Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds *

Elena Dumitrescu, Sullivan Hué, Christophe Hurlin, Sessi Tokpavi

Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds*

Dumitrescu, Elena[†], Hué, Sullivan[‡], Hurlin, Christophe[§], Tokpavi, Sessi[¶]

February, 2020

Abstract

Decision trees and related ensemble methods like random forest are state-of-the-art tools in the field of machine learning for credit scoring. Although they are shown to outperform logistic regression, they lack interpretability and this drastically reduces their use in the credit risk management industry, where decision-makers and regulators need transparent score functions. This paper proposes to get the best of both worlds, introducing a new, simple and interpretable credit scoring method which uses information from decision trees to improve the performance of logistic regression. Formally, rules extracted from various short-depth decision trees built with couples of predictive variables are used as predictors in a penalized or regularized logistic regression. By modeling such univariate and bivariate threshold effects, we achieve significant improvement in model performance for the logistic regression while preserving its simple interpretation. Applications using simulated and four real credit defaults datasets show that our new method outperforms traditional logistic regressions. Moreover, it compares competitively to random forest, while providing an interpretable scoring function.

JEL Classification: G10 C25, C53
Keywords: Risk management; Credit scoring; Machine Learning; Interpretability; Econometrics.

*This paper has previously circulated under the title "Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects". We thank the ANR programs MultiRisk (ANR-16-CE26-0015-01) and CaliBank (ANR-19-CE26-0002-02) for supporting our research.

[†]EconomiX-CNRS, University of Paris Nanterre, 200 Avenue de la République, 92000 Nanterre, France. E-mail: elena.dumitrescu@parisnanterre.fr

[‡]Corresponding author, Univ. Orléans, CNRS, LEO (FRE 2014), Rue de Blois, 45067 Orléans. E-mail: sullivan.hue@univ-orleans.fr

[§]Univ. Orléans, CNRS, LEO (FRE 2014), Rue de Blois, 45067 Orléans. E-mail: christophe.hurlin@univ-orleans.fr

[¶]Univ. Orléans, CNRS, LEO (FRE 2014), Rue de Blois, 45067 Orléans. E-mail: sessi.tokpavi@univ-orleans.fr

1 Introduction

Credit scoring is a fairly widespread practice in banking institutions, whose main objective is to discriminate between borrowers based on their creditworthiness. Borrowers (retails or corporates) with high scores are qualified as safer and get access to credit, while those with low scores are rationed or get access to credit in less favorable terms. In a world with asymmetric information, such practices are used to allocate default risk by avoiding underpricing (overpricing) bad (good) loans.

The quest for good models that predict credit worthiness is motivated not only by the analysis of credit scoring economic costs and benefits (see Berger et al., 2005; Stein and Jordao, 2003; Stein, 2005; Blöchlinger and Leippold, 2006, for example), but also, most importantly, by its implications for the banking system. Indeed, credit scoring is important for banks and regulators as the Basel III (Basel Committee on Banking Supervision, 2011) reinforces capital requirements for the coverage of credit risk. Hence, in absence of performant credit scoring models, the true levels of credit risk and hence capital requirements could be underestimated, which would render the banking system less resilient to financial crises and more exposed to systemic events if it is not capitalized enough (Engle et al., 2015; Acharya et al., 2017). On the contrary, banks' level of capital requirements could be overstated, hence raising its credit cost and/or decreasing the loan volume (Rochet, 1992) with further negative effects on the real sector.

Traditionally, borrowers' default probability is estimated with predictive statistical methods and regression models such as discriminant analysis (Altman, 1968), proportional hazard or logistic regression models (Steenackers and Goovaerts, 1989; Stepanova and Thomas, 2001), the latter model appearing as the benchmark econometric model, mainly because of its simplicity and its flexibility in providing sensitivity analysis through marginal effects of explanatory variables. Machine learning techniques have also been shown to successfully forecast credit scores. Some early examples are the k-nearest neighbor (Henley and Hand, 1996, 1997), neural networks (Desai et al., 1996; West, 2000; Yobas et al., 2000), decision trees (Yobas et al., 2000), and support vector machine (Baesens et al., 2003). However, the empirical results were mixed (Thomas, 2000; Hurlin and Pérignon, 2019). But the big data revolution renewed interest in some algorithms introduced in the late 1990s[1]. The two most widely used ones are Bagging (Breiman, 1996) and Boosting (Schapire et al., 1998), and their domains of application are rather scattered, including face detection and recognition, genes selection, medical imaging, weather forecast, fraud detection, etc. Bagging and Boosting are ensemble (aggregation) methods that aim at improving the predictive performance of a

[1]See Óskarsdóttir et al. (2019) or Frost et al. (2019) for a general discussion about the value of big data for credit scoring. In this article, we limit ourselves to the use of machine learning algorithms with "traditional data" for credit risk analysis without any reference to "new data" (social or communication networks, digital footprint, etc.) and/or "big data".

given statistical or machine learning algorithm (weak learner) by using a linear combination (through averaging or majority vote) of predictions from many variants of this algorithm rather than a single prediction. The two methods differ mainly in their aggregation scheme. For a review of Bagging and Boosting methods, see Hastie et al. (2001) and Bühlmann (2012).

Applications of such methods to credit scoring can be found in Finlay (2011), Paleologo et al. (2010), and Lessmann et al. (2015). Finlay (2011) finds that bagging and boosting methods outperform simple classifiers or models among which the logistic regression. Paleologo et al. (2010) propose an ensemble classification technique called subagging which is shown in an empirical application on credit scoring to improve significantly the performance of traditional classifiers. Similar conclusions arise from the benchmarking study proposed by Lessmann et al. (2015). Relying on various assessment criteria and a large number of credit-scoring datasets, they found, among others, that random forest, i.e. the randomized version of bagged decision trees (Breiman, 2001), outperforms logistic regression and seems to be the benchmark ensemble method in terms of predictive performance both in academia and credit risk management industry (Grennepois et al., 2018).

Nevertheless, as random forest's decision rules arise from the aggregation of individual decision tree rules, they are hardly interpretable. Consequently, although they perform very well in default prediction, random forests can be less relevant in credit scoring applications where decision makers and regulators need parsimonious and interpretable rules (e.g. marginal effects or scorecards) like those based on logistic regression. Recently, many Model-Agnostic Methods have been proposed to make the "black box" machine learning models explanaible and/or their decisions interpretable, see Molnar (2019) for a complete overview[2]. We can cite here among many others, the Partial Dependencies Plots (PdP), the global or local surrogate models (such as the LIME for instance) which consist in interpretable models that are trained to approximate the predictions of a black box model, etc. In the credit scoring industry (see for instance Bracke et al. (2019) or Grennepois and Robin (2019)), the Shapley value is often used. This method assumes that each feature value of an individual is a player in a game where the prediction is the payout and distributes the payout among features (Lundberg and Lee, 2017). Although this method is attractive, getting the Shapley values requires a lot of computing time because the number of coalitions grows exponentially with the number of predictive variables, and computational shortcuts that exist are based on coalitions' sampling that only provides approximate and unstable solutions. Another approach is the "InTrees" method proposed by Deng (2019). The framework extracts, measures, prunes, selects, and summarizes rules from a tree ensemble, and calculates frequent variable interactions. This helps detecting simple decision rules from the forest that are important in predicting the outcome variable. Nevertheless, the algorithms

[2]In the sequel, we will not distinguish explainability from interpretability.

underlying the extraction of these rules are not simple to disclose.

Our approach aims to avoid the traditionnal arbitrage between interpretability and forecasting performances. We propose here to restrict intrinsic complexity of credit score models, rather than apply interpretability methods to analyze the model after training. To do so, we exploit the fact that ensemble methods like random forests consistently outperform logistic regression because the latter method fails to fit non-linear effects. Indeed, random forest benefits from the recursive partitioning underlying decision trees and hence, by design, accommodates unobserved multivariate threshold effects. The trick of our approach consists in using these algorithms to pre-treat our predictors instead of modeling the default probability directly with machine learning methods. Thus, our approach takes benefit from machine learning algorithms for data pre-processing and feature engineering, while keeping the credit score model fully interpretable, as recommended by the regulators. To the best of our knowledge, this is the first time that such an approach is applied for credit scoring.

The *Penalized Logit Tree Regression* model, hereafter PLTR, is based on a logistic regression with predictors extracted from decision trees. Formally, these predictors are binary rules (leafs) outputted by the short-depth decision trees built with couples of original predictive variables. To handle a possibly large number of such decision tree rules and to proceed to variables selection, an Adaptive Lasso logistic regression model (Zou, 2006; Friedman et al., 2010), i.e., a penalized version of the classical logistic regression, is estimated. Firstly, we propose several Monte Carlo experiments to illustrate the inability of standard parametric models, i.e. standard logistic regression models with linear specification of the index or with quadratic and interaction terms, to well-capture the non-linear effects (thresholds and interactions) which could arise in credit-scoring data. Furthermore, these simulations allow us to evaluate the relative performance of the PLTR in presence of non-linear effects, while controlling for the number of predictors. We show that the PLTR clearly outperforms the traditional logistic regression in terms of forecasting accuracy. Moreover, it compares competitively to random forest and even surpasses it in some cases, while providing an interpretable scoring function. Secondly, we apply the PLTR and five other benchmark credit-scoring methodologies (random forest, linear logistic regression, non-linear logistic regression, non-linear logistic regression and Adaptative Lasso) on four real datasets. The empirical results confirm those of the simulations, as the PLTR yields a very good forecasting performance for all the datasets, unlike other benchmark models. This conclusion is robust to the various predictive accuracy indicators considered by Lessmann et al. (2015). Finally, we show that the PLTR also leads to more cost reductions than alternative credit-scoring models.

Our approach can be viewed as a systematization of a common practice in the deployment of credit scoring solutions that traditionally use logistic regression. Credit risk managers usually introduce non-linear effects in logistic regression by using ad-hoc or heuris-

tic preteatments and feature engineering methods such as the discretization of continuous variables, merger of categories, identification of non-linear effect by cross-product variables, etc.[3] The merit of our contribution is to propose a systematic approach to the modeling of such unobserved non-linear effects by using short-depth decision trees. Lastly, it is worth stressing that our contribution differs from those arising from the so-called Logit-Tree models, i.e., trees that contain logistic regressions at the leaf nodes. Examples are the Logistic Tree with Unbiased Selection (LOTUS) in Chan and Loh (2004) and the Logistic Model Tree (LMT) in Landwehr et al. (2005). Moreover, although similar in spirit, our PLTR method contrasts with the hybrid CART-Logit model of Cardell and Steinberg (1998) too. Indeed, to introduce multivariate threshold effects in logistic regression, they use a single non-pruned decision tree. But the large depth of this unique tree complicates the interpretability of the results and may lead to predictors inflation that is not controlled for (e.g. through penalization as in our case).

The rest of the article is structured as follows. Section 2 analyses the performance of logistic regression and random forest in the presence of univariate and multivariate threshold effects through Monte Carlo simulations. In Section 3 we introduce the PLTR credit scoring method and assess through Monte Carlo simulations its accuracy and interpretability (parsimony) in the presence of threshold effects. Section 4 is devoted to an empirical application with a benchmark dataset. Robustness of the results through datasets is explored in Section 5. Section 6 compares the models from an economic point of view, while the last Section concludes.

