Applying economic measures to lapse risk management with machine learning approaches

Stéphane Loisel, Pierrick Piette, Jason Tsai

Applying economic measures to lapse risk management with machine learning approaches

Stéphane Loisel*[1], Pierrick Piette†[1,3,4], and Cheng-Hsien Jason Tsai‡[2]

[1]Univ Lyon, Université Claude Bernard Lyon 1, Institut de Science Financière et d'Assurances (ISFA), Laboratoire SAF EA2429, F-69366, Lyon, France.
[2]National Chengchi University (NCCU), College of Commerce, Department of Risk Management andd Insurance, Taipei, Taiwan.
[3]Sorbonne Université, CNRS, Laboratoire de Probabilités, Statistique et Modélisation, LPSM, 4 place Jussieu, F-75005 Paris, France.
[4]Sinalys, 6 rue de Téhéran, 75008 Paris, France.

Abstract

Modeling policyholders lapse behaviors is important to a life insurer since lapses affect pricing, reserving, profitability, liquidity, risk management, as well as the solvency of the insurer. Lapse risk is indeed the most significant life underwriting risk according to European Insurance and Occupational Pensions Authority's Quantitative Impact Study QIS5. In this paper, we introduce two advanced machine learning algorithms for lapse modeling. Then we evaluate the performance of different algorithms by means of classical 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 for lapse risk management. We apply different algorithms to a large real-world insurance dataset. Our results show that XGBoost and SVM outperform CART and logistic regression, especially in terms of the economic validation metric. The optimization after transformation brings out significant and consistent increases in economic gains.

JEL Classification: C52; C53; G22.

Keywords: lapse; machine learning; SVM; XGBoost; life insurance.

*Email: stephane.loisel@univ-lyon1.fr. The author acknowledges support from research chair DAMI (Data Analytics and Models for Insurance) sponsored by BNP Paribas Cardif.

†Email: pierrick.piette@gmail.com.

‡Email: ctsai@nccu.edu.tw. The author 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.

1 Introduction

Lapse risk is the most significant risk associated with life insurance when compared with longevity risk, expenses risk, and catastrophe risk. 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), cause underwriting expenses unrecovered (Tsai et al., 2009; Pinquet et al., 2011), impair the effectiveness of an insurer's asset-liability management (Kim, 2005c; Eling and Kochanski, 2013) and bring in liquidity threats as experienced by US life insurers in the late 1980s.

When lapses vary with interest rates as suggested by Dar and Dodds (1989), Kuo et al. (2003), Kim (2005a), Kim (2005b), 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, e.g., Albizzati and Geman (1994), Grosen and Løchte 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.

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, 2005a; Kim, 2005b; 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 environmental 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 include the characteristics of policyholders and the features of life insurance products/policies (see Renshaw and Haberman, 1986; Kagraoka, 2005; Cerchiara et al., 2008; Milhaud et al., 2011; Pinquet et al., 2011; Eling and Kiesenbauer, 2014, among others) . Eling and Kochanski (2013) and Campbell et al. (2014) provide extensive reviews of the literature on lapses[1].

This paper extends the micro-oriented line of literature in two ways. Firstly, we introduce machine learning algorithms including Extreme Gradient Boosting (XGBoost) and Support Vector Machine (SVM) to lapse behavior modeling. These two advanced algorithms have their merits over other approaches used in the literature such as generalized linear models (i.e., binomial and Poisson models and logistic regression), Classification and Regression Tree (CART) analysis, and the proportional hazards model. Secondly, we adopt economic measures in addition to statistical accuracy in evaluating the performance of different algorithms. Such an adoption better demonstrates how different algorithms may benefit the insurer.

Thirdly, we transform the optimization objective from classification accuracy to economic gains to demonstrate the benefit of integrating modeling with profit maximization. Such an integration

[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).

can increase life insurers' profitability, improve insurers' customer management through taking preventive measures to reduce lapses, and retain more of the so-called Contractual Service Margin (CSM) in International Financial Reporting Standard (IFRS) 17. It also links us to the literature on churn management and its impact on the customer lifetime value (e.g., Neslin et al., 2006; Lemmens and Croux, 2006; Lemmens and Gupta, 2017).

The results from applying different algorithms to a large dataset consisting of more than six hundred thousand life insurance policies show that XGBoost and SVM outperform CART and logistic regression with respect to statistic accuracy. The results further show that XGBoost is the most robust across training samples.

The advantages of XGBoost and SVM are more apparent with respect to retention gains. The retention gain takes into account the costs of providing incentives to policyholders to reduce their propensities towards lapses, the benefits of retaining policies, and the costs of false alarms. XGBoost and SVM generate much higher retention gains than logistic regression and CART do.

Last but not least, we confirm that economic gains can be further enhanced when the optimization is done on a function linked to the gains rather than on statistic accuracies. The resulted retention gains are 126% of those from applying XGBoost to pursue classification accuracies, and the increase in retention gains remains to be significant under an alternative policyholder retention scheme. An insurer, therefore, should apply robust machine learning algorithms like XGBoost to 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 logistic regression. In Section 3 we delineate two performance metrics to be used. One is the commonly seen accuracy, i.e., 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 profit objectives with those from optimization statistic accuracy. Section 7 summarizes and concludes 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, i.e., $y_i \in \{0, 1\}$ Popular predictive models include logistic regression and CART models. More advanced machine learning models that we introduce in this paper are SVM and XGBoost.

2.1 XGBoost

XGBoost is an extension of the gradient boosting introduced by Friedman (2001). The gradient boosting tree is an ensemble method, i.e., multiple weak learners h are combined to become a strong learner F in order to achieve a better predictive performance. The following descriptions are summarized from Friedman (2002)).

