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

X Milhaud, S. Loisel, V Maume-Deschamps

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

X. Milhaud*, S. Loisel and V. Maume-Deschamps
Universite de Lyon, Universite Lyon 1, ISFA, Laboratoire SAF

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 endowment policies from a life insurance portfolio: the Logistic Regression model and the Classification And Regression Trees model. 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 the portfolio in classes regarding the surrender risk. Results make it explicit that duration and profit benefit option are essential. Finally, we explore and discuss the main differences of both models in terms of operational results.

Résumé - Certaines caractéristiques jouent un rôle majeur dans la décision de l'assuré de racheter son contrat d'assurance. Ses conditions de souscription, son âge, sa profession ainsi que d'autres facteurs propres à sa situation influencent ses décisions. Deux modèles de segmentation nous ont permis de développer ces idées sur les contrats mixtes d'un portefeuille d'Assurance-Vie : les arbres de classification et de régression, et la régression logistique. Nous présentons dans un premier temps les fondamentaux de chacun des modèles ainsi que leurs hypothèses et limites. Puis nous testons différents facteurs comme possibles déclencheurs du rachat, dans le but de segmenter le portefeuille en classe de risque : l'ancienneté fiscale et la la garantie de participation au bénéfice apparaissent comme des éléments essentiels. En dernière partie, nous discutons des différences entre les deux modélisations en termes de résultats numériques et d'un point de vue opérationnel.

I Introduction

Understanding the dynamics of surrender (sometimes lapse) rates is a crucial point for insurance companies, who may look towards several problems. First, policy lapse might make the insurer unable to fully recover her initial expenses due to costs of procuring, underwriting, and issuing new business. Actually the insurer pays expenses at or before the contract issue date but earns profits over its life, so that she might incur losses from early lapsed policies. Indeed, the time profile is very important because the costs of a surrender change over it. 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 the so-called "moral hazard" and "adverse selection" where there only remain "bad risks" (Bluhm (1982)). Third, massive early surrenders or policy lapses pose a liquidity threat to the insurer who is subjected to interest rate risk (because interest rate is likely to change over the period of the contract). Imagine that financial events and a general loss of confidence of investors provokes a high increase of interest rate, say r_t plus a liquidity premium λ_t. Borrowing money in order to pay back the surrender values to policyholders is thus more expensive for the insurer who could undergo a series of undesirable 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 earn money.

What causes lapses has attracted certain academic interest for some time. Originally two main hypotheses have been suggested to explain lapse behavior. On one hand, the emergency fund hypothesis contends that policyholders use cash surrender value as emergency fund when facing personal 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: the investor acts as the opportunity cost for owning insurance contracts. Interest rates rise makes equilibrium premiums to decrease, so there is definitely 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. Engle & Granger (1987) suggest to separate the potential long-term relationship between lapse rate, interest rate and unemployment rate from their short-term adjustment mechanisms thanks to the cointegrated vector autoregression approach. From a financial engineer-

*email address: xavier.milhaud@gmail.com

ing perspective, it may be difficult to accept that kind of arbitrage opportunities are not used by policyholders. Even if policyholders are still far from being rational, one cannot exclude that in the near future, policyholders may become more rational and may be helped by journalists or financial analysts to optimize the use of their life insurance portfolios.

Modeling precisely 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 (fixed maturity) and whole life products: assumptions about it have to be made in Asset and Liability Management, particularly for projections of the European Embedded Value (EEV). Product designers generally assume an expected level of lapsation thanks to *data mining techniques*. To fully exploit the information of an insurance company dataset is typically a hard task for practitioners, who must deal with various sources of complexity: missing data, mixture of data types, high dimensionality and heterogeneity between policyholders. This complexity often prevents companies from getting to the maximum productivity because they only collect part of the information from observations. The challenge is thus to select salient features of the data and feed back summaries of the information.

The use of two complementary segmentation models, the Classification And Regression Trees (CART) model by Breiman et al. (1984) and the Logistic Regression (LR) model (see Hilbe (2009)), could give clues to managers regarding the surrender risk, in order to adapt product features and penalty fees. In the literature, 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 compared with policy features and policyholder's characteristics in this framework.

Our paper aims at i) determining what segmentation method could be the most appropriated to an insurance portfolio dataset by looking at the gap in classification errors between CART and LR, ii) investigating potential surrender triggers in a classical economic regime. Those triggers could be very different in a disturbed period (financial crisis, reputation issues): we clearly have in mind that there exists a bias in the segmentation analysis because we do not consider dates (and thus forget cohort effects) as well as exogenous factors possibly playing a (big) role on surrender behaviors (financial indexes for instance). We go back to this remark and suggest some extension of the application with external dynamic factors at the end of the paper to make temporal predictions. However this is absolutely not our purpose here.

The paper is organized as follows: we first present theoretical results about CART method that are useful for our practical problem in Section II. Section III more briefly recalls the basics of logistic regression, as it has been more widely used in many fields. In Section IV, we compare both approaches on a real-life insurance portfolio embedding endowment contracts and discuss their limits. We provide numerical indicators, and determine the main reasons for a policyholder to surrender in a classical economic situation as well as predictors of the individual surrender probability. Section V finally presents some possible further extensions.

II The CART model

The CART method, an iterative and recursive flexible nonparametric tool, 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. In classification issues, binary trees provide an illuminating way of looking at data and results. The novelty of CART is in its algorithm to build the tree: there is no arbitrary rule to stop its construction, contrary to the previous uses of decision trees. The two main goals of a classification process are basically to uncover the predictive structure of the data and to produce an accurate classifier. Depending on the problem, there is usually a trade-off to find between the predictive power and the fit. The opportunity to make predictions particularly with regression trees 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 which are not developed herein). We find interesting to provide a clear chronological methodology when using CART as it is somewhat quite difficult to get it summarized in the literature.

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). Let*

- *$\forall x \in \mathbb{X}$, the classification process class(.,ϵ) classifies x in a group $j \in C$.*
- *The prior 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) $= card\{(x_n, j_n) \in \epsilon, x_n \in t\}$.*
- *$N_j(t)$=$card\{(x_n, j_n) \in \epsilon, j_n$=$j$ given that $x_n \in t\}$.*

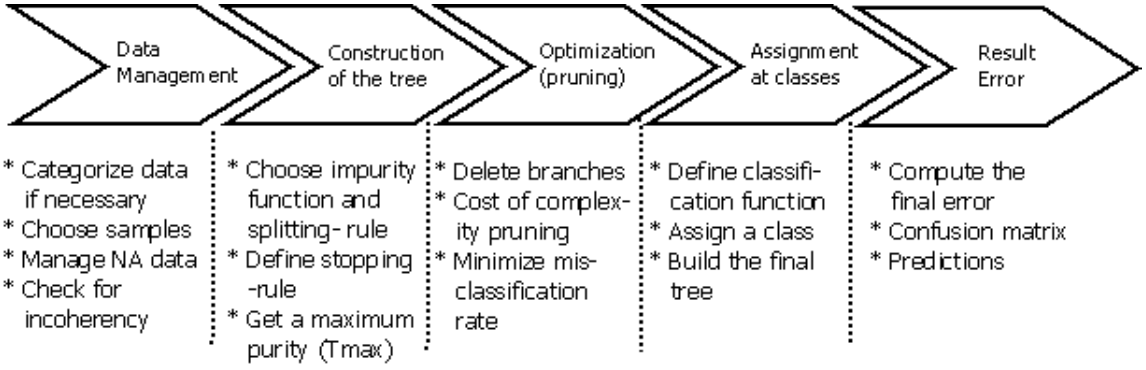


Figure 1: Ordered steps of CART procedure

- *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* $p(t) = \sum_{j=1}^{J} p(j,t)$.
- $P(j \mid t)$ *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 gender. The policyholder whose characteristic is x is either a female or male, and t could be the modality "female" (binary rules specification is provided in Appendix $A.II.1$).

Actually we start with $\mathbb{X}$ called *root* which is divided into two disjoint subsets called *nodes* denoted by t_L and t_R. Each node is then divided in the same way (if it has at least two elements). At the end, we have a partition of $\mathbb{X}$ into 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 illustration in Figure 2).