2 Threshold effects in logistic regression

2.1 Non-linear effects and logistic regression model

Let (x_i, y_i), $i = 1, ..., n$, be a sample of size n of independent and identically distributed observations where $x_i \in \mathbb{R}^p$ is a p-dimensional vector of predictors and $y_i \in \{0, 1\}$ is a binary variable taking the value one when the i-th borrower defaults and zero otherwise. The goal of a credit scoring model is to provide an estimate of the posterior probability $\Pr(y_i = 1 \,|x_i)$ that borrower i defaults given his attributes x_i. The relevant characteristics of the borrower vary according to its status: household or company. For corporate credit risk scoring, the candidate predictive variables $x_{i,j}$, $j = 1, ..., p$, may include balance-sheet financial variables that cover various aspects of the financial strength of the firm, like the firm's operational performance, its liquidity, and capital structure (Altman, 1968). For instance, using a sample of 4,796 Belgian firms, Bauweraerts (2016) shows the importance of taking into account the level of liquidity, solvency and profitability of the firm in forecasting its bankruptcy risk. For small and medium enterprises (SMEs) specific variables related to

[3] See Hurlin and Pérignon (2019).

the financial strength of the firm's owner are also shown to be important (Wang, 2012). For retail loans, financial variables such as the number and amount of personal loans, normal repayment frequency of loans, the number of credit cards, the average overdue duration of credit cards and the amount of housing loans are combined with socio-demographic factors. A typical example is the FICO score, which is widely used in the US financial industry to assess the creditworthiness of individual customers.

Regardless of the type of borrower, the conditional probability of default is generally modeled using a logistic regression with the following specification

$$\Pr\left(y_i = 1|x_i\right) = F\left(\eta\left(x_i;\beta\right)\right) = \frac{1}{1+\exp\left(-\eta\left(x_i;\beta\right)\right)}, \tag{1}$$

with $F\left(.\right)$ the logistic cumulative distribution function, and $\eta\left(x_i;\beta\right)$ the so-called index function defined as

$$\eta\left(x_i;\beta\right) = \beta_0 + \sum_{j=1}^{p}\beta_j x_{i,j}, \tag{2}$$

where $\beta = (\beta_0, \beta_1, ..., \beta_p) \in \mathbb{R}^{p+1}$ is an unknown vector of parameters. The estimator $\widehat{\beta}$ is obtained by maximizing the convex log-likelihood function

$$\mathcal{L}(y_i;\beta) = \sum_{i=1}^{n}\Big\{y_i \log\left\{F\left(\eta\left(x_i;\beta\right)\right)\right\} + \left(1-y_i\right)\log\left\{1-F\left(\eta\left(x_i;\beta\right)\right)\right\}\Big\}. \tag{3}$$

Under some regular weak assumptions, the estimator $\widehat{\beta}$ is consistent and has a Gaussian limiting distribution which allows for simple inferential procedures.

The main advantage of the logistic regression model is its simple interpretation. Indeed, this model searches for a single linear decision boundary in the predictors' space. The core assumption for finding it is that the index $\eta\left(x_i;\beta\right)$ is linearly related to the predictive variables. In this framework, it is easy to evaluate the relative contribution of each predictor to the probability of default. This is achieved by computing marginal effects as

$$\frac{\partial \Pr\left(y_i = 1\,|x_i\right)}{\partial x_{i,j}} = \beta_j \frac{\exp\left(\eta\left(x_i;\beta\right)\right)}{\left[1+\exp\left(\eta\left(x_i;\beta\right)\right)\right]^2}, \tag{4}$$

with estimates obtained by replacing β by $\widehat{\beta}$. Thus, a predictive variable with positive (negative) significant coefficient has a positive (negative) impact on the borrower's default probability.

Obviously, this simplicity comes at a cost when significant non-linear relationships exist between the default indicator, y_i, and the predictive variables, x_i. A very common type of non-linearity can arise from the existence of an univariate threshold effect on a single predictive variable but it can also be generalized to a combination of such effects (multivariate threshold effects) across variables. A typical example of the former case in the context of credit scoring is the income "threshold effect", which implies the existence of an endogenous income threshold below (above) which default probability is more (less) prominent. The

income threshold effect can obviously interact with other threshold effects, leading to highly non-linear multivariate threshold effects. The common practice to approximate non-linear effects in credit scoring applications is to introduce quadratic and interaction terms in the index function $\eta(x_i;\beta)$. However, such a practice is not successful when unobserved threshold effects are at stake. Below, we run Monte Carlo simulation experiments to provide more insight into this issue.

Formally, we first generate p predictive variables $x_{i,j}$, $j = 1, ..., p$, $i = 1, ..., n$, where the sample size is set to $n = 5000$. Each predictive variable $x_{i,j}$ is assumed to follow the standard Gaussian distribution. The index function $\eta(x_i;\Theta)$ is simulated as follows

$$\eta(x_i;\Theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \mathbf{1}(x_{i,j} \leq \gamma_j) + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \beta_{j,k} \mathbf{1}(x_{i,j} \leq \delta_j) \mathbf{1}(x_{i,k} \leq \delta_k), \tag{5}$$

where $\mathbf{1}(.)$ is the indicator function and $\Theta = (\beta_0, \beta_1, ..., \beta_p, \beta_{1,2}, ..., \beta_{p-1,p})'$ is the vector of parameters, with each component randomly drawn from an uniform $[-1, 1]$ distribution, and $(\gamma_1, ..., \gamma_p, \delta_1, ..., \delta_p)'$ are some thresholds parameters, whose values are randomly selected from the support of each generated predictive variable while excluding data below (above) the first (last) decile. The default probability is then obtained for each individual by plugging (5) into (1). Subsequently, the simulated target binary variable y_i is obtained as

$$y_i = \begin{cases} 1 & \text{if} \quad \Pr(y_i = 1 | x_i) > \pi \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where π stands for the median value of the generated probabilities.

Our objective is to assess logistic regression's performance to detect default in presence of univariate and bivariate threshold effects as introduced in (5). For this, we divide the simulated sample into two sub-samples of equal size at each replication. The first (second) sub-sample is labeled as the learning (test) sample. We estimate logistic regression models on the learning sample and evaluate their forecasting abilities with the test sample.

The first model that we estimate is the classical logistic regression, with linear effects, whose index is given in (2). The second logistic model we estimate has been designed specifically to capture non-linear effects and for this reason it is very often used in credit scoring applications. This specification is based on a non-linear index function that incorporates quadratic and interaction terms

$$\eta^{(nl)}\left(x_i;\Theta^{(nl)}\right) = \alpha_0 + \sum_{j=1}^{p} \alpha_j x_{i,j} + \sum_{j=1}^{p} \xi_j x_{i,j}^2 + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \zeta_{j,k} x_{i,j} x_{i,k}, \tag{7}$$

where $\Theta^{(nl)} = (\alpha_0, \alpha_1, ..., \alpha_p, \xi_1, ..., \xi_p, \zeta_{1,2}, ..., \zeta_{p-1,p})'$ is the unknown vector of parameters.

However, we argue here that both approaches fail to accurately model non-linearity in presence of univariate and bivariate threshold effects such as those in (5). We evaluate the

out-of-sample performance of the models by relying on the probability of correct classification (PCC) as evaluation criterion.[4]

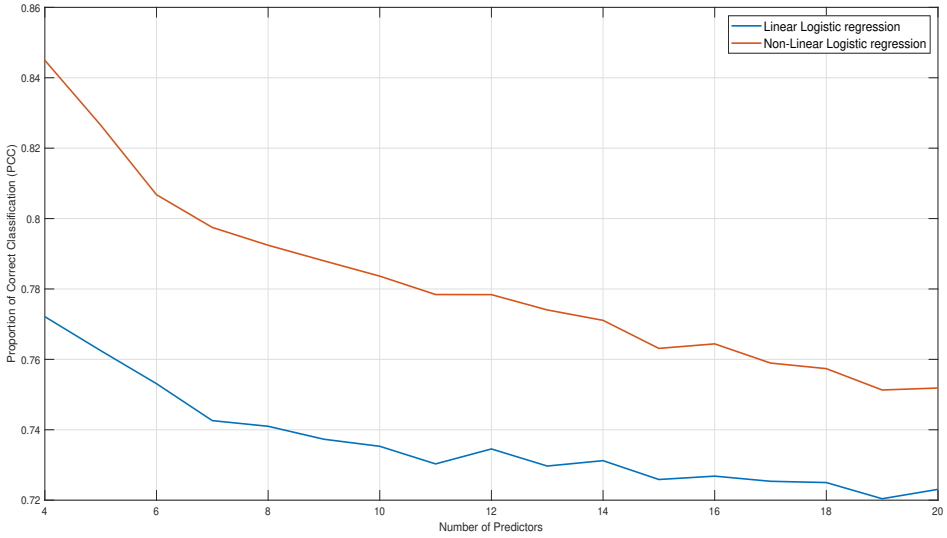

Figure 1: Comparison of performances under univariate and bivariate threshold effects: linear and non-linear logistic regressions

Figure 1 displays the average value of the PCCs of these two models over 100 simulations and for different number of predictors $p = 4, ..., 20$. We observe that the proportion of correct classification decreases with the number of predictors for both models. This suggests that in presence of univariate and bivariate threshold effects involving many variables, as in our DGP, logistic regression with linear index function, eventually augmented with quadratic and interaction terms, fails to discriminate between good and bad loans. Indeed, in the case where $p = 20$ the PCCs are equal to 72.30% and 75.19%, respectively. Hence, adding quadratic and interaction terms improves the predictive power, but the overall performance remains low when the number of predictors increases.

2.2 Machine Learning for non-linear effects

In the following we show that ensemble or aggregation methods for decision trees such as random forests perform much better in a framework with threshold effects. The out-performance of random forest arises from the non-linear "if-then-else" rules underlying decision trees. Indeed, the latter is a non-parametric supervised learning method based on a divide and conquer greedy algorithm that recursively partitions the training sample into smaller subsets, so as to group together as accurately as possible individuals with the same

[4]Note that all these models give the estimated probabilities of default $\widehat{p}_i$ for the N individuals. To compute the PCC, we need the estimated value of y_i, i.e., $\widehat{y}_i$. This is done by comparing $\widehat{p}_i$ to an endogenous threshold $\widehat{\pi}$. As usual, we set $\widehat{\pi}$ to a value such that the number of predicted defaults in the learning sample is equal to the observed number of defaults.

behaviour, i.e. the same value of the binary target variable "y_i". Formally, for a given tree, l, the algorithm proceeds as follows. Let $\mathcal{D}_{m,l}$ be the data (sub)set at a given node m of this tree. We denote by $\theta_{m,l} = (j_{m,l}, t_{m,l,j})$ a candidate split, where $j_{m,l} = 1, ..., p$ indicates a given predictive variable and $t_{m,l,j}$ is a threshold value in the support of this variable. The algorithm partitions the data $\mathcal{D}_{m,l}$ into two subsets $\mathcal{D}_{m,l,1}(\theta_{m,l})$ and $\mathcal{D}_{m,l,2}(\theta_{m,l})$, with[5]

$$\mathcal{D}_{m,l,1}(\theta_{m,l}) = (x_i, y_i) \mid x_{i,j} < t_{m,l,j}, \tag{8}$$

$$\mathcal{D}_{m,l,2}(\theta_{m,l}) = (x_i, y_i) \mid x_{i,j} \geq t_{m,l,j}, \tag{9}$$

where the parameter estimates $\widehat{\theta}_{m,l}$ satisfy

$$\widehat{\theta}_{m,l} = (\widehat{j}_{m,l}, \widehat{t}_{m,l,j}) \;=\; \arg\max_{\theta_{m,l}} \mathcal{H}(\mathcal{D}_{m,l}) - \frac{1}{2}\Big(\mathcal{H}(\mathcal{D}_{m,l,1}(\theta_{m,l})) + \mathcal{H}(\mathcal{D}_{m,l,2}(\theta_{m,l}))\Big), \tag{10}$$

with $\mathcal{H}(.)$ a measure of diversity, e.g. the Gini criterion, applied to the full sample and averaged across the two sub-samples, respectively. $\widehat{\theta}_{m,l}$ appears hence as the value of $\theta_{m,l}$ that reduces diversity the most within each subset resulting from the split. The splitting process is repeated until the terminal sub-samples, also known as leaf nodes, contain homogeneous individuals according to a predefined homogeneity rule. We denote by M_l the total number of splits in tree l and by $|T_l|$ the corresponding number of leaf nodes.