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

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

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

$$\begin{aligned} F_M(\mathbf{x}) &= \sum_{m=0}^{M} \beta_m h\left(\mathbf{x}, \{R_{lm}\}_1^L, \bar{y}_{lm}\right) \\ &= \sum_{m=0}^{M}\sum_{l=1}^{L} \beta_m \bar{y}_{lm} \mathbb{1}\left(\mathbf{x} \in R_{lm}\right), \end{aligned} \tag{2.2}$$

where $\{R_{lm}\}_1^L$ and $\bar{y}_{lm}$ are the L-disjoint regions and the corresponding split points determined by the mth regression tree, respectively, and $beta_m$ are the expansion coefficients. This strong learner is estimated through a stage-wise method that begins with an initial guess $F_0(\mathbf{x})$. Then the pseudo-residuals for $m = 1, 2, \ldots, M$ are computed:

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

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

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{lm}} \boldsymbol{\Psi}\left(y_i, F_{m-1}(\mathbf{x}_i) + \gamma\right). \tag{2.4}$$

At the final stage, the strong learner is updated,

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu.\gamma_{lm} \mathbb{1}\left(\mathbf{x} \in R_{lm}\right), \tag{2.5}$$

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

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

Inspired by previous general works on statistical learning, many extensions to the gradient boosting tree method have been developed. 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 $\left\{\tilde{y}_{\pi(i)m}, \mathbf{x}_{\pi(i)}\right\}_1^{\tilde{N}}$. Another way to inject randomness that has been popularized by Breiman (2001) 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 $\{\tilde{y}_{im}, P^*(\mathbf{x})_i\}_1^N$ in which $P^*(\mathbf{x}) = \left\{x_{\pi^*(1)}, \ldots, x_{\pi^*(\tilde{n})}\right\}$.

To avoid overfitting, some extensions follow the general idea of the ridge regression (Hoerl and Kennard, 1970) and lasso regression (Tibshirani, 1996) and adopt the penalized optimization

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

Figure 1: Pseudocode of the Gradient Tree Boosting algorithm (Friedman, 2002).

point of view. Instead of optimizing a loss function $\mathbf{\Psi}(y, F(\mathbf{x}))$, the problem is modified as the optimization on an "objective" function $\boldsymbol{O}$ that is the sum of a loss function $\mathbf{\Psi}$ and a regularization term $\mathbf{\Omega}$:

$$\boldsymbol{O}(y, F(\mathbf{x})) = \mathbf{\Psi}(y, F(\mathbf{x})) + \mathbf{\Omega}(F). \tag{2.6}$$

Among all the boosting packages that have been developed, the XGBoost system (Chen and Guestrin, 2016) has become the most popular due to its flexibility and computing performances. It has also become the most popular machine learning algorithm in data science challenges such as Kaggle for structured data. We list the main parameters that need to be tuned, using the package's terminology and the notation of (Friedman, 2002), as follows.

1. *nrounds* is the number of trees to grow: M;
2. *eta* is the shrinkage parameter: $M\nu$;
3. *gamma* is the regularization parameter which is used in $\mathbf{\Omega}$;
4. *max_depth* is the number of nodes of a tree: L;
5. *min_child_weight* is the minimal number of observations in a node and $\min_{l,m} \sum_{i=1}^{N} \mathbb{1}(\mathbf{x}_i \in R_{lm})$ should be higher than this value;
6. *subsample* is the relative size of the random subsample used in the case of a stochastic gradient boosting: $\tilde{N}/N$;
7. *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:

$$\mathbf{\Psi}(y, \hat{y}) = \sum_{i=1}^{N} \left[y_i \ln\left(1 + e^{-\hat{y}_i}\right) + (1 - y_i) \ln\left(1 + e^{-y_i}\right) \right], \tag{2.7}$$

and the error function as the metric for cross-validation:

$$error(y, \hat{y}) = \frac{\sum_{i=1}^{N} \mathbb{1}\left(y_i \neq round\left(\hat{y}_i\right)\right)}{N}, \tag{2.8}$$

where $round\left(\hat{y}_i\right) = \begin{cases} 1 \text{ if } \hat{y}_i \geqslant 0.5, \\ 0 \text{ if } \hat{y}_i < 0.5. \end{cases}$

The tuning method that we adopt consists of two nested cross-validations. We first perform a grid search on the parameters except ***nrounds*** with a 2-folds cross-validation (the grid of values is reported in Appendix 1). Then we determine the best ***nrounds*** through a 5-folds cross-validation up to 200 for every possible set of parameters in the grid.

2.2 SVM

The theory of SVM was introduced in the 1990's by Boser et al. (1992) and Cortes and Vapnik (1995). 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).

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 hyper-plane. To deal with misclassifications, a soft margin (i.e., a penalty determined by the user) is imposed upton 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 by using the notation of Hsu et al. (2003). Given a training sample $\{y_i, \mathbf{x}_i\}_1^N$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$, the SVM algorithm is the solution of the following optimization problem:

$$\min_{\omega, b, \xi} \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\omega} + C \sum_{i=1}^{N} \xi_i, \tag{2.9}$$

with the constraints

$$y_i \left(\boldsymbol{\omega}^\top \phi(\mathbf{x}_i) + b\right) \geqslant 1 - \xi_i, \xi_i \geqslant 0. \tag{2.10}$$

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 $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$.

In our case we choose to consider the radial basis function kernel (also called RBF kernel) that is the most commonly used in practice and determined by

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2\right), \tag{2.11}$$

with $\gamma > 0$ being the kernel parameter.

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 2-folds cross-validation and adopt

the misclassification error function as the validation metric. The grid of values is reported in Appendix 2.

2.3 CART

CART was first introduced by Breiman et al. (1984). The underlying idea is straight forward: 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 two disjointed subspaces. At each step or node of this top-down construction, the explanatory variable and the dividing point are chosen to minimize the Gini impurity of the node.

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 in the node is defined by $p_l = \frac{1}{N_l}\sum_{i\in l} y_i$. Then use an algorithm to partition the parent node into two nodes l_L and l_R by maximizing

$$I_G(l) - \left[I_G(l_L) - I_G(l_R)\right], \tag{2.12}$$

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

$$I_G(l) = N_l p_l (1 - p_l). \tag{2.13}$$

This construction is applied up to obtaining a node for every observation point. The tree obtained is thus designated as the saturated model. Although fitting the response on the training sample perfectly, it generally leads to low predictive performance when applied to new samples. Hence the tree needs to be pruned, i.e., the number of final nodes needs to be reduced to increase its predictive power.

