Developing Transparent Credit Risk Scorecards More Effectively: An Explainable Artificial Intelligence Approach

Gerald Fahner
Analytic Science
FICO
San Jose, USA
geraldfahner@fico.com

***Abstract*—Complex Machine Learning (ML) models can be effective at analyzing large amounts of data and driving business value. However, these models can be nonintuitive, their parameters meaningless, their potential biases difficult to detect and even harder to mitigate, and their predictions and decisions difficult to explain. Lenders, regulators, and customers need explainable models for automating credit decisions. Lack of algorithmic transparency is a broad concern beyond lending, which has led to much interest in "explainable artificial intelligence" [1]. This paper discusses a model family which warrants explainability and transparency by design: the Transparent Generalized Additive Model Tree (TGAMT). Many credit risk models used in the US and internationally belong to this family. Today, these credit scores are developed painstakingly by teams of data scientists and credit risk experts in a tedious interplay of "art and science" in order to simultaneously achieve high predictive performance and intuitive explanations of how the scores are arrived at. The main contribution of this paper is to automate the learning of TGAMT models. We also report benchmark results indicating that TGAMT's achieve strong predictive performance similar to complex ML models while being more explanation-friendly.**

Keywords - explainable artificial intelligence; algorithmic transparency; machine learning; gradient boosting; neural nets; credit risk scoring; scorecard; segmentation; constraining models.

I. INTRODUCTION

Credit scoring has been an early, highly successful and pervasive data mining application. For a comprehensive survey of credit scoring see [2]. Business users in the Financial Services industry frequently rely on the scorecard format to compute scores and to create robust, interpretable and easily deployable scoring solutions for a wide range of applications including marketing targeting, origination, behavior scoring, fraud detection and collections scoring. Scorecards are deployed through automated, rule-based systems to effect impactful, high volume decisions on consumers, such as what product to offer, accept/reject, pricing, limit setting, card authorization and collection treatment, thereby impacting a large part of the economy. Because of the high responsibility shouldered by these systems, model developers and users familiar with the domain seek a high level of transparency and confidence into reasonableness and robustness of the deployed models.

Because no database is perfect, and because future operational conditions tend to differ from past conditions under which data were collected, it has been recognized that incorporation of domain expertise into the data mining process is often essential [3]. It is indeed crucial for scorecard development to strike an appropriate balance between the desire to "let the data talk" and the necessity to engineer the models for deployment. Scorecard technology supports inclusion of domain knowledge into the models, by allowing users to impose constraints, such as monotonicity, on the fitted functional relations.

Modelers who value interpretability nevertheless desire a high degree of flexibility in their scoring algorithms to capture complex behavior patterns and to enable discovery of new, unexpected relationships. This is important in a highly competitive environment characterized by high volumes of automated, high stakes decisions. Being able to capture fainter and more complex predictive patterns that may otherwise escape simplistic models, can make a substantial difference to the bottom line of a business. Segmented scorecards are one response of the scoring industry to these needs. Unlike a single scorecard, which is additive in the predictors, these models can capture interactions between the variables used to define the segments, and the predictors used in the segment-level scorecards. For example, the FICO® Score is constructed as a system of more than a dozen segmented scorecards.

Designing a segmented scorecard system has traditionally been a labor-intensive and rather ad-hoc process during which several segmentation schemes are hypothesized from domain experience and guided by exploratory data analysis. Candidate segmentations are tested and refined to the extent possible given development resources. This process could benefit greatly from more objective and productive approaches.

Independent from these developments in the credit industry and with a different focus, ML ensemble methods, such as stochastic gradient boosting [4] and random forests [5], have been devised in academia. These procedures can automatically learn highly complex relations from data, and despite their flexibility, generalize well to new data if drawn from the same population. These procedures are attractive for ambitious scorecard developers who desire to "leave no stone unturned", because they are automated and scalable, make minimal functional assumptions on consumer

behavior, and can generate insights into the learned relationships through various diagnostics aiding variable selection and interaction detection.

But these procedures are not designed to support inclusion of subtle domain knowledge. The resulting models, and how the scores are computed from the inputs for each particular case, can defy simple explanation. While a single shallow classification and regression tree is a transparent model structure which can be understood by direct inspection, this no longer holds true for modern ensemble learners that often yield more accurate predictions by combining hundreds or thousands of trees. Such opacity can render tree ensembles unfit for deployment. In order to engineer successful solutions, businesses need to look beyond off-the-shelf algorithms to customizable scoring procedures. This raises the methodological question of how to design a productive analytic process pipeline that takes full advantage of modern ensemble learning and associated diagnostic procedures, while supporting inclusion of domain expertise into the modeling process.

