Université Catholique de Louvain
Institut de Statistique

Data science for finance and insurance

Individual project

November 30, 2020

LAMY Lionel

Contents

1 Introduction

In the field of insurance, we are often confronted with the question of how much an insurance premium a client has to pay. In an attempt to move towards an answer to this question, it may be interesting to try to determine what would be the frequency of accidents according to different criteria.

The aim of this work is to go through and summarize various modern methods and techniques that can be used to predict accident frequency. To do this, we will start by describing the data set we will use to build our models and base our estimates. We will then begin the main part of the work by presenting each method used and their results on the training set as well as the predictions obtained on another set (testSet) of 30,000 observations. At the end, you would find our best predictions and a small conclusion.

2 Datasets

2.1 Description

The dataset we will use to train our models has 70,000 observations and 10 variables as follows:

Variable	Type	Description
Gender	Factor	Male or Female
DriverAge	Numeric	18 to 82 in year
CarAge	Numeric	0 to 16 in year
Area	Factor	Suburban, Urban, Country side L.Altitude, Country side H.Altitude
Leasing	Factor	Yes or No
Power	Factor	Horsepower: Low, Normal, Intermediate, High
Fract	Factor	Splitting of the premium : Monthly, Quarterly, Yearly
Contract	Factor	Type of guarantee: Basic, Intermediate, Full
Exposure	Numeric	0.08 to 9.25 in year
Nbclaims	Numeric	0 to 4 in occurrence

Table 1: Dataset variables description

The testing set is structurally the same as the training set presented above with the only difference that we do not have the observations for *Nbclaims* [A.3.1]. Actually, this is the variable to be predicted in order to be able to compute our key ratio, the claim frequency, which is calculated as follows:

$$\text{Frequency} = \frac{\text{Nbclaims}}{\text{Exposure}}$$

2.2 Basic analysis

First of all, let's look at the variable of interest, i.e. *Nbclaims*. Its mean is equal to 0.068 and its variance to 0.070. Since the mean and the variance are very close and since we are facing counting data, we will argue that the assumption of equidistance is validated and that we can therefore use a poisson law to describe it. We observe that the latter is distributed in such a way that almost all observations have a value equal to 0. We also note the large difference between each count. Comparing this with a poisson law with a lambda equal to the average *Nbclaims* ($\lambda = 0.068$), we obtain very close results. However, having 93.45% of the observations with no claim cause the overall average *Frequency* to be very low as well and barely reaching 0.023.

Value	0	1	2	3	4
Count	65 418	4384	188	9	1
Percent	93.45%	6.26%	0.27%	0.01%	0.00%
Poisson	93.38%	6.39%	0.22%	0.00%	0.00%

Table 2: Distribution and comparison of the claims

Of the 4,582 people who have at least one claim, 70 have less than one year's insurance coverage. These people, and especially the 3 in the first group of the table below, have a much higher frequency than those with an Exposure greater than 1 and than the overall frequency. We are therefore faced with a difficulty: an enormous number of observations have a low frequency while, on the other hand, a small minority have a very high frequency.

Exposure	[0,0.2)	[0.2,0.6)	[0.6,0.8)	[0.8,1)	[1,2)	[2,4)	[4,6)	[6,10)	Total
Claims	3	12	19	36	428	2252	1620	212	4582
Mean Freq	11.4	2.57	1.39	1.11	0.66	0.34	0.23	0.17	2.23

Table 3: Distribution and mean frequency of the claims by group of Exposure

Hence, we want to know if relationships between variables appear spontaneously. The figure below has three parts: the histogram of each variable in diagonal, a bivariate scatterplot with a linear regression drawn in red on the left and the Pearson correlation of the regression on the right. Results are rather disappointing since we can see that only the *Exposure* seems to have an impact and explains - only up to 13% - the variable of interest *NbClaims*. We also find that no particular pattern or shape emerges.

