APPLYING ECONOMIC MEASURES TO LAPSE RISK MANAGEMENT WITH MACHINE LEARNING APPROACHES

BY

STÉPHANE LOISEL, PIERRICK PIETTE AND CHENG-HSIEN JASON TSAI

ABSTRACT

Modeling policyholders' lapse behaviors is important to a life insurer, since lapses affect pricing, reserving, profitability, liquidity, risk management, and the solvency of the insurer. In this paper, we apply two machine learning methods to lapse modeling. Then, we evaluate the performance of these two methods along with two popular statistical methods by means of statistical accuracy and profitability measure. Moreover, we adopt an innovative point of view on the lapse prediction problem that comes from churn management. We transform the classification problem into a regression question and then perform optimization, which is new to lapse risk management. We apply the aforementioned four methods to a large real-world insurance dataset. The results show that Extreme Gradient Boosting (XGBoost) and support vector machine outperform logistic regression (LR) and classification and regression tree with respect to statistic accuracy, while LR performs as well as XGBoost in terms of retention gains. This highlights the importance of a proper validation metric when comparing different methods. The optimization after the transformation brings out significant and consistent increases in economic gains. Therefore, the insurer should conduct optimization on its economic objective to achieve optimal lapse management.

KEYWORDS

Lapse, life insurance, machine learning, economic measure.

JEL codes: C52; C53; G22

1. INTRODUCTION

Lapse risk is the most significant risk associated with life insurance when compared with longevity risk, expenses risk, and catastrophe risk (EIOPA, 2011).

Astin Bulletin 51(3), 839-871. doi:10.1017/asb.2021.10

Policyholders of life insurance may choose to surrender their policies at any time for cash values or opt to stop paying premiums and leave policies to become invalid eventually. Lapses have significant impacts on the profitability, or even on the solvency, of a life insurer as many studies demonstrate. They may reduce expected profits (Hwang and Tsai, 2018), let underwriting expenses unrecovered (Tsai *et al*., 2009; Pinquet *et al*., 2011), impair the effectiveness of an insurer's asset–liability management (Kim, 2005a; Eling and Kochanski, 2013), and lead to liquidity threats as experienced by US life insurers in the late 1980s.

When lapse rates vary with interest rates as identified by Dar and Dodds (1989), Kuo *et al*. (2003), Kim (2005b,c), and Cox and Lin (2006), they become even more detrimental to life insurers (Tsai *et al*., 2009). Many papers argue that the option to surrender a policy for the cash value might account for a large proportion of the policy value, for example, Albizzati and Geman (1994), Grosen and Jørgensen (2000), Bacinello (2003), Bauer *et al*. (2006), Gatzert and Schmeiser (2008), and Consiglio and Giovanni (2010). The above reasoning and finding may be the reasons why the fifth Quantitative Impact Study (QIS5), conducted by the European Insurance and Occupational Pensions Authority (EIOPA) in 2011 regarding the implementation of Solvency II reports that lapse risk accounts for about 50% of the life underwriting risks (EIOPA, 2011).

The significance of lapse risk draws attentions of scholars to study what causes policyholders to lapse their policies. We may classify the literature into being macro- or micro-oriented. Macro-oriented papers (e.g., Dar and Dodds, 1989; Kuo *et al*., 2003; Kim, 2005b,c; Cox and Lin, 2006) focus on how lapse rates (the proportion of lapsed policies to the total number of sampled policies within a period of time) are affected by macroeconomic variables such as interest rates, unemployment rates, gross domestic product, and returns in capital markets, as well as by company characteristics like size and organizational form.

Micro-oriented papers secure data from insurers on individual policies to investigate the determinants of the lapse propensities/tendencies. The identified determinants including the characteristics of policyholders and the features of life insurance products/policies (see Renshaw and Haberman (1986); Kagraoka (2005); Cerchiara *et al*. (2009); Milhaud *et al*. (2011); Pinquet *et al*. (2011), and Eling and Kiesenbauer (2014) among others.). Eling and Kochanski (2013) and Campbell *et al*. (2014) provide extensive reviews of the literature on lapses.[1] More recently, Jamal (2017) apply machine learning algorithms to identify groups with homogeneous lapse risks and Aleandri (2017) compare the prediction results of logistic regression (LR) with those of a bagging classification tree.

This paper extends the micro-oriented line of literature in three ways. First, we introduce machine learning algorithms including Extreme Gradient Boosting (XGBoost) and support vector machine (SVM) to lapse behavior modeling. These two algorithms became popular recently because they have good performance in (binary) classifications (Nielsen, 2016; Vafeiadis

et al., 2015; Wainer, 2016). Their underlying theories differ from LR and Classification and Regression Tree (CART) analysis that were used in the literature (e.g., Renshaw and Haberman, 1986; Milhaud *et al.*, 2011). Furthermore, they have not yet been applied to predicting lapses of life insurance.

Second, we adopt economic measures in addition to statistical accuracies when evaluating the performance of different algorithms. What a life insurer cares is not only how well the model predicts the insureds' behaviors but also how the predictions may generate profits. In that regard, Campbell *et al.* (2014) highlight the fact that "*very often actuaries are still more focused on 'fitting a curve' to past experience, with less emphasis on the 'why' and 'so what?' aspects*" (page 14). The adoption of economic measures will reveal the comparative benefits generated by alternative algorithms for the insurer.

Third, we transform the optimization objective from classification accuracy to economic gains to demonstrate the benefit of integrating modeling with profit maximization. Such an integration can motivate a life insurer to improve its customer management through taking preventive measures to reduce lapses and retain more of the contractual service margin specified by International Financial Reporting Standard (IFRS) 17. It also links us to the literature on churn management and its impact on the customer lifetime value (CLV) (e.g., Lemmens and Croux, 2006, Lemmens and Gupta, 2017; Neslin *et al.*, 2006).

The results from applying different algorithms to a large dataset consisting of more than 600,000 life insurance policies show that XGBoost and SVM outperform CART and LR with respect to statistical accuracies. The results further show that XGBoost is the most robust across training samples.

Results are less straightforward when considering the economic measure of retention gains. The retention gains take into account the costs of providing incentives to policyholders to reduce their propensities toward lapses, the benefits of retaining policies, and the costs of false alarm. While a good overall accuracy logically leads to better performance, our assumed economic metrics with different parameter settings favor algorithms with low false alarm rates in many cases. We thus find that LR performs as well as XGBoost in terms of retention gains. This highlights the importance of a proper validation metric when comparing different models.

Last but not least, we find that economic gains can be further enhanced when the optimization is done on a function linked to the gains rather than on statistical accuracies. The retention gains increase about 20% from the models having the best statistical accuracies in the two benchmark cases. The results from sensitivity analyses on the parameters used in calculating retention gains show that the optimization done on the gain function delivers higher retention gains than the best models on statistical accuracies by an average of 51%. The increase is particularly significant when binary models deliver negative retention gains.

The above findings highlight the importance of the right validation metric and objective function. High statistical accuracies do not guarantee high retention gains. The models aiming statistical accuracies may even end up

with negative retention gains. Therefore, actuaries as well as scholars should broaden their interests from new techniques and prediction accuracies to the economic gains of the insurer or even to the overall gains of all stakeholders. The insurer should conduct optimization on its economic objective to achieve optimal lapse management.

The organization of the paper is as follows. Section 2 contains explanations about XGBoost and SVM, followed by brief descriptions on CART and LR. In Section 3, we delineate the two performance metrics to be used. One is the commonly seen accuracy, that is, a statistical validation metric, while the other one is an economic metric considering the expected profits and costs of lapse management. We describe the data obtained from a medium-sized life insurer in Section 4. Section 5 displays the comparison results across the four algorithms in terms of the statistical and economic metrics. We explain how to integrate algorithms with the profit maximization goal at the beginning of Section 6 and then compare the results from optimizing profits with those from optimizing statistical accuracies. In Section 7, we summarize and conclude the paper.

2. Binary classification algorithms

The problem that we want to tackle is detecting whether a policyholder will lapse her/his policy or not, that is, $y_i \in \{0, 1\}$, by the end of the sampling period. Popular predictive models are LR and CART models. More advanced machine learning models that we introduce in this paper are SVM and XGBoost.[2]

2.1. XGBoost

In this subsection, we first describe the gradient boosting technique. Then, we explain stochastic gradient boosting and XGBoost. We describe how we tune the hyperparameters of XGBoost at the end.

2.1.1. *Gradient boosting*

Gradient boosting was introduced by Friedman (2001). Boosting builds models in an iterative way from individuals called "weak learners," and the gradient is used to minimize a loss function. More specifically, the gradient boosting is an ensemble method, that is, multiple weak learners h are combined to become a strong learner F in order to achieve a better predictive performance. In this study, we employ the gradient tree boosting that is a specific case of gradient boosting in which weak learners are decision trees. The following descriptions are summarized from Friedman (2002).

Given a training sample $\{y_i, \boldsymbol{x}_i\}_1^N$ where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$, one would like to find a strong learner $F^*(\boldsymbol{x})$ which minimizes a loss function $\boldsymbol{\Psi}(y, F(\boldsymbol{x}))$:

$$F^*(\boldsymbol{x}) = \arg\min_{F(\boldsymbol{x})} E_{y,\boldsymbol{x}} [\boldsymbol{\Psi}(y, F(\boldsymbol{x}))]. \tag{2.1}$$

The strong learner is an additive expansion of M weak learners $h(\boldsymbol{x}, \{R_{lm}\}_1^L, \overline{y}_{lm})$ that will be a L-terminal node regression tree in our case:

$$F_M(\boldsymbol{x}) = \sum_{m=0}^{M} \beta_m h\left(\boldsymbol{x}, \{R_{lm}\}_1^L, \overline{y}_{lm}\right) = \sum_{m=0}^{M} \sum_{l=1}^{L} \beta_m \overline{y}_{lm} 1(\boldsymbol{x} \in R_{lm}), \quad (2.2)$$

where $\{R_{lm}\}_1^L$ and $\overline{y}_{lm}$ are the L-disjoint regions and the corresponding response values determined by the mth regression tree, respectively, and β_m are named as the expansion coefficients since Equation (2.2) approximates Equation (2.1) by an additive expansion form.

This strong learner is approximately estimated through a stage-wise method that begins with an initial guess $F_0(\boldsymbol{x})$. Then the pseudo-residuals for $m = 1, 2, \ldots, M$ are computed:

$$\tilde{y}_{im} = -\left[\frac{\delta \boldsymbol{\Psi}(y_i, F(\boldsymbol{x}_i))}{\delta F(\boldsymbol{x}_i)}\right]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})}, \quad (2.3)$$

and the optimal value of β_m can be determined by the following equation:

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{N} \boldsymbol{\Psi}\left(y_i, F_{m-1}(\boldsymbol{x}_i) + \beta h\left(\boldsymbol{x}, \{R_{lm}\}_1^L, \overline{y}_{lm}\right)\right), \quad (2.4)$$

given the function h.

