Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects

Article · July 2017

CITATIONS
0

READS
2,470

3 authors, including:

Hue Sullivan
Université d'Orléans
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Sessi Tokpavi
Université d'Orléans
34 PUBLICATIONS **178** CITATIONS
SEE PROFILE

Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects

Dumitrescu, Elena[*], Hue, Sullivan[†], Hurlin, Christophe[‡], Tokpavi, Sessi[§]

February, 2018

Abstract

Decision trees and related ensemble methods like random forest are state-of-the-art tools in the field of machine learning for predictive regression and classification. However, they lack interpretability and can be less relevant in credit scoring applications, where decision-makers and regulators need a transparent linear score function that usually corresponds to the link function in logistic regressions. In this paper, we propose to improve the framework of logistic regression by using information from decision trees. Formally, rules extracted from various short-depth decision trees built with different sets of predictive variables (singletons and couples) 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 while preserving its simple interpretation. Applications using simulated and real data sets for credit scoring show that the new method outperforms traditional logistic regression. Moreover, it compares competitively to random forest, while providing an interpretable scoring function.

Keywords: Credit risk scoring, Logistic regression, Penalization, Decision trees, Threshold effects, Random forest, Parsimony.

[*]elena.dumitrescu@parisnanterre.fr, EconomiX, Paris Nanterre University, France.
[†]sullivan.hue@univ-orleans.fr, LEO-CNRS, University of Orléans, France.
[‡]christophe.hurlin@univ-orleans.fr, LEO-CNRS, University of Orléans, France.
[§]Corresponding author, sessi.tokpavi@univ-orleans.fr, LEO-CNRS, University of Orléans, France.

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 costs and benefits of credit scoring have been analyzed in the literature in terms of credit availability and default risk. For instance, Berger, Frame, and Miller (2005) suggest that small business credit scoring increases access to credit for relatively risky credits that tend to pay relatively high prices for funding. Their conclusion arises from the observation of greater loans associated with higher risks for small business credits under $100K offered by banks that lean their lending decision on a credit scoring system. From an economic viewpoint, these stylized facts rationalize credit scoring, given that small and medium enterprises are major contributors to the strength of local economies. Other papers question the impact of credit scoring in terms of profitability among lenders (Stein and Jordao, 2003; Stein, 2005; Blöchlinger and Leippold, 2006). For example, Blöchlinger and Leippold (2006) analyze the economic benefit of credit scoring and observe that even small differences in the quality of credit scoring models can lead to economically significant differences in the performance of credit portfolios. Moreover, an improvement in one bank's credit scoring model lowers the profits of its competitors.

Credit scoring is also important for banks and regulators in the financial sphere, as the Basel III (Basel Committee on Banking Supervision, 2011) reinforces capital requirements for the coverage of credit risk. The level of capital requirements is linked to the banks overall credit risk and could be overstated in absence of a performing credit scoring model. Its main consequence would be a rise in banks' funding cost and a potential increase in credit cost and/or a decrease in loan volume (Rochet, 1992). This is a subject of concern both for banks (micro-economic level) and governments (macro-economic level). Capital requirements can also be understated if banks in lack of a credit scoring model underestimate the true level of credit risk. Regulators dislike this scenario since the banking system becomes less resilient to financial crises and more exposed to systemic events if it is not capitalized enough (Engle, Jondeau, and Rockinger, 2015; Acharya, Pedersen, Philippon, and Richardson, 2017).

These theoretical and empirical reasons for using credit scoring explain why a large number of works propose models that predict credit worthiness. These models use borrower-specific information to estimate the probability that they will default in the future. Early

attempts to estimate borrowers default probability rely on predictive statistical methods and regressions. Altman (1968) applied discriminant analysis to relate financial ratios to the probability of default for corporates. Steenackers and Goovaerts (1989) propose a scoring model for personal loans based on logistic regression. Stepanova and Thomas (2001) compare the power of proportional hazard models and logistic regression in predicting defaults.

Machine learning techniques are also shown to be successful in forecasting credit scores. Examples are the k-nearest neighbor (Henley and Hand, 1996, 1997), neural networks (Desai, Crook, and Overstreet Jr, 1996; West, 2000; Yobas, Crook, and Ross, 2000), decision trees (Yobas, Crook, and Ross, 2000), and support vector machine (Baesens, Gestel, Viaene, Stepanova, Suykens, and Vanthienen, 2003). The empirical results are however mixed. For instance, Baesens, Gestel, Viaene, Stepanova, Suykens, and Vanthienen (2003) compare the performance of these models or algorithms by using data for major financial institutions in Benelux and UK, and they find, among others, that support vector machine appears to perform best according to the area under the receiver operating characteristic (ROC) curve, albeit some simple models (logistic regression, discriminant analysis) are also successful in predicting failure probability.

More recently, with the revolution of big data and its uncontroversial positive effect on businesses, there is a renewed interest in some statistical and machine learning algorithms introduced in the early 2000s. The two most widely used ones are Bagging (Breiman, 1996) and Boosting (Schapire, Freund, Bartlett, and Lee, 1998), and their domains of applications 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 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. While Bagging uses the aggregation of predictions obtained from bootstrapped samples of the original sample, Boosting is an iterative method that uses data subsamples to train the learner. In particular individuals that were misclassified in the previous iteration are given more weight so that in the subsequent iteration the learner focuses more on them during training. For a review of Bagging and Boosting methods see Hastie, Tibshirani, and Friedman (2001) and Bühlmann (2012).

Applications of ensemble methods to credit scoring can be found in Finlay (2011), Paleologo, Elisseeff, and Antonini (2010), and Lessmann, Baesens, Seow, and Thomas (2015). Finlay (2011) compares several multiple classifiers (ensemble methods) in terms of their predictive performances in discriminating between good and bad borrowers. He shows that

bagging and boosting methods outperform simple classifiers or models among wich the logistic regression. Paleologo, Elisseeff, and Antonini (2010) propose an ensemble classification technique called subagging which is shown in an empirical application on credit scoring to improve significantly the performance of base classifiers (kernel support vector machines, nearest neighbors, decision trees, Adaboost). Similar conclusions arise from the intensive empirical applications in Lessmann, Baesens, Seow, and Thomas (2015). Relying on the average robust performance of each method in a large number of credit-scoring data sets, they found, among others, that random forest (Breiman, 2001), i.e. the randomized version of bagged decision trees, outperforms logistic regression. From an economic viewpoint, they compare the classification costs, measured by the weighted sum of the false positive rate and the false negative rate, and show that random forest implies a larger drop in cost reduction relative to the logistic regression.