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 $A.II.2$. In our case, the impurity in T of a node t is the quantity

$$impur(t) = g(p(1|t), p(2|t), ..., p(J|t)), \tag{1}$$

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

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

where Impur(t) = p(t)impur(t).

A splitting-rule Δ of node t gives p_L=$p(t_L)/p(t)$ observations in t_L and p_R=$p(t_R)/p(t)$ observations in t_R. We want to maximize the *purity variance*:

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

Each time a split is made, the purity of the tree has to increase. Then it looks quite natural from this process to require the following constraint

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

Do we always respect it? "Yes" if g is concave. In our applications and in most of them, we consider the Gini index of diversity (Appendix $A.II.3$), which 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 (Appendix $A.II.3$), but there is no convincing justification for a particular choice (they all fulfill the requirements of an impurity function). Besides, the properties of the final tree are usually surprisingly insensitive to the choice of this impurity function! For further explanations, see Breiman et al. (1984). Traditionally, the optimal division $\Delta_t^\star$ of a node t stands for

$$\Delta_t^\star = \underset{\Delta \in D}{argmax}\ (\delta\ impur(\Delta, t)), \tag{4}$$

where $argmax\ (\delta\ impur(\Delta, t))$ 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. 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. 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 descen-

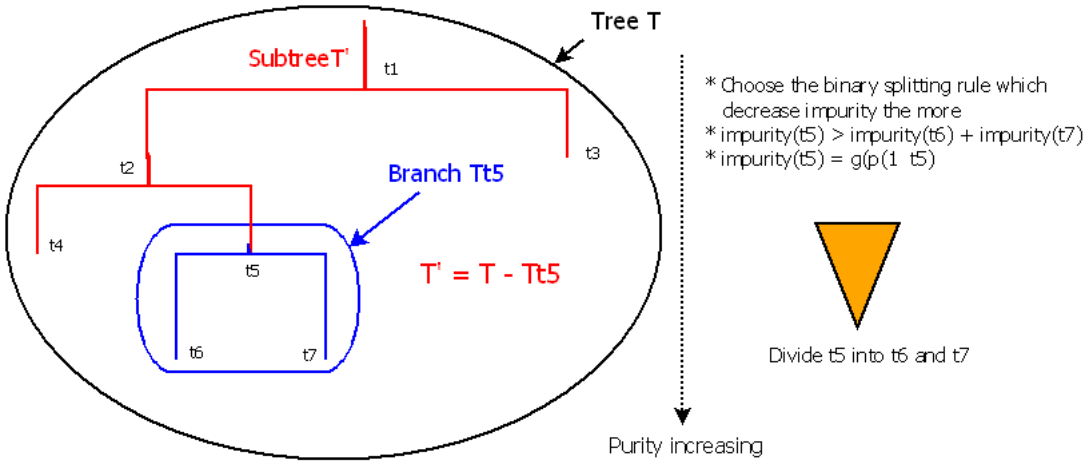


Figure 2: Construction of a binary tree

dant 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 impurity fluctuation F of the tree T is

$$\begin{aligned} F &= Impur(t) - Impur(t_L) - Impur(t_R) \\ &= \delta Impur(\Delta, t) \\ &= p(t)\delta impur(\Delta, t). \end{aligned} \quad (5)$$

Indeed, it results from the probability to be present in this node multiplied by the decrease of impurity given by the split Δ. The following step is now: when do we have to stop the splits? The user can choose among different rules to stop the division process. Some of them are natural, others are purely arbitrary: i) obviously, the divisions stop as soon as the observations of the explanatory variables are the same in a given class (not possible to split once more); ii) define a minimum number of observations in each node (the smaller it is, the bigger the number of leaves is); iii) choose a threshold λ as the minimum decrease of impurity: let $\lambda \in \mathbb{R}_+^*$,

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

Actually there is no stopping-rule in CART; we build the largest tree (T_{max}) and we prune it. A comprehensive procedure to make it is provided in Appendix $A.II.6$.

A.2 The classification function

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

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

with $B_j = \{x \in \mathbb{X}; class(x, \epsilon) = j\}$, so that we can class the policyholder (given its characteristics "x") in a set B_j to predict the response. 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 belong, one uses the following rule:

$$class(x, \epsilon) = \underset{j \in C}{argmax}\ p(j|t). \quad (6)$$

We recognize the so-called *Bayes rule* which maximizes the *a-posteriori* probability of being in class j given that we are in the node t. This process defines the classification function and therefore allows predictions. The estimation of assigning a wrong class to an observation present in the node t (with respect to the class observed for this observation) then reads

$$r(t) = 1 - class(x, \epsilon) = 1 - \max_{j \in C}\ p(j|t), \quad (7)$$

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 given that we are in the node t. It turns out that the general misclassification rate is

$$\hat{\tau}(T) = \sum_{t \in \tilde{T}} \hat{\tau}(t). \quad (8)$$

To put it in a nutshell, we can summarize the four stages to be defined in the tree growing procedure:

1. a set of binary questions like {is $x \in S$?}, $S \in \mathbb{X}$ (quite hard task numerically speaking),
2. an impurity function for the goodness of split criterion (arbitrary choice),
3. a stop-splitting rule (natural stopping-rule is then

1 case by leaf, hard task because arbitrary choice),

4. a classification rule to assign every terminal node to a class (easy to define).

As we have seen, CART builds the maximal tree T_{max} and then prune it (to avoid arbitrary stopping-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, but various estimators exist in the literature (Ghattas (1999)). The expression of the misclassification rate depends on the learning sample chosen to run the study (details in Appendix $A.II.4$). There exists for now three types of prediction error estimate:

- the **resubstitution** estimate of the tree misclassification rate: we consider all observations ϵ in the learning sample. Achievements are overestimated because we class the same data (as those used to build the classification function) to test the efficiency of the procedure. This is of course the worse estimator for predictions.
- the **test sample** estimate: let W $\subset \epsilon$ be a test (witness) sample whose size is $N' < N$ (N is the size of ϵ). Usually $N^{'}$=$N/3$ so that the size of the learning sample equals $2/3 * N$. The learning sample is used to build the classifier and the test sample is used to check for its accuracy. This estimator is better but requires a larger initial dataset.
- the **cross-validation** technique: suppose that ϵ is divided into K disjointed subgroups $(\epsilon_k)_{1 \leq k \leq K}$ of approximately same size. Let us define K new learning datasets such that $\epsilon^k = \epsilon - \epsilon_k$. The idea is to build a classification function on each sample ϵ^k such that $class^k(.) = class(., \epsilon^k)$. This technique is highly recommended when we lack data, because it is more realistic (final error is the mean of K errors).

Hereafter, $\tau(T)$ 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: i) no restriction on the type of data (both categorical and numerical variables accepted); ii) simple final form, compactly stored and displayed; iii) 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, monotonous transformations of ordered variables do not alter the results. CART is not a parametric model and thus do not require a particular specification of the relationship nature between the outcome and the predictor variables (no linearity assumption for example). Moreover, it often successfully identifies interactions between predictors.

However, each split is based on one single variable and when the class structure depends on combinations of variables, the standard tree algorithm will do poorly at uncovering this structure. Besides, the effect of one variable can be hidden by others when looking at the final tree. To avoid this, there exists solutions as ranking the variables in function of their potential in the splitting process: this is what is called the *secondary* and *surrogate splits* (also used with missing data, see Breiman et al. (1984)). There also remains some additional issues: i) 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: overfitting); ii) CART gives 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 dataset can cause different classifiers, a big constraint to make predictions because of its instability.

For sure, we would like to avoid that a variable could be considered very important with a given dataset, and be absent in the tree in another quasi-similar one! The first point i) can be solved thanks to the introduction of a complexity cost in the pruning algorithm (see Appendix $A.II.6$) and the second one ii) using cross-validation, *bagging predictors* or *arcing classifiers*.

C Bagging predictors

The bad robustness of the CART algorithm when changing the original dataset has already been discussed. To experiment different optimal final classifiers can be challenged using resampling techniques. 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. The final classifier cannot be represented as a tree, but is extremely robust.

