Default Models and "Small" Data Sets

Leveraging Multi-tree Methods for Reliable Scorecards

Dan Steinberg, Nicholas Scott Cardell, Mikhail Golovnya

September, 2005

SALFORD SYSTEMS

Small Data Problems at Financial Institutions

- Financial institutions are generally thought to have massive volumes of data permitting ultra-refined and precise statistical models
  - American Express reports more than 50 million customers worldwide
  - Moody's notes that a mid-sized European bank lends 4 billion Euros annually
  - Foster and Stine's bankruptcy study is based on 600,000 credit card accounts
- Not every credit risk model can be based on such large files
  - Corporate lending may be extended to only a few thousand accounts
  - Many smaller banks still in existence
  - Models of commercial bank failures and country defaults are necessarily based on smaller samples
- Problem compounded when data is frequently missing

What is a Small Sample? The Sample Rule of Four

- To simplify the discussion we focus on the two class binary outcome (0/1)
- A sample is small when the number of observations of the smaller outcome class is small
  - In our SME sample there were only 250 defaults
  - In a fraud model our client located 60 examples of documented fraud
  - Even in a recent large scale bankruptcy study there were 2,244 bankrupts
- We offer as a rule of thumb this definition of "effective sample size" for highly unbalanced sample

$$ESS = 4 \times \text{smaller class}$$

- Regardless of the sample size of the larger (less interesting) class
- Thus, in our definition, a file with 2,500 defaults has an effective sample size of 10,000 regardless of the actual number of "goods"

Effective Sample Size

- Logic is straightforward: Consider a data file with 250 defaults and 1,500 goods. How much benefit would we expect to derive from say 100 additional good accounts?
- Our rule of thumb suggests no benefit at all (which overstates the case)
- But going from 250 defaults to 350 is obviously a major benefit
- The table on the right is sobering in emphasizing that the group that matters should dominate our thinking about sample size
- Casual simulation experiments suggest that beyond 10:1 the value of an additional GOOD is literally 0.
- But certainly up through ratios of 4:1 there is clear value of additional GOODs

GOODS	BADS	ESS
10,000	60	240
1,500	250	1,000
7,000	500	2,000
600,000	2,500	10,000

Why the factor 4?

- The Sample Rule of Four is based on the following heuristic argument::
  - Suppose that X is in fact a relevant predictor of Y (1=default) and we wish to measure the precise difference in the mean of X between GOODS and BADS
  - Assuming that Var(X|Y)=Var(X)

$$Var(\underline{X}_1 - \underline{X}_0) = Var(X) * \{ (1/N_0) + (1/N_1))$$

if $N_0 = N_1 = .5*N$ sample evenly divided between 0's and 1's

$$Var(\underline{X}_1 - \underline{X}_0) = Var(X) * (4/N)$$

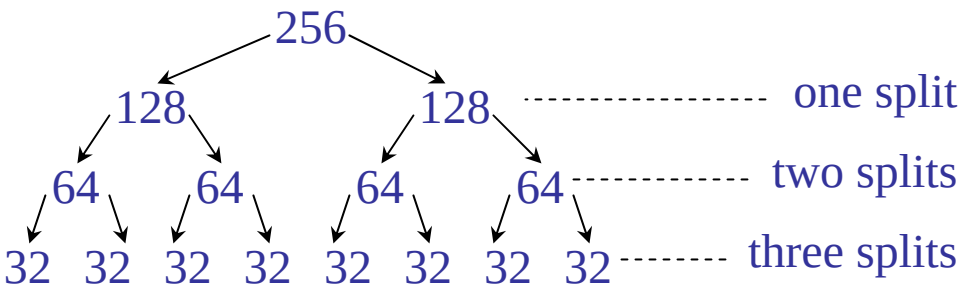
  - Thus, if N_0 is substantially larger than N_1 then

$$Var(\underline{X}_1 - \underline{X}_0) \approx Var(X) * (1/ N_1))$$

  - Thus with many 0's and few 1's the variance of the estimator is about the same as that for a balanced sample of size $4* N_1$

