A Machine Learning Model for Lapse Prediction in Life Insurance Contracts

Michele Azzone[‡+] Emilio Barucci[‡−] Giancarlo Giuffra Moncayo[‡⋄]
Daniele Marazzina[‡∗]

April 11, 2021

(‡) Politecnico di Milano, Department of Mathematics, 32 p.zza L. da Vinci, Milano, Italy
(∗) Corresponding author, E-mail daniele.marazzina@polimi.it
(+) E-mail michele.azzone@polimi.it
(−) E-mail emilio.barucci@polimi.it
(⋄) E-mail giancarlo.giuffra@polimi.it

Abstract

We use the Random Forest methodology to predict the lapse decision of life insurance contracts by policyholders. The methodology outperforms the logistic model, even if features interactions are considered. We use global and local interpretability tools to investigate how the model works. We show that non economic features (the time passed from the incipit of the contract and the time to expiry, as well as the insurance company) play a significant effect in determining the lapse decision while economic/financial features (except the disposable income growth rate) play a limited effect. The analysis shows that linear models, such as the logistic model, are not adequate to capture the heterogeneity of financial decisions.

Keywords: machine learning, life insurance, interpretability, lapse

1 Introduction

The analysis of financial decisions is an active field of research. Classical finance theory mostly provides normative insights answering questions like "what is the optimal portfolio/insurance coverage for a person facing a risk represented by a random variable?" In this paper, we deal with the decision to lapse a life insurance contract by a policyholder. The decision is investigated through Machine Learning (ML) techniques that allow us to consider a large set of explanatory variables (and their non linear interactions) going beyond linear models and classical financial decision theories.

The key reference in financial decision making is provided by the hypotheses of expected utility theory and full rationality: the person exactly knows the probability space around him and maximizes the expected utility of wealth. Empirical research on financial decision making has shown that people do not follow this approach. On one hand, the expected utility framework is too narrow as financial decisions are traced back to a limited set of factors including wealth, risk aversion, discount factor, and risk. The empirical literature has shown that other features of decision-makers affect their portfolio/insurance decisions. Among them, we have education, with well-educated people being more active in financial and insurance markets, wage, wealth, and age, with middle age people being more active. On the other hand, it turns out that psychological attitudes affect the behavior of investors. We are referring to the Behavioral Finance literature, which shows that people take decisions considering a reference status or referring to meta-goals. These considerations introduce a significant degree of heterogeneity in financial decisions and weak normative insights: in short, we cannot find a strong connection between a feature of an agent (e.g., wealth) and the demand for insurance or investment.

The empirical literature on lapses of insurance contracts exploiting classical techniques is quite large. Three main hypotheses on the decision to withdraw from a contract have been investigated in the literature, see Bauer *et al.* (2017), Eling and Kochanski (2013). The Interest Rate Hypothesis (IRH) relates the lapse decision to financial market conditions and to the value of the contract: the policyholder withdraws from the contract in response to changes in market interest rates, and, in particular, the lapse rate should positively depend on the interest rate because its increase renders other safe options more interesting and premiums of new policies cheaper. This hypothesis is integrated with the hypothesis that the decision to lapse is driven by the possibility to replace the contract with another contract at better conditions (Policy Replacement Hypothesis, PRH). Economic and liquidity conditions may also affect the decision of a policyholder to withdraw from a life contract. According to the Emergency Fund Hypothesis (EFH), a policyholder gives up the life insurance contract when she is facing a difficult personal economic/financial situation (unemployment, health/long term care problems).

In this perspective, ML tools (such as the Random Forest methodology employed in this paper) provide an interesting approach. The appealing feature is that they are model-free, they allow to consider a large set of variables (and their non linear interactions) and therefore let the data speak for themselves. These techniques are deeply used in different fields, including finance. Just to mention few examples, ML techniques, and, in particular, Random Forest (RF), are used in credit scoring and prediction of mortgage default (Bemš *et al.* (2015), Malekipirbazari and Aksakalli (2015), Barboza *et al.* (2017), Moscatelli *et al.*

(2020)) as well as of market movements (Krauss *et al.* (2017), Zhang *et al.* (2018), Fischer and Krauss (2018), Cui *et al.* (2020)). In these fields, ML tools have been used to cope with the heterogeneity and complexity of financial phenomena: the above papers show that classification tools such as RF provide an advancement over more traditional tools, e.g., logistic regression.

As in large part of applications of ML to finance, we show that RF outperforms logistic regression in the prediction of the lapse event, even if interactions between features are taken into account. We then concentrate on the explainability of ML outcomes. To overcome the black-box approach, explainability issues are becoming more and more important in ML applications to finance shading light on the structure of the model and on input-output responses, see Borgonovo and Plischke (2016), Antoniano-Villalobos *et al.* (2020). According to our analysis, important drivers of the lapse decision are the time passed from the incipit of the contract and the time to expiry, the contract size, and premium, as well as the insurance company, e.g., launching of new products or the frequency with which products are proposed to the customers. Gender, age of the policyholder, and region in which the policyholder lives do not play an important role in the lapse decision. These results are interesting because all the above variables turned out to be statistically significant in Barucci *et al.* (2020), where the same dataset is analyzed considering classical linear models, with the purpose not to predict but to analyze the lapse rates with respect to major factors of lapses. They confirm that linear models have a weak capability to capture the heterogeneity of financial decisions and that ML techniques may provide a significant improvement.

This paper contributes to the application of ML techniques to the insurance market. As far as we know, there are few examples of applications in this environment. Guelman *et al.* (2012, 2014a,b, 2015) and Guelman and Guillen (2014) employ uplift tree to evaluate marketing activity for customer retention, while Jeong *et al.* (2018) apply the method of association rule learning to investigate the association between policyholder's switching after a claim (motor insurance) and the change in premium. The articles closest to our are Milhaud *et al.* (2011), Lally and Hartman (2016), Babaoglu *et al.* (2017). In particular, Milhaud *et al.* (2011) analyze the surrender of life insurance policies considering a dataset from the Spanish market, as in our analysis they consider logistic regression as well as RF. They focus on testing different policy and policyholder's features as lapse triggers, the goal is to segment the portfolio of policyholders in classes regarding the surrender risk. They find that old age, a periodical premium, a low income, having finished the tax benefit period, and possessing a profit benefit option contribute positively to the risk of surrendering the policy. In contrast, the gender of the policyholder is not a discriminant. Regarding the models, the authors compare logistic regression, Classification and Regression Tree (CART), and RF obtaining the best results for the latter. Similarly, in Babaoglu *et al.* (2017) authors deal with a ten years dataset of life insurance contracts, and compare logistic regression, naive Bayes classifier and RF, showing that the latter outperforms the others in predicting lapse decisions. Finally, Lally and Hartman (2016) report the inadequacy of the Poisson regression model both for mortality and lapse data. As alternatives to the Poisson model, the authors consider several Generalized Linear Models (GLMs) and the RF. They find that Tweedie GLM and RF provide the best results in terms of average predictive Accuracy. Lally and Hartman (2016) prefer GLMs to the RF because it is possible to identify the contribution of the different

variables (features). Unfortunately, the gain in interpretability comes at the expense of performance. One of our contribution is to tackle the interpretability problem of the RF, investigating the decision drivers in our ML lapse rate model exploiting both local and global interpretability techniques.

The paper is organized as follows. In Section 2 we describe the dataset. The methodology is presented in Section 3, where we address three crucial ingredients of our analysis: the models used to investigate the lapse phenomenon, the performance indicators used to select the model, and the methodologies adopted to provide an interpretation of the models. In Section 4 we report the results of our analysis, while Sections 5-6 concentrate on the interpretability of the RF model and on its robustness, respectively. Section 7 concludes the paper.