This led to the development of "Random Forest" algorithms which were developed by Breiman (2001) and fol-

lows the same idea as bagging predictors: 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, the best the ability of this algorithm (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]. We prefer to use it in our applications instead of the `ipred` package [2] because it enables to compute the importance of each explanatory variable. For more precision on these theories, please refer to Breiman et al. (1984) and Breiman's webpage.

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 to predict the probability of occurrence of a binary event by fitting data (either numerical or categorical) to a logistic curve. The logistic regression is a choice model used for binomial regressions, and is mainly used in medical and marketing worlds (for instance to predict the customer's propensity to cease a subscription). Actuaries sometimes also model the mortality of an experienced portfolio with it, which is a way for them to segment their portfolio regarding death risk. Here, the goal is to model the surrender decision of 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 [0,1]:

$$\Phi(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z}. \tag{9}$$

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 the regression coefficients by $\beta = (\beta_0, \beta_1, ..., \beta_k)'$.

Remark 1. *:*

- $\forall i = 1, ..., k; \beta_i$ *represents the regression coefficient associated to the risk factor i (say age 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 exists the polytomic or multinomial regression when the response variable has more than two levels,*
- *other link-functions have been proposed historically,*
- *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.*

B Estimation of parameters

To estimate the regression coefficients, the ordinary least square estimation is the most famous technique. However, the fact that we want to estimate a probability (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. Anyway, it can be shown that maximum likelihood and least square principles are equivalent in this scope.

B.1 Maximum likelihood estimation

Let n be the number of independent observations. By definition, the likelihood function for a binomial law is

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

where Φ is defined in (9). The log-likelihood then reads

$$\ln(L(X, \beta)) = \sum_{i=1}^{n} Y_i (X_i \beta') - \ln(1 + e^{X_i \beta'}) \tag{10}$$

The maximum likelihood estimator $\hat{\beta}$ satisfies $\frac{\partial \ln(L)}{\partial \hat{\beta}} = 0$. This condition yields to a system of equations that are not in a closed form. We usually run the Newton-Raphson algorithm to find the solutions (see Appendices $B.III$ and $B.IV$ for further details).

B.2 The final probability

The individual estimation of probability to surrender is inferred from the previous estimates of coefficients,

$$\hat{p} = \Phi(\hat{\beta}_0 + \hat{\beta}_1 X_1 + ... + \hat{\beta}_k X_k), \tag{11}$$

where the $\hat{\beta}_i$ are the regression coefficients estimated by maximum likelihood. Thus each insured has her own estimated probability to surrender given its characteristics. We now want to determine the confidence interval for the surrender probability on the whole portfolio. In

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

a collective framework, the usual way is to use the Binomial law approximation which considers that the number of surrenders among n policyholders follows a Normal distribution. However this technique requires that: i) $n \to \infty$ (big size of portfolio), ii) probability p_i to surrender is comparable for all i individuals (homogeneity). The first point is usually not a problem in insurance (portfolios are often huge by nature). The second point is a direct consequence of the Central Limit Theorem (CLT): the sum of i.i.d. random variables follows a Gaussian law. Actually a portfolio is almost always heterogeneous. Anyway, imagine that the n policyholders in the portfolio are divided into i homogeneous groups (of size n_i) of policyholders. Within each group i, policyholders are considered independent and have the same characteristics: they are therefore homogeneous (same probability p_i to surrender). The number of surrenders N_i^s in group i embedding n_i policyholders is thus binomially (by property) or normally (CLT) distributed (sum of i.i.d. Bernouilli variables), although the assumption of independence is quite wrong because the environment is likely to affect lots of policyholders in the same time (see Loisel & Milhaud (2011) for further details). Hence,

$$\mathbb{E}[N_i^s] = \sum_{i=1}^{n_i} p_i = n_i p_i \text{ (binomial law prop.)}, \quad (12)$$

$$\mathbb{V}ar[N_i^s] = \sum_{i=1}^{n_i} p_i(1-p_i) = \sum_{i=1}^{n_i} p_i q_i = n_i p_i q_i. \quad (13)$$

From (12) and (13) we can get the confidence interval of $\hat{p_i} = N_i^s/n_i$ within the i^{th} homogeneous group by using the one of a Normal standard distribution. The total number N^s of surrenders over the whole portfolio is the sum of surrenders in those homogeneous groups: $N^s = \sum_i N_i^s$. The Normal law is stable under summation, so that N^s is still normally distributed. Finally, a good approximation of $\hat{p} = N^s/n$ is

$$\hat{p} = \sum_i N_i^s/n \sim N\Big(\frac{1}{n}\sum_i n_i p_i, \frac{1}{n^2}\sum_i n_i p_i(1-p_i)\Big),$$

which yields to the confidence interval (at level 5%)

$$[A - 1.96 \times B, A + 1.96 \times B] \quad (14)$$

where $A = \dfrac{\sum_i n_i p_i}{n}$, $B = \sqrt{\dfrac{\sum_i n_i p_i(1-p_i)}{n^2}}$, i is the index of the homogeneous group, and p_i is the estimated probability to surrender within group i.

B.3 Deviance and tests

The most famous tests are the likelihood ratio test and the Wald test, they are detailed in Appendix $B.V$.

C Interpretations

The regression coefficients values give us some information on the 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 variables). 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 it decreases the probability of this outcome. A large $\beta_i/\sigma(\beta_i)$ (where $\sigma(\beta_i)$ denotes the standard deviation of the coefficient estimation) means that the risk i strongly influences the probability of the outcome, and conversely. The regression coefficients have to be compared to the reference profile, for which $\beta = 0$ except for the intercept.

Practitioners are used to focusing on the odd-ratio indicators: they represent the ratio of probabilities $\frac{p}{1-p}$. Let us see an example to understand this quantity.

Example 1. *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. p/q=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* 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 (**??**) we get for one policyholder $p/q = e^{\beta_0+\beta_1 X_{age}+\beta_2 X_{region}}$. All terms disappear between both policyholders except age, the odd-ratio between them aged 40 and 30 years old is thus*

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

Generally speaking, we notice that a unit additive change in the values of explanatory variables should change the odds by constant multiplicative figures. The odd-ratios represent the difference in terms of surrender probability when explanatory variables change, and thus are a very useful operational tool to define risk classes.

D Limits of the model

Required assumptions define the main limits of the model. The policies $(Y_i|X_i)$ are considered conditionally independent with respect to the explanatory variables. Explanatory variables must also 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% (otherwise singularity in matrix inversion). Modalities of a categorical variable are considered independent, which is generally true except in case of erroneous data. Moreover, a lot of data should

be available for the robustness of the modeling. Well, this is not really the point here because insurance portfolios used to be large enough. However, applying the logistic regression over a whole portfolio of life-insurance contracts could lead us to very strange results. Indeed, if this is a run-off portfolio (no new business) which covers a very long period (say 50 years), almost all the policyholders would have lapsed and the regression would make no sense! We checked that this is not the case in our application (new business is being issued all along the observation period).

To sum up, 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 independence assumption between policyholders, but a crucial advantage is the opportunity to make precise predictions. Some examples of application can be found in Huang & Wang (2001) and Kagraoka (2005). Other quasi-similar segmentation models like Tobit model (Cox & Lin (2006)) or Cox model (Cox (1972)) could have been explored. For further details, a comparison of these different models is available in Austin (2007).

IV Application on a Life Insurance portfolio

Depending on the country, the database provides information on policyholder's characteristics (birth date, gender, marital status, smoker status, living place...) and policy features (issue date, termination date, type of contract, premium frequency, sum insured, distribution channel...) of life insurance contracts. Here, a real life portfolio was collected thanks to the Spanish entity of a large French insurer. In our study we have information on the gender, the birth date of the policyholders, the premium frequency, the face amount, the premium; the type of contract, its issue date, its termination date and the reason of the termination. The face amount is an indicator of the policyholder's wealth, and the premium 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. Thus with certain endowment products covering the death, the risk premium is the mortality rate times the sum-at-risk (amount added to the reserve in case of death), all discounted. The saving premium is the investment made by the policyholder.
We used the package `rpart` of `R` to implement the CART method and obtain the results in the sequel. The functions to implement the logistic regression are included in the core of the `R` programs.