The reminder of this paper is organized as follows: Section II reviews scorecard technology and discusses examples of imposing domain knowledge into scorecards. Section III describes the TGAMT method to grow segmented scorecard trees guided by a stochastic gradient boosting model, and Section IV reports experiments comparing the US FICO® Score against ML models and TGAMT.

II. SCORECARD TECHNOLOGY

FICO uses its own proprietary scorecard development platform for supervised learning applications including ranking, classification and regression. The platform is designed to facilitate variable transformations (binning), model selection, fitting, incorporation of domain knowledge through functional constraints, validation, reporting and deployment. In the following, we will briefly discuss the main building blocks from a conceptual perspective. Reference [6] provides more details.

The predictive variables in a scorecard are called characteristics. A characteristic is composed of a set of mutually exclusive and exhaustive bins or attributes that comprise the possible values of the underlying predictor. Characteristics can represent continuous predictors after binning them into intervals, discrete or categorical predictors whereby subsets of values can be grouped together into bins, or hybrid predictors, which have interval bins for their continuous value spectrum and categorical bins for their discrete values. Missing values, even different types of missing or special values, are incorporated naturally as additional bins into characteristics. For example, a missing value could be an indicator of risk if a consumer declined to answer a question, in which case it should not be ignored. Or there might be a temporary issue in the historic data which may not replicate itself in the future (an example of nonstationary distributions scorecard development sometimes has to deal with) in which case missingness should be treated differently, for example by imputation or by assigning a neutral score contribution to such a bin, which is part of "score engineering". Before raw variables can be used as scorecard predictors, they need to be binned. Binning of continuous variables allows a scorecard to model nonlinear relationships between inputs and score through flexible stair functions defined over the bins. Categorical variables with thinly populated categories may be binned into coarser categories, and missing and special values may receive their own bins. Various methods exist for binning and improvements to binning have been proposed [7]. Domain expertise frequently enters the binning process.

While methodologies vary between in-house teams, consultants, and software vendors, often the score is computed as a weighted sum over dummy indicator variables associated with the characteristic bins, plus an intercept term.

$$
\begin{aligned}
Score &= S_0 + \sum_{j=1}^{p} H_j(c_j) \\
S_0 &= Intercept \\
H_j(c_j) &= \sum_{i=1}^{q_j} S_{ij} x_{ij}(c_j) = Characteristic\ score\ of\ characteristic\ j \qquad (1) \\
S_{1j}, S_{2j}, \ldots, S_{q_j j} &= Score\ weights\ associated\ with\ the\ bins\ of\ characteristic\ j \\
x_{1j}, x_{2j}, \ldots, x_{q_j j} &= Dummy\ indicator\ variables\ for\ the\ bins\ of\ characteristic\ j
\end{aligned}
$$

Each of the p characteristic scores is a stair function defined over the q bins of the characteristic. The stair heights are given by score weights associated with the bins. The model structure is similar to dummy variable regression [8]. However, dummy variable regression has no notion of "characteristics", and variable selection happens on the level of the dummies, which tends to put "holes" into binned variables. In contrast, scorecard development technologies can select on the characteristic level. This can makes the models easier to interpret. In addition, some scorecard development platforms allow to constrain characteristic scores to desired shapes, such as monotonicity, which can be applied globally, or involve ranges or subsets of bins.

A powerful feature of scorecards is their ability to model nonlinear effects through nonparametric stair functions. At the same time, scorecards, and how the scores for each case are calculated, remain easy to explain. A scorecard can be represented in tabular form made up of the characteristics, their bins (or attributes), and their associated score weights, as illustrated by Fig. 1. This scheme shows one variable from each of the five key categories that compose the US FICO® Score and is for illustrative use only. "Points" refer to a scaled version of the score weights in eq. (1). For a given applicant, points are added according to his/her attributes across all characteristics, to compute the total score. The assignment of points to attributes is guaranteed to follow explainable patterns which can be reinforced by the model developer via constraints, for example constraints may be necessary to smooth noise or to mitigate data biases. A typical scorecard may contain between 12 and 20 characteristics. How the variables combine with each other to impact the score is very clear, and explainable. The simplicity of the scorecard format was historically important and still is today, to gain business users' trust in these models, and to facilitate the inclusion of domain expertise into the modeling process.