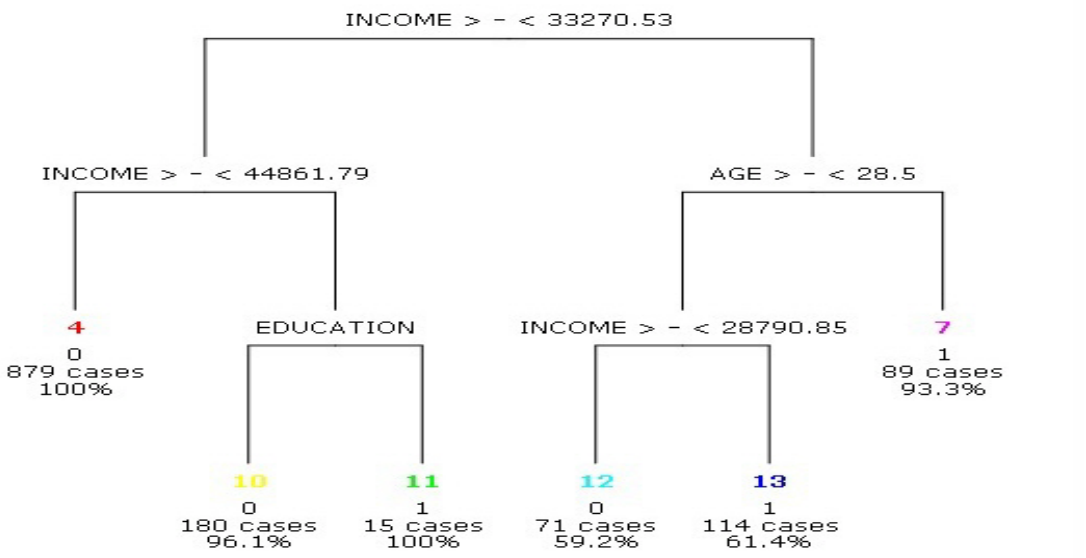

Figure 2: Example of decision tree for credit scoring

An illustrative example of a decision tree is given below in Figure 2. We observe that at the first iteration (or split), $m = 1$, $\widehat{\theta}_{m,l}$ is defined by $(\widehat{j}_{m,l}, \widehat{t}_{m,l,1})$, with $\widehat{j}_{m,l}$ the index of the variable "income" and $\widehat{t}_{m,l,1} = 33270.53$. The other iterations also include "age"

[5]To simplify the description of the algorithm we focus only on quantitative predictors. A similar procedure is available for qualitative predictors.

and "education" for further refinements. The process ends with a total number of 5 splits and 6 leaf nodes labeled 10, 11, 12, 13, 4 and 7, respectively. Each leaf node $\mathcal{R}_t$, $t = 1, ..., |T_l|$ includes a specific proportion of individuals belonging to each class of borrowers (1="default", 0="non default"). For instance, leaf node "7" contains 89 individuals, 93.3% of them having experienced a default event. Note that each of these individuals has an income lower than 33270.53 and is less than 28.5 years old. The predominant class in each leaf defines the predicted value of y_i for individuals i which belong to that particular leaf. Formally, we define the predicted default value for the i^{th} individual as

$$h_l(x_i; \widehat{\Theta}_l) = \sum_{t=1}^{|T_l|} c_t \mathcal{R}_{i,t}, \tag{11}$$

where $\Theta_l = (\theta_{m,l}, m = 1, ..., M_l)$ is the parameter vector for tree l, $\mathcal{R}_{i,t} = 1_{(i \in \mathcal{R}_t)}$ indicates whether individual i belongs to leaf $\mathcal{R}_t$, and c_t is the dominant class of borrowers in that leaf node. For example, in leaf node 7 the "default" class is dominant and hence the predicted value $h_l(x_i)$ is equal to 1 for all the individuals that belong to this leaf node. Notice that this simple tree allows to identify both interactions and theshold effects. For instance, in the simple example of Figure 2, the predicted value can be viewed as the result of a kind of linear regression[6] on the product of two binary variables that takes a value one if the income is lower than 33270.53 and the age is less than 28.5.

The random forest method is a bagging procedure that aggregates many non correlated decision trees. It exploits decision trees power to detect univariate and multivariate threshold effects while reducing their instability. Its superior predictive performance springs from the variance reduction effect of bootstrap aggregation for non correlated predictions (Breiman, 1996). Let L trees be constructed from bootstrap samples (with replacement) of fixed size drawn from the original sample. To insure a low level of correlation among those trees, the random forest algorithm chooses the candidate variable for each split in every tree, $j_{m,l}$ with $m \in \{1, \ldots, M_l\}$ and $l \in \{1, \ldots, L\}$, from a restricted number of randomly selected predictors among the p available ones. The default prediction of the random forest for each borrower, $h(x_i)$, is obtained by the principle of majority vote, that is $h(x_i)$ corresponds to the mode of the empirical distribution of $h_l(x_i; \widehat{\Theta}_l)$, $l = 1, ..., L$.

Numerous empirical papers have stressed random forests' performance in the context of credit scoring (see Lessmann et al., 2015, among others). We illustrate its relative performance in our Monte Carlo simulations setup. We consider the data generating process with non-linearity given in Equation (5) and then we estimate random forests based on the simulated data. The proportion of correct classification for the random forest algorithm, displayed as a yellow line in Figure 3, is computed over the same test samples of length 2500 as the PCCs of the logistic regressions previously discussed. The optimal number of

[6]This equivalence is only true in the case of a regression tree when the target variable y is continuous.

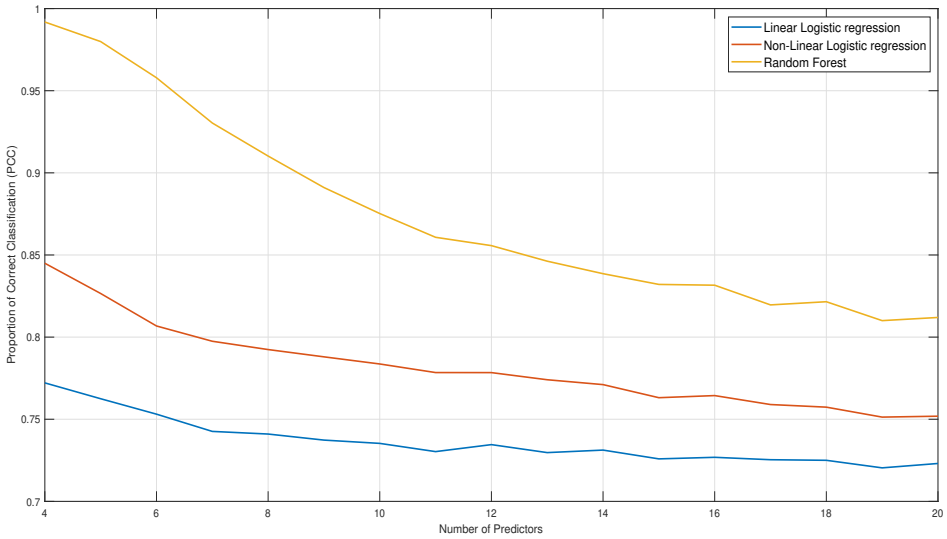

Figure 3: Comparison of performances under univariate and bivariate threshold effects: linear and non-linear logistic regressions, and Random Forest

trees in the forest, L, is tuned using the out-of-bag error. See Breiman (2001) for more information on this out-of-sample measure of performance. In presence of non-linear effects, the random forest outperforms not only the linear logistic regression as expected, but also the non-linear logistic regression. This result is due to the fact that the non-linear logistic regression model neglects the threshold effects, while taking into account the interactions between the predictors. On the contrary, the random forest is able to well-capture both features. This result is valid whatever the number of predictors, even if the differences in classification performance of the three models tend to decrease with the number of predictors. As the number of predictors increases, the complexity and the non-linearity of the DGP also increases, inducing a reduced performance for all the classifiers. For instance, the PPCs are equal to 99.18% (resp. 84.50%) for the random forest (resp. logistic regression with quadratic and interaction terms) in the case with 4 predictors, against 81.20% (resp. 75.19%) in the case with 20 predictors. Although the difference reduces from 14.68% to 6.01%, the PCC of the random forest is always higher than that of the logistic regression, confirming the empirical results generally found in the literature (see Lessmann et al., 2015, for example).

Despite insuring good performance, the aggregation rule (majority vote) underlying random forest leads to a prediction rule that lacks interpretation. This opaqueness is harmful for credit scoring applications, where decision makers and regulators usually need simple score functions like the linear index function from the logistic regression whose economic content is transparent.

The key question here is how to find a good trade-off between predictive performance

and interpretability. To gauge this issue, two lines of research can be explored. First, one can try to diminish the complexity of the random forest's aggregation rule by selecting (via an objective criterion) only some trees or decision rules in the forest.[7] Second, we can preserve the simplicity of logistic regression while improving its predictive performance with univariate and bivariate endogenous threshold effects. We opt here for the second line of research and leave the first one for further research. We propose to preserve the simplicity of logistic regression while improving its predictive performance with univariate and bivariate endogenous threshold effects. To be more precise, rules extracted from various short-depth decision trees built from couples of predictive variables are used as predictors in (regularized) logistic regression. These rules are dummy variables associated to leaf nodes from the various decision trees, and allow us to endogenously model univariate and bivariate threshold effects. The next section is devoted to the presentation of the proposed credit-scoring method.

3 Penalized Logit Tree Regression

3.1 Description of the methodology

In this paper, we propose to build a parsimonious logistic regression model from endogenous univariate and bivariate threshold effects, that we call "Penalized Logistic Tree Regression", henceforth PLTR. Both types of effects are obtained from short decision trees that rely on each possible couple of predictive variables at a time, where the dependent variable y_i measures borrower's default status. The algorithm proceeds in two steps.