A Static analysis

We mean by static analysis a "photograph" of the portfolio in December 2007. The types of long-term contracts are either pure saving or endowment products, but we focus on the 28506 endowment policies hereafter. These Term Annually Renewable (TAR) products cannot be surrendered in the first year following the underwriting (except in very special cases), and there is no tax constraint in Spain concerning life insurance saving contracts. It means that the policyholders can surrender their contracts at each anniversary date without any fee, but are penalized otherwise. We will see later (in Figure 6) that these features are big incentives and drive the surrender profile in function of the contract duration.

The study covers the full period 2000-2007. This means that the characteristics of policyholders and contracts that we extract from the database are those observed either at the surrender date or in December 2007 (if the policyholder has not surrendered yet). Recall that we first would like to have an idea of the possible triggers of the surrender decision, by explaining *surrenders* as a function of *other variables*. It will thus enable us to detect the "risky" policyholders at a given date.

Remark 2. *The static analysis raises some burning questions: what is the composition of the portfolio? Is it at maturity? What is the part of new business?*
For example if the duration of the contract is one of the main explanatory factor for surrenders (and it is!), one has to be careful to cover a sufficiently long period to experience a normal surrender rate, say 10% 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 surrender rate would be very close to 0%. Indeed we do not have a dynamical view of the phenomenon, this static analysis is just a simple way to point out the more discriminant factors of the surrender decision, even if some bias still exists as we have seen in Introduction. We go back to this problem in Section IV.B where the monthly study reflects that policyholders often wonder whether they should surrender their contract (say at least twice a year). However, eight years of experience here seems to be ok for our purpose.

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:

- CART gives us the most discriminant variables regarding the surrender in descending order (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 (although we could get the probability to be in each class and thus the probability to surrender);
- LR offers a more precise result, the probability (propensity) for this policyholder to surrender her contract given its characteristics, and sensitivities of surrender decisions when explanatory variables change

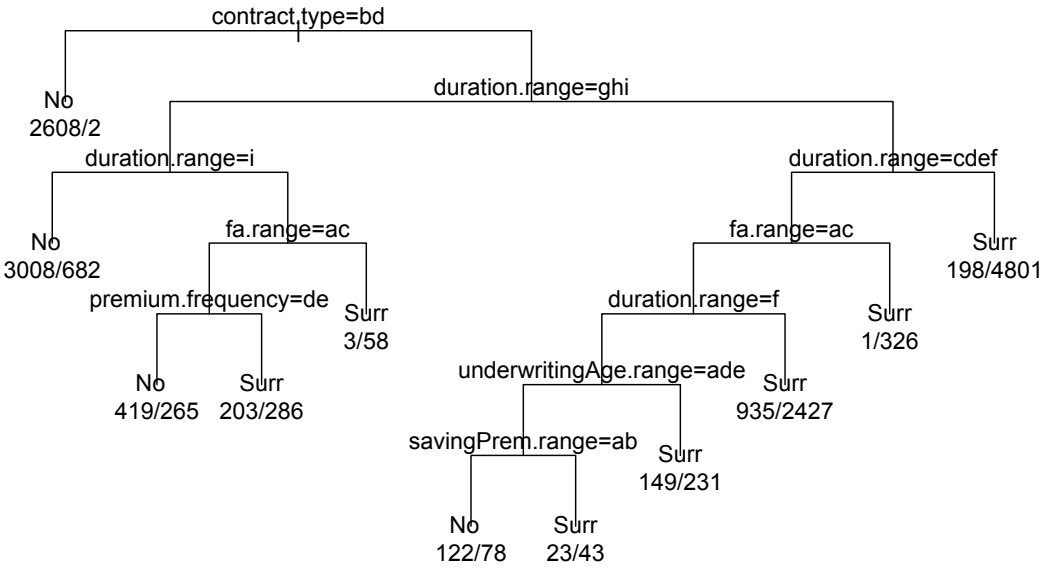


Figure 3: The final classification tree. Binary response variable: surrender. The first splitting-rule *contract.type* = *bd* means that the contract type is the most discriminant variable (*bd* correspond to the 2^{nd} and 4^{th} categories, like in alphabetic order). Continuous explanatory variables have been previously categorized for the modeling.

(thanks to the odd-ratios technique and the regression coefficients).

A.1 CART results

In R, one performs the analysis 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 does not seem important because results should not much differ. There is no misclassification cost (see Appendix $A.II.5$) in our application.

We proceed like in theory:

1. first, T_{max} is built with no complexity cost (by setting 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 Section $II.A.3$, which is useful to estimate better the real prediction error but not to build an optimal tree). We randomly create the learning and validation datasets, 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%, corresponding to non diagonal terms of the confusion matrix given in Table 1. This tree has too many leaves and its representation is too complex, so that we have to prune it. The choice of the complexity parameter α in the pruning algorithm (see Appendix $A.II.6$) is a trade-off between the final size of the tree and the minimum misclassification rate required by the user. Table 7 and Figure 9 in Appendix $A.I$ 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. Notice that minimizing the learning error (by cross-validation) and its standard deviation requires setting $\alpha \in]1.04e^{-04}, 1.30e^{-04}]$, but the corre-

1. http://cran.r-project.org/web/packages/rpart/index.html, developed by T. M. Therneau and B. Atkinson

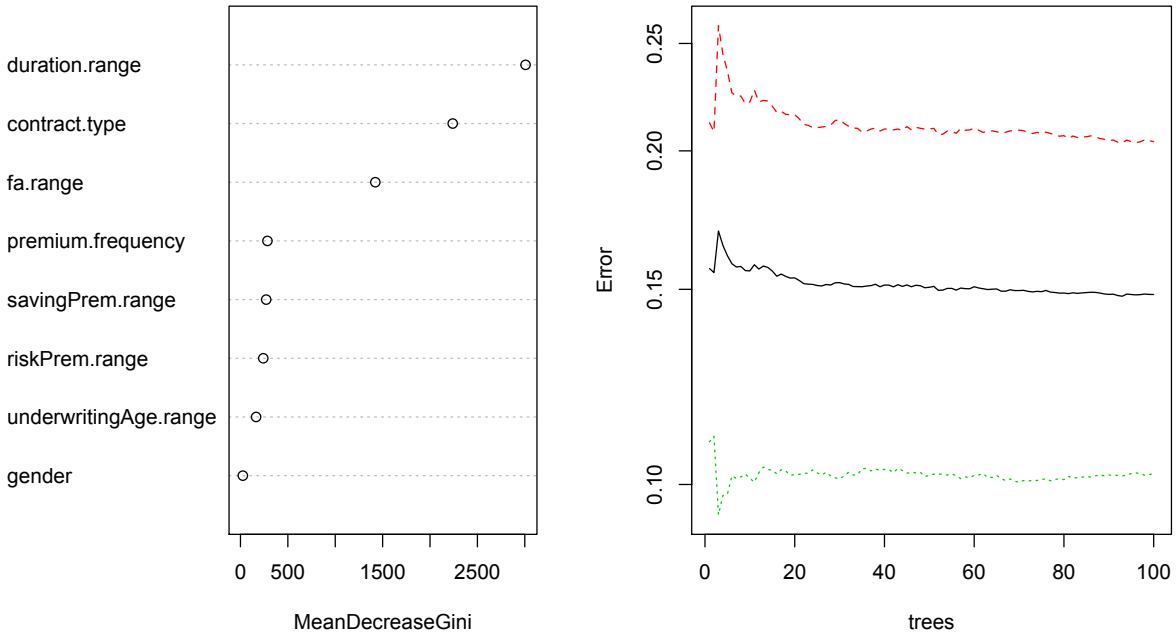
Table 1: Confusion matrix (T_{max}) on validation sample.

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

Table 2: Confusion matrix (pruned), validation sample.

