Student code: Student:

**Machine Learning Project**

Pre-processing:

Beginning the analysis by preparing the data collected for the training set in order to make the algorithm more performant.

The process of preparation is made from:

1. Treating incomplete data
2. Division between numerical and categorical data
3. Treating categorical data
   1. Bivariate analysis
   2. Creation of dummies data
4. Treating numerical data
   1. Univariate analysis
   2. Treating outliers
   3. Multivariate analysis
   4. Standardization
5. Merging numerical and dummies data
6. Test-Training set separation
7. Downsampling: Training set
8. Applying models
9. Overfitting test

**1.Treating incomplete data**

Immagine che contiene testo

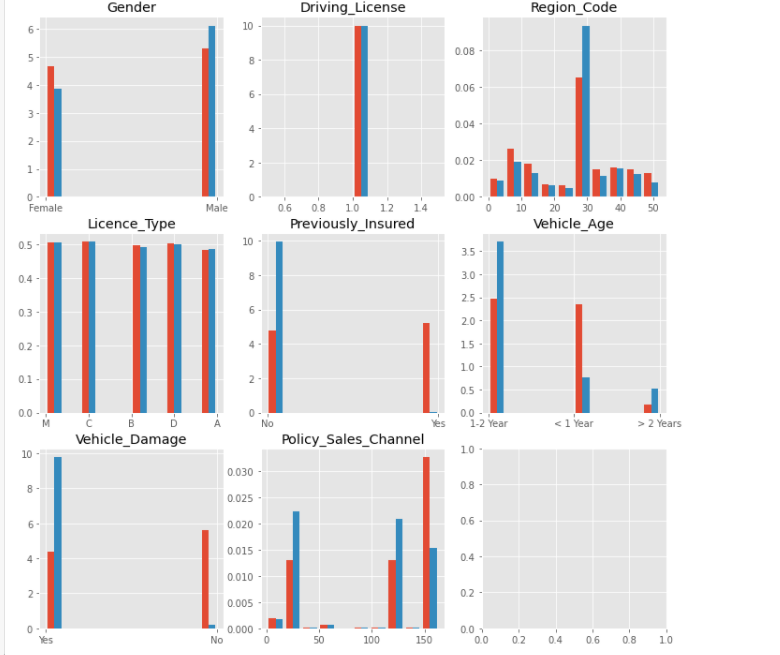
Descrizione generata automaticamente

**2.Division between numerical and categorical data**

In order to execute the analysis on the data, it’ s needed to divide the categorical and numerical data. Due their nature, it’s necessary to convert some numerical variables (Policy\_Sales\_Channel, Region\_code, Driving\_License).

**3.Treating categorical data**

**3.1.Bivariate analysis**

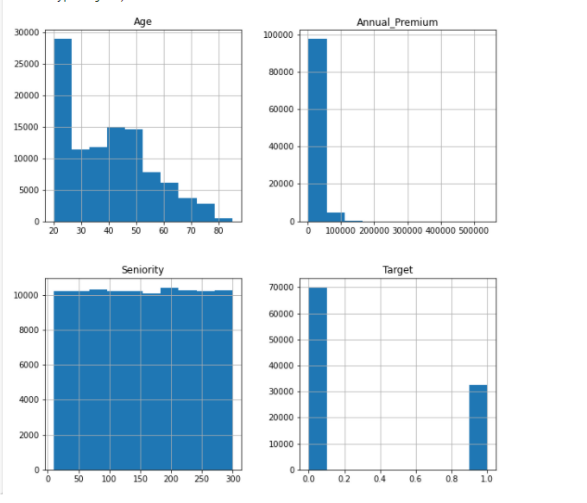
****

The first step in the Data exploration is the graphical analysis of the different variables. Here are presented the histograms of the categorical variables compared to the dependent variable “Target”. It’s clear that License\_type and Driving\_License are uncorrelated with the frequency of the Target, hence they have been discarded.

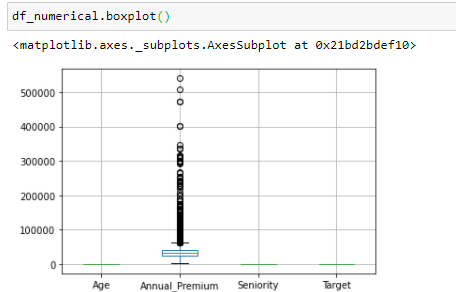
**3.2.Creation of dummies data**

In order to include also the categorical variables among the predictors of the target variable, it’s used a representation based on dummy variables.

**4.Treating numerical data**

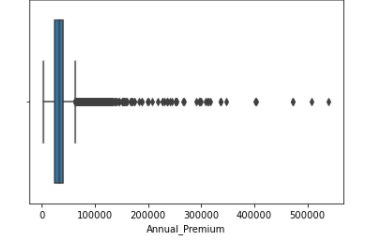
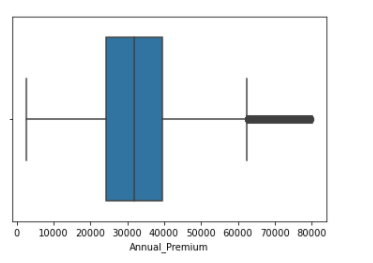
**4.1 Univariate analysis**

From the graphical analysis it seems that the Annual\_Premium variable is the one which needs more attention.