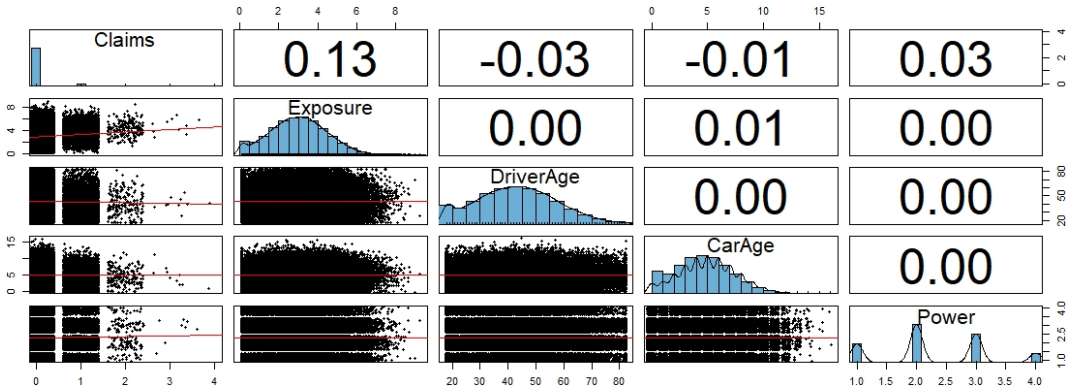

Figure 1: Histogram, regression and correlation matrix made with the package psych

Note: we have chosen to show only the variables Exposure, DriverAge, CarAge and Power, however the same illustration has been made for all the variables simultaneously and is available in the appendix [A.3.2]. Results are similar. We will see if more complex models manage to make us see more interactions later on.

3 Quality assessment

3.1 Criterions

To judge the estimation quality of the model, we will primarily use the Deviance, denoted D, which is a measure of how well the model fits the data. If the model fits well, the predicted means μ_i will be close to their real and observed values y_i, and so the deviance will be small.

$$D = 2\sum_{i=1}^{n}\{y_i \ \log(y_i) - y_i \, log(\mu_i) - y_i + \mu_i\}$$

An other criterion that we may use during the work to assess the quality of a model is the Akaike information criterion (AIC). Let k be the number of estimated parameters in the model and let $\hat{L}$ be the maximized value of the likelihood function, so the lower the AIC, the better the goodness of fit of the model.

$$AIC = 2k - 2\log(\hat{L})$$

We might also use the Bayesian information criterion (BIC) which strongly resembles the AIC. Both of them try to prevent the model from being overfitted by penalizing the addition of parameters, but the BIC penalizes more strongly the number of parameters.

$$BIC = -2\log(\hat{L}) + k\log(n)$$

3.2 Validation

Another way to verify the predictive power and quality of our models is to use cross-validation. These approaches are useful to know if the model would perform well on another dataset. We will use different techniques depending on the case and the facilities offered by the packages being employed, but primarily :

3.2.1 K-fold

With this approach, the data is divided into K groups, called folds. These folds are generated in such a way that none of them have the same data. Then, for each of the folds, we remove it from the data set and rotate the model on the remaining folds, then calculate the error obtained (MSE, Deviance, ..) and average it. We usually use 10 folds.

3.2.2 Leave-one-out

The principle is the same as the K-folds except that the number of folds is equal to the number of observations. We thus remove only one observation for each iteration.

4 Methods

4.1 Generalized Linear Models

4.1.1 Description

The generalized linear models (GLM), developed in 1972 and expanded by Nelder in 1983, allow to study the link between a dependent variable and a set of explanatory variables. This link specifies how the mathematical expectation of Y noted μ is linked to the linear predictor constructed from the variables explanations. In our case, we consider that the variable Y follows a Poisson's law and will therefore use the canonical link function, namely, $g(\mu) = log(\mu)$. The outcome is thus assumed to have mean μ_i, with

$$\log(\mu) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n$$

However, if we consider two observations both having a single claim but with a different number of contract years ($Exposure$), i.e. one and five years : a claim spread over one or five years should not have the same influence nor expected frequency. To remedy this problem, we use a so-called offset as follows (with $t = Exposure$):

$$\log(\frac{\mu}{t}) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n$$

$$\log(\mu) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \log(t)$$

Finally, since the link function is logarithmic, to obtain the value of μ, we need to take the inverse function, known as the exponential.

4.1.2 Models

As a first naive model (**1**), we simply took into account all the predictors of Table [1] without modification, using a poisson regression and $\log(Exposure)$ as the offset. We then tried the same model (**2**) with the only difference that this time we separated the continuous variables using the KMeans clustering algorithm with respectively 9 and 6 groups for the drivers and vehicles age. This number of clusters was not randomly selected and corresponds to the value that minimizes the variability of observations (within-cluster sum of squares or wss) while not being too large and providing what we believe to still be a noticeable improvement - i.e. the slope is not too flat.