Reminder of the implications of small samples

- In a decision tree based on median splitters the BAD sample size in the nodes goes



- If every split involves a different variable we still only have 6 variables in the model
- If the model is tested via 10-fold cross-validation each test fold is based on just 26 BADS.
- Even if all BADS are concentrated into as few as 3 nodes we are drawing conclusions based on 8 BADS or fewer in a node during CV testing

Small Sample Modeling Challenge

- Until recently the best methods available for small samples have been classical statistical models such as logistic regression based on a handful of predictors
- Fully parametric models impose substantial a priori structure on the model and are typically highly biased (model assumptions are incorrect)
- But parametric models offer high stability and low variance. Often the low variance is sufficient to offset the model bias to deliver a low MSE
  - Biostatistics literature is replete with studies of N=30 or 60 or 200
  - All make use of regression and logistic regression
- Problem becomes far more difficult when many predictors have missing values, and complete records may represent less than 50% of data

Modern Methods

- Modelers have been eager to benefit from advances in machine learning and data mining
  - Offer low bias models faithful to the data
  - Capture substantial nonlinearities, interactions
  - Strong ability to select predictors from large number of candidates
- For example, decision trees such as CART® have effective methods of dealing with missing values without requiring imputation
- But small samples are still a challenge for a single decision tree
  - 3 or 4 splits can easily exhaust the data yielding a model based on “too few” factors
- Problem is now resolved through the introduction of multi-tree methods

Multi-tree methods and their single tree ancestors

- Best known trees include CART (Breiman, Friedman, Olshen and Stone, 1984) and C4.5 (Quinlan, 1995)
- Multi-tree methods have been under development since the early 1990s. Most important variants (and dates of published articles) are:
  - Bagger (Breiman, 1996, "**B**ootstrap **Ag**gregation")
  - Boosting (Freund and Schapire, 1995)
  - **M**ultiple **A**dditive **R**egression **T**rees (Friedman, 1999, aka MART™ or TreeNet™)
  - RandomForests™ (Breiman, 2001)

Multi-tree Methods: Simplest Case

- Simplest example:
  - Grow a tree on training data
  - Find a way to grow another different tree (change something in set up)
  - Repeat many times, eg 500 replications
  - Average results or create voting scheme. Relate PD to fraction of trees predicting default for a given case

Prediction

- Beauty of the method is that every new tree starts with a ***complete*** set of data.
- Any one tree can run out of data, but when that happens we just start again with a new tree and ***all*** the data.

Automated Multiple Tree Generation

- Earliest multi-model methods recommended taking several good candidates and averaging them. Examples considered as few as 3 trees.
- Too difficult to generate multiple models manually
- How do we generate different trees?
  - Bagger: random re-weighting of the data via bootstrap resampling
    - Reweight at random and regrow. Every repetition independent of others
  - RandomForests: Random splits. Tree itself is grown at least partly at random
  - Boosting: weighting based on prior success in correctly classifiying a case. High weights on difficult to classifiy cases
    - Reweighting depends on how succesfully a record was previously classified
  - TreeNet: Boosting with refinements. Each tree attempts to correct errors made by predecessors
    - Each tree is linked to predecessors. Like a series expansion where the addition of terms progressively improves the predictions

TreeNet (aka MART)

- We focus on TreeNet because
  - It is the method used in the real world studies we report here
  - We have found it to be more accurate than the other methods
    - Has placed first in at least two major data mining competitions
  - Building blocks are SMALL trees regardless of training sample size
    - In small samples we have no choice but Treenet ***prefers*** small trees
  - Very strong resistance to errors in the data including “mislabeled target”
  - Mislabeled target will occur
    - When GOOD/BAD analysis is conducted on relatively new accounts and thus many BADs appear to be GOOD
    - In fraud studies where not all fraud is properly identified. Some fraud confused with legitimate default

