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

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

November, 2017

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 considered as predictors in a penalized or regularized logistic regression. By modeling such univariate and bivariate threshold effects we achieve significant improvement in model performance. 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.
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 benefits or costs 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, knowing 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 the absence of a performing credit scoring model. The 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 concern for both 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 risk credit. Regulators dislike this scenario, since the banking system becomes less resilient to financial crises and more exposed to systemic events if it is not enough capitalized (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 for the prediction of credit worthiness. These models use specific information about borrowers to estimate the probability that the latter will

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). Results from empirical applications 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 Benelux and UK major financial institutions, and find among others that support vector machine appears to perform best according to the area under the receiver operating characteristic (ROC) curve, albeit simple models (logistic regression, discriminant analysis) are also successful in predicting failure’s 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 2 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 to train the learner. Here 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. They show
that bagging and boosting methods outperform simple classifiers or models, including logistic regression. Paleologo, Elisseeff, and Antonini (2010) propose an ensemble classification technique called subagging which is shown in an empirical application to 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). Using a large number of credit scoring data sets and average robust performance across data sets, they found among others that random forest (Breiman, 2001), i.e. the randomized version of bagged decision trees, outperforms logistic regression. Moreover, from an economic viewpoint, they compare the classification costs, as 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 decisions 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 logistic regression, i.e. increase it to the same order of magnitude as that of random forest, while keeping the simple interpretability of the former method. To do so, we first recognize that ensemble methods like random forest consistently outperform logistic regression, due to the failure of the latter method 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 based on a simple logistic regression with predictors extracted from decision trees. Formally, rules extracted from various short-depth decision trees built with different sets of variables including the original predictive variables (singletons and couples) are considered as predictors in a logistic regression. 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 the new method outperforms traditional logistic regression. Moreover it compares competitively to random forest, while providing an interpretable scoring function. This picture 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 deployment of credit scoring solutions using logistic regression. Credit risk managers usually
consider non-linear effects in logistic regression using ad-hoc or heuristic methods of discretization. The merit of our contribution is to propose a systematic approach to model such non-linear effects using short-depth decision trees. Our 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 method underlying the new methodology for credit scoring, and Sections 4 and 5 are devoted to empirical applications. The last Section concludes the article.

2 Logistic regression under threshold effects

Consider a sample of size \( n \) of independent and identically distributed observations \((x_i, y_i), i = 1, ..., n\), 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 \( \pi \) 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 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 it 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 $\Pr(y_i = 1 \mid x_i) \equiv \Pr(x_i; \beta)$ by

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

(1)

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

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

(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

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

(3)

with $F(.)$ the logistic function. The estimator $\hat{\beta}$ is given by the maximizer of the convex log-likelihood function

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

(4)

with $n$ the sample size. Under some weak assumptions, the estimator $\hat{\beta}$ 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, logistic regression searches for a single linear decision boundary in the predictors space. The core assumption for finding such linear decision boundary 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 such a 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

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

(5)

with estimates obtained by replacing $\beta$ by $\hat{\beta}$. Thus, a predictive variable with positive (negative) 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 $x_i$. One type of non-linear
relationships can arise from the existence of an univariate threshold effect involving a single predictive variable, but also from the combination of these effects (multivariate threshold effects) across variables. A typical example of the former case in the context of credit scoring is the ”income threshold” effect, with i.e. the existence of an endogenous threshold below (above) which default probability is more (less) prominent. The income threshold effect can obviously interfere 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 $\eta(x; \beta)$.

We advocate that such a practice should not be successful when threshold effects are at stake. Below, we run Monte Carlo simulation experiments to give more insight into this point.

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

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

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, with values randomly selected from the support of each predictive variable generated, 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

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

with $\pi$ 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. For each experiment, we divide the simulated sample of size $n = 5000$ into two sub-samples of equal size. The first (second) sub-sample is considered as the learning (test) sample. Based on the learning sample we estimate two different models. The first is the classical logistic regression with linear effects for which the index has the following expression

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

with $\beta = (\beta_0, \beta_1, ..., \beta_p)'$ the unknown parameters. The second model is based on a non
linear index function that incorporates quadratic and interaction terms, i.e.,

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

(9)

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 last 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: Performances of logistic regressions under univariate and bivariate threshold effects](image)