The regions $\{R_{lm}\}_1^L$ are obtained by estimating the mth L-terminal node regression tree on the sample $\{\tilde{y}_{im}, \boldsymbol{x}_i\}_1^N$. The product $\beta_m \overline{y}_{lm} = \gamma_{lm}$ is set to optimize the loss function $\boldsymbol{\Psi}$:

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{\boldsymbol{x}_i \in R_{lm}} \boldsymbol{\Psi}(y_i, F_{m-1}(\boldsymbol{x}_i) + \gamma). \quad (2.5)$$

At the final stage, the approximation of the strong learner is updated:

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \nu . \gamma_{lm} 1(\boldsymbol{x} \in R_{lm}), \quad (2.6)$$

where $\nu \in (0, 1]$ is a shrinkage parameter that controls how much information is used from the new tree.

The gradient tree boosting method may be summarized as the following algorithm extracted from Friedman (2002).

	Algorithm 1: Gradient_TreeBoost
1	$F_0(\mathbf{X}) = \arg\min_{\gamma} \Sigma_{i=1}^{N} \boldsymbol{\Psi}(y_i, \gamma)$
2	For $m = 1$ to M do:
3	$\tilde{y}_{im} = -\left[\frac{\partial \boldsymbol{\gamma}(y_i, F(\mathbf{X}_i))}{\partial F(\mathbf{X}_i)}\right]_{F(\mathbf{X})=F_{m-1}(\mathbf{X})}, i = 1,N$
4	$\{R_{lm}\}_1^L = L -$ terminal node$tree(\{\tilde{y}_{im}, \mathbf{X}_i\}_1^N)$
5	$\gamma lm = \arg\min_{\gamma} \sum_{\mathbf{X}_i \in R_{lm}} \boldsymbol{\Psi}(y_i, F_{m-1}(\mathbf{X}_i) + \gamma)$
6	$F_m(\mathbf{X}) = F_{m-1}(\mathbf{X}) + \nu \cdot \gamma lm 1(\mathbf{X} \in Rlm)$
7	endFor

2.1.2. *Extensions to gradient boosting*

Inspired by previous works on statistical learning, many extensions to the gradient (tree) boosting method have been developed. For instance, training each ensemble on a subset of the training set can help improve generalizability of the model. This extension is called the stochastic gradient boosting technique. More specifically, the stochastic gradient boosting technique (Friedman, 2002) is based on the same principle as the bagging technique (Breiman, 1996). It introduces randomness in the observation: given a random permutation π of the integers $\{1, \ldots, N\}$ and $\tilde{N} < N$, the new weak learner tree is estimated on the random subsample $\{\tilde{y}_{\pi(i)m}, \boldsymbol{x}_{\pi(i)}\}_1^{\tilde{N}}$.

Another way to inject randomness that has been popularized by Breiman (2001), that is, random forest, is randomly selecting a subspace of the explanatory variables. More specifically, given a random permutation π^* of integers $\{1, \ldots, n\}$ and $\tilde{n} < n$, the new weak learner tree is estimated on $\{\tilde{y}_{im}, P^*(\boldsymbol{x})_i\}_1^{\tilde{N}}$ in which $P^*(\boldsymbol{x}) = \{x_{\pi^*(1)}, \ldots, x_{\pi^*(\tilde{n})}\}$.

To avoid overfitting, some extensions follow the general idea of the ridge regression (Hoerl and Kennard, 1970) and lasso regression (least absolute shrinkage and selection operator regression; Tibshirani, 1996) and adopt the penalized optimization point of view. Instead of optimizing a loss function $\boldsymbol{\Psi}(y, F(\boldsymbol{x}))$, the problem is modified as the optimization on an "objective" function $\boldsymbol{O}$ that is the sum of a loss function $\boldsymbol{\Psi}$ and a regularization term $\boldsymbol{\Omega}$:

$$\boldsymbol{O}(y, F(\boldsymbol{x})) = \boldsymbol{\Psi}(y, F(\boldsymbol{x})) + \boldsymbol{\Omega}(F). \tag{2.7}$$

2.1.3. *XGBoost*

XGBoost stands for Extreme Gradient Boosting. It may be regarded as a specific implementation of the gradient boosting method in which more accurate approximations are used to find the best tree model. In particular, XGBoost takes account of the second-order gradients, that is, the second partial derivatives of the loss function. This provides more information about the direction of gradients and how to get to the minimum of the loss function. XGBoost further adopts advanced regularization to improve model generalization. XGB can be trained fast and be parallelized/distributed across clusters, which provides additional advantages.

Therefore, the XGBoost system (Chen and Guestrin, 2016) has become the most popular due to its flexibility and computing performances (Nielsen, 2017). The system includes stochastic gradient descent, bagging, random forest, tree pruning, regularization, parallel processing, etc. The broad collection of statistical learning tools allows users to better tackle the bias–variance issue that is one of the main issues in machine learning. Furthermore, the corresponding parameters are made easily adaptable by the packages. XGBoost thus has become the most popular machine learning algorithm in data science challenges such as Kaggle for structured data (Nielsen, 2016).

We list the main parameters that need to be tuned in implementing XGBoost, using the R package's terminology (Chen *et al.*, 2015) and the notation of Friedman (2002), as follows:

(i) *nrounds* is the number of trees to grow: M;
(ii) *eta* is the shrinkage parameter: ν;[3]
(iii) *gamma* is the regularization parameter which is used in $\mathbf{\Omega}$;
(iv) *max_depth* is the number of nodes of a tree: L;
(v) *min_child_weight* is the minimal number of observations in a node and $\min_{l,m} \sum_{i=1}^{N} 1\,(\boldsymbol{x}_i \in R_{lm})$ should be higher than this value;
(vi) *subsample* is the relative size of the random subsample used in the case of a stochastic gradient boosting: $\tilde{N}/N$;
(vii) *colsample_bytree* is the relative size of the random subspace of explanatory variables selected at each new tree: $\tilde{n}/n$.

Since we are interested in a binary classification in this paper, we use the logistic loss function:

$$\mathbf{\Psi}(y, F(\boldsymbol{x})) = \sum_{i=1}^{N} [y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)], \tag{2.8}$$

where $p_i = \frac{1}{1+e^{-F(\boldsymbol{x}_i)}}$ is the probability score, that is, the sigmoid function applied to the output of the model $F(\boldsymbol{x}_i)$.

For the cross-validation metric, we retain the error function:

$$error(y, \hat{y}) = \frac{\sum_{i=1}^{N} 1(y_i \neq \hat{y}_i)}{N}, \tag{2.9}$$

where $\hat{y}_i = \begin{cases} 1 \text{ if } F_M(\boldsymbol{x}_i) > 0.5 \\ 0 \text{ if } F_M(\boldsymbol{x}_i) \leq 0.5 \end{cases}$ and $F_M(\boldsymbol{x}_i)$ is the output of the XGBoost model, that is, a probability.

The tuning method that we adopt consists of two nested cross-validations. We first determine the best *nrounds* through a fivefold cross-validation up to 200 for every possible set of parameters of the retained grid values which are reported in Appendix A. Then we perform grid searches on the parameters with a twofold cross-validation.[4]

2.2. SVM

The theory of SVM was introduced in the 1990s by Boser *et al.* (1992) and Cortes and Vapnik (1995). The underlying idea, as will be described in this section, is substantially different from regression or tree-based models. It has become a popular algorithm for classification problems and for churn prediction in particular (e.g., Zhao *et al.*, 2005; Xia and Jin, 2008). Its predictive power is rather good compared to other classification algorithms

(e.g., Vafeiadis *et al*., 2015; Wainer, 2016). Despite of its good performances in binary classifications, SVM has never been applied to life insurance lapses.

The SVM algorithm can be described by geometrical terms. The main idea is to find a hyperplane that separates the observation space into two homogeneous subspaces that is as far apart from each other as possible. This solution is defined as the maximum-margin hyperplane. To deal with misclassifications, a soft margin (i.e., a penalty determined by the user) is imposed upon the SVM. Another way to deal with classification errors is to project the data to a higher-dimensional space through a kernel function. A more complete geometrical description of SVM can be found in Noble (2006).

In the following, we adopt a formula-based description of the SVM using the notation of Hsu *et al*. (2003). Given a training sample $\{y_i, \boldsymbol{x}_i\}_1^N$ in which $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$, the SVM algorithm is the solution of the following optimization problem:

$$\min_{\omega,b,\xi} \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + C\sum\nolimits_{i=1}^{N} \xi_i, \tag{2.10}$$

with the constraint:

$$y_i\big(\boldsymbol{\omega}^T\phi(\boldsymbol{x}_i) + b\big) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{2.11}$$

The separating hyperplane is determined by the orthogonal vector $\boldsymbol{\omega}$ and constant b. The soft margin penalty cost is denoted as C. The data may be projected to a higher dimension space by the function ϕ, and the underlying kernel function is defined by $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.

In our case, we choose to consider the radial basis function kernel that is the most commonly used in practice and determined by:

$$K\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) = \exp\big(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\big), \tag{2.12}$$

with $\gamma > 0$ being the kernel parameter.[5]

Then we use the "e1071" R package (Meyer *et al*., 2015) to implement the SVM algorithm. To tune the SVM parameters (C, γ), we perform a grid search on a twofold cross-validation and adopt the misclassification error function as the validation metric.[6] The grid of values is reported in Appendix B.

2.3. CART

CART was first introduced by Breiman (1984) and was used by Milhaud *et al*. (2011) to model lapse behavior. The underlying idea is straightforward: defining a class by following a list of decision rules on the explanatory variables. To determine these rules, the data space is iteratively separated by binary split into disjoint subspaces. At each step or node of this top-down construction, the explanatory variable and the dividing point are chosen to minimize a defined

loss function. Since we are interested in a binary classification, we choose the Gini impurity of the node as the loss function.

More specifically, given a node l of N_l observations of response $y_i \in \{0, 1\}$ with $i \in l$, the proportion of observations of response type 1 in the node is defined by $p_l = \frac{1}{N_l} \sum_{i \in l} y_i$. Then, one may use an algorithm to partition the parent node into two nodes l_L and l_R by maximizing

$$I_G(l) - [I_G(l_L) + I_G(l_R)], \tag{2.13}$$

where I_G is the Gini impurity of the node and computed by

$$I_G(l) = N_l p_l (1 - p_l). \tag{2.14}$$

This split estimation can be applied up to obtaining a node for every sample. The obtained tree is called the saturated model. Although fitting the response on the training sample perfectly, the saturated model generally leads to low predictive performance when applied to new samples. Hence, the tree needs to be pruned, that is, the number of final nodes needs to be reduced to increase its predictive power.

Many criteria can be used to prune the tree, for example, the minimum number of samples in a final node. We estimate L, the number of terminal nodes that minimizes the misclassification error as defined by Equation (2.9), by a 10-fold cross-validation methodology. Lastly, we use the "rpart" R package (Therneau *et al.*, 2018) to implement CART.