TreeNet Process

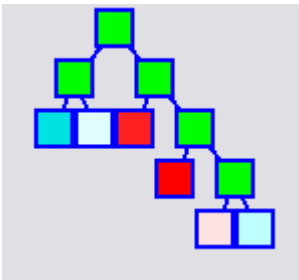
- Begin with a very small tree as initial model
  - Could be as small as ONE split generating 2 terminal nodes
  - Typical model will have 3-5 splits in a tree, generating 4-6 terminal nodes
  - Output is a probability (eg of default)
- Compute "residuals" for this simple model (prediction error) for every record in data
- Grow second small tree to predict the residual derived from first
- New model is

$$\text{Tree}_1 + \text{Tree}_2$$

- Compute residuals from this new 2-tree model and grow 3rd tree to predict these revised residuals

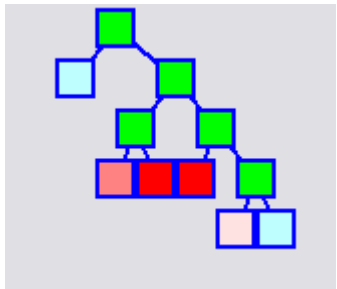
TreeNet: Trees incrementally revise predicted scores

Tree 1



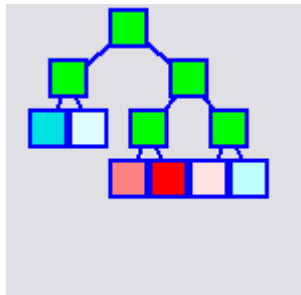
+

Tree 2



+

Tree 3



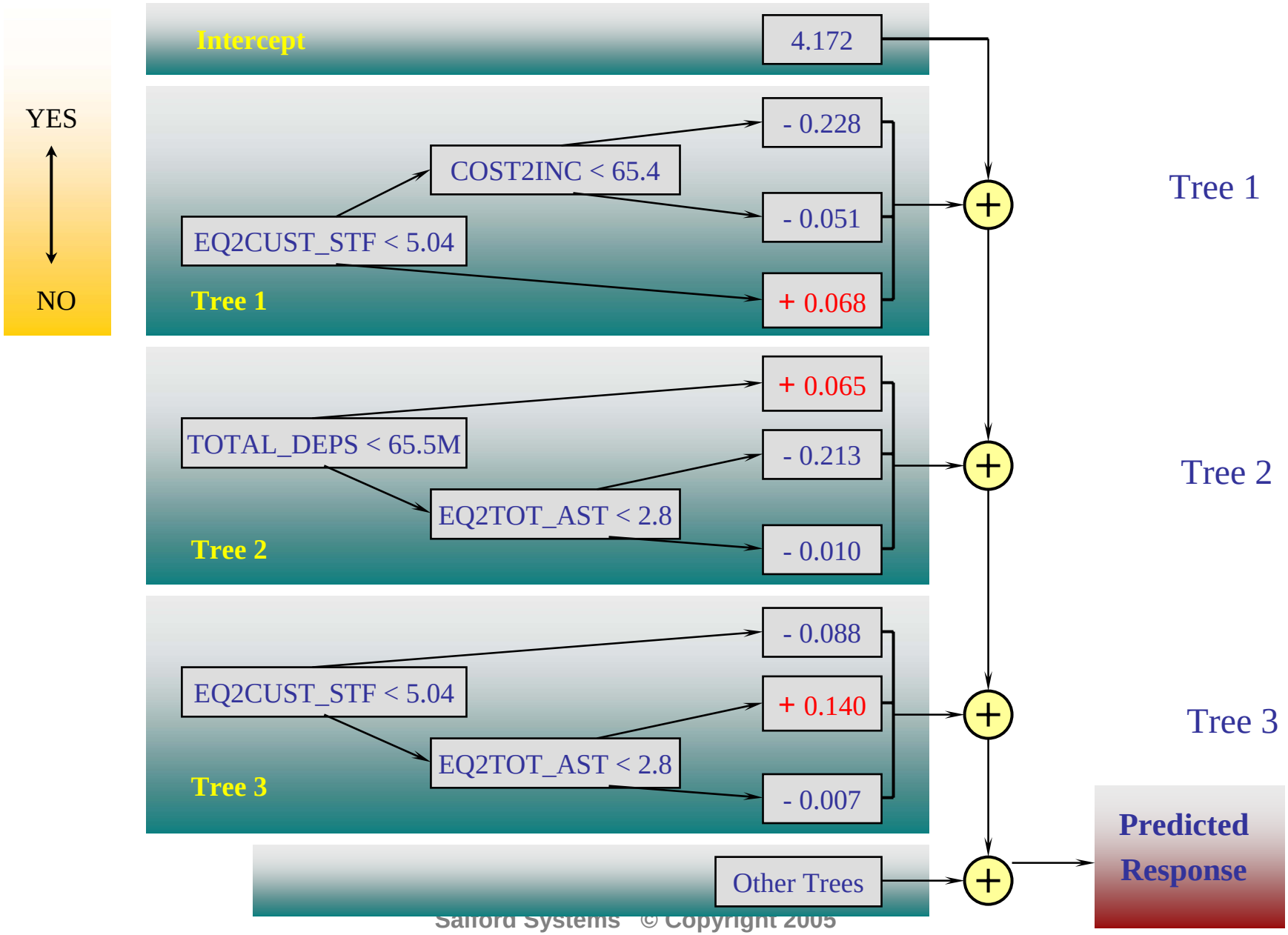
First tree grown on original target. Intentionally "weak" model

