Imposing Domain Knowledge on Algorithmic Learning

An Effective Approach to Construct Deployable Predictive Models

Dr. Gerald Fahner
FICO
August 28, 2013

Objective

» What lenders seek from models:

1. Accurate predictions of customer behavior
2. Insights into fitted relationships
3. Ability to impose domain knowledge before deploying models

» Algorithmic learning procedures address 1. and 2., but pay scant attention to 3.

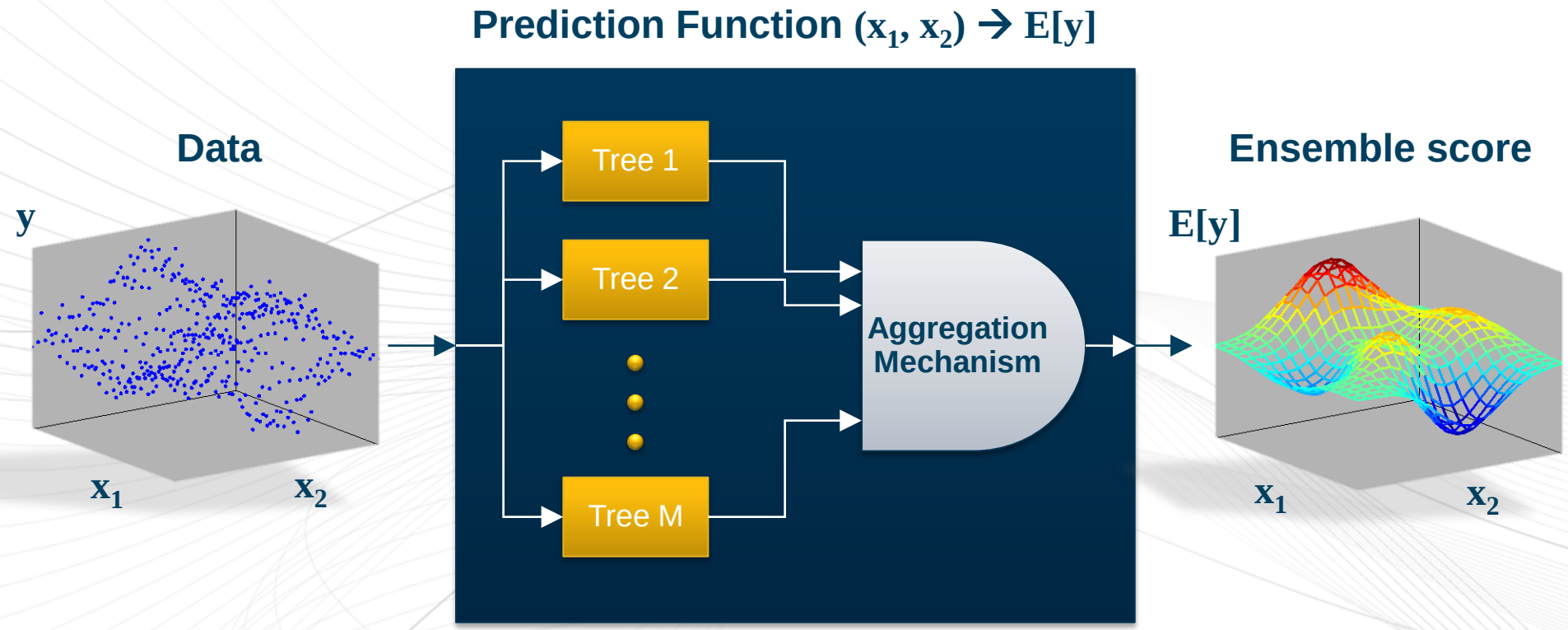
» As a consequence, algorithmic models may not be deployable for some credit scoring applications

➢ We propose an effective approach to impose domain knowledge on algorithmic learning that paves the way for deployment

Agenda

- » Algorithmic Learning Illustration
- » Case Study - Modeling US Home Equity Data
  - » Diagnosing Complex Models
  - » Algorithmic Learning-Informed Construction of Palatable Scorecards
- » Summary

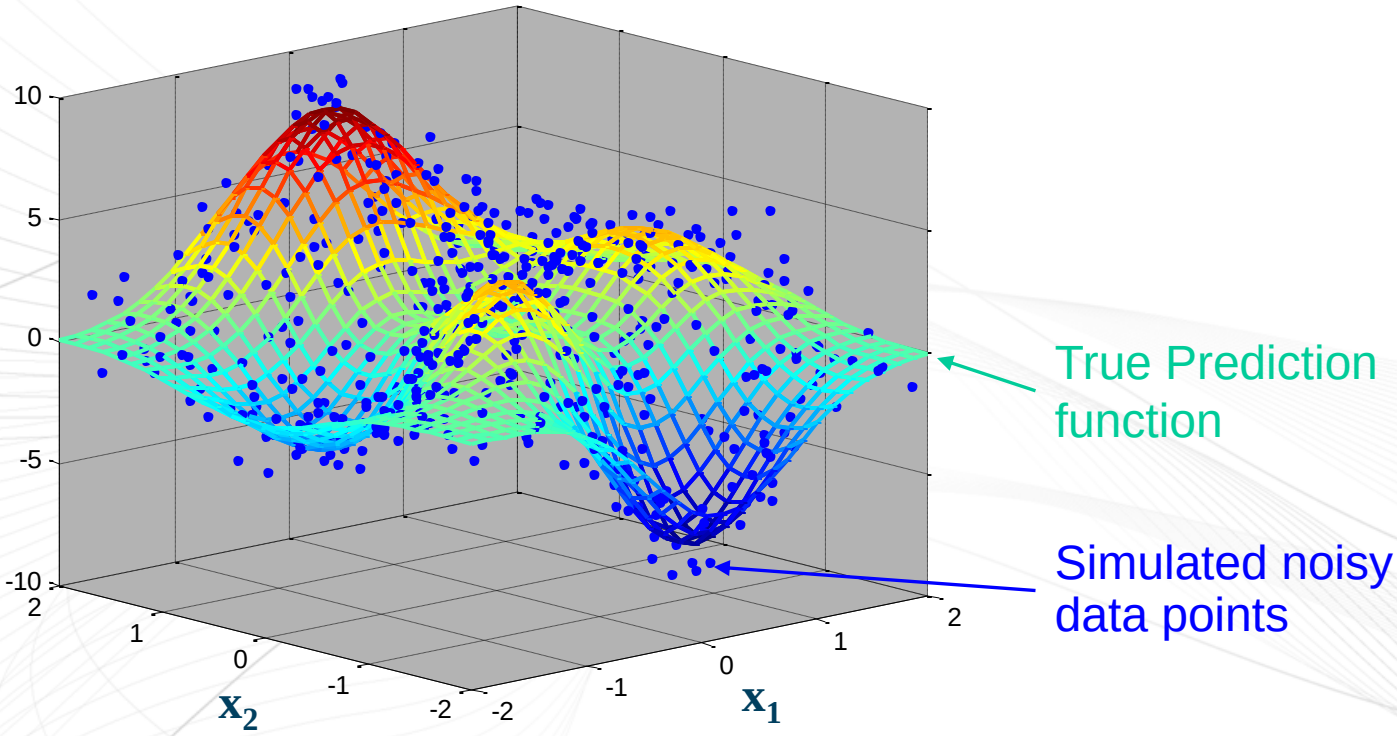
Sketch of Tree Ensemble Models



Examples:
Random Forest [1]
Stochastic Gradient Boosting [2]

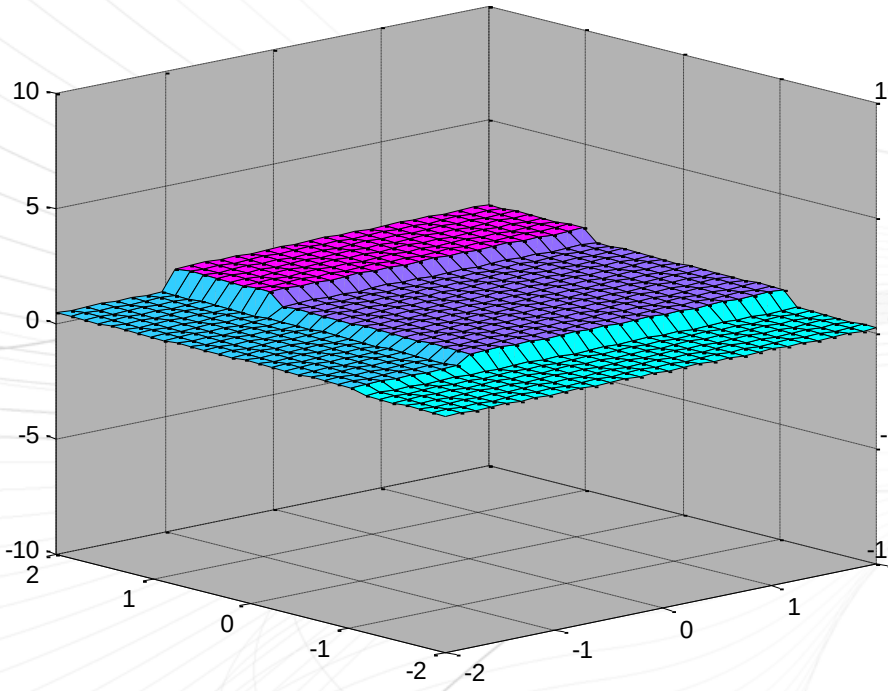
Demonstration Problem

Simulated noisy data from an underlying “true” prediction function to create a synthetic development data set