Two patterns emerge from Figure 1. First, for both models, we observe that the proportion of correct classification decreases with the number of predictors. This suggests that in presence of univariate and bivariate threshold effects (see equation 6) involving many variables, logistic regression with linear index function, eventually augmented with quadratic and interaction terms, fails to discriminate between good and bad loans. Indeed, with $p = 10$ 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 square 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 $D_{m,l}$ 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 $D_{m,l}$ into two subsets $D_{m,l,1} (\theta_{m,l})$ and $D_{m,l,2} (\theta_{m,l})$, with

$$D_{m,l,1} (\theta_{m,l}) = \{ x_i, y_i \mid x_{i,j} \leq t_{m,l,j} \}, \quad (10)$$

$$D_{m,l,2} (\theta_{m,l}) = \{ x_i, y_i \mid x_{i,j} > t_{m,l,j} \}. \quad (11)$$

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

$$\hat{\theta}_{m,l} = (\hat{j}_{m,l}, \hat{t}_{m,l,j}) = \arg \max_{\theta_{m,l}} \{ H(D_{m,l}) - H(D_{m,l,1} (\theta_{m,l}), D_{m,l,2} (\theta_{m,l})) \}, \quad (12)$$

where $H(D_{m,l})$ is a measure of diversity in the sample $D_{m,l}$, and $H(D_{m,l,1} (\theta_{m,l}), D_{m,l,2} (\theta_{m,l}))$ is the same measure averaged across the two sub-samples $D_{m,l,1}$ and $D_{m,l,2}$. Diversity is usually approximated by the so-called Gini criterion. Remark that in (12) $\hat{\theta}_{m,l}$ is 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), $m = 1$, $\hat{\theta}_{m,l}$ is defined by $\hat{j}_{m,l} = \text{"income"}$ and $\hat{t}_{m,l,j} = 33270.53$. 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, along with the number of individuals. 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 $h_l(x_i; \Theta_l) = h_l (x_i)$ 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

\footnote{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

leaf node 7, because the "default class" is dominant. Thus, if we denote $|T_l|$ the number of
leaf nodes for the decision tree number $l$, the prediction rule for an individual $i$ is given by

$$h_l(x_i) = \sum_{t=1}^{\left|T_l\right|} c_t R_{i,t},$$

where $R_{i,t} = 1_{(i \in R_t)}$, with $R_t$, $t = 1, ..., \left|T_l\right|$ the leaf nodes, and $c_l$ the dominant class in the
leaf node $t$.

The decision tree method is known to be powerful at detecting univariate and multi-
variate 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 correlations, the first component of
$\theta_{m,t}$ 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 predictors.
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 re-
duction 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.\(^2\) 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).

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: Performances 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, the simplicity of logistic regression can be kept while improving its predictive performance with univariate and bivariate threshold effects. We opt here for the second line of research and

\(^2\)See Breiman (2001) for more information on the concept of out-of-bag error, which is an out-of-sample measure of performance.
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 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 \( x_{i,j}, j = 1,\ldots,p \) are available, with \( i = 1,\ldots,n \), \( n \) 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 and for each of the \( p \) predictive variables, our methodology runs \( p \) decision trees with only one split to obtain threshold effects. For instance, let the predictive variable number \( j \) be the income. The decision tree ends up with two binary variables \( R_{i,t}^{(j)}, t = 1,2, \) each related to a terminal node. The first (second) binary variable \( R_{i,1}^{(j)} (R_{i,2}^{(j)}) \) takes the value one (zero) when the example \( i \) 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 \( R_{i,1}^{(j)} \) including some levels of this variable, and \( R_{i,2}^{(j)} \) the remainder. Note that at the end of this step, we have \( p \) binary variables \( R_{i,1}^{(j)}, j = 1,\ldots,p \) 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 \( V_{i,t}^{(j,k)}, t = 1,2,3, \) each associated to a terminal node. The first binary variable \( V_{i,1}^{(j,k)} \) could take value one when the income is lower than an estimated threshold, and zero otherwise. The second (third) binary variable \( V_{i,2}^{(j,k)} (V_{i,3}^{(j,k)}) \) 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 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. We retain the first two binary variables \( V_{i,1}^{(j,k)} \) and \( V_{i,2}^{(j,k)} \) for inclusion in our logistic regression. If we denote \( Q \) the total number of couples, this leads
to 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 should be useful 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 only consider singletons and couples, and short-depth decision trees involving one and two splits. Our logistic regression with univariate and bivariate threshold effects has the following specification

