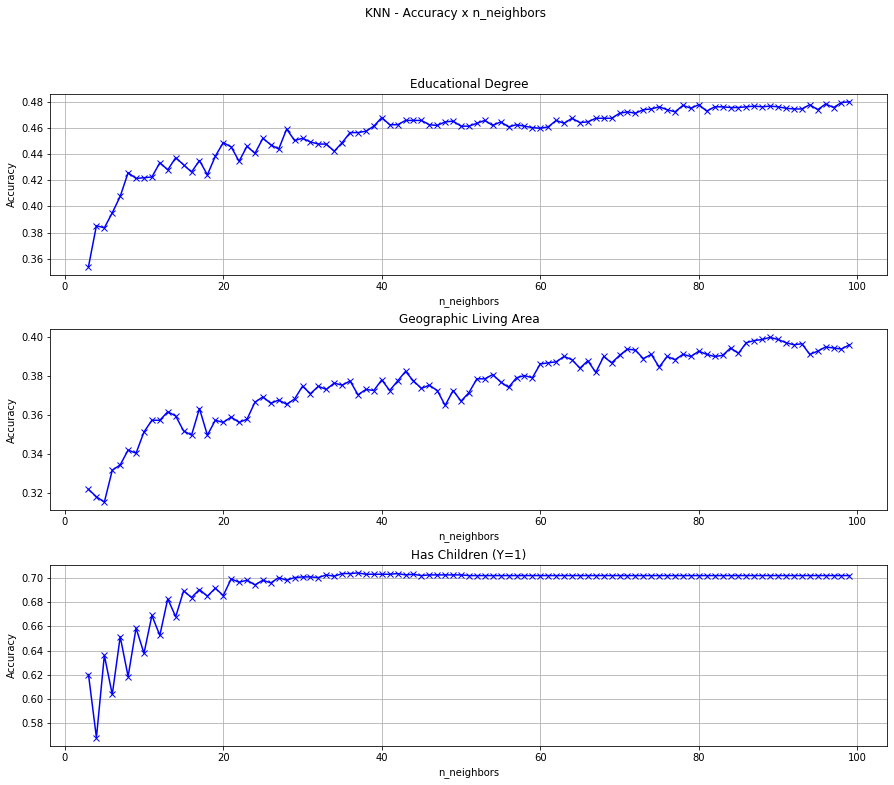


Figure 3 - Classifier accuracy for each categorical feature

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

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

# Numerical features regression

(RENNAN)

# Analysis

## Quantile Matrix

## Boxplots

# Predictive Modeling

# Summary