MEASURING THE IMPACT OF A BINARY VARIABLE ON A QUANTITATIVE RESPONSE IN A NON-PARAMETRIC FRAMEWORK

Version 1.4 of 13/05/2020

Auriol Wabo[1] **Frédéric Planchet**[2] **Maxence de Lussac**[3]

ISFA - SAF Laboratory[β]

University of Lyon - Claude Bernard University Lyon 1

ABSTRACT

This paper proposes a method for quantifying the effect of a qualitative explanatory variable on a binary response that is more flexible to use than the simple coefficient of a multiplicative GLM model; the objective is to have a measure that does not require the assumption of GLM proportionality and that is completely decorrelated from the effect of the other explanatory variables included in the model. The approach is illustrated using motor vehicle property loss cost data, for which an attempt is made to quantify the impact of the adjuster who assessed the claim on the valuation amount.

1. Introduction

The insurer often has to measure the impact of a binary explanatory variable on a quantitative response: do policyholders arriving through a given channel have a better loss ratio than others? Do owners of a two-wheeler have a lower customer value than those who have only insured one car? Answering these types of questions requires correcting for the effects of other variables influencing risk, since the typologies of

[1] Auriol Wabo is a consultant at PRIM'ACT. Contact: auriol.wabo@gmail.com.
[2] Frédéric Planchet is Professor at ISFA and Associate Actuary at PRIM'ACT. Contact: frederic@planchet.net.
[3] Maxence de Lussac is a consultant at PRIM'ACT. Contact: maxence.delussac@gmail.com.
[β] Univ Lyon, Université Claude Bernard Lyon 1, Institut de Science Financière et d'Assurances (ISFA), Laboratoire SAF EA2429, F-69366, LYON, France.

insured and/or claims differ for the two modalities of the characteristic under consideration.

In property and casualty insurance, the use of GLM regression models (see BESSON and PARTRAT [2004]) makes it possible to decorrelate the different variables and thus provides a simple answer in the form of a single coefficient synthesizing the effect of the variable, as long as a log link function is used.

The validity of this coefficient, however, presupposes that the hypothesis of proportionality of the effects is verified, since the conditional expectation of the response is written as a product of coefficients each attached to the modalities of the variables, which can be quite constraining. When this hypothesis is not verified and the relative effect of the variable of interest varies from one segment to another, it is necessary to construct other measures and the realization of a GLM per segment leads to an increase in the volatility of the estimators, making it an inefficient solution when the number of segments increases.

Two non-parametric approaches, based on *gradient boosting* (RIDEGWAY [1999]) and SHAP values (*see* SHAPLEY [1953] and LUNDBERG and LEE |2017]), are proposed in this work to construct this measure of influence.

In order to be able to construct the influence measure, the model for predicting the costs of the claims considered here must first be reconstructed using suitable techniques, in practice the *boosting gradient*. To do this, we place ourselves in the context of the study carried out by DE LUSSAC [2018] and extended by KHOUGEA [2019], which consists of measuring the impact of a network of experts on the cost of a motor vehicle equipment claim.

This paper is organized as follows: in the first part, the working context is recalled (section 2) and the algorithms for constructing predictors that serve as a basis for influence measurements are described (section3). Section4 is devoted to the description of the two proposed measures of influence. A comparison of the different approaches with the results of using a GLM is proposed in section 5, before concluding and formulating a recommendation (section 6). Readers interested in a broader presentation of this work can refer to WABO [2019].

2. Ratings and context of the study

This study is set in the following context, which can easily be transposed to many other situations: we seek to measure the influence of two networks of adjusters, A and B, on the cost of motor vehicle material claims, in order to determine the network driving, all

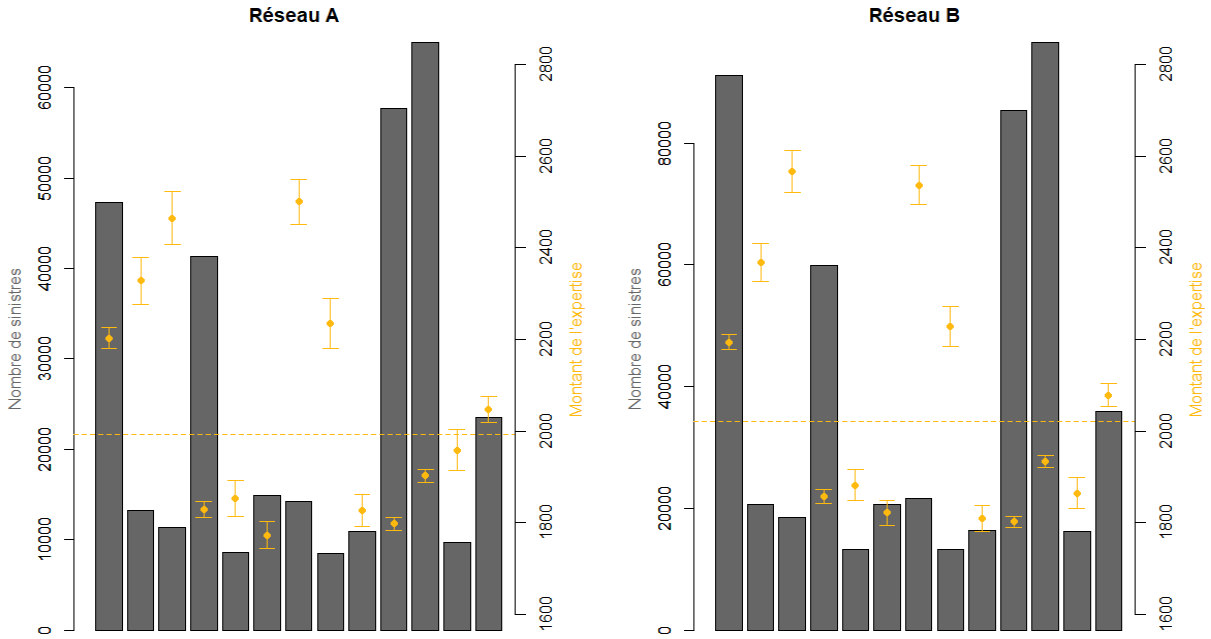
other things being equal, at the lowest cost, which will be described as "more efficient[4]". In the following, the cost difference between A and B is arbitrarily favoured.

The objective of the modelling is to measure the performance gap between two networks of experts when evaluating the cost of material claims. This evaluation consists of reconstructing the cost of claims (response variable) using nine characteristics (explanatory variables) that provide information about the vehicle (mileage, age and make), the type of accident, the hourly rate of the garage, the place and date of occurrence of the accident, the insurer that assumed responsibility for the claim and the network of experts that intervened, the variable of which has been named "target".

The following are some statistical summaries of the data used for the digital illustrations[5]:

- yellow discontinuous lines: average cost of claims appraised by network A (or B) ;
- dots in yellow: average costs for a given modality, with the associated error bars representing the 95% confidence interval[6] ;
- Grey diagrams: number of claims per modality.

Fig. 1 : Distribution of the networks' portfolio according to vehicle make

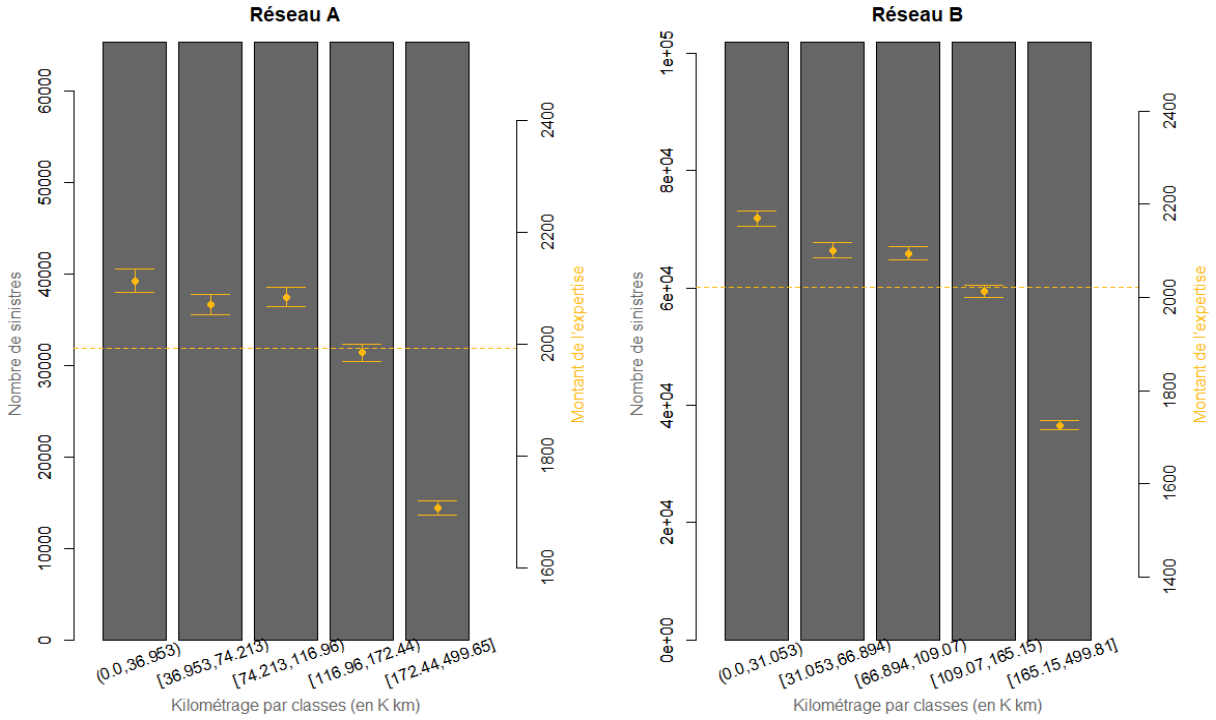


For each network, the figure above shows the composition of the claims portfolio for each make of vehicle and the associated average cost. There are a total of thirteen brands, twelve of which are main brands (AUDI, PEUGEOT, BMW, *etc.*), the other brands having been grouped in a modality named OTHER.

[4] Justifying the link between a notion of performance and lower costs is beyond the scope of this study. We can just indicate here that the experts respect a set of specifications and professional standards that should ensure an identical service to the insured, whether through network A or network B.
[5] A detailed presentation of the data is provided in WABO [2019].
[6] The confidence interval was obtained using a *Student*'s Law with n-1 degrees of freedom.