Category	Characteristics	Attributes	Points
Payment History	Number of months since the most recent serious delinquency	No serious delinquency	75
		0 – 5	10
		6 – 11	15
		12 – 23	25
		24+	55
Outstanding Debt	Overall utilization on revolving trades	No revolving trades	30
		Under 6%	65
		7 – 19%	50
		20 – 49%	45
		50 – 89%	25
		90% or more	15
Credit History Length	Number of months in file	Below 12	12
		12 – 23	35
		24 – 47	60
		48 or more	75
Pursuit of New Credit	Number of inquiries in the last 6 months	0	70
		1	60
		2	45
		3	25
		4+	20
Credit Mix	Number of bankcard trade lines	0	15
		1	25
		2	55
		3	60
		4+	50

Figure 1. Simplified version of a scorecard.

Estimation of score weights in eq. (1) is possible using many approaches. FICO's technology accommodates various objective functions including penalized maximum likelihood for regression, ranking and classification, and penalized maximum divergence (related to discriminant analysis) for classification and ranking. Regression applications include normal, logistic and Poisson regression, whereby the score models the linear predictor as in Generalized Linear Models. In the logistic regression case with a dichotomous dependent variable, the score models log(Odds). The score weights are the decision variables of the ensuing optimization problems. Nonlinear programming techniques can be used to optimize the score weights subject to linear equality and linear inequality constraints. These constraints provide mechanisms to incorporate subtle domain knowledge into the models. For example, inequality constraints between neighboring bins of an ordinal variable can be used to restrict a fitted relationship between the variable and the score to be monotonic. Consider the characteristic score for 'Time On Books (TOB)', assuming for simplicity only 3 TOB bins:

$$H_{TOB} = S_{1,TOB}1\{0 \le TOB < 60\} + S_{2,TOB}1\{60 \le TOB < 120\} + S_{3,TOB}1\{TOB \ge 120\}$$

To enforce a monotonic increasing relation between 'TOB' and the score, specify inequality constraints as follows:

$$S_{1,TOB} \le S_{2,TOB} \le S_{3,TOB}$$

With this, the optimization will solve for the optimal score weights subject to the desired monotonic shape. Monotonicity can be useful for various reasons:

(a) Dependencies between predictors and score can be restricted to intuitive shapes. For example, everything else being equal one might expect equal or higher credit quality associated with longer TOB, or one might expect lower credit quality associated with higher frequency of late payments.

(b) Constraints reduce the hypothesis space and effective degrees of freedom of the model family, hence if constraints are applied sensibly, a constrained model can be less prone to over-fitting [9].

(c) Constraints may be necessary to ensure legal compliance. For example, the US Equal Opportunity Act implies that elderly applicants must not be assigned lower score weights than the younger. An empirical derived, flexible model may not be compliant, in which case a monotonicity constraint can rectify the desired relation.

(d) Imposing constraints may be necessary when adverse decisions, such as credit rejections, need to be justified to customers.

Monotonicity constraints can be imposed over the entire range of an ordinal variable, or they can be imposed over specific intervals, for example to allow for unimodal functional forms, as illustrated by 'Number of bankcard trade lines' in Fig. 1.

The scorecard format comprises a flexible family of functions capable of modeling nonlinear effects of predictors on the score by means of constrainable stair functions. However, eq. (1) specifies an additive function of the predictors which cannot capture interactions between predictors. If the true relationship is characterized by substantial interactions then this model is biased and might under-fit the data. To overcome this limitation, several approaches exist, including:

(a) Creation of derived predictors, such as ratios between the original predictor variables. This is part of data pre-processing or featurization and outside the scorecard model.

(b) Inclusion of cross-characteristics into the models, which generate products of bin indicator variables.

(c) Segmented scorecards, whereby different scorecards apply to different segments of a population.

In the following, we will focus on segmented scorecards, which are most widely used in the financial services, likely because the models are easy to inspect, to interpret and to engineer.