Many criteria can be used to prune the tree, e.g., the minimum number of observations in a final node. We choose L, the number of terminal nodes, that minimizes the misclassification error:

$$error(y, \hat{y}) = \frac{\sum_{i=1}^{N} \mathbb{1}\left(y_i \neq \hat{y}_i\right)}{N}. \tag{2.14}$$

L is estimated by a 10-folds cross-validation methodology. We use the `rpart` R package (Therneau et al., 2018) to implement CART.

2.4 Logistic Regression

The logistic regression is a special case of the generalized linear models (Nelder and Wedderburn, 1972) obtained with the Bernoulli distribution. 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 $\{y_i, \mathbf{x}_i\}_1^N$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{0,1\}$, the regression model is specified as:

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

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

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

When applying the estimated logistic regression model to a classification problem, it doesn't directly lead to labeled responses but to estimated probabilities. To determine the forecasted class, we chose the common threshold of 0.5, i.e.,

$$\hat{y}_i^* = \begin{cases} 1 \text{ if } \hat{y}_i \geqslant 0.5, \\ 0 \text{ if } \hat{y}_i < 0.5. \end{cases} \tag{2.17}$$

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 $(1,0)$ while the number of each case is usually laid out in the so-called confusion matrix. Denote $N(j,k)$ as the coefficients of the confusion matrix in which $j \in \{0,1\}$ stands for the observed lapse indicator and $k \in \{0,1\}$ the predicted lapse indicator. Given a set of response variables $\{y_i, \hat{y}_i\}_1^N$, we estimate $N(j,k)$ as:

$$N(j,k) = \sum_{i=1}^{N} \mathbf{1}\left(y_i, \hat{y}_i = k\right). \tag{3.1}$$

3.1 Statistical metric

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

$$\begin{aligned} accuracy(y, \hat{y}) &= \frac{N(1,1) + N(0,0)}{N} \\ &= 1 - error(y, \hat{y}). \end{aligned} \tag{3.2}$$

3.2 Economic metric

Although we adopt mathematic 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 estimate the impacts of different classification results on the expected profits from policies, also called customer lifetime values. 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 ($\Theta_i \in \mathbb{N}$). 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:

$$CLV_i = \sum_{t=0}^{\Theta_i} \frac{p_{i,t} F_{i,t}}{(1+d_t)^t},, \tag{3.3}$$

where d_t is the discount rate.

Assuming a deterministic time horizon $T \in \mathbb{N}$, we ddefine 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 interest rates respectively. Given the four vectors, the customer lifetime value is

$$CLV_i\left(\boldsymbol{p}_i, \boldsymbol{F}_i, \boldsymbol{r}_i, \boldsymbol{d}\right) = \sum_{t=0}^{T} \frac{p_{i,t} F_{i,t} r_{i,t}}{(1+d_t)^t},, \tag{3.4}$$

The lapse management strategy is modelled 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:

1. $\boldsymbol{p}_i$ are the same for all policies and denoted as $\boldsymbol{p}$ hereafter;
2. $\boldsymbol{\delta}_i$ are the same for all contacted policies and denoted as $\boldsymbol{\delta}$ hereafter;
3. $p_{i,t}$, $F_{i,t}$ and d_t remain constant across time;
4. $\boldsymbol{r}_i$ equals to $\boldsymbol{r}_{lapse}$ or $\boldsymbol{r}_{stay} = (1, 1, \ldots, 1)$ and $\boldsymbol{r}_{lapse}$ is estimated on the dataset and will be given in Section 5.2;
5. if $\boldsymbol{r}_i = \boldsymbol{r}_{stay}$, the incentive is accepted with probability $\gamma_i = 1$ and $\boldsymbol{r}_i^* = \boldsymbol{r}_{stay}$;
6. if $\boldsymbol{r}_i = \boldsymbol{r}_{lapse}$, the incentive is accepted with probability $\gamma_i = \gamma$ and $\boldsymbol{r}_i^* = \boldsymbol{r}_{stay}$.[2] Policyholders who reject the offers (probability $= 1 - \gamma$) will lapse their policies, i.e. $\boldsymbol{r}_i^* = \boldsymbol{r}_{lapse}$.

The application of a segmentation 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,

$$F(j,k) = \sum_{i=1}^{N} F_i . \mathbf{1}\left(y_i, \hat{y}_i = k\right), \tag{3.5}$$

while N is defined in Equation 3.1.

[2] 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 difficut 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

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

$$RPV = CLV\left(\boldsymbol{p}, F(0,0) + F(0,1), \boldsymbol{r}_{stay}, \boldsymbol{d}\right) + CLV\left(\boldsymbol{p}, F(1,0) + F(1,1), \boldsymbol{r}_{lapse}, \boldsymbol{d}\right). \tag{3.6}$$

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

$$\begin{aligned} LMPV(\boldsymbol{\delta}, \gamma, c) = {} & CLV\left(\boldsymbol{p}, F(0,0), \boldsymbol{r}_{stay}, \boldsymbol{d}\right) + CLV\left(\boldsymbol{p}, F(1,0) + (1-\gamma)F(1,1), \boldsymbol{r}_{lapse}, \boldsymbol{d}\right) \\ & + CLV\left(\boldsymbol{p} - \boldsymbol{\delta}, F(0,1) + \gamma F(1,1), \boldsymbol{r}_{stay}, \boldsymbol{d}\right) - c\left(N(0,1) + N(1,1)\right). \end{aligned} \tag{3.7}$$

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

$$RG(\boldsymbol{\delta}, \gamma, c) = LMPV(\boldsymbol{\delta}, \gamma, c) - RPV, \tag{3.8}$$

that can be simplified as

$$\begin{aligned} RG(\boldsymbol{\delta}, \gamma, c) = {} & \gamma\left[CLV\left(\boldsymbol{p} - \boldsymbol{\delta}, F(1,1), \boldsymbol{r}_{stay}, \boldsymbol{d}\right) - CLV\left(\boldsymbol{p}, F(1,1), \boldsymbol{r}_{lapse}, \boldsymbol{d}\right)\right] \\ & + CLV\left(\boldsymbol{\delta}, F(0,1), \boldsymbol{r}_{stay}, \boldsymbol{d}\right) - c\left(N(0,1) + N(1,1)\right). \end{aligned} \tag{3.9}$$

4 Data

Our data come from a medium-size 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 31/08/2013. 243,152 policies out of all samples were lapsed, and 5,486 insureds died during the sampling period.