2nd tree grown on residuals from first. Predictions made to improve first tree

3rd tree grown on residuals from model consisting of first two trees

Every tree produces at least one *positive* and at least one *negative* node. Red reflects a relatively large positive and deep blue reflects a relatively negative node. Total "score" is obtained by finding relevant terminal node in every tree in model and summing across all trees

TreeNet: Sample Individual Trees



TreeNet methdology: Key points

- Trees are kept small
- Updates are small (downweighted). Like a partial adjustment model. Update factors can be as small as .01, .001, .0001 or even smaller. This means that the model prediction changes by very small amounts in each training cycle
- Use random subsets of the training data in each cycle. Never train on all the training data in any one cycle
- Highly problematic cases are IGNORED. If model prediction starts to diverge substantially from observed data, that data will not be used in further updates
- Cross-validation used for self-test in small data sets
- Model can be tuned to optimize
  - Area under the ROC curve
  - Logistic likelihood (deviance)
  - Classification Accuracy
  - Lift achieved in a specified percentile of the predicted-probability ranked data

Why does TreeNet work?

- Slow learning: the method “peels the onion” extracting very small amounts of information in any one learning cycle
- TreeNet can leverage hints available in the data across a large number of predictors, making use of many of them
- TreeNet self-protects against errors in the dependent variable (vital for fraud studies). If a record is actually a “1” but is misrecorded in the data as a “0” and TreeNet recognizes it as a 1 it will not attempt to get this record correct.
- Can capture substantial nonlinearity and complex interactions

Multiple Additive Regression Trees

- Friedman originally named his methodology MART™ as the method generates small trees which are summed to obtain an overall score
- The model can be thought of as a series expansion approximating the true functional relationship

$$F(X) = F_0 + \beta_1 T_1(X) + \beta_2 T_2(X) + \ldots + \beta_M T_M(X)$$

- We can think of each small tree as a mini-scorecard, making use of possibly different combinations of variables. Each mini-scorecard is designed to offer a slight improvement intended to correct and refine its predecessors
- Because each tree starts at the “root node” and can use all of the available data a TreeNet model can never run out of data no matter how many trees are built.