The objective of the first step is to identify threshold effects from trees with two splits. For illustration, take income and age to be the j^{th} and k^{th} explanatory variables, and assume that income is more informative than age in explaining credit default. For each individual i, the corresponding decision tree will generate three binary variables, each associated to a terminal node. The first binary variable $\mathcal{V}_{i,1}^{(j)}$ will account for univariate threshold effects and could take value one when the income of individual i is higher than an estimated income threshold, and zero otherwise. The second (third) binary variable $\mathcal{V}_{i,2}^{(j,k)}$ ($\mathcal{V}_{i,3}^{(j,k)}$), representing bivariate threshold effects, would be equal to one when the person's income is lower than its threshold and at the same time his/her age is lower (higher) than an estimated age threshold, and zero otherwise.[8] Note that this particular form of splitting should arise when both variables are informative, i.e. each of them is selected in the iterative process of

[7]Note that this is the approach underlying the so-called "inTrees" method of Deng (2019) who proposes a methodology to render the outputs of random forest interpretable, by extracting simple rules from a tree ensemble.

[8]It is also possible that the univariate threshold variable $\mathcal{V}_{i,1}^{(j)}$ takes value one when the income is lower than an estimated income threshold, and zero otherwise. In that case, the bivariate threshold effect $\mathcal{V}_{i,2}^{(j,k)}$ ($\mathcal{V}_{i,3}^{(j,k)}$) would be equal to one when the individual's income is higher than its threshold and at the same time his/her age is lower (higher) than an estimated age threshold, and zero otherwise.

splitting. If the second variable is non-informative (age), the tree will rely twice on the first informative variable (income). Figure 4 gives an illustration of the splitting process.

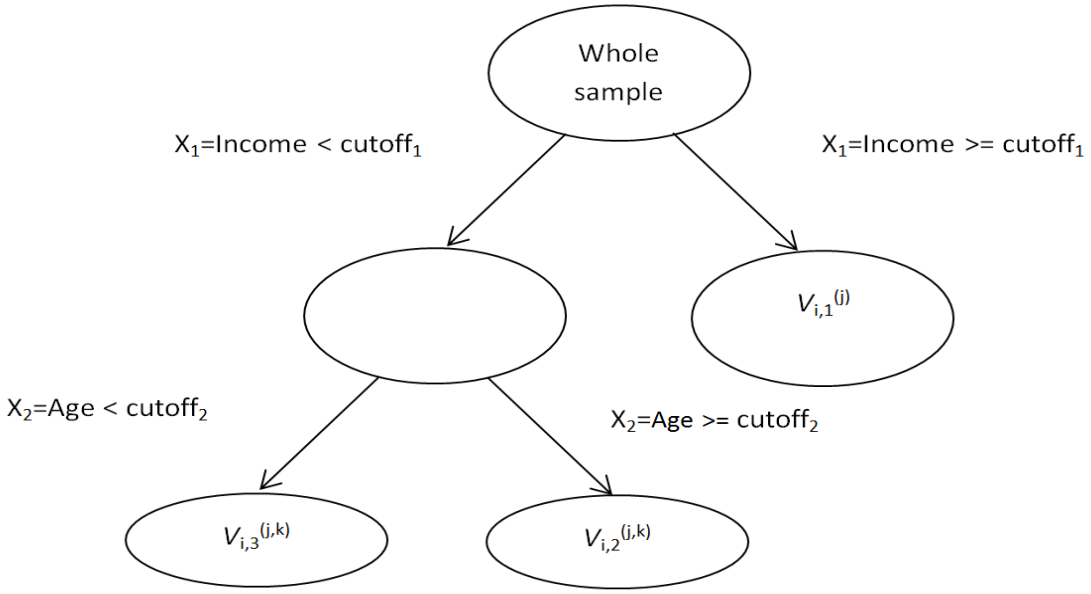

Figure 4: Illustration of the two-stage splitting process

One leaf of each of the two branches originating from the root of the tree is retained so as to cover both one and two splits, i.e. the first two binary variables $\mathcal{V}_{i,1}^{(j)}$ and $\mathcal{V}_{i,2}^{(j,k)}$ in the example above. We count at most $p+q$ threshold effects for inclusion in our logistic regression, where p represents the number of predictive variables and q denotes the total number of couples of predictive variables[9]. This is the case because the univariate threshold effects $\mathcal{V}_{i,1}^{(j)}$ are generated only by the variables retained in the first split irrespective of the variables retained in the second split. Some predictive variables may be selected in the first split of several trees, while others may never be retained. The latter do not produce any univariate threshold effects, while the former deliver identical univariate threshold effects, $\mathcal{V}_{i,1}^{(j)}$, out of which only one will be included in the logistic regression.

Note that one could also go beyond two splits by analyzing triplets or quadruplets of predictive variables. Such a procedure would allow the inclusion of more complex non-linear relationships in the logistic regression. Nevertheless, the expected uprise in performance would come at the cost of increased complexity of the model towards that of random forests which would plunge its level of interpretability. For this reason, in our PLTR model we use only short-depth decision trees involving two splits.

In the second step, the endogenous univariate and bivariate threshold effects previously obtained are plugged in the logistic regression

$$\Pr\left(y_i = 1|\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta\right) = \frac{1}{1+\exp\left[-\eta(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)\right]}, \tag{12}$$

[9] At most, $q = \frac{p\times(p-1)}{2}$.

with

$$\eta(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \mathcal{V}_{i,1}^{(j)} + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \gamma_{j,k} \mathcal{V}_{i,2}^{(j,k)} \tag{13}$$

the index and $\Theta = (\beta_0, \beta_1, ..., \beta_p, \gamma_{1,2}, ..., \gamma_{p-1,p})'$ the set of parameters to be estimated. The corresponding log-likelihood is

$$\begin{aligned} \mathcal{L}(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) &= \frac{1}{n} \sum_{i=1}^{n} \Big[y_i \log \Big[F\Big(\eta(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)\Big)\Big] \\ &\quad + (1 - y_i) \log \Big[1 - F\Big(\eta(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)\Big)\Big]\Big], \end{aligned}$$

where $F(\eta(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta))$ is the logistic CDF (see (1) for its general expression). The estimate $\widehat{\Theta}$ is obtained by maximizing the above log-likelihood with respect to the unknown parameters Θ. Remark that the length of Θ depends on p, the number of predictive variables and can be relatively high. For instance, there are 45 couples of variables when $p = 10$; this leads to a maximum number of 55 univariate and bivariate threshold effects that play the role of predictors in our logistic regression.

To prevent overfitting issues in this context with a large number of predictors, a common approach is to rely on penalization (regularization) for both estimation and variable selection. Called *penalized logistic tree regression* in our case, this method consists in adding a penalty term to the negative value of the log-likelihood function, such that

$$\mathcal{L}_p(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = -\mathcal{L}(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) + \lambda P(\Theta), \tag{14}$$

with $P(\Theta)$ the additional penalty term and λ a tuning parameter that controls the intensity of the regularization and which is selected in such a way that the resulting model minimises the out of sample error. The optimal value of the tuning parameter λ is usually obtained by relying on grid-search with cross-validation or by using some information criteria. At the same time, several penalty terms $P(\Theta)$ have been proposed in the related literature (Tibshirani, 1996; Zou and Hastie, 2005; Zou, 2006), but the most popular one is still the L1-penalty ($P(\Theta) = \sum_{j=1}^{m} |\theta_j|$) of Tibshirani (1996) that corresponds to the Least Absolute Shrinkage and Selection Operator (Lasso). This method has the advantage of performing both feature selection and regularization of coefficients while being computationally feasible in high dimensional data.

Nonetheless, the Lasso estimator does not satisfy the oracle property (Fan and Li, 2001): the probability to exclude relevant variables and to select irrelevant ones is not zero. For this reason, we decide to estimate our PLTR model by relying on an extension of the Lasso that solves the above mentioned pitfall, i.e. the Adaptive Lasso estimator of Zou (2006). Indeed, the Adaptive Lasso has oracle properties as it penalizes more (less) the coefficients that are small (big) in magnitude. The corresponding penalty term is $P(\Theta) = \sum_{v=1}^{V} w_v |\theta_v|$ with $w_v = |\widehat{\theta}_v^{(0)}|^{-\nu}$, where $\widehat{\theta}_v^{(0)}$, $v = 1, ..., V$, are consistent initial estimators of the parameters,

and ν is a positive constant. The Adaptive Lasso estimators are obtained as

$$\widehat{\Theta}_{alasso}(\lambda) \;=\; \arg\min_{\Theta} -\mathcal{L}\left(\mathcal{V}_{i,1}^{(j)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta\right) + \lambda \sum_{v=1}^{V} w_v |\theta_v|. \tag{15}$$

In practice, we set the parameter ν to 1, the initial estimator $\widehat{\theta}_j^{(0)}$ to the value obtained from the logistic-ridge regression (Hoerl and Kennard, 1970), and the only free tuning parameter, λ, is found via 10-fold cross-validation. Besides, different estimation algorithms have been developed in the literature to estimate regression models with the adaptive lasso penalty (for a given value of λ): the quadratic programming technique (Shewchuk et al., 1994), the shooting algorithm (Zhang and Lu, 2007), the coordinate-descent algorithm (Friedman et al., 2010), and the Fisher scoring algorithm (Park and Hastie, 2007). Most of them are implemented in software like Matlab and R, and we rely here on the algorithm based on Fisher scoring. See the reference for more details on this optimization algorithm (McIlhagga, 2016).

3.2 PLTR under threshold effects: Monte Carlo evidence

In this subsection we assess the accuracy and interpretability (parsimony) of the PLTR model relatively to the traditional logistic regression and random forest in the presence of the threshold effects that were introduced in the Monte Carlo simulation setup of section 2.

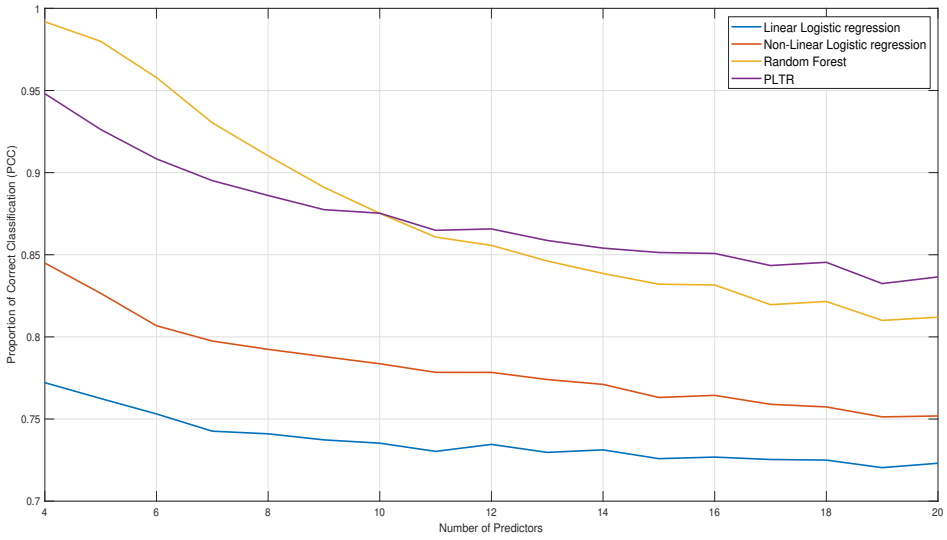

Figure 5: Comparison of performances under univariate and bivariate threshold effects: linear and non-linear logistic regressions, Random Forest and PLTR

We first assess the forecasting performance of this new credit-scoring method. The purple curve in Figure 5 represents the proportion of correct classification (PCC) for our PLTR method computed over the same test samples of length 2500 that were generated with the

DGP in (5)-(6). The conclusion is clear-cut: the PLTR method significantly outperforms the two versions of the logistic regression, i.e., with and without quadratic and interaction terms. When the number of predictors, p, is low, the PCC curve of the PLTR is lower than that of the random forest algorithm, but as p increases, the performance of the PLTR approaches and appears to even surpass that of random forest. For example, the PCCs are equal to 94.81 for our new method and 99.18 for the random forest with $p = 4$, against 83.65 and 81.20 for $p = 20$, respectively. In practice, the latter case is more realistic as credit scoring applications rely on quite a large set of predictors.