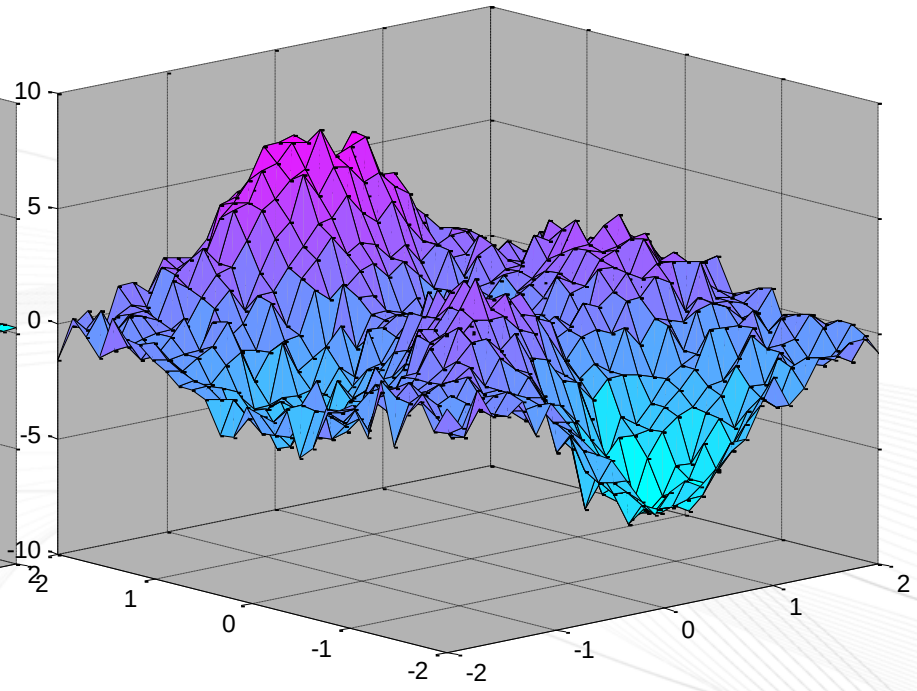


Next fit Stochastic Gradient Boosting to approximate the data

Stochastic Gradient Boosting
1 Tree

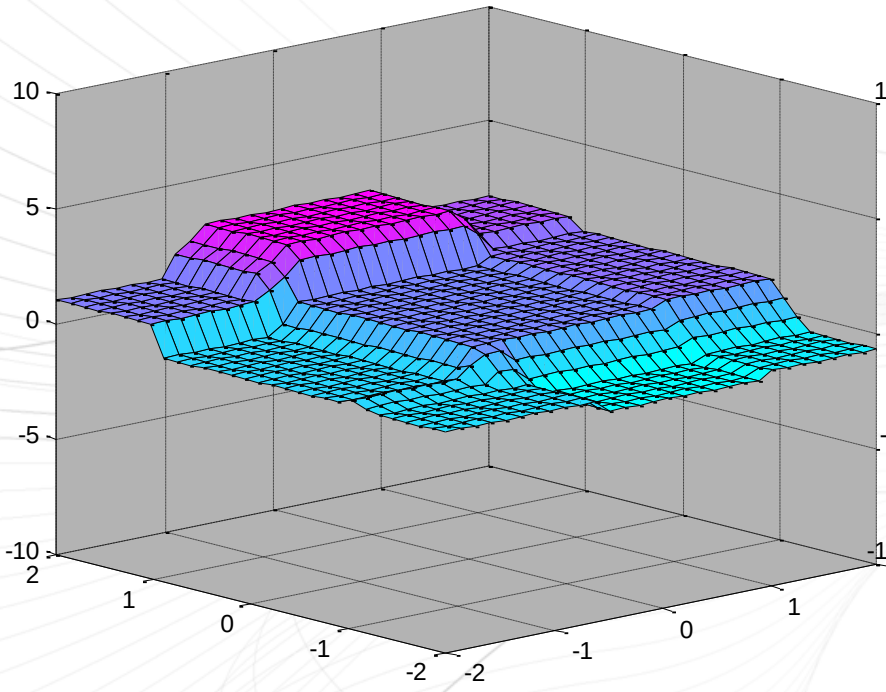


Prediction Function

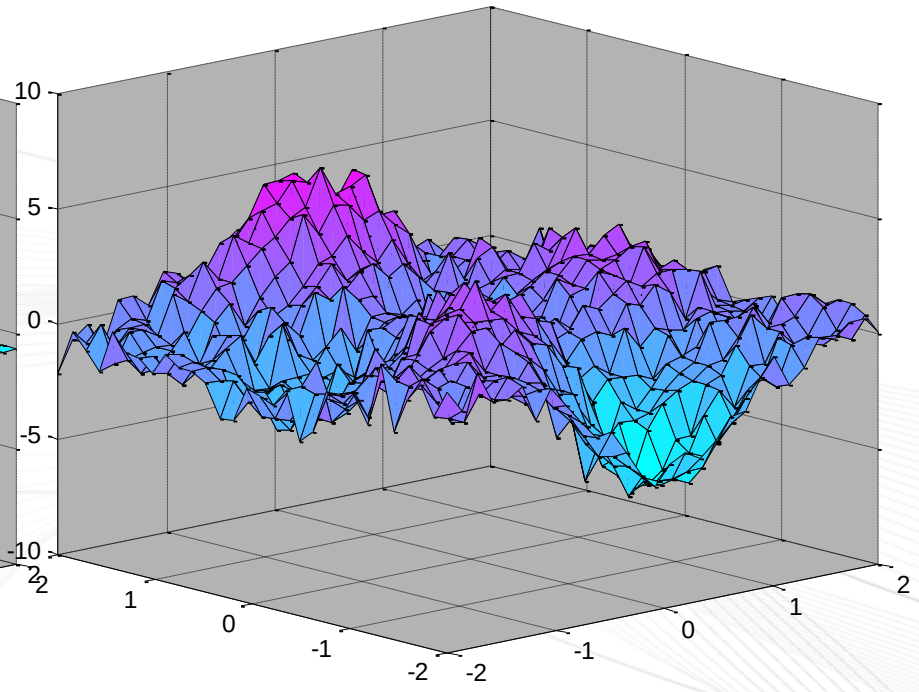


Residual error

Stochastic Gradient Boosting
5 Trees

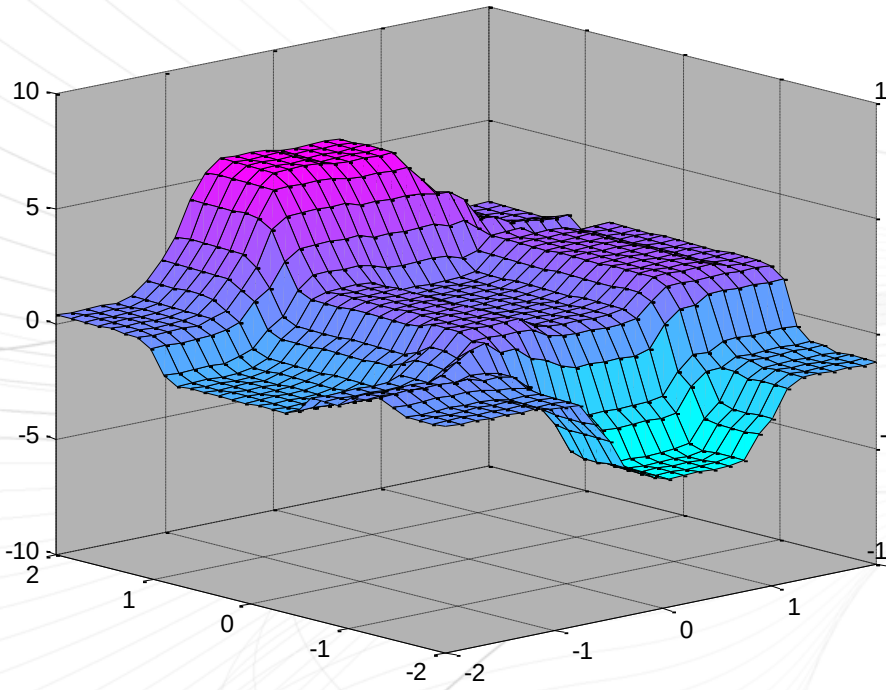


Prediction Function

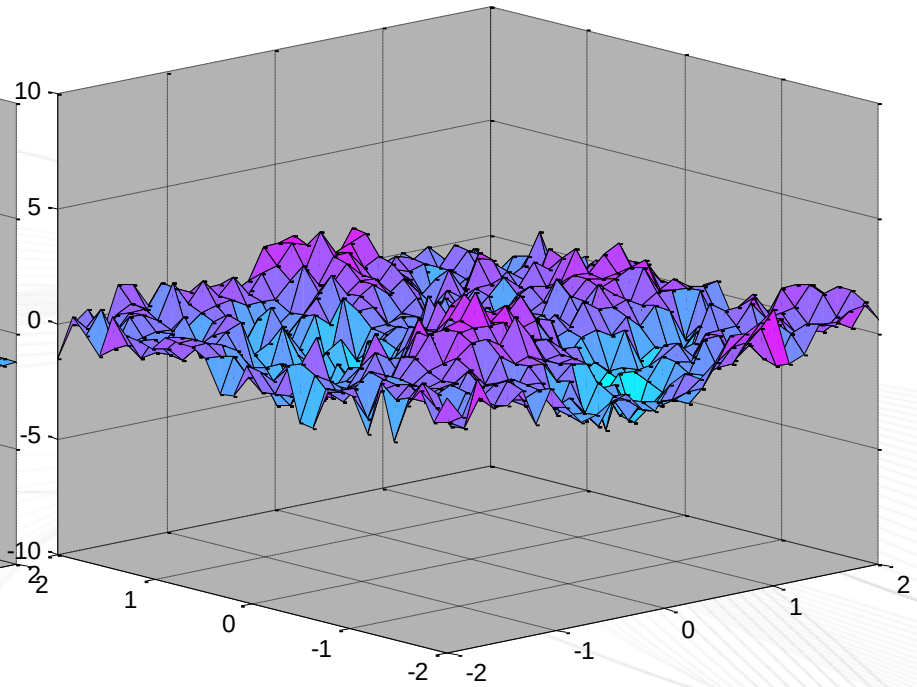


Residual error

Stochastic Gradient Boosting
25 Trees

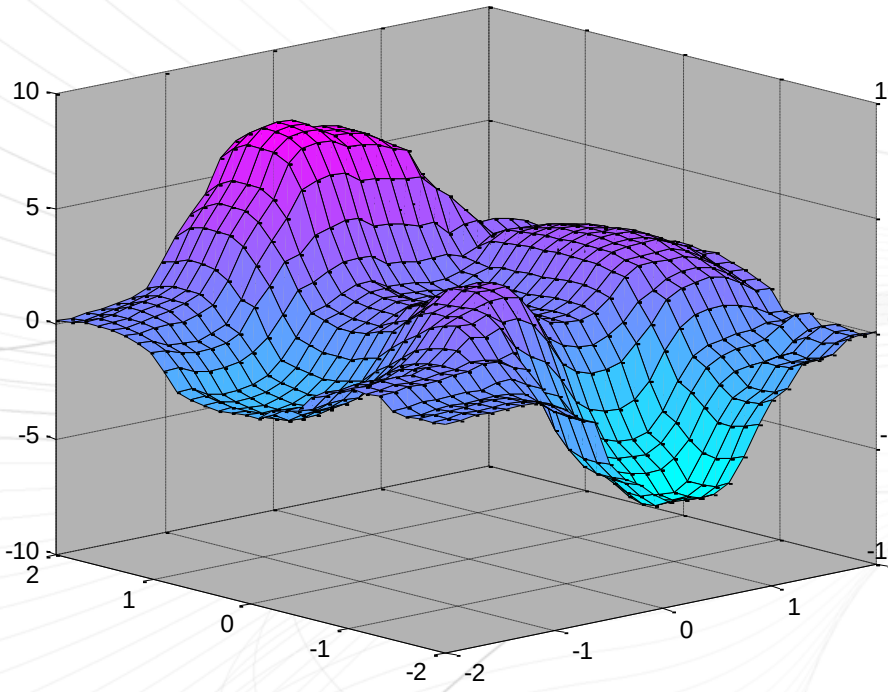


Prediction Function

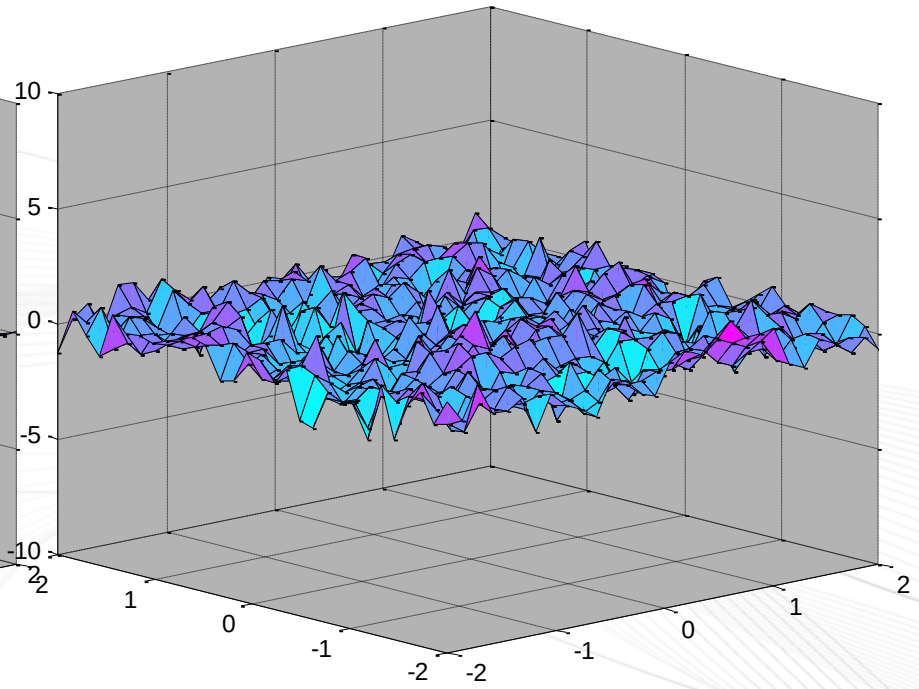


