Surrender triggers in life insurance: what main features affect the surrender behavior in a classical economic context?

Xavier Milhaud, Stéphane Loisel, Véronique Maume-Deschamps

SURRENDER TRIGGERS IN LIFE INSURANCE : CLASSIFICATION AND RISK PREDICTIONS

X. Milhaud, S. Loisel and V. Maume-Deschamps [*]

Abstract - This paper shows that some policy features are crucial to explain the decision of the policyholder to surrender her contract. We point it out by applying two segmentation models to a life insurance portfolio : the Logistic Regression model and the Classification And Regression Trees model. Protection as well as Savings lines of business are impacted, and results clearly explicit that the profit benefit option is highly discriminant. We develop the study with endowment products. First we present the models and discuss their assumptions and limits. Then we test different policy features and policyholder's characteristics to be lapse triggers so as to segment a portfolio in risk classes regarding the surrender choice : duration and profit benefit option are essential. Finally, we explore the main differences of both models in terms of operational results and discuss about it.

I Introduction

Understanding the dynamics of the surrender (or lapse) rates is a crucial point for insurance companies. Several problems appear : first, policy lapse might make the insurer unable to fully recover her initial expenses (cost of procuring, underwriting, and issuing new business). Actually the insurer pays expenses at or before the time of issue but earns profits over the life of the contract, so she might incur losses from lapsed policies. Second, policyholders who have adverse health or other insurability problems tend not to lapse their policies, causing the insurer to experience more claims than expected if the lapse rate is high. This is also called the moral hazard and adverse selection : there remain only "bad risks" (Bluhm (1982)). Third, massive early surrender or policy lapse poses a liquidity threat to the insurer who is subjected to interest rate risk, because the interest rate is likely to change over the period of the contract. Imagine that financial events and a general loss of confidence of investors are at the origin of a high increase of the interest rate, say r_t plus a liquidity premium λ_t. Borrowing money in order to pay back the surrender value to the policyholder is thus more expensive for the insurer who could undergo a series of indesirable effects : no time to recover initial expenses, obligation to borrow at a high cost and finally necessity to liquidate assets at the worst moment. However, the surrenders are not always a bad thing for the insurer because policyholders renounce to some guarantees, which makes the insurer to earn money.

What causes lapses has attracted certain academic interest for some time. Originally two main hypotheses have been suggested to explain lapse behavior. The first one, the emergency fund hypothesis (Outreville (1990)), contends that policyholders use cash surrender value as emergency fund when facing personnal financial distress. Outreville (1990) develops an ordinary least square method for short term dynamics whose testable implication would be an increasing surrender rate during economic recessions. On the other hand, the interest rate hypothesis conjectures that the surrender rate rises when the market interest rate increases, because the

investor acts as the opportunity cost for owning insurance contracts.

When interest rates rise, equilibrium premiums decrease, so there is a greater likelihood that a newly acquired contract provides the same coverage at a lower premium. Indeed policyholders tend to surrender their policy to exploit higher yields (or lower premiums) available in the market. Another insight developed by Engle & Granger (1987) is to separate the potential long-term relationship between the lapse rate, interest rate and unemployment rate from their short-term adjustment mechanisms thanks to the cointegrated vector autoregression approach.

Modeling lapse behavior is therefore important for insurer's liquidity and profitability. The lapse rate on life policies is one of the central parameters in the managerial framework for both term and whole life products : assumptions about lapse rate have to be made in Asset and Liability Management, particularly for projections of the Embedded Value.

To design life products, managers and their teams assume an expected level of lapsation thanks to *data mining techniques*. But collecting just a part of the information from the observations prevents companies from getting to a maximum productivity, and to fully exploit the information is not so easy because of the data set complexity. For instance in a typical database of an insurance company there are missing data, mixtures of data types, high dimensionality, heterogeneity between policyholders. The challenge is thus to select salient features of the data and feed back summaries of the information.

The idea of this paper is to give clues to product designers (or managers) regarding the surrender risk thanks to the use of two complementary segmentation models : the Classification And Regression Trees (CART) model (Breiman et al. (1984)) and the Logistic Regression (LR) model (Hilbe (2009)). In the litterature, Kagraoka (2005) and Atkins & Gallop (2007) applied respectively the negative binomial and the zero-inflated models as counting processes, and

Kim (2005) applied the logistic regression model with economic variables to explain the lapses on insurance policies during the economic crisis in Korea. To the best of our knowledge, CART and LR have not been run with policy and insured's characteristics in this framework.

Our paper is organized as follows: we first present theoretical results about CART method that are useful for our practical problem. We more briefly recall the basics of logistic regression, as it has been more widely used in many fields. In Section IV, we analyze a real-life insurance portfolio embedding endowment contracts with these two methods and determine the main reasons for a policyholder to surrender, as well as predictors of the individual surrender probability. Numerical figures are given, both approaches are compared and their limits are discussed.

II   The CART model

The CART method was developed by Breiman et al. (1984) in order to segment a population by splitting up the data set step by step thanks to binary rules. It is an iterative and recursive flexible nonparametric tool, binary trees provide an illuminating way of looking at data and results in classification problems. The novelty of the CART method is in its algorithm to build the tree : there is no arbitrary rules to stop its construction, contrary to the previous uses of decision trees using stopping rules (see $A.1$). Depending on the studied problem, the two main goals of a classification process are to uncover the predictive structure of the problem and to produce an accurate classifier.

The opportunity to make predictions particularly with regression trees technique is also very useful; but CART should not be used to the exclusion of other methods.

A   The model

We present in this section how to construct the classification tree. Figure 1 shows the different stages to follow. The appendix details each of the steps and the underlying concepts.
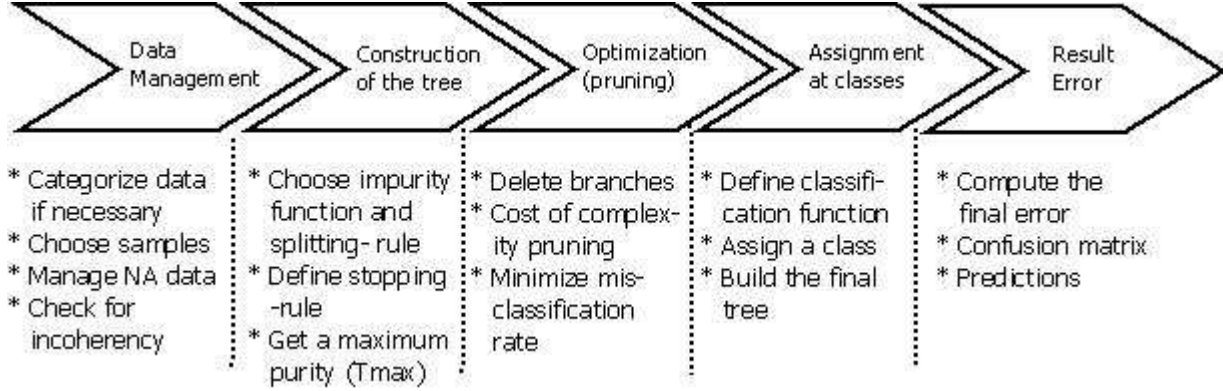
Figure 1: Ordered steps of CART procedure

A.1 Building the classification tree

Notation 1. *Let $\epsilon = (x_n, j_n)_{1 \leq n \leq N}$ be a sample of size N, where j_n are the observations of the outcome variable Y ($Y \in C = \{1, 2, ..., J\}$ and $x_n = \{x_{n_1}, x_{n_2}, ..., x_{n_d}\}$ the observations of X in $\mathbb{X}$ which are the d explanatory variables ($\mathbb{X} = \prod_{i=1}^{d} \mathbb{X}_i$ where $\mathbb{X}_i$ is a set of categorical or continuous variable).*

Definition 1. *Let*

- *$\forall x \in \mathbb{X}$, the classification process class(.,ϵ) classifies x in a group $j \in C$.*

- *The a-priori probability of group j is defined by $\pi_j = \frac{N_j}{N}$ where $N_j = card\{j_n | j_n = j\}$.*

- *Given $t \subset \mathbb{X}$ (t finite subset of $\mathbb{X}$), let us denote $N(t) = card\{(x_n, j_n) \in \epsilon, x_n \in t\}$.*

- *$N_j(t) = card\{(x_n, j_n) \in \epsilon, j_n = j$ knowing that $x_n \in t\}$.*

- *An estimator by substitution of P(j,t), denoted p(j,t), is given by $p(j,t) = \pi_j \frac{N_j(t)}{N(t)}$.*

- *An estimator by substitution of P(t), denoted p(t), is given by $p(t) = \sum_{j=1}^{J} p(j,t)$.*

- *P(j | t) is the a-posteriori probability of class j, is estimated by $\frac{p(j,t)}{p(t)} = \frac{N_j(t)}{N(t)} = \frac{p(j,t)}{\pi_j}$.*

How to begin ? The principle is to divide $\mathbb{X}$ into q classes, where q is not given *a-priori*. The method builds an increasing sequence of partitions of $\mathbb{X}$; the transfer from one part to another is given by the use of *binary (or splitting) rules* such as :

$$x \in t, \text{ for } t \subset \mathbb{X}.$$

For example, the first partition of $\mathbb{X}$ could be the sex. Here the policyholder whose characteristics are x is either a female or male, and t could be the modality "female".

Criterion 1. *These rules only depend on one "threshold" μ and one variable x_l, $1 \leq l \leq d$:*

- *$x_l \leq \mu, \mu \in \mathbb{R}$ in the case of an ordinal variable (if we have m distinct values for x_l, the set of possible sections card(D) is equal to M - 1);*

- *$x_l \in \mu$ where μ is a subset of $\{\mu_1, \mu_2, ..., \mu_M\}$ and μ_m are the modalities of a categorical variable (in this case the cardinal of the subset D of possible binary rules is $2^{M-1} - 1$).*

Actually we start with $\mathbb{X}$ called *root* which is divided into two disjoint subsets called *nodes* and denoted by t_L and t_R. Each of the nodes is then divided in the same way (if it has at least two elements !). At the end, we have a partition of $\mathbb{X}$ in q groups called *terminal node* or *leaf*. In the following, we denote by $\tilde{T}$ the set of *leaves* of the tree T; T^t is the set of *descendant nodes* of the ancestor node t in the tree T (see Figure 2).

The impurity concept The quality of the division from a node t to t_L and t_R is measured

thanks to the *impurity criterion*. This concept is explained in more details in Appendix II.1.
In our case, the impurity of a node t of a tree T is the quantity

$$impur(t) = g(p(1|t), p(2|t), ..., p(J|t)), \quad (1)$$

where g is an impurity function.
By consequence, the impurity of a tree T is

$$Impur(T) = \sum_{t \in \tilde{T}} Impur(t) \quad (2)$$

where Impur(t) = p(t)impur(t).
A binary rule Δ (or splitting-rule) of a node t gives $p_L = \frac{p(t_L)}{p(t)}$ observations in t_L and $p_R = \frac{p(t_R)}{p(t)}$ observations in t_R. We want to maximize the *purity variance* :

$$\delta \ impur(\Delta, t) = impur(t) - p_L impur(t_L)$$
$$- p_R impur(t_R) \quad (3)$$

Each time a split is made, the purity of the tree must increase : intuitively, it means that as many observations as possible should belong to the same class in a given node. The maximum decrease of impurity defines what splitting rule must be chosen.