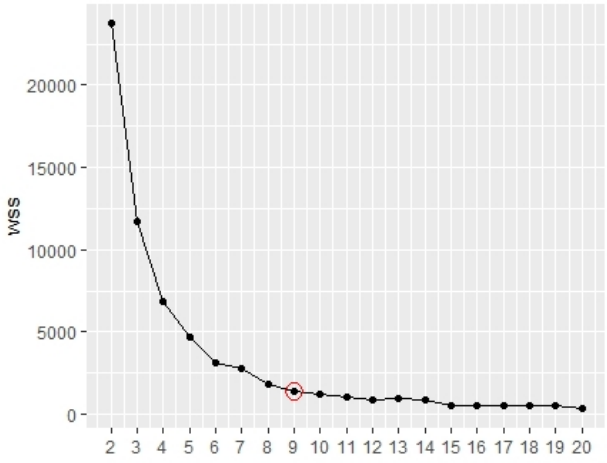

Figure 2: Example for the choice of DriverAge number of clusters

We obtained the following results:

Model	Built-in Deviance	Our Deviance	AIC	BIC	Min	Max
1	24 512.32	12 511.67	33 831.89	33 969.24	0.0007	0.3818
2	24 511.81	12 508.05	33 851.38	34 080.28	0.0000	0.4003

We immediately notice that our quality criteria seems to be pretty bad. Moreover, the maximum frequency is estimated at 0.40 for the 2nd model while the observed frequencies go up to 12.5 in the training set. However, the frequencies produced by the GLM (λ) correspond to the average expected values. Thus, by taking the sum of the probabilities of having $NbClaims$ equal to x where x range from 0 to 4 for a poisson distribution, we obtain the total number of $Nbclaims = x$ predicted by the model. The R command used is

sum(**dpois**(x, lambda=**predict**(model, type='response'))

and the results, similar for both models, are respectively: 65 425, 4367, 200, 8 and 0. Which is close to the observed values [2]. Cross-validating with 10 folds the model 2 gives, in average, 0.0685 with the MSE as the loss function and 0.3518 with the deviance.

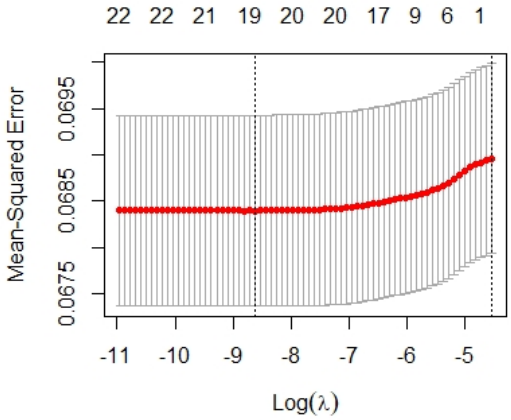

Figure 3: MSE loss

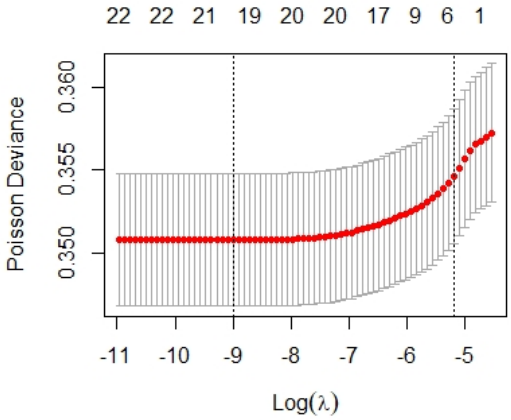

Figure 4: Deviance loss

4.2 Generalized Additive Models

Generalized Linear Models (GLM) has given us the ability to fit linear relationships. However, our data do not seemed to have strictly linear relationships [1]. With Generalized Additive Models (GAM), we can now fit data with 'splines', which are functions that can take multiple shapes. GAM fit to complex, nonlinear relationships and can make good predictions in these cases but in return we need to be more careful to not over-fit the models.

Unfortunately, the only numerical variables we can model with splines are the age of the driver and the car, leaving only a few possibilities. The model takes the following form:

$$g(\mu) = \beta^T x + s(x_k) + ... + s(x_j)$$

and in our case $s(x_k)$ and $s(x_j)$ are two cubic regression splines with 5 sub-intervals for $DriverAge$ and 3 for $CarAge$. The adjusted r^2 obtained is only 0.0242 and the Deviance explained 1.9%. The AIC, BIC and Deviance are worse than the GLM models, so we decided to drop this method because it does not seem to provide more predictive or explanatory power.