Residual error

Stochastic Gradient Boosting
100 Trees

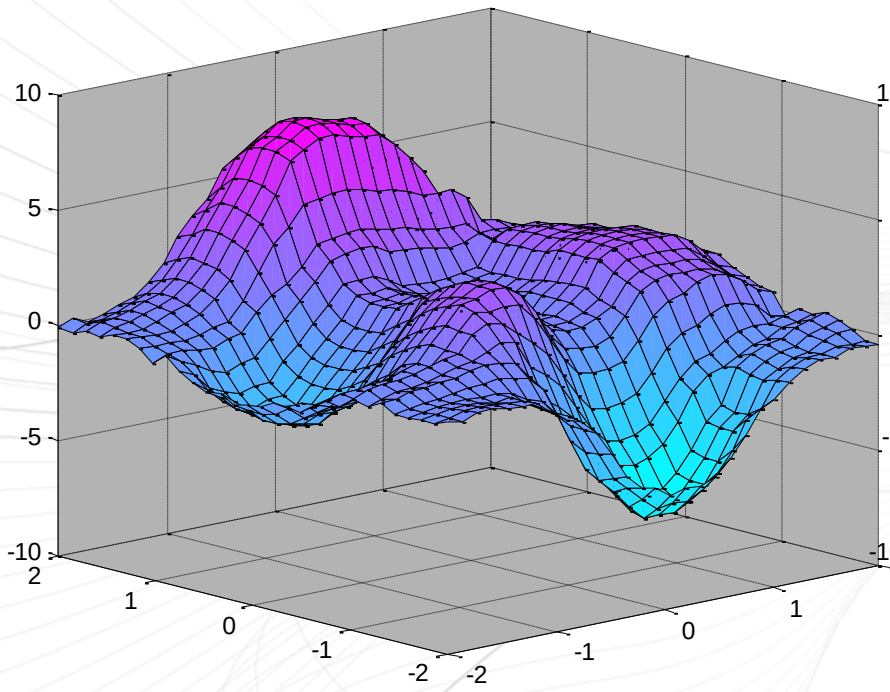


Prediction Function

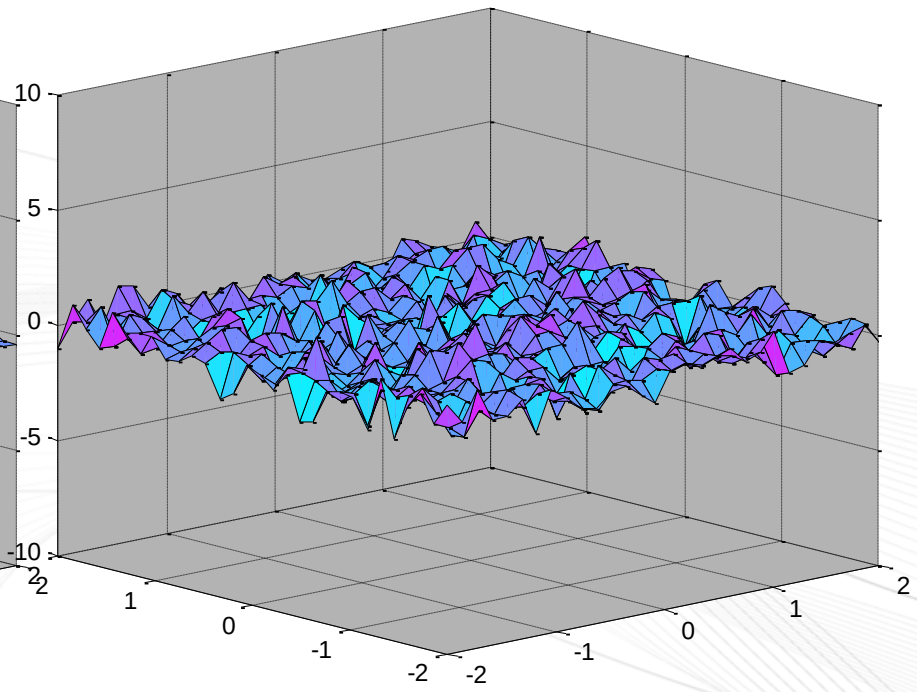


Residual error

Stochastic Gradient Boosting
200 Trees



Prediction Function



Residual error
pure noise

Agenda

- Algorithmic Learning Illustration
- Case Study - Modeling US Home Equity Data
  - Diagnosing Complex Models
  - Algorithmic Learning-Informed Construction of Palatable Scorecards
- Summary

Problem and Data

Predict likelihood of default on US home equity loans. Binary Good/Bad target.