$$ \Pr \left( y_i = 1 \mid \mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta \right) = \frac{1}{1 + \exp \left\{ -\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) \right\} }, \quad (14) $$

with $\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)$ the link function equal to

$$ \eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = \beta_0 + \sum_{j=1}^{P} \beta_j \mathcal{R}_{i,1}^{(j)} + \sum_{j=1}^{P} \sum_{k=1, k \neq j}^{P} \psi_{j,k} \mathcal{V}_{i,1}^{(j,k)} + \sum_{j=1}^{P} \sum_{k=1, k \neq j}^{P} \gamma_{j,k} \mathcal{V}_{i,2}^{(j,k)}, \quad (15) $$

where $\Theta = (\theta_1, \ldots, \theta_m)' = (\beta_0, \beta_1, \ldots, \beta_p, \psi_{1,2}, ..., \psi_{p-1,p}, \gamma_{1,2}, ..., \gamma_{p-1,p})'$ is the set of $m$ parameters to be estimated. The corresponding log-likelihood is

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

with $n$ the sample size, and $F(\eta(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta)) = \Pr \left( y_i = 1 \mid \mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta \right)$. The estimate $\hat{\Theta}$ of $\Theta$ is obtained by maximizing the above log-likelihood with respect to the unknown parameters $\Theta$. Remark that the length of $\Theta$ depends on $p$, the number of predictive variables, and can be relatively high. For instance with $p=10$, there are 45 couples of variables. 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 is based on the addition of a penalty term to the negative value of the log-likelihood function, such that

$$ \mathcal{L}_p(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) = -\mathcal{L}(\mathcal{R}_{i,1}^{(j)}, \mathcal{V}_{i,1}^{(j,k)}, \mathcal{V}_{i,2}^{(j,k)}; \Theta) + \lambda P(\Theta), \quad (16) $$

with $\lambda P(\Theta)$ the additional term that penalizes the estimates during the estimation process. This penalty term depends on the tuning parameter $\lambda$ 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

\[
\hat{\Theta}_{\text{lasso}}(\lambda) = \arg \min_{\Theta} \left\{ -L \left( R_{i,1}, V_{i,1}, V_{i,2}; \Theta \right) + \lambda \sum_{j=1}^{m} |\theta_j| \right\}.
\]

(17)

Remark that if \( \lambda = 0 \), then \( \hat{\Theta}_{\text{lasso}}(\lambda) \) 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 \( \lambda \) and hence \( \hat{\Theta}_{\text{lasso}}(\lambda) \) is usually obtained by relying on cross-validation exercise or by using some information criteria.

It is worth mentioning that the Lasso estimator \( \hat{\Theta}_{\text{lasso}}(\lambda) \) 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 (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 \( w_j = |\tilde{\theta}_j^{(0)}|^{-v} \), where \( \tilde{\theta}_j^{(0)} \), \( j = 1, \ldots, m \) are initial consistent estimators of the parameters, and \( v \) a positive constant. Hence, the Adaptive Lasso estimators are obtained as

\[
\hat{\Theta}_{\text{alasso}}(\lambda) = \arg \min_{\Theta} \left\{ -L \left( R_{i,1}, V_{i,1}, V_{i,2}; \Theta \right) + \lambda \sum_{j=1}^{m} w_j |\theta_j| \right\}.
\]

(18)

We set the parameter \( v \) to 1, the initial estimator \( \tilde{\theta}_j^{(0)} \) to the value obtained from the ridge regression (Hoerl and Kennard, 1970), and the only free tuning parameter \( \lambda \) 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 \( \lambda \)) 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 update scheme at iteration \( t \) given by

\[
\Theta^{(t+1)} = \Theta^{(t)} - \left( \frac{1}{n} H(\Theta^{(t)}) - \lambda \mu \right)^{-1} \left( \frac{1}{n} S(\Theta^{(t)}) - \lambda P'(\Theta^{(t)}) \right),
\]

(19)

with \( S(\Theta^{(t)}) \) the score of the likelihood with respect to \( \Theta \), \( P'(\Theta^{(t)}) \) the derivative of the penalty term \( P(\Theta) \) with respect to \( \Theta \), \( H(\Theta^{(t)}) \) the hessian of the likelihood with respect to
$\Theta$ and $\mu$ the hessian of the penalty term $P(\Theta)$ with respect to $\Theta$. In practice, the hessian matrix is replaced by the negative of the Fisher information matrix, $-I$ such that

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

(20)

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 method. As above, 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.