Nevertheless, as decision rules from random forest arise from the aggregation of individual (non-pruned) decision tree rules, they can be less relevant in credit scoring applications, where decision makers and regulators need parsimonious and interpretable scorecards like those based on logistic regression.

The objective of this article is precisely to propose a simple extension of the logistic regression that improves the predictive performance of the baseline model, i.e. increase it to the same order of magnitude as that of random forest, while preserving the simple interpretability of logistic regression. To do so, we first recognize that ensemble methods like random forest consistently outperform logistic regression because the latter method fails in fitting non linear effects. Indeed, random forest benefits from the recursive partitioning underlying decision trees and hence, by design, accommodates univariate and multivariate threshold effects. We then introduce a new credit scoring modeling approach that is based on a simple logistic regression with predictors extracted from decision trees. Formally, the predictors in the logistic regression are chosen to be the rules extracted from various short-depth decision trees built with different sets of variables including the original predictive variables (singletons and couples). We then use the adaptive Lasso logistic regression (Zou, 2006; Friedman, Hastie, and Tibshirani, 2010), a penalized version of the classical logistic regression, to handle the large number of decision trees rules and to proceed to variables selection. Applications based on simulated and real data sets show that our new method, entitled Penalized Logit Tree Regression, hereafter PLTR, outperforms traditional logistic regression. Moreover, it compares competitively to random forest while providing an interpretable scoring function. This conclusion holds for the various predictive accuracy indicators in Lessmann, Baesens, Seow, and Thomas (2015).

Our approach can be considered as a systematization of a common practice in the de-

ployment 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 heuristic methods of discretization. The merit of our contribution is to propose a systematic approach to the modeling of such non-linear effects by using short-depth decision trees. Our PLTR methodology, although similar in spirit, contrasts with the hybrid CART-Logit model of Cardell and Steinberg (1998). To introduce multivariate threshold effects in logistic regression, they consider non-pruned single decision tree, while our goal is to achieve simple model interpretation by using short depth decision trees from singleton or couples of variables with limited splits. Moreover, they do not control at all for predictors inflation through penalization. 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, Hall, and Frank (2005).

The rest of the article is structured as follows. Section 2 presents the logistic regression model, and shows through Monte Carlo simulations that its predictive performance shrinks in the presence of univariate and multivariate threshold effects. In Section 3 we present the new methodology for credit scoring while Sections 4 and 5 are devoted to empirical applications. The last Section concludes the article.

2 Logistic regression under threshold effects

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. This probability is compared to a threshold value π by using the following rule: reject the loan if $\Pr(y_i = 1 \,|x_i) > \pi$, accept it otherwise.

For corporate credit risk scoring, the candidate predictive variables $x_{i,j}$, $j = 1, ..., p$, 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). More precisely, variables such as cash-flows and profits, debt ratios and quick ratio are generally used to predict the default of large corporate firms. These variables are also shown to be important determinants of the default prediction for small and medium enterprises (SMEs). 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 SMEs, specific variables related to the financial strength of the firm's owner are also shown to be important (Wang, 2012). These variables include, among others, 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. These latter variables combined with socio-demographic factors are also important for the credit risk analysis of retail loans.

From the specification viewpoint, logistic regression models the conditional probability $\Pr(y_i = 1 | x_i) \equiv \Pr(x_i; \beta)$ by

$$\log\left\{\frac{\Pr(x_i;\beta)}{1-\Pr(x_i;\beta)}\right\} = \eta(x_i;\beta), \tag{1}$$

with $\eta(x_i;\beta)$ the so-called index function defined as

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

where $\beta = (\beta_0, \beta_1, ..., \beta_p) \in \mathbb{R}^{p+1}$ is an unknown vector of parameters. In terms of probability, this specification can be rewritten as

$$\Pr(x_i;\beta) = F(\eta(x_i;\beta)) = \frac{1}{1+\exp(-\eta(x_i;\beta))}, \tag{3}$$

with $F(.)$ the logistic function. The estimator $\widehat{\beta}$ is obtained by maximizing the convex log-likelihood function

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} y_i \log\{F(\eta(x_i;\beta))\} + (1-y_i)\log\{1-F(\eta(x_i;\beta))\}, \tag{4}$$

with n the sample size. Under some 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 is its ease of interpretation. Indeed, this model searches for a single linear decision boundary in the predictors space. The core assumption for finding it is that the probability of default, $\Pr(x_i;\beta)$, has a logistic functional form whose argument is the index $\eta(x_i;\beta)$, which 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

$$\frac{\partial \Pr(y_i = 1 | x_i)}{\partial x_{i,j}} = \beta_j \frac{\exp(\eta(x_i;\beta))}{1+\exp(\eta(x_i;\beta))}, \tag{5}$$

with estimates obtained by replacing β by $\widehat{\beta}$. Thus, a predictive variable with positive (negative) significant estimated coefficient has a positive (negative) impact on the 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", wich implies the existence of an endogenous 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 in the application of credit scoring to capture non linear effects is to introduce quadratic and interaction terms in the index function $\eta(x_i;\beta)$. We advocate that such a practice is not successful when 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

$$\eta(x_i;\Theta) = \beta_0 + \sum_{j=1}^{p} \beta_j (x_{i,j} \leq \gamma_j) + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \beta_{j,k} (x_{i,j} \leq \delta_j)(x_{i,k} \leq \delta_k), \tag{6}$$

where $\Theta = (\beta_0, \beta_1, ..., \beta_p, \beta_{1,2}, ..., \beta_{p-1,p})'$ 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, whose values are randomly selected from the support of each predictive variable generated, while excluding data below (above) the first (last) decile. Thus, for each individual, we obtain the probability of default from (3). The target binary variable y_i is simulated as

$$y_i = \begin{cases} 1 & \text{if} \quad \Pr(y_i = 1 \,|x_i) > \pi \\ 0 & \text{else,} \end{cases} \tag{7}$$

with π equal to the median value of generated probabilities. Note that with the above specification one generates a sample of data with default events that arise from univariate and bivariate threshold effects. At each replication, we divide the simulated sample into two sub-samples of equal size. The first (second) sub-sample is designated as the learning (test) sample. We next estimate two different models on the learning sample. The first is the classical logistic regression with linear effects whose index is given in (2). The second model is based on a non linear index function that incorporates quadratic and interaction terms, i.e.,

$$\eta(x_i;\Theta) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} + \sum_{j=1}^{p} \gamma_j x_{i,j}^2 + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \delta_{j,k} x_{i,j} x_{i,k}, \tag{8}$$

where $\Theta = (\beta_0, \beta_1, ..., \beta_p, \gamma_1, ..., \gamma_p, \delta_{1,2}, ..., \delta_{p-1,p})'$ is the unknown vector of parameters. This specification is the one that is generally used in empirical applications of credit scoring to capture non linear effects. Our goal here is to show that it fails to model non-linearity in presence of univariate and bivariate threshold effects (see equation 6). The performance of these two models is evaluated on the test sample. We consider here the probability of correct classification (PCC) as the evaluation criterion. Figure 1 gives the average value of the PCCs for both models over 50 simulations and for different values of $p = 4, ..., 10$, the number of predictive variables.