Reference [10] includes an overview of reasons and practices for undertaking model segmentation. The authors report mixed results with a research algorithm for finding good segmentations for credit risk score development data sets. The findings cast doubt over whether segmentations are as useful as they are widely thought to be, when looking at the benefits from a purely predictive standpoint (in terms of improving model fit). There are also other reasons for creating segmented models, such as availability of different variables for different customer types, or a need for subpopulation homogeneity in the segments for managerial reasons. On the other hand, the findings in [11] are more upbeat about the predictive benefits of 2-way segmentations

for improving the discriminatory power of the resulting score. According to the omnipresent bias-variance tradeoff, and since a single scorecard is already a flexible model, it stands to reason that segmentation may indeed sometimes do more harm than good, because the larger hypothesis space for the segmented model family makes it easy to over-fit the data, eventually outweighing the benefits of reducing the single-scorecard structural bias. For this reason it is important to carefully navigate the bias-variance tradeoff during segmented scorecard development.

Scorecard segmentations can be represented as binary tree structures. The root node represents the entire population. Starting from the root node, the population is split into child nodes, defined by a first split variable, such as 'TOB' in the example of a simple segmentation given by eq. (2). One of the two child nodes there is further split according to 'Number of accounts' which results in an asymmetric binary tree. Splitting eventually stops and leaf nodes are created. The leaf nodes represent nonoverlapping population segment which together make up the full population. Each leaf node contains a dedicated scorecard denoted by $Scorecard_1$, $Scorecard_2$, and $Scorecard_3$ for the 3 leaf nodes in this example:

$$Score = \begin{cases} Scorecard_1 \; if \; TOB < 24 \\ Scorecard_2 \; if \; TOB \geq 24 \; and \; Number \; of \; accounts < 2 \\ Scorecard_3 \; if \; TOB \geq 24 \; and \; Number \; of \; accounts \geq 2 \end{cases} \quad (2)$$

Another example of a segmented scorecard tree is graphically illustrated by Fig. 2 (bottom left).

In principle, any candidate predictor can define a split. In practice, model developers and domain experts may avoid certain split variables, such as variables that are less trusted, difficult to interpret, or highly volatile. Segment scorecards can be developed independently from each other which is can speed up model development by a team.

To score out a new case, its segment is first identified and then the case is scored out using the associated scorecard, keeping computation light.

The deeper the segmentation tree, the higher the order of interactions the model can capture, and the more degrees of freedom can be devoted to refining the interactions. A single split at the root node (e.g. at 'TOB' = 24) can capture 2-way interactions between 'TOB' and all other characteristics. Further splitting a child node (e.g. 'TOB' >= 24), say, by 'Number of accounts' allows capturing 3-way interactions between 'TOB', 'Number of accounts' and all other predictors, etc. If a segmentation tree is allowed to grow infinitely deep, it can approximate arbitrary orders of interactions, rendering this model family a universal approximator. This is an asymptotic consideration. In practice segmented scorecard trees tend to be rather shallow. One quickly runs out of data, and deeper segmentation trees may underperform shallower ones due to over-fitting. In contrast to traditional classification and regression trees, which can grow deep and become difficult to comprehend, segmented scorecard trees tend to be rather shallow with no more than a few levels. This makes segmented scorecard trees easier to comprehend and to explain than traditional trees. Segmentation variables are typically selected not just with predictive power in mind, but also to make the resulting population segments easy to describe. The US FICO® Score uses more than a dozen segments tuned to distinctly different population segments, such as consumers with:

- Short credit history
- Long credit history without major blemishes
- Long credit history with major blemishes

and other segments that are defined by a segmentation tree of modest depth.

In summary, the family of constrained, segmented scorecards provides a very flexible, yet easy to interpret functional form, capable of representing complex predictive relations characterized by nonlinearities and interactions, whereby subtle domain knowledge can be imposed onto the structure of the segmentation and onto the functional forms of the segment scorecards. Interpretable shallow segmentation schemes with easy to explain scorecards associated with each segment thus form a Transparent Generalized Additive Model Tree (TGAMT). It is state of the art in the credit scoring industry to develop TGAMT's painstakingly by teams of data scientists and credit risk experts in a tedious interplay of "art and science" to simultaneously achieve high predictive performance and intuitive explanations how the scores are computed. The next section introduces a novel algorithmic approach to automatically learn TGAMT's.