Sample size: N(total) = 5,960, N(bad) = 1,189. 12 predictor candidates

Predictor Candidates	Description
REASON	'Home improvement' or 'debt consolidation'
JOB	Six occupational categories
LOAN	Amount of loan request
MORTDUE	Amount due on existing mortgage
VALUE	Value of current property
DEBTINC	Debt-to-income ratio
YOJ	Years at present job
DEROG	Number of major derogatory reports
CLNO	Number of trade lines
DELINQ	Number of delinquent trade lines
CLAGE	Age of oldest trade line in months
NINQ	Number of recent credit inquiries

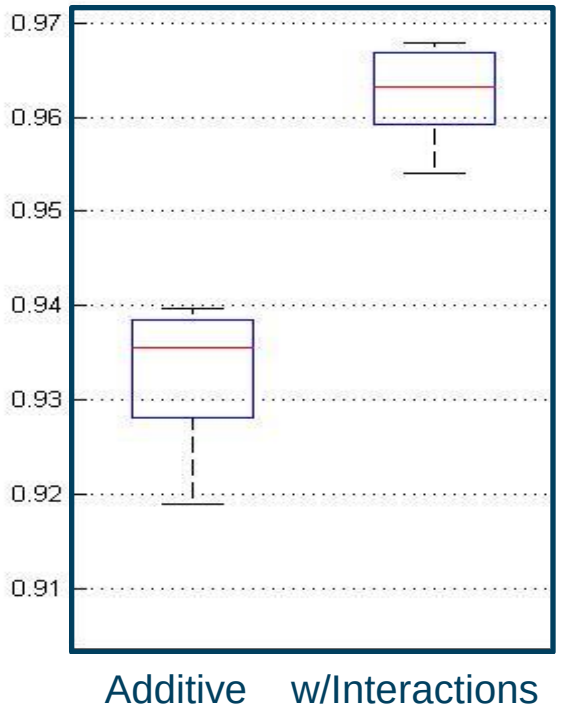
Data source as of 07/25/2013: http://old.cba.ua.edu/~mhardin/DATAMiningdatasets2/hmeq.xls

Stochastic Gradient Boosting (SGB) Experiments

» Use SGB to approximate log(Odds) of being “Good” [3]

» Will use Area Under Curve (AUC) as evaluation measure throughout

5-fold cross-validation of AUC



1. Model that is additive in the predictors (trees have 2 leaves)
2. Model cable of capturing interactions between predictors (trees have 15 leaves)

➢ Interactions appear to be substantial

Eminently Useful Model Diagnostics
Following [4]

Variable Importance
Relative to most important variable

DEBTINC	1.00
DELINQ	0.49
VALUE	0.45
CLAGE	0.40
DEROG	0.34
LOAN	0.25
CLNO	0.24
MORTDUE	0.23
NINQ	0.23
JOB	0.20
YOJ	0.20
REASON	0.07

Interaction Test Statistics
Whether a variable interacts with any other variables

DEBTINC	0.029
VALUE	0.022
CLNO	0.014
CLAGE	0.013
DELINQ	0.010
YOJ	0.008
MORTDUE	0.007
LOAN	0.007
DEROG	0.006
JOB	0.006

Score out complex model at regular grid points across the range of one predictor at a time (here x_1), while keeping all other predictors fixed

Here we fix the other predictors to their values for observation 1

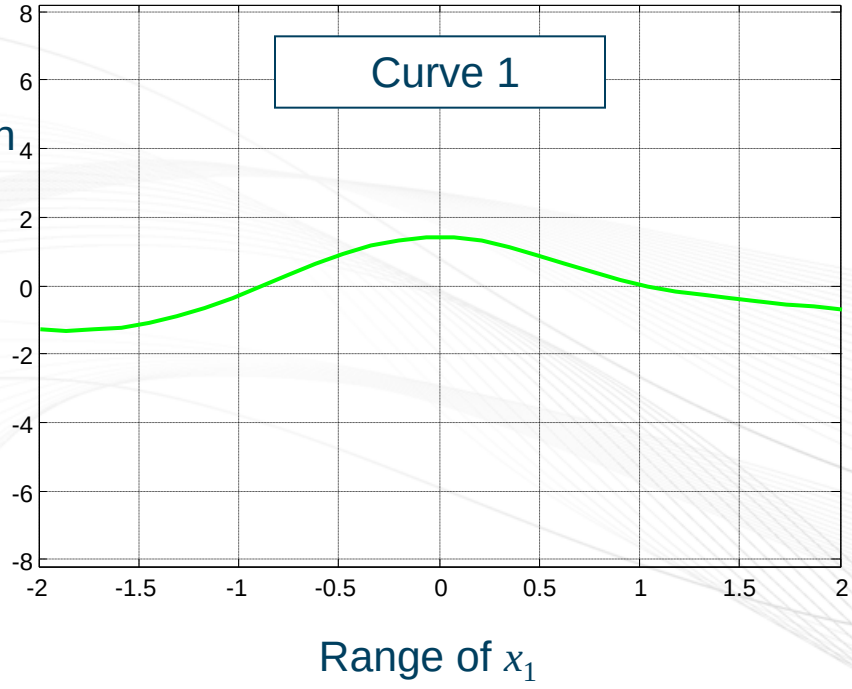
Sweep through range of x_1

$(x_1\ x_{12} \ldots x_{1P})$

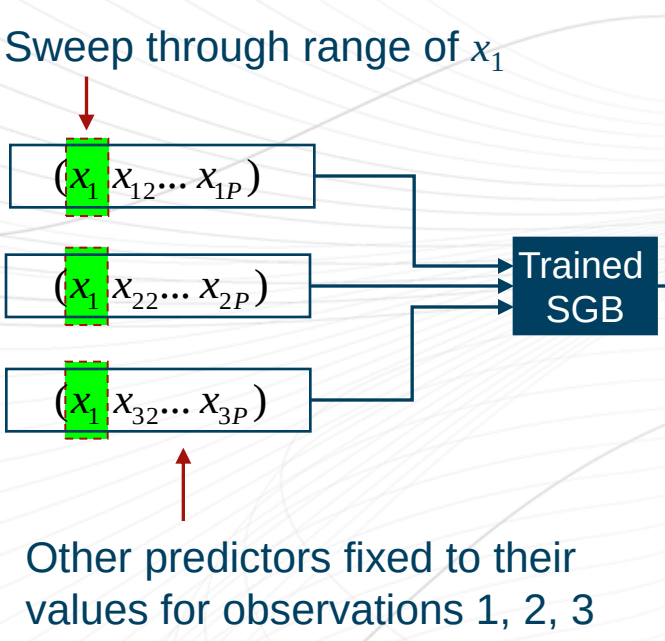
Other predictors fixed to their values for observation 1

Trained SGB

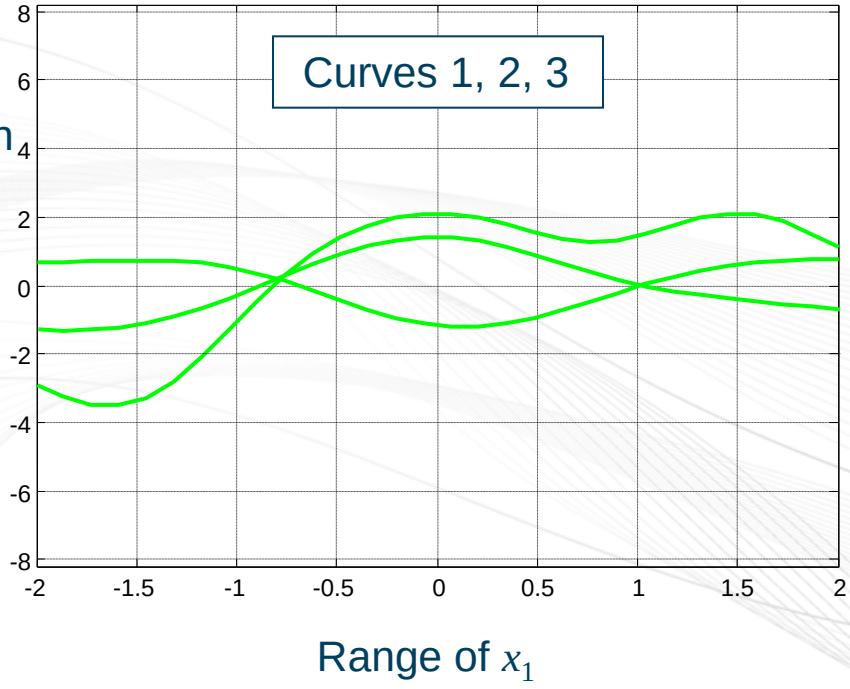
Score (logOdds) as function of x_1, after centering