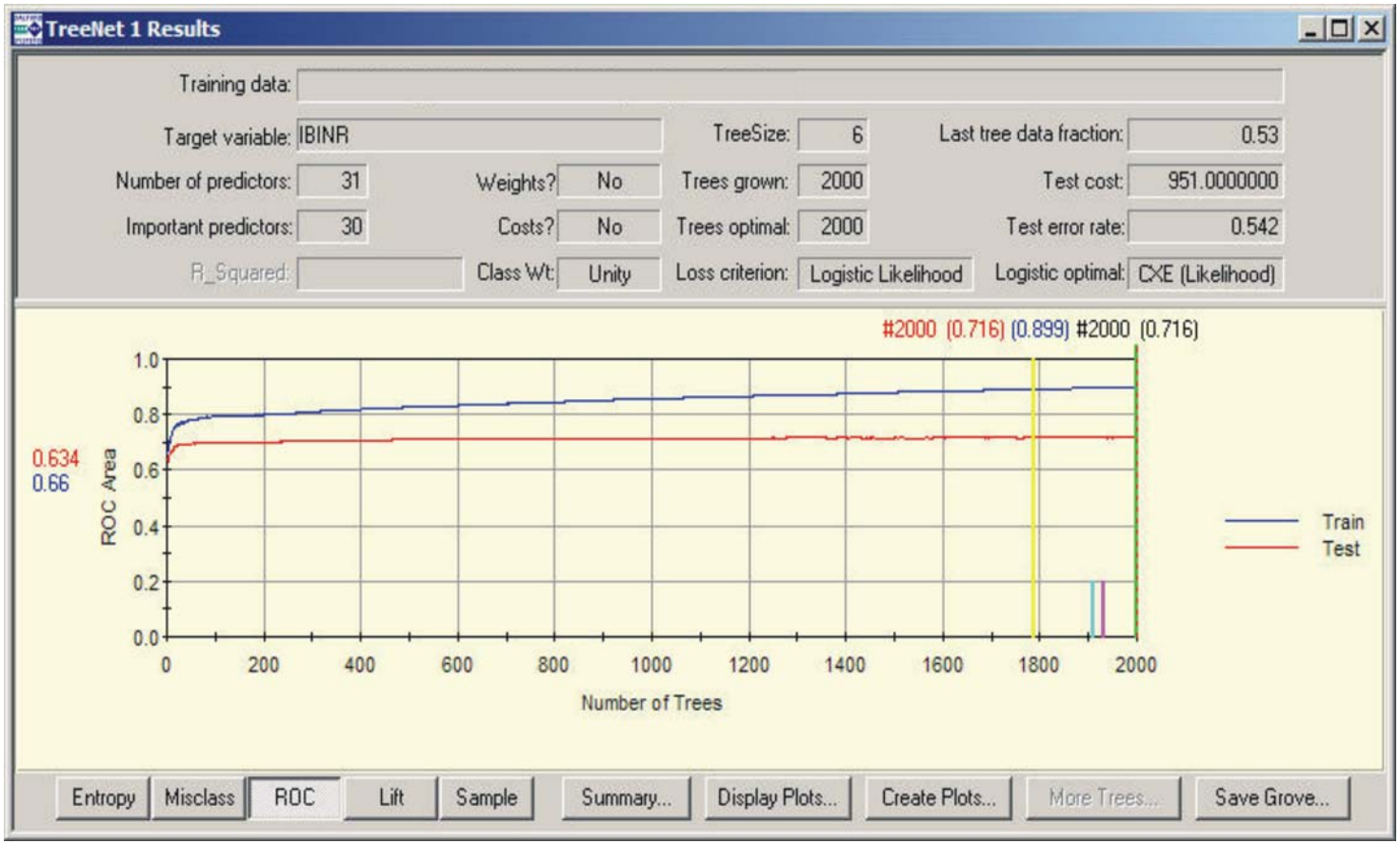
Examples

- Corporate default
- Analysis of bank ratings

Corporate Default Scorecard

- Mid-sized bank
  - Around 200 defaults
  - Around 200 indeterminate/slow
- Classic small sample problem
- Standard financial statements available, variety of ratios created
- All key variables had some fraction of missing data, from 2% to 6% in a set of 10 core predictors, and up to 35% missing in a set of 20 core predictors
- Single CART tree involves just 6 predictors
  - yields cross-validated ROC= 0.6523

TreeNet Analysis of Corporate Default Data: Summary Display



Summary reports progress of model as it evolves with an increasing number of trees. Markers indicate models optimizing entropy, misclassification rate, ROC