2.4. Logistic regression (LR)

The LR is a special case of the generalized linear models (Nelder and Wedderburn, 1972) obtained with the Bernoulli distribution. It was employed by Renshaw and Haberman (1986) and Milhaud *et al.* (2011) to identify product and policyholder characteristics relating to lapses. The goal is to model the probability of a binary event such as the lapse probability p_i of the policyholder i. Given a training sample $\{y_i, \boldsymbol{x}_i\}_1^N$ in which $\boldsymbol{x} \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$, the regression model is specified as:

$$\ln \frac{p_i}{1 - p_i} = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}. \tag{2.15}$$

The parameters $(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^n$ can be estimated by the maximum-likelihood method:

$$\mathcal{L} = \prod_{i=1}^{N} \left(\frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta} + \boldsymbol{\beta_0}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta} + \boldsymbol{\beta_0}}} \right)^{y_i} \left(\frac{1}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta} + \boldsymbol{\beta_0}}} \right)^{1 - y_i}. \tag{2.16}$$

When applying the estimated LR model to a classification problem, it does not directly lead to labeled responses but to estimated probabilities. To determine the forecasted class, we choose a threshold $th \in (0, 1)$ so that:

$$\hat{y}_i^* = \begin{cases} 1 \text{ if } \hat{y}_i > th; \\ 0 \text{ if } \hat{y}_i \leq th. \end{cases} \tag{2.17}$$

To obtain the optimal threshold th, we conduct a fivefold cross-validation method on the training set according to the error function.

3. Validation metrics

For each policy, the observed lapse y_i and the forecasted lapse $\hat{y}_i$ are binary variables: $(y_i, \hat{y}_i) \in \{0, 1\}^2$. The four different outputs of a binary classification model are named true positive $(1, 1)$, true negative $(0, 0)$, false positive $(0, 1)$ and false negative $(1, 0)$ respectively, while the number of each case is usually laid out in the so-called confusion matrix. Denote $N(j, k)$ as the numbers of the confusion matrix in which $j \in \{0, 1\}$ stands for the observed lapse indicator and $k \in \{0, 1\}$ for the predicted lapse indicator. Given a set of response variables $\{y_i, \hat{y}_i\}_1^N$, we calculate $N(j, k)$ as:

$$N(j, k) = \sum\nolimits_{i=1}^{N} \mathbf{1}\big(y_i = j, \hat{y}_i = k\big). \tag{3.18}$$

3.1. Statistical metric

Based on the confusion matrix, different metrics can be developed. We first focus on an accuracy metric, the ratio of correctly classified predictions over the total number of predictions:

$$accuracy\big(y, \hat{y}\big) = \frac{N(1, 1) + N(0, 0)}{N} = 1 - error\big(y, \hat{y}\big)\,. \tag{3.19}$$

We are aware that other statistical metrics can be used to compare one binary classification model with another. Nevertheless, our study is to introduce economic metric rather than merely comparing alternative statistical measures. We adopt the above accuracy metric for its simplicity and its widespread use in binary classification applications.

3.2. Economic metric

Although we adopt mathematical algorithms to predict lapses, the risk is an economic issue after all. We thus would like to analyze and compare the classification algorithms by an economic metric. More specifically, we will evaluate

the impacts of different classification results on the expected profits from policies that are also called customer lifetime values (CLVs). In order to do so, we plan to adopt an economic model inspired by Neslin *et al.* (2006) and Gupta *et al.* (2006).

Suppose that policy i stays Θ_i years in the portfolio ($\Theta_i \in \mathbf{N}$). The profitability ratio at time t can be represented by $p_{i,t}$ and the face amount by $F_{i,t}$. The lifetime value for policy i is computed as:

$$CLV_i = \sum_{t=0}^{\Theta_i} \frac{p_{i,t}F_{i,t}}{(1+d_t)^t}, \tag{3.20}$$

where d_t is the discount rate.

Further assuming a deterministic time horizon T(T $\in$ **N**), we define the $(T+1)$-dimensional real vectors $\boldsymbol{p}_i$, $\boldsymbol{F}_i, \boldsymbol{r}_i$, and $\boldsymbol{d}$ for profitability ratios, face amounts, retention probabilities, and discount rates, respectively. Given the four vectors, the CLV is

$$CLV_i\left(\boldsymbol{p}_i, \boldsymbol{F}_i, \boldsymbol{r}_i, \boldsymbol{d}\right) = \sum_{t=0}^{T} \frac{p_{i,t}F_{i,t}r_{i,t}}{(1+d_t)^t}\,. \tag{3.21}$$

The lapse management strategy is modeled by the offer of an incentive $\boldsymbol{\delta}_i \in \mathbb{R}^{T+1}$ to policyholder i who is contacted with a cost c. The incentive is accepted with the probability γ_i, and the acceptance will change the vector of the probabilities of staying in the portfolio from $\boldsymbol{r}_i$ to $\boldsymbol{r}_i^* \in \mathbb{R}^{T+1}$. We further make the following simplifying assumptions:

(i) $\boldsymbol{p}_i$ are the same for all policies and denoted as $\boldsymbol{p}$ hereafter;
(ii) $\boldsymbol{\delta}_i$ are the same for all contacted policies and denoted as $\boldsymbol{\delta}$ hereafter;
(iii) $p_{i,t}$, $F_{i,t}$, and d_t remain constant across time;
(iv) $\boldsymbol{r}_i$ equals to $\boldsymbol{r}_{lapse}$ or $\boldsymbol{r}_{stay}$ in which $\boldsymbol{r}_{stay} = (1, 1, \ldots, 1)$ and $\boldsymbol{r}_{lapse}$ is estimated using the dataset and will be given in Section 5.2;
(v) if $\boldsymbol{r}_i = \boldsymbol{r}_{stay}$, the incentive is accepted with probability γ_i=1 and $\boldsymbol{r}_i^* = \boldsymbol{r}_{stay}$;
(vi) if $\boldsymbol{r}_i = \boldsymbol{r}_{lapse}$, the incentive is accepted with probability γ_i=γ and $\boldsymbol{r}_i^* = \boldsymbol{r}_{stay}$.[7] Policyholders who reject the offers (with probability = 1- γ) will lapse their policies, that is, $\boldsymbol{r}_i^* = \boldsymbol{r}_{lapse}$.

The application of a classification algorithm to the tested samples produces two confusion matrices: one with respect to number of policies, while the other in term of face amount. For the latter matrix, we denote $F(j,k)$ as the coefficients of the matrix with regard to face amount, where j stands for the indicator of the policyholder's lapse in real life, k the indicator by the algorithm's prediction, and $(j,k) \in \{0,1\}^2$. More specifically,

$$F(j,k) = \sum_{i=1}^{N} F_i.\mathbf{1}\left(y_i = j, \hat{y}_i = k\right), \tag{3.22}$$

while N is defined in Equation (3.18).

We define the reference portfolio value (RPV) as the CLV of all policies if no customer relationship management about lapses are carried out to be

$$\begin{aligned} RPV &= CLV\big(\boldsymbol{p}, F(0,0)+F(0,1)\,, \boldsymbol{r}_{stay}, \boldsymbol{d}\big) \\ &\quad + CLV\big(\boldsymbol{p}, F(1,0)+F(1,1)\,, \boldsymbol{r}_{lapse}, \boldsymbol{d}\big)\,. \end{aligned} \tag{3.23}$$

Given a classification algorithm, we compute the lapse managed portfolio value (LMPV) by:

$$\begin{aligned} LMPV(\boldsymbol{\delta}, \gamma, c) &= CLV\big(\boldsymbol{p}, F(0,0)\,, \boldsymbol{r}_{stay}, \boldsymbol{d}\big) + CLV(\boldsymbol{p}, F(1,0) \\ &\quad + (1-\gamma)\,F(1,1)\,, \boldsymbol{r}_{lapse}, \boldsymbol{d}\big) + CLV(\boldsymbol{p}\text{-}\boldsymbol{\delta}, F(0,1) \\ &\quad + \gamma F(1,1)\,, \boldsymbol{r}_{stay}, \boldsymbol{d}\big) - c(N(0,1)+N(1,1))\,. \end{aligned} \tag{3.24}$$

Then, we define the economic metric of the algorithm as the retention gain:

$$RG(\boldsymbol{\delta}, \gamma, c) = LMPV(\boldsymbol{\delta}, \gamma, c) - \ RPV, \tag{3.25}$$

that can be simplified as:

$$\begin{aligned} &\gamma\left[CLV\big(\boldsymbol{p}-\boldsymbol{\delta}, F(1,1)\,, \boldsymbol{r}_{stay}, \boldsymbol{d}\big) - CLV\big(\boldsymbol{p}, F(1,1)\,, \boldsymbol{r}_{lapse}, \boldsymbol{d}\big)\right] \\ &\quad - CLV\big(\boldsymbol{\delta}, F(0,1)\,, \boldsymbol{r}_{stay}, \boldsymbol{d}\big) - c(N(0,1)+N(1,1))\,. \end{aligned} \tag{3.26}$$

4. Data

Our data come from a medium-sized life insurance company in Taiwan that had total assets over 15 billion US dollars at the end of 2013. The data contain 629,331 life insurance policies sold during the period from 1998 to 2013. The data-providing insurer tracked changes in the statuses of policies including death and lapse. The last tracking date is 8/31/2013, and 243,152 policies out of all samples were lapsed.[8] The prediction is to detect whether a policyholder will lapse her/his policy within the limited time window of data starting from the inception of individual policies to the end of policy lives or 2013 whichever is earlier.

We specify several variables based on the literature and the data provided by the insurer as input to the algorithms of Section 2. First, we are able to identify the age, gender, and occupation of an insured at the time when the policy was issued. Female is designated as 1, while male as 0 for the dummy variable Gender. Then we designate the dummy variable Occupation as 1 for the occupations that the insurers in Taiwan would undertake extra screening/underwriting. The data also record whether the insured is required to have a physical examination when purchasing life insurance and how many non-life policies (health and long-term care) an insured has.

The data contain the inception date and face amount of each policy. There are three types of policies. The most popular type is conventional

policies like term life, whole life, and endowment. Investment-linked and interest-adjustable types of products appeared in 2000s. We are also able to identify whether a policy is a single-premium one or not. There are three cases with regard to participation: mandatory participation, non-participation, and participation. It was not until 2004 that insurers were allowed to sell non-participating policies. The policies sold before the end of 2003 are thus designated as Mandatory Participating. Starting from 2004, policies may be classified into participating and non-participating. Most policies sold in Taiwan are dominated in New Taiwan Dollar (NTD); there are some policies dominated in other currencies.

