31 May – 03 June 2016
at ISEG – Lisbon School of Economics
and Management

Modeling the structural component of lapse rate : Nonlinear approximation and models aggregation

Salma Jamal

Actuarial consultant — KPMG

This presentation has been prepared for the ASTIN Colloquium Lisboa 2016.
IAP wishes it to be understood that opinions put forward herein are not necessarily those of the Institute and the Council is not responsible for those opinions.

Table of contents

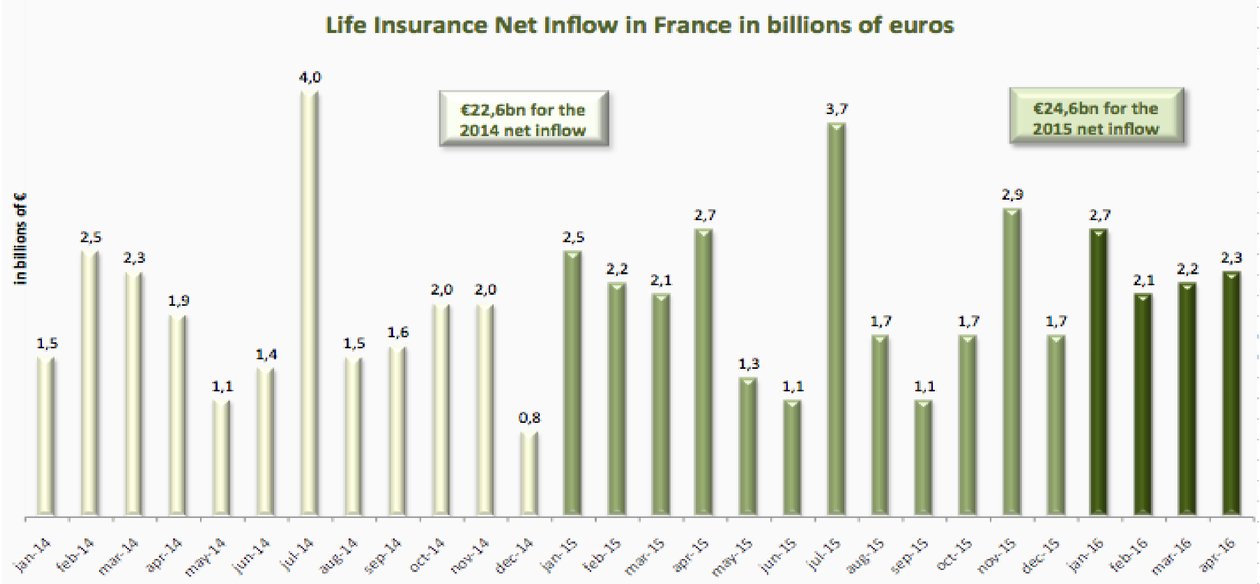
1. General framework
   - Topic and purpose of the study
2. Scope and data processing
   - Perimeter of the study
   - Missing data imputation
3. Portfolio description
   - Factor Analysis of Mixed Data (FAMD)
4. Lapse rate analysis
   - Regression trees
   - Neural networks
   - Models aggregation
5. Models Validation
6. Conclusion

General framework

Topic and purpose of the study

A thriving market

The outstanding amounts of Life Insurance are not static : emergency cash needs, an evolving fiscal framework, new products onto the market,... and figures attest to this. In France, for the only month of April 2016, the Life Insurance net inflow was close to 2.3 billions of euros.



Evolution of the Life Insurance net inflow in France January 2014 — February 2016 (Press Conference of the Association Française de l'Assurance 28/01/2016)

Lapses

In this market, individual contracts have, among others, the following particularity : policyholders can recover all or part of their initial contract value, eventually reduced by contractual penalties and eventually increased by a guaranteed rate. **Lapse** designates this operation.

Topic and purpose of the study

As part of the European directive Solvency II, accuracy prevails on caution, especially for the lapse risk. To meet this solvency requirement, insurers should harness in an optimal way the information contained by their data.

Purpose of the study :

Modeling a Best-Estimate lapse law with a process which highlights 3 issues :

- the necessity to model lapse rates in a Best Estimate way to fulfill Solvency II requirements ;
- the use of mathematical methods which can approach heterogeneous data so that we avoid the bias generated from a variables conversion ;
- the added-value given by the use of nonlinear approximation after testing a one dimensional approach.

Scope and data processing

Perimeter of the study

Perimeter

For this study, we have analyzed a portfolio of Individual Savings contracts. These contracts were redeemed between 2005 and 2010 and on 2013 (as 2011 and 2012 data could not be tapped). The study was realized in two times :

- **Description phaze** : common to the whole portfolio.
- **Lapse rate analysis** :
  - **Models building** Contracts redeemed between 2005 and 2010 ;
  - **Models validation** Contracts redeemed on 2013.

Variables used :

- Policyholders ages
- Gender
- Customer segment (marketing variable)
- Product range
- Seniority of the contract
- Tax category
- Multi-investment vehicle contract or one investment vehicle (in euros) ;
- Minimum Guaranteed rate (MGR) of each contract ;
- Lapses amounts and types
- Mathematical reserves (at the beginning of each period)

Missing data imputation

Methods for the imputation of heterogeneous missing data :

- **LOCF (Last Observation Carried Forward)** Each value is replaced by the last known value of the concerned variable.
- **kNN (k Nearest Neighbors)** Each missing data is replaced by the mean of the values of its k nearest neighbors. This method is based on the distance (the Euclidean distance or the one of Mahalanobis) between the missing data and the observed ones.
- **MICE (Multiple Imputation by Chained Equations)** [5] The main assumption of this method is the probability of missing data of one variable given the state of the other variables of the database.
- **Random Forest** [2] : A first rudimentary imputation is made with any imputation method. A group of random forests is then gradually adjusted until a stopping criterion is reached.