III. LEARNING TRANSPARENT GENERLIZED ADDITIVE MODEL TREES

A. CART-like Greedy Recursive Search

Given a pre-existing segmentation scheme, developing the associated scorecards is a relatively easy task as long as the segments contain a sufficient number of informative training examples. Finding a good segmentation scheme is however a difficult problem, because the space of possible segmentations is extremely large. Domain knowledge tends to be insufficient to decide on an appropriate segmentation scheme. Due to the large number of possible solutions, it is unlikely that the "best" scheme with "optimal" score performance will ever be found. This is also not likely to be necessary, as there can be many good solutions that are close enough to optimal for all practical purposes. Similar to growing classification and regression trees [12] we apply a greedy recursive search heuristic to grow a TGAMT. Starting with the root node a set of candidate split variables and a finite set of split locations for each split candidate variable (e.g. taken at distribution deciles) are considered to split the current data set tentatively into two parts. It is evaluated whether there is a performance gain by fitting separate scorecards for each subset of the data instead of fitting a single scorecard to the entire current data. If so, the winning split that offers the greatest performance gain is made permanent. This process is performed recursively to grow a TGAMT until there is no more split that provides a performance gain exceeding some threshold, or until the number of training examples in the resulting segments falls below some minimum counts threshold. In the following, we

distinguish between two broad approaches to grow the tree: direct and ensemble-guided approaches.

(a) Direct approaches decide on splits based on measures relating to the original dependent variable, characterizing discriminatory power, such as divergence, KS (Kolmogorov-Smirnov statistic) or AUC (Area Under the Curve) for binary dependent variables, or closeness of fit (likelihood statistics) for binary or continuous dependent variables.

(b) Ensemble-guided approaches use a new dependent variable which is the ensemble learner's prediction of the original dependent variable. When the original dependent variable is binary it is sensible to generate the new dependent variable on the log(Odds) scale. A reasonable performance measure for this approach is closeness of fit between the segmented scorecard score and the ensemble prediction in the least squares sense.

Both approaches can employ cross-validation and use out-of-bag estimates to obtain unbiased empirical distributions of gains in the objectives associated with the tentative split. One can thus account for statistical significance when making split decisions. Corrections for multiple comparison testing are also possible. Cross-validation has benefits for smaller data sets (or at nodes in a deeper tree where data become scarce), as it stabilizes split decisions further thus mitigating the risk of over-fitting.

B. Challenges for Direct Approaches

From our experiments, direct approaches face challenges if the dependent variable is very noisy, which is often the case when predicting consumer credit behavior:

(a) As the tree grows, segment volumes decrease rapidly and variances of performance measures increase fast, making split or stopping decisions fraught with uncertainty. This can result in over-fitting and unreliable, unstable segmentation solutions.

(b) Setting minimum counts threshold too low makes it likely to over-fit. Setting the threshold too high makes it likely to under-fit because the tree may not be able to grow deep enough to capture complex interaction effects adequately.

(c) Results are sensitive to choice of the performance gain threshold.

(d) There is no notion of how close or how far away a heuristically derived segmentation solution is from the "optimum".

C. Benefits of Ensemble-Guided Approach

To mitigate these challenges, we developed the hybrid approach, as outlined in Fig. 2 (models shown in the figure are for illustrative use only.) First, the ensemble model is trained involving optimal hyper-parameter search. Various diagnostics for the best ML models are then generated. Data records are scored out by the best ML model and a "Best Score" variable is appended to the data. Next, the TGAMT is grown with the objective to approximate the "Best Score" in the Least Squares sense. Once a segmentation is accepted by domain experts, the segment scorecards can be fine-tuned

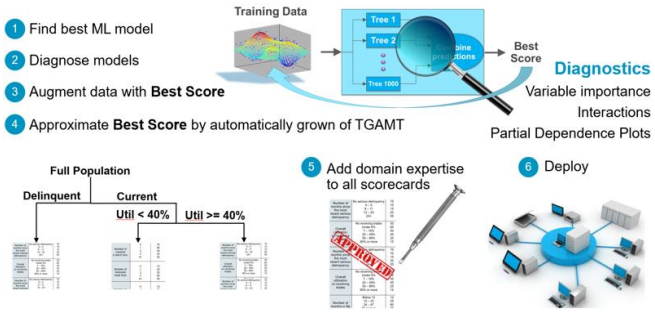

Figure 2. Process flow for learning TGAMT.

based on domain expertise. For example, characteristic selections, binnings, or constraints might be adapted with a view of the specific segment. Finally, the segmented scorecard system is deployed.

In our experience, replacing the original noisy binary dependent variable by a regression-smoothed "Best Score" as new dependent variable, greatly reduces sampling variance in scorecard parameters and uncertainties in split decisions when growing the tree, thus mitigating the risk of overfitting.