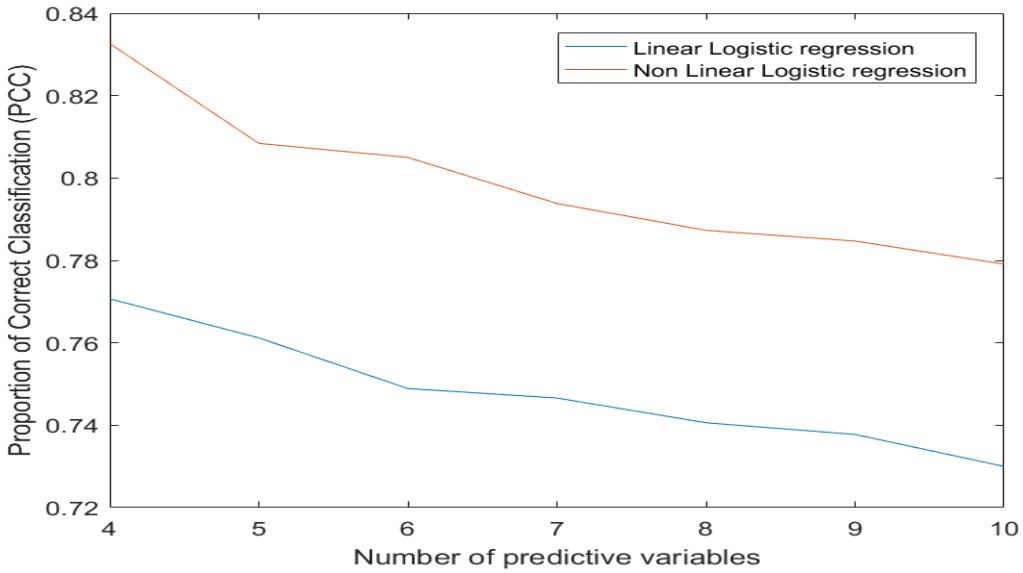

Figure 1: Performance of logistic regressions under univariate and bivariate threshold effects

Two patterns emerge from Figure 1. First, 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 $p = 10$ the proportions of correct classification are only equal to 73% and 77.91% for the first and second model, respectively. Second, the model with linear index has the lowest predictive power of all. Hence, adding quadratic and interaction terms improves the predictive power, but the overall performance remains low when the number of predictors increases.

Ensemble or aggregation methods for decision trees like random forest are shown to be more successful in such a framework. The out-performance of random forest arises from the non linear "if-then-else" rules underlying decision tree. 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, such that the individuals with the same value of the binary target variable "y_i" are grouped together. Formally, for a given tree with index l, the algorithm proceeds as follows. Let $\mathcal{D}_{m,l}$ be the data at a given node or iteration m for the tree l. We denote $\theta_{m,l} = (j_{m,l}, t_{m,l,j})$ a candidate split, with $j_{m,l} = 1, ..., p$ referring to a given predictive variable and $t_{m,l,j}$ 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[1]

$$\mathcal{D}_{m,l,1}(\theta_{m,l}) = (x_i, y_i) \mid x_{i,j} \leq t_{m,l,j}, \tag{9}$$

$$\mathcal{D}_{m,l,2}(\theta_{m,l}) = (x_i, y_i) \mid x_{i,j} > t_{m,l,j}. \tag{10}$$

The estimate $\widehat{\theta}_{m,l}$ of the parameter $\theta_{m,l}$ is found such as

$$\widehat{\theta}_{m,l} = (\widehat{j}_{m,l}, \widehat{t}_{m,l,j}) \;=\; \arg\max_{\theta_{m,l}} \left\{ \mathcal{H}(\mathcal{D}_{m,l}) - \mathcal{H}\Big(\mathcal{D}_{m,l,1}(\theta_{m,l}), \mathcal{D}_{m,l,2}(\theta_{m,l})\Big) \right\}, \tag{11}$$

where $\mathcal{H}(\mathcal{D}_{m,l})$ is a measure of diversity in the sample $\mathcal{D}_{m,l}$ and $\mathcal{H}(\mathcal{D}_{m,l,1}(\theta_{m,l}), \mathcal{D}_{m,l,2}(\theta_{m,l}))$ is the same measure averaged across the two sub-samples $\mathcal{D}_{m,l,1}$ and $\mathcal{D}_{m,l,2}$. Diversity is usually approximated by the so-called Gini criterion. $\widehat{\theta}_{m,l}$ appears hence as the value of $\theta_{m,l}$ that reduces diversity the most after 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. 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 $(\widehat{j}_{m,l}, \widehat{t}_{m,l,j})$, with $\widehat{j}_{m,l}$ the index of the variable "income" and $\widehat{t}_{m,l,j} = 33270.53$. The second iteration ($m = 2$) also includes "age" and "education" for a further refinement. The process ends with a total number of splits equal to 5, and 6 leaf nodes labeled 10, 11, 12, 13, 4 and 7, respectively. The distribution of the two classes (1='default', 0='non default') is given for each leaf node. For instance, the 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 old. Let $\Theta_l = (\theta_{m,l}, m = 1, ..., M_l)$ be the set of parameters for tree l, where M_l is the total number of splits for this tree. Denote $h_l(x_i; \widehat{\Theta}_l) \equiv h_l(x_i)$ the predicted value of y_i for individual i. It corresponds to the most frequent class of the leaf node individual i belongs to. For example in Figure 2 the predicted value $h_l(x_i)$ is equal to 1 for an individual that belongs to the leaf node 7 because the "default class" is dominant. Thus, if we denote $|T_l|$ the number of leaf nodes in the l^{th} decision tree, the prediction rule for an individual i is given by

$$h_l(x_i) = \sum_{t=1}^{|T_l|} c_t \mathcal{R}_{i,t}, \tag{12}$$

[1]We simplify the description of the algorithm restricting the focus on quantitative predictors. The idea is similar for qualitative predictors.

