Combining Ensemble Learning and Human Expertise to Engineer Better Scorecards

Gerald Fahner
FICO
181 Metro Drive, Suite 700
San Jose, CA 95110 USA
+1 (408) 535-1500
geraldfahner@fico.com

ABSTRACT
Credit scoring applications are often based on the flexible, interpretable, and constrainable (in terms of functional form) model family of segmented scorecards. In this paper, we will briefly review scorecard technology and use cases for engineering scorecards based on domain knowledge. We will then describe a novel methodology to grow palatable segmented scorecard trees guided by ensemble learning, which we will illustrate by means of examples from credit scoring. Our methods are applicable in any supervised learning domain where the predictive content in historic data needs to be balanced with human expertise in order to meet requirements for future deployment.

Categories and Subject Descriptors
I.5.2 [**Pattern Recognition**]: Models - *statistical*, Design Methodology - *classifier design and evaluation, feature design and selection, pattern analysis.*

General Terms
Algorithms, Performance, Design, Reliability, Human Factors, Legal Aspects.

Keywords
Credit Scoring, Scorecards, Predictive Models, Segmentation Trees, Model Interpretation, Domain Knowledge, Constraining Functional Dependencies, Ensemble Learning, Stochastic Gradient Boosting.

1. INTRODUCTION
Credit scoring has been an early, highly successful and pervasive data mining application. For a comprehensive survey of credit scoring see [1]. 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 assignment, 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 confidence into reasonableness and robustness of the deployed models.

Conference'10, Month 1–2, 2010, City, State, Country.

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 [2]. 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. As described in section 2., scorecard technology supports inclusion of subtle 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 allow for 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 ten segmented scorecards. Designing a segmented scorecard system has traditionally been a labor-intensive and fairly subjective process, during which several segmentation schemes are hypothesized from domain knowledge and guided by exploratory data analysis, then tested and refined to the extent possible given development resources, which could benefit greatly from more productive approaches.

Independent from the industrial developments and with a different focus, powerful machine learning ensemble methods, such as stochastic gradient boosting [3] and random forests [4], have been devised in academia. These procedures can automatically learn almost arbitrarily complex relations from data, and despite their high flexibility, generalize well to new data if drawn from the same population. These procedures are very 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 model diagnostics. Straightforward applications include variable selection and interaction detection. But these procedures and the resulting models can be harder to explain to users or regulators, and they don't support inclusion of subtle domain knowledge into the models. This can render them 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 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 2. reviews the segmented scorecard technology and gives examples of imposing domain knowledge into scorecards. Section 3. describes a novel methodology to grow segmented scorecard trees guided by the predictive content of a stochastic gradient boosting model. Section 4. reports experiments and results from credit scoring and insurance claims fraud projects, before discussing other business applications and concluding.

2. 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 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.

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 finite 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 discrete values. Missing values, even different types of missing or special values, are incorporated naturally as additional bins into characteristics. For example, the missing values could be an indicator of risk if a consumer didn't answer a question, in which case it should not be ignored. Or they could be a temporary issue in the historic data which may not replicate itself in the future (an example of non-stationary 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 any variables can be used as scorecard predictors, they need to be binned. Various methods exist for binning and improvements to binning have been proposed [5].

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.

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

where each of the P characteristic scores is a stair function defined over the q bins of its underlying characteristic. The heights of the stairs are given by score weights associated with the bins. The structure of this model is similar but not identical to dummy variable regression [6]. While dummy variable regression has no notion of characteristics and dummy variable selection happens on the level of dummy variables, which tends to put "holes" into binned characteristics, some scorecard technologies perform variable selection on the characteristic level. This makes the resulting models easier to interpret. In addition, FICO's scorecard development platform has the capability to constrain characteristic scores to desired shapes, such as monotonicity, over any subsets of bins.

A powerful feature of scorecards is their ability to model nonlinear effects through nonparametric stair functions. Model eq.(1) is easy to interpret and to calculate as it can be represented in tabular form made up of the characteristics, their bins, and their associated score weights. This was historically important and still is today, to gain business users' trust in these models.

Estimation of the score weights in eq.(1) is possible using many approaches. FICO's technology allows for specification of various objective functions, including penalized maximum likelihood for regression, ranking and classification, and penalized maximum divergence, which is 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 are used to optimize the score weights subject to linear equality and linear inequality constraints. These constraints provide a key mechanism 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 'Age', assuming for simplicity only 3 age bins:

$$H_{Age} = S_{1,Age} 1\{18 \le Age \le 25\} + S_{2,Age} 1\{26 \le Age \le 46\} + S_{3,Age} 1\{Age \ge 47\}$$

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

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

With this, the optimization will solve for the best possible set of score weights, subject to the desired monotonicity.