2 The dataset

The dataset considered in our analysis comes from one of the Italian largest insurance companies: it covers the $2008 - 2016$ time interval with over one million life insurance contracts. A piece of information is collected for each customer detaining an insurance policy, that is a policyholder/contract belongs to the dataset since the subscription of the contract and remains in the dataset up to the year in which the policyholder decides to lapse the contract or the contract is terminated for other reasons (death or expiration of the contract). Insurance contracts in the dataset belong to two broad families: 60% are standard contracts with a guarantee (non-negative minimum guarantee rate) and 40% are unit-linked contracts. The contracts refer to three different companies belonging to the same insurance group. As the companies refer to different distributors, either bank assurance relationship or financial advisors, the lapse phenomenon is company-specific because the lapse decision may be driven by commercial policies of distributors (e.g., the launch of new products). This dataset was analyzed through classical techniques in Barucci *et al.* (2020). The lapse rate per year (lapses over the number of contracts) exhibits an increasing trend over the sample with an outlier (peak) in 2011 due to significant lapses for Company A. We refer to Barucci *et al.* (2020) for a complete statistical analysis of the dataset.
The dataset contains many interesting pieces of information about each contract/policyholder. The full set of exogenous variables is made up of the following information, for an analysis of the literature on the impact of these features on the lapse rate we refer again to Barucci *et al.* (2020):

- **Gender** : policyholder's gender (54% male and 46% female in the dataset).
- **Age**: age of the policyholder. We compute it as the policyholder's age at the beginning of the contract plus the time passed from the beginning of the contract.
- **Region**: one of the 4 macro-regions where the policyholder lives, North-West (Region=1), North-East (2), Center (3), and South & Islands (4).
- **Company**: the insurance company issuing the policy. The parent company develops its business through three companies. We identify them as company A (COMP=1), B (2), and C (3).

- **Time from start**: the time passed from the incipit of the contract.
- **Time to expiry**: the time left (in years) to the expiry date of the contract. We also have perpetual contracts with no expiration date (in this case we set the value of the variable to -1 in RF analysis while we consider a dummy variable in case of a logistic model to capture a perpetual contract).
- **Contract size**: the insured capital.
- **Premium**: policy premium. This variable is given by the contract size if there is only one payment and by the contract size divided by the number of payments if there are multiple payments.
- **Product type**: the dataset contains several different types of policies, we categorize them into two classes: traditional contracts (PT=0) and unit-linked (1).

We also consider macroeconomic variables that may drive the lapse decision.

- **Disposable income**: the yearly growth rate of Italian disposable income.
- **Inflation**: the yearly Italian inflation rate.
- **Eurostoxx**: the yearly growth rate of the Eurostoxx index.
- **Interest rate**: twelve months BOT (Italian zero coupon bond) rate.

For each contract, we have the following information: starting date, expiry date, and the eventual lapse date. For each contract, we have an observation for each year in which the contract is active (i.e., it is started and not expired). Our dataset contains 9 309 755 observations.

3 The methodology

In this section, we address three ingredients of our analysis: the models used to investigate the lapse phenomenon, the performance indicators used to select the model, and the methodologies adopted to provide an interpretation to the models.

3.1 The models

We compare the ability of two classification models for the lapse decision: a logistic regression (considering regularization and interaction among features) and a RF model.

3.1.1 Logistic regression

The logistic regression is the most common technique for classification problems, see Kleinbaum *et al.* (2002). The model is a generalized linear model with a logistic link function. Consider a binary random variable Y (in our case the lapse random variable: $Y = 0$ no lapse, 1 lapse) and X the input vector of

random variables (the features of each contract/policyholder as well as the macroeconomic variables). The logistic regression for the probability to lapse ($Y = 1$) is modeled as follows

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-(a+B^\top x)-\varepsilon}} \ ,$$

where $a \in \mathbb{R}$, B is a vector with size as the vector X, and ε is the error. It is possible to connect this equation to the classical linear regression model observing that the model aims to estimate the logarithm of the odds ratio:

$$\log\left(\frac{P(Y = 1|X = x)}{1 - P(Y = 1|X = x)}\right) = a + B^\top x + \varepsilon.$$

As is standard in the literature, in the logistic regression case, we substitute categorical variables with the corresponding dummy variables. To avoid collinearity effects the logistic regression model is estimated maximizing the log-likelihood with an l1 (Lasso) or an l2 (Ridge) penalization, see, e.g., Melkumova and Shatskikh (2017).

Further, we take into account also the interactions between features by including in the model the pairwise products of original features. More precisely, we consider both the logistic regression model with interactions among all the features, as well as the model with interaction among the six most important features. The ranking of the original features is obtained using the Recursive Feature Elimination (RFE) algorithm, see, e.g., Kuhn and Johnson (2013).

Summarizing, we deal with the following logistic regression models:

1. Ridge and Lasso standard logistic regression;
2. Ridge and Lasso logistic regression with interaction between the six most important features;
3. Ridge and Lasso logistic regression with interaction between all the features.

3.1.2 Random Forest

Given the binary nature of the target variable, our RF classifier is based on a combination of classification trees and of tree bagging methods (see, e.g., Liaw *et al.* 2002). Classification trees are recursive algorithms that split the dataset into smaller sets (*nodes*) in a step by step fashion thanks to binary rules defined on the features of the observations. The complete dataset is the root node of the tree. To split the node, a feature is selected and a binary rule, e.g., if the value of the feature is larger or smaller than a given threshold, is defined on it in order to obtain two disjoint datasets. These two datasets become nodes of the tree. This procedure is repeated for each new node until a stopping criterion is met, e.g., the specified maximum depth of the tree is reached. So a classification tree is a growing tree with nodes refining the information about the exogenous variables to classify an item in the database (lapse or no lapse).

Each node is associated with a measure of *Impurity*. Such a measure is high when the dataset representing the node contains observations belonging to different classes of the endogenous variable (0 or 1), and reaches its lowest value (zero) when the node only contains observations belonging to a single class. In our setting,

as we have only two classes, we use the *Gini impurity*. Given a node x, the Gini impurity, denoted by $G(x)$, for a binary classifier is defined as

$$G(x) = p_{lapse}(1 - p_{lapse}),$$

where p_{lapse} denotes the proportion of lapse observations contained in node x. p_{lapse} can be interpreted as the probability of lapse in the node. We aim to minimize the Gini index; its maximum (1/4) corresponds to the case in which a node is perfectly balanced (50% of the observations in the node are lapses and 50% non lapses, and therefore $p_{lapse} = 1/2$).

The feature (exogenous variable) and the binary rule that are used for the split at a node are defined through a search process that maximizes the impurity decrease after the split. Given a split rule r that divides node x (observations of the dataset) into the child nodes x_L and x_R, we define the impurity decrease after the split, $\delta(r, x)$, as

$$\delta(r, x) = G(x) - p_L G(x_L) - p_R G(x_R) \ , \tag{1}$$

where p_L (p_R) is the percentage of observations in x that after the split belong to node x_L (x_R). The last nodes of the tree (the arrival point of the algorithm, i.e., the nodes that are not split anymore) are called *leaves*. The leaves contain the estimation of the tree (in our case lapse or no lapse decision), which is given by the most recurrent class in that leaf: if more than 50% of the observations in the leaf are lapses, then the leaf is a lapse leaf, otherwise, it is a non-lapse leaf. We also get a probability of lapse for each leaf as the proportion of lapse observations in that leaf.

Bagging (Bootstrap Aggregation) is used to reduce the variance of classification trees. The methodology builds several datasets from the training set choosing randomly observations with re-entry from the original dataset (statistical bootstrap). Each dataset is used to train a tree. As a result, we obtain an ensemble of trees-models and the final classifier is obtained averaging the predictions from the different trees yielding a more robust outcome than the one obtained from a single tree.

A RF classifier is an extension of the bagging methodology. The approach considers random subsets of observations (each observation consisting of the thirteen features detailed in Section 2) from the original dataset and it also randomly selects the features rather than using all the features to construct each tree. The output of the RF classifier is the probability of observing a lapse. This probability is computed averaging the predictions of all the trees, i.e., the probability of lapse is the average of the lapse probabilities provided by all the classification trees in the RF.

To calibrate the parameters of the RF we have to set a priori some hyper-parameters. The parameters of the model are then calibrated on the training set. The hyper-parameters are chosen by evaluating the performance of the calibrated models on the validation set. In our analysis we consider the following hyper-parameters:

- **maximum tree depth**: the maximum depth for a tree;
- **minimum leaf size**: the minimum number of observations contained in a leaf. A split will only be considered if the number of observations belonging to each child node will be higher than the threshold;

- **minimum decrease of impurity after a split**: a node is split if it generates an impurity decrease greater than or equal to the threshold;
- **minimum split size**: the minimum number of observations belonging to a node required to split it;
- **number of trees**: the number of trees in the forest.

3.2 Model selection

In the case of linear regression, the performance of the model can be evaluated by means of Mean Square Error (MSE) or mean absolute percentage error. These metrics cannot be used in the case of statistical methods dealing with a discrete endogenous random variable. Usually, the output of a classifier (e.g., RF or logistic regression) is a vector of length equal to the number of classes of the endogenous variable, where each component corresponds to an estimate of the probability of belonging to that class: in our case, as we are dealing with two classes, the vector consists of two elements, the probability of observing a lapsed contract and the probability of observing a non-lapsed contract. In this context, the metrics used to evaluate the performance of an algorithm are often defined through a threshold value: if the class probability of an observation is greater than the threshold, then the observation is marked as positive (value 1 of the binary dependent variable). By varying this threshold value, the metrics give a complete overview of the classifier's performance.