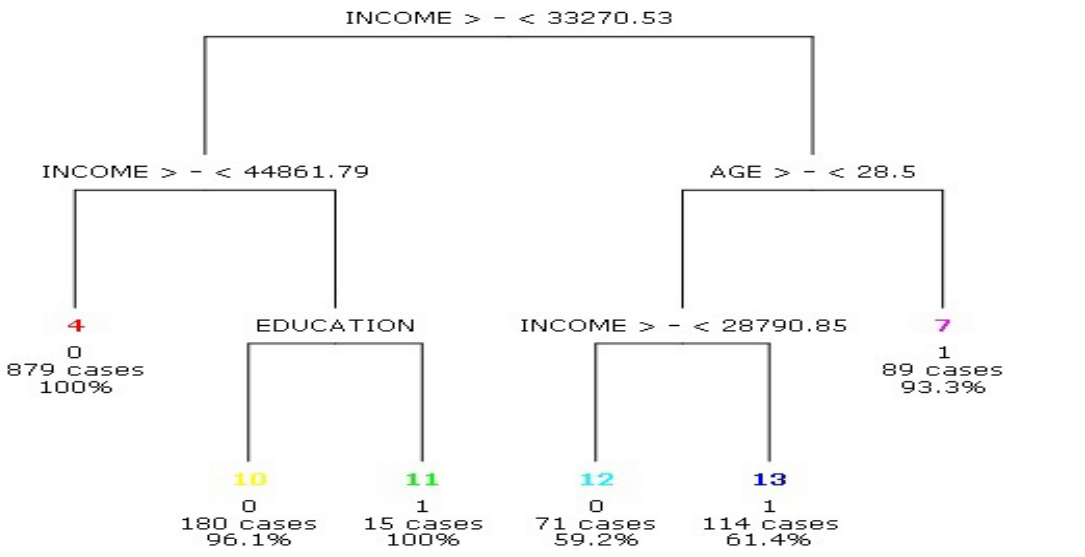

Figure 2: Example of decision tree for credit scoring

where $\mathcal{R}_{i,t} = 1_{(i \in \mathcal{R}_t)}$, with $\mathcal{R}_t$, $t = 1, ..., |T_l|$ the leaf nodes, and c_t the dominant class in the leaf node $\mathcal{R}_t$.

The decision tree method is known to be powerful at detecting univariate and multivariate threshold effects. Nevertheless, its generalization power can be limited due to high variance and instability. Random forest is a bagging procedure that averages many non correlated decision trees to reduce instability. Specifically, assume that L trees are learned, each using a bootstrapped (with replacement) sample of size n drawn from the original sample. To ensure that those trees have a low level of correlation, the first component of $\theta_{m,l}$, i.e. the candidate variable for the split number m when learning the tree l is chosen from a restricted number of randomly selected predictors among the p available ones. Random forest uses the principle of majority vote to form the final prediction $h(x_i)$ based on the L decision tree predictions $h_l(x_i)$, $l = 1, ..., L$. Specifically, $h(x_i)$ corresponds to the mode of the empirical distribution of $h_l(x_i)$.

Random forest is a strong learner that improves upon the predictive performance of the weak learner decision tree. The out-performance springs theoretically from the variance reduction effect of bootstrap aggregation for non correlated predictors (Breiman, 1996). Many empirical papers stressed its performance in the context of credit scoring (e.g. Lessmann, Baesens, Seow, and Thomas, 2015). We illustrate the relative performance of random forest using our Monte Carlo simulations setup. Figure 3 completes the results displayed in Figure 1 by adding the proportion of correct classification for the random forest algorithm. The latter statistic is computed over the same 50 test samples of length 2500 generated. The

optimal number of trees in the forest is tuned using the out-of-bag error.[2] We find that the predictive performance of random forest decreases as the number of predictive variables increases, but it remains high compared to those of logistic regressions. For example, with the largest number of predictors, the proportion of correct classification is equal to 83.05% (resp. 77.91%) for random forest (resp. logistic regression with quadratic and interaction terms).

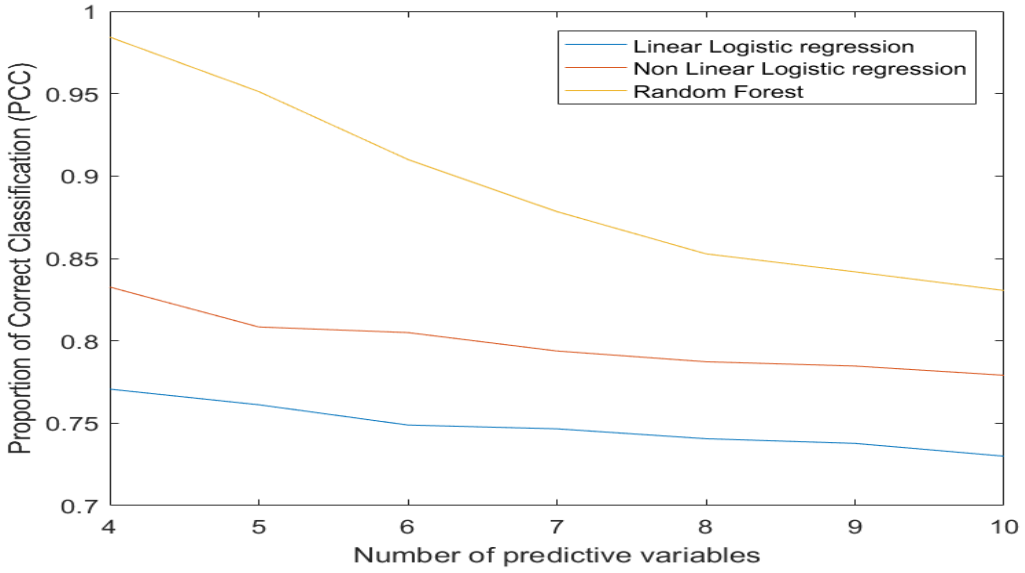

Figure 3: Performance of random forest and logistic regressions under univariate and bivariate threshold effects

Nevertheless, the aggregation rule (majority vote) underlying random forest leads to a prediction rule that lacks interpretation. This opaqueness is harmfull 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 issue here is that of parsimony, namely the search for an optimal 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 in the forest. Second, we can preserve the simplicity of logistic regression while improving its predictive performance with univariate and bivariate threshold effects. We opt here for the second line of research and leave the first one for further research. To be more precise, rules extracted from various short-depth decision trees built with different sets of predictive variables (singletons and couples) are considered as predictors in (regularized) logistic regression. These rules are

[2]See Breiman (2001) for more information on the concept of out-of-bag error, which is an out-of-sample measure of performance.

dummy variables associated to each leaf node from the various decision trees, and allow us to model univariate and bivariate threshold effects. The next section is devoted to the presentation of our methodology.