To provide motivation and evidence for the effectiveness of the ensemble-guided approach through a simplified experiment, consider the problem of binning the characteristics of a scorecard. If the binnings are too coarse, the relationship between the variables and the score becomes too inflexible to capture the signal accurately; the model is biased. Coarse step function approximations of true relationships, which are often expected to be smooth, may also not be palatable. If however the binnings are too fine, variances of fitted score weights tend to increase and models starts to over-fit. Noisy step function approximations are again not palatable. Typically, scorecard developers may use 5 to 15 bins per characteristic depending on the size of the training sample.

For both the direct and the ensemble-guided approach, we developed 10 scorecards each, all using the same fixed set of predictive variables (mostly ordinal continuous types), but distinguished by the granularity of their binning, ranging from an average of 3.8 bins/characteristic up to 40.9 bins per characteristic, which is a far finer binning than typical. Fig. 3 illustrates the over-fitting problem encountered by the direct approach, as the number of bins is increased. Predictive performance is measured by 5-fold cross-validated AUC. Findings for other common measures of score performance are qualitatively similar. The direct approach was implemented by training the scorecards to maximize divergence, using the original binary dependent variable. Findings for logistic regression are qualitatively similar. The direct approach reaches a performance plateau where it no longer improves beyond 8 bins/characteristic, and starts to over-fit the data beyond 20 bins/characteristic. In contrast, the ensemble-guided approach shows no signs of over-fitting within the tested range. For fine binnings with more than 12 bins/characteristic it is able to improve beyond the best models trained by the direct approach. Our findings indicate

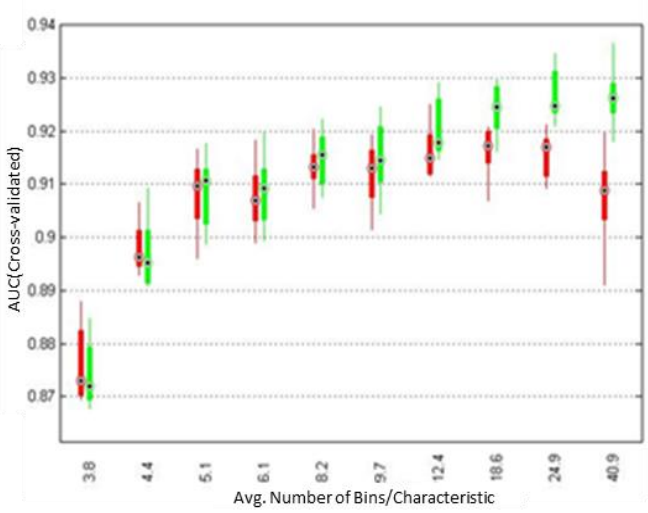

Figure 3. Effect of model degrees of freedom on performance for direct approach (red boxplot) versus ensemble-guided approach (green boxplot).

that the ensemble-guided approach is more resistant to over-fitting than direct approaches and therefore has a potential to train more flexible and more powerful scorecard models. These findings indicate that the ensemble-guided approach to growing TGAMT's will mitigate over-fitting challenges encountered by the direct approach and therefore lead to more stable and improved segmentation solutions.

The ensemble-guided approach has additional practical benefits over the direct approach by informing the learning of TGAMT through diagnostics obtained from ML:

(a) The list of candidate characteristics considered for inclusion into the TGAMT scorecards can be curtailed when it is found that certain variables may be unimportant as predictors in the ML models. Reference [13] proposes a statistic for input variable importance in the context of ensemble learning. Reducing the number of predictor candidates speeds up the learning of TGAMT.

(b) The list of candidate variables for segmentation splits can be informed by interaction diagnostics. Reference [13] proposes a statistic for testing whether any given variable interacts with one or more other variables. Variables with a high value of this statistic may be good split candidates. Variables that do not interact significantly with other variables may be removed from the candidate list of splitters, as no interactions need to be captured involving these variables. This can drastically reduce the search space and further speed up learning of TGAMT.

(c) ML models trained to maximize predictive power provide a "Best Score" upper bound on predictive performance. This bound informs TGAMT developers about the tradeoff they are willing to make between predictive performance and simplicity and transparency of the resulting TGAMT model.

IV. BENCHMARKING THE US FICO® SCORE AGAINST OPAQUE AND EXPLAINABLE MACHINE LEARNING APPROACHES

FICO® Scores are based on characteristics derived from credit bureau reports. The scores are designed to rank order the odds of repayment of credit while being easy to explain, such that higher scores indicate better credit quality. For our case study we chose the latest version of the US FICO® Score which is FICO 9. It is of interest whether ML models that are not restricted to be explainable might outperform FICO 9 by a substantial margin.