Moreover, performance is not the only essential criterion for credit-scoring managers. The other fundamental characteristic of a good scoring model is interpretability. But learning interpretable predictive models is challenging because interpretability and accuracy are generally two competing objectives. The first is favoured by simple models, while the latter by complex ones. In our case, if the results of the less performing logistic regressions can be immediately interpreted in terms of marginal effects, elasticities and even transformed in a transparent scorecard, those of the outperforming random forest are very difficult to interpret for two reasons. First, the forest relies on many trees, with many splits, which involve many complicated *if-then-else* rules. Second, the rules obtained from the trees are aggregated via the majority vote.

In this context, our PLTR method appears as a parsimonious solution to the tradeoff between performance and interpretability. Its good performance was emphasized in Figure 5. On top of that, the scoring decisions are simple to interpret through marginal effects (as well as elasticities and scorecards) similar to those of traditional logistic regression. This is facilitated by the simple decision rules obtained in the first step of the procedure from short-depth decision trees. Indeed, the skeleton of our PLTR is actually a logistic regression with binary indicators that account for endogenous univariate and bivariate threshold effects. The complete loan-decision process based on the PLTR method is illustrated in Figure 6. The input of the method includes all the predictive variables from the loan applicant, while the output is fundamentally the decision to accept or to reject the credit application based on the default risk of the person. Additionally, the mapping from the inputs to the output allows one to transform the internal set of rules of the PLTR into transparent feedback about the weaknesses and strengths of the application.

To give more insights about the interpretability, we compare our PLTR model and the random forest in the same Monte-Carlo setup as in Section 2 with p fixed to 20. Various measures to quantify the interpretability of a classifier have been proposed in the literature. One measure is the size of the set of decision rules needed for prediction. The fewer the rules, the easier to interpret the results. The size of a given rule in a decision set is a complementary measure. If the number of predicates in a rule is too large, it will loose its natural interpretability. Across the 100 simulations, the random forest registers an

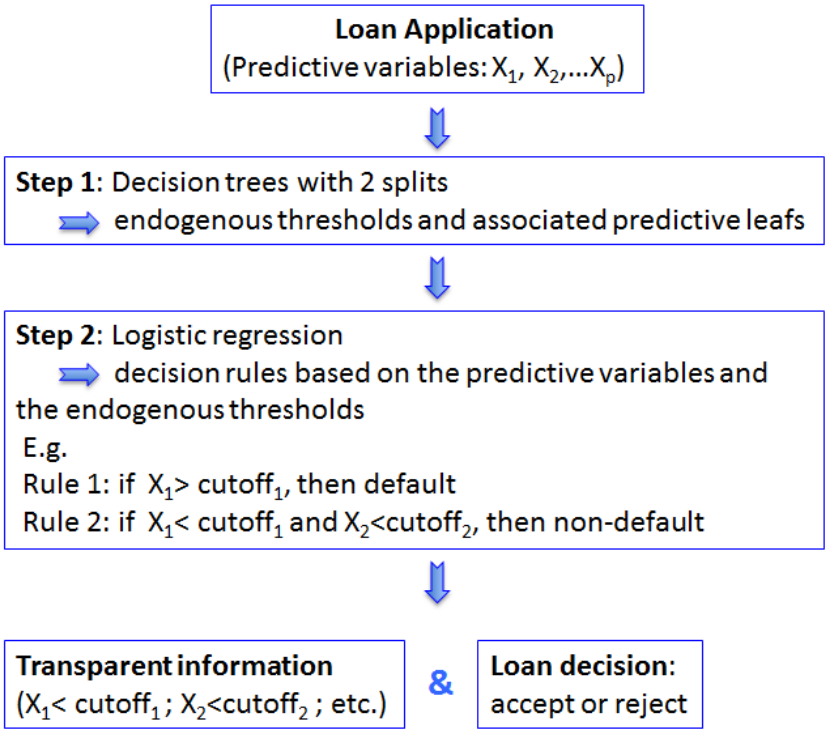

Figure 6: PLTR inference process

average number of 160.85 trees, each with an average number of 410.47 terminal nodes. This leads to a decision set of 410.47×160.85 binary decision variables or rules that can be used for prediction with this method. Across the same simulations, the average number of active binary decision variables in our penalized logistic regression is equal to 146.90.[10] Moreover, the number of predicates involved in each of these binary decision variables for our PLTR method varies between 1 and 2 by construction, whereas the maximum number of predicates in a rule of the random forest is 14.52 on average. Hence, our PLTR outperforms the random forest. In this sense, it is comparable to the non-linear logistic regression[11] in terms of interpretability.

Furthermore, marginal effects and elasticities can be easily obtained in the PLTR due to the linearity of the link function in (13) with respect to the parameters. On the one hand, this greatly simplifies significance testing as well as the implementation of out-of-sample exercises. On the other hand, this allows credit institutions to easily explain, in a transparent way, the main reasons behind a loan decision (see the example rules in Figure 6 that guarantee transparent information).

[10] Notice that for $p = 20$ predictors, the maximum number of binary variables is equal to $20 + \frac{20 \times 19}{2} = 210$. This result illustrates the selection operated through the adaptative lasso regression.

[11] The major difference between these two methods is the endogenous character of the thresholds that characterize variable interactions in our framework.

4 Model performance with a benchmark dataset

One could argue that the Monte Carlo simulations were designed to favor the PLTR method over the logistic regressions as the first implicitly handles univariate and bivariate threshold effects. In this section we evaluate the out-of-sample accuracy and interpretability of our method relative to that of its competitors by using a benchmark credit default dataset.

4.1 Data description and processing

To gauge the out of sample performance and to illustrate the interpretability of the PLTR method, we use a popular dataset provided by a financial institution for the Kaggle competition "Give me some credit", and which is often used in credit scoring applications (Baesens et al., 2003). The dataset includes several predictive variables and a binary response variable measuring default. The predictive variables provide information about the customers (age, monthly income, the number of dependents in family), and the application form (number of mortgage and real estate loans, the monthly debt payments, the total balance on credit cards, etc.). The dataset contains 10 quantitative predictors. See Table A.1 in Appendix A for the description of the variables in the dataset.

The number of instances in the dataset is equal to $150,000$ loans out of which $10,026$ defaults, leading to a prior default rate of 0.067. It is well known that class imbalance impedes classification: some classifiers may focus too much on the majority class and neglect the minority group (of interest). They could hence exhibit good overall performance despite poorly identifying the minority group, i.e. the borrowers that default.[12] A common solution to this issue consists in resampling methods such as undersampling or oversampling (as the SMOTE for example). Nonetheless, we choose not to resample the datasets as our PLTR method is designed to be an operational tool for credit-officers, that insures a good balance between predictive accuracy and interpretability.

Lastly, we need to prepare the raw dataset for use in this empirical application. To do so, we replace each missing value by the mean of the predictive variable. At the same time, we discuss data partitioning as it is an important step in our evaluation scheme. In particular, we use the so-called $N \times 2$-fold cross-validation of Dietterich (1998), which involves randomly dividing the dataset in two sub-samples of equal size. The first (second) part is used to build the model, while the second (first) part is used for evaluation. This procedure is repeated N times and the evaluation metrics are averaged. This method of evaluation produces more robust results compared to the classical single data partitioning. We set $N = 5$ for computational reasons.

[12] In the worst case, some classifiers could even misclassify all the members of the minority group and still exhibit good global performance.

4.2 Statistical measures of performance and interpretability

To evaluate the performance of each classifier we consider five accuracy measures: the area under the ROC curve (AUC), the Brier Score (BS), the Kolmogorov-Smirnov statistic (KS), the percentage of correctly classified (PCC) cases, and the Partial Gini Index (PGI). We rely on these indicators because they are the most popular evaluation metrics used in many empirical applications evaluating statistical models for credit scoring (Lessmann et al., 2015). Moreover, they are related to different facets of the predictive performance of scorecards, namely the accuracy of the scores as measured by the BS statistics, the quality of classification given by the PCC and KS statistics, and the discriminatory power assessed through the AUC and the PGI statistics. By using several statistics instead of a single one, we expect to obtain a robust and complete evaluation of the relative performances of the competing models.

The AUC tool evaluates the overall discriminatory performance of each model or classifier. It is a measure of the link between the False Positive Rate (FPR) and the True Positive Rate (TPR), each computed for every threshold between 0 and 1. The FPR (TPR) is the percentage of non-defaulted (defaulted) loans misclassified as defaulted (non-defaulted). Thus, the AUC reflects the probability that the occurrence of a randomly chosen bad loan is higher than the occurrence of a randomly chosen good loan.

The Gini Index is equal to twice the area between the ROC curve and the diagonal. Hence, like the AUC, it evaluates the discriminatory power of a classifier across several thresholds, with values close to one corresponding to perfect classifications. However, in credit scoring applications it is not realistic to study all possible thresholds. Informative thresholds are those located in the lower tail of the distribution of default probabilities (Hand, 2005). Indeed, only applications below a threshold in the lower tail could be granted a credit, which excludes high thresholds. The Partial Gini Index solves this issue by focusing on thresholds in the lower tail (Pundir and Seshadri, 2012). With x denoting a given threshold and $L(x)$ the function describing the ROC curve, the PGI is then defined as[13]

$$PGI = \frac{2\int_a^b L(x)dx}{(a+b)(b-a)} - 1. \tag{16}$$

The PCC is the proportion of loans that are correctly classified by the model. Its computation requires a discretization of the continuous variable of estimated probabilities of default. Formally, we need to choose a threshold π above (below) which a loan is classified as bad (good). In practice, the threshold π is fixed based on the cost of rejecting good customers/granting credits to bad customers. Since we do not have such information, we set this threshold to a value such that the predicted number of defaults in the learning sample is equal to the observed number of defaults.

[13]PGI within bounds $a = 0$ and $b = 1$ is equivalent to Gini Index. In the empirical applications, we evaluate the PGI within the $(0, 0.4)$ bounds as in Lessmann et al. (2015).

As for the Kolmogorov-Smirnov statistic, it is generally defined as the maximum distance between the estimated cumulative distribution functions of two random variables. In credit scoring applications, these two random variables measure the scores of good loans and bad loans, respectively (Thomas et al., 2002).

Lastly, the Brier Score (Brier, 1950) is defined as

$$BS = \frac{1}{n}\sum_{i=1}^{n}(\widehat{\Pr}(y_i = 1|x_i) - y_i)^2, \tag{17}$$

where $\widehat{\Pr}(y_i = 1|x_i)$ is the estimated probability of default and y_i is the target binary default variable. Note that it is the equivalent of the mean-squared error but it is designed for the case of discrete-choice models. All in all, the higher these indicators are the better the model is, except for the Brier Score for which a small value is better.