We further set up two nominal variables. First, we categorize distribution channels as Tied Agents (denoted by TA), Direct Marketing (DM), and Banks (BK)[9]. Second, premium paying methods are classified into three ways: collected by the personnel of the insurer (denoted as Insurer), automatic transfers from banks or payments by credit cards (B&C),[10] and going to the post office or convenient stores in person (P&C).

Table 1 presents the descriptive statistics of the above explanatory variables. About 20% of the insureds work in riskier occupations that call for extra underwriting. Most insureds (over 96%) were not required to go through physical examination in purchasing life insurance. The most popular way of paying premiums is through automatic/recurring transfers from bank accounts or credit cards (71%). Since post offices and convenient stores providing money transferring services are conveniently around, about 10% of our samples have premiums paid in places like these. And 46.6% of samples are mandatory-participating policies, while 37.2% are non-participating ones.

Many insureds are associated with multiple non-life policies, so that the average number of non-life policies a person are listed as the insured is 1.2. There is a person who is listed as the insured for 33 non-life policies.[11] The face amount of the sampled policies has an average of 17,165 US dollars[12] with big variations: the largest policy reaches 2 million dollars, the smallest one is only 333 dollars,[13] and the standard deviation is about 28,000 dollars.

5. Results with respect to statistical and economic metrics

We are interested in seeing the predictive performance of different algorithms under alternative metrics and thus conduct out-of-sample tests using the following procedure. First, we randomly split the dataset D into 10 subsamples $\{D_1, \ldots, D_{10}\}$ of equal size and then train an algorithm on D_k , k $\in\{1,\ldots,10\}$.[14] The estimated model is subsequently applied to other subsamples $(D - D_k)$ to obtain forecasts $\hat{y}$ of lapses. In the last step, we compare these predictions with the observed lapses y by the validation metric $\rho(y, \hat{y})$ to measure the predictive performance of the algorithm. Repeating this on k $\in \{1, \ldots,10\}$, we obtain 10 observations on out-of-sample prediction performance. This procedure enables

TABLE 1

DESCRIPTIVE STATISTICS OF EXPLANATORY VARIABLES.

Nominal variables				
Gender	Female 48%	Male 52%		
Occupation	Tier one 80.5%	Requiring extra underwriting 19.5%		
Physical examination	Exempted 96.4%	Required 3.6%		
Distribution channel	TA 93.9%	BK 3.4%	DM 2.4%	Others 0.3%
Premium payment	Single premium 3.1%	Non-single premium 96.9%		
Premium paying method	Insurer 18.8%	B&C 70.8%	P&C 10.4%	
Participation	Non-participating 37.2%	Participating 16.2%	Mandatory Participating 46.6%	
Product type	Interest-adjustable 1.7%	Investment-linked 1.2%	Conventional 97.1%	
Currency domination	NTD 88.1%	Others 11.9%		

Metric variables	Mean	Medium	Standard deviation	Minimum	Maximum
Age	28.3	27	16.8	0	80
# of non-life policies	1.2	0	2	0	33
Inception date	06/06/2005	21/04/2005	4.8(years)	01/01/1998	31/07/2013
Face amounts (in USD)	17,165	10,000	28,050	333	2,000,000

us to make sure that every observation is used, at some point of an algorithm, as both training and testing samples.

The underlying idea of the above procedure is similar to the k-fold cross-validation technique in which the training subsample is composed of $D - D_k$

TABLE 2

CROSS-VALIDATED STATISTIC ACCURACIES.

	LR	CART	SVM	XGB
Mean accuracy	77.07%	77.15%	77.82%	78.88%
Standard deviation	0.03%	0.10%	0.08%	0.03%

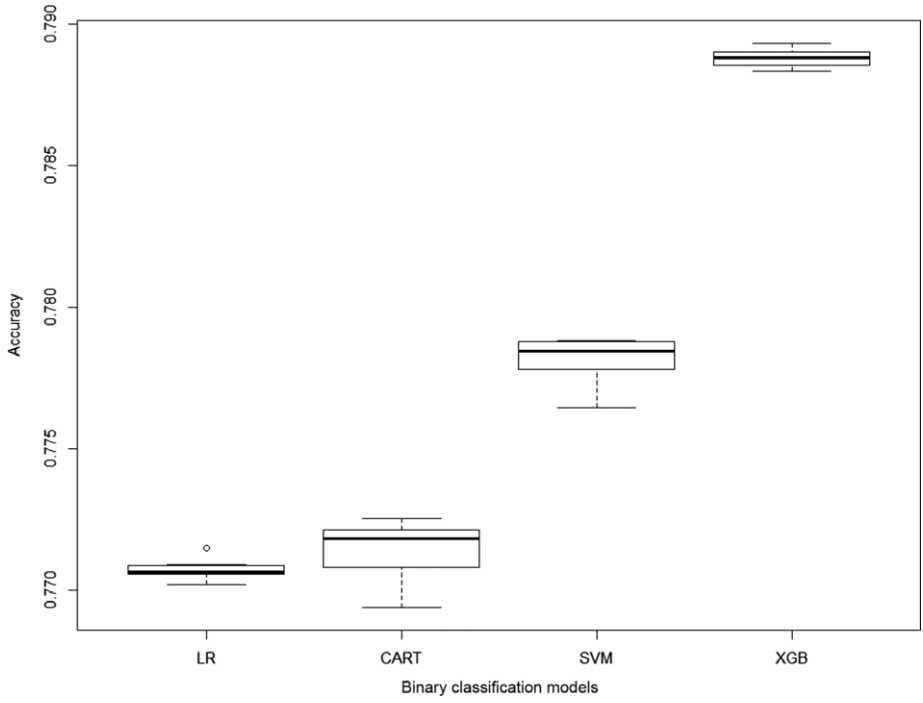

FIGURE 1: Box plot of statistic accuracies estimated on the test sets.

and the testing subsample is set to D_k. We use the k-fold cross-validation to tune parameters in training some of the algorithms.

5.1. Results with respect to the statistical metric

The mean accuracy computed using the above cross-validation procedure is displayed in Table 2 and Figure 1 for each binary classification algorithm. As expected, the more sophisticated the model is, the more accurate the predictions will be. XGBoost ranks number one, followed by SVM, CART, and LR. XGBoost surpasses LR by 1.81% on average, which represents an improvement of 10,236 correctly classified policies. Moreover, the smallest standard deviation of accuracy of XGBoost, 0.03%, indicates that XGBoost is less prone

TABLE 3

(A) AVERAGE CONFUSION MATRIX OF XGB. (B) AVERAGE CONFUSION MATRIX OF SVM. (C) AVERAGE CONFUSION MATRIX OF CART. (D) AVERAGE CONFUSION MATRIX OF LR.

		Predicted: Stay	Predicted: Lapse
Actual	Stay	309,111 54.6%	38,450 6.8%
	Lapse	81,177 14.3%	137,660 24.3%

		Predicted: Stay	Predicted: Lapse
Actual	Stay	310,258 54.8%	37,303 6.6%
	Lapse	88,339 15.6%	130,498 23.0%

		Predicted: Stay	Predicted: Lapse
Actual	Stay	296,320 52.3%	51,241 9.0%
	Lapse	78,209 13.8%	140,628 24.8%

		Predicted: Stay	Predicted: Lapse
Actual	Stay	315,184 55.6%	32,377 5.7%
	Lapse	97,486 17.2%	121,351 21.4%

to the sample splits done for conducting out-of-sample tests. This is visible in the box plot of Figure 1.

Looking at the entire confusion matrices in Tables 3a-d, we find that CART predicts the most lapses (191,869 = 51,241 + 140,628) from which it identifies the most lapses correctly but also signals the most false alarms. LR predicts the most stays (415,670 = 315,184 + 97,486) in which it identifies the most stays correctly while produces the most false security cases. On the other hand, XGBoost and SVM are more robust: they do not suffer from a high rate of false alarm nor a high rate of false securities. XGBoost outperforms SVM in capturing the to-be-lapsed policies.

5.2. Results with respect to the economic metric

To evaluate the algorithms by the economic metric, we first need to specify the parameters of the cash flows model. Since no data are available for us to estimate these parameters, we have to make assumptions. We had conducted

sensitivity analyses to be shown in section 5.3. The comparison results remain the same in general.

The time horizon T is set to be 12 years according to the length of the sampling period. We estimate the retention probability vector $\boldsymbol{r}_{lapse}$ from the dataset and obtain

Year t	0	1	2	3	4	5	6	7	8	9	10	11	12
Retention probability	0.96	0.87	0.67	0.37	0.27	0.21	0.15	0.12	0.1	0.08	0.06	0.05	0.04

Other parameters are set as follows:

- the profitability ratio $p = 0.5\%$;
- the discount rate $d = 2\%$;
- the cost to contact a policyholder $c = 10$ USD.

We propose two different incentive strategies: a strong one and a moderate one. The incentive vectors as percentage discounts on premiums are defined as below:

Year	0	1	2	3	4	5	6	7	8	9	10	11	12
Strong Incentive	0%	0%	0.030%	0.030%	0.060%	0.060%	0.090%	0.090%	0.120%	0.120%	0.150%	0.150%	0.180%
Moderate Incentive	0%	0%	0.015%	0.015%	0.030%	0.030%	0.045%	0.045%	0.060%	0.060%	0.075%	0.075%	0.090%

We further assume that the probabilities for policyholders to accept incentives and maintain the validity of policies are $\gamma_1 = 20\%$ and $\gamma_2 = 10\%$, respectively.[15]

The results from comparing different classification algorithms by the economic metric with the strong incentive strategy are displayed in Table 4 and Figure 2. There is no clear winner. CART underperforms significantly, while XGBoost, SVM, and LR generate similar retention gains.

Notice that the differences across algorithms are wider in terms of the economic metric than the statistical metric. The accuracies of alternative algorithms are between 77.07% and 78.88%, which means an improvement ratio of 2.3%. The retention gains, on the other hand, range from 2.68 and 5.33 million USD, indicating an enhancement of 99%. Therefore, choosing a good algorithm is more important in terms of economic reality (dollar amount) than by statistical accuracy.

It appears that CART produces the lowest retention gain: $2,680,012. This is mostly because CART has the highest false alarm rate (cf. Table)) which means offering the incentive to many policyholders who have no intention to lapse their policies. Furthermore, CART leads to the highest contacting cost since it predicts the highest lapses. The profits are thus reduced. In contrast, LR has the lowest false alarm rate and predicts the lowest lapses. It thus generates the highest retention gain: $5,327,911.

TABLE 4

CROSS-VALIDATED RETENTION GAINS WITH THE STRONG STRATEGY.