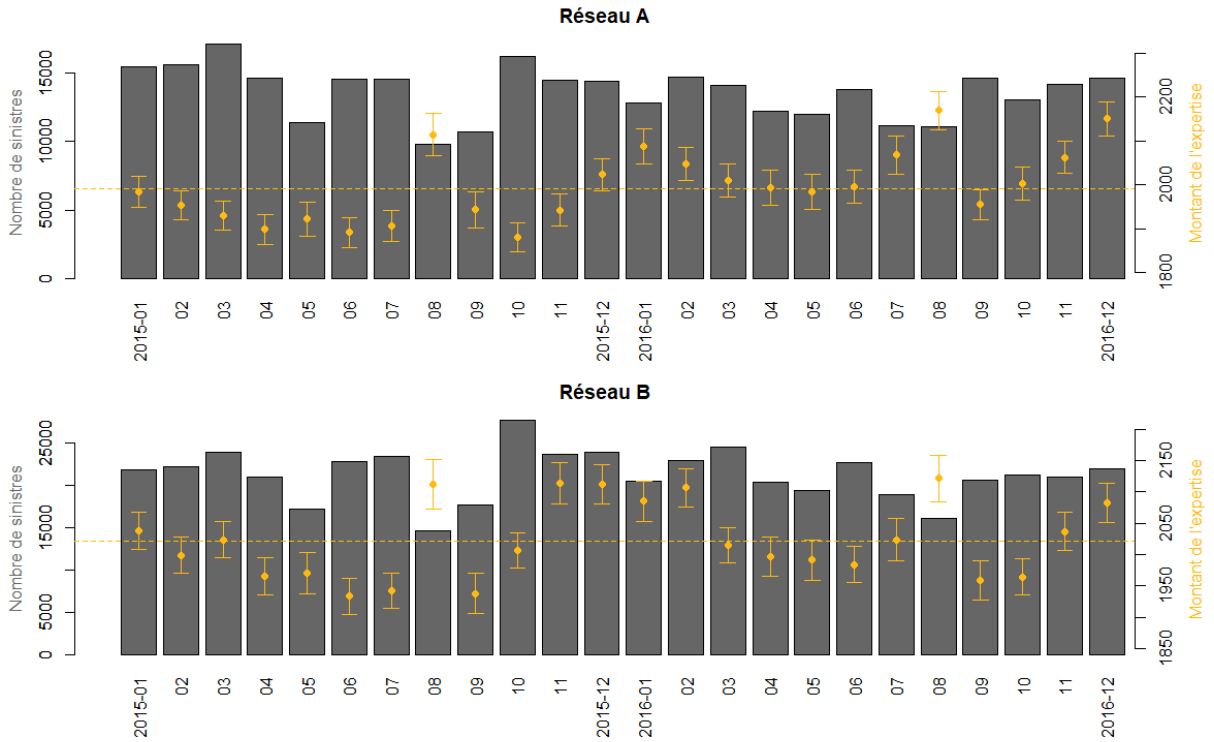
There is a similarity between the two networks.

Fig. 2 : Distribution of the network portfolio according to mileage



Mileage was categorized into five classes using quantiles. This discretization concerns only the previous graphical representation. In general, the older a vehicle is, the more likely it is to have travelled a long distance, hence the decrease in average cost observed above.

Fig. 3 : Distribution of the networks' portfolio according to the period of occurrence of the accident



The figure above shows a fairly similar trend in claims for the two networks.

Measuring the performance gap by comparing their average claims costs is only valid if the claims appraised by the networks are homogeneous, which is not the case in practice. This is why it is necessary to take into account the dissimilarities relating to the nature of the claims appraised by each network. For this purpose, a model is constructed to predict the costs of claims based on their characteristics.

A first possible approach is to use a GLM with a log link function. The difference in performance obtained is then -2.4% in favour of network A (cf. De Lussac [2018]). However, the author pointed out that the assumption of proportionality of GLM effects is questionable and that the *Gradient Boosting Model is* preferable for modelling average cost.

3. The prediction model: Gradient Boosting Models (GBM)

The studies carried out by De Lussac [2018] and Khougea [2019] were mainly aimed at reconstructing claims costs in a P&C (motor) context using a predictive model. In order to find the most suitable predictive model, they compared several models such as the *Generalized Linear Model* (GLM), the *Random Generalized Linear Model* (RGLM), the *Gradient Boosting Model* and the *Random Vector Functional Link* (RVFL) neural network.

The comparison of these models was based on the *Mean Square Error (MSE*) and the L1 error obtained on the test sample :

$$\text{MSE} = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}}(y_i - \hat{y}_i)^2 \quad \text{Error } L_1 = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}}|y_i - \hat{y}_i|$$

n_{test} is the number of observations in the test sample. For $i \in \{1, \dots, n_{test}\}$ y_i and $\hat{y}_i$ represent observed and predicted values respectively.

The best performing predictive model is the WBG (see RIDGEWAY [1999]) with a root mean square error of 2,411,535 and an L1 error of 651. Compared to other models, the WBG had the lowest errors. For this reason, this model was chosen for the modelling of this study.

Nevertheless, significant discrepancies between predicted and observed values remain. These are partly explained by the inadequacy of the variables available for building a high-quality predictive model, which leads to a lack of information to differentiate between certain claims that appear similar but for which the expert appraisal amounts are significantly different. However, these discrepancies do not call into question the consistency and value of the study, which is based on the measurement of a relative gap.

Before detailing the settings made to build the optimal WBG, let's start by presenting the framework.

a. Predictor construction

The *Gradient Boosting Models* (GBM) is a family of algorithms based on the *boosting*[7] and gradient of a supposedly convex and differentiable loss function. Their basic principle is to build a sequence of adaptive models so that at each step, each model added to the combination of previous models appears as a step towards a better solution. This step is taken in the direction of the gradient of the loss function in order to improve the convergence properties.

The objective of the WBG is to build an aggregated model f minimizing the quadratic loss function L.

$$f(x) = \underset{\psi}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\big(y_i - \psi(x_i)\big)^2 = \underset{\psi}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}L\big(y_i, \psi(x_i)\big)$$

Among the many versions of *boosting*, the one used in this study is *Stochastic Gradient Boosting,* which differs from classic *Gradient Boosting* by the addition of a random draw without resizing. $\tilde{n}\ (< n)$ at each iteration to create an adaptive model. In FRIEDMAN [2002], several cases have shown the relevance of this parameter. The estimate $\hat{f}$ from f shall be obtained using the standard algorithm set out in the Annex.

The WBG algorithms implemented in R (*gbm)* and Python (*scikit-learn*) are relatively easy to use when sticking to non massive data or simple analysis based on default parameters

[7] *Boosting* is a method of automatic learning that consists in iteratively applying a weak learning algorithm to optimize its performance.

(maximum number of iterations Mthe proportion of randomly selected observations pthe maximum depth of the regression trees K and the rate of learning or *shrinkage* λ). Nevertheless, both of these cases are rare in reality. For this reason, it is necessary to find an alternative with an algorithm optimized in terms of memory management and allowing an advanced parallelization of the calculations: the GBM package of the H2O module meets these constraints and, in this study, all the GBM algorithms were executed using this module.

We will then summarize the methodology used and the results obtained (*cf.* DE LUSSAC [2018]). The idea is to first apply the model with default settings, then optimize them to obtain the optimal predictive model.

b. The predictive model used

The parameterization of a model depends on two main constraints:

- the complexity of the model: the algorithm must run in a reasonable time while providing good results. The complexity is mostly controlled by the number of iterations of the algorithm.
- overlearning: the algorithm will naturally adapt to the learning sample as iterations go by, hence the need to define a stopping criterion associated with slow learning so that each iteration brings only a slight modification of the previous model.

In this study, the stopping criterion chosen is the control of the number of iterations by the root mean square error on the validation sample. Specifically, the algorithm stops as soon as the MSE on the validation sample no longer increases after 50 iterations. It is also possible to slow down the learning of the model by decreasing the learning rate and/or the maximum depth of the regression trees.

The first application of the WBG algorithm was done under R with the following parameters : M = 10,000 , p = 50 % , K = 5 and λ = 0,1. The prediction errors obtained on the test sample are as follows: MSE = 2,477,139 and Error L1 = 661. For information, the number of iterations that have been performed is 2,822 and the execution time[8] is 1 minute and 31 seconds.

The results obtained are based on a default setting. However, good forecast quality depends on an optimal configuration of the parameters, carried out using a method known as "grid search": values are proposed for each parameter, thus creating numerous

[8] the previous and following calculations were carried out using an optimised calculation server under Linux, provided by Prim'Act. This server is equipped with 64 GB RAM, and a 32-core processor with 2.6 GHz operating frequency and 40 MB cache. Performance issues are not central here, when they do become central, the approach can be adapted by considering the approach of LUNDBERG et al [2019].

possible configurations which will all be browsed through to obtain the best configuration. This optimization is quite delicate for two reasons :

- the existence of numerous parameters: in addition to the four parameters presented above, there are others such as the number of variables to be randomly selected, the minimum number of observations per sheet, etc. ;
- the majority of parameters are linked: adjusting the value of one changes the value of the other.

As the number of parameters is high and some of them play redundant roles in controlling overlearning, it is not necessary to optimize all of them. Only seven parameters were configured (see appendix f.

In order to find the optimal parameter values, one should first have an idea of the finite set of values from which to choose the model fitting value. For each parameter, the associated search interval has been obtained through numerous tests. These consist of calculating the model MSE of the model on the validation sample according to several values that are sufficiently distant to identify (expertly speaking) a set of values where the mean of the calculated MSEs is the lowest. The resulting set is the ideal set for grid search.

The predictive model used in this study having been recalled, it is now time to respond to the problem through the two approaches presented below.

4. Influence measurements

The proposed measures of influence are first constructed in a static manner, to provide an estimate, over a given period of time, of the impact on the response level of the binary variable of interest. The second step is to look at the use of these estimates in a dynamic framework, conducting them quarterly over a two-year period to see if any trends emerge.