We specify several variables based on the literature and the data provided by the insurer as input to the algorithms of Section 2. Firstly we are able to identify from the data the age, gender, and occupation of an insured at the time when the policy was issued. Female is designated as 1 while male 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) a person are listed as the insured (since a person may purchase multiple policies).

The data also contain the inception date and face amount of each policy. There are three types of policies. The most popular type is traditional policies like term life, whole life, and endowment. Investment-linked and interest-adjustable types of products appeared in 2000s. We also able to identify whether a policy is a single-premium one or not. There are three cases with regard to participation. It was not until 2004 that insurers were allowed to sell non-participating policies. The policies sold by the end of 2003 are thus designated as Mandotory 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. Firstly, we categorize distribution channels as Tied Agents (denoted by TA), Direct Marketing (DM), and Banks (BK)[3]. Secondly, 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)[4], and going to the post office or convenient stores in person (P&C).

Table 1 and 2 present the descriptive statistics of the above explanatory variables. The average age of the sampled insureds is 28 and the standard deviation of the insureds' age is 17. The minimum, medium, and maximum age is 0, 27, and 80, respectively. The samples consist of relatively equivalent portions of male and female insureds. 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. 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.

The mean and medium of policy inception dates are in the second quarter of 2005, and the standard deviation around this quarter is almost 5 years. The face amount of the sampled policies has an average of 17,165 US dollars[5] with big variations: the largest policy reaches 2 million dollars, the smallest one is only 333 dollars[6], and the standard deviation is about twenty-eight thousand dollars. Around 3% of the samples are single-premium policies. 46.6% of samples are mandatory-participating policies while 37.2% are non-participating ones. Almost all policies are traditional types of products ; interest-adjustable and investment-linked types of products are merely 3% of our samples. 88% of policies are dominated in NTD.

Table 1 also shows that selling life insurance through tied agents is the major way (94%) of this insurer while the sampled policies sold through direct marketing are smaller than 3%. It further shows that 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.

5 Result with respect to statistical and economic metrics

Our focus is on the predictive performance of different algorithms. We 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, $k \in \{1, \ldots, \}$. The estimated model is subsequently applied to the other subsamples 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. This procedure enables us to make sure that every observation is used, at some point of an algorithm, as both training and testing samples. It is similar to the k-fold cross-validation technique in which the training subsample is composed of $D - D_k$ 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.

[3]Few policies are also sold by independant agents, brokers that we gather in the same category.

[4]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.

[5]The exchange rate used in the paper is 30 NTD/1 USD.

[6]This policy is a whole life insurance with a one-year old insured and the death benefit of ten thousand NTD (a little over three hundred USD). There are other small policies with death benefits smaller than three thousand USD. These policies constitute less than one percent of our samples.

Table 1: Descriptive statistics of categorical explanatory variables.

Variables	**Category**	**Percentage**
Gender		
	Female	48
	Male	52
Occupation		
	Tier one	80.5
	Requiring extra screening	19.5
Physical Examination		
	Exempted	96.4
	Required	3.6
Distribution Channel		
	TA	93.9
	BK	3.4
	DM	2.4
	Others[7]	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
	Traditional	97.1
Currency Domination		
	NTD	88.1
	Others	11.9

Table 2: Descriptive statistics of continuous explanatory variables.

	Mean	**Medium**	**St. Dev.**	**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

5.1 Results with respect to the statistical metric

The mean accuracy computed using the above cross-validation procedure is displayed in the Table 3 and Figure 2 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 logistic regression (LR). XGBoost surpasses logistic regression by 2.24% on average, which represents a significant improvement of 12,684 correctly classified policies. Moreover, the smallest standard deviation of accuracy of the XGBoost, 0.03%, indicates that XGBoost is less prone to sample selection. This is visible in the box plot of Figure 2.

Table 3: Cross-Validated Statistic Accuracies.

	LR	CART	SVM	XGB
Mean Accuracy	76.64%	77.15%	77.82%	78.8%
Standard Deviation	0.07%	0.10%	0.08%	0.03%

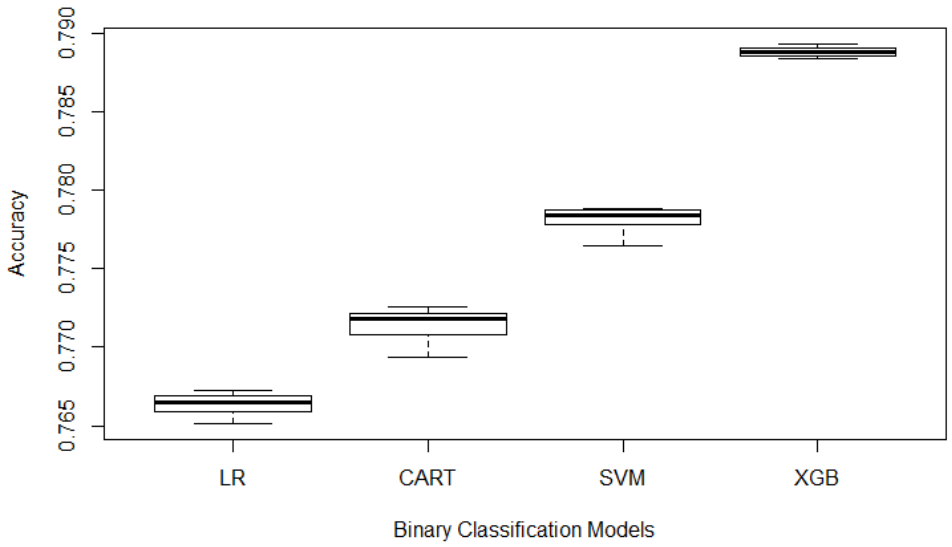

Figure 2: Box plot of statistic accuracies.