4.3 Regression Trees

4.3.1 Description

The previous two methods assumed that the data had a certain fixed structure. In this section we discuss a non-parametric method call 'regression trees'. They are powerful and commonly used because they require little data preparation. However, one of the major problems with trees is that if not tuned properly or if the data does not split well, they can become easily complex and not generalize well.

With the 'rpart' package in R that we used we can control some parameters, for instance : the minimum number of observations in a node for a split to be attemped, the maximum depth of the tree, and the most important one, the complexity parameter (cp). The latter is described in the documentation as "any split that does not decrease the overall lack of fit by a factor of cp." We also used the 'xval' parameter set to 10 for the algorithm to perform cross-validation.

4.3.2 Models

For this model, we will take the dataset with the unretouched age variables. It is no longer necessary to give an offset but instead we pass $Exposure$ directly with $Nbclaims$ as response variable. As a first try we let all the parameters by default and unfortunately the algorithm was not able to separate any observation and resulted in a tree with only one root. So we lowered the value of cp and tried 0.001 and 0.0005 and finally got these two trees:

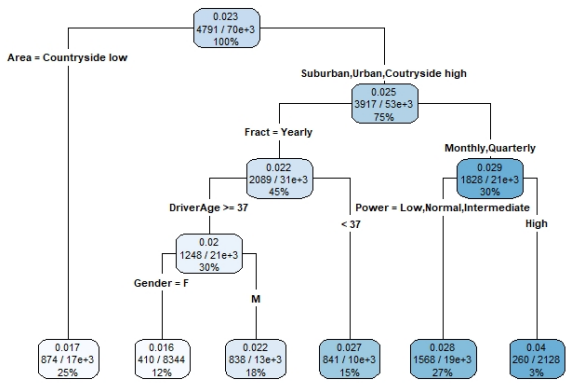

Figure 5: Tree with cp = 0.001

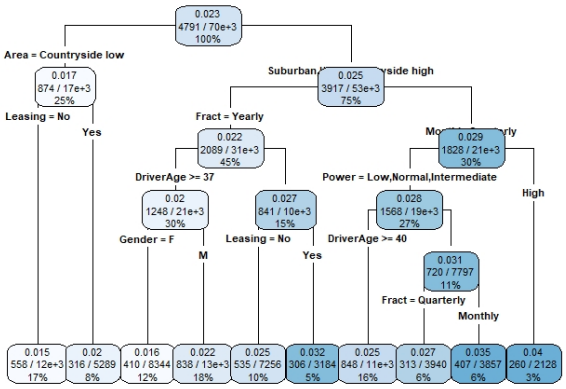

Figure 6: Tree with cp = 0.0005

We notice that in both cases, the first separation is done at the area level and that Countryside Low takes 25% of the data. We also observe that the highest predicted value is in the group on the far right, with people not paying annually and with a high power engine. On the contrary, women paying annually

and over 37 years of age seem to have one of the lowest frequency.

Performing cross-validation on both of them has, however, shattered our hopes. Indeed, although each split managed to slightly decrease the relative error, it still remains at 98.6% despite 10 splits. Moreover, we notice that the cross error (xerror) starts to increase again from 8 splits and doesn't go below the 99% error bar, which indicates that the trees are definitely not good.

	CP	Rel. Error	X.Error	X.St-dev
1	0.00500941	1.00000	1.00003	0.0097246
2	0.00267026	0.99499	0.99523	0.0096903
3	0.00183082	0.99232	0.99326	0.0096921
4	0.00109530	0.99049	0.99205	0.0096923
5	0.00100821	0.98939	0.99186	0.0096928
6	0.00063569	0.98839	0.99117	0.0096819
7	0.00056670	0.98775	0.99131	0.0097068
8	0.00055762	0.98718	0.99123	0.0097109
9	0.00052419	0.98663	0.99132	0.0097149
10	0.00050000	0.98610	0.99178	0.0097218

Table 4: Cross validation for the tree with cp=0.0005

4.4 Random Forest

Random forests are the aggregation of several regression trees. The objective is to reduce the tendency of a trees to overfit by taking the average prediction.

We have chosen to use a package made by Florian Pechon named rfCountData and available on github. The particular feature of this implementation is that it is specially made for counting data distributed according to a poisson law - that is our case. Another more than important advantage is that the algorithm considers that the best split is the one that maximizes the decrease of the poisson deviance.

We built a small random forest with only 300 trees, however, the average value of poisson deviance tends to already stabilize around 0.353 from 30 trees as you can see in the left figure below. By looking at the figure on the right, we notice that the most important variables remained the same as in the single trees we had generated earlier. That does not surprise us since a Random forest is basically composed with trees.