3 Penalized Logit Tree Regression

Consider that p predictive variables $x_{i,j}$, $j = 1, ..., p$ are available, with $i = 1, ..., n$, n being the number of individuals. As already stressed, in this article we propose to build a logistic regression model based on univariate and bivariate threshold effects. The latter are obtained using decision trees that rely on each predictive variable (singleton) and each couple of predictive variables at a time, with y_i the dependent variable measuring default.

In a first step our methodology runs p decision trees with only one split to obtain threshold effects for each of the p predictive variables. For instance, let the predictive variable number j be the income. The decision tree outputs two binary variables $\mathcal{R}_{i,t}^{(j)}$, $t = 1, 2$, one for each terminal node. The first (second) binary variable $\mathcal{R}_{i,1}^{(j)}$ ($\mathcal{R}_{i,2}^{(j)}$) takes the value one (zero) when the individual i is such that his income is lower than an estimated threshold, and zero (one) otherwise. By convention, we retain the first binary variable for inclusion in our logistic regression. Needless to say that the procedure is similar for a qualitative variable, with $\mathcal{R}_{i,1}^{(j)}$ including some levels of this variable, and $\mathcal{R}_{i,2}^{(j)}$ the remainder. Note that at the end of this step we have p binary variables $\mathcal{R}_{i,1}^{(j)}$, $j = 1, ..., p$, that summarize univariate threshold effects.

The objective of the second step is to build bivariate threshold effects from decision trees based on each couple of predictive variables, i.e. with two splits. For illustration, if two such variables j and k are income and age, respectively, the decision tree generates three binary variables $\mathcal{V}_{i,t}^{(j,k)}$, $t = 1, 2, 3$, each associated to a terminal node. The first binary variable $\mathcal{V}_{i,1}^{(j,k)}$ could take value one when the income is lower than an estimated income threshold, and zero otherwise. The second (third) binary variable $\mathcal{V}_{i,2}^{(j,k)}$ ($\mathcal{V}_{i,3}^{(j,k)}$) could be equal to one when the income is higher than the above threshold and at the same time the age is lower (higher) than an estimated age threshold, and zero otherwise. 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 splitting. If the second variable is uninformative, the tree would rely on the (first) informative one. Overall, from these decision trees with two splits we retain the first two binary variables $\mathcal{V}_{i,1}^{(j,k)}$ and $\mathcal{V}_{i,2}^{(j,k)}$ for inclusion in our logistic regression. If we denote by Q the total number of couples, we count $2Q$ bivariate threshold effects.

Remark that one could extend these two steps by considering triplets and quadruplets of variables, i.e. more than two splits. Such a procedure could be useful in order to include more

complex non linear relationships in the logistic regression. Nevertheless, as our objective is to build a model that is less complex than random forest in terms of interpretability, we use only short-depth decision trees involving one and two splits based on singletons and couples of variables.

Finally, our logistic regression with univariate and bivariate threshold effects has the following form

$$\Pr\left(y_i = 1|\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta\right) = \frac{1}{1+\exp\left\{-\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)\right\}}, \tag{13}$$

with

$$\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = \beta_0 + \sum_{j=1}^{p}\beta_j\mathcal{R}_{i,1}^{(j)} + \sum_{j=1}^{p}\sum_{k=1,k\neq j}^{p}\psi_{j,k}\mathcal{V}_{i,1}^{(j,k)} + \sum_{j=1}^{p}\sum_{k=1,k\neq j}^{p}\gamma_{j,k}\mathcal{V}_{i,2}^{(j,k)}, \tag{14}$$

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

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

where $F(\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)) = \Pr\left(y_i = 1|\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta\right)$. The estimate $\widehat{\Theta}$ of Θ 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 total number of $m = 100$ univariate and bivariate threshold effects in our logistic regression.

To prevent multicollinearity and over-fitting 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 regression* in our case, this method consists in adding a penalty term to the negative value of the log-likelihood function, such that

$$\mathcal{L}_p(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = -\mathcal{L}(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) + \lambda P(\Theta), \tag{15}$$

with $\lambda P(\Theta)$ the additional term that penalizes the estimates during the estimation process. This penalty term depends on the tuning parameter λ that controls the intensity of the regularization. Several penalty terms have been proposed in the related literature (Tibshirani, 1996; Zou and Hastie, 2005; Zou, 2006), but the most popular is the L1-penalty ($P(\Theta) = \sum_{j=1}^{m}|\theta_j|$) from Tibshirani (1996) that corresponds to the Least Absolute Shrinkage and Selection Operator (Lasso). This method has the advantage of performing both

selection and regularization of coefficients while being computationally feasible in high dimensional data. The lasso-parameter estimators solve

$$\widehat{\Theta}_{lasso}(\lambda) \;=\; \arg\min_{\Theta}\left\{-\mathcal{L}\left(\mathcal{R}_{i,1}^{(j)},\mathcal{V}_{i,1}^{(j,k)},\mathcal{V}_{i,2}^{(j,k)};\Theta\right)+\lambda\sum_{j=1}^{m}|\theta_j|\right\}. \tag{16}$$

Remark that if λ=0, then $\widehat{\Theta}_{lasso}(\lambda)$ comes down to the maximum likelihood estimator. Otherwise, if $\lambda \to \infty$, the Lasso estimator approaches the null vector. The optimal value of the tuning parameter λ and hence $\widehat{\Theta}_{lasso}(\lambda)$ is usually obtained by relying on cross-validation exercise or by using some information criteria.

It is worth mentioning that the Lasso estimator $\widehat{\Theta}_{lasso}(\lambda)$ has some drawbacks, the most important being the lack of oracle properties (Fan and Li, 2001). More precisely, the probability to exclude relevant variables and to select irrelevant ones is not zero for this estimator. Therefore, we use the Adaptive lasso estimator of Zou (2006) instead. This method is an extension of the Lasso that solves the above mentioned pitfall. 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 $P(\Theta) = \sum_{j=1}^{m} w_j|\theta_j|$ with $w_j = |\widehat{\theta}_j^{(0)}|^{-v}$, where $\widehat{\theta}_j^{(0)}$, $j = 1, ..., m$, are consistent initial estimators of the parameters, and v is a positive constant. Hence, the Adaptive Lasso estimators are obtained as

$$\widehat{\Theta}_{alasso}(\lambda) \;=\; \arg\min_{\Theta}\left\{-\mathcal{L}\left(\mathcal{R}_{i,1}^{(j)},\mathcal{V}_{i,1}^{(j,k)},\mathcal{V}_{i,2}^{(j,k)};\Theta\right)+\lambda\sum_{j=1}^{m}w_j|\theta_j|\right\}. \tag{17}$$