Gaps between the proposed models will therefore be analysed in terms of level and trend.

c. First approach: Imputation of missing responses

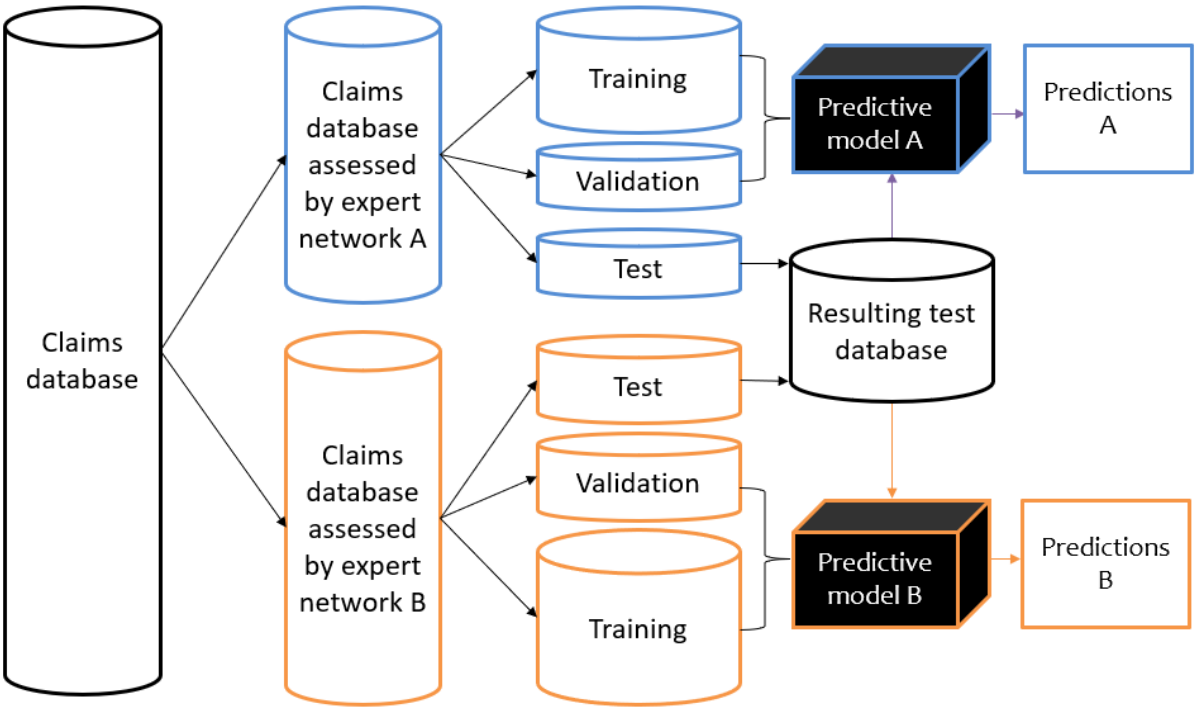
This approach involves comparing the two networks on a single claims basis to eliminate heterogeneity bias. However, there is no claims base where both networks have been involved, since a claim is appraised by A or B but never by A and B. It is therefore necessary to estimate the amount of the missing cost for each claim in the base in question. For this purpose, a model built on the basis of the claims appraised by each network is chosen to correspond to each network. Using these predictive models, a total of 3 amounts per claim will be available: the observed amount, the amount predicted by

Network A, and the amount predicted by Network B.

The construction of the two predictive models is performed by separating the claims of network A and those of network B, thus obtaining models constructed without the "target[9]" variable (as performed below). This process only works if there are similarities in the claims being appraised. It would also have been possible to construct both models with a single predictive model incorporating the "target" variable. The advantage of this process is that it can be used even if the claims assessed are totally different for each network. In the present work, the two-model approach is used, but the choice of either approach is irrelevant.

The cost prediction by the two networks was as follows :

Fig. 4 : Steps in predicting the cost of claims for Networks A and B



The division of the samples was done according to the proportions 3/5 (for the training sample), 1/5 (validation sample) and 1/5 (test sample). The concatenated test bases are the claims basis on which the comparison is made.

Two possibilities of comparison are proposed :

- the first is to compare the networks on the basis of the costs predicted for each network, the observed cost being taken as a benchmark without directly intervening in the assessment of the cost difference ;

[9] As a reminder, this is the name of the variable corresponding to the network of experts involved.

- the second is to compare the two networks on the basis of their observed and predicted costs, with the predicted value simply supplementing the data when it is not provided.

Comparison on the same claims experience - method n°1

Here, the evaluation is made on the basis of the predictions made and on the basis of the measurements. M and Err defined below. This method requires *a priori the* definition of reference amounts, i.e. amounts expected by the insurer. Thus, we look at the network (through its constructed predictive model) that has obtained on average the lowest cost of claims while having the reconstructed costs closest to those expected.

$$\mathrm{M}(\hat{y}_1, \dots, \hat{y}_n) = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i,\ \mathrm{Err}\big((y_1, \hat{y}_1), \dots, (y_n, \hat{y}_n)\big) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (4.1)$$

where y_i and $\hat{y}_i$ for $i \in \{1, \dots, n\}$ correspond respectively to the observed and predicted amounts. n is the number of claims contained in the resulting test database.

A simple numerical application of the previously defined measures to the claims costs gives :

$$\mathrm{M}\big(\hat{y}_1^A, \dots, \hat{y}_n^A\big) = €2{,}002.43 \qquad \mathrm{Err}\left(\big(y_1, \hat{y}_1^A\big), \dots, \big(y_n, \hat{y}_n^A\big)\right) = 2{,}453{,}265$$

and

$$\mathrm{M}\big(\hat{y}_1^B, \dots, \hat{y}_n^B\big) = €2{,}016.95 \qquad \mathrm{Err}\left(\big(y_1, \hat{y}_1^B\big), \dots, \big(y_n, \hat{y}_n^B\big)\right) = 2{,}417{,}710$$

The average value of the claims costs predicted by the model for network A is lower than the average value of the costs predicted by network B. This observation leads to the conclusion that the presence of network A reduces costs on average by about 0.72% (this is a similar interpretation to the linear models) or the performance gap is -0.72%. The calculation of this coefficient is detailed in section 4.

However, the error measurement obtained for network A is higher than that of network B. Taking into account the fact that the average cost of the observed claims is €2,006.93, it can be said that network A has been more efficient than network B because the average cost of the latter is further away from the average cost observed compared to network A.

Nevertheless, one cannot overlook the fact that these calculations were made on the basis of all model predictions (good and bad). This is why the comparison is made again, but this time according to the quality of the predictions. To do this, a *K-means* is applied to construct homogeneous classes of claims with respect to the measure d_1 defined below :

$$d_1\big(y_i, \hat{y}_i^A, \hat{y}_i^B\big) = \max\left(\left|\frac{y_i - \hat{y}_i^A}{\hat{y}_i^A}\right|, \left|\frac{y_i - \hat{y}_i^B}{\hat{y}_i^B}\right|\right) \qquad (4.2)$$

where y_i, $\hat{y}_i^A$ and $\hat{y}_i^B$ correspond respectively to the observed claims costs, predicted by model A and predicted by model B.

Classification *using* the *K-means* algorithm requires a number of classes to be defined before executing the algorithm. We have determined the number of classes so that the explained inertia[10] is greater than 95%. which leads us to retain 10 of them, homogeneous for the measurement. d_1. These classes are numbered from 1 to 10 in ascending order of class centres.

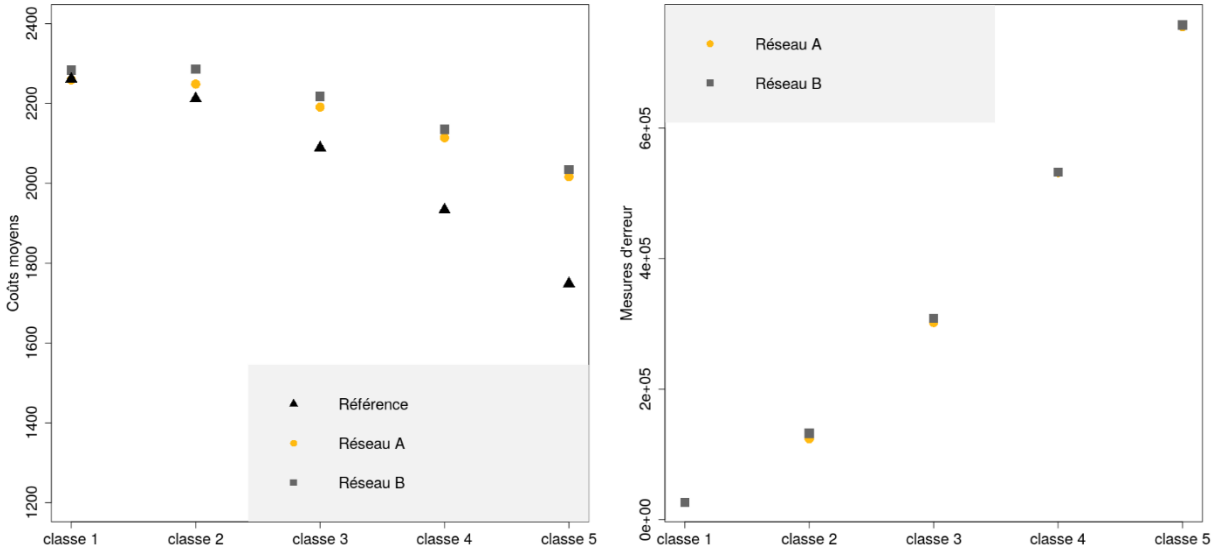
Tab. 1. Values of the centres of the constructed classes

Class numbers	1	2	3	4	5	6	7	8	9	10
Centre values	0.06	0.13	0.21	0.28	0.35	0.43	0.50	0.58	0.91	3.69

Given that in this first method, the estimates provided by each of the two predictive models are considered to correspond to the amounts that the networks would have provided after appraisal, it was considered reasonable to focus only on the claims contained in the first five classes. For these classes, the numerical applications of M and Err have been made. This procedure makes it possible to limit the biases associated with the average prediction quality of the model considered globally due to the limited number (10) of explanatory variables of the model.

The results obtained are as follows:

Fig. 5 : Measures M and Err calculated in the top five classes



The graph on the left above shows three average costs per class, that of the observed claims and that of the claims predicted by models A and B. It can be seen that the estimates provided by network A are on average lower than those provided by network B in each class. In addition, the graph on the right shows an increasing gap between

[10] The explained inertia is equal to the inter-class variance divided by the total variance.

average network costs and average expected costs. This growing gap can be explained by the poorer quality of the forecasts over the classes. Only in class 1, network B has a lower error measurement than network A.

Equivalent to that used to obtain the previous performance gap of -0.72%, here we obtain -1.26% taking into account only classes 1 to 5. That is to say, network A lowers the cost of claims by an average of 1.26% compared to network B. The overall (considering all predictions) or restricted (considering the best predictions) analyses lead to the same conclusion, with a difference in appreciation of the magnitude of the gap.