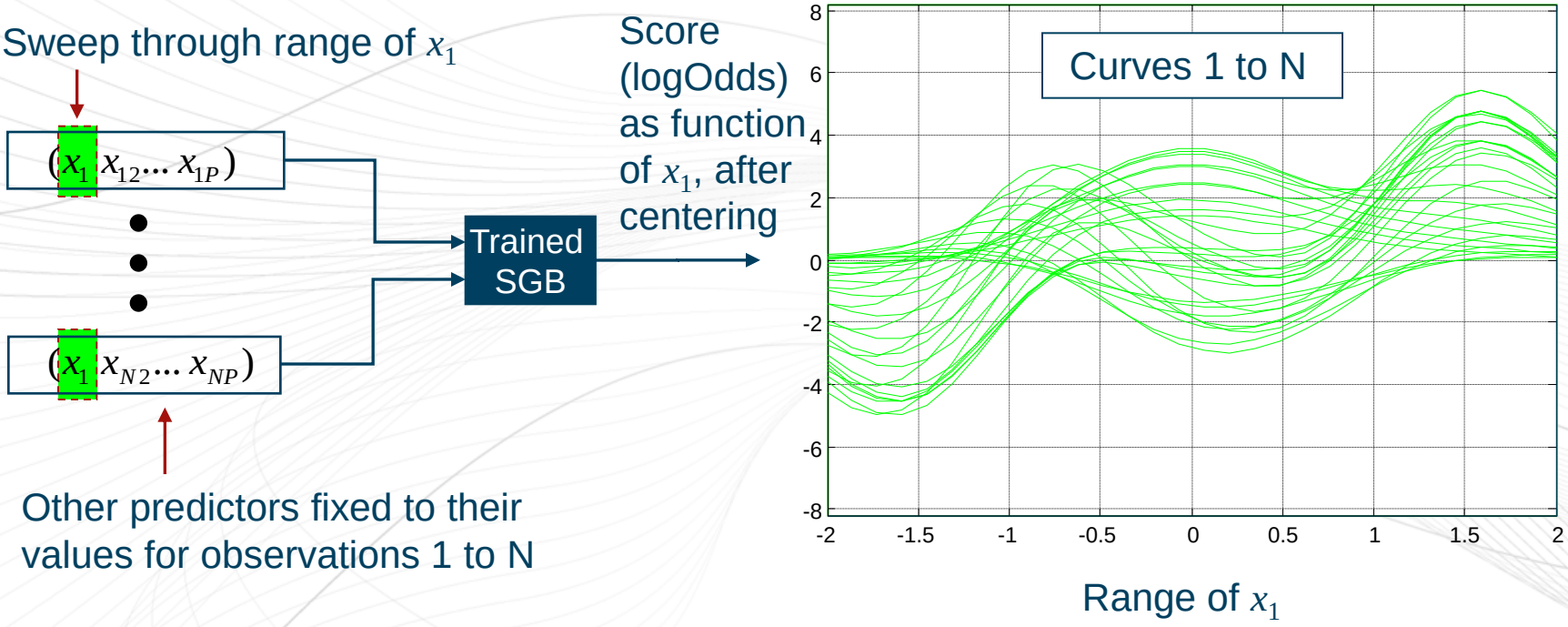
Diagnosing Predictive Relationships Learned by Complex Models



Score (logOdds) as function of x_1, after centering

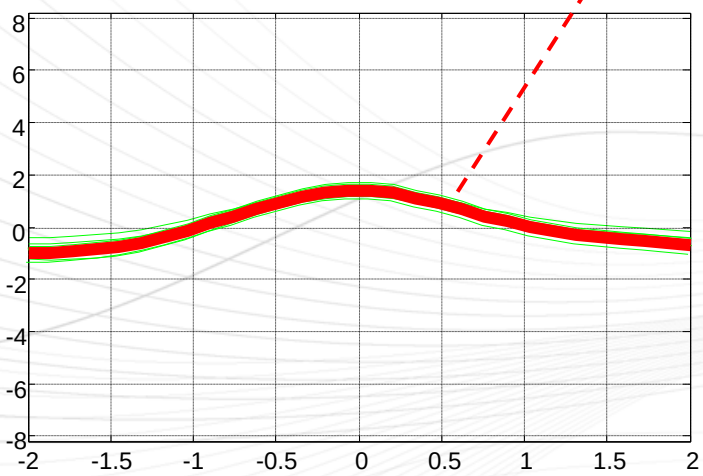


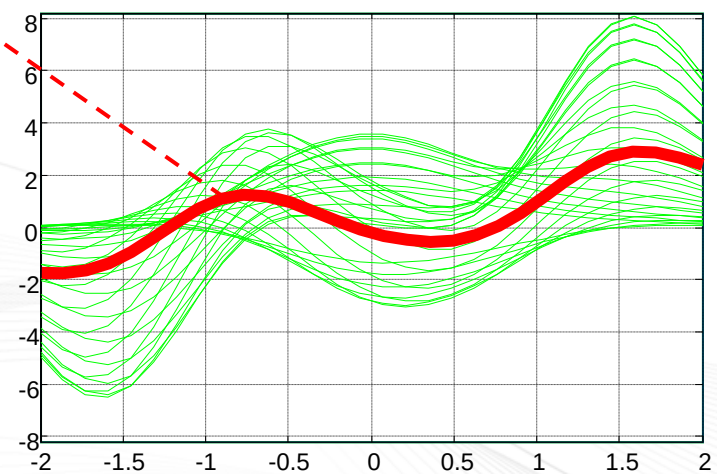
Conditioning on Observations 1 to N



Partial Dependence Plot

Defined in [4] as sample average over the N curves





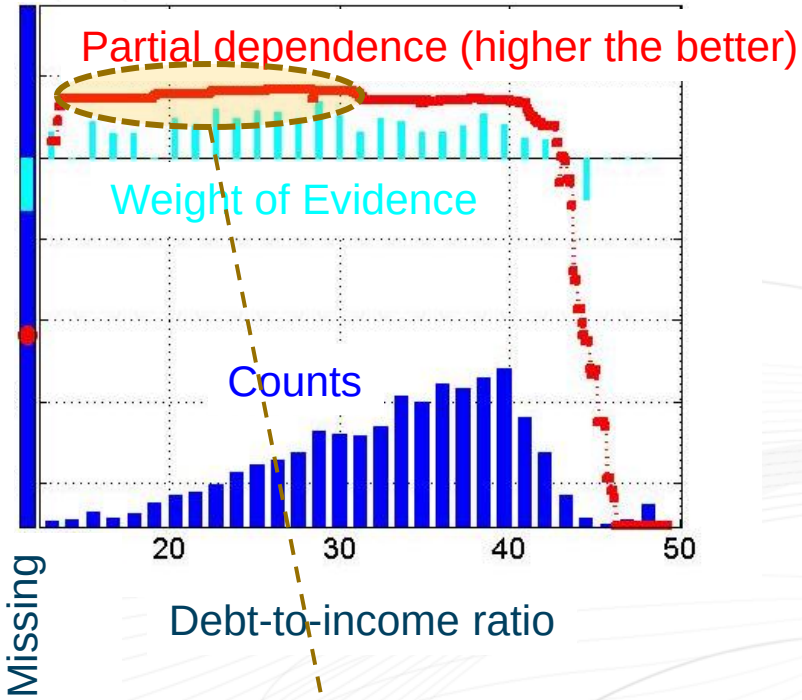
When x_1 enters the model *additively*…

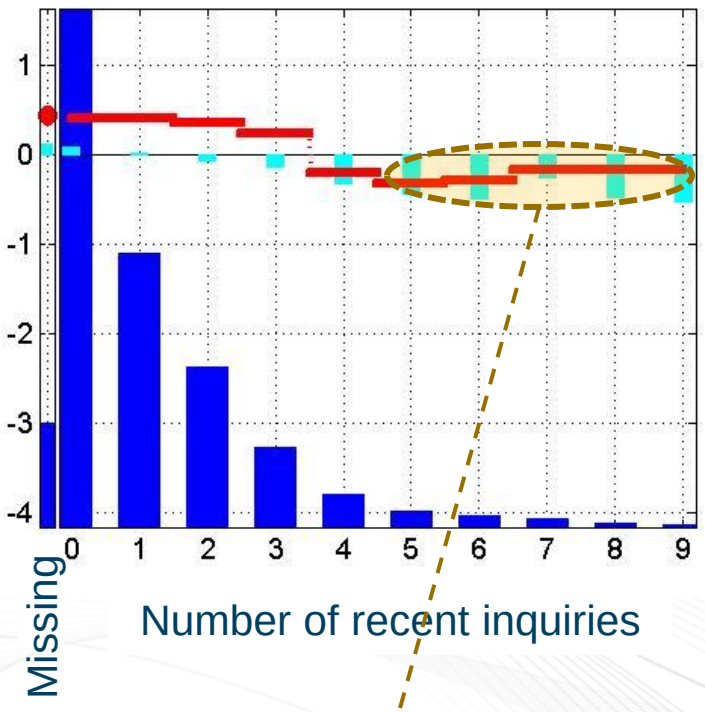
… then this plot summarizes influence of x_1 on score well