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

Figure 4: 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".



sponding 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. The corresponding tree is plotted on Figure 3. 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. Selected variables in the tree construction are the contract type, the duration, the face amount, the premium frequency, the saving premium and the underwriting age. Finally, gender and risk premium don't appear in the final tree, because they should not be relevant. 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 difference between these numbers, the better the segmentation. 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).*

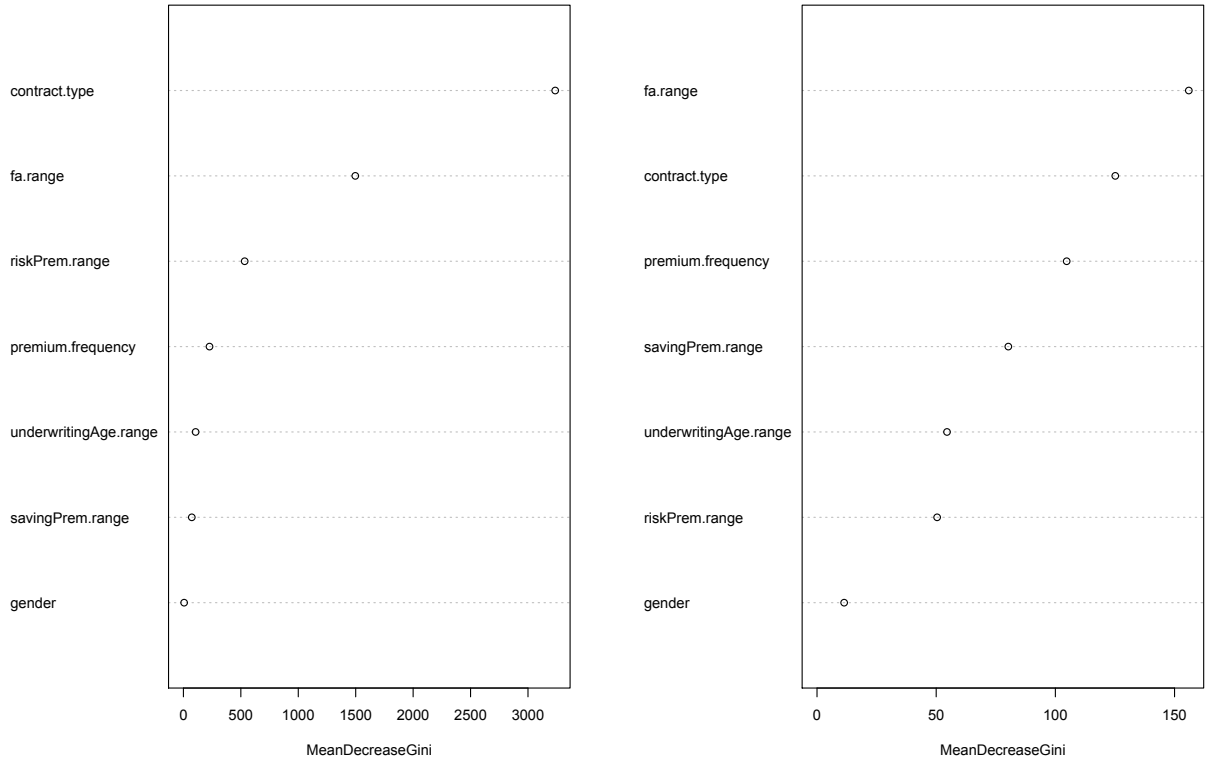
Example 2. *Let us consider someone whose characteristics are a periodical premium and a contract with profit benefit. The duration of her contract is today observed in the seventh range and her face amount belongs to the second range. The tree predicts that this policyholder is today in a risky position given its characteristics (58/61 $\simeq$ 95% of people with these characteristics have surrendered their contract).*

Looking at Figure 3, it is clear that the most discriminant factor regarding the surrender risk here is the profit benefit option. The misclassification rate (learning error) of this tree is 15% (33.1% $\times$ 45.4%, where 45.4% is

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: Importance of explanatory variables excluding the duration effect. On the left policyholders whose final contract duration corresponds to peaks in Figure 6, and others on the right.



the root error when no split) according to relative errors in Table 7 presented in Appendix $A.I$. The prediction error can be estimated via the confusion matrix in Table 2. It is quite satisfying: 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 interesting because pruning the tree from 175 leaves to 11 leaves causes a less than 1%-increase of the prediction error!

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 correlation and the strength; thus there is an optimal m that can be found with the *out-of-bag* error. We cannot represent the new final classifier as a tree, but it gives best results (all these concepts are explained on Breiman's webpage [1]). Table 3 summarizes the results on the entire original dataset (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 4, as well as the necessary number of trees in the forest for the *out-of-bag* error to be stabilized (which seems to be here about 50 trees). These results confirms what we expected: the duration and the type of contract are the most meaningful variables to explain the policyholder's decision to surrender her life insurance contract. To be sure that the importance of these factors is not biased by the duration effect, we decided to run the analysis excluding the duration and splitting data into two subsets: policyholders whose final contract duration corresponds to peaks in Figure 6, and others policyholders left. We thus look at surrenders due to penalty fees as well as other surrenders without penalty constraints. Not surprisingly Figure 5 shows that we get the same most important factors (with the order slightly differing), meaning that the duration effect is not correlated to another risk factor

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 variables (e.g. duration) have previously been categorized.

Odd-ratios	Reference	Other modalities							
Duration	[0,12]	]12,18]	]18,24]	]24,30]	]30,36]	]36,42]	]42,48]	]48,54]	> 54
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			
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		
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*						
surrenders	5361	684	3154						
empirical OR		0.14	0.12						
modeled OR		0.003	0.0008						
Risk prem.	#1*	#2*	#3*						
surrenders	3941	2987	2271						
empirical OR		1.50	0.92						
modeled OR		1.43	1.30						
Saving prem.	#1*	#2*	#3*						
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					
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.

and thus does not lead to biased viewpoints.

A.2 The LR model

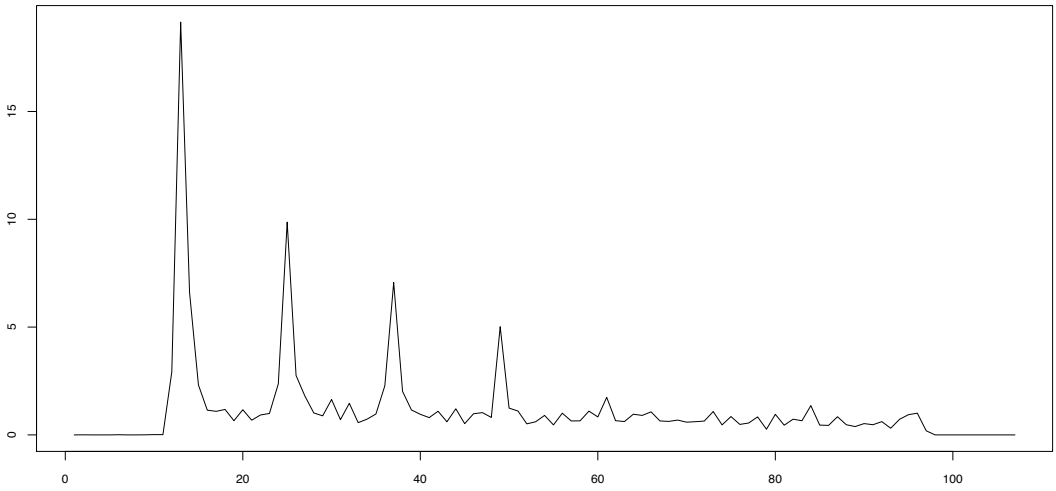
Consider that X is the matrix of explanatory variables for each observation, that is to say a line of the matrix X represents a policyholder and a column represents the observed value for a certain risk factor (e.g. the age). The response vector $\mathrm{Y} = (Y_1, Y_2, ..., Y_n)'$ represents the surrender decisions of the 28506 insureds (policyholders). In the classical regression framework, the problem can be written in the matrix form:

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

We ran the logistic regression in `R` thanks to the function `glm`. The output of the model is the effect of each variable, the standard deviation of the estimated regression coefficients, and the deviance of the model (see Appendices $B.III$, $B.IV$ and $B.V$).