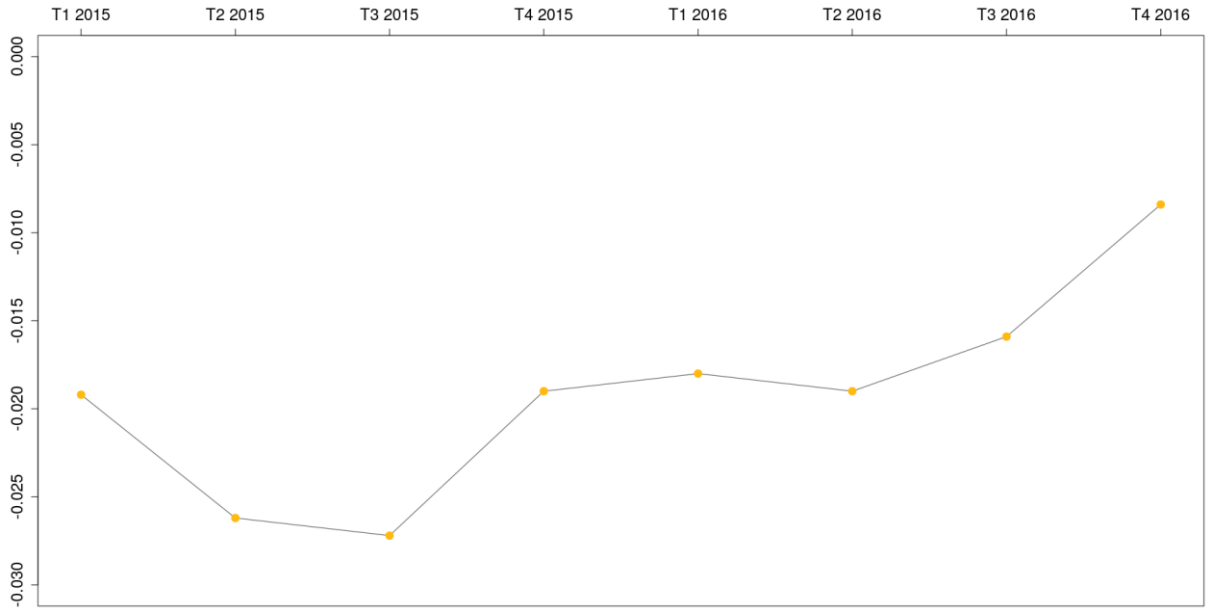
The time trend in the performance gap is now being studied to see if a trend emerges. The temporal study takes place over the quarters of the years 2015 and 2016, and concerns only the best predictions. The following table summarizes the performance variance by quarter.

Tab. 2. Performance gap measurement by quarter in the top five *clusters*

	2015				2016			
	T1	T2	T3	T4	T1	T2	T3	T4
Difference (in %)	-1.92	-2.62	-2.72	-1.90	-1.80	-1.90	-1.59	-0.84

The corresponding graph is as follows :

Fig. 6 : Evolution of the performance gap over the quarters in the top five *clusters*



There is an increasing trend, synonymous with a degradation of performance, this point will be commented on further above.

Comparison on the same claims experience - method n°2

In contrast to the previous method, which requires *a priori* a reference amount, in this method we compare the two networks on the basis of observed and predicted claims costs. For example, for a claim appraised by network A, the observed amount is the amount in the database and the predicted amount is the amount from model B.

We therefore calculate the cost that the other network would have obtained if it had intervened in place of the network that actually intervened. This calculation is performed *using* the predictive model of the corresponding network. By construction, this approach has the advantage of eliminating the bias for the observed values.

The comparison here is made using the following measure :

$$M'(y_1, \dots, y_m, \hat{y}_{m+1}, \dots, \hat{y}_n) = \frac{1}{n}\left(\sum_{k=1}^{m} y_k + \sum_{k=m+1}^{n} \hat{y}_k\right) (4.3)$$

where the y_i for $i \in \{1, \dots, m\}$ are consistent with the observed costs and the $\hat{y}_i$ for $i \in \{m+1, \dots, n\}$ correspond to the predicted claims costs. With m which is equal to the number of claims appraised by a network (those observed), and n is the number of claims contained in the test base (the one on which the networks are evaluated).

We get :

$$M'\left(y_1^A, \dots, y_m^A, \hat{y}_{m+1}^A, \dots, \hat{y}_n^A\right) = €1{,}998.88 \qquad M'\left(y_1^B, \dots, y_m^B, \hat{y}_{m+1}^B, \dots, \hat{y}_n^B\right) = €2{,}016.98$$

The results show that network A has on average a lower cost than network B. This makes the A network the more efficient of the two. More specifically, we can conclude that the presence of the latter decreases the cost by an average of about 0.90%, or 0.18 points more than the measure obtained in the first method.

This result is obtained on the basis of both good and bad predictions. But there is not a sufficiently reliable and relevant technique to select the best predictions. This limit can be assimilated to predictions *via* a linear model, in particular the calculation of all predictions is carried out using the same regression coefficients.

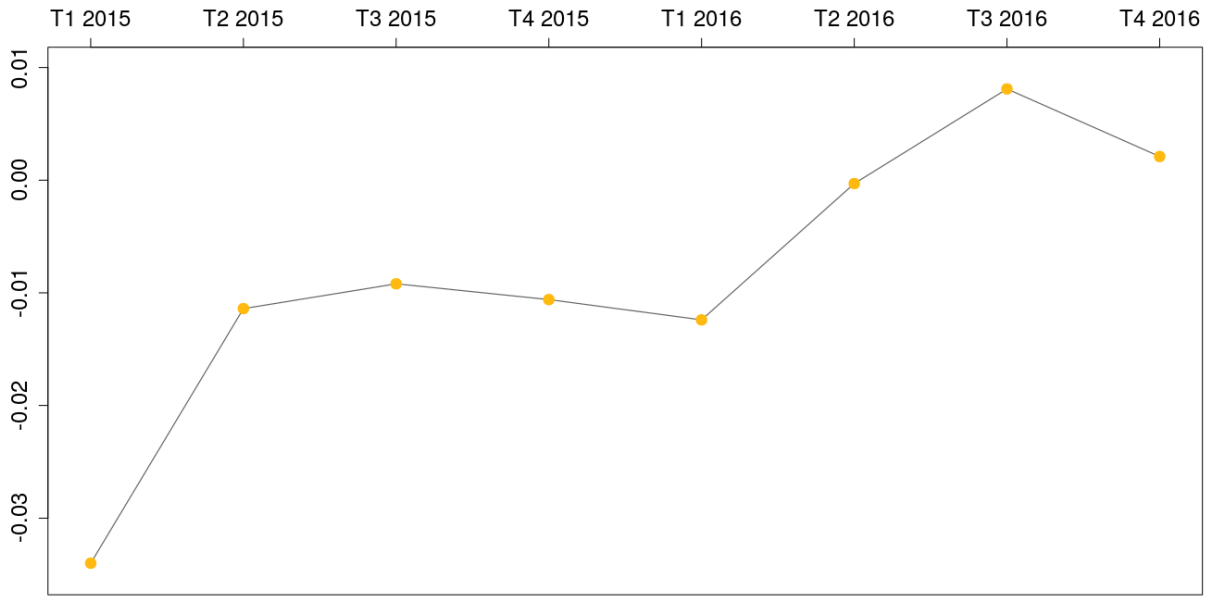
As in the previous method, it is useful to study the quarterly evolution of the performance gap. To this end, the measure defined in (4.3) has been applied to network costs on a quarter-by-quarter basis :

Tab. 3. Performance gap measurement by quarter

	2015				2016			
	T1	T2	T3	T4	T1	T2	T3	T4
Difference (in %)	-3.40	-1.14	-0.92	-1.06	-1.24	-0.03	0.81	0.21

A graphical view of these differences gives :

Fig. 7 : Evolution of the performance gap over the quarters



The above results indicate a trend towards a narrowing of the gap between the two networks, this time not always in favour of network A as in the first method.

Both methods lead to the same overall conclusion which is in favour of Network A. The second method has the main limitation that it cannot be restricted to the good predictions made by the model. Both methods provide better predictions than those of a linear model and (using a simple calculation that is detailed in Section 4) provide information (using a simple calculation that is detailed in Section 4) on the performance gap between the systems, which can be calculated globally or on any segment, which the GLM does not allow.

d. Second approach: using SHAP values

The pragmatic approach proposed above, while simple to implement, has an important limitation, however, which is that the resulting measure of deviation depends on the heterogeneity structure of the sets of claims on which it is calculated. To get around this difficulty, we will calculate the contributions of the variables for a predicted value, in order to better isolate that of the "target" variable.

Contributions are calculated using *Kernel* SHAP, a method derived from the Shapley value (see SHAPLEY [1953]). The latter provides an equitable distribution between the contributions of the variables and the solid mathematical theory of *Kernel* SHAP.

Kernel SHAP is a technique that allows to explain a prediction of a machine learning model, by building locally a linear model around this prediction (*cf.* RIBEIRO et *al.* [2016]). Its principle is to find the best linear model that reflects the behaviour of the predictive model for a given prediction (see LUNDBERG and LEE [2017]) :

$$g(x') = \underset{\varepsilon \in G_L}{\operatorname{argmin}} L(f, \varepsilon, \pi_{x'}) \qquad (4.4)$$

where f and g are respectively the model to be explained and the explanatory model ; G_L is the class of linear models;

x' is the vector corresponding to the absence/presence of the[11] explanatory variables of the instance of interest x in the prediction obtained;

And the parameters $\pi_{x'}$[12](proximity measurement) and L (loss function) are given below.

$$\begin{cases} L(f, \varepsilon, \pi_{x'}) = \sum_{z' \in Z} \pi_{x'}(z') \left(f\left(h_x(z')\right) - \varepsilon(z') \right)^2, \\ \pi_{x'}(z') = \dfrac{m-1}{|z'|(m-|z'|)} \binom{m}{|z'|}^{-1}. \end{cases}$$

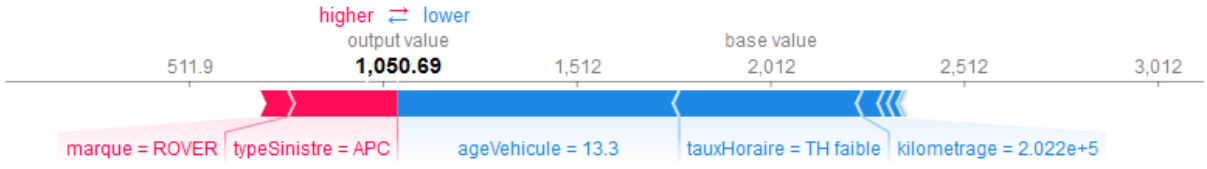
z' is the vector corresponding to the absence/presence of the m explanatory variables of the created neighboring instance zin the prediction obtained. z' is of value in $\{0,1\}^m$;
Z is the set of row vectors of the binary matrix representing all possible combinations[13] of z' ;
h_x is a function such that $z = h_x(z')$ and $|z'|$ corresponds to the number of non-zero elements of z'.