Looking at the entire confusion matrices in Tables 4 to 7, we find that CART predicts the most lapses (191,869 = 51,241 + 140,628) from which it identifies the most lapses correctly (140,628) but also signals the most false alarms (51,241). SVM predicts the most stays (398,597 = 310,258 + 88,339) in which it identifies the most stays correctly (310,258) while produces many false security cases (88,339). XGBoost is rather robust on the other hand. It is ranked the second in terms of all aspects: correctly identifying lapses (137,660), correctly identifying stays (309,111), not producing false alarms (38,450), and not producing false securities (81,177).

Table 4: Average confustion matrix of XGB.

		Predicted	
		Stay	Lapse
Actual	Stay	309,111	38,450
	Lapse	81,177	137,660

Table 5: Average confustion matrix of SVM.

		Predicted	
		Stay	Lapse
Actual	Stay	310,258	37,303
	Lapse	88,339	130,498

Table 6: Average confustion matrix of CART.

		Predicted	
		Stay	Lapse
Actual	Stay	296,320	51,241
	Lapse	78,209	140,628

Table 7: Average confustion matrix of LR.

		Predicted	
		Stay	Lapse
Actual	Stay	304,025	43,537
	Lapse	88,775	130,062

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 is available for us to estimate these parameters, we have to make assumptions. We had conducted sensitivity analyses and confirmed that the comparison results remain the same in general.

The time horizon T is set to 12 years according to the length of the sampling period. We estimate the retention probability vector $\boldsymbol{r}_{lapse}$ from the dataset and obtain the vector displaid in Table 8.

Table 8: Estimated retention probability $\boldsymbol{r}_{lapse}$.

Year	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	0.10	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: an aggressive one and a moderate one. The incentive vectors are described in Table 9

Table 9: Incentive strategies.

Year	0	1	2	3	4	5	6	7	8	9	10	11	12
Incentive 1 (in bp)	0	0	3	3	6	6	9	9	12	12	15	15	18
Incentive 2 (in bp)	0	0	1.5	1.5	3	3	4.5	4.5	6	6	6	6	6

We further assume that the probabilities of accepting the incentives for a would-lapse policyholder are $\gamma_1 = 20\%$ and $\gamma_2 = 10\%$ respectively.

The results from comparing different classification algorithms by the economic metric with the aggressive incentive strategy are displayed in Table 10 and Figure 3. The winner looks to be XGBoost: it has the highest retention gain with the smallest standard deviation across subsampling. Figure 3 further illustrates that XGBoost and SVM lead to similar retention gain compared to logistic regression and CART.

Notice that the differences across the algorithms are wider in terms of the economic metric than the statistical metric. The accuracies of the models are between 76.64% and 78.88%, which means an improvement ratio of 2.9%. The retention gains, on the other hand, range from 2.7 and 5.2 million USD, indicating an enhancement of 96%. 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 3c) 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.

Table 10: Cross-validated retention gains with the aggressive strategy.

	LR	CART	SVM	XGB
Mean Retention Gain	4,046,602	2,680,012	5,028,737	5,243,913
Standard Deviation	133,993	209,220	139,102	115,415

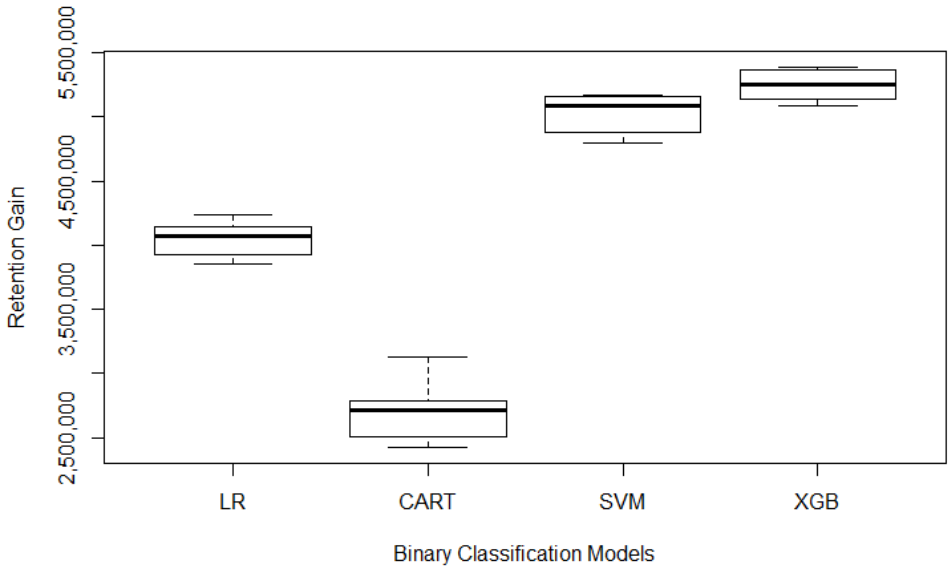

Figure 3: Box plot of retention gains with the aggressive strategy.

Then we look at algorithms' performances when the incentive strategy is moderate and leads to lower acceptance probabilities. The results are displayed in the Table 11 and the Figure 4. We first notice XGB and SVM remains to be ranked No. 1 and No. 2, respectively. Next we observe that the improvement ratio of the best algorithm over the worst is smaller but remains to be significant (56%). Thirdly, retention gains are significantly lower with the moderate incentive strategy. For instant, XGB achieves a gain of 5.2 million dollars with the aggressive incentive strategy but the gain reduces to 3.3 million dollars when incentives offered to policyholders are moderate. Under our assumptions, the company should rather set the aggressive incentive strategy up to optimize her gains. However, in practice, one would need a more complete sensitivity study on the incentive to be offered and the corresponding acceptance probability to fully optimize the lapse management.