Problem 1. $\delta \ impur(\Delta, t)$ *positive ? Or*

$$impur(t) \geq p_L \ impur(t_L) + p_R \ impur(t_R) \ ?$$

Solution 1. *The answer is always "yes" if g is concave.*

In our applications and in most of them, one uses the Gini index of diversity :

$$impur(t) = \sum_{j \neq k} p(j|t)p(k|t) \quad (4)$$

The Gini diversity index can be interpreted as a probability of misclassification. It is the probability to assign an observation selected randomly from the node t to class k, times the estimated probability that this item is actually in class j.
There also exits other impurity functions with an easier interpretation (see Appendix II.2) and there is no convincing justification for those

choices, except that they satisfy the conditions of an impurity function. Besides, the properties of the final tree are usually surprisingly insensitive to the choice of the impurity function ! For further explanations, see Breiman et al. (1984).

Dividing a node t The optimal division $\Delta_t^\star$ is given by

$$\Delta_t^\star = \underset{\Delta \in D}{argmax} \ (\delta \ impur(\Delta, t)), \quad (5)$$

where $\underset{\Delta \in D}{argmax} \ (\delta \ impur(\Delta, t))$ denotes the splitting rule Δ which maximizes $\delta \ impur(\Delta, t)$. At each step, the process is run in order to lower the impurity as fast as possible. Maximizing the gain in purity (homogeneity) dividing the node t is the same as maximizing the gain of purity on the overall tree T. Hence by dividing the parent node t into descendant nodes (t_L, t_R) with the rule Δ, one gets the more branched tree T' (see Figure 2) and from (2):

$$Impur(T') = \sum_{w \in \tilde{T} - \{t\}} Impur(w) + Impur(t_L)$$
$$+ Impur(t_R)$$

So the fluctuation of the impurity of the tree T is given by :

$$Impur(T) - Impur(T')$$
$$= Impur(t) - Impur(t_L) - Impur(t_R)$$
$$= \delta Impur(\Delta, t)$$
$$= p(t) \delta impur(\Delta, t) \quad (6)$$

Indeed, it results from the probability to be present in this node multiplied by the decrease of impurity given by the split Δ.

When to stop the splits ? Different rules exist to stop the division process. Some of them are natural, others are purely arbitrary and result from the choice of the user :

- obviously, the divisions stop as soon as the observations of the explanatory variables are the same in a given class (because it is not possible
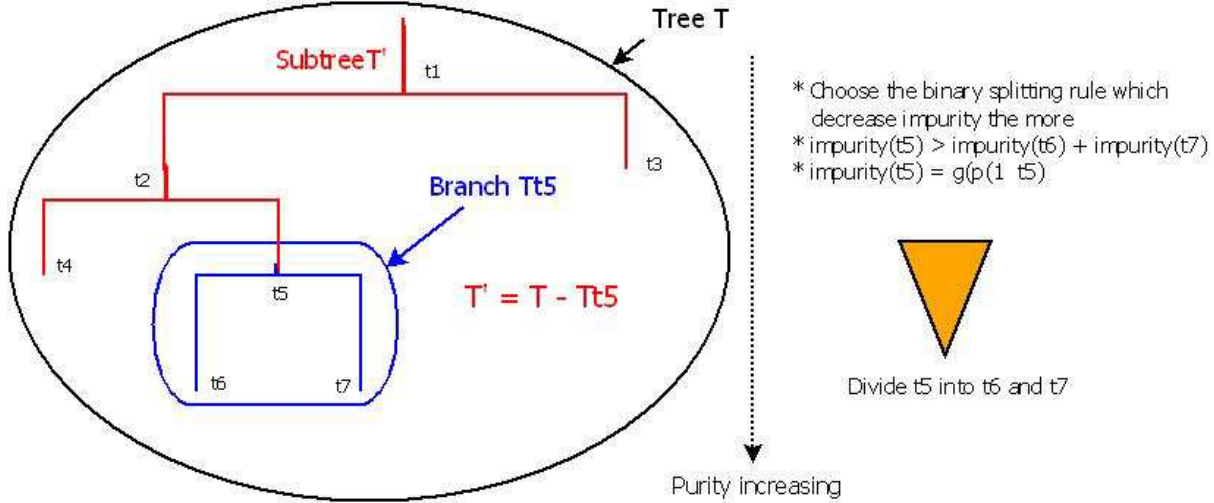
Figure 2: Construction of a binary tree

to go on splitting data !) ;

- define a minimum number of observations in each node. The smaller it is, the bigger the number of terminal nodes (leaves) is.

- choose a threshold λ as the minimum decrease of the impurity : let $\lambda \in \mathbb{R}_+^*$,

$$\max_{\Delta \in D} \delta \, Impur(\Delta, t) < \lambda \Rightarrow \text{stop the division.}$$

We will see that actually there is no stopping-rules with CART algorithm; we build the largest tree (T_{max}) and then we prune it.

A.2   The classification function

Problem 2. *The aim is to build a classification function, denoted by class(.,ϵ), such that*

$$
\begin{aligned}
class \quad &: \quad \mathbb{X} \to C \\
&x \to class(x, \epsilon) = j \\
with \quad &B_j = \{x \in \mathbb{X}; class(x, \epsilon) = j\}
\end{aligned}
$$

The idea is that we can class the policyholder (given its characteristics "x") in a set B_j to predict the result. This function must provide insight and understanding into the predictive structure of the data and classify them accurately.

Consider that the optimal tree has been built ; to know at what class the terminal nodes corre-

spond, use the following rule :

$$class(x, \epsilon) = \underset{j \in C}{argmax} \; p(j|t) \qquad (7)$$

In fact this is just the *Bayes rule* : it maximizes the *a-posteriori* probability of being in class j knowing that we are in the node t. This process defines the classification function, which then allows predictions. The estimation of classing an observation present in the node t in a wrong class (with respect to the class observed for this observation) is therefore

$$r(t) = 1 - class(x, \epsilon) = 1 - \underset{j \in C}{max} \; p(j|t), \quad (8)$$

Let the misclassification rate at node t be $\hat{\tau}(t) = p(t)r(t)$.

For each node of the tree, it represents the probability to be in the node t multiplied by the probability to wrongly class an observation knowing that we are in the node t.

It turns out that the general misclassification rate on the tree A is given by

$$\hat{\tau}(T) = \sum_{t \in \tilde{T}} \hat{\tau}(t) \qquad (9)$$

To put it in a nutschell, Figure 3 shows the four stages to be defined in the tree growing pro-

cedure. The last point is easy to define whereas others are much more difficult because of arbitrary choices and the necessity to adapt the questions to the problem. In fact, the CART method builds first the maximal tree T_{max} and then prune it. It enables to remove the arbitrary stop-splitting rules.

A.3 Prediction error estimate

The *prediction error* is assessed by the probability that an observation is classified in a wrong class by class(.,ϵ), that is to say :

$$\tau(class) = P(class(X, \epsilon) \neq Y)$$

The classification process, the predictor and its efficiency to get the final tree are based on the estimation of this error. The true misclassification rate $\tau^*(class)$ cannot be estimated when considering the whole data set to build the classification function. Various estimators exist in the litterature (Ghattas (1999)); and the expression of the misclassification rate depends on the learning sample chosen to run the study (some details and remarks are given in Appendix II.3).

Resubstitution estimate of the tree misclassification rate : the learning sample is the total sample of observations, ϵ. The part of observations wrongly classed by the function *class* is :

$$\hat{\tau}(class) = \frac{1}{N} \sum_{(x_n, j_n) \in \epsilon} \mathbb{1}\{class(x_n, \epsilon) \neq j_n\}$$

$$(10)$$

Achievements are overestimated because we finally class the same data (as those used to build the classification function) to test the efficiency of the procedure. Not surprisingly, this is the worse estimator in terms of prediction.

Test sample estimate : let W $\subset \epsilon$ be a witness (test) sample whose size is $N' < N$ (N is the size of the original data set ϵ). Usually $N' = \frac{N}{3}$ and so the size of the learning sample equals to $\frac{2}{3}N$. The test sample is used as in (10) :

$$\hat{\tau}^{ts}(class) = \frac{1}{N'} \sum_{(x_n, j_n) \in W} \mathbb{1}\{class(x_n, \epsilon) \neq j_n\}$$

$$(11)$$

The learning sample is used to build the classifier *class* and the test sample is used to check for the accuracy of *class*. This estimator is better but requires a larger initial data set.

By cross-validation : suppose that the original sample ϵ is divided into K disjointed subgroups $(\epsilon_k)_{1 \leq k \leq K}$ of same size and let us define K new learning data sets such that $\epsilon^k = \epsilon - \epsilon_k$. We can build a classification function on each sample ϵ^k such that $class^k(.) = class(., \epsilon^k)$. One still uses the same idea :

$$\hat{\tau}^{cv}(class) = \frac{1}{N} \sum_{k=1}^{K} \sum_{(x_n, j_n) \in \epsilon_k} \mathbb{1}\{class(x_n, \epsilon^k) \neq j_n\}$$

$$(12)$$

This technique is highly recommended when we do not have a lot of data available.

Notation 2. $\tau(T)$ *is the prediction error on T* ; $\hat{\tau}(T)$, $\hat{\tau}^{ts}(T)$ *and* $\hat{\tau}^{cv}(T)$ *its estimations.*

B Limits and improvements

The classification tree method offers some interesting advantages like no restriction on the

---

1. a set of binary questions like { is $x \in S$?}, $S \in \mathbb{X}$,

2. an impurity function for the goodness of split criterion,

3. a stop-splitting rule (or not, natural stopping-rule is then one case by leaf),

4. a classification rule to assign every terminal node to a class.

Figure 3: Necessary stages for the tree growing procedure

type of data (both categorical and numerical explanatory variables accepted), the final classification has a simple form and can be compactly stored and displayed.

By running the process to find the best split at each node, the algorithm does a kind of automatic stepwise variable selection and complexity reduction. In addition, one can transform ordered variables without changing the results if the transformation is monotonous. Moreover CART is not a parametric model and thus do not require a particular specification of the nature of the relationship between the outcome and the predictor variables, it successfully identifies interactions between predictor variables and there is no assumption of linearity.

However, the splits are on single variables and when the class structure depends on combinations of variables, the standard tree algorithm will do poorly at uncovering the structure. Besides, the effect of one variable can be masked by others when looking at the final tree. To avoid this, there exists solutions as ranking the variables in function of their potential : this is what is called the *secondary* and *surrogate splits* (also used with missing data, see Breiman et al. (1984)). There exists other difficulties, particularly :

- sometimes the final tree is difficult to use in practice because of its numerous ramifications. The more you split the better you think it is, but if one sets the stop-splitting criterion so as to get only one data point in every terminal node, then the estimation of the misclassification rate would not be realistic (equal to 0 because each node is classified by the case it contains, one overfits the data);

- the CART method provides a way to have an idea of the prominence of each explanatory variable. As a matter of fact, reading the final tree from the root to the leaves gives the importance of variables in descending order. But Ghattas (2000) criticizes the bad reliability of the method : a small modification of the data sample can cause different classifiers,

a big constraint to make predictions because of its instability.

For sure, we do not want this kind of behaviours because it means that a variable could be considered very important with a given data set, and be absent in the tree in another quasi-similar one ! The first point can be solved thanks to the introduction of a complexity cost in the pruning algorithm (see Appendix II.5) and the second one using cross-validation, learning and test samples (see *A*.3), *bagging predictors* or *arcing classifiers*.

C Bagging predictors

The bad robustness of the CART algorithm when changing the original data set has already been discussed. This can cause to experiment different optimal final classifiers, but this drawback can be challenged using resampling techniques.

The bootstrap is the most famous of them (sample N cases at random with replacement in an original sample of size N), and the bagging is just a bootstrap aggregation of classifiers trained on bootstrap samples. Several studies (Breiman (1996), Breiman (1994) and Breiman (1998)) proved the significance and robustness of *bagging predictors*. The final classifier assigns to an observation the class which has been predicted by a majority of "bootstrap" classifiers. This classifier cannot be represented as a tree, but is extremely robust.