Missing data imputation

Imputation steps :

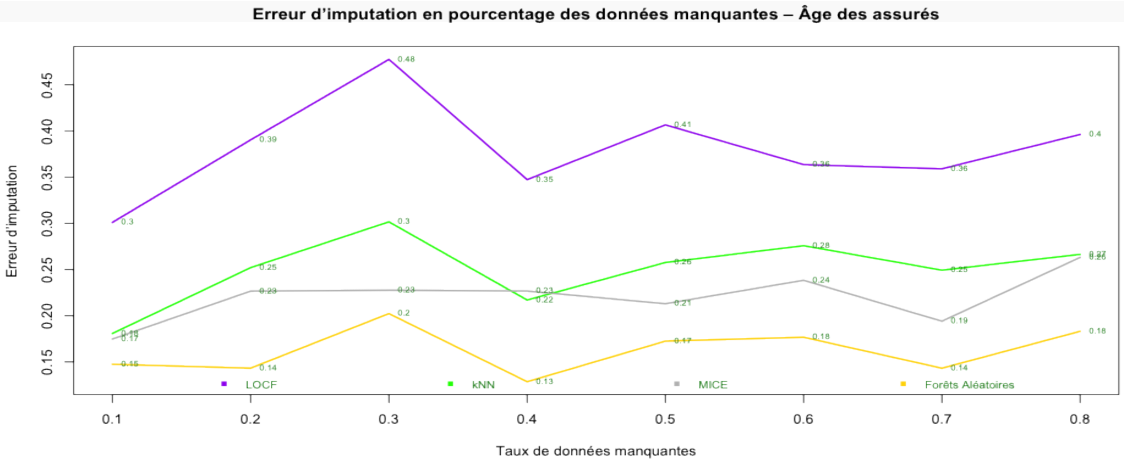
1. We first suppress each row that contains missing data ;
2. From the new complete database, we randomly generate missing data.
3. We apply the methods presented above to the new database ;
4. After the imputation of missing data, we assess the error of imputation by measuring the difference between imputed data and those suppressed. The assessment of the imputation error, in the case of :
   - Qualitative variables, consists in minimizing the Hamming distance :

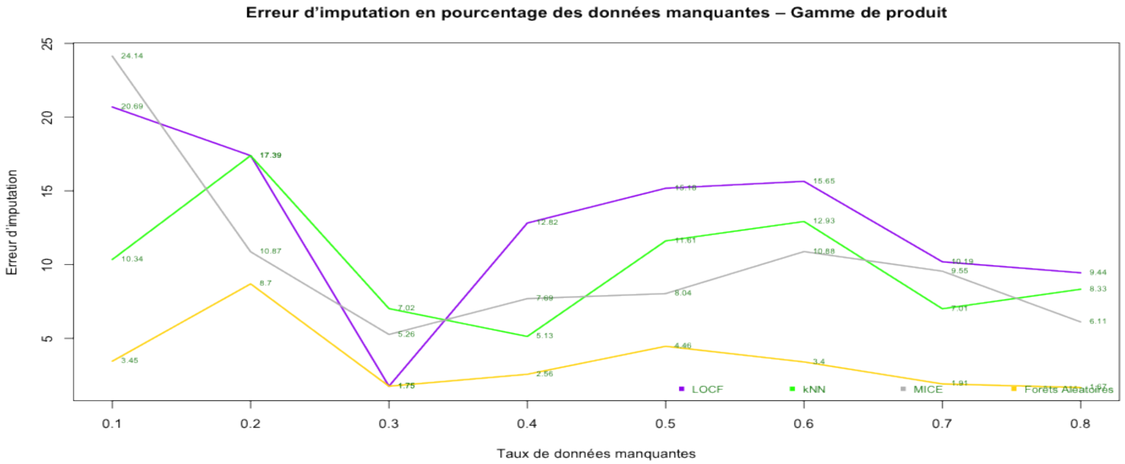
     $$d_H\left(\hat{T}_i, T\right) = \frac{\sum_{j=1}^{p} \mathbb{1}_{\{\hat{T}_{i_j} \neq T_{i_j}\}}}{\frac{Card\, T_i}{100}}$$

     where T is the $n \times p$ data set, T_{i_j} the missing categories and $\hat{T}_{i_j}$ the imputed ones.
   - Numeric variables, consists in minimizing the absolute value of the difference between the initial data and the imputed ones.

Missing data imputation

Example : Policyholders ages and products ranges :





Curves :

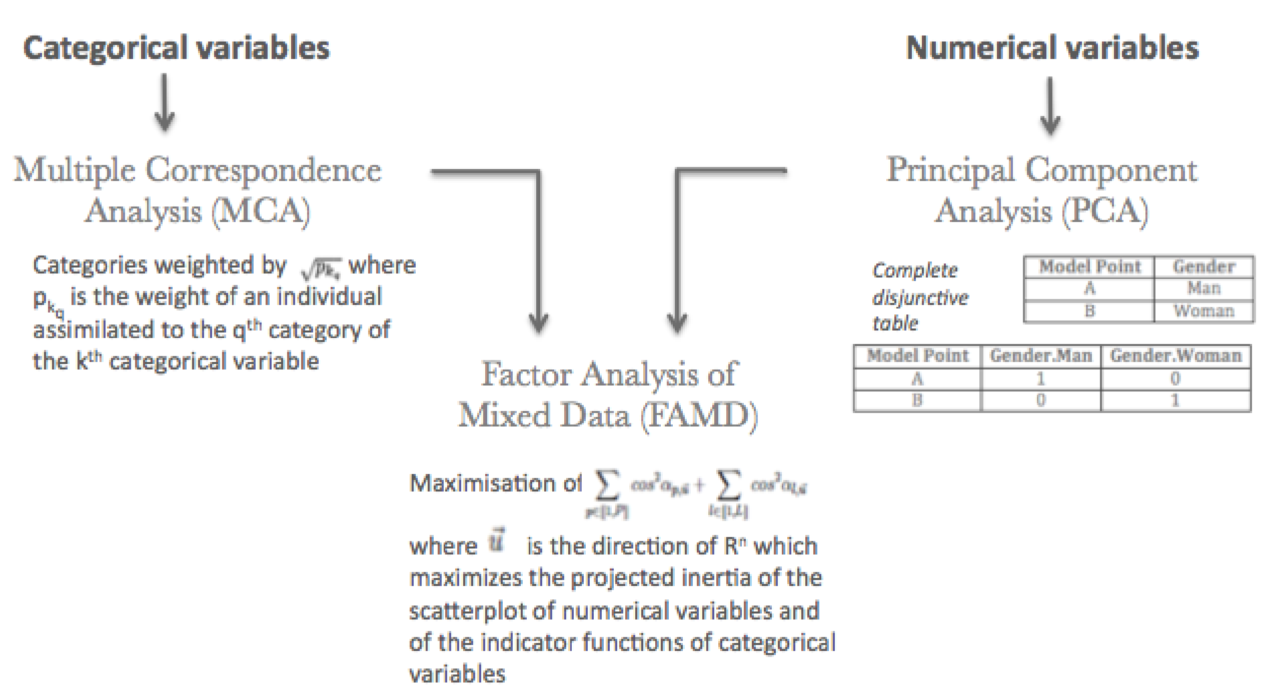
🟪	LOCF	◼	MICE
🟩	kNN	🟨	Forêts aléatoires

The **Random forest** imputation method minimizes, overall, the error due to imputation. To optimize its results, we make a first imputation by the kNN method before applying the random forest one.