Application example

Kernel SHAP consists in explaining the difference between a predicted value and the mean predicted value. More precisely, it assigns to each explanatory variable a SHAP value, a value corresponding to its contribution to the difference obtained between the predicted value and the mean predicted value.

As the diagram below illustrates, this is a prediction made on a claim with an observed cost of €1,057.16.

Fig. 8 : Decomposition for an observation



Positive contributions (SHAP values) are shown in red, and negative contributions are shown in blue. Moreover, we can see that the predicted value (noted *output value* on the diagram) is €1,050.69 and the predicted average value (noted *base value on the* diagram)

[11] If x is big mso... x' is of value in $\{0,1\}^m$.
[12] It allows the creation of neighbouring instances to the instance of interest x.
[13] For example, if $m = 2$ so... $Z = \{(0,0);\ (0,1);\ (1,0);\ (1,1)\}$.

is €2,012. All the contributions have been well represented on the above diagram but not all of them are written due to lack of space.

These SHAP values were calculated using the Python function of the H2O module *predict_contributions*. This function can only be used for predictive models such as GBM, XGBoost obtained with the H2O package. Its syntax is as follows :

$$\phi = model.predict\_contributions(data)$$

where model and data correspond respectively to the predictive model built[14] under H2O and the test sample and $\phi = (\phi_0, \dots, \phi_m)$ (with ϕ_j column vector to n_{test} values) corresponds to a matrix of size n_{test} lines and m columns (m being the number of explanatory variables), of which the $m - 1$ The last columns correspond to the SHAP values of the predicted values. ϕ_0 corresponds to the mean predicted value, it is the same value throughout the column, we can assimilate it to the *intercept* in a linear model.

Modeling

The main interest of this approach is the contribution of the "target" variable (variable representing the network of experts involved). Specifically, we are looking at which network has the lowest average contribution. The network with the lowest average contribution will be the best performing network.

However, in order to remain in line with the performance gaps calculated so far (in percentage terms), these average contributions should be divided by the average predicted cost :

$$sH_A = \frac{{}^1/_{n_{test}^A} \sum_{i=1}^{n_{test}^A} \hat{\phi}_{ip}^A}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}} = -0.79\,\%$$

$$sH_B = \frac{{}^1/_{n_{test}^B} \sum_{i=1}^{n_{test}^B} \hat{\phi}_{ip}^B}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}} = 0.53\,\%$$

with $p \in \{1, \dots, m\}$ column index of the variable "target". The $\hat{\phi}_{1p}^A, \dots, \hat{\phi}_{n_{test}^A p}^A$ s represent the approximate values of the contributions of Network A and the $\hat{\phi}_{1p}^B, \dots, \hat{\phi}_{n_{test}^B p}^B$ are those of Network B's contributions.

$\left(\hat{\phi}_{ij}\right)_{1 \le i \le n_{test}; 1 \le j \le m}$ is the matrix of SHAP values obtained on the test sample. These contributions are approximate values to avoid long calculation times.

From the above results it can be concluded that network A is more efficient than network B. Equivalent to the interpretation made in the linear models, it can be said that the presence of network A reduces on average the cost of claims by 1.32% or that the

[14] The construction of the model was carried out using the learning and validation samples.

performance gap is -1.32%. This coefficient corresponds to the difference between sH_A and sH_BThe calculation of the coefficient will be explained in detail in section 4.

Although the contributions of the explanatory variables are evenly distributed, it is worth asking whether the same conclusion can be reached by looking only at the best predictions. Indeed, the contributions of the "target" variable from poor predictions may be greater or less than expected.

These scores are calculated again, but this time based on the quality of the predictions. For this purpose, homogeneous classes are constructed with respect to the measurement d_2 :

$$d_2(y_i, \hat{y}_i) = \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right|$$

where y_i and $\hat{y}_i$ correspond respectively to observed and predicted claims costs.

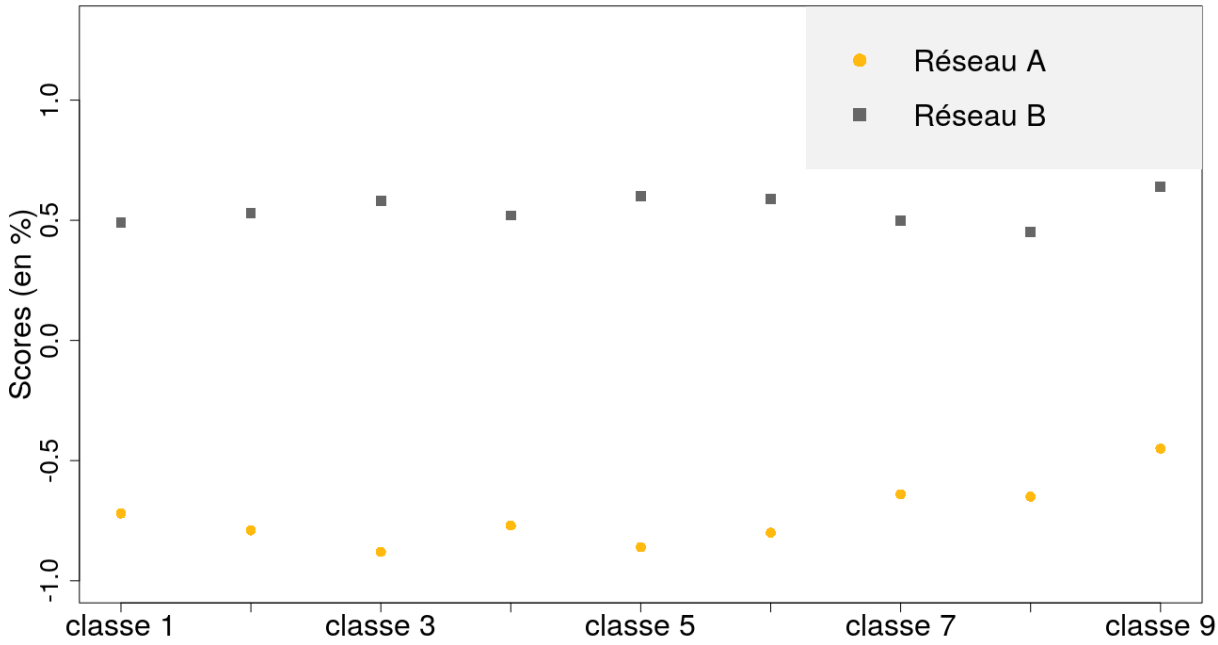
The classification was done using the *K-means* algorithm. The method used to find the number of classes is the same as the one used in the first approach: we take a number of *clusters* such that the explained inertia is greater than 95%. The *K-means* algorithm is applied with K = 9. The nine homogeneous classes with respect to the measure d_2 have been numbered from 1 to 9 in ascending order of class centre values (see below).

Tab. 4. Values of the centres of the constructed classes

Class numbers	1	2	3	4	5	6	7	8	9
Centre values	0.10	0.31	0.53	0.75	1.33	2.19	3.54	5.89	10.95

The two networks are compared again not on the basis of the whole test, but in each constructed group. As a result, the following results are obtained :

Fig. 9 : Graphical representation of the scores calculated by class, for each network



Looking at the graph above, we can see that network A has a better score than network B in each group. In addition, there is some fluctuation in scores, probably due to prediction errors. Since the distribution of contributions depends on the predicted values, if the predicted values are bad, the contributions of the "target" variable will probably be bad too. For this reason, it was considered reasonable to focus only on the first-class results, thus coming as close as possible to a reliable distribution. In this class, you get :

$$sH_A = -0.72\,\% \qquad sH_B = 0.49\,\%$$

In this case, the performance gap between the two networks is -1.21%, an increase of 0.11 points over the performance gap calculated on the basis of the entire test.

However, it is not useful to ask whether by comparing the two networks again over a larger number of well-predicted claims, the same conclusion will be reached. For this reason, we have applied the SHAP *Kernel* to the whole database, with the same predictive model as the one used in the application to the test database. The contributions obtained were ranked according to the quality of the predictions.

The best prediction class here includes, in addition to the number of previously well-predicted values, about one hundred and fifty thousand other well-predicted values. A numerical application of the resulting contributions gives :

$$sH_A = -0.71\,\% \qquad sH_B = 0.48\,\%$$

With one difference, the same values are obtained as those previously obtained. Despite

a larger number of well-predicted values, the conclusion remains unchanged, so the approach has some robustness[15].