The "Random Forest" tool was developed by Breiman (2001) and follows the same idea as bagging predictors : this is a combination of tree predictors such that each tree is built independently from the others. The final classification decision is obtained by a majority vote law on all the classification trees, the forest chooses the classification having the most votes over all the trees in the forest. The larger the number of trees is, the more the ability of this algorithm is good (until a certain number of trees). We usually speak about the *out-of-bag* error when using Random Forest algorithm : it represents for each observation the misclassification rate

of predicted values of the trees that have not been built using this observation in the bagging scheme. This error tends to stabilize to a low value.

The bagging method can be implemented with the `randomForest` R package[1]. It offers the opportunity to compute the importance of each explanatory variable, that is why we prefer to use it in our applications instead of the `ipred` package[2].
For more precision on these theories, refer to Breiman et al. (1984).

III   The LR model

The logistic regression (Hosmer & Lemeshow (2000), Balakrishnan (1991)) belongs to the class of generalized linear models (McCullagh & Nelder (1989)). Using this technique yields a mean to predict the probability of occurrence of an event by fitting data to a logistic curve. As this is the case in CART method, either numerical or categorical explanatory variables can be introduced. The logistic regression is used for binomial regression and thus is considered as a choice model. The main domains in which it is used are medical and marketing worlds, for instance for the prediction of a customer's propensity to cease a subscription. As it is often used with binary events, sometimes actuaries also model the mortality of an experienced portfolio with this tool. It is a mean for them to segment their portfolio regarding this risk. Here, the goal is to model the surrender decision of the policyholders.

A   Why the logistic function : a first explanation

The logistic function is very useful because from an input z which varies from negative infinity to positive infinity one gets an output $\Phi(z)$ confined to values between 0 and 1. Its expression is :

$$\Phi(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z}$$

Because we want to model a probability (represented by $\Phi(z)$ above), this is the first explanation of this choice. The requirement of a non-decreasing function for cumulative distribution function is satisfied. Actually z represents the exposure to some set of risk factors, and is given by a common regression equation

$$z = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k,$$

where the X_i are the explanatory variables (e.g. age). Hereafter, we denote by β the vector of regression coefficients $(\beta_0, \beta_1, ..., \beta_k)^{'}$.

B   Another approach

We could also introduce this technique considering the strict regression approach. The idea is to transform the output of a common linear regression to be suitable for probabilities by using a *logit* link function as :

$$logit(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k, \tag{13}$$

Remark 1. *:*

- *$\forall i = 1, ..., k; \beta_i$ represents the regression coefficient associated to the risk factor i (say the sex for instance),*

- *the inverse of the logit function is the logistic function : $\Phi^{-1}(p) = \beta_0 + \sum_{j=1}^{k} \beta_j X_j$,*

- *there are also the so-called polytomic or multinomial regression when the variable to explain (response variable) has more than two levels,*

- *$\frac{p}{1-p} \in [0, +\infty[\Rightarrow \ln\left(\frac{p}{1-p}\right) \in \ ]-\infty, +\infty[,$*

- *other link-functions exist.*

C   Estimation of parameters

Because we are not exactly working in the same framework as in the linear regression case,

---

1. available at http://cran.r-project.org/web/packages/randomForest/index.html
2. available at http://genome.jouy.inra.fr/doc/genome/statistiques/R-2.6.0/library/ipred/html/bagging.html

we use a different technique to estimate the parameters. The ordinary least square estimation is the most famous one to get the regression coefficients, but the fact that we want to estimate a probability (number of surrenders $\sim B(n, \Phi(\beta_0 + \beta_1 X_1 + ... + \beta_k X_k)))$ implies that we usually estimate the coefficients thanks to the maximum likelihood principle. Besides, an ordinary least square estimation would not be well-adapted.

C.1   The maximum likelihood and the regression coefficients

Let n be the number of observations (policyholders). By definition, the maximum likelihood principle gives with a binomial law

$$L(X, \beta) = \prod_{i=1}^{n} \Phi(X_i \beta')^{Y_i} (1 - \Phi(X_i \beta'))^{1-Y_i}$$

The log-likelihood is then

$$
\begin{aligned}
\ln(L(X, \beta)) &= \sum_{i=1}^{n} Y_i \ln(\Phi(X_i \beta')) \\
&+ \sum_{i=1}^{n} (1 - Y_i) \ln(1 - \Phi(X_i \beta')) \\
&= \sum_{i=1}^{n} Y_i \ln\left(\frac{e^{X_i \beta'}}{1 + e^{X_i \beta'}}\right) \\
&+ \sum_{i=1}^{n} (1 - Y_i) \ln\left(1 - \frac{e^{X_i \beta'}}{1 + e^{X_i \beta'}}\right) \\
&= \sum_{i=1}^{n} Y_i (X_i \beta') - \ln(1 + e^{X_i \beta'}) \quad (14)
\end{aligned}
$$

To find the right β which maximizes the likelihood (or the log-likelihood!), let us find the β, denoted by the estimator $\hat{\beta}$, such that

$$\frac{\partial \ln(L)}{\partial \hat{\beta}} = \frac{\partial l}{\partial \hat{\beta}} = 0 \quad (15)$$

This condition yields to a system of equations that are not in a closed form. The use of the Newton-Raphson algorithm to find its solution is advised (see Appendix C and Appendix D for further details).

C.2   The final probability

The individual estimation of the final probability is inferred from the previous estimations,

$$\hat{p} = \Phi(\hat{\beta}_0 + \hat{\beta}_1 X_1 + ... + \hat{\beta}_k X_k) \quad (16)$$

where the $\hat{\beta}_i$ are the regression coefficients estimated by maximum likelihood.

Each insured has an estimated probability to surrender given its characteristics. Policyholders with same characteristics are therefore homogeneous and have the same probability to surrender.

Next step is to determine the confidence interval for the surrender probability on the whole portfolio. In a collective framework, the usual way is to use the Binomial law approximation which considers that the number of surrenders among n insureds follows a Normal distribution. However this technique requires that :

- probability p_i to surrender her contract is comparable for all i in the group (homogeneity) ;
- $n \rightarrow \infty$, which means that the portfolio size is big (n insureds).

The first point is a direct consequence of the Central Limit Theorem (TCL) : the number of surrenders follows a Binomial law, which is a sum of Bernouilli laws. Hence, if these Bernouilli laws are independent and identically distributed we can apply the TCL formula which tells us that the number of surrenders is normally distributed. But a portfolio is heterogeneous. Imagine that the n policyholders in the portfolio are divided into homogeneous groups of policyholders, then each group is normally distributed and the sum of these groups consists in the portfolio. But the sum of normally distributed laws is a normally distributed law, that is why we still can use the Normal approximation. The second point is not a problem in insurance (portfolios are huge by nature).

The exact estimation of the expectation and the variance of the Binomial law yields to a correct final approximation using the confidence interval of a Normal Standard law. Consider that the individual surrender decision of a policyholder follows a Bernouilli law then the number of surrenders N_i^s in a given homogeneous group i embedding n_i policyholders is binomially and

identically distributed, $N_i^s \sim B(n_i, p_i)$. Hence,

$$\mathbb{E}[N_i^s] = \sum_{i=1}^{n_i} p_i = n_i p_i,$$

$$\mathbb{V}ar[N_i^s] = \sum_{i=1}^{n_i} p_i(1 - p_i) = \sum_{i=1}^{n_i} p_i q_i = n_i p_i q_i,$$

$$\sigma[N_i^s] = \sqrt{\mathbb{V}ar[N_i^s]} = \sqrt{\sum_{i=1}^{n_i} p_i q_i} = \sqrt{n_i p_i q_i}$$

Denote by $\hat{p}_i = \dfrac{N_i^s}{n_i}$ the surrender rate, we get

$$\mathbb{E}[\hat{p}_i] = \frac{\mathbb{E}[N_i^s]}{n_i} = \frac{\sum_{i=1}^{n_i} p_i}{n_i} = p_i, \qquad (17)$$

$$\sigma[\hat{p}] = \sigma\left[\frac{N_i^s}{n_i}\right] = \frac{1}{n_i}\sigma[N_i^s] = \sqrt{\frac{p_i q_i}{n_i}} \qquad (18)$$

From (17) and (18) we can get the classical confidence interval of a Normal distribution within a homogeneous group i. But the sum of independent Normal distributions is still a Normal distribution (the total number of surrenders is the sum of surrenders of homogeneous subgroups, $N^s = \sum_i N_i^s$), thus we can generalize to the whole portfolio and get the confidence interval of $\hat{p} = \dfrac{N^s}{n}$ (95% confidence level)

$$CI(p) = \left[\hat{p} - 1.96\sqrt{\frac{\hat{p}(1 - \hat{p})}{n}}, \hat{p} + 1.96\sqrt{\frac{\hat{p}(1 - \hat{p})}{n}}\right]$$
$$(19)$$

C.3   Deviance and tests

Famous tests of likelihood ratio and Wald test are available in more details in Appendix E.

D   Interpretations

The regression coefficients give us some information on effect of each risk factor.
The intercept β_0 is the value of z for the reference risk profile, this is the expected value of the outcome when the predictor variables correspond to the reference modalities (for categorical variables) and thresholds (for continuous vari-

ables).
Then the coefficients β_i (i = 1,2,...,k) describe the contribution of each risk : a positive β_i means that this risk factor increases the probability of the outcome (lapse), while a negative one means that risk factor decreases the probability of that outcome. A large $\dfrac{\beta_i}{\sigma(\beta_i)}$ (where $\sigma(\beta_i)$ denotes the standard deviation of the coeficient estimation) means that the risk i strongly influences the probability of that outcome, and conversely. In the case of a categorical variable, the regression coefficient has to be interpreted as compared to the reference category, for which $\beta = 0$.

Focus on the odd-ratio indicators They represent the ratio of probabilities $\frac{p}{1-p}$.

Example 1. *Let us say that the probability of success $p = P(Y = 1|X)$ is 0.7. Then the probability of failure $q = P(Y = 0|X)$ is 0.3. The odds of success are defined as the ratio of these two probabilities, i.e. $\frac{p}{q} = \frac{0.7}{0.3} = 2.33$; it means that with the same characteristics (vector X), the success is 2.33 more likely to happen than the failure (obviously the odds of failure are $\frac{0.3}{0.7} = 0.43$).*
Now consider that only one explanatory variable differ from one policyholder to another, say the age (among age and region). From (13) we get for one policyholder $\frac{p}{q} = e^{\beta_0 + \beta_1 X_{age} + \beta_2 X_{region}}$. All terms disappear between the two policyholders except age because they are equal, thus the odd-ratio between them aged 40 and 30 years old is defined by

$$\frac{\dfrac{P(Y = 1|X_{age} = 40)}{P(Y = 0|X_{age} = 40)}}{\dfrac{P(Y = 1|X_{age} = 30)}{P(Y = 0|X_{age} = 30)}} = \frac{e^{40\beta_1}}{e^{30\beta_1}} = e^{10\beta_1}$$

More generally, looking at the equation giving the odd-ratio, we notice that a unit additive change in the values of explanatory variables should change the odds by constant multiplica-

tive figures.

$$
\begin{aligned}
e^{logit(p)} &= \frac{p}{1-p} = \frac{p}{q} = e^{\beta_0 + \beta_1 X_1 + ... + \beta_k X_k} \\
&= e^{\beta_0} e^{\beta_1 X_1} ... e^{\beta_k X_k} \quad (20)
\end{aligned}
$$

From (20), for instance with a binary explanatory variable x_i (categorical variable) :

$$
\begin{cases}
\text{if } x_i = 0 \Rightarrow e^0 = 1, \text{and the term disappears,} \\
\text{else if } x_i = 1 \Rightarrow e^{\beta_i x_i} = e^{\beta_i}
\end{cases}
$$

If our explanatory variables x_i are all binary, we are left in (20) with terms for only the x_i that are true, and the intuition here is that if I know that a variable is true, then that will produce a constant change in the odds of the outcome : $x_1 = x_2 = 1$ and $x_k = 0 \ \forall k \neq 1,2 \Rightarrow$ odd-ratio $= e^{\beta_1} e^{\beta_2}$.
This is the same idea with a continuous variable (see Example 1).