Enforcing monotonicity can be useful for various reasons: (i) To make a model adhere to theoretical expectations, for example, everything else being equal one might expect higher credit quality associated with higher income in a risk score development, or one might expect lower demand with higher price in a marketing response model. (ii) Constraints reduce the hypothesis space or the effective degrees of freedom of the model family, hence if constraints are applied sensibly, a constrained model can be less prone to over-fitting [7]. (iii) 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 contradict this, in which case a monotonicity constraint can rectify the desired relation. Similar considerations apply when adverse decisions such as credit rejections need to be justified to customers.

The scorecard format comprises a flexible family of functions capable of modeling nonlinear effects of predictors on the score

in as constrainable stair functions. However, in the form written in eq.(1) it is an additive function of the predictors and cannot capture interactions. If the true relationship is characterized by strong interactions, then the model will under-fit the data. To overcome this limitation, several approaches are in use, including: (i) Creation of derived predictors such as ratios between the original predictor variables, which is part of data pre-processing and outside the scorecard. (ii) Inclusion of cross-characteristics into the models, which generate products of bin indicator variables. (iii) Segmented scorecards. In the following we will focus on segmented scorecards, which are most widely used in the financial services, probably because the models are easy to inspect, to interpret and to engineer.

[8] includes an overview of reasons for undertaking segmentation and discusses current practices. It features a study reporting mixed results with a research algorithm for finding good segmentations for 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 homogeneity in the segments for various managerial reasons. On the other hand, the findings in [9] are more upbeat about the predictive benefits of 2-way segmentations for improving the discriminatory power of the resulting more flexible 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 much 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, we believe that mitigation of over-fitting during segmented scorecard development might be necessary for segmentation success.

Scorecard segmentations are often represented as binary tree structures. The root node represents the entire input space. Below it, the population is split into child nodes, as defined by a first split variable. Child nodes may be split further until splitting eventually stops and leaf nodes are created. The leaf nodes contain the scorecards. In principle, any candidate predictor can be used to define a split, although a scorecard developer may shy away from certain variables less trusted. For example:

$$Score(X) = \begin{cases} Score_1(X) \; if \; Age < 30 \\ Score_2(X) \; if \; Age >= 30 \; and \; Income < \$20{,}000 \\ Score_3(X) \; if \; Age >= 30 \; and \; Income >= \$20{,}000 \end{cases} \quad (2)$$

Given a segmentation, the associated scorecards are trained independently for each segment. To score out a new case, its segment must be identified, and the case then scored out by the respective scorecard.

The deeper the segmentation tree, the higher the order of interactions a segmented scorecard can capture and the more degrees of freedom can be devoted to a given interaction. A single split at the root node (e.g. at 'Age' = 30) can capture 2-way interactions between 'Age' and all other characteristics. Further splitting a child node (e.g. 'Age' >= 30), say, by 'Income' allows capturing 3-way interactions between 'Age', 'Income' and all other predictors, etc. If a segmented Scorecard tree is allowed to grow infinitely deep, it can represent functions of arbitrary order. This is an asymptotic consideration because in practice and with limited data, segmented scorecard trees tend to be rather shallow. One quickly runs out of data and a deeper segmentation tree may greatly underperform a shallower one due to over-fitting. Shallowness is a virtue on the other hand because it aids interpretability.

In summary, the family of constrained segmented scorecards provides a potentially powerful, yet easy to interpret functional form to capture complex predictive relations characterized by nonlinearities and interactions, whereby subtle domain knowledge can be imposed onto the functional forms of the segment scorecards. Over-fitting appears however as a significant threat to segmentation success. We now propose a novel way to mitigate over-fitting during the search for a segmented scorecard tree.

3. GROWING SEGMENTED SCORECARD TREES

3.1 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. Similar to growing classification and regression trees [10], we apply a greedy recursive scorecard segmentation search heuristic to grow a segmented Scorecard tree. Starting with the root node, a set of candidate split variables and a finite set of split locations (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 part of the data instead of fitting a single scorecard to the current data. If so, the winning split that offers the greatest performance gain is made permanent. The process is performed recursively 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. Direct approaches make split decisions based on measures relating to the original dependent variable, characterizing either discriminatory power such as such as Divergence, KS, or ROC Area in the case of binary dependent variables, or closeness of fit (likelihood statistics) for binary or continuous dependent variables.