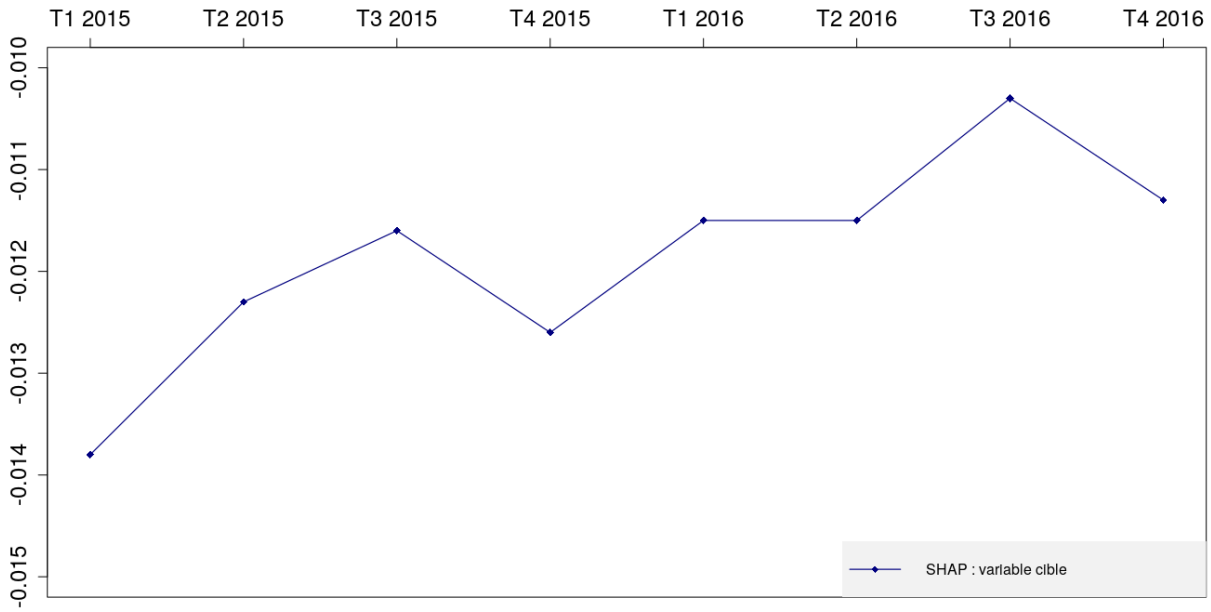
As in the previous approach, the quarterly evolution of the performance gap is presented. In other words, the scores previously obtained were calculated again, but this time on a quarter-by-quarter basis. By digital application, we obtain :

Tab. 5. Measurement of the performance gap by quarter in the SHAP approach

	2015				2016			
	T1	T2	T3	T4	T1	T2	T3	T4
Difference (in %)	-1.21	-0.79	-0.74	-1.07	-0.77	-0.63	-0.85	-0.99

The corresponding graphical representation is as follows :

Fig. 10 : Evolution of the performance gap over the quarters in the SHAP approach



This graph shows the same trend as each of the graphs performed in the previous methods. In addition, there is some stability in the spreads over time. We'll come back to that in section 4.

The results obtained in the proposed approaches are compared with those obtained with generalized linear models.

[15] *Kernel* SHAP has been applied to the test base and to the whole base, and in each case the contributions obtained are approximate values. The calculation time for the SHAP values of the test base (comprising approximately one hundred and sixty thousand observations) was approximately 7 minutes and 20 seconds, while that for the entire base (comprising approximately eight hundred and fifty thousand observations) was approximately 8 minutes and 30 seconds.

5. Synthesis of approaches and rapprochement with the GLM

As stated in the first part of section 2, the problem of the study had been addressed by DE LUSSAC [2018] in the context of predictions from generalized linear models.

Like the GLM, the two approaches previously presented allowed us to determine the most effective network of experts. Before comparing the results of the different approaches, the modelling carried out in the GLM is recalled below.

Performance gap from GLM

This part provides the modelling process carried out and the results obtained (*see* DE LUSSAC [2018]).

As a distribution, the gamma distribution was chosen because it is well suited to asymmetric and positive distributions. The logarithmic function was used to obtain multiplicative tariffs. Indeed :

$$log(\mathbb{E}[Y|X]) = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m$$

$$\mathbb{E}[Y|X] = exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m)$$

$$\mathbb{E}[Y|X] = exp(\beta_0) \times \left(exp(\beta_1)\right)^{X_1} \times \ldots \times \left(exp(\beta_m)\right)^{X_m}$$

The logarithmic link function also allows you to reason in percentage terms.

The implementation of GLM does not only require the definition of the distribution and the link function. Indeed, it is important to reduce the complexity of the calculations. The database used includes discrete variables with a fairly large number of modalities. Since the GLM calculates a modal regression coefficient for each discrete variable, it was deemed necessary to categorize these variables.

We can also note the choice of other parameterizations carried out such as the choice of the variable selection procedure or the criterion of penalization of the log-likelihood of the model. For more information, *see* DE LUSSAC [2018].

Modelling using the GLM leads to the conclusion that the A system decreases the predicted values on average by approximately 2.4 per cent compared to the B system. In other words, there is a performance gap of -2.4% (estimated value of the regression coefficient). This measure falls within a 95% confidence interval of plus or minus 0.4%.

Performance gap from the WBG

In the modeling done with the log link function GLM, the performance gap is obtained by reading the regression coefficient of the variable corresponding to the expert network involved. However, the modeling done with the *Gradient Boosting Model* does not

explicitly provide this performance gap. The latter was obtained as follows:

- **Performance** gap in the **first approach**: here, the performance gap between the two networks is obtained from the following formula :

$$écart_1 = \frac{\bar{y}_A - \bar{y}_B}{\bar{y}} \tag{4.5}$$

Where $\bar{y}_A$ and $\bar{y}_B$ correspond respectively to the mean predicted value of network A and the mean predicted value of network B. And $\bar{y} = 1/n_{test} \sum_{i=1}^{n_{test}} y_i$, the y_i are the observed costs.

In order to measure the difference in performance between the two networks, the coefficient of variation specific to each network is first calculated, i.e. the coefficients k_A and k_B such as $\bar{y}_A = \bar{y}(1 + k_A)$ and $\bar{y}_B = \bar{y}(1 + k_B)$. Then, the difference of the two coefficients was calculated, hence the expression in (4.4).
By numerical application of the formula (4.4) in the first method of Approach 1, we obtain :

$$écart_1 = \frac{2\,002{,}43 - 2\,016{,}95}{2\,006{,}93} = -0.72\ \%$$

In the second method of the same approach, we obtain :

$$écart_1 = \frac{1\,998{,}88 - 2\,016{,}98}{2\,006{,}93} = -0.90\ \%$$

- **Performance gap in the second approach**: here, the formula constructed to measure the performance between the two networks is :

$$écart_2 = sH_A - sH_B \tag{4.6}$$

where sH_A and sH_B correspond to the scores for networks A and B, calculated in section 3, which is the variation between the average contribution of a network and the total average contribution (or predicted average value). The idea is the same as in the previous approach: we start by measuring in percentage (for each network) the variation between the average contribution of the network and the total average contribution. That is, the coefficients k_A and k_B such as $C_A = \bar{y}(1 + k_A)$ and $C_B = \bar{y}(1 + k_B)$ (with C_A and C_B the average contributions of Networks A and B). Then, the difference between these two coefficients is made, hence the result in (4.5). By digital application, we obtain :

$$écart_2 = -0.79\ \% - 0.53\ \% = -1.32\ \%$$

Summary of results

The table below summarizes the performance gaps achieved :

Tab. 6. Summary table of the performance gap in the four methods

	Performance gap
GLM	-2.4 %
Identical loss experience 1	-0.72 %
Identical loss experience 2	-0.90 %
SHAP	-1.32 %

The variances in the above table are interpreted as the percentage reduction in the average cost of System A compared to System B.

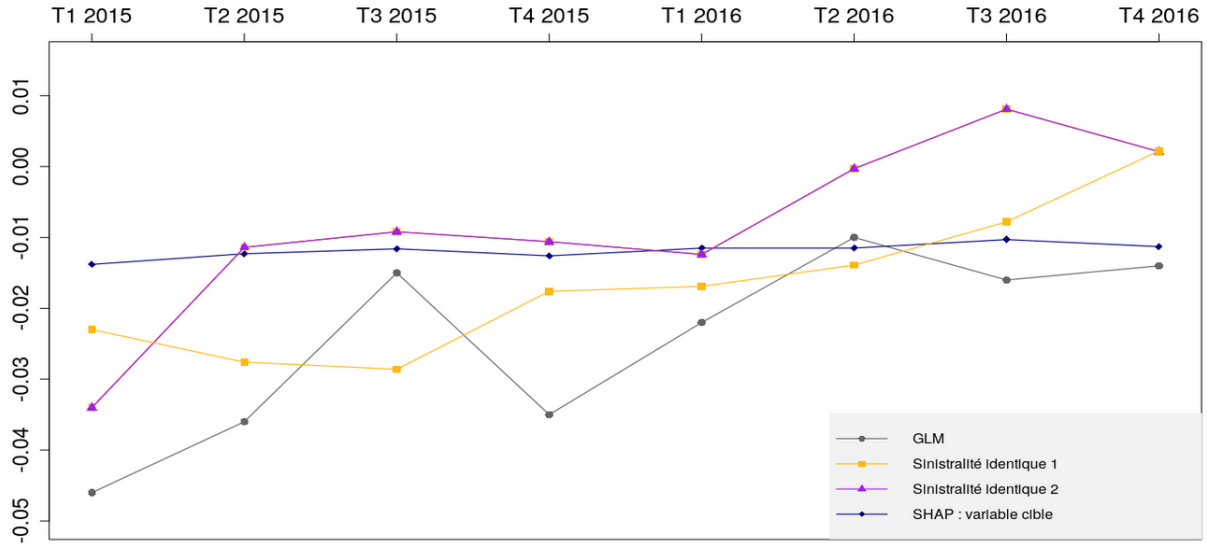
The above is a static comparison. It would be interesting to make a dynamic comparison, specifically a quarter-by-quarter comparison. Before showing the temporal evolution of the deviations of the four methods on the same graph, we recall the deviation measurements obtained using the GLM :

Tab. 7. Performance gap measurement by quarter in a GLM

	2015				2016			
	T1	T2	T3	T4	T1	T2	T3	T4
Difference (in %)	-4.6	-3.6	-1.5	-3.5	-2.2	-1.0	-1.6	-1.4
Standard deviation (%)	0.50	0.54	0.59	0.51	0.52	0.55	0.57	0.54

The following graph shows the quarterly evolution of the performance gaps calculated in the four methods :

Fig. 11 : Quarterly evolution of the performance gap in the four methods



A similar trend can be observed for deviations calculated *using* the GLM approach and

approach 1. In addition, there is a certain stability on the deviations calculated *using* the SHAP approach. But this approach also reflects the same trend as the other three curves, as shown in Figure 6.

The stability observed in the SHAP approach can be explained by the fact that the SHAP approach succeeds in dissociating the (minimal) correlation effects between the explanatory variables. This is something the GLM cannot do because it assumes total independence between the variables. However, in some practical cases there is a weak correlation between the variables and *Kernel* SHAP allows thanks to the Shapley value to better isolate the contribution of a variable. Approach 1 also does not allow to dissociate the correlation effect between the variables.