Categorical variables are split into dummy variables corresponding each one to a modality (same process as in CART) to build the so-called "design matrix". 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). The R procedure `stepAIC` from the package `MASS` allows us to drop non significant variables from the model and to add relevant ones. We finally get the optimal model with the minimum number of relevant variables. The learning sample still contains the randomly chosen 16868 policyholders and the validation sample the 11638 ones left. As usual, the regression coefficients were estimated on the learning sample whereas the predictions were made on the validation dataset. Table 8 in Appendix $B.I$ summarizes the regression coefficients of the explanatory variables, their standard deviation, and the p-value of the Wald test (confidence in the estimation and relevance of the

Figure 6: Surrender rate (%) VS duration (in month) for Mixtos products. Effect of penalty fees and tax constraints (contracts can be surrendered at each anniversary date without fees, which explains the peaks).



regression coefficients, see Appendix $B.V.2$). We can deduce from the estimates of regression coefficients that the variables which seem to have the main effects (biggest absolute values) are once again the duration, the contract type, but also the face amount. This suggests that the results are consistent with CART, and that historical data should no longer be used if the surrender profile with respect to contract duration changes (due to regulatory's decisions for example). The odd-ratios presented in Section $III.C$ should be compared to 1 (value corresponding to the reference category). Looking at Table 4, we clearly see that the modeled odd-ratios are a quite bad representation of the reality: they are very different from the empirical ones (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 all other things being equal. The experience shows that in fact they are 3.28 times more likely to surrender! The good point is that the estimated odd-ratios very often have the same trend as the observed ones (as compared to the reference category). This is the case with duration: Figure 6 shows the surrender profile with respect to duration (ratio of surrenders within each duration range), and is obviously in line with odd-ratio estimations of Table 4: indeed the risk is very high at the beginning and goes decreasing with time. The model has globally 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 accurately the reality in most of cases. As we have previously seen, there is a trade-off between the goodness of fit and the predictive power: in our case good results in terms of prediction are clearly favored since the goal is to make classification predictions. The confusion matrix given in Table 5 gives the number of misclassified policyholders and represents the predictive power of the method. Of course good predictions still appear in the diagonal of the table. 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 15.04% and is quasi-similar to the one gotten with CART method.

It is also interesting to compare the two methods with other usual performance criteria: 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* corresponds to a predicted response equal to 0 and the observed one 1. *correct rejections* corresponds to an observed and a predicted response equal to 0, and finally *false risky policyholder* stands for a predicted response equal to 1 and an ob-

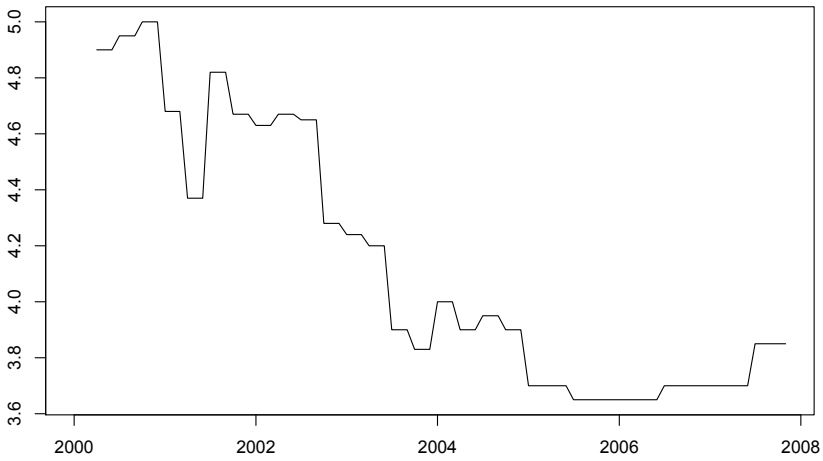
Table 5: Confusion matrix (LR model).

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

Table 6: 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 7: Monthly average credited rate for Mixtos products. This credited rate encompasses the mean guaranteed rate, plus the mean profit benefit rate.



served one to 0. 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 our three CART applications, results are quite similar and errors are well-balanced. The compromise between sensitivity and specificity is better in CART but the number of *misses* is higher. Hence the most prudential model is the LR model (10%) for us.

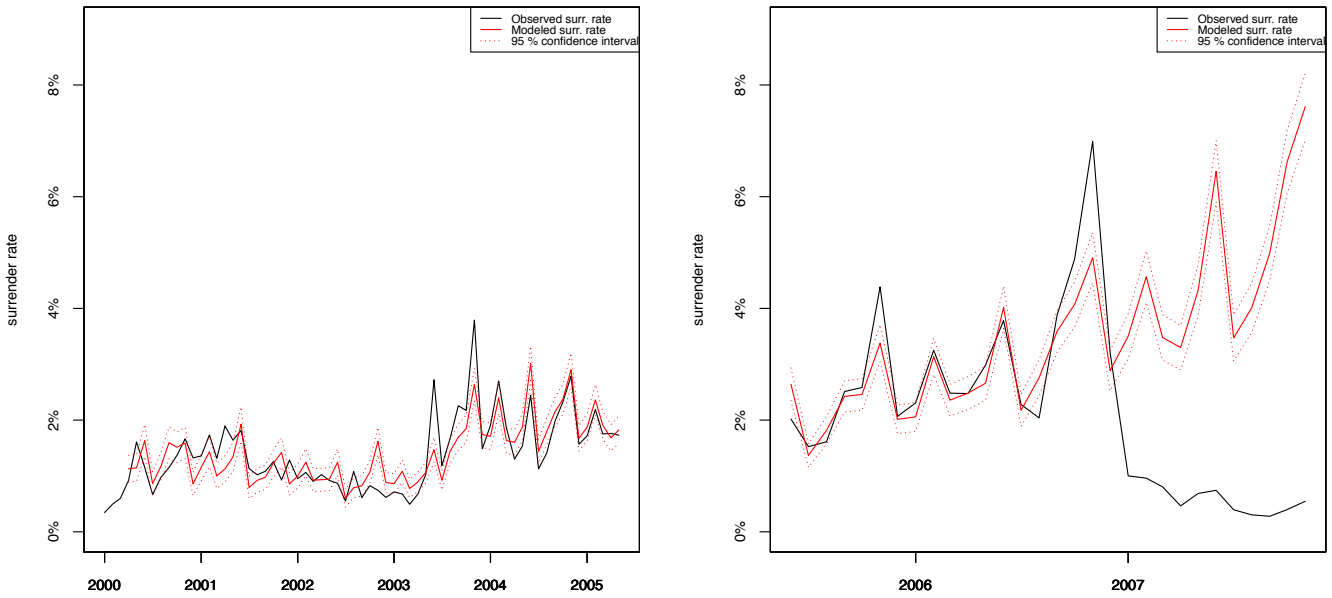
This first static analysis is helpful to understand which policyholders' characteristics and contract' features are relevant, but has a big drawback: we cannot quantify the impact of the economical and financial context on surrender behaviors since we only look at the portfolio at a given date. We could state that in a classical economic and financial regime, behaviors are not driven by economy and hence the static analysis is enough. This is obviously not the case during a crisis where it is extremely hard to anticipate behaviors and thus surrender rates (policyholders' decisions may be correlated, see Milhaud et al. (2010) for a discussion on this). The modeling also becomes much more difficult to handle. To provide a comprehensive model that enables to capture well all effects (endogenous and exogenous) would be tempting but is not in the scope of this paper.

B Further developments: a dynamical analysis

This section is not the heart of the paper, but aims at proving that a static analysis could lead to huge errors in terms of future surrender rate predictions. Practitioners can robustly use segmentation models to get risky profiles, but should be very careful when dealing with time predictions which are strongly dependent on a moving environment. The dynamical analysis better reflects the evolution of economic conditions faced by policyholders and allows us to model their monthly decisions. Therefore the surrender rate is modeled each month on the whole portfolio by aggregation of individual decisions. In this part of the paper we only consider the LR model because we can easily input economic indexes so as to make future predictions.