In our analysis, we consider four performance metrics: Accuracy, Area Under Curve (AUC), Area Under Precision Recall curve (AUPR), Logloss.

3.2.1 Accuracy

The *Accuracy* of a classifier is the percentage of correctly classified items:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \ ,$$

where TP is the number of true positives (the algorithm output is 1 and the observation for the endogenous variable is 1) and FN is the number of false negatives (the algorithm output is 0 and the observation for the endogenous variable is 1); analogously, FP and TN stand for false positives and true negatives, respectively. This performance measure does not distinguish between the classifier's predictive power applied to 0 and 1 observations. In particular, Accuracy can be misleading in the case of an imbalanced dataset where there is a significant disparity between the number of 0 and 1 observations. For example, if the percentage of 0 observations is extremely high, then it is possible to obtain a high Accuracy classifying all observations as 0.

As the output of the logistic regression and of the RF is the probability to observe a lapsed contract, Accuracy is computed considering a threshold value. In our analysis, we will consider two thresholds: 50%, which is equivalent to classify an observation according to the highest class probability (e.g., if the classifier output is smaller or equal than 0.5 then the observation belongs to class 0, otherwise to class 1) and 75% (e.g., if the classifier output is smaller or equal to 0.75 then the observation belongs to class 0, otherwise

to class 1). In the case of the RF, the output is 1 if this outcome is obtained for more than 50% or 75% of the trees of the forest. The latter threshold can be employed if we want to handle FN and FP in a different way: adopting a 75% threshold we diminish the possibility to find a FP (the algorithm returns 1 only if the computed probability of having a lapse is greater than 75%), at the cost of increasing the number of FN. This approach can be adopted for commercial reasons: the insurance company contacts a policyholder to prevent his/her lapse only in case the probability of having a lapse, according to the algorithm, is high enough (higher than 75%).

3.2.2 Receiver Operating Characteristic and Area Under Curve

The true positive rate (TPR) is the percentage of positive observations correctly classified, while the false positive rate (FPR) is the percentage of negative observations classified as positive:

$$TPR = \frac{TP}{TP + FN} , \qquad FPR = \frac{FP}{FP + TN} .$$

These ratios depend on the threshold used to define the classification outcome of the RF algorithm. The *Receiver Operating Characteristic* (ROC) curve is built plotting the false positive rate (FPR) against the true positive rate (TPR), moving the threshold from 0 to 1. The ROC curve illustrates the predictive ability of a binary classifier, visualizing the trade-off between TPR and FPR, and thus suggesting an optimal threshold that minimizes the misclassification error. For example, when the threshold is 0, every observation is marked as positive. This setting corresponds to the point $(1, 1)$ in the ROC curve ($TN = FN = 0$). The optimal classifier, i.e., the one which provides exact predictions, corresponds to the point $(0, 1)$, i.e., $FPR = 0,\ TPR = 1$ (the classifier has neither FP nor FN). Therefore, the ROC curve of a good classifier is close to the top left of the graph, i.e., point $(0, 1)$, allowing to achieve high TPR and low FPR (at an optimal threshold).

The *Area Under Curve* (AUC) is the integral of the ROC curve. The AUC of the random classifier (a classifier that randomly classifies each item) is 0.5 while the AUC of the perfect classifier (a classifier that correctly classifies each observation) is 1. The higher is the AUC, the better is the performance of the classifier. Notice that the AUC overcomes the problem of selecting a threshold when evaluating a classifier but it could be misleading in the case of an imbalanced dataset where the difference between good and bad predictors are less significant (see, e.g., Saito and Rehmsmeier 2015).

3.2.3 Area under Precision Recall curve

The *Precision Recall* (PR) curve is obtained by plotting the *Precision* (the percentage of observations classified as positive that are indeed positive)

$$Precision = \frac{TP}{TP + FP}$$

against the *Recall* (the percentage of positive observations classified as positive)

$$Recall = \frac{TP}{TP + FN}$$

as the threshold defining the classifier output varies from 0 to 1. The *Area Under PR curve* (AUPR) is the integral below the PR curve. The AUPR of the random classifier is equal to the percentage of positive observations, i.e., observations belonging to class 1 in the dataset (in our case 17%), while for the perfect classifier the AUPR yields 1. The higher is the AUPR index, the better is the performance of the classifier. This measure of performance is recommended for imbalanced datasets (see, e.g., Davis and Goadrich 2006).

3.2.4 LogLoss

The LogLoss is the loss function used to evaluate the performance of a logistic regression. It is based on the cross-entropy and it is defined as

$$LogLoss = -\sum_{n=1}^{N} y_n \log(p_n) + (1 - y_n) \log(1 - p_n) \ ,$$

where N is the number of observations, y_n is 1 if the n-th observation corresponds to a lapse and 0 otherwise, and p_n is the probability that the n-th observation belongs to class 1 according to the classifier. The LogLoss builds on the class probabilities estimated by the classifier: the lower is the LogLoss, the better is the performance of the algorithm. As it does not take values in a bounded range, the LogLoss can only be used in relative terms to compare two different models and not as an absolute measure of the goodness of a classifier. Notice that it is not possible to define, a priori, the log-loss of the random classifier.

3.3 Random forest explainability

The main limit of ML tools is that they appear as "black boxes" that are difficult to be interpreted in terms of causal relationships between input and output variables. Several methodologies have been proposed in the literature to cope with the explainability/interpretability of the ML models and of the results. In this paper, we use a "local" approach proposed by Lundberg and Lee (2017) and a simple "global" approach.

Given an observation (in our case the features of an insurance policy/policyholder, the macroeconomic factors and the endogenous variable), by a local approach, we mean an analysis of how the RF classifier behaves for small perturbations of the observation's features. Looking at a specific observation and its prediction, we analyze how the outcome of the RF classifier depends linearly on some features, rather than having a complex dependence on them.

By a global approach, we mean that we train a simple and easily interpretable model (in our case a single regression tree) on the RF predictions, that is we connect the features of an observation to its RF estimated output, and we use this model to understand how the "black box" algorithm works.

3.3.1 Local Explainability (SHAP values)

The SHAP (SHapley Additive exPlanations) method allows us to capture the impact of the different variables/features on the ML classifier output, see Lundberg and Lee (2017). The method borrows from cooperative game theory and consists in the calculation of SHAP value, which represents a measure of the

importance of a feature. More precisely, the SHAP value of a feature measures how much it contributes, either positively or negatively, to the classifier prediction.

The goal of the SHAP method is to explain a prediction computing the contribution of each feature to the prediction itself. More precisely, the method shows the contribution of each feature to push the model output from the base value (the average model output over the training dataset) to the model output associated with the observation. Given a single observation, a set of SHAP values, one for each feature, is calculated. Notice that SHAP values are additive: their sum is equal to the difference between the predicted class probability and the class base value.

The SHAP method is characterized by several interesting features: it is an additive feature attribution method, i.e.,it can be written as a linear function of binary variables (Lundberg and Lee 2017), it mimics the evaluation procedure of people as shown in several experiments in Lundberg and Lee (2017), Lundberg *et al.* (2020), its implementation for tree-based algorithms is efficient, see Lundberg *et al.* (2020).

3.3.2 Global Explainability

"Local" methods are useful to investigate the results connected to a specific observation. In some applications, it is interesting to have a global view of the model's behavior. To accomplish this task, we fit a single regression tree on the predicted probabilities of our RF. Regression trees have basically the same structure as classification trees, but the dependent variable is a continuous variable (e.g, the probability to lapse), and the measure of impurity is the MSE of the observations in the node. The quantity predicted by each leaf is the average of the values of the target variable of the observations in the leaf. In our case, the idea is to estimate a regression tree, with low depth, in such a way that its output is as close as possible to our "black box" output (the outcome of the RF). The scope is to produce a human-readable tree that can represent the main drivers of the "black box" classification. The approach is close to the Local Interpretable Model-agnostic Explanations (LIME) method presented in Ribeiro *et al.* (2016), which is used in a local explainability perspective. We extend it to develop a global explainability approach.