Portfolio description

Factor Analysis of Mixed Data (FAMD)

FAMD principle :



Categorical variables

Multiple Correspondence Analysis (MCA)

Categories weighted by $\sqrt{p_{k_q}}$ where p_{k_q} is the weight of an individual assimilated to the q^{th} category of the k^{th} categorical variable

Numerical variables

Principal Component Analysis (PCA)

Complete disjunctive table

Model Point	Gender
A	Man
B	Woman

Model Point	Gender.Man	Gender.Woman
A	1	0
B	0	1

Factor Analysis of Mixed Data (FAMD)

Maximisation of $\sum_{p\in[1,P]} \cos^2\alpha_{p,\vec{u}} + \sum_{l\in[1,L]} \cos^2\alpha_{l,\vec{u}}$ where $\vec{u}$ is the direction of R^n which maximizes the projected inertia of the scatterplot of numerical variables and of the indicator functions of categorical variables

The Multiple Correspondence Analysis (MCA) and the Principal Component Analysis (PCA) give an interesting theoretical basis to tackle with heterogeneous data. The Factor Analysis of Mixed Data (FAMD) [4] is a technique which applies the reasoning of an MCA to a PCA, combining their advantages and reducing their shortcomings.

One of the most important advantages of the FAMD is that **whatever the number of a qualitative variable categories, its weight will be the same than the one of a numeric variable**. In fact, when a qualitative variable is projected on the subspace generated by the scatterplot of its categories, its inertia is always equal to one.

Lapse rate analysis

Regression trees

CART building steps [3]

- **Sampling** :

Samples	Proportion of the initial basis
Learning	80%
Test	20%
Validation	Recent basis : 2013 data

- **Segmentation rules** : Tree segmentation is performed by successive choices of contributory variables to the explanation of the variable of interest. The ramification stopping criterion is defined from indicators of the model accuracy. In the case of a CART model, the Gini coefficient is one of these possible indicators :

$$I_{Gini} = \sum_{i \neq j} P(i|t)P(j|t) \qquad (1)$$

where $P(i|t)$ represents the proportion of the i^{th} class elements assigned to the t^{th} node.

- **Individuals assignment** Individuals are then distributed on the tree branches. This distribution can be carried out by minimizing misclassification costs which are directly included in Gini coefficient computation :

$$I_{Gini} = \sum_{i \neq j} C_i(j)P(i|t)P(j|t) \qquad (2)$$

where $C_i(j)$ represents the misclassification cost of the i^{th} class into the j^{th} class.

- **Tree pruning** Tree pruning consists in lightening the tree from the irrelevant leafs for the tree analysis. For this study, we opt for the standard deviation rule, established by Breiman and al.(1984), which consists in reducing the complexity parameter (cp) of the final tree.

Regression trees

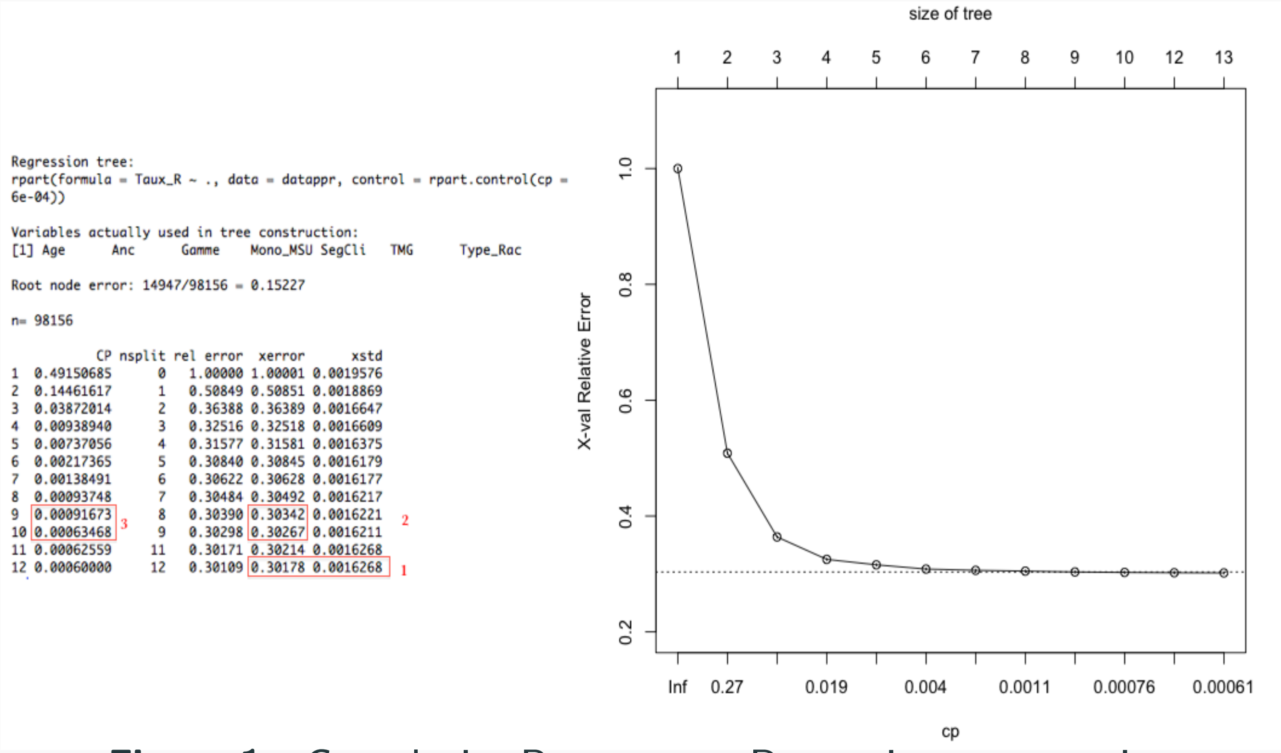
Complexity Parameter

In our case (Figure 1), the cp increases when the learning error (*rel error*) increases and when the nodes number (size of the tree) decreases.