We set the parameter v to 1, the initial estimator $\widehat{\theta}_j^{(0)}$ to the value obtained from the ridge regression (Hoerl and Kennard, 1970), and the only free tuning parameter λ is found via 5-fold cross-validation. From the viewpoint of estimation, different methods or algorithms have been developed in the literature to estimate (for a given value of λ) regression models with the adaptive lasso penalty: the quadratic programming technique (Shewchuk et al., 1994), the shooting algorithm (Zhang and Lu, 2007), the coordinate-descent algorithm (Friedman, Hastie, and Tibshirani, 2010), and the Fisher scoring algorithm (Park and Hastie, 2007). Most of these algorithms are implemented in softwares like Matlab and R. We rely here on the algorithm based on Fisher scoring. It is a minimization algorithm based on Newton iterations with an updating scheme at iteration t given by

$$\Theta^{(t+1)} = \Theta^{(t)} - (\frac{1}{n}H(\Theta^{(t)}) - \lambda\mu)^{-1}(\frac{1}{n}S(\Theta^{(t)}) - \lambda P^{'}(\Theta^{(t)})), \tag{18}$$

with $S(\Theta^{(t)})$ the score of the likelihood with respect to Θ, $P'(\Theta^{(t)})$ the derivative of the penalty term $P(\Theta)$ with respect to Θ, $H(\Theta^{(t)})$ the hessian of the likelihood with respect to Θ and μ the hessian of the penalty term $P(\Theta)$ with respect to Θ. In practice, the hessian

matrix is replaced by the negative of the Fisher information matrix, $-\mathcal{I}$, such that[3]

$$\Theta^{(t+1)} = \Theta^{(t)} + (\frac{1}{n}\mathcal{I} + \lambda\mu)^{-1}(\frac{1}{n}S(\Theta^{(t)}) - \lambda P'(\Theta^{(t)})). \tag{19}$$

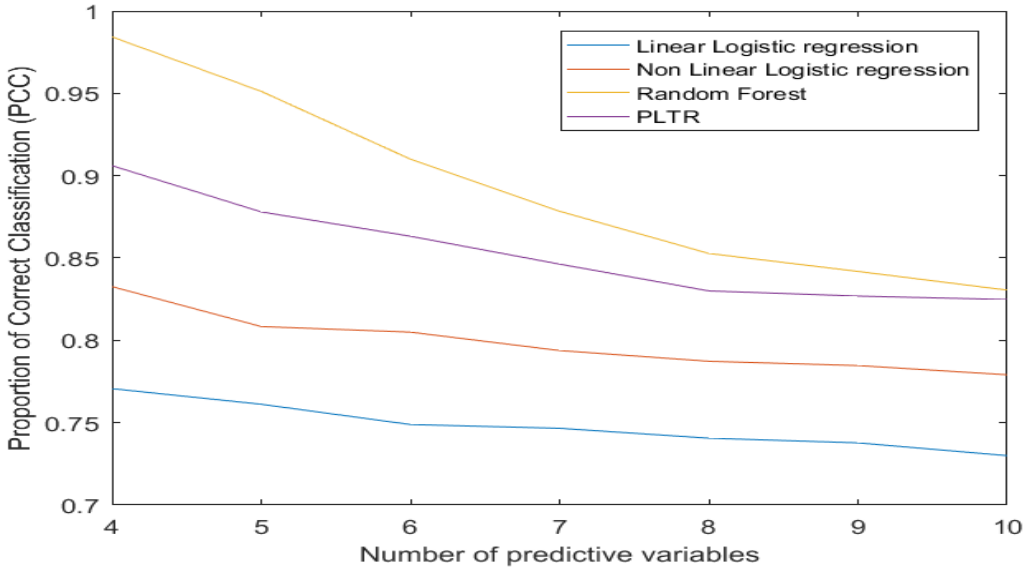

Figure 4: Comparison of performances under univariate and bivariate threshold effects

Figure 4 completes the results displayed in Figure 3 by adding the proportion of correct classification (PCC) for our PLTR method. As in the previous section, PCCs are computed over the same 50 test samples of length 2500 generated. The predictive performance of all methods decreases as the number of predictive variables increases. Our method significantly outperforms the two versions of the logistic regression, i.e., the traditional one that relies only on linear predictors and the one that includes quadratic and interaction terms. The PCCs of the new method are lower than the ones from the random forest algorithm, but asymptotically ($p \to \infty$) both methods perform similarly[4].

Beyond predictive performances it is important to stress that our method is more parsimonious than random forest. First, it relies on univariate and bivariate threshold effects obtained from short-depth decision trees and which are easily interpretable. In contrast, random forest aggregates, via the majority vote, many non-pruned long-depth decision trees and hence lacks parsimony. For instance, with $p = 10$, the average number of trees across the simulations is equal to 86.74, each having an average number of terminal nodes equal to 558.8. This leads to a total of 558.8×86.74 binary decision variables that can be used by random forest for prediction. Across the same simulations, the average number of active

[3]See Park and Hastie (2007) for some discussions about convergence and how to deal with the non-differentiability of the penalty function.

[4]With p the number of initial predictors.

variables in our penalized logistic regression is only equal to 86.76. Second, marginal effects can be easily obtained in our method, because of the linearit of the link function in (14) with respect to the parameters. This greatly simplifies significance testing as well as the implementation of out-of-sample exercises.

One can argue that relative to the two versions of the logistic regressions, our Monte Carlo simulations design favors the new method which is elaborated to handle univariate and bivariate threshold effects. In the next section, we evaluate the predictive performance of our method relative to that of its competitors by using real datasets for credit scoring.

4 Statistical performances with real data sets

4.1 Data description and processing

To gauge the efficiency of the new method, we use two popular data sets. 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 has been provided by a financial institution for the Kaggle competition "Give me some credit" and is often used in credit scoring applications (Baesens, Gestel, Viaene, Stepanova, Suykens, and Vanthienen, 2003). Each of the data sets includes several predictive variables and a binary response variable measuring default. The predictive variables provide information about the customers (age, number of years at the present job, etc.) and the application form (amount of the loan, number of recent credits secured, etc.). The Housing data set includes 12 explanatory variables two out of wich are qualitatives, while the Kaggle data set contains 10 predictive variables that are all quantitatives. A description of the variables is provided in the Appendix.