4 Analysis

ML models are prone to overfitting. In the case of the RF classifier, for example, it is possible to select the hyper-parameters in a way that each observation of the training set is correctly classified. To address this issue, it is a standard practice to randomly split the dataset into three sets: training, validation and test sets (see, e.g., James *et al.* 2013, Chapter 5).

The training set is used to estimate the model parameters. If the model presents additional hyper-parameters, as in the case of a RF or the penalization term λ of the Ridge (Lasso) logistic regression, the validation set is used to determine the optimal hyper-parameters: given a set of hyper-parameters, the model is calibrated on the training set, its performance is evaluated on the validation set. Then the set of hyper-parameters is chosen according to a performance metric evaluated on the validation set. Finally, the test set is used to evaluate the performance of the ML algorithm out of sample.

	Accuracy (50%)	AUC	AUPR	LogLoss
Ridge	82.8152 %	69.7749 %	31.0182 %	42.6375 %
Lasso	82.8170 %	69.7734 %	31.0302 %	42.6366 %
Interactions(6) + Ridge	85.2556 %	78.4835 %	50.3491 %	37.3262 %
Interactions(6) + Lasso	85.2587 %	78.4810 %	50.3536 %	37.3263 %
Interactions(All) + Ridge	85.2525 %	79.4836 %	53.3510 %	36.3531 %
Interactions(All) + Lasso	85.4682 %	79.7551 %	53.4549 %	36.3262 %

Table 1: Performance metrics for the different logistic regressions on the test dataset.

We split our dataset as follows: training set (70%), validation set (15%), test set (15%). The split is randomized and the seed of the random number generator is fixed for reproducibility. In the analysis, as is standard in ML applications, we standardize each feature via the min-max normalization procedure (see, e.g., Patro and Sahu 2015).

4.1 Logistic regression

We report in Table 1 the performance metrics defined in Section 3.2 for the different logistic regressions on the test dataset. We notice that Ridge and Lasso logistic regressions show comparable performances, and a relevant improvement (specifically in term of AUC and AUPR) considering the interactions between the six most important features (disposable income, time to expiry, time from start, company, inflation and Eurostoxx) with respect to the standard logistic regression case. By considering the interactions between all variables, we get only an additional 3% improvement in terms of AUPR. Unfortunately in this case, the computational cost explodes (more than ten times the case with the interaction among the six most important features) and it is difficult to understand the contributions of the different features (due to the high number of estimated coefficients). For this reason, we opt for the logistic regression model with interactions among the six most important features and a Lasso regularization[1]. Therefore from now on we will refer to this logistic regression model.

In Table 2 we report the performance metrics on the training set, on the validation set, and on the test set for the selected logistic regression. The metrics show that the logistic model outperforms the random classifier both in terms of AUC and AUPR. We notice that there are only negligible differences in the metrics on the training and on the test set. This result indicates that logistic regression is not affected by overfitting even if the interactions between features are considered.

In Figure 1.(left) we plot the PR curve for the logistic regression on the training set and on the test set. Notice that the two curves overlap showing again that there is no evidence of overfitting: the performance of the model on the training set is similar to that obtained on the test set. For recall levels greater than 50% (the percentage of positive observations classified as positive), we observe that the precision of the logistic classifier (the percentage of observations classified as positive that are indeed positive) is always

[1]We do not observe a significant difference among Ridge and Lasso regularization.

	Training	Validation	Test
Accuracy (50%)	85.3306 %	85.2877 %	85.2587 %
Accuracy (75%)	84.1485 %	84.1071 %	84.0773 %
AUC	78.4298 %	78.4195 %	78.4810 %
AUPR	50.2855 %	50.2592 %	50.3536 %
LogLoss	37.2635 %	37.3102 %	37.3263 %

Table 2: Performance metrics for the logistic regression on the training, validation and test dataset.

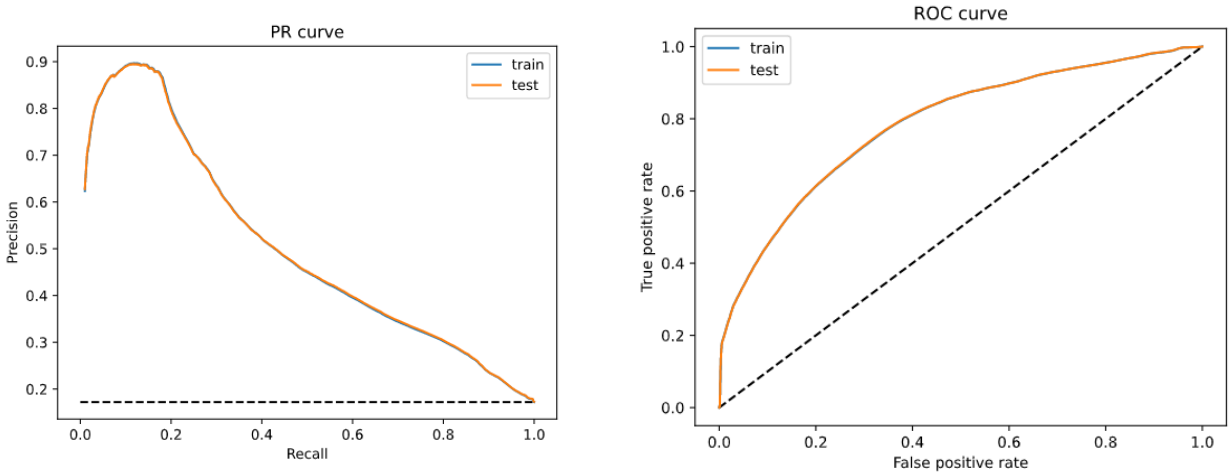


Figure 1: PR and ROC curve for the logistic regression on the training set and on the test set. The dashed line corresponds to the random classifier. Notice that the train and the test curves overlap.

lower than 45%. These results are quite unsatisfactory: in order to correctly classify at least 50% of the lapse observations (Recall greater than 0.5), less than 45% of the observations classified as lapses are true positive (TP), i.e., precision smaller than 45%, and therefore more than 55% are false positive (FP).

To conclude, in Figure 1.(right) we plot the ROC for the logistic regression on the training set and on the test set. As above, the two curves overlap confirming that there is no evidence of overfitting. Notice that for false positive rates smaller than 20%, the true positive rates are smaller than 60%, i.e., to have less than 20% of non lapse events classified as lapses, we have to set a threshold that classifies less than 60% of true lapses as lapses.

4.2 Random Forest

The selection of the RF works as follows. The RF model is trained on the training set for different sets of hyper-parameters. We have to select the best combination of hyper-parameters of the model. To this end, we perform a discrete grid search in the hyper-parameter space as defined in Table 3. For each combination of hyper-parameters, we train a RF model on the training set and we evaluate its performance

Maximum tree depth	5, 10, **50**, 100
Minimum leaf size	5, **10**, 50
Minimum decrease of impurity after a split	**0**, 0.1, 0.2, 0.3
Minimum split size	5, **10**, 50, 100
Number of trees	1, 2, 5, 10, **50**, 100

Table 3: Hyper-parameters used in the grid search procedure

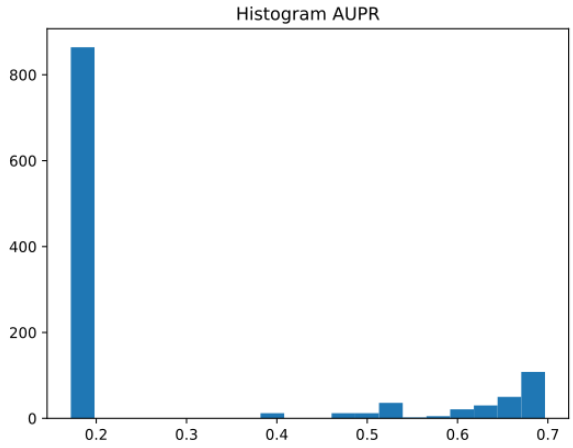


Figure 2: Histogram of AUPR for the sets of hyper-parameters on the grid search: number of combinations of hyper-parameters that fall into AUPR bins.

on the validation set. We select the best set of hyper-parameters evaluating the AUPR: in Figure 2 we show the histogram of the AUPRs on the validation set for all the RF models calibrated through the grid search. Notice that we have a cluster of sets of hyper-parameters of the RF model (with an AUPR slightly less than 70%) with a good performance on the validation set, the rightmost candle in Figure 2. From this cluster of RF models, we select the less complex one in terms of the number of trees and maximum tree depth. The selected hyper-parameters are: maximum tree depth 50, minimum leaf size 10, minimum decrease of impurity after a split 0, minimum split size 10, number of trees 50 (in bold in Table 3).