A. Predictive Performance Comparisons

We created "apples-to-apples" comparisons by developing a Stochastic Gradient Boosting (SGB) model and a multilayer Neural Network (NN) based on the same data set used to develop FICO 9, which consists of millions of credit reports. We allowed the same predictive variables that enter the FICO 9 model to enter the ML models. These comparisons thus provide insights into the potential impact of enforcing explainability constraints on score performance.

We also created "more data" comparisons for which we developed SGB and TGAMT models to investigate the potential performance gains possible for ML when taller and wider data are available for model development. For this we increased the number of candidate variables for the ML models to ca. 10 times as many variables than are input into the FICO 9 model (the additional variables are typically somewhat different versions of the variables used in the "apples-to-apples" comparisons). At the same time we also doubled the number of development records by sampling additional records from the same population from which the FICO 9 development data were sampled.

All important ML hyper-parameters, including learning rates, number of trees, depth of trees, minimum leaf size, number of random features for splitting, and number of hidden neurons, were tuned on a validation sample using multidimensional grid searches in order to warrant best possible performance of these models.

Table 1. compares model performance measures for discriminatory power (AUC, KS) for the various models and comparison scenarios, on bankcard accounts. All models are evaluated on an independent test sample that was not touched for model training and hyper-parameter tuning.

For the "apples-to-apples" comparisons the ML models mildly outperform FICO 9. It has been argued that marginal accuracy improvements observed under "laboratory conditions" may not carry over to the field where they can easily be swamped by other sources of uncertainty, such as

TABLE I. PERFORMANCE COMPARISON

Technology/Comparison	AUC	KS
FICO 9 Segmented scorecards	0.893	61.96
"Apples-to-apples" SGB	0.899	63.07
"Apples-to-apples" NN	0.895	62.48
"More data" SGB	0.902	63.92
"More data" TGAMT	0.894	62.19

changes to the environment and uncertain misclassification costs [14]. Therefore, from a practical perspective, these performance differences are minor. This finding supports that segmented scorecards are a very flexible model class capable of capturing nonlinear and interaction effects similar to complex ML models. Interestingly, explainability constraints on the FICO 9 model impact performance only slightly. This finding is in agreement with an often-made experience by scorecard developers, namely that enforcing explainability constraints, such as monotonicity, on the models often has little or no impact on score performance. This can be explained theoretically by the "Flat Maximum Effect", according to which "often quite large deviations from the optimal set of weights will yield predictive performance not substantially worse than the optimal weights" [14].

For the "more data" comparisons we observe a further mild performance improvement by SGB. The "Best Score" from this SGB model was used to guide the learning of the "more data" TGAMT, as described in Section 3. The resulting TGAMT performs practically on par with the FICO 9 Score. Inspection of its segmentation structure reveals a similar segmentation scheme as implemented by the FICO 9 model. The similarity between the automatically learned TGAMT structure and the laboriously derived FICO 9 segmentation structure is quite remarkable and illustrates the potential of TGAMT learning to increase the effectiveness of credit risk model development.

B. Opaqueness of Unconstrained Machine Learning

The opaqueness of ML models can be illustrated by exploring the input-output relationships captured by the models. It is possible to gain insights into the inner workings of ML models by plotting partial dependence functions [15]. These capture the average influences of single predictors, or sets of two or more predictors, on the score. 1-dimensional plots provide a summary of the average contributions of each predictor to the score, as illustrated for two variables in Fig. 4. We chose these variables as representative to illustrate certain problems with explaining opaque SGB models:

(i) Having longer 'Time on Books' intuitively should increase the score (reflecting higher credit quality). This experience is borne out from many score developments. This general directionality is indeed captured by our SGB model, except for many wiggles—presumably capturing noise—that cannot be explained. In our research, partial dependence functions for the more important predictors turned out to be directionally intuitive, except for noisy wiggles.

(ii) The contribution of 'Number of Trade Lines 30 Days Late' is directionally counterintuitive. It is very difficult to explain an increasing score with more trade lines showing late payments. Some of the less important predictors exhibited such counterintuitive behavior in our studies.

There are also two- and higher-dimensional versions of partial dependency plots that summarize joint effects of two or more predictors on the score. In our experience relating to credit scoring, these plots are often difficult to rationalize.