Breiman rule recommands an error threshold less than 0,30178 + 0,0016268 = 0,3034068

In other words, a cp between 0,00063468 and 0,00091673.

- We chose, thus, a cp equal to 0,0007



```
Regression tree:
rpart(formula = Taux_R ~ ., data = datappr, control = rpart.control(cp =
6e-04))

Variables actually used in tree construction:
[1] Age      Anc      Gamme    Mono_MSU SegCli   TMG      Type_Rac

Root node error: 14947/98156 = 0.15227

n= 98156

           CP nsplit rel error  xerror      xstd
1  0.49150685      0   1.00000 1.00001 0.0019576
2  0.14461617      1   0.50849 0.50851 0.0018869
3  0.03872014      2   0.36388 0.36389 0.0016647
4  0.00938940      3   0.32516 0.32518 0.0016609
5  0.00737056      4   0.31577 0.31581 0.0016375
6  0.00217365      5   0.30840 0.30845 0.0016179
7  0.00138491      6   0.30622 0.30628 0.0016177
8  0.00093748      7   0.30484 0.30492 0.0016217
9  0.00091673 3    8   0.30390 0.30342 0.0016221  2
10 0.00063468      9   0.30298 0.30267 0.0016211
11 0.00062559     11   0.30171 0.30214 0.0016268
12 0.00060000     12   0.30109 0.30178 0.0016268  1
```

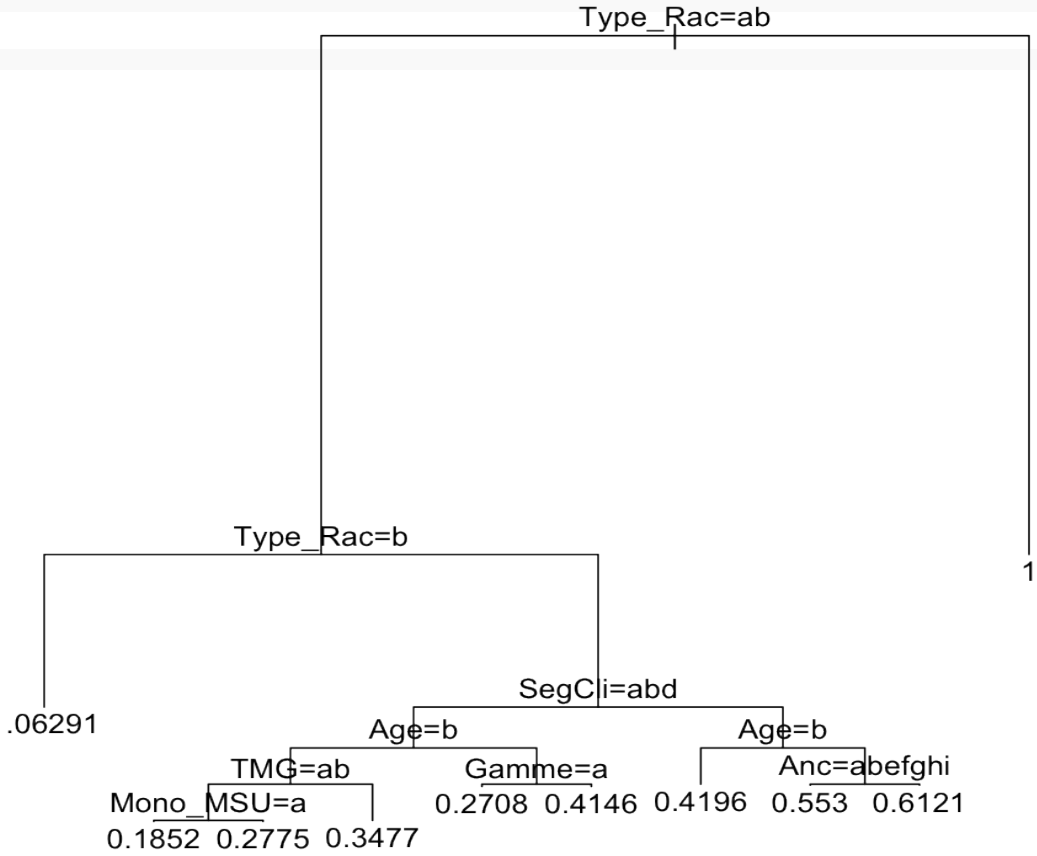
Figure 1: Complexity Parameter - Regression tree pruning

Regression trees

Tree pruning

The tree retained is composed of 9 segmentations, which correspond to 10 leafs instead of 13 for the initial tree.

The figures shown at each leaf correspond to a mean value of the lapse rate for the cohorts identified by the tree. Realistic rates can be obtained by computing, for the identified groups of risk, the lapse risk, starting from the initial lapse amounts.



Pruned regression tree

Neural networks

Formalism

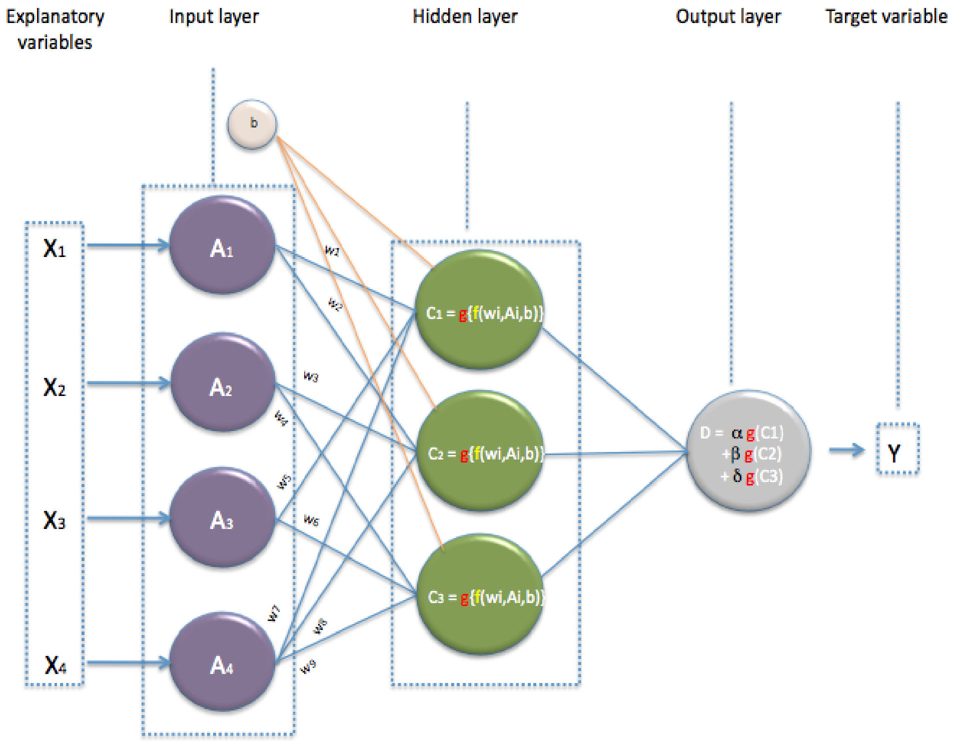
A neural network is composed by an input layer, an output layer and eventually one or many hidden layers **It is the presence of hidden layers that gives a nonlinear dimension to neural network outputs.**. Each layer contains neurons. Input layer neurons are the explanatory variables of a database.