When x_1 participates in significant *interactions*…

… then explore relationship further by creating two-dimensional partial dependence plots

One-dimensional Partial Dependence Plots
Examples from Case Study



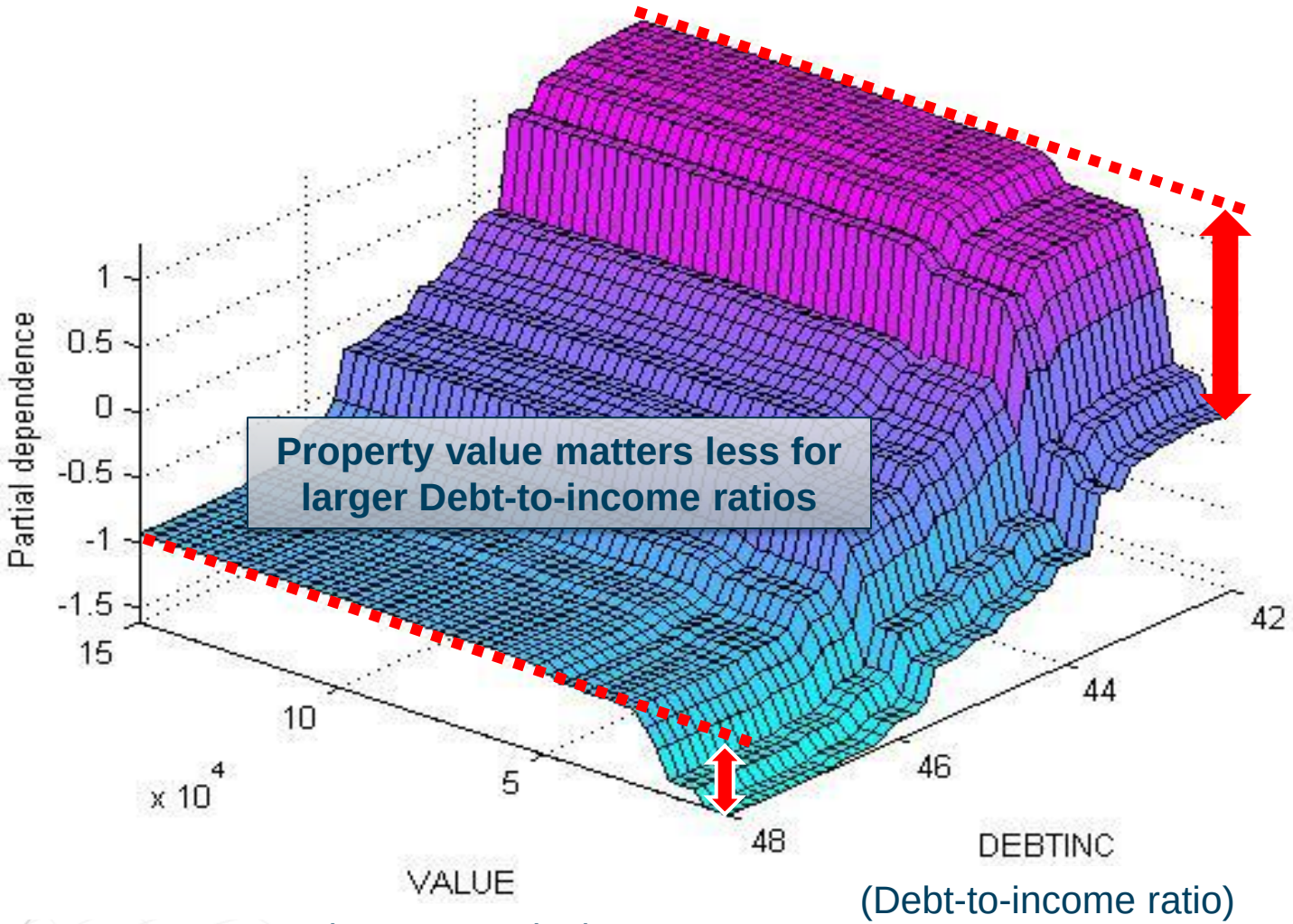


Concerned about increasing score?

Concerned about increasing score?

Two-dimensional Partial Dependence Plot

Interaction between Debt-to-income ratio and Property value



Pros and Cons of Algorithmic Learning

Pros
Objective, no strong assumptions
Accurate fit to historic data
Push-button procedures (close to)
Informative diagnostics, considerable insights

Cons
Predictive relationships may not be palatable
No obvious way to impose domain expertise
May not be deployable

Agenda

- Algorithmic Learning Illustration
- Case Study - Modeling US Home Equity Data
  - Diagnosing Complex Models
  - Algorithmic Learning-Informed Construction of Palatable Scorecards
- Summary

Alternative Modeling Strategies to Fit a Scorecard

Comparing two fitting objectives

1. Approximate log(Odds) of being "Good"
   - Penalized Bernoulli Likelihood
2. Approximate ensemble score from Stochastic Gradient Boosting
   - Penalized Least Squares

Structure of score formula

Score = Sum of staircase functions in 12 binned predictors

Categorical and missing values receive their own bins

Continuous predictors are binned into approximately equal-size bins

Constraints on score formula

Apply linear inequality $('\geq')$ constraints between score weight of neighboring bins covering these intervals:

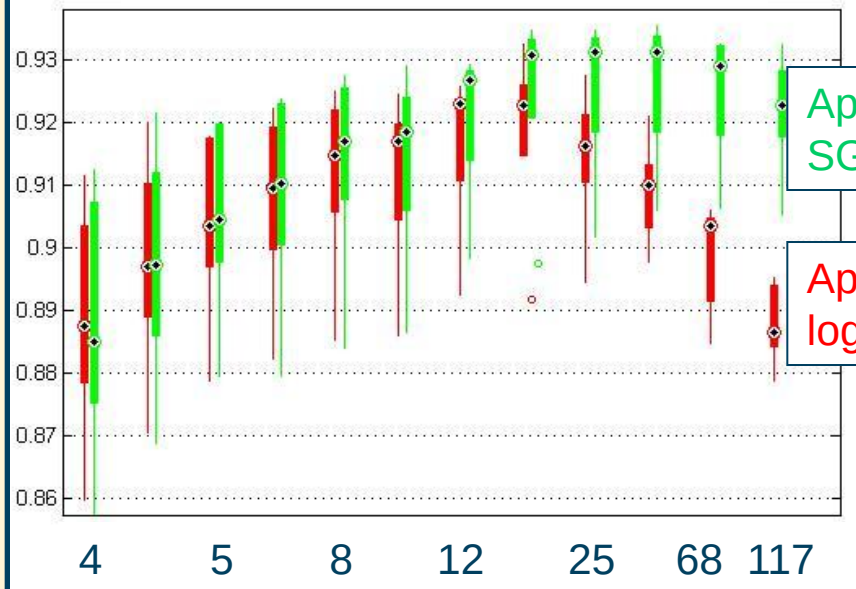
Debt ratio in [5%, inf)

Number of inquiries in [0, inf)

to force monotonic decreasing patterns

Constraining staircase functions is an effective approach to avoid counterintuitive score weight patterns