Ensemble-guided approaches relate to a new dependent variable which is the ensemble learner's prediction of the original dependent variable. When the original dependent variable is binary, then 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 empirical distributions of gains in the objectives associated with the tentative split. One can thus account for statistical significance when making spit 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.

3.2 Challenges for Direct Approaches

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

- As the tree grows, segment volumes decrease rapidly and variances of performance measures increase fast, making split or stopping decisions fraught with uncertainty, which can result in over-fitting and unreliable, unstable segmentation solutions
- Setting minimum counts threshold too low makes it likely to over-fit either to training data or to test data, whatever data set is used to evaluate split performance
- Setting minimum counts 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
- Results are sensitive to choice of performance gain threshold
- There is no notion of how close or how far away a heuristically derived segmentation solution is from the "optimum"

3.3 Ensemble-Guided Approach

To mitigate the aforementioned challenges, we developed a hybrid approach, whereby an ensemble model is trained and its predictions ("ideal score function") used as the new dependent variable for developing a segmented scorecard tree (Figure 1). A segmented scorecard tree is then trained with the objective to approximate the ensemble prediction in the least squares sense. Replacing the original dependent variable with the regression-smoothed ensemble prediction greatly reduces sampling variance in the scorecards, performance measures, and split decisions, and therefore mitigates the risk of over-fitting.

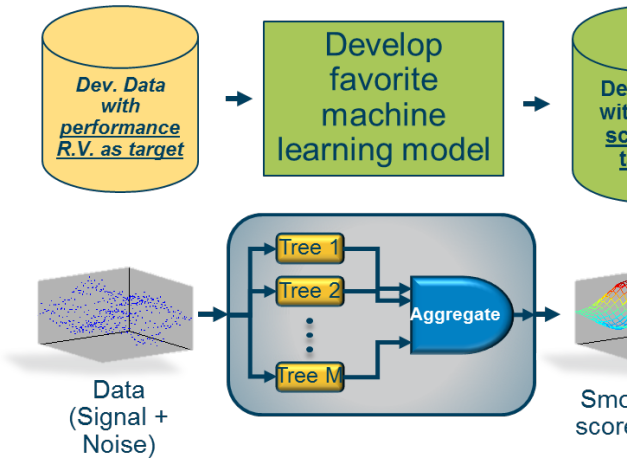

Figure 1. Ensemble-guided appr

To provide motivation and evidence for the effectiveness of the ensemble-guided approach in a simpler context, consider the problem of binning variables 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. Coarse step function approximations of true relationships, which are often expected to be smooth, may also not be very palatable. If the binnings are too fine, the variances in the fitted score weights tend to increase and the model starts to over-fit. Noisy step function approximations are again not very palatable. Typically one may find between 5 to 10 bins per characteristic in scorecards. 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 3.8 bins/characteristic up to 40.9 bins per characteristic, which is far more than one would normally find in a scorecard. Figure 2. illustrates the over-fitting problem for the credit scoring data set presented in section 4, where we picked AUC to compare the models.

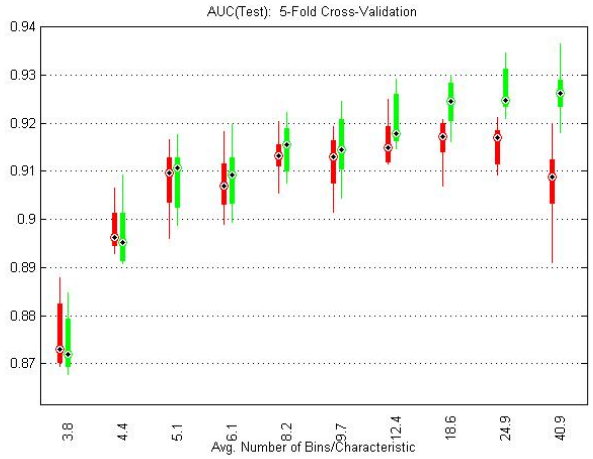

Figure 2. Effect of model degrees of freedom on performance, direct approach (red bars) Vs. ensemble-guided approach (green bars)

Findings for other common measures of score performance are qualitatively similar to Figure 2. The direct approach was implemented by training the scorecards to maximize Divergence. Findings for logistic regression are qualitatively similar. The direct approach reaches a performance plateau where it no longer improve beyond 8 bins per characteristic, and starts to over-fit the data beyond 20 bins per characteristic. In contrast, the ensemble-guided approach shows no signs of over-fitting within the tested range. For very fine binning, it improves beyond the best models trained by the direct approach. Our findings indicate that the ensemble-guided approach is rather resistant to over-fitting and therefore has a potential to train more flexible and more powerful scorecard models which are less encumbered by over-fitting troubles. This raises hopes that the ensemble-guided approach to growing segmented scorecard trees will mitigate challenges for the direct approach and lead to more stable and improved segmentation solutions.