The RF model selected according to the above procedure is evaluated on the test set. In Table 4 we report the performance metrics for the selected RF model on the training set, validation set, and test set. Notice that there are only slight differences in the error metrics on the training and validation set showing that overfitting problems are not significant. Comparing Table 4 to Table 2, we observe that the RF model outperforms the logistic regression.

In Figure 3 we plot the PR and the ROC curves for the RF model on the training and on the test set. Notice that the performances on the two datasets are similar but, as expected, the performance on the training set is slightly better than the performance on the test set. In both cases, the model outperforms considerably the random classifier and the logistic regression, see Figure 1. Considering the test set, for

	Training	Validation	Test
Accuracy (50%)	89.0997 %	88.0406 %	88.0241 %
Accuracy (75%)	87.2702 %	87.0098 %	86.9367 %
AUC	92.4511 %	88.2419 %	88.3102 %
AUPR	76.2799 %	69.5422 %	69.7343 %
LogLoss	25.0665 %	28.7677 %	28.7936 %

Table 4: Performance metrics for the selected RF models on the training, validation and test sets.

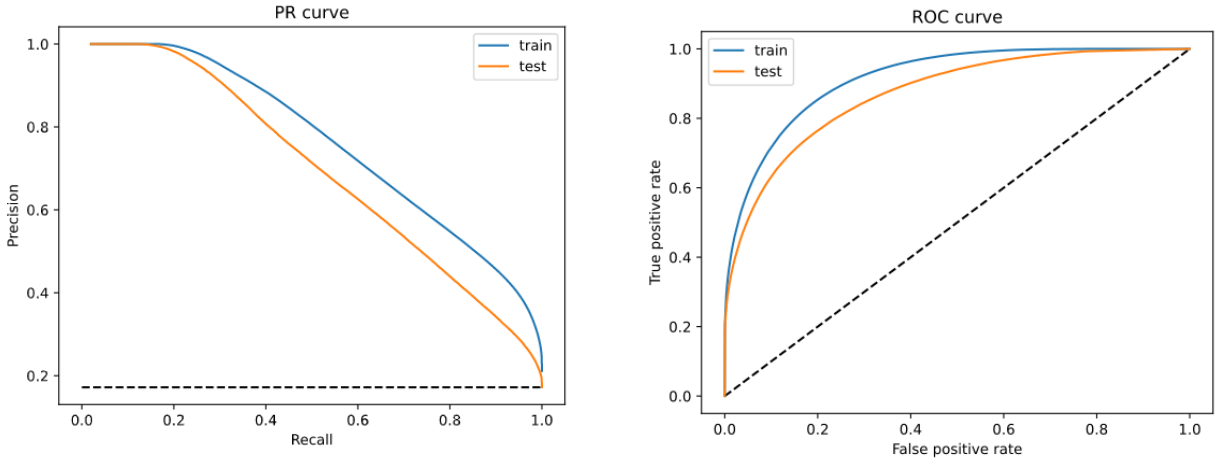


Figure 3: PR curve for the RF on the training and test set. The dashed line is the PR curve of the random classifier.

recall values equal to 50%, precision is nearly 75%, i.e., to correctly classify 50% of the observed lapses, nearly 75% of the observations classified as lapse are true positive (for logistic regression it was nearly 45%). On the other hand, for a false positive rate equal to 20%, the true positive rate of the model is nearly 80%, i.e., to have 20% of non lapse observations classified as lapses, nearly 80% of true lapses are classified as lapses (for logistic regression it was 60%).

5 Random Forest explainability

In this section we address the interpretability issue of the RF model estimated according to the above methodology. We first deal with the importance of the thirteen exogenous variables (features) presented in Section 2. Then we address global interpretability of the model. To conclude we deal with the local interpretability approach based on the SHAP values.

1	Company (COMP)	15.5361 %	8	Age	6.1452 %
2	Time from start	15.0721 %	9	Interest rate	4.9981 %
3	Time to expiry	11.5236 %	10	Region	4.5519 %
4	Premium (P)	9.7135 %	11	Inflation	4.3068 %
5	Contract Size (CS)	9.1357 %	12	Eurostoxx	3.7103 %
6	Product Type (PT)	7.7191 %	13	Gender (SEX)	0.5970 %
7	Disposable income	6.8855 %			

Table 5: Importance of the features in the RF model (percentage). Among brackets the abbreviation used in the plot of the tree.

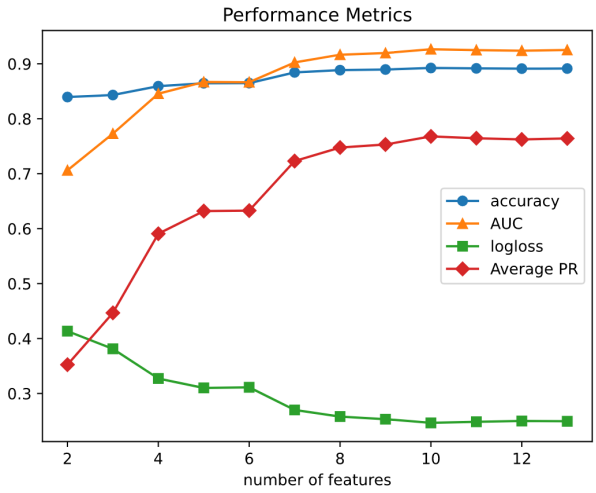


Figure 4: Random Forest performance on the training set as a function of the number of input features.

5.1 Features importance

In Table 5 we report the features ranked according to their impurity-based importance. The higher is the influence of a feature on the RF model output, the higher is its importance, see, e.g., Breiman (2001). All the variables (except Gender) give a non-negligible contribution to the RF model. According to our analysis, the most important drivers of the lapse decision are the insurance company, the time passed from the incipit of the contract, the time to expiry, and the contract size and premium.

In Figure 4 we plot the performance of the RF model on the training set according to various metrics as a function of the number of features included in the analysis. The order of the features inserted in the RF model is the one provided in Table 5. In the previous section, the model was trained using all the features, in Figure 4 the first n most important ones are selected and the performance metrics are computed considering the RF model estimated on the training set only considering the selected features. For example, if $n = 2$, we consider a training set having as features only the time from the start and the company, and as target-dependent variable the binary variable lapse/no-lapse. The performance improves

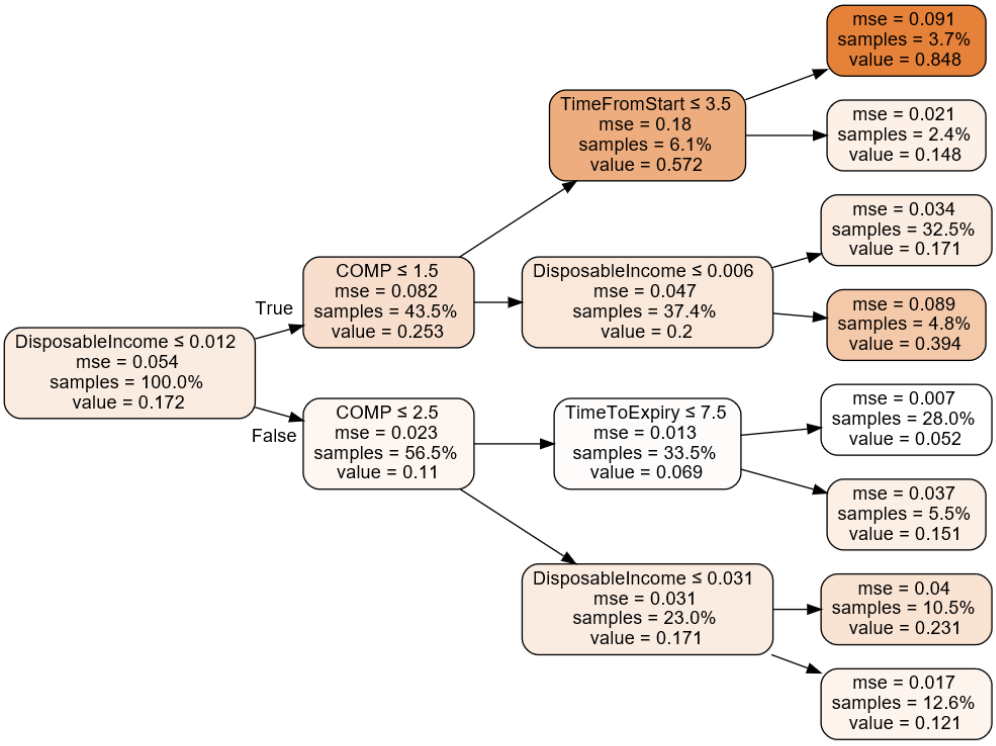


Figure 5: Regression tree fitted on the random forest output.

as the number of features increases, however, the rate of improvement rapidly decreases when more than eight features are inserted. We can assess that, in line with the results presented in Table 5, the first eight explanatory variables contribute substantially to the performance of the RF classifier, while adding the remaining variables we obtain little improvement. We also notice the important role of the seventh feature, the disposable income growth rate: adding this feature results in an important increase in the performance of the RF. This last result also puts forward the limits of the impurity-based importance (in Figure 4 the disposable income growth rate, ranked seventh, seems to play a more important role than product type, ranked sixth), suggesting the necessity of more advanced ML interpretability approaches.