The output layer neurons depends on the type of the network : if the network is predictive, there are as many neurons in the output layer as target variables (or as categories of the target variables if these ones are qualitative). The link between two neurons is called **synaptic weight**.

At the level of the input layer, each node is connected to synaptic weights. Weights are then linked to the nodes of the following layer via a **combination function**.

An **activation function** will, from these sums, assess the activity of each node. Finally, **biases**, also called *neurons threshold activation*, can exist at each node.

The figure below shows a simple neural network with an input layer with 4 entries, an output layer linked to one interest variable and a hidden layer with 3 neurons. The combination function is noted **f** while the activation function is noted **g**



Elementary neural network {*4 3 1*}

Neural networks

Retained Neural Network

In this study, we have focused on the case of the **Multilayer Peceptron using the Backpropagation Algorithm** [8] (noted **MLP-BA**).

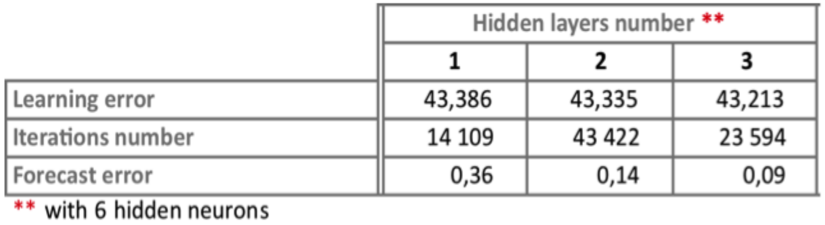
MLP-BA building steps

- **Data processing** :
  - We apply the logarithm function and standardize numeric variables. The logarithm function limits the crushing of the average values by potential extreme values ;
  - We create binary variables for the qualitative variables (complete disjunctive table) ;
  - Discrete variables are transformed by replacing each i^{th} value by $\sum_{j=0}^{i} 2^{-j}$.
- **Weights initialization :** this initialization is made randomly since weights are adjusted during the learning step by the Backpropagation Algorithm.
- The BA is applied to the learning sample. During this step, the algorithm :
  - applies the combination and the activation functions to the inputs and the hidden layers ;
  - assesses the cost function of the model ;
  - back-propagates the global error to compute the individual errors of the model ;
  - adjusts the neurons weights starting from the neurons of the output layer to the ones of the inputs layer ;
  - repeats the operation until reaching a stopping criterion (threshold for the partial derivatives of the error function) or a specific number of iterations ;
  - applies the retained model to the test sample to verify its generalization capacities.

Neural networks

Sensitivity of a MLP-BA

- *Sensitivity of the MLP-BA to the number of hidden layers*

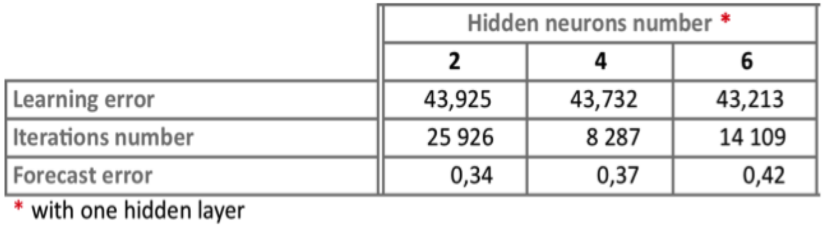


	Hidden layers number **		
	1	2	3
Learning error	43,386	43,335	43,213
Iterations number	14 109	43 422	23 594
Forecast error	0,36	0,14	0,09

** with 6 hidden neurons

The 1st test concerns the **number of hidden layers**. We increase it from 1 to 3. For each hidden layer, the number of neurons is set to 6 : when the network consists of 2 (respectively 3) hidden layers, each one consists of 3 (respectively 2) neurons.

- *Sensitivity of the MLP-BA to the number of hidden neurons*

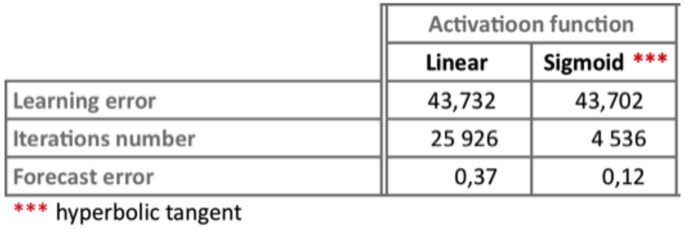


	Hidden neurons number *		
	2	4	6
Learning error	43,925	43,732	43,213
Iterations number	25 926	8 287	14 109
Forecast error	0,34	0,37	0,42

* with one hidden layer

The 2nd test concerns the **number of hidden neurons**. Starting from one hidden layer, we increase the number of hidden neurons from 2 to 6, by 2. This test is tricky because there are many theories on the effect of the hidden neurons number on the generalization capacities of a neural network.