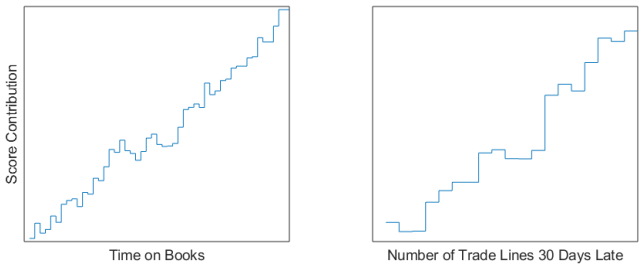

Figure 4. 1-dimensional partial dependence functions derived from the "more data" SGB model, for two predictive variables.

In our experiments these opaqueness phenomena were not artefacts of a specific model, but persisted under variations of hyper-parameter settings for SGB.

V. CONCLUSION

Complex modern ML techniques compete well with state-of-the-art credit scoring systems in terms of predictive power. However, their lack of explainability hampers trust and creates barriers for relegating high-stakes consumer lending decisions to these algorithms. In the artificial intelligence community the notion of an "explainable artificial intelligence" has been popularized whose lines of reasoning and decisions aim to be easily understood by humans, while hopefully not sacrificing substantial performance.

Our contribution is in a similar vein. We demonstrated how performance similar to that of complex and opaque ML models can be achieved within the family of explainable Transparent Generalized Additive Model Trees. The structure of these models was motivated by state-of-the-art credit risk scoring models. We discussed how TGAMT's can be learned automatically and effectively being guided by modern ML techniques. This contrasts with the rather painstaking, high-effort analytic processes, by which many credit risk scoring systems are being developed today. What makes TGAMT's different and more explanation-friendly than complex ML models, is that subtle domain expertise can be easily imposed into the model during its construction. Whereas opaque ML models search for the most predictive model in very large and less structured function spaces, TGAMT searches for the most predictive model in a smaller, more structured subspace of segmented, explainability-constrained scorecard models. We found that TGAMT's sacrifice very little predictive power compared to unconstrained ML models for the credit scoring problem we investigated. Our methods provide an effective approach to develop explainable credit risk scores, by effectively combining the benefits of data-driven ML with diagnostic information and with domain expertise.

This approach might also benefit other application areas where domain knowledge exists, where operational context needs to be taken into account during model construction, and where predictions and decisions need to be accurate, transparent, and easy to explain.

REFERENCES

[1] D. Gunning, "Explainable artificial intelligence research at DARPA," http://sites.nationalacademies.org/cs/groups/pgasite/documents/webpage/pga_184754.pdf [accessed November 2018].

[2] L. C. Thomas, "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers," International Journal of Forecasting, vol. 16, pp. 149-172, 2000.

[3] A. Feelders, H. Daniels, and M. Holsheimer, "Methodological and practical aspects of data mining," Journal Information and Management vol. 37, issue 5, pp. 271-281, 2000.

[4] J. H. Friedman, "Stochastic gradient boosting," Computational Statistics and Data Analysis, vol. 38, 2000, pp. 367-378, 2000.

[5] L. Breiman, "Random forests," Machine Learning, Volume 45, Issue 1, pp. 5-32, 2001.

[6] FICO White Paper, "Introduction to Model Builder Scorecard," http://www.fico.com/en/latest-thinking/white-papers/introduction-to-model-builder-scorecard [accessed November 2018].

[7] G. Scallan, "Selecting characteristics and attributes in logistic regression," Credit Scoring Conference CRC, pp 1-32, Edinburgh, 2011.

[8] M. A. Hardy "Regression with dummy variables," Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-093. Newbury Park, CA: Sage.

[9] E.E. Altendorf, A. C. Restificar, and T. G. Dietterich, "Learning from sparse data by exploiting monotonicity constraints," in Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh, pp 18-25, 2005.

[10] K. Bijak and L. Thomas, " Does segmentation always improve model performance in credit scoring?," Expert Systems with Applications, vol. 39, issue 3, pp. 2433-2442, 2012.

[11] D. J. Hand, S. Y. Sohn, and Y. Kim, "Optimal bipartite scorecards." Expert Systems with Applications, vol. 29, issue 3, pp. 684-690, 2005.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees, Wadsworth, 1984.

[13] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," Technical report, Department of Statistics, Stanford University, 2005.

[14] D. J. Hand, "Classifier technology and the illusion of progress," Statistical Science, vol. 21, number 1, pp. 1–15, 2006.

[15] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," The Annals of Statistics, vol. 29, number 5, pp. 1189–1232, 2001.