5.2 Global interpretability

Given the hyper-parameters selected above, the RF model trained on the training set is applied to all the observations in the training set, storing the RF predictions, i.e., the estimated lapse probabilities. Then we look for the regression tree which better fits the RF, i.e., features as input and the RF predictions as output in the training set. We would like to stress that we are approximating a RF made up of fifty trees, each one with (maximum) depth 50, with a single regression tree with a much smaller depth.

In Figure 5 we provide the fitted regression tree assuming a depth of the tree equal to three; each node of the tree contains the following pieces of information:

1. the split rule that identifies the variable considered for the split;

2. the MSE that measures the pureness of the node, computing the distance between the estimated lapse probability of the node (the average of the lapse probabilities of all the observations belonging to the node) and the lapse probabilities of the observations contained in the node, i.e., if there are N observations belonging to a node, and p_i is the lapse probability of observation i, $i = 1, \cdots, N$, then we compute
$$\frac{1}{N}\sum_{i=1}^{N}\left(p_i - \frac{1}{N}\sum_{j=1}^{N} p_j\right)^2;$$

3. the percentage of observations contained in the node (samples);

4. the estimated lapse probability of the node (value).

We observe that this regression tree only uses four of the variables of the RF model. We notice that observations with disposable income growth rate smaller than 0.012 are more likely to lapse. We also notice the relevance of the Company (COMP) which acts at the second level of the tree. For example, if the disposable income growth rate is low and the policy is stipulated with Company A (COMP=1), then the probability to lapse increases considerably, reaching a 57% probability. We can conclude that, according to the selected RF model, these are the main drivers of the lapse decision among those identified in Table 5.

According to our analysis, the leaf with the highest lapse probability (84.8%) is obtained with a disposable income growth rate lower (or equal) than 0.012, a policy stipulated with Company A and with time from start smaller (or equal) than 3.5 years. On the other hand, the leaf with the smallest lapse probability (5.2%) is obtained with a disposable income growth rate higher than 0.012, a policy stipulated with Company A or B, and with time to expiry smaller (or equal) than 7.5 years.

Notice that the role of disposable income growth rate is controversial: while in the first node a low disposable income growth rate results in an increase of the probability to lapse (from 17.2% to 25.3%), the opposite happens in the second node of the third level. We can conclude that the contribution of the growth rate of the disposable income to the lapse decision is nonlinear and can not be fully captured through the coefficient of a linear model, as in the logistic regression. Notice that also Company A is crucial both in determining a lapse and a non-lapse event. We can conclude that the interactions among variables are truly nonlinear.

In Table 6, varying the depth of the regression tree, we report the Accuracy, which measures how well the single regression tree reproduces the RF output, as well as the number of variables that the algorithm selects to build the tree. More precisely, in this case, an observation is a TP if it is a lapse both for our RF model and for the regression tree, while it is a FP if it is a lapse for the regression tree but not for the RF, according to a threshold of 50%. As expected, by increasing the depth of the regression tree, both the Accuracy and the number of features increase. All the features included in the model are used if the depth of the tree is ten or higher. Therefore, we have a trade off between regression tree readability (lower

depth	2	3	4	5	6	10	25
Accuracy	25.57%	45.21%	55.93%	64.16%	73.34%	89.24%	99.41%
# Features	2	4	6	10	11	13	13

Table 6: Global interpretability decision tree varying its depth: Accuracy in reproducing the "black box" output, and number of features used to fit the random forest with a single decision tree.

is the depth, easier is the readability of the tree), and the ability of the tree to reproduce the "black-box" output.

Regression trees with depth five and ten are available in the online material. Even if the split selection relies partially on a random seed, we observe that the first three splits of these trees coincide with those in Figure 5. This result reinforces the importance of the selected variables and in particular of the role of the disposable income growth rate on the lapse decision, even if it ranks in 7^{th} position over thirteen features according to the impurity analysis (see Table 5) .

5.3 Local interpretability

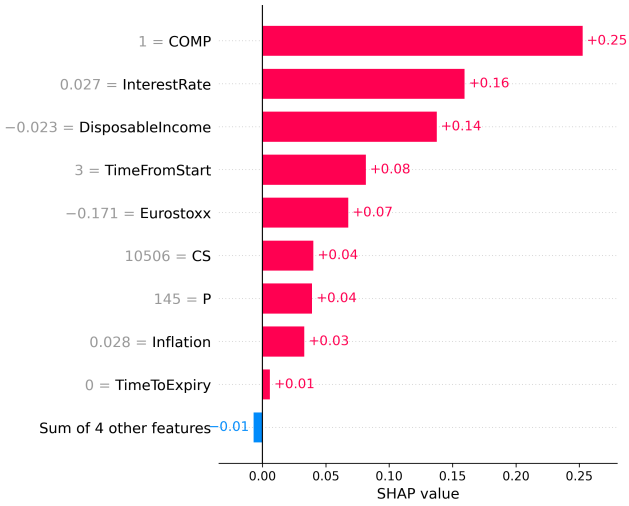
In Figure 6, we report the SHAP values for a true positive, a false positive, a true negative, and a false negative prediction. The values are calculated for the positive class (i.e., lapse class 1) and the observations are taken from the test set. For each observation, the SHAP values are represented graphically using the method introduced in Lundberg *et al.* (2018): we can visualize feature attributions obtained by Shapley values as "forces". Each feature value is a force that either increases or decreases the lapse prediction. The prediction starts from the baseline, i.e., the percentage of positive observations in the sample. As an example, in Figure 6(a) we have that the probability of a lapse of the observation is obtained by adding to the baseline probability, which is equal to the percentage of observations in the training set belonging to class 1 (17.2%), the following quantities:

- 0.25 due to the fact that the contract is proposed by company A (COMP=1);
- 0.16 associated with the fact that the risk free interest rate is 0.027;
- 0.14 associated with the fact that the disposable income growth rate is -0.023;
- 0.8 associated with the time from start;

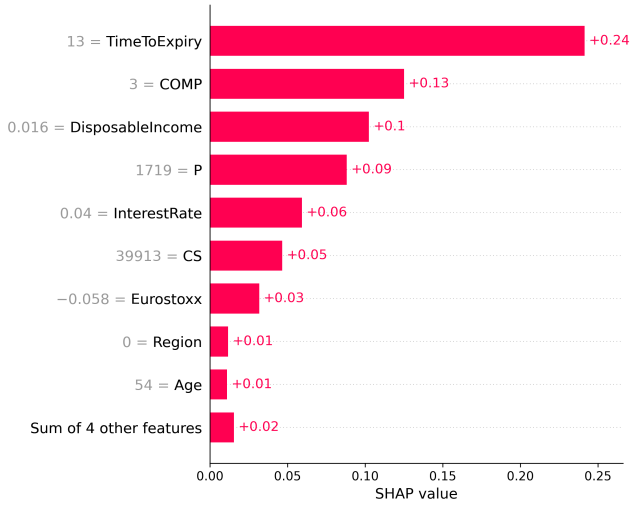
and so on. Each Shapley value pushes to increase (positive value) or decrease (negative value) the prediction from the baseline probability.

Results are coherent with those obtained through the global methodology. The variables used in the nodes of the regression tree in Figure 5 (Disposable Income, Time to Expiry, Time from Start, and Company) are among the variables with the largest (in modulus) SHAP value for these four observations.