- *Sensitivity of the MLP-BA to the activation function*

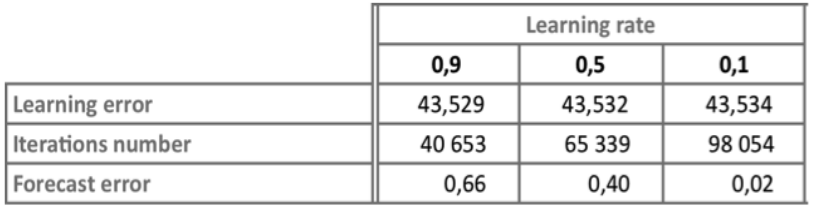


	Activatioon function	
	Linear	Sigmoid ***
Learning error	43,732	43,702
Iterations number	25 926	4 536
Forecast error	0,37	0,12

*** hyperbolic tangent

The 3rd test concerns the choice of the **activation function**. As mentioned earlier, sigmoid function outputs are nonlinear. Such a function lets the learning error decreases but, in particular, reduces the number of iterations before the model stabilization : the time saving is 5 times better than in the case of a linear function.

- *Sensitivity of the MLP-BA to the learning rate*



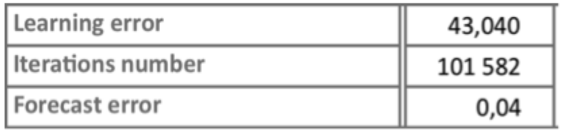
	Learning rate		
	0,9	0,5	0,1
Learning error	43,529	43,532	43,534
Iterations number	40 653	65 339	98 054
Forecast error	0,66	0,40	0,02

The 4th test concerns one of the most sensitive parameters of a neural network : the **learning rate**. This rate represents the speed with which the model crosses the whole database.

Neural networks

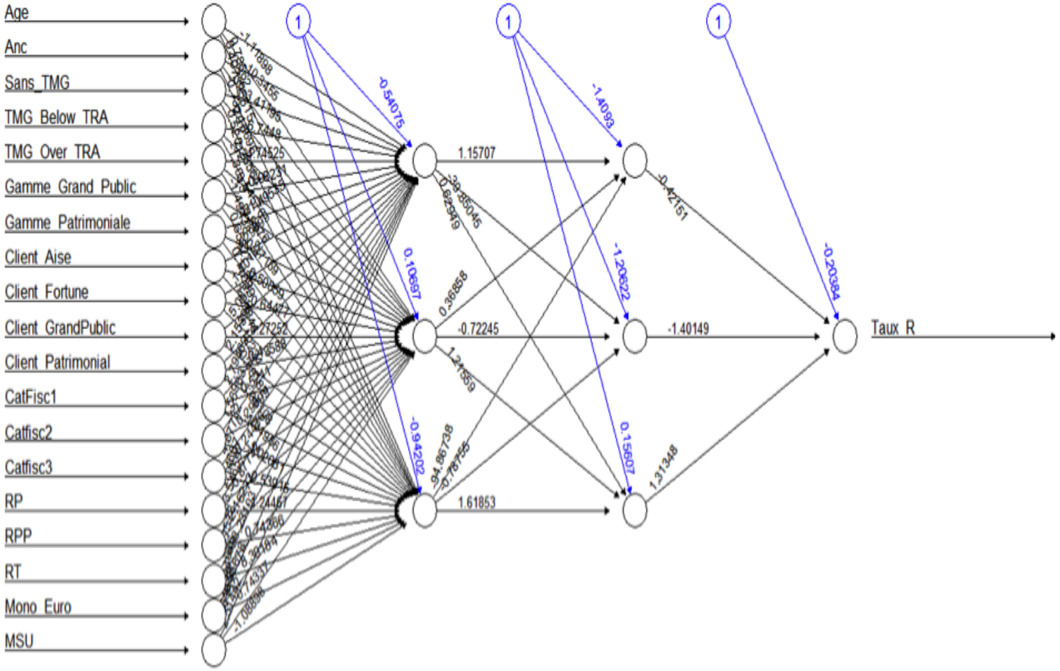
Retained MLP-BA

The learning step leads us to choose a model with 2 hidden layers with 6 hidden neurons (3 per hidden layer). The sigmoid function was selected as an activation function and the learning rate was set to 0.25.



Learning error	43,040
Iterations number	101 582
Forecast error	0,04

Final adjustment - MLP-BA {3 3}



MLP-BA with 2 hidden layers, with 6 hidden neurons

Models aggregation

Some models aggregation methods

The quality of a statistical model is linked to its accuracy and to its sturdiness. Increasing the first one consists in decreasing the bias of the model. Increasing the second one : in decreasing its variance.

Models aggregation allows to strengthen these aspects, especially in the case of sensitive models like regression trees and neural networks. [1].

We present here the most popular aggregation methods : bagging, a particular bagging method : random forests, and boosting.

Bagging Bagging is a random process of models aggregation developed by Breiman on 1996 [3]. After a first classification or regression step, the bagging find the average model which fit the best the variables of interest. The model selected by the bagging algorithm is not the one with the best performances among all the constructed models but its performances are better than the average ones.

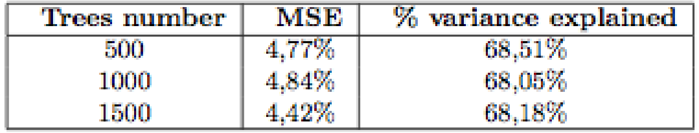
Random forest : A random forest is an aggregation of decision trees. The specificity of a random forest, comparing to a classic bagging algorithm, is the additional step of a random draw on the whole explanatory variables.

Boosting Developed on 1990 by Schapire [7], than by Freund et Shapire on 1996 [6], the Boosting algorithm is a deterministic (most of the time) and an adaptive process, that is to say : at each iteration, the model is a an adapted version of the previous one.

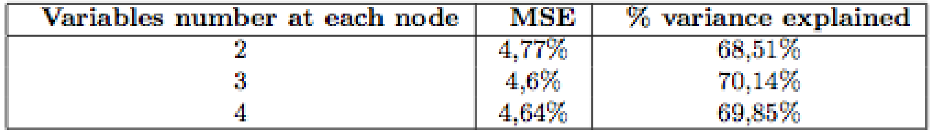
Models Aggregation

Random forests

A random forest is mainly function of the forest trees number and of the number of variables at each node of the trees forest. We vary these parameters and observe the stability of the forest thanks to the percentage of the variance explained and to the Mean Squared Error (MSE).