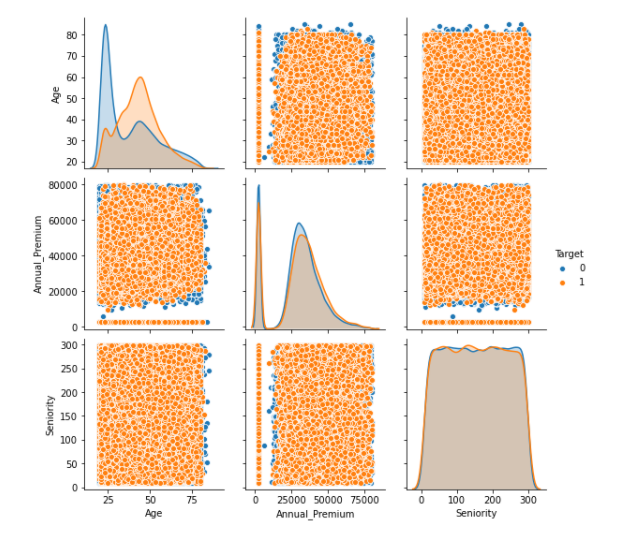


The boxplot’s analysis confirms the hypothesis that the Annual\_Premium variable needs to be treated, hence searching for outliers.

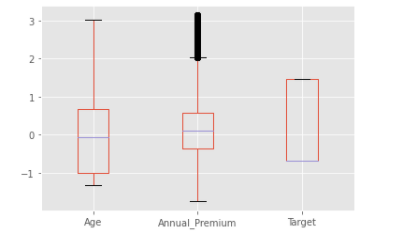
**4.2 Treating Outliers**

Using the Chebyshev’s Theorem with γ=3, we take at least 89% (in our particular case 99,26%) of the values.  
In fact 657 rows have been discarded, how we can see in the following image.

**4.3 Multivariate analysis**

****The relationship among the different pairs of variables respect to the Target show the uncorrelation between them and the Target.

Moreover , Seniority doesn’t have any correlation with the Target so only Age and Annual\_Premium have been chosen as components of the numerical subset.  
  
  
  
  
  
  
  
**4.4 Standardization**

****

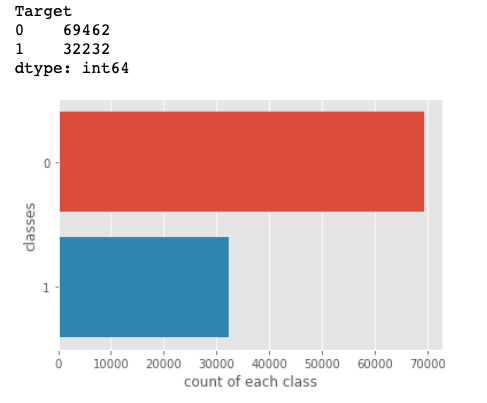
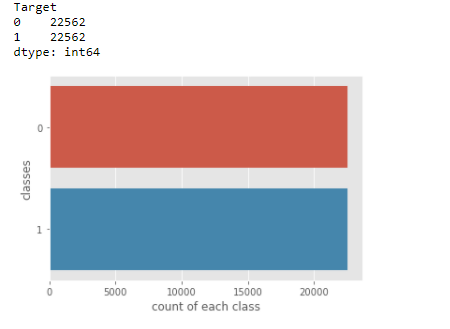
In order to make the algorithm more accurate it’s applied the z-index scaling method.  
  
  
  
  
  
  
  
  
**5. Merging numerical and dummies data**

In order to obtain a single subset to train on the model, numerical and categorical attributes are merged.

**6. Test-Training separation**

The available dataset is splitted into two subsets. One, which contain the 30% of the records, is used as Test – Set and the other one, containing the remaining 70% of the elements is used as Training – Set.

**7. Downsampling: Training set**



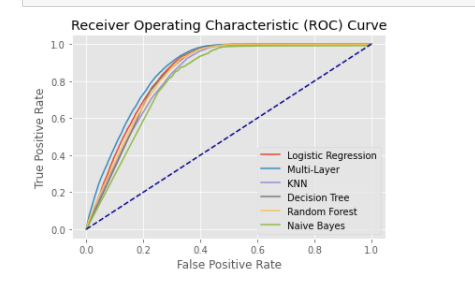
The Downsampling method could be useful, this becomes evident if checking the imbalance.   
After the Downsampling a balanced training set is obtained.  
  
  
  
**8. Applying models:**

The models tested are:

* KNN
* Decision Tree
* Random Forest
* Adaboost
* Logistic Regression
* Naïve-Bayes
* Multi-Layer Perceptron Classifier

For each model there are been searched the best hyperparameters in order to maximize F1.

Analysing all the F1 tests, the Multi-layer Perceptron Classifier is been proven to be the one with the best performance, also analysing all the ROC curves the Multi-layer Perceptron Classifier is been proven to be the one with the major Area under the curve.



**9. Overfitting test**

On the chosen model (Multi-Layer perceptron classifier), is made an analysis for the overfitting.

It’s not fair to compare a balanced train (X\_train) with an unbalanced test (X\_test). So the model is been trained with the balanced data (X\_train) while is been tested on the unbalanced data (X\_train0 and X\_Test) the results were:

F1\_X\_Train0= 0.7072775360464125

F1\_X\_Test = 0.7056620313184842

Hence the conclusion is that there is not overfitting.

1. **Conclusions**

In conclusion the model chosen is the Multi-Layer Perceptron Classifier with the best hyperparameters founded as:

using {'alpha': 0.1, 'hidden\_layer\_sizes': (10, 5), 'max\_iter': 2000}, trained on the downsampled Train – Set.