Table 11: Cross-validated retention gains with the moderate strategy.

	LR	CART	SVM	XGB
Mean Retention Gain	2,618,396	2,085,599	3,113,900	3,261,029
Standard Deviation	63,693	85,184	54,169	45,928

In summary, XGB and SVM consistently perform better than CART and LR no matter which performance index, statistical accuracy or retention gains with alternative incentive strategies, is used. The drawbacks of XGB and SVM relative to CART and LR that we may think of are not

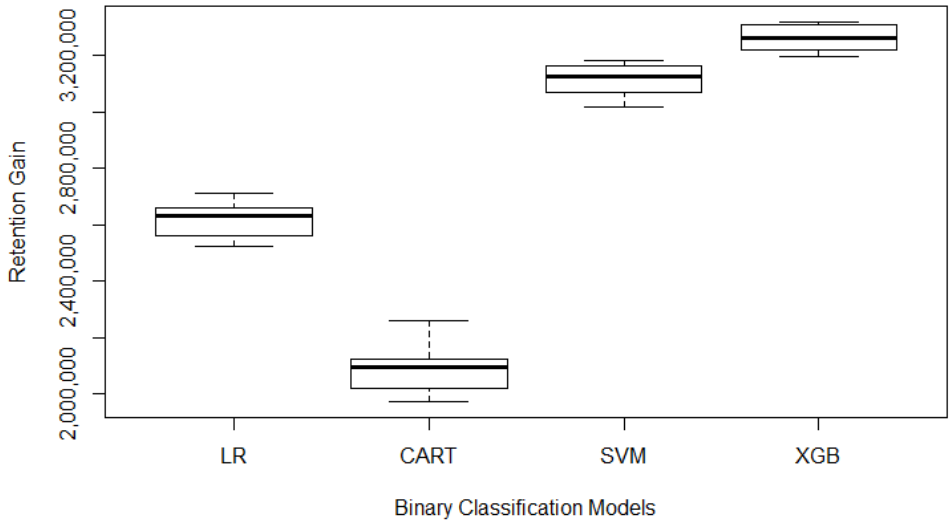

Figure 4: Box plot of retention gains with the moderate strategy.

related to performance. For instance, XGB and SVM are less transparent, more complex, demanding more computing power, and more difficult to be comprehended by inexperienced persons than CART and LR.

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 resulted 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 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 (2017) modify the usual loss function into a profit-based function to optimize economic gains. They obtain a significantly increase in the expected profit of a retention campaign. Learning from the churn literature, we transform the above classificaton 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 resulting from proposing the incentive $j \in \{1, 2\}$ (cf. Section 5.2) to policyholder i. More specifically, we define $z_i^{R_j}$ as

$$z_i^{R_j} = \begin{cases} -CLV\left(\boldsymbol{\delta}_j, F_i, \boldsymbol{r}_{stay}, \boldsymbol{d}\right) - c \text{ if } y_i = 0, \\ \gamma_j.\left[CLV\left(\boldsymbol{p} - \boldsymbol{\delta}_j, F_i, \boldsymbol{r}_{stay}, \boldsymbol{d}\right) - CLV\left(\boldsymbol{p}, F_i, \boldsymbol{r}_{lapse}, \boldsymbol{d}\right)\right] - c \text{ if } y_i = 1. \end{cases} \tag{6.1}$$

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

$$\mathbf{\Psi}\left(z^{R_j}, \hat{z}^{R_j}\right) = \frac{1}{N}\sum_{i=1}^{N}\left[z_i^{R_j} - \hat{z}_i^{R_j}\right]^2, \tag{6.2}$$

and as the metric for cross-validation.

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

$$\hat{y}_i = \begin{cases} 1 \text{ if } \hat{z}^{R_j} \geqslant 0, \\ 0 \text{ if } \hat{z}^{R_j} < 0, \end{cases} \tag{6.3}$$

By this way we can apply the same metrics described in previous sections. Here $\hat{y}_i$ is better to be understood as the estimation of the profitability about offering an incentive to the policyholder i rather than the forecast on the policyholder's lapse.

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 apply to estimating the parameters is described in Appendix 3.

6.2 Results

Table 12 and Figure 5 display the prediction accuracies. Table 12 shows that XGB_R1 and XGB_R2 produce relatively low mean accuracy of respectively 76.7% and 75.7% While XGB_R2 is clearly the worst model in term of accuracy, XGB_R1 generates similar results to the logistic regression which is the worst binary classification model regarding the accuracy measure. These seemingly unsatisfied results are understandable since both XGB_R1 and XGB_R2 are not designed to predict whether a policy would be lapsed or not. What they aim for are economic gains.

Table 12: Cross-Validated Statistic Accuracies.

	LR	CART	SVM	XGB	XGB_R1	XGB_R2
Mean Accuracy	76.64%	77.15%	77.82%	78.8%	76.67%	75.71%
Standard Deviation	0.07%	0.10%	0.08%	0.03%	0.07%	0.06%

The numbers in Table 13 and 14 tell us more about why XGB_R1 and XGB_R2 performs badly in statistical accuracy. They result in the smallest correct identifications on lapses (resp. 104,889 and 99,432) and produce the most false-sense-of-security (resp. 113,948 and 119,405). However, we will see very soon that XGB_R1 and XGB_R2 stand out when we switch focus to retention gain.