We have already discussed about the problem of the static analysis (Introduction and Section $III.D$): depending on the period covered and the phenomenon modeled, it could be largely erroneous: if the period covered is longer than the "term" of the phenomenon, the binary response variable would everytime equal 1. By consequence, the model would not be true to life; this is the first argument to run a monthly study. Another one is that we model a dynamical decision: we may think that policyholders is likely to wonder each month if they should keep their contract in force.
However, the dynamical analysis raises a robustness and stability problem because of the additional underlying assumption of 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 her decision at date t (very strong hypothesis which is obviously not reasonable in reality). In the new dataset (whose size is 991 010), policyholders have been duplicated each month while they were present in the portfolio (no surrender and no other reason to leave), and their characteristics were up-dated (duration of the contract, economic indexes...). It gives birth to another bias which does not really alter the results from our experience: characteristics of people with long durations are over-represented in the sample. Anyway, we perform the LR on this new dataset after being sure that the model is built on a representative period

Figure 8: Predictions of the portfolio surrender rate with economic indexes added in explanatory variables. On the left, the predictions on the learning sample and on the right predictions on the validation sample.



(the portfolio is at maturity).

We check the accuracy and the quality of the predictions by comparing the predicted surrender rate and the observed one each month. The final dataset is divided into the following learning and validation samples: the learning sample (whose size is 629357) covers the period from January 2000 to March 2005, and the validation sample covers the period from April 2005 to December 2007 (size equals 361653). To build the model, we add the month of observation (seasonality effects), economical index (unemployment rate) and financial indexes (credited rate, Spanish market index Ibex 35, 1Y and 10Y risk-free interest rates) to the same explanatory variables as in the static study. We neglect the death of policyholders in the portfolio when making future predictions even if they are exposed to this risk, since this event is sufficiently rare (about $2e^{-4}$) in our portfolio.

As a matter of fact, we see on Figure 8 that the observation period has a big influence: the model fits quite well the data in the "learning period" but is a bit far from the reality when predicting the future, especially in 2007. As we can see in Figures 7 and 8, it is rather interesting to note that the average lapsation level increases as the profit benefit is decreasing (2003-2004), which shows a clear relation between credited and lapse rates. The results seem to be acceptable except that it works very bad in extreme situations. During an economic crisis, financial indicators should be the main explanatory variables of surrender decisions. Besides, the assumptions of independence (between policyholders and dates) are not at all realistic when considering extreme events. Here, the beginning of the financial crisis led the surrender rate of endowment products in Spain to drop in 2007, which is not predicted by the model and shows that the economic framework is crucial. Actually we realize that the model does not capture the right effects, especially concerning economy.

This gap between predictions and observed surrender rate is certainly due to the fact that the user has to make an assumption when predicting: what will be the average level of lapsation in the coming months and years 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 average level of lapsation during the learning period will stay the same in the validation period (2005, 2006 and 2007) and then we predict the surrender decisions of policyholders taking into account individual characteristics, economy and seasonality (introduced via the "month" explanatory variable). Indeed a good prediction partially 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? The conclusion is that if future economic conditions are significantly different from the past, the findings of the statistical predictions are often useless, which justifies why statistical predictions for surrender rates are not so popular in actuarial theory and practice.

V Discussion and improvements

The goal of this paper is to give insights on discriminant contract features and policyholder's characteristics regarding the surrender behavior. So what's new?
Our study has brought out some typical risky profiles: **oldest people tend to surrender more than others**, as well as those who have a **periodical premium** ("annual" and "bi-monthly" are the worst cases). Policyholders with **low income** are more likely to surrender their contracts: poor insureds have to pay for fees and regular premiums but they do not have the money for it, whereas rich people may not really pay attention to this. In general the biggest risks are concentrated on the first periods following the termination of a tax constraint: **if the duration of the contract has reached the tax or penalty relief delay, 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 classification predictions can be performed by running either the LR model or the CART model. 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 often not really significant. 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, since we have seen that the results of both models were consistent and complementary. Another improvement in the LR model could be to *re-balance* the dataset (Ruiz-Gazen & Villa (2007)) which is extremely unbalanced in the dynamical analysis: we observe 15571 surrenders among 991010 observations, meaning that surrenders only represent 1.57% of the whole dataset. 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 predictions, but this is not always optimal (Lemmens & Croux (2006)).

Most of professionals know that the duration of the contract is a meaningful factor in explaining the surrender because of tax constraints. At underwriting, we do not have any information on it because the contract is newly acquired. Hence, duration as an input of the model enables us to get reasonable predictions of surrender rates but could not be considered when we want to segment the population of policyholders at underwriting. However this is not really a point since we just have to remove the duration in the modeling to segment policyholders at underwriting process.
Besides, the results of these two segmentation models are true at a fixed date t (when the model is built). To improve this and take nicely into account the duration and the economic context, it could be preferable to use a functional data analysis, or to try some models used in survival analysis like the Cox model family: we could have access to the intensity to surrender at $t + dt$, where dt can be big. The moral hazard, the adverse selection and hidden variables such as the competition on the market (Albrecher et al. (2010)) could be considered as well, but are much more difficult to measure and collect. Finally, there still remains the question on how to model accurately the surrender decisions in all contexts (including a disturbed one) and what kind of model to use to adjust the level of lapsation. Structural effects as well as the conjuncture have both to be considered when modeling surrender rates, which is quite a challenge since they are different by nature. It suggests for instance the use of two separated processes with possible jumps.

Acknowledgement. *We would like to thank an anonymous referee for very useful suggestions. This work is partially funded by the reinsurance company AXA Global Life and the ANR (reference of the French ANR project: ANR-08-BLAN-0314-01).*

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), 'Re-thinking 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, D. (1972), 'Regression models and life tables (with discussion)', *Journal of the Royal Statistical Society: Series B* (34), 187–220.

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.

Loisel, S. & Milhaud, X. (2011), From deterministic to stochastic surrender risk models: impact of correlation crises on economic capital. working paper.

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

Milhaud, X., Gonon, M.-P. & Loisel, S. (2010), 'Les comportements de rachat en assurance vie en régime de croisière et en période de crise', *Risques* (83).

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

Ruiz-Gazen, A. & Villa, N. (2007), 'Storms prediction: logistic regression vs random forest for unbalanced data', *Case Studies in Business, Industry and Government Statistics* **1**(2), 91–101.

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 9.

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.*

Table 7: Complexity parameters

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

II Deeper in CART theory

II.1 Specification of binary rules

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 $2^{M-1} - 1$).*

II.2 What is an impurity function?

Definition 1. *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: $argmax\ g(p_1, p_2, ..., p_J) = \left(\frac{1}{J}, \frac{1}{J}, ..., \frac{1}{J}\right)$,*
2. *the minimum of g is given by the "dirac": $argmin\ g(p_1, p_2, ..., p_J) \in \{e_1, ..., e_J\}$, where e_j is the j^{th} element in the canonical basis of $\mathbb{R}^J$.*

II.3 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 $\frac{p_L p_R}{4}\left[\sum_j |p(j|t_L) - p(j|t_R)|\right]^2$;*

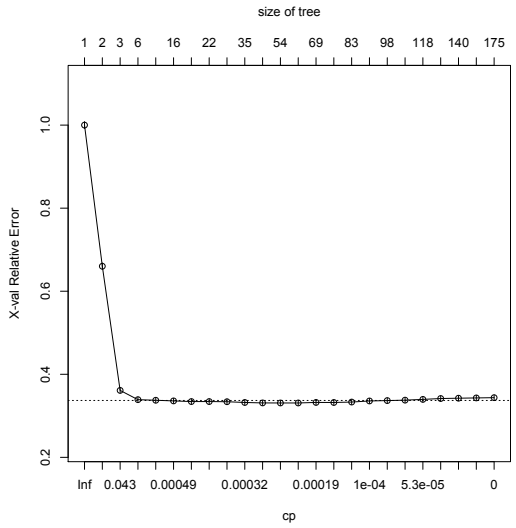


Figure 9: The cross-validated misclassification estimator of the optimal tree in function of the complexity parameter cp (or α). T_{max} contains here 175 leaves and corresponds to $cp = 0$. Notice that there is an initial sharp drop of error followed by a "flat" plateau and a slow rise.