Regarding the interpretability of the scoring models, the size of the decision set and the average size of rules in a decision set are the criteria retained, as discussed in Subsection 3.2, to compare the interpretability of the PLTR and the random forest.

4.3 Statistical evaluation results

Table 1 presents the average value of each statistic across the 5×2 cross-validation test samples. We compare the performance of the PLTR to those of the traditional logistic regressions and random forest. Three different versions of the logistic regression are implemented: the simple linear logistic regression, its non-linear version which includes as additional variables, quadratic and interaction terms,[14] and a penalized version of this last model to avoid overfitting due to the large number of predictors. We use the adaptive Lasso penalty as described above.

Table 1: Average values of Statistical performance indicators: Kaggle dataset

Methods	AUC	PGI	PCC	KS	BS
Linear Logistic Regression	0.6983	0.3964	0.9082	0.3168	0.0576
Non-Linear Logistic Regression	0.7660	0.5255	0.9127	0.4173	0.0649
Non-Linear Logistic Regression + ALasso	0.8062	0.6102	0.9208	0.4751	0.0535
Random Forest	0.8529	0.6990	0.9260	0.5563	0.0500
PLTR	0.8568	0.7076	0.9247	0.5647	0.0496

Note: The non-linear logistic regression includes linear, quadratic and interaction terms. The method labelled "Non-Linear Logistic Regression + ALasso" corresponds to a penalized version of the non-linear logistic regression with the adaptive Lasso penalty.

The results displayed in Table 1 show that random forest performs better than the three versions of the logistic regression, and this holds for all statistical measures considered. This

[14] As already stressed, this non-linear model is the one that is generally used to capture non-linear effects in the framework of logistic regression.

is expected given that random forest is the benchmark method in terms of performance for credit scoring applications (Lessmann et al., 2015). In particular, the differences are more pronounced for the AUC, PGI and KS statistics. Most importantly, our PLTR method also outperforms the three versions of the logistic regression irrespective of the performance measure. This is particularly the case of AUC, PGI and KS metrics for which the dominance is stronger. This stylized fact is important as it suggests that our method has better predictive abilities compared to the benchmark models currently used by firms. The main message here is that combining decision trees with a standard model like logistic regression provides a valuable statistical modeling solution for credit scoring. In other words, the non-linearity captured by univariate and bivariate threshold effects obtained from short-depth decision trees can improve the out-of-sample performance of the traditional logistic regression.

The results in Table 1 also show that our method compares competitively to random forest. All statistical performance measures are of the same order or slightly better for our method. The main conclusion to draw from this illustration is hence that one should use our method instead of random forest, at least for this dataset. The rational of this assertion springs from the performance of the PLTR together with its parsimony that contrasts with the complexity underlying the prediction rule of random forest. Indeed, the average number of trees in the random forest across the 5×2 cross-validation test samples is equal to 173.9. These trees have on average $5,571.1$ terminal nodes, with a total of $5,571.1 \times 173.9$ binary variables for prediction (via the majority vote). By contrast, the average number of bivariate threshold effects selected by our penalized logistic regression is only equal to 40. More importantly, these bivariate threshold effects are easily interpretable because they arise from short-depth decision trees. In addition, the PLTR rules are built from only 2 predicates at most, whereas the rules from random forest are built from an average number of 32.15 predicates at most. These differences in terms of size of the decision set and size of the rules are the cost to pay in order to catch more non-linear effects, although such effects do not seem to play a significant role in this dataset.

It is worth stressing that the results above emphasize the importance of using different measures of performance when comparing several credit scoring methods. The conclusions may be slightly different according to the evaluation approach that is used. For example, if the unique objective is to obtain accurate probabilities of default (measured by Brier's score), the methods are almost equivalent. Nonetheless, the performance of the three versions of the logistic regression is much inferior in terms of discriminatory ability (measured by both AUC and PGI).

Lastly, in order to highlight the advantages of our method, especially in terms of interpretability, we report in Table 2 the 10 most important decision rules from the short-depth decision trees, which are selected by the Adaptive Lasso in the implementation of our PLTR method. These decision rules are those associated with the largest absolute values

of the marginal effects (averaged across individuals). A positive (negative) value of a given marginal effect provides information about the strength of increase (decrease) of the probability of default. We observe that three univariate threshold variables are selected, i.e., "NumberOfTime60-89DaysPastDueNotWorse < 0.5", "NumberOfTimes90DaysLate<0.5" and "RevolvingUtilizationOfUnsecuredLines<0.69814", the first one appearing as the most important in term of marginal effect. Referring to the description of this variable in Table A.1, we are able to infer that the probability of defaulting is 3.92% less important when the number of times a borrower has been between 60 and 89 days past due (but not worse in the last 2 years) is lower than 0.5 compared to the reference case when this number is higher than 0.5. Moreover, seven bivariate threshold effects are selected by the models as being important in explaining credit default. This kind of analysis that helps measuring through marginal effects the importance of the decision rules from the short-depth decision trees is an important added value of our PLTR model in term of interpretability.

5 Robustness across datasets

In this section, we evaluate the robustness of the above empirical results across datasets. To this end, we consider three popular additional datasets. The first one, named "Housing", is available in a SAS library and has been used by many authors for illustrative examples (Matignon, 2007). The second one labeled "Australian dataset" concerns credit card applications and is a UCI (University of California at Irvine) dataset provided by Quinlan, and one of the Credit Approval Databases which were used in the Statlog project.[15] Lastly, the third dataset labeled "Taiwan dataset" is also a UCI dataset that collects information about default payments in Taiwan.

The Housing dataset includes $5,960$ loans, $1,189$ of which had defaulted. Therefore, the prior default rate is 19.95%. In the Australian (Taiwan) dataset there are 690 ($30,000$) instances out of which 307 ($6,636$) defaults, leading to a prior default rate of 44.49% (22.12%). In the Housing dataset, there are 12 explanatory variables, two out of which are nominal. The Australian dataset includes 6 numerical and 8 nominal predictors. As for the Taiwan dataset, there are 23 predictors, nine out of which are nominal. Tables A.2 and A.3 display the list of the predictive variables for the Housing and the Taiwan datasets, respectively. We do not provide this information for the Australian dataset, as all attribute names and values have been changed to meaningless symbols to respect the confidentiality of the data.

We rely on the same $(N \times 2)$ comparison setup used for the benchmark Kaggle dataset, with $N = 5$. Table 3 displays the values of the five statistics retained for the comparison of the alternative models. For the Australian dataset, we remark that the two best performing

[15]The StatLog is an international project, which involves comparing the performances of machine learning, statistical, and neural network algorithms on data sets from real-world industrial areas including medicine, finance, image analysis, and engineering design.

Table 2: Decision rules and average marginal effects: Full sample Kaggle dataset

#	Decision Rules	Average marginal effects
1	"NumberOfTime60-89DaysPastDueNotWorse < 0.5"	-0.0392
2	"NumberOfTimes90DaysLate<0.5" & "RevolvingUtilizationOfUnsecuredLines<0.59907"	-0.0389
3	"NumberOfTimes90DaysLate<0.5" & "NumberOfTime60-89DaysPastDueNotWorse<0.5"	-0.0342
4	"NumberOfTime60-89DaysPastDueNotWorse<0.5" & "NumberOfTime30-59DaysPastDueNotWorse<0.5"	-0.0326
5	"NumberOfTimes90DaysLate<0.5"	-0.0326
6	"NumberOfTime60-89DaysPastDueNotWorse>=0.5" & "NumberOfTime60-89DaysPastDueNotWorse<1.5"	-0.0300
7	"RevolvingUtilizationOfUnsecuredLines>=0.69814" & "RevolvingUtilizationOfUnsecuredLines<1.001"	-0.0285
8	"RevolvingUtilizationOfUnsecuredLines<0.69814"	-0.0281
9	"NumberOfTimes90DaysLate<0.5" & "NumberOfTime30-59DaysPastDueNotWorse<0.5"	-0.0277
10	"NumberOfTimes90DaysLate<0.5" & "NumberOfTime30-59DaysPastDueNotWorse<0.5"	-0.0231

Note: The table provides the list of the decision rules associated with the 10 largest absolute values of the marginal effects (with respect to the probability of defaulting) derived from the PLTR model estimated using the full sample. See Table A.1 in Appendix A for a precise description of the variables.

models are the PLTR and the random forest, with similar values for all the five statistics[16]. This stylized fact confirms once again the relevance of our approach that leads to a performant model which inherits the ease of interpretation of the logistic regression. The same picture is observed for the Taiwan dataset with the PLTR model appearing as efficient as the random forest.

Lastly, for the Housing dataset, random forest and our PLTR method appear once again as the best performing models. However, in contrast to the results obtained for the other datasets, it now appears that random forest outperforms our method. But since our method is based on a compromise between statistical performance and interpretability, the previous mixed result is not so detrimental. Indeed, using the same arguments as above, the average number of active variables (univariate and bivariate threshold effects) in our penalized logistic regression is equal to 47.60, while random forest relies on average on 343.8×110.5 binary variables for prediction.[17] Hence, the PLTR is much more parsimonious.

Other results, available upon request, show that by relaxing the constraint of parsimony via the inclusion of tri-variate and quadri-variate threshold effects, the performance of our penalized logistic regression increases and reaches that of random forest. This suggests that complex non-linear relationships that go beyond univariate and bi-variate threshold effects are at stake in this dataset. In view of this result, it is important to stress that our article offers a highly flexible framework to credit risk managers, as they can tune their model according to the desired level of parsimony. The predictive performance can be significantly improved but at the cost of less interpretable results.

6 Economic evaluation

In the previous section we found that random forest and the PLTR introduced in this article have better statistical performances than logistic regressions and that out of the two, the PLTR also remains easily interpretable. A valuable key question for a credit risk manager is to what extent these statistical performance gains have a positive impact at a financial level for a credit company. The best way to evaluate these economic consequences is to calculate the amount of regulatory capital from the estimated default probability series. A similar comparison approach has been proposed by Hurlin et al. (2018) for LGD models. However, this task requires computing other parameters like the loss given default (LGD) and the exposure at default (EAD), and hence needs specific information about the consumers and

[16] For the Non-Linear Logistic Regression, we find that all fitted probabilities are higher than 0.6. Therefore, as we compute the PGI within $(0, 0.4)$, this statistic cannot be computed. Unlikely to happen in practice, this bad performance can also be observed through the high value of the BS statistic compared to those of the other methods.

[17] In this dataset we identify on average 110.5 trees in the forest, with an average number of terminal nodes equal to 343.8 for each tree. Furthermore, at most 18.82 predicates are used on average in the rules of the random forest against 2 at most for the PLTR model.