The Housing data set includes $5,960$ loans, $1,189$ of them having defaulted. Therefore, the prior default rate is 0.2. In the Kaggle data set, there are $150,000$ loans out of wich $10,026$ defaults, leading to a prior default rate of 0.067, wich means that the loan classes are highly imbalanced. It is well known that class imbalances can affect the results of a classification. In fact, some classifiers may pay too much attention to the majority class and neglect the minority group. Such classifiers could hence exhibit good overall performances despite having bad results for the minority group.[5] To solve this problem, one can use resampling methods such as undersampling, oversampling or the SMOTE (Verbeke, Dejaeger, Martens, Hur, and Baesens, 2012). Nonetheless, we choose not to resample the data sets for a simple reason. Given that our objective is to propose a method that is easily interpretable and usable by managers, we decide to work in the same conditions as the companies and

[5]Depending on the degree of imbalances, some classifiers can even misclassify every member of the minority group and still have good global performances.

resampling is not very popular outside academia. Moreover, this approach allows us to see whether our method performs well compared to firms' benchmark (Logistic regression) and machine learning techniques' benchmark (Random Forest).

Lastly, we prepare the datasets for use in our empirical applications. To do so, we replace each missing value by the mean if the predictive variable is numeric, and by the mode otherwise. 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) wich involves randomly dividing the data set 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 classical data partitioning, particularly when data sets are relatively small. We set $N = 10$ for the Housing data set and $N = 3$ for the Kaggle data set for computational reasons.

4.2 Statistical measures of performance

To evaluate the performance of each classifier we consider five 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. 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[6]

$$PGI = \frac{2\int_a^b L(x)dx}{(a+b)(b-a)} - 1. \tag{20}$$

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 the optimal operating point of the ROC curve wich is the best trade-off between the FPR and TPR.

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, Edelman, and Crook, 2002).

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

$$BS = \frac{1}{n}\sum_{i=1}^{n}(\hat{\Pr}(y_i = 1|x_i) - y_i)^2, \tag{21}$$

where $\hat{\Pr}(y_i = 1|x_i)$ 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, he higher these indicators are the better the model is, except for the Brier Score for wich a small value is better.

4.3 Results and analysis

Table 1 presents the average value of each statistic across the 3×2 cross-validation test samples for the Kaggle data set. We compare the performance of the PLTR to those of the traditional logistic regression and random forest. Two different versions of the logistic regression are implemented: the simple linear logistic regression and its non linear version, wich includes as additional variables quadratic and interaction terms. As already stressed,

[6]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, Baesens, Seow, and Thomas (2015).

this last model is the one that is generally used to capture non linear effects in the framework of logistic regression. We also include the MARS (multivariate adaptive regression splines) model of Friedman (1991) in the comparison. Indeed, this method is an extension of linear models that enables one to catch non linear relationships and interaction between predictors through hinge functions (see Friedman, 1991). Although different from our method, it presents some similarities and is considered as one of the most powerful modern statistical learning algorithms.

Table 1: Average values of Statistical performance indicators : Kaggle data set

Method	AUC	PGI	PCC	KS	BS
Linear Logistic Regression	0.6982	0.3961	0.9341	0.3166	0.0576
Non Linear Logistic Regression	0.7648	0.5268	0.9336	0.4104	0.0574
MARS	0.8570	-	0.9367	0.5591	0.2594
Random Forest	0.8386	0.6695	0.9350	0.5364	0.0515
PLTR	0.8519	0.6980	0.9364	0.5577	0.0497

Note: The non linear logistic regression includes linear, quadratic and interaction terms.

The results displayed in Table 1 show that random forest performs better than the two versions of the logistic regression, and this holds for all statistical measures considered. This is expected given that random forest is the benchmark method in terms of performance for credit scoring applications (Lessmann, Baesens, Seow, and Thomas, 2015). In particular, the differences are more pronounced for the AUC, KS, and especially PGI statistics. Most importantly, note that our PLTR method outperforms both versions of the logistic regression irrespective of the performance measure. This is particularly the case of AUC, KS and PGI metrics for wich 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 result is that one should use our method instead of random forest, at least for this data set. The rational of this assertion springs from the parsimony of the PLTR that contrasts with the complexity underlying the prediction rule of random forest. Indeed, the average number of trees in the random forest across the 3×2 cross-validation test samples is equal to 86. These trees have on average $8,773$ terminal

nodes, with a total of $8,773 \times 86$ binary variables for prediction (via the majority vote). By contrast, the average number of univariate and bivariate threshold effects selected by our penalized logistic regression is only equal to 78.1667. More importantly, these univariate and bivariate threshold effects are easily interpretable because they arise from short-depth decision trees.

Note also that the PLTR and MARS have similar performances. Indeed, the AUC, PCC and KS performance statistics do not differ much in this case. Nonetheless, the BS statistic of the MARS algorithm is much higher, indicating that our method beats the MARS from this perspective. This result also implies that the estimated probabilities from MARS are quite far from realistic probabilities. At a closer look we find that indeed all fitted probabilities from MARS are higher than 0.6, situation wich is unlikely to happen in practice.[7] Moreover, from the viewpoint of interpretability, the univariate and bivariate threshold effects underlying our method are easier to disclose compared to the hinge functions which are the building block of MARS.

The results above confirm the importance of using different measures of performance when comparing several credit scoring methods. Depending on the approach considered the conclusions may be different. For example, if the objective is to obtain accurate probabilities of default, the methods are almost equivalent (except for MARS). Nonetheless, the performance of the two versions of the logistic regression is much inferior in terms of discriminatory ability.

Table 2: Average values of Statistical performance indicators : Housing data set

Method	AUC	PGI	PCC	KS	BS
Linear Logistic Regression	0.7910	0.5524	0.8347	0.4426	0.1230
Non Linear Logistic Regression	0.8092	0.5677	0.8561	0.4773	0.1130
MARS	0.8866	-	0.8805	0.6350	0.2549
Random Forest	0.9501	0.8364	0.9090	0.7800	0.0670
PLTR	0.8977	0.7271	0.8828	0.6599	0.0868

Note: The non linear logistic regression includes linear, quadratic and interaction terms.

Table 2 displays the same measures of performance but in the case of the Housing data set. As previously observed, random forest, MARS and our method always outperform the two versions of the logistic regression. The results also suggest that the PLTR seems to compete well with MARS. Indeed, in view of the AUC, PCC and KS performance statistics and especially the BS statistic, our method clearly dominates the MARS algorithm.