	LR	CART	SVM	XGB
Mean retention gain	5,327,911	2,680,012	5,028,737	5,243,913
Standard deviation	149,000	209,220	139,102	115,415

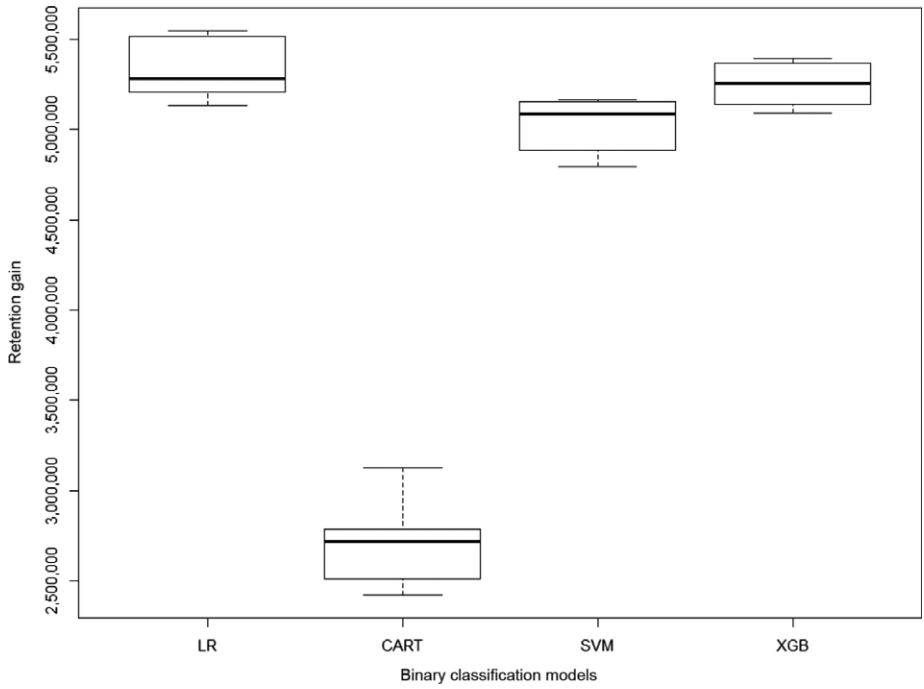

FIGURE 2: Box plot of retention gains with the strong strategy estimated on the test sets.

Then, we look at algorithms' performances when the incentive strategy is moderate and leads to lower acceptance probabilities. The results are displayed in Table 5 and Figure 3. We first notice that LR, SVM, and XGBoost significantly outperform CART again. Second, we observe that the improvement ratio of the best algorithm over the worst is smaller but remains to be significant (56%). Third, retention gains are significantly lower with the moderate incentive strategy. For instance, XGB achieves a gain of 5.2 million dollars with the strong incentive strategy, but the gain reduces to 3.3 million dollars when incentives offered to policyholders are moderate. Under our assumptions, the company should adopt the strong incentive strategy to optimize its gains. In practice, one would need more comprehensive sensitivity studies on the incentives to be offered and the corresponding acceptance probabilities to fully optimize the lapse management.

TABLE 5

CROSS-VALIDATED RETENTION GAINS WITH THE MODERATE STRATEGY.

	LR	CART	SVM	XGB
Mean marketing gain	3,178,087	2,085,599	3,113,900	3,261,029
Standard deviation	60,255	85,184	54,169	45,928

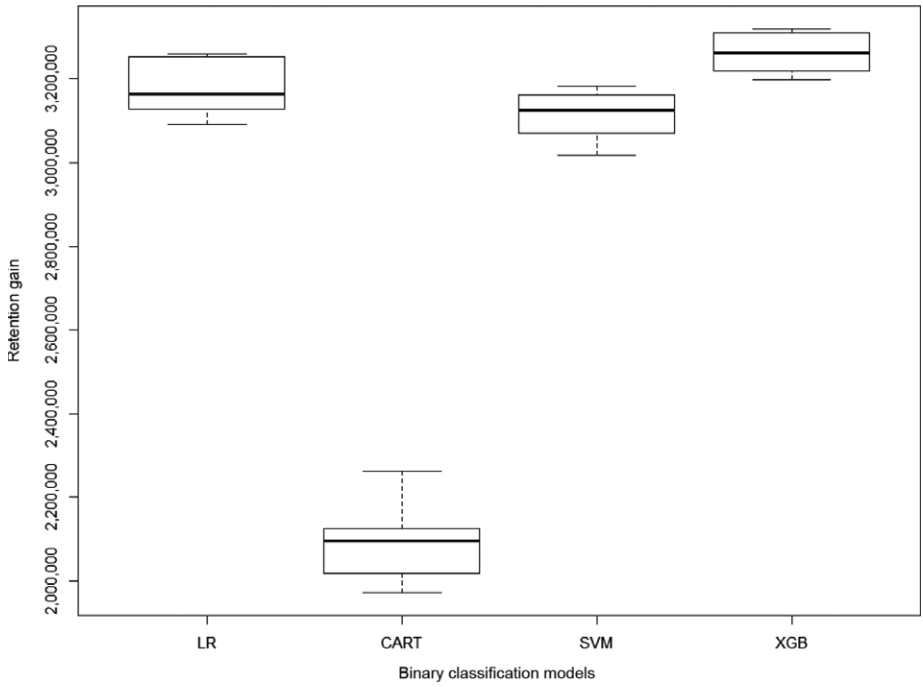

FIGURE 3: Box plot of retention gains with the moderate strategy estimated on the test sets.

In summary, we first observe that XGB and SVM outperform CART and LR by accuracy metrics. Then, we see that LR levels up to XGB and SVM when we switch to economic metrics. This is because our economic metric settings tend to favor a low false alarm rate upon which LR has the best result. The performance rank changes highlight the importance of choosing a proper metric. A "naïve" actuary who limits himself/herself on a statistical point of view will keep pursuing advanced algorithms like XGB; a more "pragmatic" risk manager would have a broader view than merely statistical accuracy and leads to more robust results.

5.3. Sensitivity analyses

We are aware that the above results may be sensitive to the assumption that we make. We thus conduct sensitivity analyses on all parameters, except for the

retention probability vector since it is estimated on empirical data, and present the results in Table 6.

The sensitivity analyses on the profitability ratio (p) indicate that a lower p will increase the importance of low false alarm rate and thus enlarge the outperformance of LR over XGB (becoming -$942,888 vs. -$1,958,561 in Table 6 from the case of $5,327,911 vs. $5,243,913 in Table 4). This is because the incentives offered to the policyholders that have no intentions to lapse their policies are wasted. When expected profits are low, these "wasted" incentives would reduce retention gains by a significant portion or even result in negative gains. The algorithm that generates a low false alarm rate would have good performance as a result. This reasoning also implies that the significance of the false alarm rate would be reduced by p, and we do see from Table 6 that XGB and SVM outperform LR when p is raised to 1% from 0.5% ($23,250,096 and $22,111,997 vs. $21,004,909).

Stronger incentive strategies will also increase the importance of low false alarm rate. When an insurer offers larger incentives to insureds for not lapsing their policies, offering these incentives to "wrong" insureds (i.e., those who have no intentions to lapse their policies in the first place) increases costs but generates no benefits. We therefore see that LR outperforms XGB (-$546,631 vs. -$2,090,205) when an aggressive incentive strategy (starting from 0.05% in the second year and then increase 0.05% every 2 years up to year 12) is adapted; LR would underperform XGB if a weak incentive strategy (starting from 0.01%) is used ($11,202,453 vs. $12,578,031).

A higher probability of accepting the offer to keep the policy implies a more effective incentive without raising the cost. This reduces the importance of a low false alarm and vice versa. We thus observe from Table 6 that XGB outperforms LR when the probabilities of accepting the incentives for a would-lapse policyholder (γ) increase to 25% and underperform LR when γ decreases to 15%, given the strong incentive strategies.

A higher cost of contacting the policyholder (c) would increase the importance of a low false alarm rate. We therefore see that LR outperforms/underperforms XGB when c = $50/$1. The discount rate d merely affects the present value of a future cash flow. A larger or smaller value therefore does not alter the ranking of alternative algorithms as we observe from Table 6.

6. Optimization on profitability instead of classification

It is obvious that insurers would not seek to optimize the classification accuracy but focus on economic gains that result from the classification algorithms when forming a lapse management strategy. When our aim is to maximize the profitability of the lapse management strategy, binary classifications might be unsuitable since they are not designed to meet such a need. Ascarza *et al.* (2018) emphasize the difference between the at-risk population (e.g., customers with high churn probabilities) and the targeted population (e.g., customers that the

TABLE 6

RETENTION GAINS WITH ALTERNATIVE PARAMETERS.

Profitability (p)	Incentive (second year)	Probability (gamma)	Cost (c)	Discount rate (d)	LR	CART	SVM	XGB
0.30%	0.03%	20%	10	2.0%	−942,888	−4,746,915	−1,804,567	−1,958,561
1.00%	0.03%	20%	10	2.0%	21,004,909	21,247,328	22,111,997	23,250,096
0.50%	0.05%	20%	10	2.0%	−546,631	−6,632,394	−1,888,940	−2,090,205
0.50%	0.01%	20%	10	2.0%	11,202,453	11,992,418	11,946,414	12,578,031
0.50%	0.03%	25%	10	2.0%	8,331,308	6,237,135	8,301,545	8,693,534
0.50%	0.03%	15%	10	2.0%	2,324,515	−877,111	1,755,929	1,794,291
0.50%	0.03%	20%	50	2.0%	−821,181	−4,994,768	−1,683,291	−1,800,463
0.50%	0.03%	20%	1	2.0%	6,711,457	4,406,837	6,538,943	6,828,897
0.50%	0.03%	20%	10	5.0%	4,268,910	2,177,299	4,039,478	4,214,841
0.50%	0.03%	20%	10	1.0%	5,741,453	2,864,649	5,412,485	5,642,858

company should focus her retention campaign on in order to optimize her profits) from an economic point of view. Along this line of churn literature, Lemmens and Gupta (2020) modify the usual loss function into a profit-based function to optimize economic gains. They obtain a significant increase in the expected profit of a retention campaign. Learning from the churn literature, we transform the above classification problem into a regression question in this section.

6.1. Methodology

Let the new response variable $z_i^{R_j}$ represents the retention gain or loss that results from proposing the incentive $j \in \{1, 2\}$ (cf. Section 5.2) to policyholder i. More specifically, we define $z_i^{R_j}$ as:

$$z_i^{R_j} = \begin{cases} -CLV\left(\boldsymbol{\delta}_j, F_i, \boldsymbol{r}_{stay}, \boldsymbol{i}\right) - c & \text{if } y_i = 0, \\ \gamma_j . \left[CLV\left(\boldsymbol{p}\text{-}\boldsymbol{\delta}_j, F_i, \boldsymbol{r}_{stay}, \boldsymbol{i}\right) - CLV\left(\boldsymbol{p}, F_i, \boldsymbol{r}_{lapse}, \boldsymbol{i}\right)\right] - c & \text{if } y_i = 1; \end{cases} \tag{6.27}$$

Then, we may apply the XGBoost algorithm to $\{z_i^{R_j}, \boldsymbol{x}_i\}_1^N$ and use the mean squared error as the loss function:

$$\boldsymbol{\Psi}\left(z^{R_j}, \widehat{z^{R_j}}\right) = \frac{1}{N} \sum_{i=1}^{N} \left[z_i^{R_j} \text{-} \widehat{z^{R_j}}_i\right]^2, \tag{6.28}$$

and as the metric for cross-validation.