Thus, the odd-ratios represent the difference in terms of probability regarding the modeled event (here the surrender) when explanatory variables change and thus is a very useful operational tool.

E Limits of the model

Some problems are raised when using this modeling : concerning assumptions, the policies $(Y_i | X_i)$ are considered independent knowing the explanatory variables. Explanatory variables must be independent, which is never totally right in reality. Fortunately calculations can be done in practice if the Pearson correlation coefficient is not equal to 100% (in this case singularity in matrix inversion). Modalities of a categorical variable are considered independent, which is generally true except in case of erroneous data. Another limit is that a lot of data should be available for the robustness of the modeling. Well, this is not our problem because insurance portfolios are by definition huge.

Other topics have to be questioned in our context, especially : applying the logistic regression over a whole portfolio of life-insurance con-

tracts could lead us to strange results. Indeed, if the portfolio covers a period of 100 years, almost all the policyholders would have lapsed and the regression would make no sense ! That is why we divide the study into several observation periods in the second part of applications (see B). But this leads to duplicate contracts on each observation period, which is maybe an unpleasant source of problems : we can cite the possible existing correlation between periods for a given contract (it is as if we would consider this contract as another one when we change the period). Finally, memory size and time of computations could also be a problem.

The logistic regression is a great tool to model the differences of the outcome variable considering the differences on the explanatory variables. The big drawback is the assumption of independence between explanatory variables, but a crucial advantage is the opportunity to make predictions. Some example of applications can be found in Huang & Wang (2001) and Kagraoka (2005). There exists some other segmentation models in the same family as Tobit model (see Cox & Lin (2006)) and Cox model. A comparison of these different models is also available in Austin (2007). For further details, please refer to the bibliography.

IV Application on a Life Insurance portfolio

Depending on the country, the database provides typically data on policyholder's characteristics (birth date, gender, marital status, smoker status, living place...) and policy features (issue date, end date, type of contract, premium frequency, sum insured, distribution channel...) of life insurance contracts. Here, a Spanish real life portfolio was collected thanks to AXA Seguros. In our study we have information on the gender, the birth date of the policyholders ; the type of contract, its issue date, its termination date and the reason of the termination, the premium frequency, the face amount which is an indicator of the wealth of the policyholder and the premium which encompasses the risk premium and

the saving premium.

The risk premium is commonly the product of the sum-at-risk (sum paid back to the policyholder in case of guarantee) by the probability for the guarantee to be triggered. Typically with certain endowment products covering the death, the risk premium is the product of the sum-at-risk by the mortality rate. The saving premium is the investment made by the policyholder.

All simulations have been performed with R, an open-source statistical software that you can find on the web[1]. We used the package rpart to implement the CART method and obtain the following results. The useful functions to implement the logistic regression are included in the core of the R program.

A Static analysis

We mean by static analysis a "photograph" at a given date of the state of the portfolio. There are 28506 policyholders in this portfolio, the types of long-term contract are either pure saving or endowment products but we focus on endowment policies hereafter. The study covers the period 1999-2007 and the "photo" is taken in December 2007. This means that the characteristics of policyholders and contracts that we extract from the database for the study are those observed either at the date of their surrender or in December 2007 if the policyholder has not surrendered yet.

First, we would like to have an idea of the possible triggers in the surrender decision. Actually we just want to explain the *surrender* in function of *other variables*. The static model enables us to detect the "risky" policyholders regarding the surrender at a given date.

Remark 2. *The static analysis raises some burning questions like : what is the composition of the portfolio ? Is the portfolio at maturity ? What is the part of new business ?*
For example if the duration is one of the main explanatory risk regarding the surrender (and this is !), one has to be careful to cover a suf-

ficiently long period to experiment a normal surrender rate, say 15% a year. If the contract duration is almost always at least 15 months (before the surrender), looking at surrenders statistics twelve months after the issue date of the contracts would not be realistic because the annual lapse rate would be very close to 0%.
Actually we do not have a dynamical view of the phenomenom (surrender), the static analysis is just a simple way to point out the more discriminant factors of the surrender decision. Even if nine years of experience in our study seems to be ok, we will run in B the study monthly to reflect the fact that policyholders often wonder if they should surrender their contract (say at least twice a year).

In December 2007, 15571 of the 28506 endowment contracts present in the database have been surrendered. The two segmentation models provide us with two different information :

- first, the CART model gives us the more discriminant variables regarding the surrender in descending order (when reading the classification tree from the root to the leaves). Finally, one can class a policyholder as "risky" at the underwriting process or later but the predicted response is binary (we can get the probability to be in a given class but not the probability to surrender);

- the LR model offers a more precise result, the probability for this policyholder to lapse his contract in the future given its characteristics. Hence the response is not binary, this is a kind of intensity or propensity to surrender the contract. One can also compare the effect of changing the modality (for categorical variable) or the value (continuous variable) of an explanatory variable thanks to the odd-ratios technique.

In the following, the duration is an input of the model (explanatory variable) to highlight its importance in the surrender decision. If the question concerns the segmentation at the underwrit-

---

1. www.r-project.org/

Table 1: The confusion matrix for T_{max} on the validation sample.

	observed Y = 0	observed Y = 1
predicted Y = 0	4262	1004
predicted Y = 1	728	5644

Table 2: The confusion matrix for the pruned tree on the validation sample.

	observed Y = 0	observed Y = 1
predicted Y = 0	4188	1078
predicted Y = 1	664	5708

ing process, this variable should not be input in the model because it is unknown.

A.1   The CART method

In R, this application is done thanks to the package rpart [1] (r-partitionning) and more precisely the procedure rpart which builds the classification tree.   By default, rpart uses the Gini index to compute the impurity of a node. As we have seen previously, this option is not important because results do not much differ. There is no misclassification cost (see Appendix II.4) in our application. We proceed like in theory :

1. first, T_{max} is built (by setting in rpart the option *cp* equal to 0) ;

2. second, this tree is pruned off to lower the number of leaves and simplify the results.

The minimum number of observations required in a leaf of T_{max} has been set to 1, the number of competitive splits computed is 2, and we use the cross-validation technique to get better and more accurate results. The number of samples for cross-validation is set to 10 in rpart.control.   Beware :   these cross-validations correspond to the misclassification rate estimated by cross-validations (and not the cross-validation estimate of the prediction error presented in $A.3$, which is just useful to estimate better the real prediction error but not to build an optimal tree).
We randomly create the learning and validation data sets, whose sizes are respectively 16868 and 11638 policyholders.

The test-sample estimate of the prediction error in the maximal tree T_{max} computed on the validation sample is 14.88%. The corresponding confusion matrix is given in Table 1. This

tree has too many leaves, its representation is too complex so we have to prune it.

The choice of the complexity parameter α in the pruning algorithm (see Appendix II.5) is a trade-off between the final size of the tree and the minimum misclassification rate required by the user. Figure 7 in Appendix I plots the learning error in function of this complexity cost. Each complexity parameter corresponds to an optimal tree whose size is specified on the graph gotten by ten cross-validations.

Table 7 in Appendix I shows that minimizing the learning error (by cross-validation) and its standard deviation requires setting $\alpha \in$ $]1.04e^{-04}, 1.30e^{-04}]$, but the corresponding number of leaves (equal to 82) is too high to represent the tree easily. Hence we have chosen to set $\alpha = 6e^{-04}$ which corresponds to 11 leaves and a very small increase of the error. Figure 4 shows this tree. The most important (discriminant) variable seems to be the type of contract (characterized by the premium type,unique or periodic; and the profit benefit option), then the duration and so on.

The variables actually input for the tree construction are the contract type, the duration, the face amount, the premium.frequency, the saving premium and the underwriting age. Finally, the gender and the risk premium are the only variables which don't appear in the final tree.
The first splitting-rule is therefore "does the policyholder own a contract with profit benefit ?". If "no" go down to the left, otherwise go down to the right. The predicted classes are written in the terminal nodes, and the proportions under this class are the number of policyholders observed as "no surrender" on the left and "surrender" on the right. Obviously the bigger the

---

1. http://cran.r-project.org/web/packages/rpart/index.html, developed by T. M. Therneau and B. Atkinson
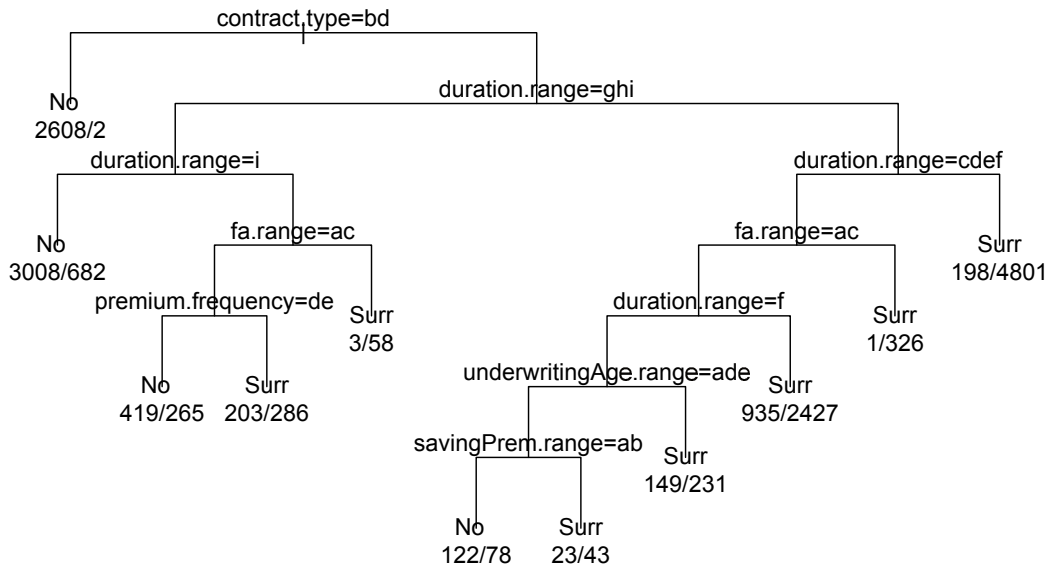
Figure 4: The final classification tree. Binary response variable : surrender. The first splitting-rule *contract.type = bd* means that the contract type is the more discriminant variable (*bd* correspond to the 2nd and 4th categories, like in alphabetic order). continuous explanatory variables have been previously categorized for the modeling.

difference between these numbers is, the better the tree segment the data. Here, if the policyholder has a contract with a periodic or unique premium and no profit benefit option (PP sin PB and PU sin PB), he probably won't surrender (2608/2610 = 99.92%). The predicted class is labeled "No".

Remark 3. *Sometimes some categories of certain explanatory variables do not appear in the final tree. In fact, the representation of the tree obliges us to hide other competitive possible splits at each node (or surrogate splits). But the complete analytic result provides the solution to this problem (it is just a display problem).*

Example 2. *Let us consider a man whose characteristics are the following : he pays a periodic premium and owns a contract with profit benefit,*

the duration of his contract is today observed in the seventh range and his face amount belongs to the second range. The tree predicts that this policyholder is today in a risky position knowing its characteristics (58/61 ≃ 95% of people with these characteristics have surrendered their contract).