- *in a two-class problem, the Gini index reduces to* $impur(t) = 2p(1|t)p(2|t)$.

II.4 Notes on prediction error

Formally, we can write the expression of the part of observations wrongly classed by the function *class* in function of the prediction error estimate chosen:

– the resubstitution estimate:

$$\hat{\tau}(class) = \frac{1}{N} \sum_{(x_n, j_n) \in \epsilon} \mathbb{1}\{class(x_n, \epsilon) \neq j_n\} \quad (2)$$

– The test sample estimate: used as in (2):

$$\hat{\tau}^{ts}(class) = \frac{1}{N'} \sum_{(x_n, j_n) \in W} \mathbb{1}\{class(x_n, \epsilon) \neq j_n\} \quad (3)$$

– the cross-validation estimate:

$$\hat{\tau}^{cv}(class) = \frac{1}{N} \sum_{k=1}^{K} \sum_{(x_n, j_n) \in \epsilon_k} \mathbb{1}\{class(x_n, \epsilon^k) \neq j_n\} \quad (4)$$

Notice also that:

$$\begin{aligned} \mathbb{E}[\hat{\tau}(class)] &= \mathbb{E}\left[\tfrac{1}{N} \sum_{(x_n, j_n) \in \epsilon} \mathbb{1}\{class(x_n, \epsilon) \neq j_n\} \right] \\ &= \frac{1}{N} \sum_{(x_n, j_n) \in \epsilon} \mathbb{E}[\mathbb{1}\{class(x_n, \epsilon) \neq j_n\}] \\ &= P(class(X, \epsilon) \neq Y) = \tau(class). \end{aligned}$$

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 re-sampling 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.5 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 2. *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:

$$class(x, \epsilon) = \underset{i \in C}{argmin} \sum_{j \in C} \Gamma(i|j)\ p(j|t) \quad (5)$$

From (5), the estimation of the misclassification rate is now

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

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

$$\hat{\tau}(T) = \sum_{t \in \tilde{T}} \hat{\tau}(t) \quad (6)$$

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*

$$\hat{\tau}(T_s) \leq \hat{\tau}(T) \quad (7)$$

Let t_L and t_R be the descendants of node t in tree T_s. From (6) and (7), it turns out that

$$\begin{aligned} \sum_{t \in \tilde{T}_s} \hat{\tau}(t) &\leq \sum_{t \in \tilde{T}} \hat{\tau}(t) \\ \sum_{t \in \tilde{T}} \hat{\tau}(t) - \hat{\tau}(t) + \hat{\tau}(t_L) + \hat{\tau}(t_R) &\leq \sum_{t \in \tilde{T}} \hat{\tau}(t) \\ \hat{\tau}(t_L) + \hat{\tau}(t_R) &\leq \hat{\tau}(t) \quad (8) \end{aligned}$$

II.6 Pruning the tree

The problem of a too complex final tree overfitting 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 (8) we get

$$\hat{\tau}(t) \geq \hat{\tau}(T^t) \tag{9}$$

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 (7), 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:

$$\hat{\tau}_\alpha(T) = \hat{\tau}(T) + \underbrace{\alpha\ Card(\tilde{T})}_{\text{complexity term}}, \text{ where } \alpha > 0. \tag{10}$$

$Card(\tilde{T})$ 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:

$$\tau_\alpha(T(\alpha)) = \min_{T \leq T_{max}} \tau_\alpha(T) \tag{11}$$

For problems of existence and uniqueness 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 (8)). 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 (9), 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

$$\hat{\tau_\alpha}(T_1^t) = \hat{\tau}(T_1^t) + \alpha\ Card(\tilde{T}_1^t) \tag{12}$$

   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,

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

   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

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
$\beta_{duration}$ (categorical)	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
	5 :]30,36]	-3.09368	0.14294	$< 2e-16$	< 0
	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
$\beta_{premium\ frequency}$ (categorical) (in month)	Monthly	0 (reference)			nul
	Bi-monthly	0.92656	0.62071	0.135504	> 0
	Quarterly	-0.03284	0.10270	0.749148	< 0
	Half-yearly	-0.22055	0.16681	0.186128	< 0
	Annual	0.43613	0.10690	4.51e-05	> 0
	Single	-0.28494	0.38155	0.455177	< 0
$\beta_{underwriting\ age}$ (categorical)	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
	4 : [40,50[	-0.26266	0.14077	0.062054	< 0
	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
$\beta_{face\ amount}$ (categorical)	1* :	0 (reference)			nul
	2* :	-5.79014	1.46592	7.82e-05	< 0
	3* :	-7.14918	1.46631	1.08e-06	< 0
$\beta_{risk\ premium}$ (categorical)	1* :	0 (reference)			nul
	2* :	0.36060	0.11719	0.002091	> 0
	3* :	0.26300	0.14041	0.061068	> 0
$\beta_{saving\ premium}$ (categorical)	1* :	0 (reference)			nul
	2* :	0.93642	0.13099	8.74e-13	> 0
	3* :	1.32983	0.14955	$< 2e-16$	> 0
$\beta_{contract\ type}$ (categorical)	PP con PB	0 (reference)			nul
	PP sin PB	-16.79213	114.05786	0.882955	< 0
	PU con PB	-7.48389	1.51757	8.16e-07	< 0
	PU sin PB	-12.43284	1.08499	$< 2e-16$	< 0
β_{gender}	Female	0 (reference)			nul
	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.

on the root node.
But what is the optimal tree in this sequence? (11) tells us that the best pruned tree is the one with the minimum misclassification rate.

B Logistic regression

I 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.

II 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 bernoulli 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:

$$\begin{aligned} logit &= \ln\left(\frac{P[Y=1|X_0=x_0,...,X_k=x_k]}{P[Y=0|X_0=x_0,...,X_k=x_k]}\right) \\ &= \beta_0+\beta_1X_1+...+\beta_kX_k \end{aligned}$$

Finally,

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

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

III The Newton-Raphson algorithm

The condition on maximizing the log-likelihood function (**??**) yields to the following system of (k + 1) equations to solve

$$\left\{\begin{aligned} \frac{\partial l}{\partial\hat{\beta}_0} &= \textstyle\sum_{i=1}^n Y_i-\Phi(\beta_0+\sum_{j=1}^k\beta_kX_k)=0 \\ \frac{\partial l}{\partial\hat{\beta}_j} &= \textstyle\sum_{i=1}^n X_{ij}(Y_i-\Phi(\beta_0+\sum_{j=1}^k\beta_kX_k))=0 \end{aligned}\right.$$

$\forall j=1,...,k$.
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 software, the Newton-Raphson algorithm to solve it is included and uses the following iterative process:

$$\beta^{(i+1)}=\beta^{(i)}-\Big(\frac{\partial^2\ln(L(\beta))}{\partial\beta\partial\beta'}\Big)^{-1}\times\Big(\frac{\partial\ln(L(\beta))}{\partial\beta}\Big) \quad (13)$$

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.

IV Estimating the variance matrix

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

$$\begin{pmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0,\hat{\beta}_1) & \cdots & Cov(\hat{\beta}_0,\hat{\beta}_k) \\ Cov(\hat{\beta}_1,\hat{\beta}_0) & Var(\hat{\beta}_1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ Cov(\hat{\beta}_k,\hat{\beta}_0) & Cov(\hat{\beta}_k,\hat{\beta}_1) & \cdots & Var(\hat{\beta}_k) \end{pmatrix} \quad (14)$$

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

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

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.

V Deviance and tests

V.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

$$\Lambda=2\times\Big(l(\beta)-l(\beta_0)\Big) \quad (15)$$

This statistic follows a χ^2_k, 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: $R^2=1-\dfrac{l(\beta)}{l(\beta_0)}$.
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.

V.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: $\beta_j\neq0$ (H_1).
We use the Wald statistic which follows a χ^2_1 to do this test: $\Lambda=\dfrac{\hat{\beta_j}^2}{Var(\hat{\beta}_j)}$.
Let us choose 5% as confidence threshold, and let us denote by $\chi^2_{95\%}(1)$ 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.