5-fold cross-validation of AUC



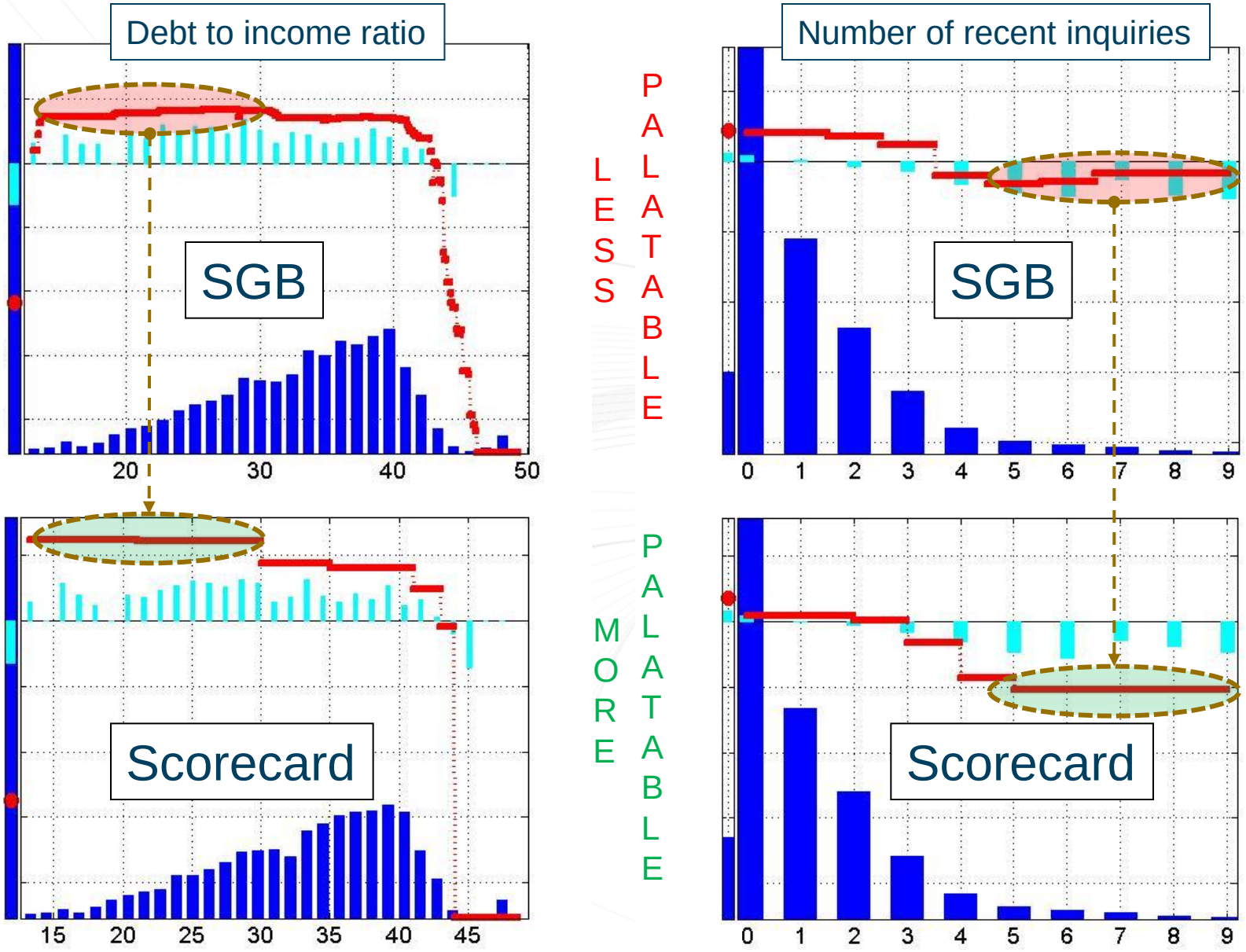
Conjecture About Statistical Benefit of Approximating Ensemble Score

- Ensemble score can be seen as well smoothed version of original dependent variable, carrying less noise than original target and also approximately unbiased
- We conjecture that variance of the estimates is reduced by predicting ensemble score instead of predicting the binary target. As a result, score performance is improved

Note:

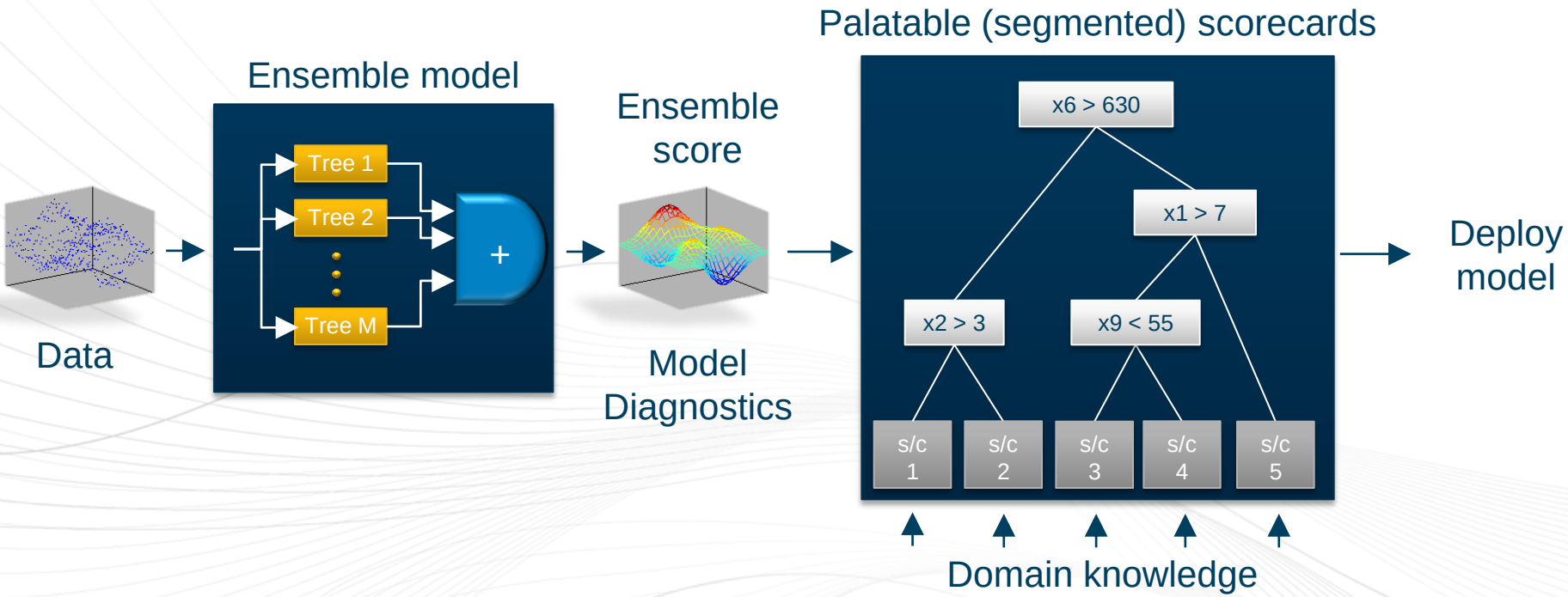
- Use of penalized MLE to approximate original target also reduces variance of estimates. But empirically we find this approach to be less effective

Constrained Scorecards Address Palatability Issues



Capturing Interactions With Segmented Scorecards

2-stage Approach for Growing Palatable Scorecard Trees



Stage I

- » Use algorithmic learning to generate ensemble score
- » Generate model diagnostics

Stage II

- » Recursively grow segmented scorecard tree to approximate ensemble score
- » Guide split decisions by interaction diagnostics
- » May impose domain knowledge via constraints on score formula

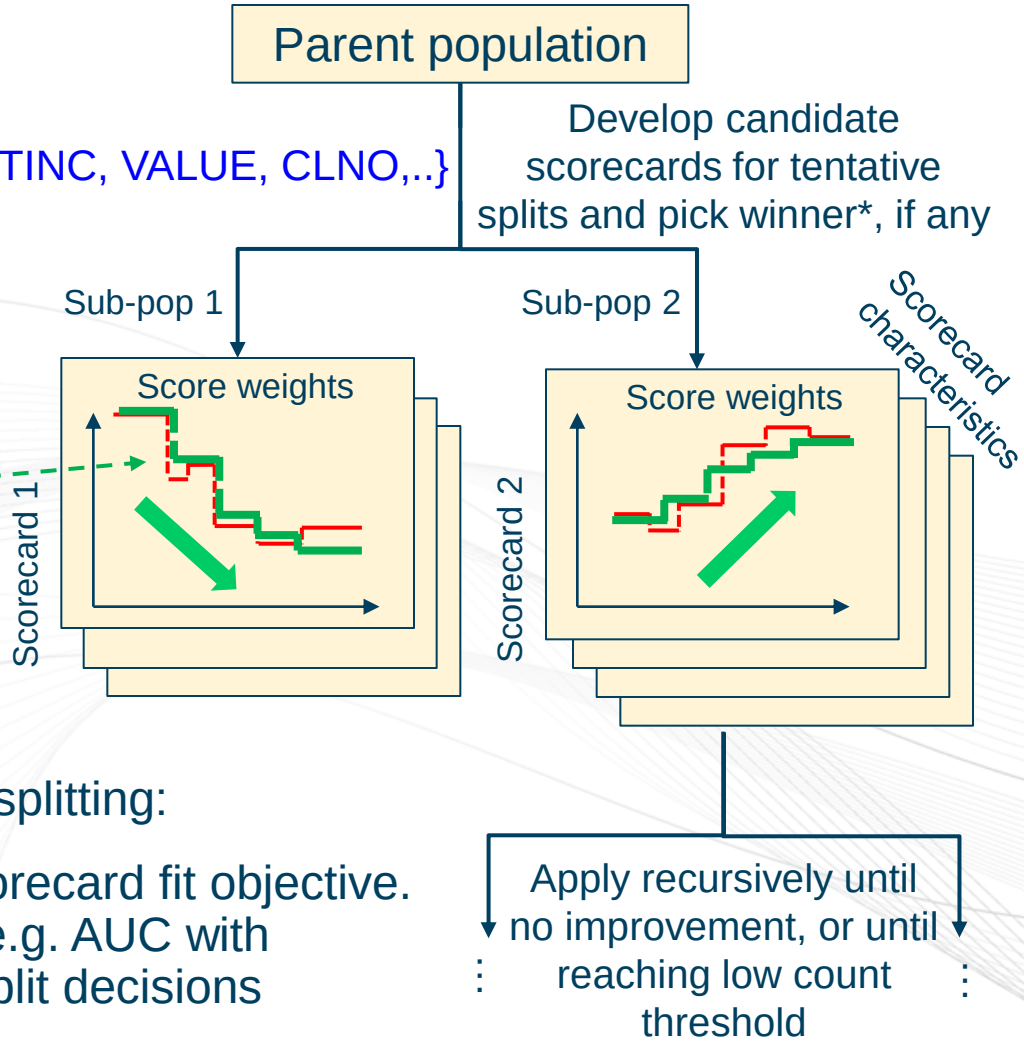
CART-like Greedy Recursive Scorecard Segmentation
Informed by Algorithmic Learning and Domain Knowledge