Now, taking into account this correlation effect in approach 2, i.e. applying the deviation calculation performed in approach 1 to approach 2, we obtain the following formula :

$$\frac{\bar{y}_A - \bar{y}_B}{\bar{y}} = \frac{{}^{1}/_{n_{test}^{A}} \sum_{i=1}^{n_{test}^{A}} \sum_{j=0}^{m} \hat{\phi}_{ij}^{A} - {}^{1}/_{n_{test}^{B}} \sum_{i=1}^{n_{test}^{B}} \sum_{j=0}^{m} \hat{\phi}_{ij}^{B}}{{}^{1}/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}} \qquad (4.7)$$

Approach 1 evaluates the networks on the basis of identical claims experience, in contrast to approach 2, which evaluates both networks through their database. However, using the same formula as in Approach 1 is not an aberration because similarities between the data from the two networks are found for each explanatory variable.

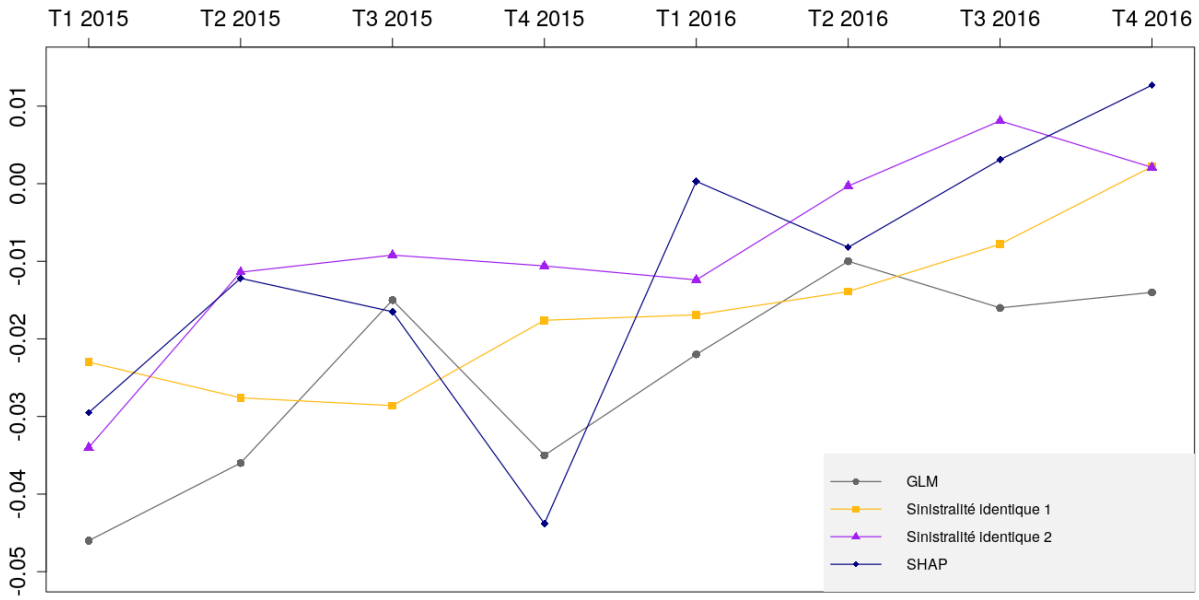
By numerical application of the formula (4.6) to the SHAP values, there is a performance gap of -1.30% (in favour of network A). Doing this calculation again, but this time quarter by quarter, we get :

Tab. 8. Performance gap measurement with correlation effect, by quarter in the SHAP approach

	2015				2016			
	T1	T2	T3	T4	T1	T2	T3	T4
Difference (in %)	-2.95	-1.22	-1.65	-4.38	0.03	-0.82	0.31	1.27

Comparing the differences obtained with those of the other three approaches, one finds :

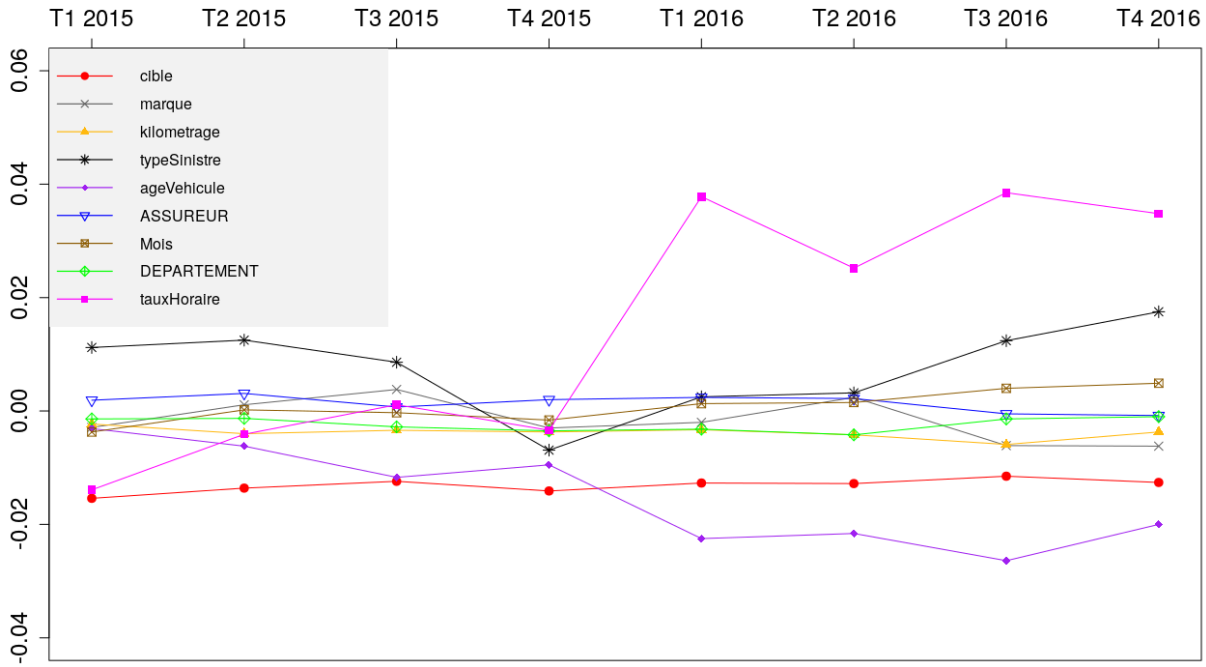
Fig. 12 : Quarterly evolution of the performance gap in the four approaches, all taking into account the correlation between variables



The curves of the four methods all describe a degradation of the performance gap, but nevertheless in favour of network A. There is a consistency and a certain adequacy between the results. Here, the trend is the same for all four approaches, which was not the case in Figure 7. In this figure, correlations between variables (however small) were not taken into account.

Using *Kernel* SHAP, it is possible to isolate the effect of other variables on the performance gap. The following graph shows the effect of each variable in the quarterly evolution of the performance gap.

Fig. 13 : Quarterly evolution of the contributions of variables on the performance gap with correlation between variables in the SHAP approach



The figure above was obtained by calculating for each quarter the deviation of each variable from the overall deviation defined in (4.6). Specifically, the reasoning is as follows :

$$\frac{\bar{y}_A - \bar{y}_B}{\bar{y}} = \frac{{}^1/_{n_{test}^A} \sum_{i=1}^{n_{test}^A} \sum_{j=0}^{m} \hat{\phi}_{ij}^A - {}^1/_{n_{test}^B} \sum_{i=1}^{n_{test}^B} \sum_{j=0}^{m} \hat{\phi}_{ij}^B}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}}$$

$$= \frac{{}^1/_{n_{test}^A} \sum_{i=1}^{n_{test}^A} \hat{\phi}_{i1}^A - {}^1/_{n_{test}^B} \sum_{i=1}^{n_{test}^B} \hat{\phi}_{i1}^B}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}} + \ldots + \underbrace{\frac{{}^1/_{n_{test}^A} \sum_{i=1}^{n_{test}^A} \hat{\phi}_{ip}^A - {}^1/_{n_{test}^B} \sum_{i=1}^{n_{test}^B} \hat{\phi}_{ip}^B}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}}}_{\textbf{Effect of the "target" variable}} +$$

$$\ldots + \frac{{}^1/_{n_{test}^A} \sum_{i=1}^{n_{test}^A} \hat{\phi}_{im}^A - {}^1/_{n_{test}^B} \sum_{i=1}^{n_{test}^B} \hat{\phi}_{im}^B}{{}^1/_{n_{test}} \sum_{i=1}^{n_{test}} \sum_{j=0}^{m} \hat{\phi}_{ij}}$$

In fact, it is the sum of the differences in each variable that explains the difference in performance obtained. *Kernel* SHAP manages to dissociate the existing effects between variables, thus attributing to each variable its marginal contribution.

The approaches developed in this study offer better predictions than the log link function GLM, with an average decrease of about 34% in the GLM MSE being found on the GLM MSE used in these approaches. This is due to the use of the GBM, a machine learning model. However, it should be noted that these two approaches, unlike the GLMs, do not

return error measurements associated with deviation measurements.

An important point to note is the modelling carried out to calculate the temporal evolution of performance gaps: the advantage of the GLM is that it can be applied as and when data become available without having to re-run previous calculations. In other words, the performance variance calculated for a quarter does not affect the variances calculated for previous quarters. The other two approaches, based on the use of the WBG, require all calculations to be restarted once new observations have been taken into account. This may result in a slight variation in the previously calculated deviation measurements. It is indeed unwise to envisage a GBM modeling per quarter, due to too few observations and thus exposing oneself to larger error measures than those obtained with the global base.

A key point justifying the interest of the proposed approaches is the possibility of modulating the impact measure on sub-segments when the hypothesis of proportionality is not verified.

6. Conclusion and discussion

At the end of this study, which aimed to measure the influence of the selected network of adjusters on the cost of motor vehicle material claims, two approaches were developed. The first is to compare the two networks on an identical claims basis and the second is to measure (using *Kernel* SHAP) the marginal contribution of a network to the predicted costs. Both approaches were implemented with a machine learning model: the *Stochastic Gradient Boosting Model*. The results are compared with those obtained by DE LUSSAC [2018] with a multiplicative GLM model.

All these approaches lead qualitatively to the same result, both in instantaneous (outperformance of A compared to B) and dynamic (deterioration of performance over the quarters of the two-year history considered).

The GLM, despite its limited assumptions or its lower prediction quality compared to that of a machine learning model, remains a sufficiently relevant model to get an idea of the comparison between two networks. Especially since its modeling is relatively simple and its interpretation understandable and easy.

In quantitative terms, different levels of performance are achieved and it is important to quantify the deviation measurement as accurately as possible. In this context, if we were to choose one measure from the four proposed methods, the one obtained with SHAP values would be preferred. It offers better predictions than the GLM, including a decrease in MSE of approximately 34% between the GLM and GBM. It is also noteworthy that it is based on a Shapley value theory framework. However, the latter does not return associated error measures as modelling *via* GLM does.