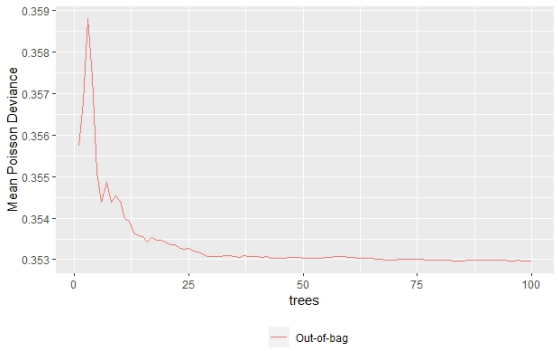

Figure 7: Poisson deviance evolution with increasing number of trees

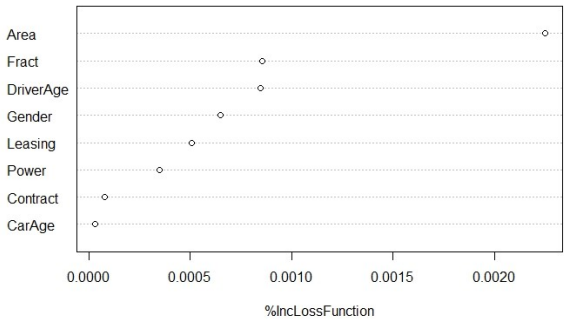

Figure 8: Variable importance in the Random forest

4.5 Gradient Boosting

Boosting is an iterative algorithm that similarly to random forests consists in training several models but this time it tries to reduce the error at each iteration rather than considering them independently. We have trained a GBM of 5000 trees with a learning rate of 0.01. The poisson deviance given by cross-validating seems to be stabilizing at 1136 iterations.

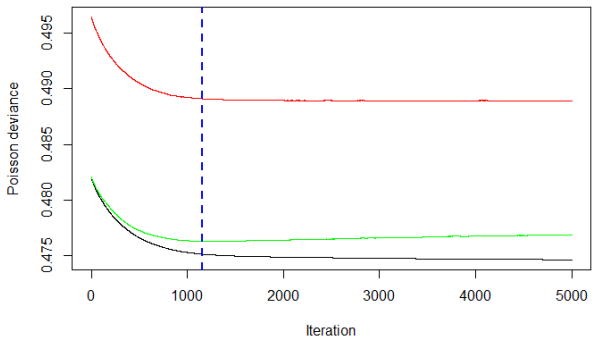

Figure 9: Evolution of the deviance with increasing number of trees

The partial plots, with the logarithm of the predicted frequencies on the y-axis are as follows:

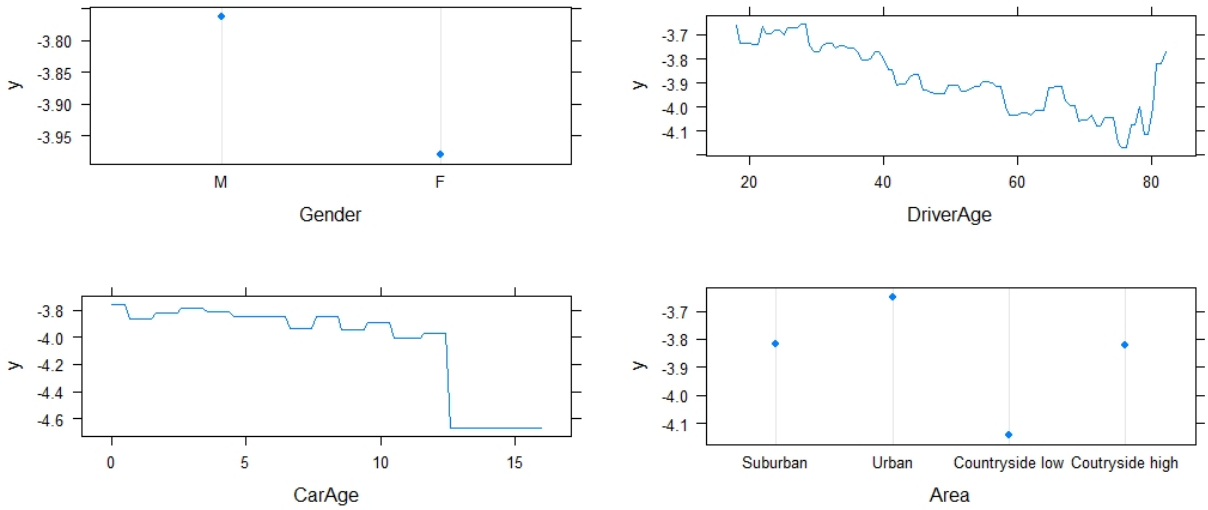

Figure 10: GBM partial plots with shrinkage = 0.01

4.6 Neural network

Trees create separations with fewer and fewer elements while neurons succeed in capturing the interactions between all covariates. A preprocessing of the data is necessary because the functions used are sigmoid which are quickly close to zero or one. We therefore scale the quantitative variables x_{ij} between 0 and 1 via :

$$x_{ij} = \frac{x_{ij} - min_{i=1,\dots,n}(x_{ij})}{max_{i=1,\dots,n}(x_{ij}) - min_{i=1,\dots,n}(x_{ij})}$$

Categorical variables must be transformed into dummy variables. We have also separated the dataset in two: a calibration set (80%) and a validation set (20%). Our best neural network looks like this:

We decided to start with 1 hidden neurons and increase the complexity of the model by 1 up to 4 hidden neurons and set the threshold to 1. Once that was done, we fitted each network to the test set 50 times in order to monitor the evolution of the deviance and ensure they did not fit too much the calibration set. The

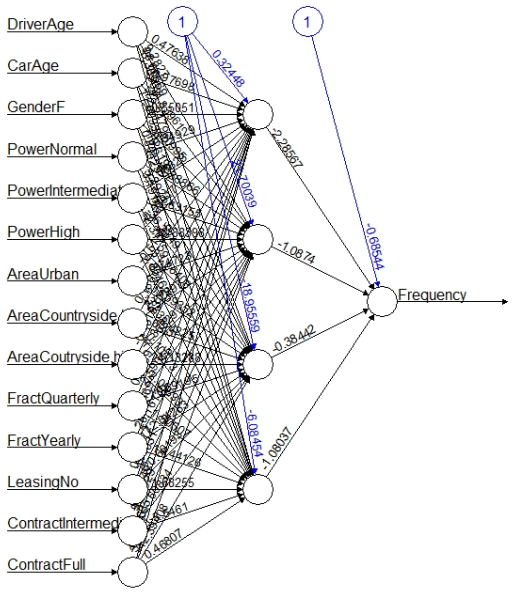

Figure 11: Neural network with 4 neurons

best one was with the 4 neurons network with the lowest variance obtained for the calibration set of 19 536 and 4811 on the validation set, summing to 24 347, which is quite an improvement.

5 Results

We have explored several models, each with their advantages and disadvantages. As we have noticed throughout the work, our dataset also has its own flaws: few claims, explanatory variables with little predictive power, and so on... which led our models to a rough ride.

We would have liked to push our analyses a little further, especially on the latest methods using evolutionary algorithms and machine learning, which seem to us more promising than basic models like glm. We would also have liked to make a comparison of the predicted average frequency for each variable with the observed frequency of the corresponding class (32 year old women with a 3 year old vehicle, living in rural areas, .. for example) but unfortunately we did not have the time.

Summaries for the predicted test-set frequencies for each method are:

	GLM Poisson	GLM Negbin	GAM	Tree	RF	GBM	NN
Min	0.0007	0.0007	0.0076	0.015	0.0011	0.0066	0.00100
Median	0.0622	0.0622	0.0216	0.0220	0.0649	0.02132	0.0211
Mean	0.0683	0.0683	0.0226	0.0226	0.0667	0.0224	0.0227
Max	0.3466	0.3472	0.0710	0.0396	0.2356	0.0762	0.2160

Table 5:

In the end, we think that it is the neural network with 4 neurons that might be the best and with which we will make our predictions on the final dataset of 30,000 other observations, available by clicking the link in appendix [A.1] below. It could happen, as that was the case in the training base, that some observations have a completely disproportionate frequency and thus far from the prediction. However, we believe that on average these predictions are not bad and since we have taken these very high values into account, the expectation of the predicted frequencies has been influenced upwards.

A Appendix

A.1 Code

Available at https://github.com/lamylio/LDATS2310 in the main.Rmd file.