FICO

Interaction test statistics provides list of segmentation candidate (domain knowledge can override)

→ {DEBTINC, VALUE, CLNO,..}

Fit (constrained) scorecards as palatable approximations of ensemble score

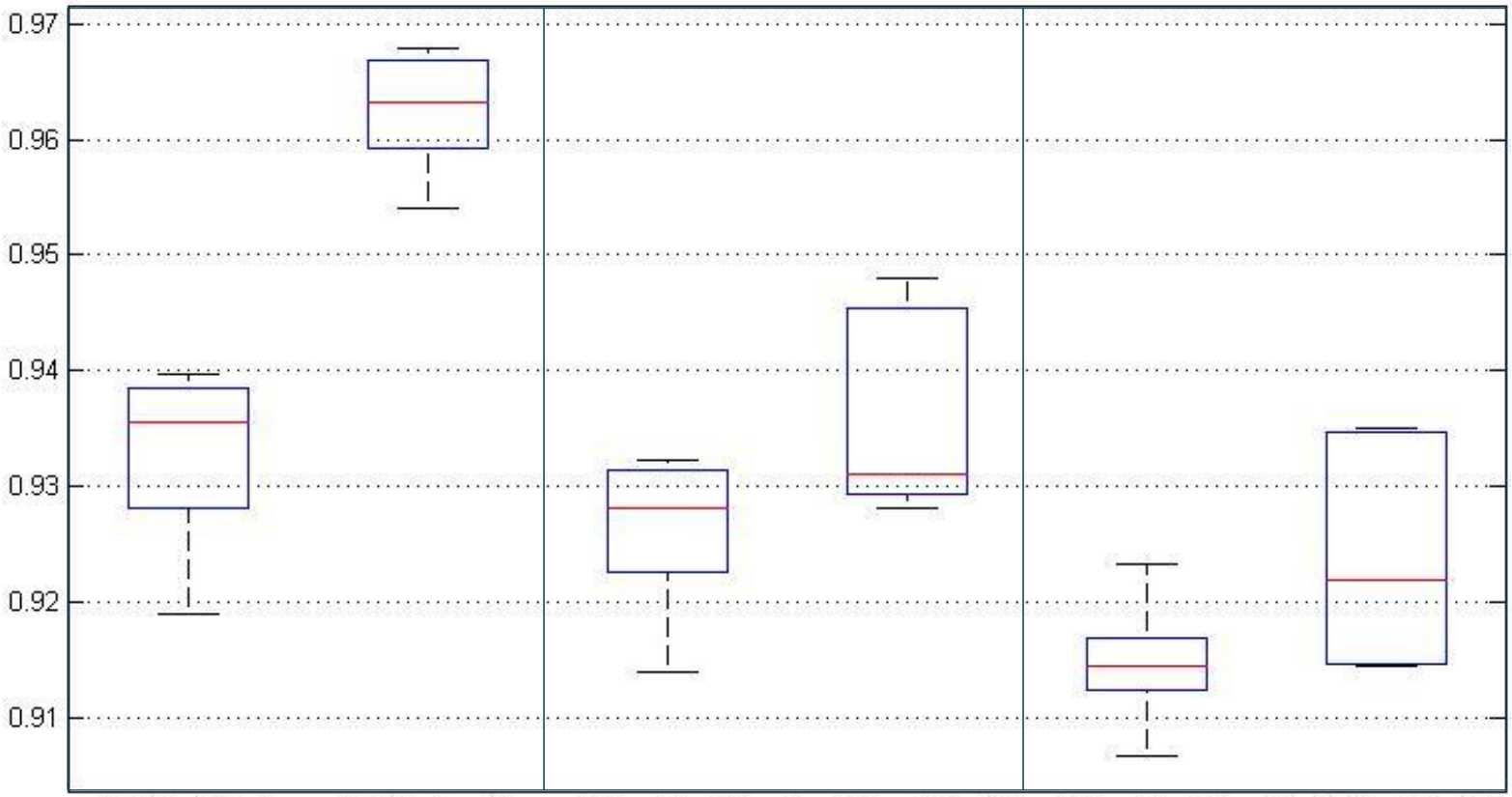


*Need objective function to decide on splitting:

- Least Squares is consistent with scorecard fit objective. But can use other favorite metrics (e.g. AUC with respect to original target) to make split decisions
- Best practice to evaluate splitting objective on a validation set (different from test set)

All Modeling Strategies Compared
Cross-validated Alternative Approaches on Same Folds

5-fold cross-validation of AUC

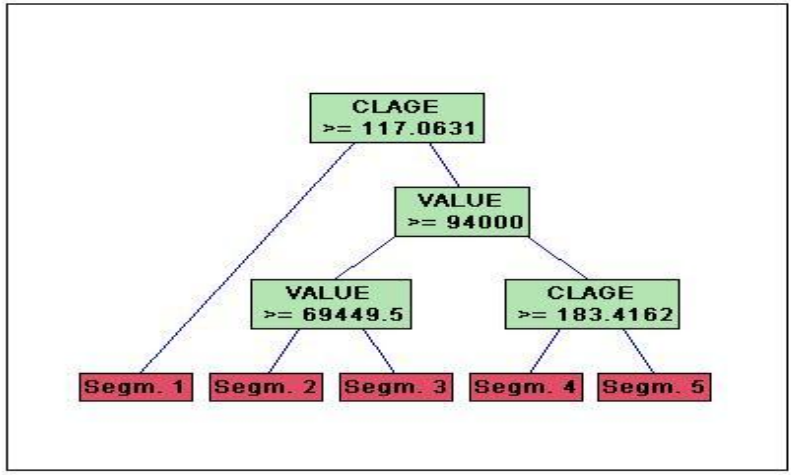


Results With Segmented Scorecards

Case study experiences

- Split decisions are noisy, but more stable when approximating SGB score instead of log(Odds)
- Various tree alternatives come up during cross-validation - finding ***the*** best tree is an illusion
- 2-stage approach effective at producing segmented scorecards with performance close to SGB, while a gap remains

Example result from 2-stage approach



Academic research on scorecard segmentation

- Segmentation sometimes but not always helps [5], [6]

Our experiences with big data sets

- Applied 2-stage approach to credit scoring and fraud problems with several million observations and 100+ predictors. Often find segmentations outperforming single scorecards by a few % in AUC, KS
- Restricting segmentation candidates based on SGB interaction diagnostics reduces hypothesis space and accelerates tree search
- Automated segmentations tend to match intuition (clean/delinquent, young/old, …) and shave weeks off tedious manual segmentation analysis
- Segmented scorecards at times achieve close to SGB performance, at other times there remains a gap

Summary

» Construction of accurate, deployable credit scoring models faces a unique challenge:

Find a highly predictive model within a flexible yet palatable model family

» Constrained (segmented) scorecards can meet this challenge

» Proposed 2-stage approach combines algorithmic learning and domain knowledge to inform the search for highly predictive, deployable (segmented) scorecards

» We are testing it for projects across credit scoring, application and insurance claims fraud and see good potential to increase the effectiveness of practical scoring models

THANK YOU

References

[1] *Random Forests*, by Leo Breiman, Machine Learning, Volume 45, Issue 1 (Oct., 2001), pp. 5-32.

[2] *Stochastic Gradient Boosting*, by Jerome Friedman, (March 1999).

[3] *Greedy Function Approximation: A Gradient Boosting Machine*, by Jerome Friedman, The Annals of Statistics, Volume 29, Number 5 (2001), pp. 1189-1232.

[4] *Predictive Learning Via Rule Ensembles*, by Jerome Friedman and Bogdan Popescu, Ann. Appl. Stat. Volume 2, Number 3 (2008), pp. 916-954.

[5] *Does Segmentation Always Improve Model Performance in Credit Scoring?,* by Katarzyna Bijak and Lyn Thomas, Expert Systems with Applications, Volume 39, Issue 3 (Feb. 2012), pp. 2433-2442.

[6] *Optimal Bipartite Scorecards*, by David Hand et. al., Expert Systems with Applications, Volume 29, Issue 3 (Oct. 2005), pp. 684-690.