Table 15 and Figure 6 show that XGB_R1 generates a significantly larger average retention gain with the aggressive incentive strategy ($6,586,357) than other algorithms as well as a significantly lower standard deviation ($53,460). The increase in retention gain is 26% (1.3 million USD) higher than that generated by XGB (the second-best algorithm) and 146% (3.9 million USD) better than that produced by CART. Looking back to Table 13, we see that XGB_R1 leads to reduce the number of false alarms (18,204) in optimizing the retention gain, even if this also reduces the

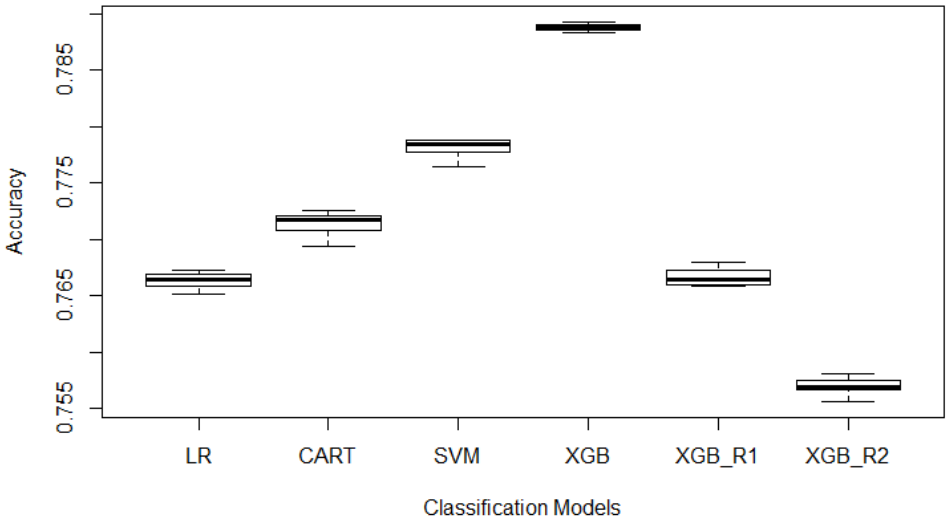

Figure 5: Box plot of statistic accuracies.

Table 13: Average confustion matrix of XGB_R1.

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

Table 14: Average confustion matrix of XGB_R1.

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

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 6.2, 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 16 and Figure 7. 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 increases with respect to the commonly seen LR and CART reach 47% and 85%.

Table 15: Cross-validated retention gains with the aggressive strategy.

	LR	CART	SVM	XGB	XGB_R1
Mean Retention Gain	4,046,602	2,680,012	5,028,737	5,243,913	6,586,357
Standard Deviation	133,993	209,220	139,102	115,415	53,460

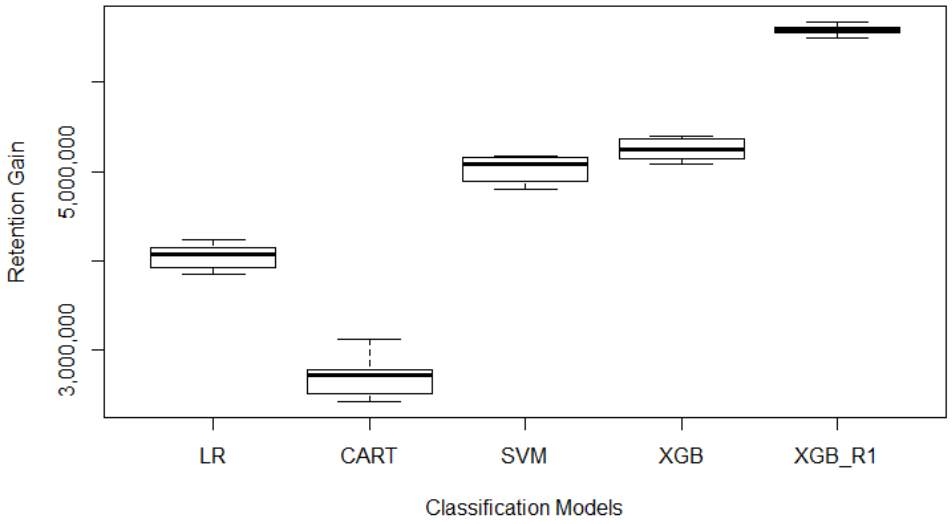

Figure 6: Box plot of retention gains with the aggressive strategy.

Table 16: Cross-validated retention gains with the moderate strategy.

	LR	CART	SVM	XGB	XGB_R2
Mean Retention Gain	2,618,396	2,085,599	3,113,900	3,261,029	3,852,782
Standard Deviation	63,693	85,184	54,169	45,928	39,163

The results in this section demonstrate the benefit of having a specific objective. If senior managers of an insurer are able to specify an objective to be optimized (e.g., maximizing retention gain), the staff should apply an advanced algorithm like XBG directly 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 mis-pricing, impair the risk management, or even pose solvency threats. Employing a good algorithm to model policyholder lapse behavior is therefore valuable.

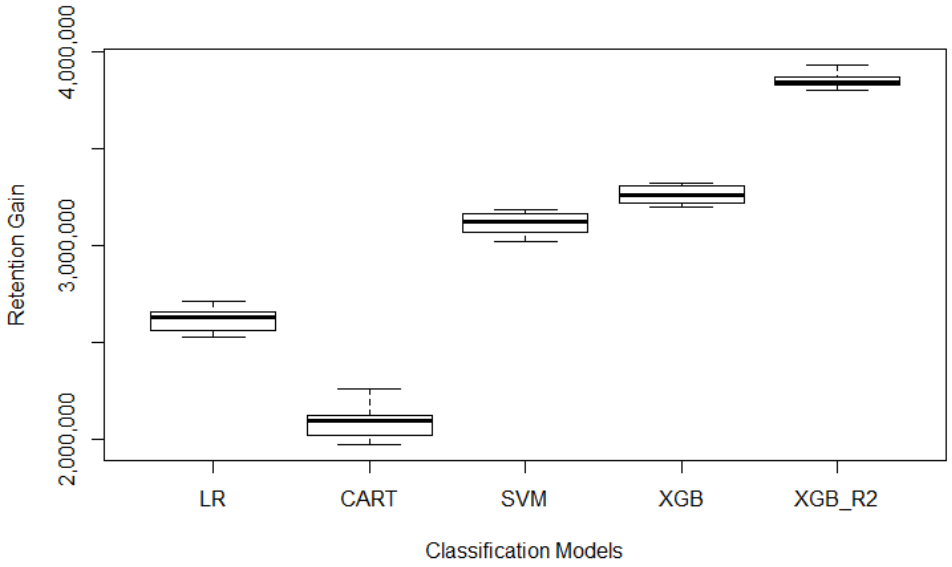

Figure 7: Box plot of retention gains with the moderate strategy.