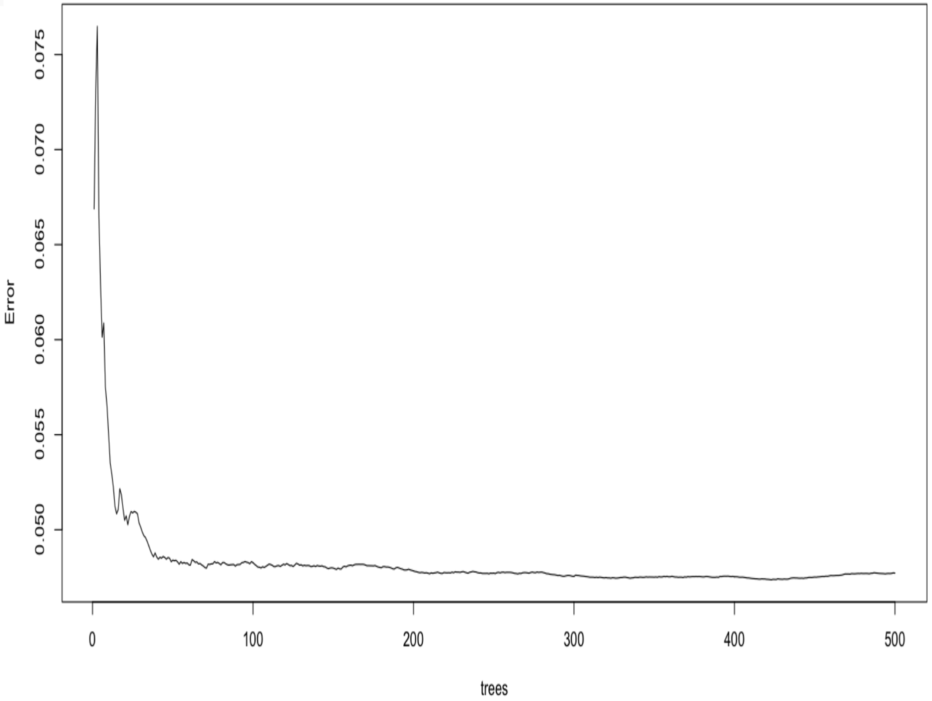


Trees number	MSE	% variance explained
500	4,77%	68,51%
1000	4,84%	68,05%
1500	4,42%	68,18%



Variables number at each node	MSE	% variance explained
2	4,77%	68,51%
3	4,6%	70,14%
4	4,64%	69,85%

The forest stability began from the threshold of 300 trees. However, the variation of the variables number at each node does not affect significantly the MSE and the explained variance. We set it to 3 as it reduces the MSE and maximizes the explained variance.

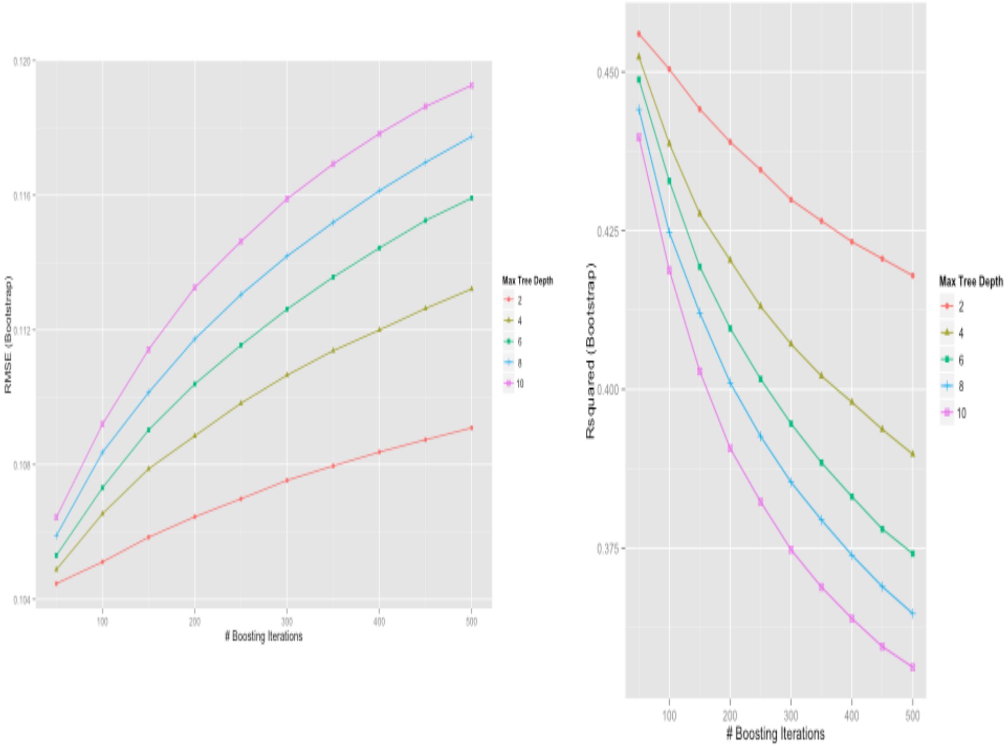


Random forest - Adjustment of parameters

Models Aggregation

Regression trees boosting

- Regression trees boosting is sensitive to the forest trees number, to the depth of the forest (i.e. the number of nodes per segmentation, at each tree of the forest) and to the learning rate.
- In the case of a regression, the model computes the Root Mean Squared Error (RMSE) and the R^2 at each iteration ;
- The algorithm selects then the model which reduces this error.



Regression trees Boosting - RMSE and R^2

Models Aggregation

Neural Networks aggregation

The neural networks aggregation requires the definition of :

- the learning algorithm number of iterations ;
- the network size (that means the number of intermediary layers and the one of hidden neurons) ;
- the **decay**. This last one is a controlling factor which penalizes weights (the network synapses) by forcing them to decrease exponentially to 0.

Reducing the overlearning is directly linked to the decrease of the hidden neurons number and to the increase of the decay.

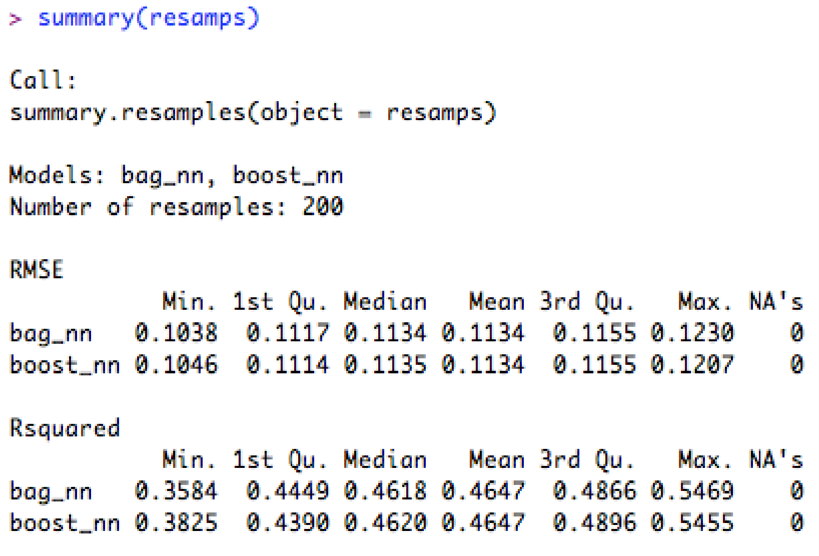
We choose the following values :

Iterations number	Hidden neurons number	Decay
200	6	0.9

Models Aggregation

Neural Networks aggregation

RMSE and R^2 levels of bagging and boosting are very close and present nearly similar values to those of regression trees boosting.