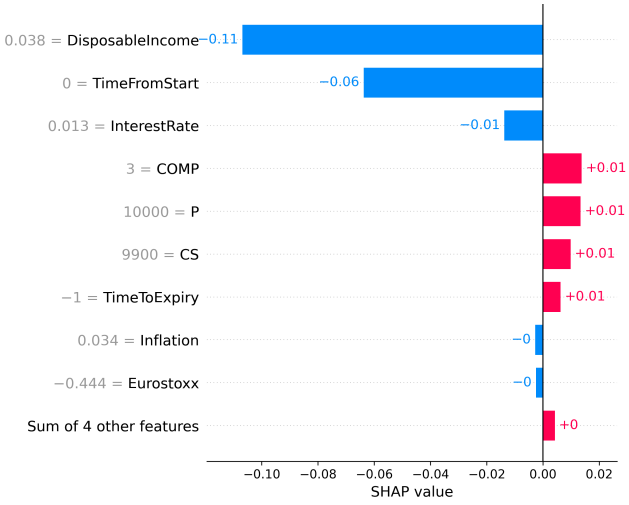
To further emphasize the coherence of the results with those obtained with the global methodology, we notice that the regression tree in Figure 5 would assign to the true positive observation in Figure 6.(a) the



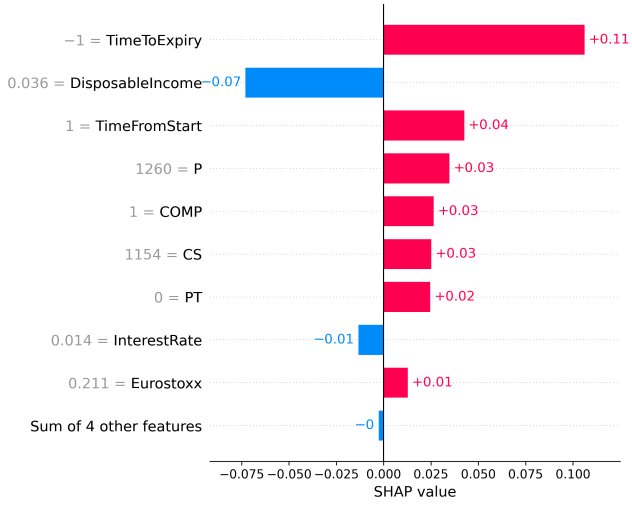
(a) true positive



(b) false positive



(c) true negative



(d) false negative

Figure 6: SHAP values of the four different observations (each feature is accompanied by the corresponding value assumed in the observation). We recall that P is the premium, CS is the contract size and PT is the product type.

leaf having the highest predicted value (the upmost leaf). The assignment would be done based on the variables Disposable Income, Company, and Time from Start, which represent respectively the third, first and fourth variable according to the modulus of the SHAP value for this particular observation. Instead, for the true negative observation, the regression tree would assign a 12.1% lapse probability.

	Lapse rate	Training		Test	
		AUC	AUPR	AUC	AUPR
COMPLETE	17.2094 %	92.4511 %	76.2799 %	88.3102 %	69.7343 %
COMP=A	20.6835 %	98.8603 %	96.5143 %	97.8568 %	94.5728 %
COMP=B	11.8177 %	90.9526 %	62.5063 %	84.8508 %	53.5381 %
COMP=C	20.4994 %	88.8943 %	68.7995 %	82.7440 %	58.5213 %
NO MACROS	17.2094 %	89.8872 %	69.1204 %	84.1821 %	59.4496 %

Table 7: AUC and AUPR on the training set and on the test set for the original dataset, for the dataset of the insurance company A, B and C and for the dataset that excludes macroeconomic variables. We report also the average lapse rate for the different sets.

6 Random forest robustness

To verify the robustness of the estimated RF model and to understand the key driver of the model performance we train and validate the selected RF model on sub-samples of the dataset (see, e.g., Heinze-Deml and Meinshausen 2017, Quionero-Candela *et al.* 2009):

1. we split the dataset into three subsets, one for each company;

2. we build a dataset that excludes the macroeconomic variables.

We compare the RF results for the different datasets in terms of AUC and AUPR and repeat the explainability analyses for the different datasets.

In Table 7 we report AUC and AUPR on the training set and test set for the original dataset, for the dataset of the insurance company A, B, and C, and for the dataset that excludes macroeconomic variables. Our approach is to fix the optimal hyper-parameters determined in the previous section and to train a RF model on each sub-sample to test the robustness of the results (with the same hyper-parameters). We observe that we have a good classifier in all the cases even if the performances are worse than in the full dataset, except for Company A. We notice that Company A has a small lapse rate in all years except 2011, this feature could explain the very good performance on this dataset. The differences between the error metrics on the training dataset and the test dataset are slightly larger than those obtained in the complete dataset (again, except for Company A), however, the degree of overfitting seems to be limited. Note that it would be possible to obtain better performances by selecting the best performing model on the specific dataset through a new grid search on the hyper-parameters. However, we prefer not to do it because it would not allow the comparison among the outcomes of the models.

Overall, our selected RF model seems to be quite stable and robust to subsample analysis. Notice that our RF model still turns out to be a good classifier, even removing macroeconomic variables, as it renders a better model - in terms of all the performance metrics - than the logistic regression estimated using the macroeconomic variables.

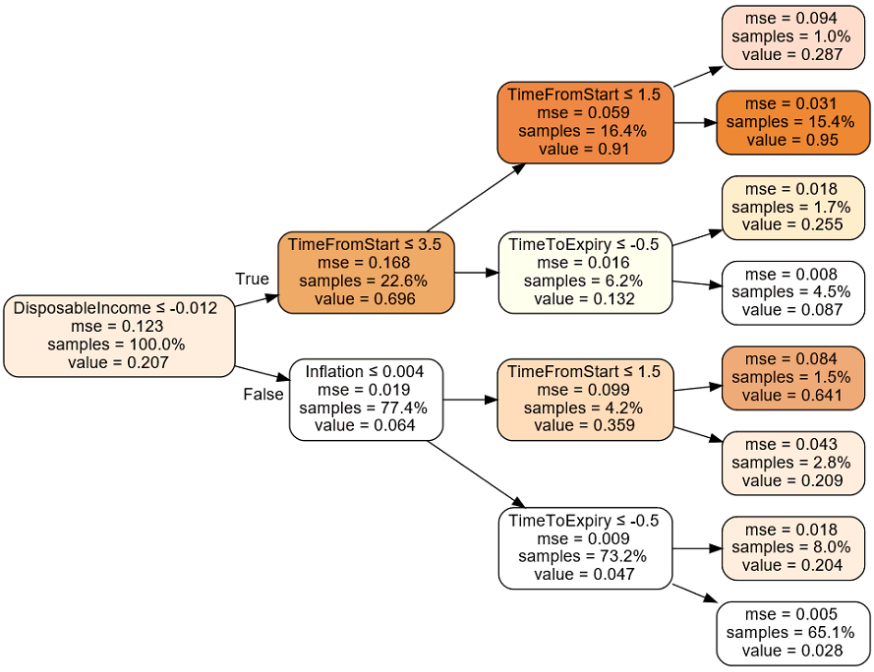


Figure 7: Regression tree fitted on the dataset of company A.

We finally investigate whether the global explainability analyses are robust varying the dataset. To accomplish this task, we apply the global explainability approach to each of the RF trained on the subsets. Thus, for each subset, we obtain a regression tree fitted on the outcome of the RF model. In Figure 7 we can observe the regression tree associated with the subset consisting of observations related to Company A, while in Figure 8 we deal with the subset which excludes macroeconomic variables. Time from start and time to expiry as well as disposable income growth rate, when macroeconomic variables are considered, are important in the estimation of the lapse decision as they are employed in the trees. These results confirm the previous analysis.

7 Conclusions

In this paper, we have used ML methods to analyze lapses of life insurance contracts by policyholders. The decision to lapse an insurance contract has been analyzed in a large literature, both from a theoretical and an empirical point of view.

A "rational" lapse decision refers to a low interest rate, that renders more convenient to switch to another life contract, or to weak economic conditions of the policyholder. Testing these hypotheses on microdata, the evidence is limited: the fraction of the variability of the lapse decision captured by proxies of these motivations is not high. Behavioral and commercial reasons seem to play a significant role. To investigate them in an agnostic way, i.e., starting from a large set of variables, it is useful to use ML tools that allow data to speak for themselves.