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

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

The advantages of XGBoost and SVM are more apparent with respect to retention gains. The retention gains incorporate the costs of providing incentives to policyholders to reduce lapse propensities and the benefits of retaining policies. XGBoost and SVM generate much higher retention gains than logistic regression and CART do. For instance, XGBoost produces 1.2 to 2.6 million dollars more economic gains than CART.

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 an aggressive incentive strategy resulted from XGB_R1 is 126% of those from applying XGBoost to pursue 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 be really optimized.

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, pp. 616–637. DOI: 10.2307/253641.

Ascarza, E. et al. (2018). In Pursuit of Enhanced Customer Retention Management: Review, Key Issues, and Future Directions. *Cust. Need. and Solut.* 5.1, pp. 65–81. DOI: 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, pp. 1–17. DOI: 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, pp. 317–331. DOI: 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, pp. 171–183. DOI: 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. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. event-place: Pittsburgh, Pennsylvania, USA. New York, NY, USA: ACM, pp. 144–152. DOI: 10.1145/130385.130401.

Breiman, L. (1996). Bagging predictors. *Mach Learn* 24.2, pp. 123–140. DOI: 10.1007/BF00058655.

— (2001). Random Forests. *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and Regression Trees. Wadsworth. DOI: 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, pp. 1–29. DOI: 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., Edwards, M., and Gambini, A. (2008). Generalized Linear Models in Life Insurance: Decrements and Risk Factor under Solvency II. *18th international AFIR colloquium*. Rome.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. event-place: San Francisco, California, USA. New York, NY, USA: ACM, pp. 785–794. DOI: 10.1145/2939672.2939785.

Consiglio, A. and Giovanni, D. D. (2010). Pricing the Option to Surrender in Incomplete Markets. *Journal of Risk and Insurance* 77.4, pp. 935–957. DOI: 10.1111/j.1539-6975.2010.01358.x.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach Learn* 20.3, pp. 273–297. DOI: 10.1007/BF00994018.

Cox H., S. 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, pp. 415–433. DOI: 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, pp. 241–269. DOI: 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, pp. 392–413. DOI: 10.1108/JRF-12-2012-0088.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29.5, pp. 1189–1232.

— (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*. Nonlinear Methods and Data Mining 38.4, pp. 367–378. DOI: 10.1016/S0167-9473(01)00065-2.

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, pp. 691–712. DOI: 10.1111/j.1539-6975.2008.00280.x.

Grosen, A. and Løchte Jørgensen, P. (2000). Fair valuation of life insurance liabilities: The impact of interest rate guarantees, surrender options, and bonus policies. *Insurance: Mathematics and Economics* 26.1, pp. 37–57. DOI: 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, pp. 139–155. DOI: 10.1177/1094670506293810.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12.1, pp. 55–67. DOI: 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*. DOI: 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*.

Kagraoka, Y. (2005). Modeling Insurance Surrenders by the Negative Binomial Model. *Working paper*.

Kim, C. (2005a). Modeling Surrender and Lapse Rates With Economic Variables. *North American Actuarial Journal* 9.4, pp. 56–70. DOI: 10.1080/10920277.2005.10596225.

— (2005b). Report to the Policyholder Behavior in the Tail Subgroups Project. *Society of Actuaries*.

— (2005c). Surrender Rate Impacts on Asset Liability Management. *Asia-Pacific Journal of Risk and Insurance* 1.1. DOI: 10.2202/2153-3792.1004.

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, pp. 489–508. DOI: 10.1111/1539-6975.t01-1-00061.

Lemmens, A. and Gupta, S. (2017). Managing Churn to Maximize Profits. SSRN Scholarly Paper ID 2964906. Rochester, NY: Social Science Research Network.

Lemmens, A. and Croux, C. (2006). Bagging and Boosting Classification Trees to Predict Churn. *Journal of Marketing Research* 43.2, pp. 276–286.

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, pp. 348–357. DOI: 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 (Formely : 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, pp. 373–398. DOI: 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, pp. 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, pp. 370–384. DOI: 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, pp. 204–211. DOI: 10.1509/jmkr.43.2.204.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology* 24.12, pp. 1565–1567. DOI: 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, pp. 983–1002. DOI: 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. *Internation Journal of Machine Learning Technology* 2.1, pp. 37–63.

Renshaw, A. E. and Haberman, S. (1986). Statistical analysis of life assurance lapses. *Journal of the Institute of Actuaries* 113.3, pp. 459–497. DOI: 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, pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

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, pp. 909–931. DOI: 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, pp. 1–9. DOI: 10.1016/j.simpat.2015.03.003.

Wainer, J. (2016). Comparison of 14 different families of classification algorithms on 115 binary datasets. *Working paper*. arXiv: 1606.00930.

Xia, G.-e. and Jin, W.-d. (2008). Model of Customer Churn Prediction on Support Vector Machine. *Systems Engineering - Theory & Practice* 28.1, pp. 71–77. DOI: 10.1016/S1874-8651(09)60003-X.

Zhao, Y., Li, B., Li, X., Liu, W., and Ren, S. (2005). Customer Churn Prediction Using Improved One-Class Support Vector Machine. *Advanced Data Mining and Applications*. Ed. by Li, X., Wang, S., and Dong, Z. Y. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 300–306.

Appendix 1 XGBoost Tuning - Binary Classification

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

- *eta*: 0.005, 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 2 SVM Tunning

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 previous section, 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 3 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_weight*: 15;
- *subsample*: 0.7;

- *colsample_bytree*: 0.8.

Then, we determine the best *nrounds* through a 5-folds cross-validation with this parameter tested up to 1,000.