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, with a total of $558.8 \times 86.74$ binary decision variables used by random forest for

---

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

4With $p$ the number of initial predictors.
prediction. Across the same simulations, the average number of active 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 linearity (with respect to the parameters) of the link function in (15). 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 using real datasets for credit scoring.

4 Statistical performances with real data sets

4.1 Data description and processing

To confirm 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; Zhou and Wang, 2012). 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 credit secured, etc.). The Housing data set includes 12 explanatory variables and two of them 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 including 10,026 defaults, leading to a prior default rate equal to 0.067: the loan classes are thus 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. Due to class imbalances, such classifiers could exhibit good overall performances despite having bad results for the minority group. 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

\footnote{Depending on the degree of imbalances, some classifiers can even misclassify every member of the minority group and still have good global performances.}
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 firms and 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 get the datasets ready to be 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. One important step in our evaluation scheme is data partitioning. For this, we used the so-called $N \times 2$-fold cross-validation by following Dietterich (1998). We randomly divide 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 using 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. Using many instead of one statistic allows for 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, considering all possible thresholds is not realistic. 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\(^6\)

\[
PGI = \frac{2 \int_a^b L(x)dx}{(a + b)(b - a)} - 1. \tag{21}
\]

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 \( \pi \) above (below) which a loan is classified as bad (good). In practice, the threshold \( \pi \) 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 which 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

\[
BS = \frac{1}{n} \sum_{i=1}^{n} (\hat{P}(y_i = 1|x_i) - y_i)^2, \tag{22}
\]

where \( \hat{P}(y_i = 1|x_i) \) is the estimated probability of default and \( y_i \) the target binary default variable, and thus corresponds to the mean-squared error.

The higher are these indicators the better is the model, except for the Brier Score for which a small value is better.

### 4.3 Results and analysis

Table 1 presents, for each statistic, the average value across the \( 3 \times 2 \) cross-validation test samples for the Kaggle data set. We compare the performance of the new method to those of the traditional logistic regression and random forest. Two different versions of the logistic regression are considered: the simple linear logistic regression, and its non linear version, \( a = 0 \) and \( b = 1 \) is equivalent to Gini Index. In empirical applications, we set \( a = 0 \), \( b = 0.4 \), as in Lessmann, Baesens, Seow, and Thomas (2015), and evaluate the (Partial) Gini Index within these bounds.
wich includes as additional variables quadratic and interaction terms. As already stressed, this last model is the one that is generally used to capture non linear effects in the framework of logistic regression. We also include in the comparison the MARS (multivariate adaptive regression splines) model of Friedman (1991). 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

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>PGI</th>
<th>PCC</th>
<th>KS</th>
<th>BS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Logistic Regression</td>
<td>0.6982</td>
<td>0.3961</td>
<td>0.9341</td>
<td>0.3166</td>
<td>0.0576</td>
</tr>
<tr>
<td>Non Linear Logistic Regression</td>
<td>0.7648</td>
<td>0.5268</td>
<td>0.9336</td>
<td>0.4104</td>
<td>0.0574</td>
</tr>
<tr>
<td>MARS</td>
<td>0.8570</td>
<td></td>
<td>0.9367</td>
<td>0.5591</td>
<td>0.2594</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.8386</td>
<td>0.6695</td>
<td>0.9350</td>
<td>0.5364</td>
<td>0.0515</td>
</tr>
<tr>
<td>PLTR</td>
<td>0.8519</td>
<td>0.6980</td>
<td>0.9364</td>
<td>0.5577</td>
<td>0.0497</td>
</tr>
</tbody>
</table>

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 new method introduced in this article outperforms both versions of the logistic regression, irrespective of the performance measure, the dominance is stronger for the AUC, KS and PGI metrics. 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 new method that contrasts with the complexity underlying the prediction rule of random forest. Indeed, the average number of trees in the random forest
across the $3 \times 2$ 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). In comparison, 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 new method 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, and hence our method appears to be more powerful in view of this statistic. This result implies that the estimated probabilities from MARS are quite far from realistic probabilities. This is the case, since all fitted probabilities from MARS are higher than 0.6 and this situation is unlikely to happen in practice.\footnote{Recall that we compute PGI with thresholds for probabilities between 0 and 0.4. Hence we cannot compute this statistics for MARS.} Moreover, from the viewpoint of interpretability, 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 in 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