Table 3: Average values of Statistical performance indicators: Three datasets

Methods	AUC	PGI	PCC	KS	BS
Australian dataset					
Linear Logistic Regression	0.8998	0.5664	0.8374	0.7135	0.1186
Non-Linear Logistic Regression	0.6090		0.6067	0.2266	0.3921
Non-Linear Logistic Regression + Alasso	0.8866	0.5092	0.8214	0.6816	0.1333
Random Forest	0.9344	0.6246	0.8603	0.7523	0.0999
PLTR	0.9299	0.6370	0.8606	0.7425	0.1029
Taiwan dataset					
Linear Logistic Regression	0.6310	0.2099	0.7586	0.2506	0.2344
Non-Linear Logistic Regression	0.5963	0.0984	0.7035	0.1927	0.2965
Non-Linear Logistic Regression + Alasso	0.7596	0.5029	0.7871	0.3926	0.1447
Random Forest	0.7722	0.4924	0.8102	0.4177	0.1362
PLTR	0.7780	0.5156	0.7959	0.4257	0.1352
Housing dataset					
Linear Logistic Regression	0.7904	0.5508	0.8103	0.4450	0.1228
Non-Linear Logistic Regression	0.7965	0.5425	0.8239	0.4650	0.1199
Non-Linear Logistic Regression + Alasso	0.8113	0.5754	0.8217	0.4815	0.1125
Random Forest	0.9387	0.8157	0.9036	0.7455	0.0736
PLTR	0.9011	0.7341	0.8818	0.6694	0.0844

Note: The non-linear logistic regression includes linear, quadratic and interaction terms. The method labelled "Non-Linear Logistic Regression + ALasso" corresponds to a penalized version of the non-linear logistic regression with the Adaptive Lasso penalty.

the terms of the loans, which are not publicly available. Consequently, we compute another measure largely accepted in the literature, i.e. the misclassification cost (see Viaene and Dedene, 2004). This cost is estimated from Type 1 and Type 2 errors weighted by their probability of occurrence.

Formally, let C_{FN} be the cost associated to Type 1 error (the cost of granting credit to a bad customer) and C_{FP} the one for Type 2 error (e.g., the cost of rejecting a good customer). Thus, the misclassification error cost is defined as

$$MC = C_{FP}FPR + C_{FN}FNR, \tag{18}$$

with FPR the False Positive Rate and FNR the False Negative Rate. There is no consensus in the literature about how to best determine C_{FN} and C_{FP}. Two alternatives have been proposed. The first method fixes these costs based on previous studies (Akkoc, 2012). For example, West (2000) sets C_{FN} to 5 and C_{FP} to 1. The second method evaluates misclassification costs for different values of C_{FN} so as to test as many scenarios as possible (Lessmann et al., 2015). Even though there is no consensus on how to determine these costs, it is well known and accepted that the cost of granting a credit to a bad customer is higher than the opportunity cost of rejecting a good customer (see Thomas et al., 2002; West, 2000; Baesens et al., 2003, among others). We chose to follow the second approach

in order to assess the performance of the competing models. We fix C_{FP} at 1 without loss of generality (Hernandez-Orallo et al., 2011) and consider values of C_{FN} between 2 and 50.

Once these misclassification costs are computed[18], we set the linear logistic regression as benchmark and compute the financial gains or cost reduction (in percentage) generated by using a given method (the two versions of the non-linear logistic regression, random forest, PLTR) instead of the benchmark. This will enable us to assess the relative performance of our PLTR method from an economic point of view.

Figures A.1-A.4 in Appendix A display the cost reduction or financial gains for the four datasets considered above. First, except for the Taiwan and the Australian datasets for which the non-linear logistic regression leads to negative values for the cost reduction, all methods deliver positive cost reductions. This means that financial institutions relying on each of these methods rather than on the benchmark linear logistic regression should save the cost of rejecting (accepting) good (bad) applicants. In view of the large number of credits in bank credit portfolios, these gains could represent substantial savings for credit institutions. The fact that the non-linear logistic regression leads to an increase in costs compared to the linear logistic regression comes from the relatively high number of variables in the two datasets (14 and 23 in the Australian and Taiwan datasets, respectively). This leads to a proliferation of predictors (squares of the variables, cross-products of the variables), and therefore to overfitting. The penalized version of the non-linear logistic regression succeeds in dealing with this issue, which is materialized by positive values of the cost reductions in all cases, except for the Australian dataset.

Second, across all datasets, our PLTR method is among the most efficient in terms of cost reduction. Indeed, it appears to be the best on the Kaggle dataset. Precisely, the cost reduction relative to the linear logistic regression is on average equal to 18.06% for the PLTR method. This result also holds in the Taiwan dataset, with an average cost reduction equal to 22.29% for PLTR. Remark that random forest leads to lower cost reduction for these two datasets, with an average cost reduction of 13.09% (11.51%) for the Kaggle (Taiwan) one. This means that although random forest has high global predictive accuracy, as given by the proportion of correct classification (see Tables 1 and 3), it fails to some extent to detect bad customers, which leads to a relative increase in costs due to more false negatives. For the other two datasets (Australian and Housing) random forest is the best performing model, followed by our PLTR method. With the Australian dataset, the average cost reduction of random forest (PLTR) method is equal to 22.71% (14.89%). As for the Housing dataset, the average values are equal to 44.56% and 38.69% for random forest and PLTR, respectively.

To conclude, the results show that our credit-scoring method compares favourably to the state-of-the-art random forest algorithm not only on the statistical side, but also from an economic viewpoint. Indeed, the cost reduction engendered by the PLTR is higher than

[18]The misclassification costs are computed from tests samples.

the one of random forest for two datasets. For the other two datasets, the random forest is better. Remember, however, that PLTR is more parsimonious than random forest, allowing for simple interpretation of results. The results for these last two datasets hence highlight the cost of PLTR's interpretability in terms of performance (relatively to random forest). Note also that for all models and datasets, the cost reductions are highly stable across the different values of C_{FN}.

7 Conclusion

The benchmark credit scoring model is still the logistic regression, which by design leads to conclusions that are easy to disclose and hence interpretable for clients, credit risk managers, and regulators. Since the Big Data revolution and the renewed interest in statistics and machine learning, several papers advocate the use of sophisticated ensemble methods like random forest, that are shown to outperform the traditional logistic regression. Nevertheless, the prediction rule underlying random forests lacks parsimony and can be less relevant in credit scoring applications where decision makers need simple and interpretable forecasting rules.

Recognizing that traditional logistic regression underperforms random forest due to its pitfalls in modeling non-linear (threshold and interaction) effects, this article introduces a penalized logistic tree regression (PLTR) with predictive variables given by easy-to-interpret endogenous univariate and bivariate threshold effects. These effects are quantified by dummy variables associated to leaf nodes of short-depth decision trees built with couples of the original predictive variables. Our main objective is to combine decision tree (from the field of machine learning) and logistic regression (from the field of econometrics) to get the best of both worlds: a performing and interpretable credit scoring model.

We show through Monte Carlo simulations and four empirical applications that the PLTR has good predictive power while remaining easily interpretable. More precisely, using several metrics to evaluate both the accuracy and the interpretability of credit models, we show that our new method outperforms traditional linear and non-linear logistic regression while being competitive compared to the difficult to interpret random forest. We also evaluate the economic benefit of using our PLTR method through the so-called misclassification costs analysis. We find that beyond parsimony, our method leads to significant reduction in misclassification costs compared to the benchmark logistic regression while being competitive with respect to the state-of-the-art random forest algorithm. Relying on an efficient trade-off between performance and interpretability, the PLTR method introduced in this article hence proves to be a useful tool for credit risk managers.