Having chosen the SHAP approach as the one most representative of reality, the problem of choosing the performance gap measure arises. Should we give preference to the gap measure that isolates the contribution of a network, and thus manages to dissociate the link between the variables? Or the one that takes into account, in addition to the contribution of a network, the correlation (however weak it may be) between the variables?

It's a pretty tough question. However, we advocate choosing the gap measure that completely isolates a network's contribution. For, it seems basic that this is the objective sought by the GLM. Except that, due to the fact that the latter neglects the minimal correlations existing between variables (assuming total independence between them), it ultimately returns a measure of deviation that informs the contribution of a network that takes into account the existing link between variables. This can be avoided with the SHAP approach.

Finally, it should be noted that all the proposed methods can be applied in a large number of situations, as soon as it is necessary to measure the impact of a binary variable on a quantitative response variable.

7. References

BERGSTRA J., BENGIO Y. [2012] Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281-305.

BESSON J.L., PARTRAT C. 2004] Non-Life Insurance - Modeling, Simulation, Collection: Insurance - Audit - Actuarial, Paris: Economica.

DE LUSSAC M. [2018] Comparaison de modèles prédictifs pour l'évaluation des coûts matériels automobiles, Dauphine, Mémoire d'actuaire.

FRIEDMAN J.H. [2002] Stochastic gradient boosting. *Computational Statistics & Data Analysis*, Volume 38, Issue 4, 28, Pages 367-378.

KHOUGEA D. 2019] Tarification IARD avec des modèles de régression avancés, Université de Strasbourg, Actuary's dissertation.

LUNDBERG S.M., ERION G.G., LEE S.U [2019] Consistent Individualized Feature Attribution for Tree Ensembles, University of Washington, Working Paper.

LUNDBERG S.M., LEE S.I. [2017] A unified approach to interpreting model predictions, *Advances in Neural Information Processing Systems*.

RIBEIRO M. T., SINGH S., GUESTRIN C. [2016] "why should I trust you?" : Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 1135-1144

RIDGEWAY G. [1999] The State of Boosting, *Computing Science and Statistics*, 31, 172–181.

RIDGEWAY G. [2007] Generalized boosted models: A guide to the gbm package. University of Pennsylvania, Working Paper.

SHAPLEY L.S. [1953] A value for n-person games. *Contributions to the Theory of Games.* 2.28:

307-317.

WABO A. [2019] Mesure de l'écart de performance entre deux réseaux d'experts en assurance automobile, Dauphine, Mémoire d'actuaire.

8. Annex

e. GBM Algorithm

Algorithm 1: *Stochastic Gradient Boosting* for Regression

Entrance(s) $\mathbf{x}$: observation to be expected ; $(\mathbf{y_i}, \mathbf{x_i})_{\mathbf{1 \le i \le n}}$ sample data ; $\mathbf{M}$: maximum number of iterations ; $\mathbf{p}$: proportion of randomly selected observations ; $\mathbf{K}$: maximum depth of regression trees ; $\boldsymbol{\lambda}$: learning rate or *shrinkage*.

Initialization : $\hat{\mathbf{f}}(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{y_j}$

Calculation : $\tilde{\mathbf{n}} = \mathbf{p} \times \mathbf{n}$

For $\mathbf{m}$ ranging from 1 **to** $\mathbf{M}$ **do**

Random and non-discounting draw of $\tilde{n}$ indices in $\{1, \ldots, n$ The indices drawn are stored in $\mathcal{O}$.

Calculation of $z_i = -\frac{\partial L(y_i, \rho)}{\partial \rho}|_{\rho = \hat{f}(x_i)}$ for $i \in \mathcal{O}$

Adjustment of a regression tree δ, maximum depth K, to couples $(x_i, z_i)_{i \in \mathcal{O}}$

Calculation of $\gamma = \underset{w}{\operatorname{argmin}} \sum_{i \in \mathcal{O}} L(y_i, f(x_i) + w\delta(x_i))$

Update : $\hat{f}(x) = \hat{f}(x) + \lambda\gamma\delta(x)$

End For

Exit(s) $\hat{\mathbf{f}}(\mathbf{x})$

f. Prediction model parameters

The parameters concerned and their associated search limits are as follows:

- **The minimum subdivision numbers** ($\mathbf{nbins}$) **and maximum** ($\mathbf{nbins\_top\_level}$) **for continuous variables**: during the construction of the regression tree, each quantitative variable is transformed into a qualitative variable whose number of subdivision factors is a parameter of the model. The greater the number of factors, the more possibilities there will be to study, thus increasing complexity, and the more precise the discretization will be. Default, $\mathbf{nbins}$ is set at 20 and $\mathbf{nbins\_top\_level}$ à 1 024. The study by De Lussac [2018] shows a small influence of these parameters on the MSE (on the validation sample) despite a large variation. For this reason, the value of $\mathbf{nbins}$ is searched in the interval [500; 1500] and that of $\mathbf{nbins\_top\_level}$ is set at 3,000.
- **The number of variables to be randomly selected** $\mathbf{col\_sample\_rate}$ This parameter allows the variables in each division to be selected randomly and without resetting. It has been shown by numerous tests on several values that the

forecast quality of the model also depends on this parameter. Thus, the defined search interval is [0.4; 1] in steps of 0.1. This interval is motivated by the results of prediction errors (on the validation sample) that were of interest starting at 0.4.

- **The minimum number of observations per sheet $\mathbf{min\_rows}$** : it defaults to 10. After several tests, the set of possible values defined is {1, 2, 5, 10, 20}.
- **The maximum depth of the trees $\mathbf{K}$** The optimal value is generally equal to 5 and it was decided to look for its optimal value between 4 and 8.
- **The proportion of randomly selected observations $\mathbf{p}$** parameter: as well as the $\mathbf{ncol_sample_rate}$The random draw is made without discount. After numerous tests, $\mathbf{p}$ is searched in the interval [0.5; 1] in steps of 0.1.
- **The learning rate $\boldsymbol{\lambda}$** It penalizes the addition of a new model in the aggregation and slows down convergence. Its optimal value is sought in the range [0.05; 0.1] in steps of 0.001.
- **The degressive factor $\boldsymbol{\lambda_m}$** It allows better control of convergence by making it sufficiently slow. It intervenes by replacing the apprenticeship rate $\boldsymbol{\lambda}$ by $\boldsymbol{\lambda} \times \boldsymbol{\lambda_m}$ in the WBG algorithm (in annex), with $\boldsymbol{\lambda_m} = \mathbf{0,999^m}$ for example, $\mathbf{m}$ being the counter of the loop. The idea for this addition was motivated by the relationship between the learning rate and the number of iterations required to obtain the optimal solution. Indeed, a low rate leads to an increase in the number of trees but generally leads to an improvement in the quality of prediction. Thus, the degressive factor $\boldsymbol{\lambda_m}$ allows the model to have a high learning coefficient for the first iterations and then allows it to learn slowly over the iterations. For that reason, $\boldsymbol{\lambda_m}$ is searched in the interval [0.99; 1] in steps of 0.001.

The parameters to be optimized and the set of values where their optimal value has been sought are summarized.

Tab. 9. Parameters to be optimized and their areas of existence

Number of subdivisions for continuous variables	$nbins$	[500 ; 1 500]
Random selection of variables	col_sample_rate	[40 % ; 100 %]
Minimum observation per sheet	min_rows	{1, 2, 5, 10, 20}
Maximum tree depth	K	[4 ; 8]
Randomly selected observations	p	[50 % ; 100 %]
Learning Coefficient	λ	[0,05 ; 0,1]
Degressive factor	λ_m	[0,99 ; 1]

Using a simple calculation and the number of possible values for each of the above parameters, there are 11×7×5×5×6×51×11 = 6,479,550 possible configurations. Going through all these combinations is not possible. The grid search will then be abandoned in favour of another method called "random grid search". This process has been theorized

by BERGSTRA and BENGIO [2012]. The basic argument is that at equivalent computing time, the random grid search can find efficient models over larger spaces than the grid search does. Moreover, BERGSTRA and BENGIO [2012] found that only certain parameters have a real impact on the performance of the algorithm. These observations justify not having to test all the values of the parameters that are not sufficiently influential, which saves a huge amount of computing time.

It is essential to define a stopping criterion for the random search method. This is either a function of the calculation time, or a function of the number of models calculated. In this study, it was chosen to set a maximum calculation time. Indeed, the more time is important, the more models will be calculated and the closer we get to the optimal solution.

Nevertheless, a 1-hour random search may yield a better optimal solution (in the sense of the MSE on the test sample) than a 4-hour random search. As an illustration, see the table below.

Tab. 10. Random grid search results according to computing time

Calculation time	Settings								MSE test
	col_sample_rate	λ	λ_m	K	nbins	min_rows	p	No. of trees	
1 hour	0,4	0,062	0,992	8	1 400	20	0,8	1 375	2 422 460
4 hours	0,4	0,072	0,996	8	1 500	1	0,8	2 792	2 428 957
12 hours	0,5	0,069	0,994	8	1 500	1	1	1 853	2 421 259

The table above gives the parameters of the best models obtained according to the number of hours performed by random search. The selected model is the one obtained after 12 hours of random search.

In order to achieve better results, the resulting model has been modified. The latter is motivated by RIDEGWAY [2007], which reports a significantly better result when a 5-sample cross-validation is performed. The principle is as follows: first aggregate the learning and validation samples, then divide the resulting sample into 5 samples of the same size, finally each of the 5 samples will serve as a validation sample while the other 4 aggregated samples will serve as a learning sample. The stopping criterion used is the same as above.

Cross-validation is most effective by lowering the[16] *boosting* MSE on the test sample to 2,411,535.

[16] However, a major disadvantage of this method is its computing time. For example, it took 36 minutes and 7 seconds longer than the simple WBG (which took 1 minute and 31 seconds to complete).

In summary, compared to the simple WBG which has an MSE on the test sample of 2,477,139, the optimized WBG now has an MSE of 2,411,535. This represents a decrease of 2.7%, which is a significant gain. In addition, a comparison was also made with the L1 standard and it was found that the predictions from the optimized WBG are about 1.51% more accurate than those from the simple WBG. This attests to the fact that a better quality model has been obtained and therefore to the usefulness of all the processes applied to obtain it.