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

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

Table 2 displays the same statistics 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 new method 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 as already stressed, our method is based 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 $506.9 \times 81.5$ binary variables for prediction.\footnote{On average in this dataset we identify 81.5 trees in the forest, with an average number of terminal nodes equal to 506.9 for each tree.} Hence, our method 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 method 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

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

with FPR the False Positive Rate and FNR the False Negative Rate. In order to determine
$C_{FN}$ and $C_{FP}$, there is no consensus about the two alternative methods available in the literature. The first method fixes these costs based on previous studies (Akkoc, 2012). For example, West (2000) sets $C_{FN}$ to 1 and $C_{FP}$ to 5. The second method evaluates misclassification costs for different values of $C_{FN}$, in order 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 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. For both data sets, non linear logistic regression, random forest, MARS and penalized logit tree regression perform well from the financial perspective. 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.

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 our new model is only slightly smaller than the one of random forest for the Kaggle dataset. Moreover, it 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 of parsimony and can be less relevant in credit scoring applications where decision makers need simple and interpretable rules of prediction.

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 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 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad</td>
<td>Binary</td>
<td>Whether the consumer had a default on the loan (1) or not (0)</td>
</tr>
<tr>
<td>Clage</td>
<td>Interval</td>
<td>Age in months of the oldest trade</td>
</tr>
<tr>
<td>Clno</td>
<td>Interval</td>
<td>Number of trades</td>
</tr>
<tr>
<td>Debtinc</td>
<td>Interval</td>
<td>Ratio of debt to income</td>
</tr>
<tr>
<td>Delinq</td>
<td>Interval</td>
<td>Number of neglectful trades</td>
</tr>
<tr>
<td>Derog</td>
<td>Interval</td>
<td>Number of major derogatory reports</td>
</tr>
<tr>
<td>Job</td>
<td>Nominal</td>
<td>Profession categories</td>
</tr>
<tr>
<td>Loan</td>
<td>Interval</td>
<td>Amount of the loan</td>
</tr>
<tr>
<td>Mortdue</td>
<td>Interval</td>
<td>Amount due on the mortgage</td>
</tr>
<tr>
<td>Ninq</td>
<td>Interval</td>
<td>Number of recent credit inquired</td>
</tr>
<tr>
<td>Reason</td>
<td>Binary</td>
<td>Whether it is for debt consolidation (DebtCon) or home improvement (HomeImp)</td>
</tr>
<tr>
<td>Value</td>
<td>Interval</td>
<td>Current property value</td>
</tr>
<tr>
<td>Yoj</td>
<td>Interval</td>
<td>Number of years at the present job</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SeriousDlqin2yrs</td>
<td>Binary</td>
<td>Person experienced 90 days past due delinquency or worse (Yes/No)</td>
</tr>
<tr>
<td>RevolvingUtilizationOfUnsecuredLines</td>
<td>Percentage</td>
<td>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</td>
</tr>
<tr>
<td>Age</td>
<td>Interval</td>
<td>Age in years of the borrower</td>
</tr>
<tr>
<td>NumberOfTime30-59DaysPastDueNotWorse</td>
<td>Interval</td>
<td>Number of time a borrower has been 30-59 days past due but not worse in the last 2 years</td>
</tr>
<tr>
<td>DebtRatio</td>
<td>Percentage</td>
<td>Monthly debt payments, alimony and living costs over the monthly gross income</td>
</tr>
<tr>
<td>MonthlyIncome</td>
<td>Interval</td>
<td>MonthlyIncome</td>
</tr>
<tr>
<td>NumberOfOpenCreditLinesAndLoans</td>
<td>Interval</td>
<td>Number of open loans (like car loan or mortgage) and credit lines (credit cards)</td>
</tr>
<tr>
<td>NumberOfTimes90DaysLate</td>
<td>Interval</td>
<td>Number of times a borrower has been 90 days or more past due</td>
</tr>
<tr>
<td>NumberRealEstateLoansOrLines</td>
<td>Interval</td>
<td>Number of mortgage and real estate loans including home equity lines of credit</td>
</tr>
<tr>
<td>NumberOfTimes60-89DaysPastDueNotWorse</td>
<td>Interval</td>
<td>Number of times a borrower has been 60-89 days but not worse in the last 2 years</td>
</tr>
<tr>
<td>NumberOfDependents</td>
<td>Interval</td>
<td>Number of dependents in family excluding themselves (spouse, children, etc...)</td>
</tr>
</tbody>
</table>
References