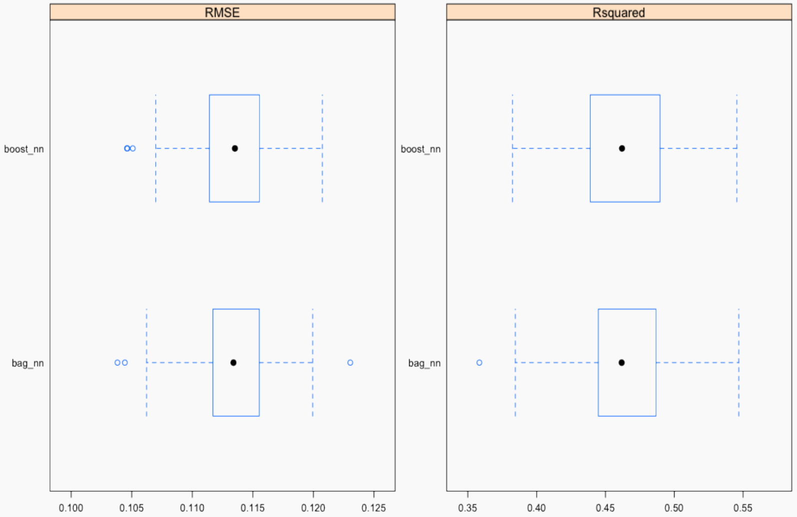
```
> summary(resamps)

Call:
summary.resamples(object = resamps)

Models: bag_nn, boost_nn
Number of resamples: 200

RMSE
           Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
bag_nn   0.1038  0.1117 0.1134 0.1134  0.1155 0.1230    0
boost_nn 0.1046  0.1114 0.1135 0.1134  0.1155 0.1207    0

Rsquared
           Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
bag_nn   0.3584  0.4449 0.4618 0.4647  0.4866 0.5469    0
boost_nn 0.3825  0.4390 0.4620 0.4647  0.4896 0.5455    0
```



Neural networks aggregation - Comparison between Bagging and Boosting - RMSE and R^2 boxplot

Models Aggregation

Forecast error

The least important forecast error is the one of **neural networks bagging**.

Regression trees		
Initial model	**Bagging**	**Boosting**
10,4%	3,18%	3,6%

Neural networks		
Initial model	**Bagging**	**Boosting**
4,43%	1,97%	3,88%

Models Validation

Models Validation

The selected models have to be traced over time. This follow-up enables to judge the consistency of the evolution of risk groups behavior and to control the sturdiness of these models.

Implementing the selected models on a recent database leads to consistent results in view of the portfolio population aging. It also lets us perceive trends reflecting commercial practice.

Forecast error

In the case of a recent database, the least important forecast error is the one of **regression trees boosting** :

Regression trees		
Initial model	**Bagging**	**Boosting**
5,26%	5,65%	2,32%

Neural networks		
Initial model	**Bagging**	**Boosting**
3,67%	3,02%	4,33%

Conclusion

Conclusion

All along this paper, we have focused on different statistical methods to the Lapse rate modeling. The structural component of the lapse was the main topic of this study but the method used are appropriate to any variable.

The calibration of our models have respected theoretical basis but, above all, remained connected to the purpose of this study : the identification of risk groups reflecting an economic reality.
As the linear approach was restrictive, we adopted specific methods of *machine learning*, commonly used in the Non-Life sector.

- Advantages
  - Forecast errors are often very low (after a meticulous calibration phase) ;
  - Detection of nonlinear links ;
  - Neural networks parcimony ;
  - Adaptation to big and heterogeneous data.
- Limits
  - Nonlinear models are sensitive to local minima and to overlearning ;
  - First calibration is tricky and implementation is time-consuming.

And then .. ?

- Our conclusions lead to define new *Model Points* at the *Asset and Liability Management* Department of a french insurer. As part of sensitivity tests, these *Model Points* were shocked to assess the impact on profitability indicators (such as The Return-on-Equity) ;
- The Factor Analysis could be followed by a hierarchical clustering to perform a first segmentation of explanatory variables ;
- We could also remain in the theme of neural networks by designing a self-organizing map (Kohonen Map) as a first part of this paper ;
- It could be interesting to model lapse rates with other learning methods such as Support Vector Machine (SVM) ;
- We could also develop an aggregation algorithm between bagging and boosting insofar as they gave us error rates relatively similar.

References I

O. D. et Maclin R.
Popular ensemble methods : An empirical study.
Journal of Artificial Intelligence Research 11, pages 169–198, 1999.

D. J. S. et P. Bülhman.
Missforest - nonparametric missing value imputation for mixed-type data.
Bioinformatics 28, pages 113–118, 2011.

R. O. e. C. S. L. Breiman, J. Friedman.
Classification and regression trees.
Wadsworth Brooks, 1984.

J. PAGES.
Analyse factorielle de données mixtes.
Revue de statistique appliquée, tome 52, n4, pages 93–111, 2004.

P. Royston.
Multiple imputation for nonresponse in surveys.
New York : Wiley, 2004.

References II

F. Y. B. P. . L. W. Schapire, R.
Boosting the margin : A new explanation for the effectiveness of voting methods.
Proceedings of the Fourteenth International Conference on Machine Learning. Nashville, TN, page 322–330, 1997.

R. Schapire.
The strength of weak learnability.
Machine Learning, 5(2), page 197–227, 1990.

P. Werbos.
Beyond regression : New tools for prediction and analysis in the behavioral sciences.
Ph. D. dissertation, Committee on Appl. Math., Harvard Univ., Cambridge, MA, 1974.