In the last step, $\hat{y}_i$ is forecasted if the estimated gain is positive:

$$\hat{y}_i = \begin{cases} 1 \text{ if } \widehat{z^{R_j}}_i > 0 \\ 0 \text{ if } \widehat{z^{R_j}}_i \leq 0 \end{cases}, \tag{6.29}$$

By this way, we apply the same metrics described in previous sections. Here, $\hat{y}_i$ is better to be understood as whether offering an incentive to the policyholder i would be profitable to the insurer or not rather than the forecast on whether a policyholder would lapse or not.

The two new classifications are denoted as XGB_R1 and XGB_R2, respectively, for applying XGBoost to z^{R_1} and z^{R_2}. The tuning method that we use to estimate the parameters is described in Appendix C.

6.2. Results

Table 7 and Figure 4 display the prediction accuracies. Table 7 shows that XGB_R1 and XGB_R2 produce relatively low mean accuracy of 76.7% and 75.7%, respectively. They are the worst models in term of accuracy. These seemingly unsatisfied results are understandable, since neither XGB_R1 nor

TABLE 7

CROSS-VALIDATED ACCURACY.

	LR	CART	SVM	XGB	XGB_R1	XGB_R2
Mean accuracy	77.07%	77.15%	77.82%	78.88%	76.67%	75.71%
Standard deviation	0.03%	0.10%	0.08%	0.03%	0.07%	0.06%

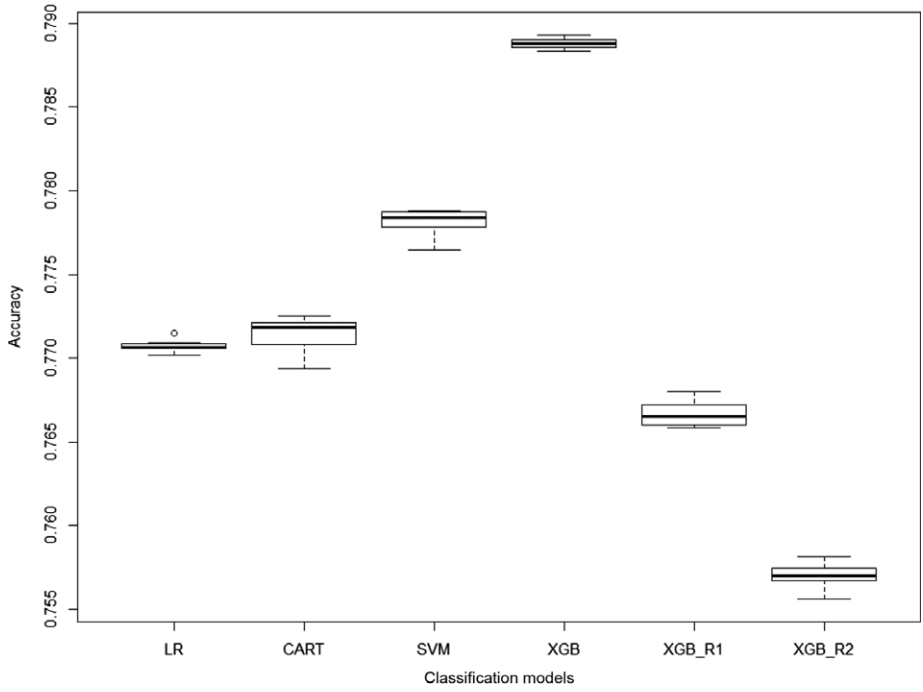

FIGURE 4: Box plot of cross-validated accuracy estimated on the test sets.

XGB_R2 are designed to predict whether a policy would be lapsed or not. What they aim for are economic gains.

The numbers in Tables 8a and 8b tell us more about why XGB_R1 and XGB_R2 do not perform well in statistical accuracies. They result in the smallest correct identifications on lapses (104,889 and 99,432, respectively) and produce the most false sense of security (113,948 and 119,405). However, we will see in the following that XGB_R1 and XGB_R2 stand out when we switch focus to retention gains.

Table 9 and Figure 5 show that XGB_R1 generates a significantly larger average retention gain with the strong incentive strategy ($6,586,357) than other algorithms as well as a significantly lower standard deviation ($53,460). The increase in retention gain is 24% (1.3 million USD) higher than that generated by LR (the second-best algorithm) and 146% (3.9 million USD) better than that produced by CART. Looking back to Table 8a, we see that XGB_R1

TABLE 8

(A) AVERAGE CONFUSION MATRIX OF XGB_R1. (B) AVERAGE CONFUSION MATRIX OF XGB_R2.

		Predicted	
		Stay	Lapse
Actual	Stay	329,357	18,204
	Lapse	113,948	104,889

		Predicted	
		Stay	Lapse
Actual	Stay	329,413	18,149
	Lapse	119,405	99,432

TABLE 9

CROSS-VALIDATED RETENTION GAINS WITH THE STRONG STRATEGY.

	LR	CART	SVM	XGB	XGB_R1
Mean retention gain	5,327,911	2,680,012	5,028,737	5,243,913	6,586,357
Standard deviation	149,000	209,220	139,102	115,415	53,460

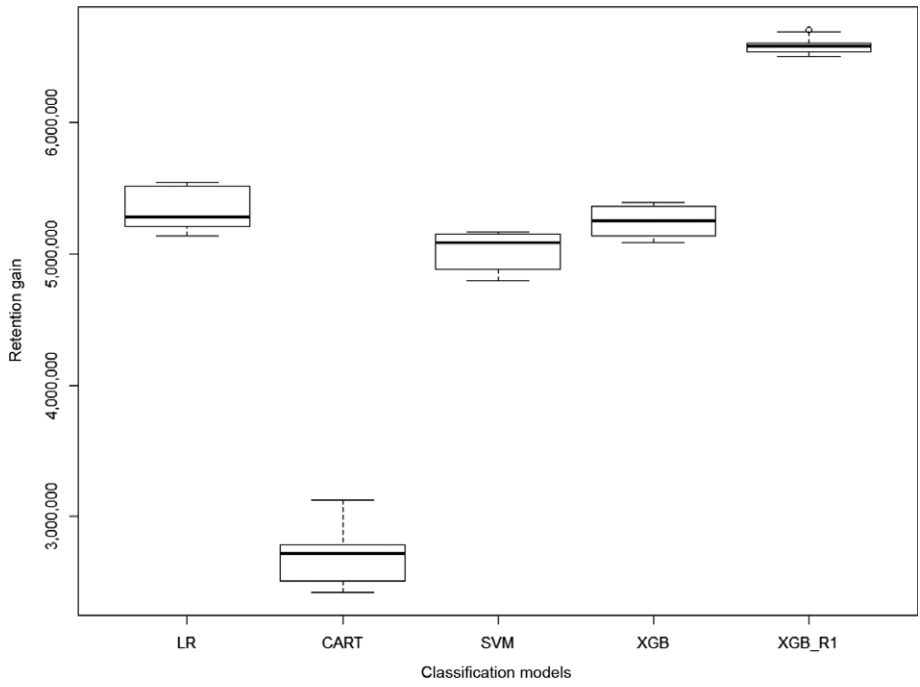

FIGURE 5: Box plot of retention gains the strong strategy estimated on the test sets.

TABLE 10

CROSS-VALIDATED RETENTION GAINS THE MODERATE STRATEGY.

	LR	CART	SVM	XGB	XGB_R2
Mean marketing gain	3,178,087	2,085,599	3,113,900	3,261,029	3,852,782
Standard deviation	60,255	85,184	54,169	45,928	39,163

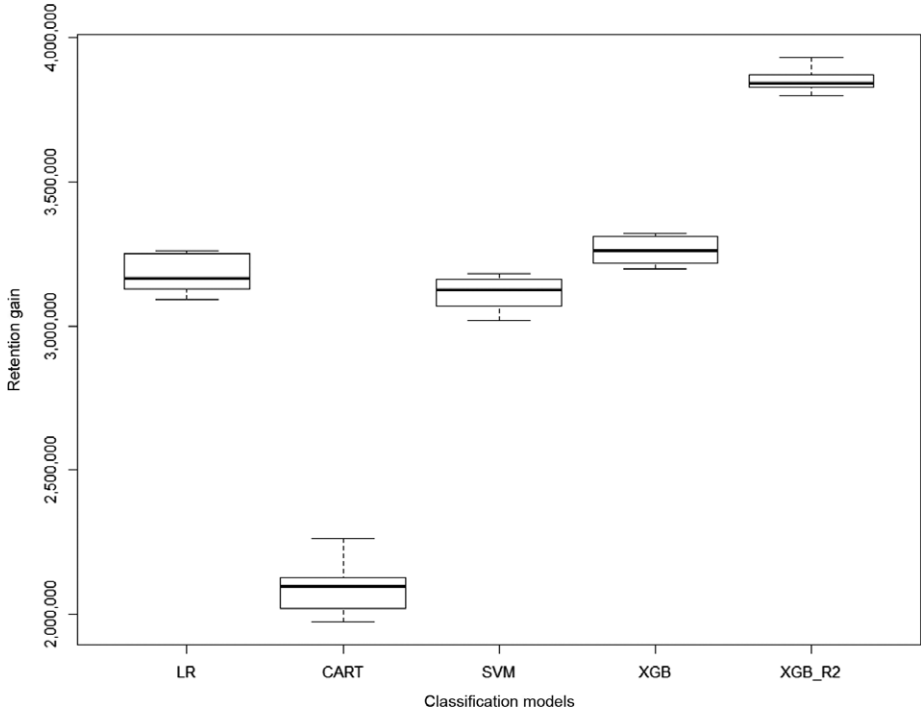

FIGURE 6: Box plot of retention gains the moderate strategy estimated on the test sets.

reduces the number of false alarms (18,204) in optimizing the retention gain even if this also reduces the correct detection (104,889). The good results of XGB_R1 in achieving retention gain demonstrate the benefit of integrating the algorithm with the goal to be achieved. The objective function for XGB_R1 to minimize, Equation (6.28), is about predicting retention gains. XGB_R1 therefore would naturally perform the best when compared with other algorithms optimizing other objectives (such as classification accuracies).

We expect that the benefit of integrating the algorithm with the goal is robust across incentive strategies. This is confirmed by the results in Table 10 and Figure 6. XGB_R2 generates retention gain of 3.9 million dollars that is nearly 600 thousand dollars more than that achieved by the second place XGB. The increase in retention gains is 18%. The increase with respect to the CART reaches 85%.

TABLE 11

IMPROVEMENTS OF XGB_R OVER THE BEST OF BINARY MODELS.