The ensemble-guided approach may be further enhanced and informed by interaction statistics derived from learned stochastic gradient boosting ensembles. [11] proposes a statistic for testing whether a specified variable interacts with one or more other variables. Variables with a high value of this statistic may be good split candidates. Such an analysis may be applied recursively to identify promising split candidates according to this statistic, locally at each node of the growing tree. However, this idea has not been further pursued in the present paper.

4. CASE STUDY

We applied our proposal to a publicly available data set 'hmeq.xls' which, as of 02/22/2013, can be downloaded from:

http://old.cba.ua.edu/~mhardin/DATAMiningdatasets2/

The data set comprises 5,960 observations of home equity loans of which 4,198 paid back the loan (Good) and 1,762 defaulted (Bad), which is encoded by a binary (Good/Bad) outcome flag. There are 12 candidate predictors some of which have a substantial number of missing values. The data were randomly split by the author into a training set comprising 70% of the observations and a remaining test set of 30%. Figure 3. shows typical results from 3 models which use all 12 predictors. We performed several model runs for the direct and the ensemble-guided approaches, by varying important parameters for the scorecard segmentation search including average number of bins per characteristic, minimum counts threshold, and performance gain threshold. We found no parameter setting where the direct approach generated a model that came close to the ensemble performance. The direct approach showed little consistency in the splits it found gainful, if any, between the different runs. We concluded that a single scorecard worked best for this problem when using the direct approach. In contrast, the ensemble-guided approach generated several segmented scorecard solutions with performances that came much closer to stochastic gradient boosting. The segmentation tree structures varied somewhat with the parameter settings, but there was more consistency between the trees in terms of the splits made, albeit sometimes in different order (Figures 4., 5.)

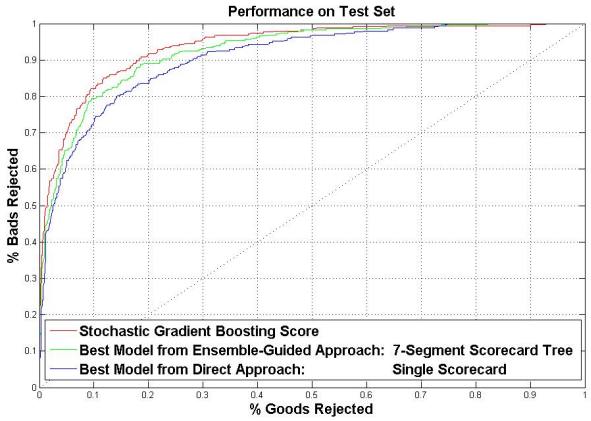

Figure 3. Comparison of 3 models. Stochastic Gradient Boosting dominates the comparison. A segmented scorecard tree with 7 segments grown by the ensemble-guided approach comes second. The best model found by the direct approach is a single scorecard which comes last.

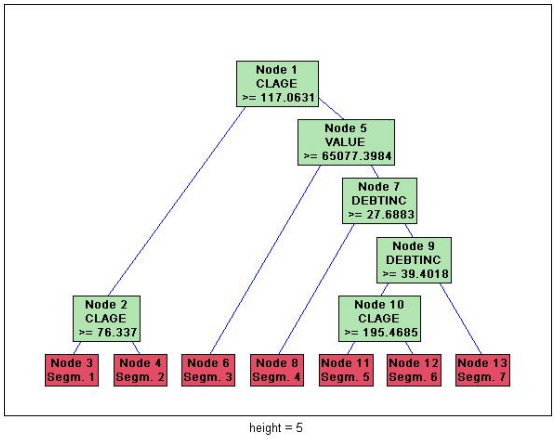

Figure 4. 7-segment solution referred to in Figure 3.