Looking at Figure 4, it is clear that the main discriminant factor regarding the surrender risk here is the profit benefit option. The misclassification rate (learning error) of this tree is 33.1% according to Table 7 presented in Appendix *I*. The prediction error can be estimated via the confusion matrix in Table 2.

This prediction is quite good because only 14.97% of predictions are wrong, which is almost equal to the prediction error on the maximal tree T_{max}. Indeed the compromise is really interest-

Table 3: The confusion matrix of the classifier by the Random Forest.

	observed Y = 0	observed Y = 1
predicted Y = 0	10327	2608
predicted Y = 1	1592	13979

Figure 5: On the left, the importance of explanatory variables. On the right, the number of trees required to stabilize the *out-of-bag* errors : the black line is the overall error, the green line is the error of the category "surrender" and the red one for the category "no surrender".
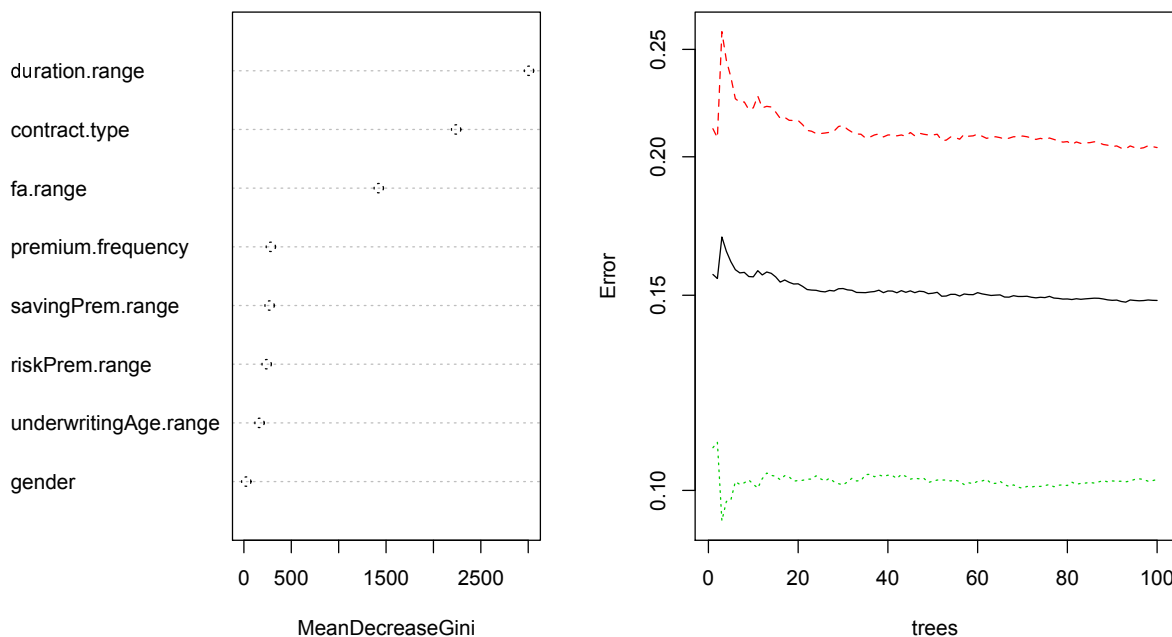
ing because pruning the tree from 175 leaves to 11 leaves increases the prediction error less than 1% !

To consolidate these results, we use the bagging predictors thanks to the `randomForest` package. The following stages in the Random Forest algorithm are performed to grow a tree : bootstrap the original sample (this sample will be the training set), split at each node with the best variable in terms of decrease of the impurity (possible m variables randomly chosen among M initial input variables, $m < M$ because m=M corresponds to the *bagging* method), grow the tree to the largest extent possible (no pruning). The forest error rate depends on the strength of each individual tree (its power to classify well) and the correlation between any two trees in the forest. When the strength increases the forest error decreases and when the correlation increases the forest error also increases. m is the only adjustable parameter to which random forests is sensitive, and reducing m reduces both the cor-

relation and the strength ; thus there is an optimal m that we can find with the *out-of-bag* error. We cannot represent the new final classifier as a tree, but it gives best results. Table 3 summarizes the results on the entire original data set (no learning and test samples because this is already a bootstrap aggregation).

The unbiased *out-of-bag* error estimate is 14.73%. The importance of explanatory variables is given in Figure 5, as well as the necessary number of trees in the forest for the *out-of-bag* error to be stabilized (here it seems to be about 50 trees). These results confirms what we expected : the duration and the type of contract are the most meaningful variables to explain the decision to surrender her life insurance contract. All the concepts developed in this section are explained on Breiman's webpage [1].

A.2   The LR model

Consider that X is the matrix of explanatory variables for each observation, that is to say a

---

1. See http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm

Table 4: Odd-ratios, endowment products (duration in month, learning sample). Contract types : PP con PB → periodic premium (PP) with profit benefit (PB), PP sin PB → PP without PB, PU con PB → unique premium (PU) with PB, PU sin PB → PU without PB. continuous explanatory variables have previously been categorized for the modeling.

Odd-ratio	Ref.	Other modalities							
Duration	[0,12]	]12,18]	]18,24]	]24,30]	]30,36]	]36,42]	]42,48]	]48,54]	> 54
nb surrenders	3062	1740	1187	791	728	400	365	244	682
empirical OR		10.56	2.89	2.69	1.82	1.16	0.96	0.68	0.19
modeled OR		0.27	0.07	0.06	0.05	0.03	0.02	0.02	0.004
Premium freq.	Monthly	Bi-monthly	Quarterly	Half-Yearly	Annual	Single			
nb surrenders	2790	12	323	92	595	5387			
empirical OR		2.22	0.93	0.66	2.39	1.60			
modeled OR		2.52	0.97	0.80	1.55	0.75			
UW. age	[0,20[	[20,30[	[30,40[	[40,50[	[50,60[	[60,70[	> 70		
nb surrenders	258	1719	2165	2002	1490	1088	477		
empirical OR		1.16	1.06	1.25	1.63	2.67	3.28		
modeled OR		1.32	0.99	0.77	0.67	0.51	0.47		
Face amount	#1*	#2*	#3*						
nb surrenders	5361	684	3154						
empirical OR		0.14	0.12						
modeled OR		0.003	0.0008						
Risk prem.	#1*	#2*	#3*						
nb surrenders	3941	2987	2271						
empirical OR		1.50	0.92						
modeled OR		1.43	1.30						
Saving prem.	#1*	#2*	#3*						
nb surrenders	3331	1762	4106						
empirical OR		1.90	2.09						
modeled OR		2.55	3.78						
Contract type	PP con PB	PP sin PB	PU con PB	PU sin PB					
nb surrenders	3840	0	5357	2					
empirical OR		0	4.75	0.0008					
modeled OR		5.6e-08	0.0006	3.9e-06					

* Note : for confidentiality reasons, the real ranges of the face amount, the risk premium and saving premium are omitted.

line of the matrix X represents a policyholder and a column represents the observed value for a certain risk factor (e.g. the age).

The response vector $Y = (Y_1, Y_2, ..., Y_n)'$ represents the surrender decisions of the 28506 insureds (policyholders).

In the classical regression framework, the problem can be written in the matrix form :

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_{1,1} & X_{1,2} & \cdots & X_{1,k} \\ 1 & X_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & X_{n,1} & \cdots & \cdots & X_{n,k} \end{pmatrix} \times \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}$$

We ran the logistic regression in R thanks to

the function `glm`, the output of the model is the effect of each variable and its confidence bands (see D and Appendix D), and the deviance of the model (see Appendix E).

Categorical variables are split into dummy variables corresponding each one to a modality. A stepwise logistic regression is carried out with a step-by-step iterative algorithm which is used to compare a model based on p' of the p original variables to any of its sub-model (with one less variable) or to any of its top-model (with one more variable). Hence the R procedure `stepAIC` from the R package `MASS` allows us to drop non significant variables from the model and to add relevant ones. We finally get the opti-

mal model with the minimum number of relevant variables.

The learning sample still contains the randomly chosen 16868 policyholders and the validation sample 11638. As usual, the regression coefficients were computed on the learning sample whereas the predictions were made on the validation data set.

Table 8 in Appendix A summarizes the regression coefficients of the explanatory variables, the standard deviation associated, and the confidence we can have in the test of the relevance of the explanatory variable (see Appendix E.2). The odd-ratios, presented in D, is an important operational tool and should be compared to 1. Looking at Table 4, we clearly see that the modeled odd-ratios are a bad representation of the reality : they are very different from the empirical odd-ratios (obtained via descriptive statistics). For instance, the model tells us that a policyholder whose underwriting age is over 70 years old is less likely to surrender than a young policyholder whose underwriting age is less than 20 years old (the reference range) all other things being equal. The experience shows that in fact they are 3.28 times more likely to lapse !

The model has therefore a bad goodness of fit since many regression coefficients estimates are not significant, and this is the reason why the modeled odd-ratios do not represent the reality in most of cases. But there is a trade-off between the goodness of fit and the predictive accuracy. In our case, we prefer to have good results in terms of prediction than for goodness of fit. To check for this, we still look at the confusion matrix given in Table 5 which gives the number of missclassified policyholders and represents the

prediction power of the method. Of course good predictions still appear in the diagonal of this table and we can get the predicted misclassification rate with it. To make such predictions, we consider that a policyholder with a modeled probability to surrender greater than 0.5 is assigned the response 1, otherwise the response 0. Here the predictions are right for 84.96% of the validation sample. Thus the prediction error equals to 15.04% and is quasi-similar to the one gotten with the CART method.

Other usual performance criteria to compare the two methods are the sensitivity (Se) and the specificity (Sp). Let *success* be the case which corresponds to a predicted and an observed response equal to 1 in the confusion matrix, *misses* correspond to a predicted response equal to 0 and the observed one 1, *correct rejections* correspond to an observed and a predicted response equal to 0, and finally when the predicted response is 1 and the observed one is 0 this is a *false risky policyholder*. The sensitivity is the number of *success* over the number of observed surrendered contracts, and the specificity is the number of *correct rejections* over the number of observed non-surrendered contracts.

Table 6 summarizes the performance criteria for each method ; we want to minimize the proportion of *misses*. The predictions of the LR model have less *misses* and more *false risky policyholders* ; in all CART models results are quite similar, errors are well-balanced and the compromise between the sensitivity and the specificity is better but the number of *misses* is higher. Hence, the most prudential model is clearly the LR model (10%).

Table 5: The confusion matrix (LR model).

	observed Y = 0	observed Y = 1
predict Y = 0	#correct rejections 4153	#misses 637
predict Y = 1	#false risky policyholder 1113	#success 5735

Table 6: The performance criteria.