A.2 Tables

DriverAge	Claims	No claims	% of claim	% without claim
[18,25)	595	7031	7.80	92.20
[25,35)	913	11 489	7.36	92.64
[35,50)	1701	25 077	6.35	93.65
[50,65)	1064	16 792	5.96	94.04
[65,80)	275	4537	5.71	94.39
[80, 100)	34	492	6.46	93.54
Total	4582	65 418	6.55	93.45

Table A.2.1: Breakdown of claims by (arbitrary) age group

By arbitrarily separating the observations by age group, we can observe that the lower age group, i.e. people between 18 and 24 years of age, seem more inclined to claim an accident. Moreover, the trend in the percentage of claims is downward, but we observe a re-emergence from the age of 80 onwards.

Area	Fract	DriverAge	Power	Leasing	Gender
125	81	62	28	27	25

Table A.2.2: Variable importance for the tree with cp = 0.0005

A.3 Figures

```
> summary(base_test)
 Gender      DriverAge         CarAge                    Area      Leasing            Power          Fract             Contract      Exposure
 M:18016   Min.   :18.00   Min.   : 0.000   Suburban       :9079   Yes: 8909   Low         : 5895   Monthly  : 5860   Basic       : 9037   Min.   :0.080
 F:11984   1st Qu.:33.00   1st Qu.: 3.000   Urban          :8962   No :21091   Normal      :12058   Quarterly: 6050   Intermediate:14971   1st Qu.:1.990
           Median :43.00   Median : 5.000   Countryside low:7552               Intermediate: 9041   Yearly   :18090   Full        : 5992   Median :3.010
           Mean   :43.38   Mean   : 5.028   Coutryside high:4407               High        : 3006                                            Mean   :3.025
           3rd Qu.:53.00   3rd Qu.: 7.000                                                                                                  3rd Qu.:4.030
           Max.   :82.00   Max.   :17.000                                                                                                  Max.   :9.260
> summary(base_train)
 Gender      DriverAge         CarAge                    Area      Leasing            Power          Fract             Contract      Exposure         Nbclaims
 M:42108   Min.   :18.00   Min.   : 0.000   Suburban       :20826   Yes:21105   Low         :13885   Monthly  :14050   Basic       :20859   Min.   :0.08   Min.   :0.00000
 F:27892   1st Qu.:33.00   1st Qu.: 3.000   Urban          :21179   No :48895   Normal      :27968   Quarterly:14079   Intermediate:35056   1st Qu.:1.99   1st Qu.:0.00000
           Median :43.00   Median : 5.000   Countryside low:17437               Intermediate:21057   Yearly   :41871   Full        :14085   Median :3.00   Median :0.00000
           Mean   :43.26   Mean   : 5.024   Coutryside high:10558               High        : 7090                                            Mean   :3.01   Mean   :0.06844
           3rd Qu.:53.00   3rd Qu.: 7.000                                                                                                   3rd Qu.:4.00   3rd Qu.:0.00000
           Max.   :82.00   Max.   :16.000                                                                                                   Max.   :9.25   Max.   :4.00000
>
```