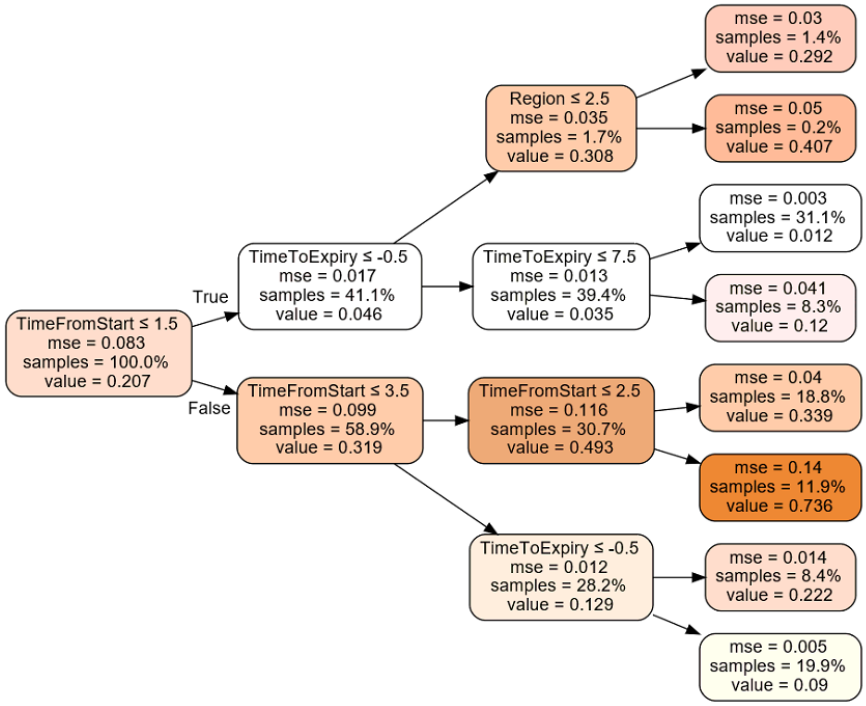


Figure 8: Regression tree fitted on the dataset with no macroeconomic variables.

We confirm that a RF performs better than the classical logistic model to predict the lapse decision, even if the interactions between features are taken into account. The ML methodology allowed us to discern the relevance of a wide set of exogenous variables to explain the lapse decision.

The main result of our analysis is that the important drivers of the lapse decision are the time passed from the incipit of the contract and the time to expiry, as well as the insurance company, the contract size, and premium. Other features of the policyholder (gender, age of the policyholder, region) or of the contracts (product type) as well as macroeconomic variables (with the exception of the disposable income growth rate with a nonlinear effect) play a limited role. This is in contrast with results obtained using linear models and confirms that traditional hypotheses that associate the lapse decision to economic convenience or to the economic condition of the policyholder have a weak capability to explain the phenomenon. These results also confirm that linear models are not able to fully capture the heterogeneity of financial decisions.

Acknowledgements

The work has been partially supported by the European Union's Horizon 2020 training and innovation program "FIN-TECH", under the grant agreement No. 825215 (Topic ICT-35-2018, Type of actions: CSA), and by the COST Action CA19130 - "Fintech and Artificial Intelligence in Finance - Towards a transparent financial industry" (FinAI).

References

Antoniano-Villalobos, I., Borgonovo, E., and Lu, X., 2020. Nonparametric estimation of probabilistic sensitivity measures, *Statistics and Computing*, 30 (2), 447–467.

Babaoglu, C., Ahmad, U., Durrani, A., and Bener, A., 2017. Predictive modeling of lapse risk: An international financial services case study, *in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 16–21.

Barboza, F., Kimura, H., and Altman, E., 2017. Machine learning models and bankruptcy prediction, *Expert Systems with Applications*, 83 (C), 405–417.

Barucci, E., Colozza, T., Marazzina, D., and Rroji, E., 2020. The determinants of lapse rates in the Italian life insurance market, *European Actuarial Journal*, 10, 149–178.

Bauer, D., Gao, J., Moenig, T., Ulm, E.R., and Zhu, N., 2017. Policyholder exercise behavior in life insurance: The state of affairs, *North American Actuarial Journal*, 21 (4), 485–501.

Bemš, J., Starỳ, O., Macaš, M., Žegklitz, J., and Pošík, P., 2015. Innovative default prediction approach, *Expert Systems with Applications*, 42 (17-18), 6277–6285.

Borgonovo, E. and Plischke, E., 2016. Sensitivity analysis: a review of recent advances, *European Journal of Operational Research*, 248 (3), 869–887.

Breiman, L., 2001. Random forests, *Machine learning*, 45 (1), 5–32.

Cui, H., Rajagopalan, S., and Ward, A.R., 2020. Predicting product return volume using machine learning methods, *European Journal of Operational Research*, 281 (3), 612–627.

Davis, J. and Goadrich, M., 2006. The relationship between precision-recall and roc curves, *in: Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA: Association for Computing Machinery, ICML '06, 233–240.

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*.

Fischer, T. and Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research*, 270 (2), 654–669.

Guelman, L. and Guillen, M., 2014. A causal inference approach to measure price elasticity in automobile insurance, *Expert Systems with Applications*, 41, 387–396.

Guelman, L., Guillen, M., and Pérez-Marín, A., 2015. Uplift random forests, *Cybernetics and Systems*, 46, 230 – 248.

Guelman, L., Guillén, M., and Pérez-Marín, A.M., 2012. Random forests for uplift modeling: An insurance customer retention case, *in:* K.J. Engemann, A.M. Gil-Lafuente, and J.M. Merigó, eds., *Modeling and Simulation in Engineering, Economics and Management*, Berlin, Heidelberg: Springer Berlin Heidelberg, 123–133.

Guelman, L., Guillén, M., and Pérez-Marin, A.M., 2014a. Optimal personalized treatment rules for marketing interventions: A review of methods, a new proposal, and an insurance case study, Tech. rep., UB Riskcenter Working Papers Series 2014-06.

Guelman, L., Guillén, M., and Pérez-Marín, A.M., 2014b. A survey of personalized treatment models for pricing strategies in insurance, *Insurance: Mathematics and Economics*, 58, 68 – 76.

Heinze-Deml, C. and Meinshausen, N., 2017. Conditional variance penalties and domain shift robustness, *arXiv preprint*, 1710.11469.

James, G., Witten, D., Hastie, T., and Tibshirani, R., 2013. *An introduction to statistical learning*, vol. 112, Springer.

Jeong, H., Gan, G., and Valdez, E.A., 2018. Association rules for understanding policyholder lapses, *Risks*, 6 (3), 69.

Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., and Klein, M., 2002. *Logistic regression*, Springer.

Krauss, C., Do, X.A., and Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500, *European Journal of Operational Research*, 259 (2), 689–702.

Kuhn, M. and Johnson, K., 2013. *Applied Predictive Modeling*, SpringerLink : Bücher, Springer New York.

Lally, N.R. and Hartman, B.M., 2016. Predictive modeling in long-term care insurance, *North American Actuarial Journal*, 20 (2), 160–183.

Liaw, A., Wiener, M., *et al.*, 2002. Classification and regression by random forest, *R news*, 2 (3), 18–22.

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.I., 2020. From local explanations to global understanding with explainable ai for trees, *Nature Machine Intelligence*, 2 (1), 56–67.

Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions, *in: Advances in neural information processing systems*, 4765–4774.

Lundberg, S.M., Nair, B., Vavilala, M.S., Horibe, M., Eisses, M.J., Adams, T., Liston, D.E., Low, D.K.W., Newman, S.F., Kim, J., and Lee, S.I., 2018. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery, *Nature Biomedical Engineering*, 2 (10), 749–760.

Malekipirbazari, M. and Aksakalli, V., 2015. Risk assessment in social lending via random forests, *Expert Systems with Applications*, 42 (10), 4621–4631.

Melkumova, L. and Shatskikh, S.Y., 2017. Comparing Ridge and Lasso estimators for data analysis, *Procedia engineering*, 201, 746–755.

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.

Moscatelli, M., Parlapiano, F., Narizzano, S., and Viggiano, G., 2020. Corporate default forecasting with machine learning, *Expert Systems with Applications*, 161, 113567.

Patro, S. and Sahu, K.K., 2015. Normalization: A preprocessing stage, *arXiv preprint arXiv:1503.06462*.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N.D., 2009. *Dataset shift in machine learning*, The MIT Press.

Ribeiro, M.T., Singh, S., and Guestrin, C., 2016. "Why should i trust you?" Explaining the predictions of any classifier, *in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Saito, T. and Rehmsmeier, M., 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets, *PloS one*, 10 (3), e0118432.

Zhang, J., Cui, S., Xu, Y., Li, Q., and Li, T., 2018. A novel data-driven stock price trend prediction system, *Expert Systems with Applications*, 97, 60–69.