In contrast to the results obtained for the Kaggle data set, it now appears that random forest outperforms our method. This result is expected because our method is based

[7] Recall that we compute PGI within $(0, 0.4)$ bounds. Hence we cannot compute this statistics for MARS.

on a compromise between statistical performance and interpretability. Indeed, using the same arguments as above, the average number of active variables (univariate and bivariate threshold effects) from our penalized logistic regression is equal to 102.65, while random forest relies on average on 506.9×81.5 binary variables for prediction.[8] Hence, the PLTR is much more parsimonious. Other results, available from the authors 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 data set. 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 at the cost of less interpretable results.

5 Economic Consequences

In the previous section we found that random forest, MARS and the PLTR introduced in this article have better statistical performances than logistic regression. 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. The best way to evaluate these economic consequences is to calculate the amount of regulatory capital from the estimated default probability series. 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 the terms of the loans, which are not publicly available. Consequently, we compute another measure largely accepted in the literature, i.e., the misclassification costs (Viaene and Dedene, 2004). These costs are 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

$$MC = C_{FP}FPR + C_{FN}FNR, \tag{22}$$

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).

[8] In this dataset we identify on average 81.5 trees in the forest, with an average number of terminal nodes equal to 506.9 for each tree.

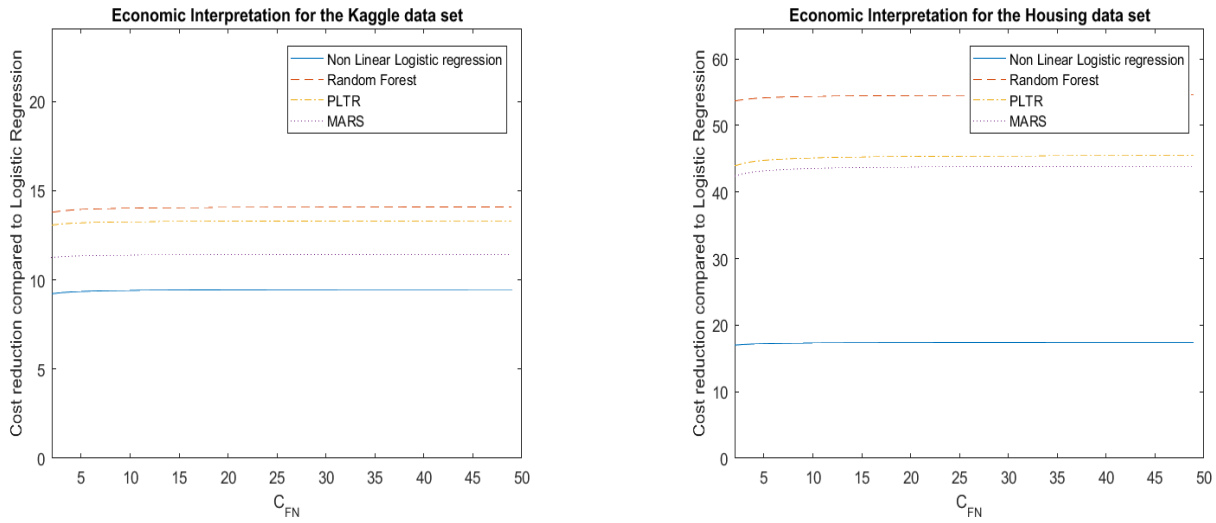

Figure 5: Economic Interpretation for Kaggle and Housing datasets

For example, West (2000) sets C_{FN} to 1 and C_{FP} to 5. The second method evaluates misclassification costs for different values of C_{FN} so as to test as many scenarios as possible (Lessmann, Baesens, Seow, and Thomas, 2015). Even though there is no consensus on how to determine these costs, it is well known and accepted that the costs of granting a credit to a bad customer are higher than the opportunity cost of rejecting a good customer (Thomas, Edelman, and Crook, 2002). We chose to use 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, Flach, and Ferri, 2011) and consider values of C_{FN} between 2 and 50.

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

Figure 5 displays the cost reduction or financial gains for the Kaggle and the Housing data sets, respectively. Non linear logistic regression, random forest, MARS and penalized logit tree regression perform well from the financial perspective for both data sets. These four methods achieve a cost reduction relative to the linear logistic regression of at least 9.0% for the Kaggle data set and 17% for the Housing data set. For all models the cost reduction is highly stable across the different values of C_{FN}. In view of the large number of credits in bank credit portfolios, these gains could represent substantial savings for credit institutions.

Figure 5 also shows that our methodology to predict default risk compares favorably 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 only

slightly smaller than the one of random forest for the Kaggle dataset. Moreover, PLTR is more parsimonious than random forest, hence allowing for simple interpretation of results. Besides, both these models outperform MARS and the non linear logistic regression. For the Housing dataset the results are similar: all cost reductions are positive, stable across the values of C_{FN} and the ranking of the models is the same.

6 Conclusion

Credit scoring has always been intriguing and is still the subject of a lot of works devoted to the development of statistical models for default's prediction. A benchmark model is the logistic regression, which by design leads to conclusions that are easy to disclose and hence interpretable by both credit risk managers and regulators. Since the Big Data revolution and the renewed interest in statistical learning, some papers advocate the use of sophisticated models like random forest, that are shown to outperform the traditional logistic regression. Nevertheless, the prediction rule underlying random forest 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 effects, this article introduces a penalized logistic regression with predictive variables given by easy-to-interpret univariate and bivariate threshold effects. These effects are quantified by dummy variables associated to the leaf nodes of short-depth decision trees built with singletons and couples of the original predictive variables.

We show through Monte Carlo simulations and two empirical applications that the penalized logit tree regression has good predictive power. More precisely, using many statistical metrics for the evaluation of credit scorecards, we observe that it outperforms traditional linear and non linear logistic regression while being competitive compared to random forest.

We also evaluate the economic benefit of using our PLTR method through the so-called misclassification costs. We find that beyond parsimony, our method leads to significant misclassification costs reduction compared to the benchmark logistic regression. Besides, it appears to be competitive with respect to the misclassification costs reduction of the state-of-the-art random forest algorithm. By making an efficient trade-off between performance and interpretability, the PLTR method introduced in this article proves to be a useful tool for credit risk managers.

A Appendix: Descriptions of the Variables

Table 3: Description of the variables for the Housing data set

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	Profession 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 4: Description of the variables for the Kaggle data set

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 but not worse in the last 2 years
NumberOfDependents	Interval	Number of dependents in family excluding themselves (spouse, children, etc...)

References

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

S. Akkoc. 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, 2012.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

J. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–141, 1991.

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

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

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

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

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

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

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

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

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

R. Matignon. *Data Mining Using SAS Enterprise Miner*. 2007.

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

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

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

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

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

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

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

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

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

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

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

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

W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1):211–229, 2012.

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

W. Wang. 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, 2012.

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

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

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

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

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