	T_{max}	T_{pruned}	$T_{RandomForest}$	LR
Se	84.9%	84.1%	84.3%	90%
Sp	85.4%	86.3%	86.7%	78.9%
(1-Se)	15.1%	15.9%	15.7%	10%

Figure 6: Predictions and confidence bands of the portfolio surrender rate. On the left, the predictions on the learning sample and on the right predictions on the validation sample.
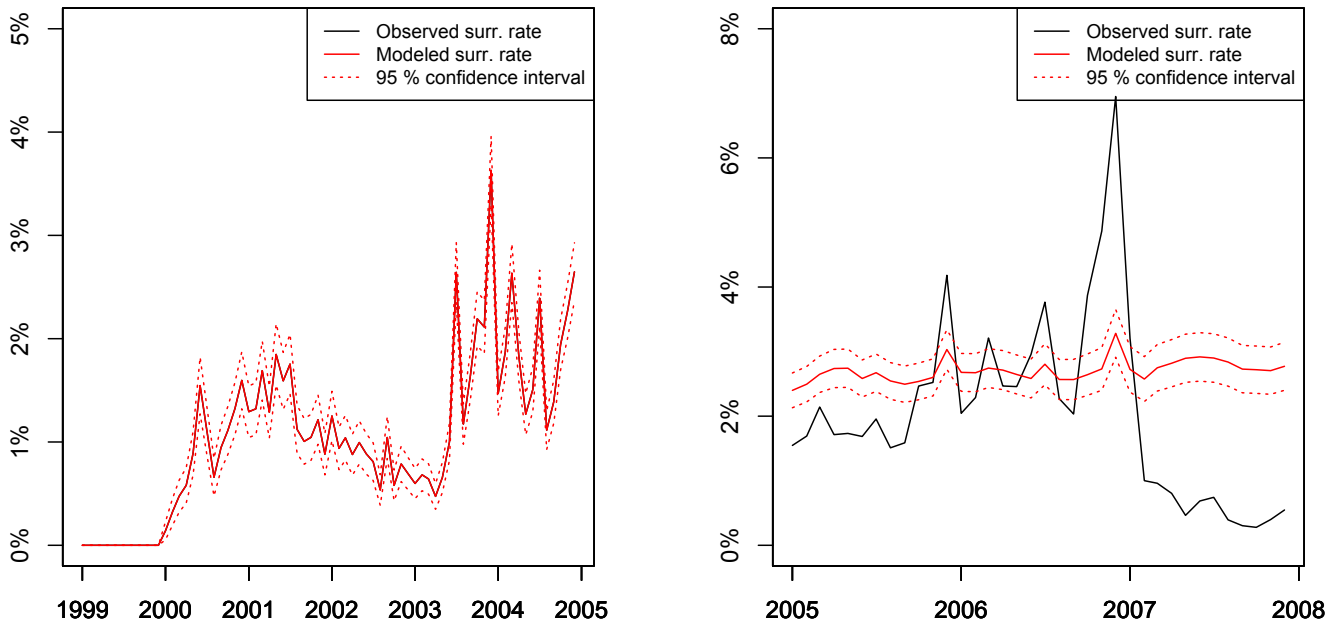
B   A dynamical analysis

In this part of the paper, the LR model is the only one used.

We have already discussed about the problem of a static analysis : depending on the period covered and the phenomenom modeled, it could be erroneous.
If the period covered is longer than the term of the phenomenom, the binary response variable would be equal to 1 everytime. By consequence, the model would not work well; this is the first explanation of running a monthly study. The second one is that we want to model a dynamical decision : we may think that the policyholder is likely to wonder each month if he has to keep in force her contract. However, a robustness and stability problem is raised : we perform the logit modeling monthly (just considering the present contracts in the portfolio at the given date), but is the resulting model reliable and stable between month ? To validate the model built on one period, you might ensure that the model is built on a representative period (the portfolio is at

maturity for example). The dynamical analysis allows us to model the monthly decisions of policyholders and thus to model the surrenders on the whole portfolio each month by aggregation of individual decisions.

The main assumption is the independence between agents (policyholders) and the added underlying assumption here is the independence in time. In practice, we consider that the decision of the policyholder at date $t + 1$ is independent of what happened before, and more precisely independent with the decision at date t. This is a very strong hypothesis which is not reasonable in reality. In the new data set (whose size is 991010), policyholders are duplicated each observed month while they are present in the portfolio (no surrender and no other reason to leave), and their characteristics are up-dated (for instance duration).
We have chosen to check for the accuracy and the quality of the predictions looking at the predicted surrender rate as compared to the observed one each month. The final data set is

divided into the following learning and validation samples : the learning sample (whose size is 629357 lines) covers the period January 1999 to December 2004, and thus the validation sample covers the period January 2005 to December 2007 (its size is 361653).

The same explanatory variables as in the static study have been input to build the model, plus the date. The month of observation is added in the modeling to enable us to predict future surrenders when assuming an expected level (either increase or decrease) of lapsation as compared to a reference date. This is a key-advantage.

The results seem to be acceptable but we should keep in mind that it should work very bad in extreme situations. Here the economic context is not considered in the model, and during economic crisis these indicators should be the main explanatory variables of the surrender decisions. The assumption of independence between policyholders and in time are not at all realistic when considering a crisis period. As a matter of fact, we see on Figure 6 that the period has a big influence : the model perfectly fits the data in the learning period but is a bit far from the reality when predicting the future. The beginning of the financial and economic crisis led the surrender rate to drop in 2007, which is not predicted by the model and shows that the economic situation is also very important.

This is certainly due to the fact that the user has to make an assumption when predicting : what will be the level of lapsation in the coming months as compared to today (or a reference date) ? Then the predicted surrender rate will be adjusted depending on this hypothesis. Here, we simply assume that the level of lapsation in December 2004 will stay the same in 2005, 2006 and 2007 and then we predict the surrender decisions of policyholders. Actually a good prediction depends on the good choice of the future expected general level of lapsation as compared to today (when the date is introduced in the model) : will it be higher ? lower ? the same ?

V   Discussion and improvements

The goal of this paper is to give insights about the discriminant contract features and policyholder's characteristics regarding the surrender behaviour, so what's new ?
Our study has brought out some typical risky profiles : oldest people tend to surrender more than others, as well as people who have a periodic premium ("annual" and "bi-monthly" are the worst cases). Unlike policyholders with an intermediate wealth, those who are very poor or very rich are not really interested in surrendering their contracts : poor insureds have to pay for fees but they do not have the money for it, and rich people may not really pay attention to the situation. But in general the biggest risks are concentrated on the first periods following the termination of a fiscality constraint : if the duration of the contract has reached the period from which the policyholder can surrender her contract without penalty, the risk is very high. Finally, the participation of the policyholder to the benefits of the insurance company plays an important role in its decision, the study has shown that people with no profit benefit option do not surrender their contract whereas people with the profit benefit (PB) option tend to surrender their contract. Three reasons could explain it : first people move to a new product which globally offers a higher PB, second a high PB in the first years of the contract enables the policyholder to overperform the initial yield and could lead her to surrender the contract and recover the surrender value, third someone with a PB option simply receives frequent information on it and on the surrender value which can prompt her to surrender. The gender of the policyholder does not seem to be discriminant.

The conclusion could be that the predictions can be performed by running either the LR model or the CART model, but risky profiles should be extracted from the descriptive statistics or the CART model more than from the LR model for which the modeled odd-ratios are not really significant here. An idea could be

to select salient explanatory variables with the CART procedure and Random Forest algorithm and then apply the LR model to make predictions and use odd-ratios. Another improvement in the LR model could be to *re-balance* the data set which is extremely unbalanced in the dynamical analysis : we observe 15571 surrenders among 991010 observations, thus surrenders represent only 1.57% of the whole data set. We can overcome it by using downsampling or oversampling (Liu et al. (2006)), or by changing the decision function (here the policyholder was assigned a surrender if the modeled probability was over 0.5 in the predictions, but this is not always optimal (Lemmens & Croux (2006))).

A lot of professionnals know that the duration is a meaningful factor in explaining the surrender because of fiscality constraints, but at the underwriting we do not have any information on this factor because the contract is newly acquired.
Hence, duration as an input of the model enables us to predict well the surrender rates but should not be applied when we want to segment the population of policyholders at the underwriting. However this is not a problem : we just have to remove the duration in the models to segment policyholders at underwriting process.
Besides, the results of these two models are true for a fixed date t, when the model is computed. But we would like a dynamical response in function of time, which could be preferable if we want to know for example not the intensity to surrender at t but the intensity to surrender at $t + dt$, where dt can be big. The next step could be to run a functional data analysis which could nicely take into account the economic situation, or to try some models used in survival analysis like the Cox model family. The hazard moral, the adverse selection and hidden variables such as the competition on the market (Albrecher et al. (2010)) could be considered as well.

References

Albrecher, H., Dutang, C. & Loisel, S. (2010), A game-theoretic approach of insurance market cycles. Working Paper.

Atkins, D. C. & Gallop, R. J. (2007), 'Rethinking how family researchers model infrequent outcomes: A tutorial on count regression and zero-inflated models', *Journal of Family Psychology* .

Austin, P. C. (2007), 'A comparison of regression trees, logistic regression, generalized additive models, and multivariate adaptive regression splines for predicting ami mortality', *Statistics in Medicine* **26**, 2937–2957.

Balakrishnan, N. (1991), *Handbook of the Logistic Distribution*, Marcel Dekker, Inc.

Bluhm, W. F. (1982), 'Cumulative antiselection theory', *Transactions of Society of actuaries* **34**.

Breiman, L. (1994), Bagging predictors, Technical Report 421, Department of Statistics, University of California.

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* (24), 123–140.

Breiman, L. (1998), 'Arcing classifiers', *The Annals of Statistics* **26**(3), 801–849.

Breiman, L. (2001), 'Random forests', *Machine Learning* (45), 5–32.

Breiman, L., Friedman, J., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Chapman and Hall.

Cox, S. H. & Lin, Y. (2006), Annuity lapse rate modeling: tobit or not tobit?, *in* 'Society of actuaries'.

Engle, R. & Granger, C. (1987), 'Cointegration and error-correction: Representation, estimation and testing', *Econometrica* (55), 251–276.

Ghattas, B. (1999), 'Previsions par arbres de classification', *Mathematiques et Sciences Humaines* **146**, 31–49.

Ghattas, B. (2000), 'Aggregation d'arbres de classification', *Revue de statistique appliquee* **2**(48), 85–98.

Hilbe, J. M. (2009), *Logistic regression models*, Chapman and Hall.

Hosmer, D. W. & Lemeshow, S. (2000), *Applied Logistic Regression, 2nd ed.*, Wiley.

Huang, Y. & Wang, C. Y. (2001), 'Consistent functional methods for logistic regression with errors in covariates', *Journal of the American Statistical Association* **96**.

Kagraoka, Y. (2005), Modeling insurance surrenders by the negative binomial model. Working Paper 2005.

Kim, C. (2005), 'Modeling surrender and lapse rates with economic variables', *North American Actuarial Journal* pp. 56–70.

Lemmens, A. & Croux, C. (2006), 'Bagging and boosting classification trees to predict churn', *Journal of Marketing Research* **134**(1), 141–156.

Liu, Y., Chawla, N., Harper, M., Shriberg, E. & Stolcke, A. (2006), 'A study in machine learning for unbalanced data for sentence boundary detection in speech.', *Computer Speech and Language* **20**(4), 468–494.

McCullagh, P. & Nelder, J. A. (1989), *Generalized linear models, 2nd ed.*, Chapman and Hall.

Outreville, J. F. (1990), 'Whole-life insurance lapse rates and the emergency fund hypothesis', *Insurance: Mathematics and Economics* **9**, 249–255.

Appendices

A CART method

I Choice of the complexity parameter

`rpart()` prunes the tree and runs a K-fold cross validation (K=10 by default) on each pruned tree (we took K=10). The policyholders in the cross-validation process are randomly selected, thus the *cptable* can slightly differ from one simulation to another. On Table 7, *relerror* measures the learning error and describes the fit of the tree, *xerror* measures the misclassification rate in the 10-fold cross validation and is considered as a better estimator of the actual error. *xstd* is the standard deviation of *xerror*. The optimal tree minimizes $err = xerror + xstd$. If two trees have the same error err, we choose the smallest. Table 7 enables to plot the learning error in function of the complexity parameter and the size of the tree in Figure 7.

Remark 4. *Notes on how to read this table :*

- *the third tree with 2 splits corresponds to* $\alpha \in]2.30, 3.10]$ *,*
- `R` *standardizes the error, that is why relative error of the root is equal to 1. The real error of the root can be obtained by printing the tree (here it is 45.465%),*
- *the maximal tree T_{max} (non-pruned) returned automatically and by default by the function* `rpart()` *corresponds to the last line of the cptable.*

II Deeper in CART theory

II.1 What is an impurity function ?