Profitability (p)	Incentive (2nd year)	Probability (gamma)	Cost (c)	Discount rate (d)	Best of binary models	XGB_R	Increase	Percentage of increase
0.30%	0.03%	20%	10	2.0%	-942,888	1,642,720	2,585,608	157%
1.00%	0.03%	20%	10	2.0%	23,250,096	23,634,878	384,782	2%
0.50%	0.05%	20%	10	2.0%	-546,631	3,213,095	3,759,725	117%
0.50%	0.01%	20%	10	2.0%	12,578,031	13,272,649	694,618	5%
0.50%	0.03%	25%	10	2.0%	8,693,534	9,477,878	784,344	8%
0.50%	0.03%	15%	10	2.0%	2,324,515	3,995,017	1,670,502	42%
0.50%	0.03%	20%	50	2.0%	-821,181	3,424,200	4,245,380	124%
0.50%	0.03%	20%	1	2.0%	6,828,897	7,782,795	953,898	12%
0.50%	0.03%	20%	10	5.0%	4,268,910	5,311,218	1,042,308	20%
0.50%	0.03%	20%	10	1.0%	5,741,453	7,104,433	1,362,981	19%

We conduct more sensitivity analyses on the parameters used in calculating retention gains and present the results in Table 11.

We see from Table 11 that XGB_R delivers higher retention gains than the best of the binary models (LR or XGB) across alternative parameter settings. The increase in retention gains ranges from 2% to 157% with an average of 51%. The increase is particularly significant when binary models deliver negative retention gains. This highlights the importance of the right objective function. High statistical accuracies do not guarantee high retention gains. The models aiming statistical accuracies may end up with negative retention gains.

The results in this section demonstrate the benefit of having a right objective. If senior managers of an insurer are able to specify an objective consistent with the value and/or mission of the company (e.g., maximizing retention gain), the staff can then apply an advanced or robust algorithm like XBG or LR to such an objective to achieve the optimum. The enhanced gain relative to the case having no specific objective other than classification accuracy can be substantial.

7. Conclusions

Lapse risk is the most significant risk associated with life insurance. Lapses may cause losses, reduce expected profits, lead to stringent liquidity, result in mispricing, impair the risk management, or even pose solvency threats. Employing a good algorithm to model policyholder lapse behavior is therefore valuable.

In this study, we adopt innovative viewpoints on lapse management in addition to introducing machine learning algorithms to lapse prediction. First, applying XGBoost and SVM to predicting whether a policyholder will lapse her/his policy is new to the literature. Second, we adopt not only a statistical metric in evaluating algorithms' prediction performance but also an economic metric based on CLV and retention gains.

The goal of classification accuracy has no direct link to the insurer's costs and profits. It thus might lead to a biased strategy (Powers, 2011). Following the churn literature, we define a specific validation metric based on the economic gains. This constitutes our third contribution: we are the first to set up a profit-based loss function so that we may directly optimize the economic gains. More specifically, we change the usual statistical idea of classification to a gain regression in which profits are to be maximized.

We are aware that the calculations on economic gains in this paper are based on subjective assumptions. This research constraint can be released as soon as an insurer caring about economic gains gathers data to estimate relevant parameters. More specifically, the aim of our paper is to provide the methodological keys of calculating economic gains and the potential impact of such calculations on lapse risk management rather than to estimate the results of empirical applications. The insurance company interested in economic metrics can estimate relevant parameters based on its in-house data and explore the impact on benefits through an A/B testing methodology.

The two machine learning algorithms, XGBoost and SVM, perform a little bit better than classic CART and LR in terms of statistical accuracy on a large dataset consisting of more than 600,000 life insurance policies with information on policy terms and policyholders' characteristics. XGBoost has another advantage over other algorithms: it is less dependent on the choice of training samples.

Nevertheless, the introduction of retention gains tempers the advantages of XGBoost and SVM over the common LR. Indeed, our retained economic metrics significantly favor models with low false rate. Thus, LR leads to similar profits performance to XGBoost or SVM, emphasizing the importance of the validation measure used when comparing machine learning algorithms.

In the last section, we demonstrate that the economic gains can be further enhanced when the optimization is done on a function linked to economic gains rather than on statistic accuracies. The results show that the retention gains with a strong incentive strategy resulted from XGB_R1 is 126% of those from applying XGBoost to pursuing classification accuracies, in particular by reducing the false alarm rates. An insurer should therefore apply advanced machine learning algorithms like XGB to its economic objective, so that lapse management can really be optimized.

Acknowledgments

Stéphane Loisel acknowledges support from research chairs DAMI and NINA sponsored by BNP Paribas Cardif, as well as from the research initiative Sustainable Actuarial Science sponsored by Milliman Paris. Cheng-Hsien Jason Tsai is grateful to the Ministry of Science and Technology of Taiwan (project number MOST 105-2410-H-004 -070 -MY3) and Risk and Insurance Research Center of National Chengchi University for the financial supports.

Notes

1. There are some papers on the subject of modeling early terminations that do not fit our macro-micro classification on empirical, explanatory studies. They impose specific assumptions on the transition probabilities to early terminations (Buchardt *et al.*, 2015), the early terminations' intensity (Barsotti *et al.*, 2016), or the early termination rates (Loisel and Milhaud, 2011; Milhaud, 2013).

2. There exist other binary classification models that may be applied to lapse predictions (see, e.g., Wainer (2016) for an extensive quantitative comparison of binary classification algorithms). On the other hand, the main purpose of our paper is to introduce an economic point of view for the lapse risk management. We thus choose to limit our study to four models.

3. In the exposition, we follow the notation of Friedman (2002) so that the reader may go back to Friedman (2002) for more details. *eta* is the terminology used in the *R* package.

4. We are aware that the twofold cross-validation is unusual although the fivefold or 10-fold is more a practice than a proven theory. We chose twofold cross-validation to keep computational times reasonable, since we tested a large number of hyperparameters on large training samples (around 60k observations). The good news about a large dataset is that it reduces the variance of cross-validated estimations and the need of a large number of folds.

5. We indeed tested the polynomial kernel and linear SVM. The linear one had poorer predictive power, and the polynomial was not significantly different from the radial basis.

6. For dataset of more than 1k observations, Wainer and Cawley (2017) show that a twofold procedure is appropriate for SVM's parameter tuning.

7. These simplifications assume that the profitability ratio, the incentive, and the probability to accept the incentive is the same across policyholders, respectively. Upon the availability of data, we may compute an expected profitability ratio for each policy. The incentive offered to each policyholder can then be set as a function of the policy's profitability. The probability of accepting the offer can also be a function of the incentive, but such a function is difficult to estimate in practice. Face amount may be variable for some products, which increases the difficulty in estimating the expected profitability ratio. The retention probabilities may change with time, and this calls for a dynamic model of lapse propensities.

8. This represents 39% of the entire sample. We thus do not suffer from the accuracy paradox of unbalanced data.

9. Few policies are sold by independent agents or brokers; we put them into the TA category.

10. Paying premiums by automatic transfers from bank accounts or by recurring payments of credit cards is indifferent to policyholders. We thus regard these two automatic/recurring payment methods as one.

11. We are aware of the few extreme values such as an 80-year-old insured and one insured listed for 33 non-life policies. The samples with such extreme values are so few relative to our sample size that they will not affect our results in later sections.

12. The exchange rate used in the paper is 30 NTD/1 USD.

13. This policy is a whole life insurance with a 1-year-old insured and the death benefit of 10,000 NTD (a little over 300 USD). There are other small policies with death benefits smaller than 3000 USD. These policies constitute less than 1% of our samples.

14. Since our dataset contain over 600,000 policies, while XGBoost and SVM have hyperparameters to be tuned, training an algorithm on a one-tenth of the large dataset seems to be a balanced choice: sufficient to train the models while maintaining reasonable computational time.

15. We are aware of the subjectivity of these assumptions, but they will not impede comparisons across classification algorithms. Furthermore, these parameters can be estimated empirically by the insurer who is interested in economic gains.

References

Albizzati, M.-O. and Geman, H. (1994) Interest rate risk management and valuation of the surrender option in life insurance policies. *The Journal of Risk and Insurance,* **61**(4), 616–637. https://doi.org/10.2307/253641

Aleandri, M. (2017) Modeling dynamic policyholder behavior through machine learning techniques. *Working paper*. https://www.dss.uniroma1.it/en/system/files/pubblicazioni/CopiaPubblicazione.pdf

Ascarza, E., Neslin, S.A., Netzer, O., Anderson, Z., Fader, P.S., Gupta, S., Hardie, B.G.S., Lemmens, A., Libai, B., Neal, D., Provost, F. and Schrift, R. (2018) In pursuit of enhanced customer retention management: Review, key issues, and future directions. *Customer Needs and Solutions*, **5**(1), 65–81. https://doi.org/10.1007/s40547-017-0080-0

Bacinello, A.R. (2003) Pricing guaranteed life insurance participating policies with annual premiums and surrender option. *North American Actuarial Journal*, **7**(3), 1–17. https://doi.org/10.1080/10920277.2003.10596097

Barsotti, F., Milhaud, X. and Salhi, Y. (2016) Lapse risk in life insurance: Correlation and contagion effects among policyholders' behaviors. *Insurance: Mathematics and Economics*, **71**, 317–331. https://doi.org/10.1016/j.insmatheco.2016.09.008

Bauer, D., Kiesel, R., Kling, A. and Russ, J. (2006) Risk-neutral valuation of participating life insurance contracts. *Insurance: Mathematics and Economics*, **39**(2), 171–183. https://doi.org/10.1016/j.insmatheco.2006.02.003

Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992) A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, *COLT 1992*, pp. 144–152. New York, NY, USA: ACM. https://doi.org/10.1145/130385.130401

BREIMAN, L. (2001) Random forests. *Machine Learning*, **45**(1), 5–32. https://doi.org/10.1023/A:1010933404324

BREIMAN, L. (1996) Bagging predictors. *Machine Learning*, **24**(2), 123–140. https://doi.org/10.1007/BF00058655

BREIMAN, L., FRIEDMAN, J., OLSHEN, R. and STONE, C. (1984) *Classification and Regression Trees*. Wadsworth. https://doi.org/10.1201/9781315139470

BUCHARDT, K., MØLLER, T. and SCHMIDT, K.B. (2015) Cash flows and policyholder behaviour in the semi-Markov life insurance setup. *Scandinavian Actuarial Journal*, **8**, 1–29. https://doi.org/10.1080/03461238.2013.879919

CAMPBELL, J., CHAN, M., LI, K., LOMBARDI, L., PURUSHOTHAM, M. and RAO, A. (2014) Modeling of policyholder behavior for life insurance and annuity products: A survey and literature review. *Society of Actuaries*.

CERCHIARA R.R., GAMBINI, A. and EDWARDS, M. (2009) Generalized linear models in life insurance: Decrements and risk factor analysis under solvency II. *Giornale dell'Istituto Italiano degli Attuari*, **72**, 100–122.