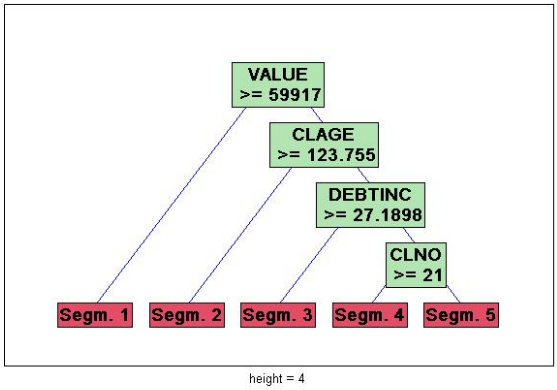

Figure 5. 5-segment solution using ensemble-guided approach, using a finer binning than for the model referred to in Figure 4. Similar variable sets are used for splitting.

We also performed cross-validation to obtain empirical distributions of predictive gains from tentative splits. The cross-

validated gains have wider confidence intervals for the direct method as compared to the ensemble-guided method, reflecting the higher variance of models and performance measures with the direct method. Wider confidence intervals produce more spurious splits that don't validate on new data, as compared to the ensemble-guided approach which produces more reliable split decisions. We believe that the ensemble-guided approach to scorecard segmentation offers several advantages over the direct approach:

- Sampling variance in scorecards and derived performance objectives is greatly reduced, owing to the replacement of a noisy with a smoothed dependent variable.
- As a consequence, split decisions become more reliable, trees become more stable and eventually grow deeper.
- As deeper trees can more reliably be grown, more of the true interaction structure of the data can be captured without this advantage being overshadowed by over-fitting.
- By comparing with the ensemble learner, we can measure how close a segmented scorecard solution remains from the best attainable prediction. This provides an informative benchmark for scorecard development.

5. OTHER BUSINESS APPLICATIONS

We're currently rolling out this methodology in "benchmarking mode" to explore it for a wider range of projects from credit scoring and fraud where results can be compared against more established approaches, in the laboratory as well as in the field. While field results have not yet been observed, we see good potential in the lab for insurance claims fraud prediction, which is traditionally an area where neural nets are predominant. Apart from possible predictive improvements over neural nets that we're seeing for several datasets, we also find secondary advantages that occasional data problems in claims fraud data are easier to spot and to remedy by using the interpretable and engineerable scorecard format. In addition, users such as claims fraud investigators can better contribute their subtle domain knowledge to aid the refinement of claims fraud scoring solutions.

6. DISCUSSION

Machine learning algorithms for prediction and classification developed in academia hold a potential for improving credit and other industrial scoring systems which drive high volume, high stakes decisions. However, off-the-shelf procedures cannot be used directly in many circumstances. There are often data limitations and operational constraints to be dealt with. A model that looks good in terms of "paper performance" may be rejected by the user because it defies common sense. It requires human expertise to engineer sound models in those circumstances. Off-the-shelf machine learning procedures almost by definition ignore human expertise. Predictive analytics practitioners with the need to incorporate domain expertise into their models don't need to choose between a complex black box and a transparent yet inflexible model. There are good intermediate options such as segmented scorecard systems which can capture complex relations yet are easy to inspect and to engineer. Scorecard segmentation has traditionally been more an art than a science, but with recent research and developments in both academia and in the industry we believe that we got a little closer to productively combining powerful tool sets from academia and industry.

7. REFERENCES

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

[2] Feelders, A., Daniels and H., Holsheimer, M. 2000. Methodological and practical aspects of data mining. *Journal Information and Management* 37, 5 (2000), 271-281.

[3] Friedman, J. H. 1999. Stochastic Gradient Boosting. *Computational Statistics and Data Analysis*, 38 (2000), 367-378.

[4] Breiman, L. and Schapire, E. 2001. Random forests, *Machine Learning*, 5-32 (2001).

[5] Scallan, G. 2011. Selecting characteristics and attributes in logistic regression. *Credit Scoring Conference CRC, Edinburgh*, (2011).

[6] Hardy, M. A. 1993. *Regression with dummy variables*. Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-093. Newbury Park, CA: Sage.

[7] Altendorf, E. E., Restificar, A. C. and Dietterich, T. G. 2005. *Learning from sparse data by exploiting monotonicity constraints*. Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, Edinburgh, Scotland.

[8] Bijak, K. and Thomas, L. 2012. Does segmentation always improve model performance in credit scoring? *Expert Systems with Applications*, 39, 3, (2012), 2433-2442.

[9] Hand D.J., Sohn S.Y. and Kim Y. 2005. Optimal bipartite scorecards. *Expert Systems with Applications*, 29 (2005) 684-690.

[10] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone., C. J. 1983. *Classification and Regression Trees.* Wadsworth.

[11] Friedman, J. H. and Popescu, B. E. 2005. *Predictive Learning via Rule Ensembles*. Technical report. Department of Statistics, Stanford University (2005).