Definition 2. *An impurity function is a real function g defined over discrete probabilities on a finite set :*

$$g : (p_1, p_2, ..., p_J) \rightarrow g(p_1, p_2, ..., p_J),$$

symmetric in p_1, p_2, ..., p_J and :

1. *the maximum of g is at equiprobability :* $argmax\, g(p_1, p_2, ..., p_J) = \left(\frac{1}{J}, \frac{1}{J}, ..., \frac{1}{J}\right),$

Table 7: Complexity parameters

CP	nsplit	rel error	xerror	xstd	CP	nsplit	rel error	xerror	xstd
3.3981e-01	0	1.000	1.000	0.0084	1.9559e-04	59	0.312	0.332	0.0060
3.0539e-01	1	0.660	0.660	0.0077	1.8255e-04	68	0.310	0.332	0.0060
5.9982e-03	2	0.354	0.361	0.0062	1.3040e-04	73	0.309	0.332	0.0060
7.8237e-04	5	0.336	0.337	0.0061	1.0432e-04	82	0.308	0.332	0.0060
5.2158e-04	10	0.331	0.333	0.0060	9.7796e-05	88	0.307	0.333	0.0060
4.5638e-04	15	0.328	0.333	0.0060	8.6930e-05	97	0.306	0.334	0.0060
3.9119e-04	19	0.326	0.333	0.0060	6.5198e-05	100	0.306	0.334	0.0060
3.6945e-04	21	0.325	0.333	0.0060	4.3465e-05	117	0.305	0.337	0.0061
3.2599e-04	32	0.319	0.333	0.0060	3.7256e-05	132	0.304	0.339	0.0061
3.1295e-04	34	0.318	0.333	0.0060	3.2599e-05	139	0.304	0.340	0.0061
2.6079e-04	39	0.317	0.332	0.0060	2.6079e-05	159	0.303	0.340	0.0061
2.1733e-04	53	0.31360	0.334	0.0060	0.0000e+00	174	0.303	0.341	0.0061

2. *the minimum of g is given by the "dirac":*
   $$argmin \; g(p_1, p_2, ..., p_J) \quad \in \quad \{e_1, ..., e_J\},$$
   where e_j is the jth element in the canonical basis of $\mathbb{R}^J$.

II.2 Existing impurity functions

We usually consider the following functions which satisfy the concavity criterium :
- $impur(t) = - \sum_{j=1}^{J} p(j|t) \ln(p(j|t))$;
- $impur(t) = \sum_{j \neq k} p(j|t) \, p(k|t)$ (Gini index)

Remark 5. *In a variance approach,*

- *the Gini diversity index also equals to $1 - \sum_j p_j^2$;*
- *we also use the twoing rule, choose Δ to maximize $\frac{p_L p_R}{4} \left[\sum_j |p(j|t_L) - p(j|t_R)| \right]^2$;*
- *in a two-class problem, the Gini index reduces to $impur(t) = 2p(1|t)p(2|t)$.*
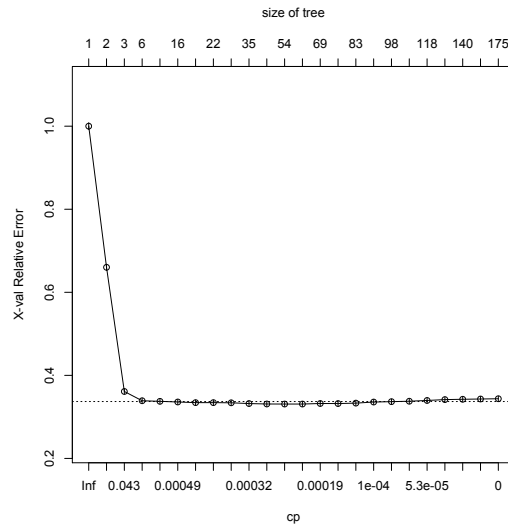

Figure 7: The cross-validated misclassification estimator of the optimal tree in function of the complexity parameter cp (or α). T_{max} contains here 175 leaves and corresponds to $cp = 0$. Notice that there is an initial sharp drop of error followed by a "flat" plateau and a slow rise.

II.3 Notes on prediction error

Notice that :

$$\mathbb{E}[\hat{\tau}(class)] = \mathbb{E}\left[\frac{1}{N}\sum_{(x_n,j_n)\in\epsilon}\mathbb{1}\{class(x_n,\epsilon)\neq j_n\}\right]$$

$$= \frac{1}{N}\sum_{(x_n,j_n)\in\epsilon}\mathbb{E}[\mathbb{1}\{class(x_n,\epsilon)\neq j_n\}]$$

$$= P(class(X,\epsilon)\neq Y) = \tau(class).$$

and all presented estimators are unbiased :
$\mathbb{E}[\hat{\tau}(class)] = \mathbb{E}[\hat{\tau}^{cv}(class)] = \mathbb{E}[\hat{\tau}^{ts}(class)]$
Prediction error and misclassification error are two different concepts. Misclassification error is the error in nodes of the tree whereas prediction error is linked to the final classification of the variable of interest and is calculated once the tree is built.

By default, R computes a cross-validation estimator of the learning error. This is the results given in the complexity parameter table. But this cross-validation procedure does not correspond to the cross-validation technique in resampling theory. The former computes the optimal tree for a given size by minimizing the learning error whereas the latter only aims at getting to a more realistic estimator of the prediction error but does not deal with the problem of finding an optimal tree.

II.4 Penalize wrong classification

Using the inaccurate resubstitution estimate (see A.3) as well as selecting too large trees have led tree structured methods to a lot of critics. In real applications, the cost of misclassifying a class j object as a class i object is not the same for all $i \neq j$. A possible improvement could be to penalize the misclassification of an observation (as compared to the response observed) by a positive factor.

Definition 3. *The cost of classifying an observation in a wrong class is defined by*

$$\Gamma : C \times C \to \mathbb{R}_+ \text{ , such that}$$
$$\Gamma(i|j) \geq 0 \text{ and } \Gamma(i|i) = 0$$

Hence, let us define

- the probability to class an observation badly by $P_{class}(i|j) = P(class(x,\epsilon) = i \mid j)$ (the function *class* classes x in the class i instead of the class j),

- $\tau_{class}(j) = \sum_i \Gamma(i|j)P_{class}(i|j)$: the mean cost of wrong classification,

We get $\tau_{class} = \tau(T)$ and

$$\tau(T) = \sum_j \pi(j)\tau_{class}(j) = \frac{1}{N}\sum_j N_j\tau_{class}(j)$$

Given this new framework, Ghattas (2000) defines the new penalized classification function to assign a class to a terminal node t :

$$class(x,\epsilon) = \underset{i\in C}{argmin}\sum_{j\in C}\Gamma(i|j)\ p(j|t) \qquad (2)$$

From (2), the estimation of the misclassification rate is now

$$r(t) = \min_{i\in C}\sum_{j\in C}\Gamma(i|j)\ p(j|t)$$

Knowing that $\tau(t) = r(t)p(t)$, the misclassification rate by substitution on the tree T is still

$$\hat{\tau}(T) = \sum_{t\in\tilde{T}}\hat{\tau}(t) \qquad (3)$$

Corollary 1. *The tree misclassification rate estimator $\hat{\tau}(T)$ becomes smaller each time a split is made, whatever the split. Thus, if we denote by T_s the tree gotten by splitting T at a terminal node, we get*

$$\hat{\tau}(T_s) \leq \hat{\tau}(T) \qquad (4)$$

Let t_L and t_R be the descendants of node t in tree T_s.

From (3) and (4), it turns out that

$$\sum_{t\in\tilde{T}_s}\hat{\tau}(t) \leq \sum_{t\in\tilde{T}}\hat{\tau}(t)$$

$$\sum_{t\in\tilde{T}}\hat{\tau}(t) - \hat{\tau}(t) + \hat{\tau}(t_L) + \hat{\tau}(t_R) \leq \sum_{t\in\tilde{T}}\hat{\tau}(t)$$

$$\hat{\tau}(t_L) + \hat{\tau}(t_R) \leq \hat{\tau}(t) \quad (5)$$

II.5   Pruning the tree

The problem of a too complex final tree over-fitting data can be easily solved. In fact looking for the right stopping-rule is the wrong way of looking at the problem, a more satisfactory procedure to get the final result consist of two key elements.

1. Don't stop the construction of the tree (forget arbitrary stopping-rules) and get the largest tree T_{max} ; then prune it upward until the root node (the criterion to prune and recombine the tree upward is much more important than the splitting criterion) ;

2. Use better estimators of the true misclassification rate to select the right sized tree from among the pruned subtrees. Use cross-validation or learning/test samples for this.

The idea is to look for subtrees of T_{max} with a minimum misclassification rate. To prune a branch T^t from a tree T means to delete all descendants of node t in T.
The resulting pruned tree is denoted by $T' = T - T^t$, and $T' < T$.
From (5) we get

$$\hat{\tau}(t) \geq \hat{\tau}(T^t) \quad (6)$$

T_{max} contains so many nodes that a huge number of distinct ways of pruning up to the root exist, thus we need to define a criterion to select the pruning procedure which gives the "best" subtree (the right-sized tree). Obviously, the natural criterion to compare same sized trees is the misclassification error : the selective pruning process starts with T_{max} and progressively

prunes T_{max} upward to its root node such that at each stage of pruning the misclassification rate of the tree is as small as possible. This work yields to a sequence of smaller and smaller trees : $T_{max} > T_1 > T_2 > ... > T_{root}$. ($T_{root}$ is just the root node)
From (4), notice that : $T_1 < T_{max} \Rightarrow \hat{\tau}(T_{max}) \leq \hat{\tau}(T_1)$. The error of the maximal tree is always less or equal to the error of the pruned tree and the aim is to lower the number of leaves of T_{max}, thus it is natural to think about penalizing a big number of leaves in the final tree. That is why we introduce in the term of the error a complexity cost representing this idea. The new misclassification rate or *cost-complexity measure* is then :

$$\hat{\tau}_\alpha(T) = \hat{\tau}(T) + \underbrace{\alpha\, Card(\tilde{T})}_{\text{complexity term}} \quad, \text{ where } \alpha > 0.$$

$$(7)$$

$Card(\tilde{T})$ is the number of terminal nodes of T. Actually we just want to find the substree $T(\alpha) \leq T_{max}$ which minimizes $\tau_\alpha(T)$ for each α :

$$\tau_\alpha(T(\alpha)) = \min_{T \leq T_{max}} \tau_\alpha(T) \quad (8)$$

For problems of existence and unicity of the tree $T(\alpha)$, please refer to Breiman et al. (1984).
α is clearly linked to the size of the final pruned tree; if α is small, then the penalty for having a lot of leaves is small and the tree $T(\alpha)$ will be large.
The critical cases are :

- $\alpha = 0$: each leaf contains only one observation (T_{max} very large). Every case is correctly classified and $\tau(T_{max}) = 0$. T_{max} minimizes $\tau_0(T)$;
- $\alpha \to +\infty$: the penalty for terminal nodes is big and the minimizing subtree will consist in the root node only.

Algorithm 1. *To know what branches to prune off and the optimal α associated,*

1. *Let terminal nodes t_L and t_R be the immediate descendants of a parent node t; starting from T_{max}, one looks for the division*

which did not lead to a decrease of error, i.e. where $\hat{\tau}(t) = \hat{\tau}(t_L) + \hat{\tau}(t_R)$ (see (5)). Prune off t_L and t_R, and do it again until no more pruning is possible. We get $T_1 < T$;

2. For T_1^t any branch of T_1, define $\hat{\tau}(T_1^t) = \sum_{t\in\tilde{T}_1^t}\hat{\tau}(t)$. According to (6), the non terminal nodes t of the tree T_1 satisfy the following property : $\hat{\tau}(t) > \hat{\tau}(T_1^t)$ (no equality because of step 1).