CHEN, T. and GUESTRIN, C. (2016) XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *KDD 2016*, pp. 785–794. New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

CHEN, T., HE, T., BENESTY, M., KHOTILOVICH, V. and TANG, Y. (2015) *XGBoost: Extreme Gradient Boosting*. *R package*

CONSIGLIO, A. and GIOVANNI, D.D. (2010) Pricing the option to surrender in incomplete markets. *Journal of Risk and Insurance*, **77**(4), 935–957. https://doi.org/10.1111/j.1539-6975.2010.01358.x

CORTES, C. and VAPNIK, V. (1995) Support-vector networks. *Machine Learning*, **20**(3), 273–297. https://doi.org/10.1007/BF00994018

COX, S.H. and LIN, Y. (2006) Annuity lapse modeling: Tobit or not tobit? *Society of Actuaries*.

DAR, A. and DODDS, C. (1989) Interest rates, the emergency fund hypothesis and saving through endowment policies: Some empirical evidence for the U.K. *The Journal of Risk and Insurance*, **56**(3), 415–433. https://doi.org/10.2307/253166

ELING, M. and KIESENBAUER, D. (2014) What policy features determine life insurance lapse? An analysis of the German market. *Journal of Risk and Insurance*, **81**(2), 241–269. https://doi.org/10.1111/j.1539-6975.2012.01504.x

ELING, M. and KOCHANSKI, M. (2013) Research on lapse in life insurance: what has been done and what needs to be done? *The Journal of Risk Finance*, **14**(4), 392–413. https://doi.org/10.1108/JRF-12-2012-0088

European Insurance and Occupational Pensions Authority (EIOPA) (2011) EIOPA report on the Fifth Quantitative Impact Study (QIS5) for solvency II.

FRIEDMAN, J.H. (2002) Stochastic gradient boosting. *Computational Statistics & Data Analysis, Nonlinear Methods and Data Mining*, **38**(4), 367–378. https://doi.org/10.1016/S0167-94730100065-2

FRIEDMAN, J.H. (2001) Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, **29**(5), 1189–1232.

GATZERT, N. and SCHMEISER, H. (2008) Assessing the risk potential of premium payment options in participating life insurance contracts. *Journal of Risk and Insurance*, **75**(3), 691–712. https://doi.org/10.1111/j.1539-6975.2008.00280.x

GROSEN, A. and JØRGENSEN, P.L. (2000) Fair valuation of life insurance liabilities: The impact of interest rate guarantees, surrender options, and bonus policies. *Insurance: Mathematics and Economics*, **26**(1), 37–57. https://doi.org/10.1016/S0167-6687(99)00041-4

GUPTA, S., HANSSENS, D., HARDIE, B., KAHN, W., KUMAR, V., LIN, N., RAVISHANKER, N. and SRIRAM, S. (2006) Modeling customer lifetime value. *Journal of Service Research*, **9**(2), 139–155. https://doi.org/10.1177/1094670506293810

HOERL, A.E. and KENNARD, R.W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

HSU, C.-W., CHANG, C.-C. and LIN, C.-J. (2003) A practical guide to support vector classification. *Working Paper*. https://doi.org/10.1007/s11119-014-9370-9

HWANG, Y. and TSAI, C. (2018) Differentiating surrender propensity from lapse propensity across life insurance products. *Taiwan Risk Theory Seminar*, Taipei, Taiwan.
JAMAL, S. (2017) Lapse risk modeling with machine learning techniques: An application to structural lapse drivers. *Bulletin Français d'Actuariat*, **17**(33): 27–91.
KAGRAOKA, Y. (2005) Modeling insurance surrenders by the negative binomial model. *Working paper*.
KIM, C. (2005a) Surrender rate impacts on asset liability management. *Asia-Pacific Journal of Risk and Insurance*, **1**(1). https://doi.org/10.2202/2153-3792.1004
KIM, C. (2005b) Modeling surrender and lapse rates with economic variables. *North American Actuarial Journal*, **9**(4), 56–70. https://doi.org/10.1080/10920277.2005.10596225
KIM, C. (2005c) Report to the policyholder behavior in the tail subgroups project. *Society of Actuaries*.
Kuo, W., Tsai, C. and Chen, W.K. (2003) An empirical study on the lapse rate: The cointegration approach. *Journal of Risk and Insurance*, **70**(3), 489–508. https://doi.org/10.1111/1539-6975.t01-1-00061
LEMMENS, A. and CROUX, C. (2006) Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, **43**(2), 276–286.
LEMMENS, A. and GUPTA, S. (2020) Managing churn to maximize profits. *Marketing Science*, forthcoming.
LOISEL, S. and MILHAUD, X. (2011) From deterministic to stochastic surrender risk models: Impact of correlation crises on economic capital. *European Journal of Operational Research*, **214**(2), 348–357. https://doi.org/10.1016/j.ejor.2011.04.038
MEYER, D., DIMITRIADOU, E., HORNIK, K., LEISCH, F., WEINGESSEL, A., CHANG, C. and LIN, C. (2015) *Misc Functions of Department of Statistics, Probability, Theory Group (Formerly: E1071)*. R package version 1.7-0.
MILHAUD, X. (2013) Exogenous and endogenous risk factors management to predict surrender behaviours. *ASTIN Bulletin: The Journal of the IAA*, **43**(3), 373–398. https://doi.org/10.1017/asb.2013.2
MILHAUD, X., LOISEL, S. and MAUME-DESCHAMPS, V. (2011) Surrender triggers in Life Insurance: what main features affect the surrender behavior in a classical economic context? *Bulletin Français d'Actuariat*, **11**(22), 5–48.
NELDER, J.A. and WEDDERBURN, R.W.M. (1972) Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, **135**(3), 370–384. https://doi.org/10.2307/2344614
NESLIN, S.A., GUPTA, S., KAMAKURA, W., LU, J. and MASON, C.H. (2006) Defection detection: measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, **43**(2), 204–211. https://doi.org/10.1509/jmkr.43.2.204
NIELSEN, D. (2016) Tree boosting with XGBoost why does XGBoost win "Every" machine learning competition? (Master's thesis), *Norwegian University of Science and Technology*, Trondheim. http://pzs.dstu.dp.ua/DataMining/boosting/bibl/Didrik.pdf.
NOBLE, W.S. (2006) What is a support vector machine? *Nature Biotechnology*, **24**(12), 1565–1567. https://doi.org/10.1038/nbt1206-1565
PINQUET, J., GUILLÉN, M. and AYUSO, M. (2011) Commitment and lapse behavior in long-term insurance: A case study. *Journal of Risk and Insurance*, **78**(4), 983–1002. https://doi.org/10.1111/j.1539-6975.2011.01420.x
POWERS, D.M. (2011) Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, **2**(1), 37–63.
RENSHAW, A.E. and HABERMAN, S. (1986) Statistical analysis of life assurance lapses. *Journal of the Institute of Actuaries*, **113**(3), 459–497. https://doi.org/10.1017/S0020268100042566
THERNEAU, T., AKTINSON, B. and RIPLEY, B. (2018) *Recursive Partitioning and Regression Trees*. R package version 4.1-13.
TIBSHIRANI, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**(1), 267–288.
TSAI, C., KUO, W. and CHIANG, D.M.-H. (2009) The distributions of policy reserves considering the policy-year structures of surrender rates and expense ratios. *Journal of Risk and Insurance*, **76**(4), 909–931. https://doi.org/10.1111/j.1539-6975.2009.01324.x

VAFEIADIS, T., DIAMANTARAS, K.I., SARIGIANNIDIS, G. and CHATZISAVVAS, K.C. (2015) A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, **55**, 1–9. https://doi.org/10.1016/j.simpat.2015.03.003

WAINER, J. (2016) Comparison of 14 different families of classification algorithms on 115 binary datasets. *Working paper*.

XIA, G. and JIN, W. (2008) Model of customer churn prediction on support vector machine. *Systems Engineering - Theory & Practice*, **28**(1), 71–77. S1874-86510960003-X S1874-86510960003-X

ZHAO, Y., LI, B., LI, XIU, LIU, W. and REN, S. (2005) Customer churn prediction using improved one-class support vector machine. In: *Advanced Data Mining and Applications* (eds. Li, X., Wang, S. and Dong, Z.Y.), Lecture Notes in Computer Science, pp. 300–306. Berlin, Heidelberg: Springer.

STÉPHANE LOISEL (Corresponding author)
Univ Lyon, Université Claude Bernard Lyon 1
Institut de Science Financière et d'Assurances (ISFA)
Laboratoire SAF EA2429, F-69366, Lyon, France
E-Mail: stephane.loisel@univ-lyon1.fr

PIERRICK PIETTE
Univ Lyon, Université Claude Bernard Lyon 1
Institut de Science Financière et d'Assurances (ISFA)
Laboratoire SAF EA2429, F-69366, Lyon, France

Seyna, 58 Rue de la Victoire, 75009 Paris, France
E-Mail: pierrickpiette@gmail.com

CHENG-HSIEN JASON TSAI
Department of Risk Management and Insurance
Risk and Insurance Research Center, College of Commerce
National Chengchi University (NCCU)
Taipei City, Taiwan
E-Mail: ctsai@nccu.edu.tw

APPENDICES

APPENDIX A: XGBOOST TUNING – BINARY CLASSIFICATION

The values of the parameters tested in the grid search for the tuning of XGBoost are as follows:

- *eta*: 0.05, 0.1, 0.15;
- *gamma*: 0, 5, 10;
- *max_depth*: 10, 15, 20, 25, 30;

– *min_child_weight*: 15, 20, 25;
– *subsample*: 1;
– *colsample_bytree*: 0.4, 0.5, 0.6.

The values of the grid search are chosen by a previous sensitivity study in which we apply the same methodology on a subsample of the whole database but with a coarser grid. Then, we focus on a finer grid to obtain better results within a reasonable time period. In addition, the fact that we only test *subsample* with the value of 1 means that we do not adopt the stochastic gradient boosting of Friedman (2002).

Appendix B: SVM tuning

The values of the parameters tested in the grid search for the tuning of SVM are as follows:

– *Cost*: 0.5, 1, 2, 5, 10;
– *gamma*: 0.25, 0.5, 0.75, 1, 1.25.

Similar to the XGBoost tuning explained in Appendix A, the values of the grid search are chosen by a previous sensitivity study in which we apply the same methodology on a subsample of the whole database but with a coarser grid. Then, we focus on a finer grid to obtain better results. This is necessary, so that the computing can be done within a reasonable time period.

Appendix C: XGBoost Tuning – Profitability

We adopt the values of most parameters generated by a previous sensitivity study as:

– *eta* = 0.005;
– *gamma* = 1;
– *max_depth* = 15;
– *min_child* = 15;
– *subsample* = 0.7;
– *colsample* = 0.8.

Then, we determine the best *nrounds* through a fivefold cross-validation with this parameter tested up to 1000.