Figure A.3.1: Summary for each dataset

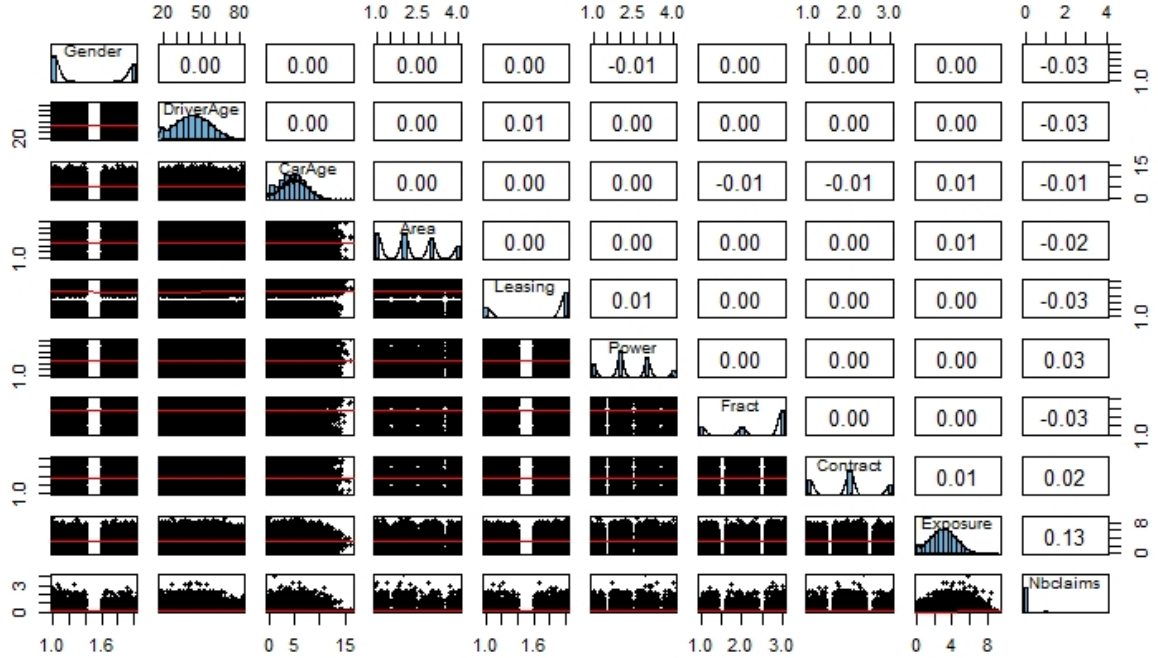

Figure A.3.2: Full scatterplot, histograms and regression

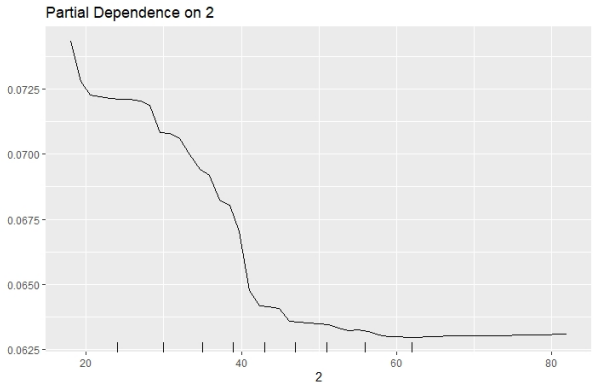

Figure A.3.3: RF Predicted frequency evolution by DriverAge

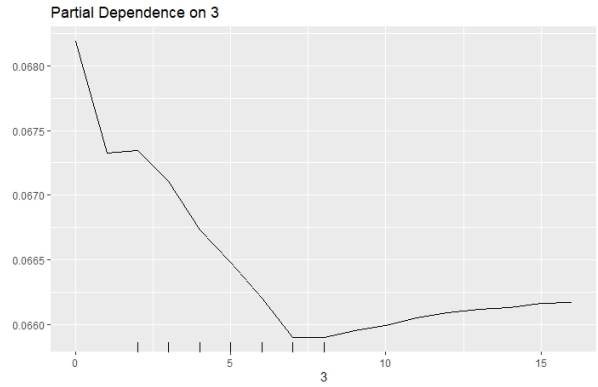

Figure A.3.4: RF Predicted frequency evolution by CarAge

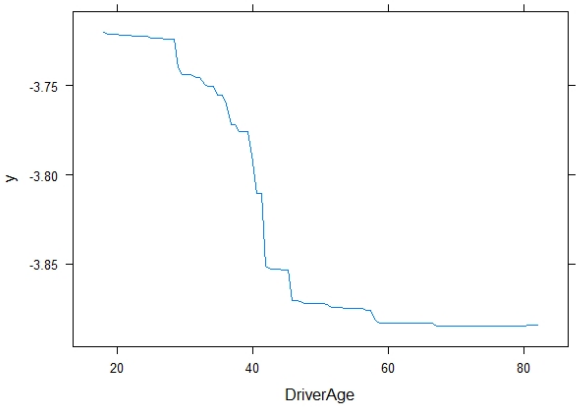

Figure A.3.5: GBM Predicted frequency evolution by DriverAge

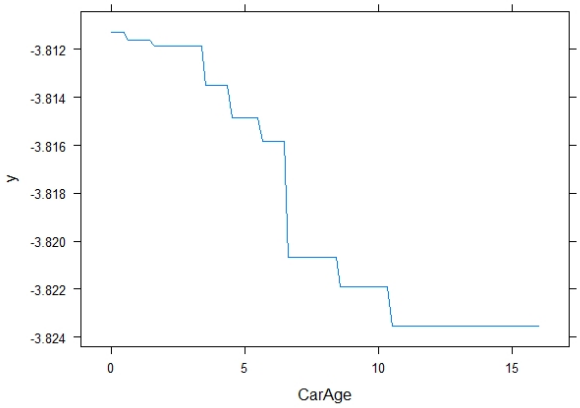

Figure A.3.6: GBM Predicted frequency evolution by CarAge