A Appendix A: Additional Figures and Tables

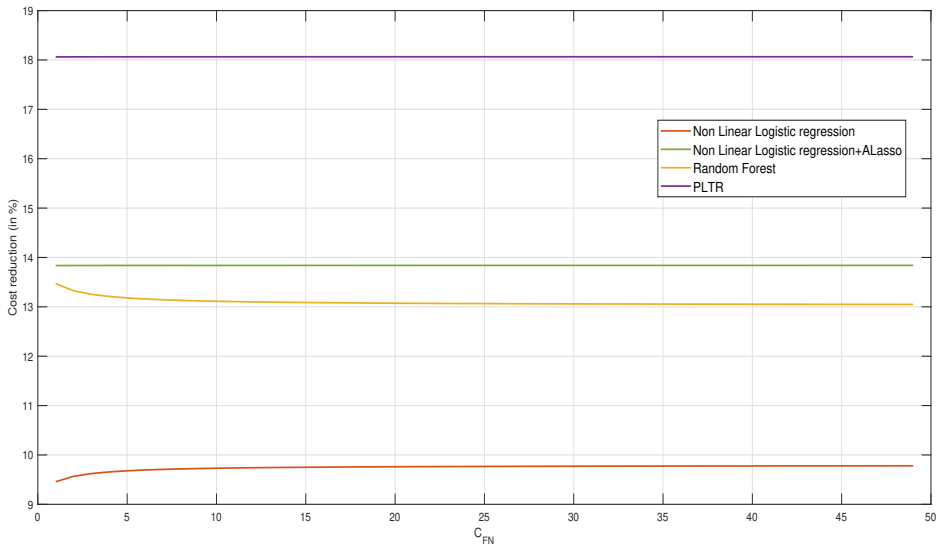

Figure A.1: Economic Evaluation for the Kaggle dataset

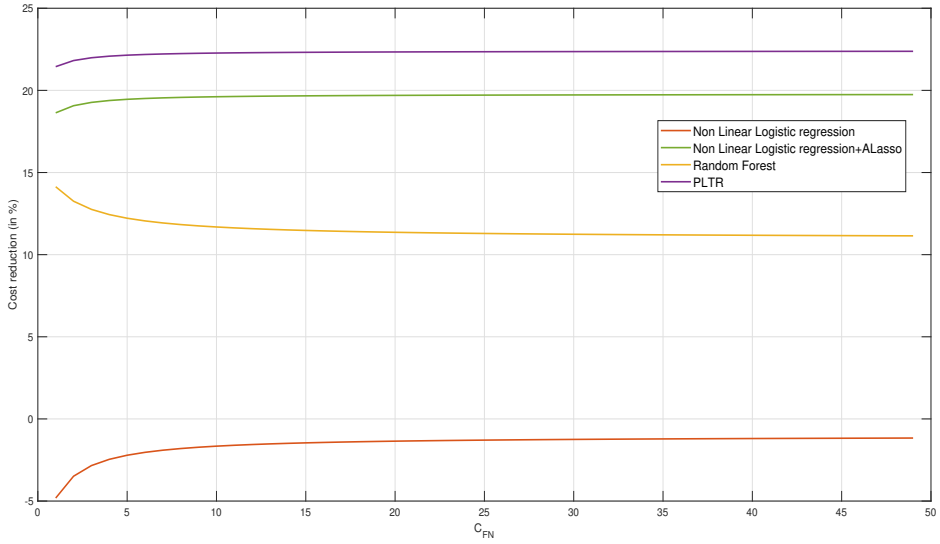

Figure A.2: Economic Evaluation for the Taiwan dataset

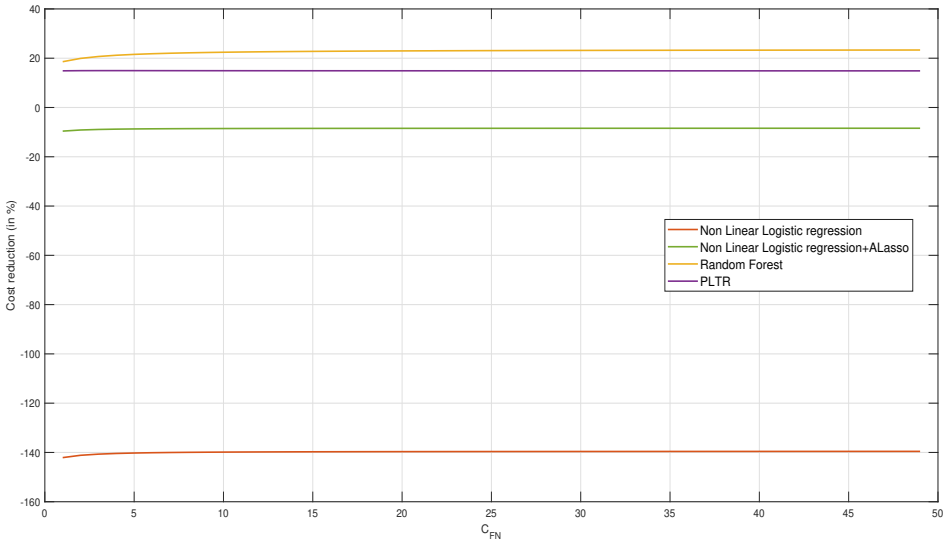

Figure A.3: Economic Evaluation for the Australian dataset

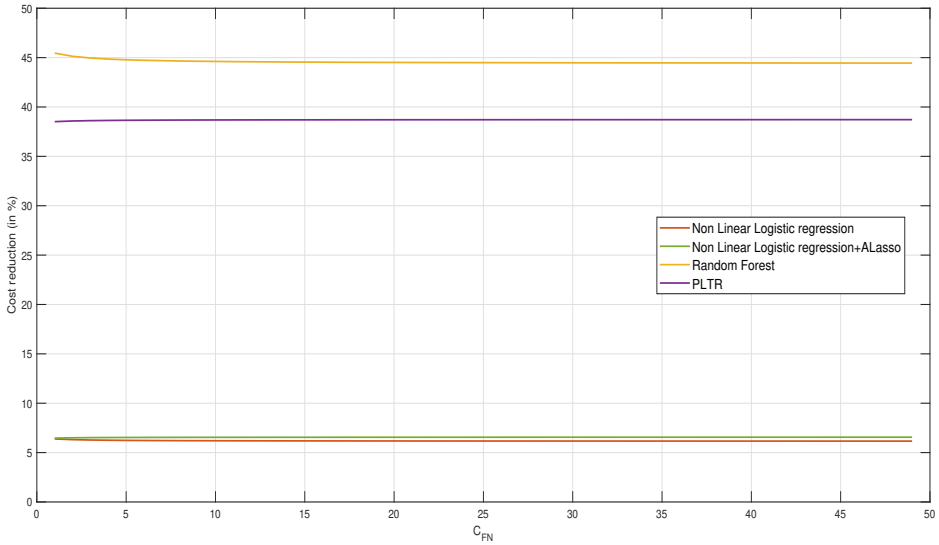

Figure A.4: Economic Evaluation for the Housing dataset

Table A.1: Description of the variables in the Kaggle dataset "Give me some credit"

Variable	Type	Description
SeriousDlqin2yrs	Binary	The person experienced 90 days past due delinquency or worse (Yes/No)
RevolvingUtilizationOfUnsecuredLines	Percentage	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits
Age	Interval	Age of the borrower (in years)
NumberOfTime30-59DaysPastDueNotWorse	Interval	Number of times a borrower has been between 30 and 59 days past due but not worse in the last 2 years
DebtRatio	Percentage	Monthly debt payments, alimony and living costs over the monthly gross income
MonthlyIncome	Interval	Monthly Income
NumberOfOpenCreditLinesAndLoans	Interval	Number of open loans (like car loan or mortgage) and credit lines (credit cards)
NumberOfTimes90DaysLate	Interval	Number of times a borrower has been 90 days or more past due
NumberRealEstateLoansOrLines	Interval	Number of mortgage and real estate loans including home equity lines of credit
NumberOfTimes60-89DaysPastDueNotWorse	Interval	Number of times a borrower has been between 60 and 89 days past due but not worse in the last 2 years
NumberOfDependents	Interval	Number of dependents in family excluding themselves (spouse, children, etc...)

Table A.2: Description of the variables in the Housing dataset

Variable	Type	Description
Bad	Binary	Whether the consummer had a default on the loan (1) or not (0)
Clage	Interval	Age of the oldest trade (in months)
Clno	Interval	Number of trades
Debtinc	Interval	Ratio of debt to income
Delinq	Interval	Number of neglectful trades
Derog	Interval	Number of major derogatory reports
Job	Nominal	Professional categories
Loan	Interval	Amount of the loan
Mortdue	Interval	Amount due on the mortgage
Ninq	Interval	Number of recent credits inquired
Reason	Binary	Whether the loan is for debt consolidation (DebtCon) or home improvement (HomeImp)
Value	Interval	Current property value
Yoj	Interval	Number of years at the present job

Table A.3: Description of the variables in the Taiwan dataset

Variable	Type	Description
Y	Binary	default payment (Yes = 1, No = 0)
X1	Quantitative	Amount of the given credit (NT dollar)
X2	Binary	Gender (1 = male; 2 = female)
X3	Nominal	Education (1 = graduate school; 2 = university; 3 = high school; 4 = others)
X4	Nominal	Marital status (1 = married; 2 = single; 3 = others)
X5	Quantitative	Age (year)
X6-X11	Nominal	X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above
X12-X17	Quantitative	Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005
X18-X23	Quantitative	Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .; X23 = amount paid in April, 2005

References

Acharya, V. V., Pedersen, L. H., Philippon, T., and Richardson, M. (2017). Measuring systemic risk. *Review of Financial Studies*, 30(1):2–47.

Akkoc, S. (2012). An empirical comparison of conventional techniques, neural networks and the three stage hybrid adaptive neuro fuzzy inference system (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. *European Journal of Operational Research*, 222(1):168–178.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54:627–635.

Bauweraerts, J. (2016). Predicting bankruptcy in private firms: Towards a stepwise regression procedure. *International Journal of Financial Research*, 7(2):147–153.

Berger, A. N., Frame, W. S., and Miller, N. H. (2005). Credit scoring and the availability, price, and risk of small business credit. *Journal of Money, Credit and Banking*, 37(2):191–222.

Blöchlinger, A. and Leippold, M. (2006). Economic benefit of powerful credit scoring. *Journal of Banking and Finance*, 30:851–873.

Bracke, P., Datta, A., Jung, C., and Sen, S. (2019). Machine learning explainability in finance: an application to default risk analysis.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26:123–140.

Breiman, L. (2001). Random forest. *Machine Learning*, 45:5–32.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1).

Bühlmann, P. (2012). Bagging, boosting and ensemble methods. *Handbook of Computational Statistics: Concepts and Methods, 2nd edition*, pages 985–1022.

Cardell, N. S. and Steinberg, D. (1998). The hybrid-CART logit model in classification and data mining. *Working paper, Salford-System*.

Chan, K. Y. and Loh, W. Y. (2004). Lotus: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics*, 13(4):826–852.

Deng, H. (2019). Interpreting tree ensemble with intrees. *International Journal of Data Science and Analytics*, 7(4):277–287.

Desai, V. S., Crook, J. N., and Overstreet Jr, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, 95(1):24–37.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Engle, R., Jondeau, E., and Rockinger, M. (2015). Systemic risk in Europe. *Review of Finance*, 19(1):145–190.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research*, 210(2):368–378.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Frost, J., Gambacorta, L., Huang, Y., Shin, H. S., and Zbinden, P. (2019). Bigtech and the changing structure of financial intermediation.

Grennepois, N., Alvirescu, M., and Bombail, M. (2018). Using random forest for credit risk models. *Deloitte Risk Advisory*.

Grennepois, N. and Robin, E. (2019). Explain artificial intelligence for credit risk management. *Deloitte Risk Advisory*.

Hand, D. J. (2005). Good practice in retail credit scorecard assessment. *Journal of the Operational Research Society*, 56(9):1109–1117.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning. data mining, inference and prediction. *Springer, New York*.

Henley, W. and Hand, D. (1996). A k-nearest-neighbour classifier for assessing consumer credit risk. *The Statistician*, 45(1):77–95.

Henley, W. E. and Hand, D. J. (1997). Construction of a k-nearest neighbour credit-scoring system. *IMA Journal of Mathematics Applied in Business and Industry*, 8:305–321.

Hernandez-Orallo, J., Flach, P., and Ferri, C. (2011). Brier curves: A new cost-based visualisation of classifier performance.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hurlin, C., Leymarie, J., and Patin, A. (2018). Loss functions for loss given default model comparison. *European Journal of Operational Research*, 268(1):348–360.

Hurlin, C. and Pérignon, C. (2019). Machine learning et nouvelles sources de données pour le scoring de crédit.

Landwehr, N., Hall, M., and Frank, E. (2005). Logistic model trees. *Machine Learning*, 59:161–205.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247:124–136.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, pages 4765–4774.

Matignon, R. (2007). *Data Mining Using SAS Enterprise Miner*.

McIlhagga, W. H. (2016). penalized: A matlab toolbox for fitting generalized linear models with penalties.

Molnar, C. (2019). *Interpretable machine learning*. Lulu.com.

Óskarsdóttir, M., Bravo, C., Sarraute, C., Vanthienen, J., and Baesens, B. (2019). The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics. *Applied Soft Computing*, 74:26–39.

Paleologo, G., Elisseeff, A., and Antonini, G. (2010). Subagging for credit scoring models. *European Journal of Operational Research*, 201(2):490–499.

Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society*, 69(4):659–677.

Pundir, S. and Seshadri, R. (2012). A novel concept of partial Lorenz curve and partial Gini index. *International Journal of Engineering, Science and Innovative Technology*, 1:296–301.

Rochet, J.-C. (1992). Capital requirements and the behaviour of commercial banks. *European Economic Review*, 36(5):1137–1170.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686.

Shewchuk, J. R. et al. (1994). An introduction to the conjugate gradient method without the agonizing pain.

Steenackers, M. and Goovaerts, J. (1989). A credit scoring model for personal loans. *Insurance: Mathematics and Economics*, 8(1):31–34.

Stein, R. M. (2005). The relationship between default prediction and lending profits: integrating roc analysis and loan pricing. *Journal of Banking and Finance*, 29:1213–1236.

Stein, R. M. and Jordao, F. (2003). What is a more powerful model worth? *Technical Report #030124, Moodys KMV, New York*.

Stepanova, M. and Thomas, L. C. (2001). Phab scores: Proportional hazards analysis behavioural scores. *The Journal of the Operational Research Society*, 52(9):1007–1016.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International journal of forecasting*, 16(2):149–172.

Thomas, L. C., Edelman, D. B., and Crook, J. N. (2002). *Credit scoring and its application*.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288.

Viaene, S. and Dedene, G. (2004). Cost-sensitive learning and decision making revisited. *European Journal of Operational Research*, 166:212–220.

Wang, W. (2012). How the small and medium-sized enterprises' owners' credit features affect the enterprises' credit default behavior? *E3 Journal of Business Management and Economics*, 3(2):90–95.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, 27(11-12):1131–1152.

Yobas, M. B., Crook, J. N., and Ross, P. (2000). Credit scoring using neural and evolutionary techniques. *IMA Journal of Mathematics Applied in Business and Industry*, 11:111–125.

Zhang, H. H. and Lu, W. (2007). Adaptive lasso for Cox's proportional hazards model. *Biometrika*, 94(3):691–703.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, 67(2):301–320.