3. Denote by $\{t\}$ the subbranch of T_1^t consisting of the single node $\{t\}$, $card(\{t\}) = 1$. Hence, $\hat{\tau_\alpha}(\{t\}) = \hat{\tau}(t) + \alpha$ and

$$\hat{\tau_\alpha}(T_1^t) = \hat{\tau}(T_1^t) + \alpha\ Card(\tilde{T}_1^t) \quad (9)$$

We have seen that $\hat{\tau}(T_1^t) < \hat{\tau}(\{t\})$, but the introduction of the complexity term makes this inequality with $\hat{\tau_\alpha}$ become not always true. While $\hat{\tau_\alpha}(T_1^t) < \hat{\tau_\alpha}(\{t\})$ it is no use to prune the tree, but there exists a threshold α_c such that $\hat{\tau_{\alpha_c}}(T_1^t) = \hat{\tau_{\alpha_c}}(\{t\})$. Therefore,

$$\hat{\tau}(T_1^t) + \alpha_c\ Card(\tilde{T}_1^t) = \hat{\tau}(t) + \alpha_c$$
$$\alpha_c = \frac{\hat{\tau}(t) - \hat{\tau}(T_1^t)}{Card(\tilde{T}_1^t) - 1}$$

While $\alpha < \alpha_c$, it is no use to prune off the tree at the node t, but as soon as $\alpha = \alpha_c$ pruning off the subbranch presents some interest because the error is the same and the tree is simpler ;

4. Do this for all t in T_1 and choose the node t in T_1 which minimizes this quantity α_c. Let α_1 be α_c. By pruning T_1 at the node t, we get $T_2 = T_1 - T_1^t$. Recursively, repeat 3. and 4. with T_2, get α_2, and so on until the root node.

Finally, we get by construction (see the critical cases) a sequence $\alpha_1 < \alpha_2 < ... < \alpha_{root}$ corresponding to the pruned trees $T_1 > T_2 > ... > T_{root}$. T_{root} consists only on the root node. But what is the optimal tree in this sequence ?

(8) tells us that the best pruned tree is the one with the minimum misclassification rate.

B Logistic regression

A Static results

The regression coefficients, their standard error, the confidence we can have in the value of the coefficients and their effect are available in Table 8. The regression coefficients of the dynamical study are not given here, there are too many coefficients because the date was included in the modeling.

B Theoretical framework

The main idea why the logit modeling seems to be relevant is that we want to model a binary event (surrender). Indeed, logistic regression analyses binomially distributed data of the form $Y_i \sim B(n_i, p_i)$, where n_i is the number of bernouilli trials and p_i the probability of "success" (surrender). If we denote by Y the variable to explain (i.e. the surrender decision), we have

$$Y = \begin{cases} 1, & \text{if the policyholder surrenders,} \\ 0, & \text{else.} \end{cases}$$

It is now possible to adapt the logistic regression equation to our environment and we get p as the probability to surrender :

$$\begin{aligned} logit &= \ln\left(\frac{P[Y=1|X_0=x_0,...,X_k=x_k]}{P[Y=0|X_0=x_0,...,X_k=x_k]}\right) \\ &= \beta_0 + \beta_1 X_1 + ... + \beta_k X_k \end{aligned}$$

Finally,

$$\left.\begin{aligned} \Phi(logit(p)) &= \Phi(\Phi^{-1}(p)) = p \\ \Phi(logit(p)) &= \Phi(\beta_0 + \sum_{j=1}^k \beta_j X_j) \end{aligned}\right\} (1)$$

$(1) \Rightarrow p = \Phi(\beta_0 + \sum_{j=1}^k \beta_j X_j)$.
This writing will help us to understand the expression of the likelihood function in C.

C The Newton-Raphson algorithm

The condition on maximizing the log-likelihood function (15) yields to the following

Table 8: Estimations of the logistic regression coefficients for "Mixtos" products. With confidential data, modalities increasing means the variable associated also increasing.

Coef. (var. type)	modality : correspondance	coefficient estimate	std error	p-value	effect
β_0 (continuous)		10.63398	1.48281	7.42e-13	> 0
	1 : [0,12] (in month)	0 (reference)			nul
	2 :]12,18]	-1.31804	0.15450	$< 2e - 16$	< 0
	3 :]18,24]	-2.66856	0.14016	$< 2e - 16$	< 0
	4 :]24,30]	-2.75744	0.14799	$< 2e - 16$	< 0
$\beta_{duration}$	5 :]30,36]	-3.09368	0.14294	$< 2e - 16$	< 0
(categorical)	6 :]36,42]	-3.54961	0.15080	$< 2e - 16$	< 0
	7 :]42,48]	-3.72161	0.14980	$< 2e - 16$	< 0
	8 :]48,54]	-4.10431	0.15772	$< 2e - 16$	< 0
	9 : > 54	-5.49307	0.14037	$< 2e - 16$	< 0
	Monthly	0 (reference)			nul
	Bi-monthly	0.92656	0.62071	0.135504	> 0
	Quarterly	-0.03284	0.10270	0.749148	< 0
$\beta_{premium\ frequency}$	Half-yearly	-0.22055	0.16681	0.186128	< 0
(categorical)	Annual	0.43613	0.10690	4.51e-05	> 0
(in month)	Single	-0.28494	0.38155	0.455177	< 0
	1 : [0,20[(years old)	0 (reference)			nul
	2 : [20,30[	0.28378	0.13912	0.041376	> 0
	3 : [30,40[	-0.01146	0.13663	0.933163	< 0
$\beta_{underwriting\ age}$	4 : [40,50[	-0.26266	0.14077	0.062054	< 0
(categorical)	5 : [50,60[	-0.42098	0.15136	0.005416	< 0
	6 : [60,70[	-0.66396	0.19531	0.000675	< 0
	7 : > 70	-0.75323	0.23417	0.001297	< 0
$\beta_{face\ amount}$	1* :	0 (reference)			nul
(categorical)	2* :	-5.79014	1.46592	7.82e-05	< 0
	3* :	-7.14918	1.46631	1.08e-06	< 0
$\beta_{risk\ premium}$	1* :	0 (reference)			nul
(categorical)	2* :	0.36060	0.11719	0.002091	> 0
	3* :	0.26300	0.14041	0.061068	> 0
$\beta_{saving\ premium}$	1* :	0 (reference)			nul
(categorical)	2* :	0.93642	0.13099	8.74e-13	> 0
	3* :	1.32983	0.14955	$< 2e - 16$	> 0
	PP con PB	0 (reference)			nul
$\beta_{contract\ type}$	PP sin PB	-16.79213	114.05786	0.882955	< 0
(categorical)	PU con PB	-7.48389	1.51757	8.16e-07	< 0
	PU sin PB	-12.43284	1.08499	$< 2e - 16$	< 0
	Female	0 (reference)			nul
β_{gender}	Male	-0.08543	0.04854	0.078401	< 0

* Note : for confidentiality reasons, the real ranges of the face amount, the risk premium and saving premium are omitted.

system of (k + 1) equations to solve

$$\begin{cases} \dfrac{\partial l}{\partial \hat{\beta}_0} = \sum_{i=1}^{n} Y_i - \Phi(\beta_0 + \sum_{j=1}^{k} \beta_k X_k) = 0 \\ \dfrac{\partial l}{\partial \hat{\beta}_j} = \sum_{i=1}^{n} X_{ij}(Y_i - \Phi(\beta_0 + \sum_{j=1}^{k} \beta_k X_k)) = 0 \end{cases}$$

$\forall j = 1, ..., k$.

The problem is that it is not in a closed form, we need to use an algorithm (often Newton-Raphson) to find its solution. In SAS and R soft-

ware, the Newton-Raphson algorithm to solve it is included and uses the following iterative process :

$$\beta^{(i+1)} = \beta^{(i)} - \left(\frac{\partial^2 \ln(L(\beta))}{\partial\beta\partial\beta'}\right)^{-1} \times \left(\frac{\partial \ln(L(\beta))}{\partial\beta}\right) \tag{10}$$

When the difference between $\beta^{(i+1)}$ and $\beta^{(i)}$ is less than a given threshold (say 10^{-4}), the iteration stops and we get the final solution.

D   Estimating the variance matrix

The variance matrix Z of coefficients $\hat{\beta}$ is

$$\begin{pmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) & \cdots & Cov(\hat{\beta}_0, \hat{\beta}_k) \\ Cov(\hat{\beta}_1, \hat{\beta}_0) & Var(\hat{\beta}_1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ Cov(\hat{\beta}_k, \hat{\beta}_0) & Cov(\hat{\beta}_k, \hat{\beta}_1) & \cdots & Var(\hat{\beta}_k) \end{pmatrix} \tag{11}$$

and is estimated by the inverse of the information of Fisher matrix, given by

$$I(\beta) = -\mathbb{E}\left[\frac{\partial^2 \ln(L(\beta))}{\partial\beta\partial\beta'}\right].$$

So we have a pretty result : the latter term also appears in the Newton-Raphson algorithm, so we can estimate the regression coefficients and their variance matrix together.

The maximum likelihood estimator $\hat{\beta}$ converges and is asymptotically normally-distributed with mean the real value of β and variance the inverse of the Fisher matrix $I(\beta)$.

The term in the expectation is called *Hessian matrix* and is also used in the significance tests of the regression coefficients β.

E   Deviance and tests

E.1   Statistic evaluation of the regression

To check the relevance of the model, we classically use the statistic of the log-likelihood ratio test : the first assumption of this test is $\beta_1 = \beta_2 = ... = \beta_k = 0$ (H_0) ;
And the alternative hypothesis is "at least one regression coefficient is not equal to 0" (H_1).
Now let us denote by l(β) the log-likelihood of the logistic regression model with $k + 1$ regression coefficients, and the log-likelihood of the simplest logistic regression model (with only the constant term associated to β_0) by l(β_0), the statistic of the log-likelihood ratio is

$$\Lambda = 2 \times \Big(l(\beta) - l(\beta_0)\Big) \tag{12}$$

This statistic follows a χ_k^2, a chi-square law with k degrees of freedom (d.f.).
To conclude, if the "p-value" is lower then the expected threshold of confidence (e.g. 5%), the model is globally statistically significant and H_0 is rejected.
More intuitively, sometimes the R^2 coefficient of MC Fadden is also used : $R^2 = 1 - \frac{l(\beta)}{l(\beta_0)}$.
As one could expect, if this coefficient is closed to 0 it is because the ratio is closed to 1, and then the log-likelihood of the complete model is closed to the simplest model one which means that this is not significant to have explanatory variables.
On the contrary, if R^2 is closed to 1 it means that there is a huge difference between the two model. In this case, the complete model is the best one.

E.2   Relevance of a given explanatory variable

The idea of this test is to compare the value of the estimated coefficient β_j (associated to the explanatory variable X_j) to its variance. This variance is taken from the Hessian matrix defined above.
Here the first assumption is : $\beta_j = 0$ (H_0) ;
Otherwise the alternative one is then : $\beta_j \neq 0$ (H_1).
We use the Wald statistic which follows a χ_1^2 to do this test : $\Lambda = \frac{\hat{\beta}_j^2}{Var(\hat{\beta}_j)}$.
Let us choose 5% as confidence threshold, and let us denote by $\chi_{95\%}^2(1)$ the 95% quantile of the chi-square law with 1 d.f. H_0 is true if the ratio is lower than this quantile, otherwise H_1 is confirmed.