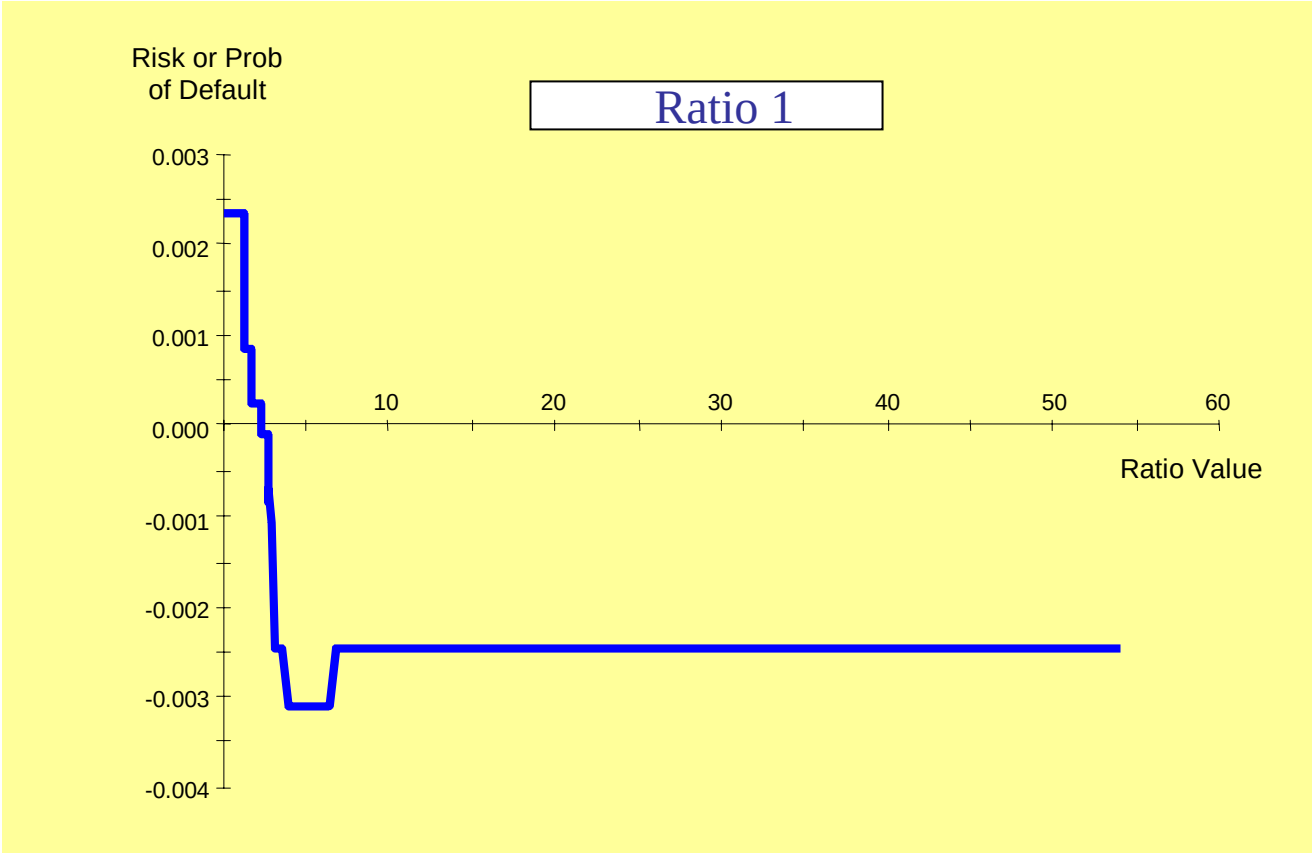
Corporate Default Model: TreeNet

- 1789 Trees in model with best test sample area under ROC curve

	CXE	Class Error	ROC Area	Lift
Optimal N Trees:	2000	1930	1789	1911
Optimal Criterion	0.534914	0.3485293	0.7164921	1.9231729

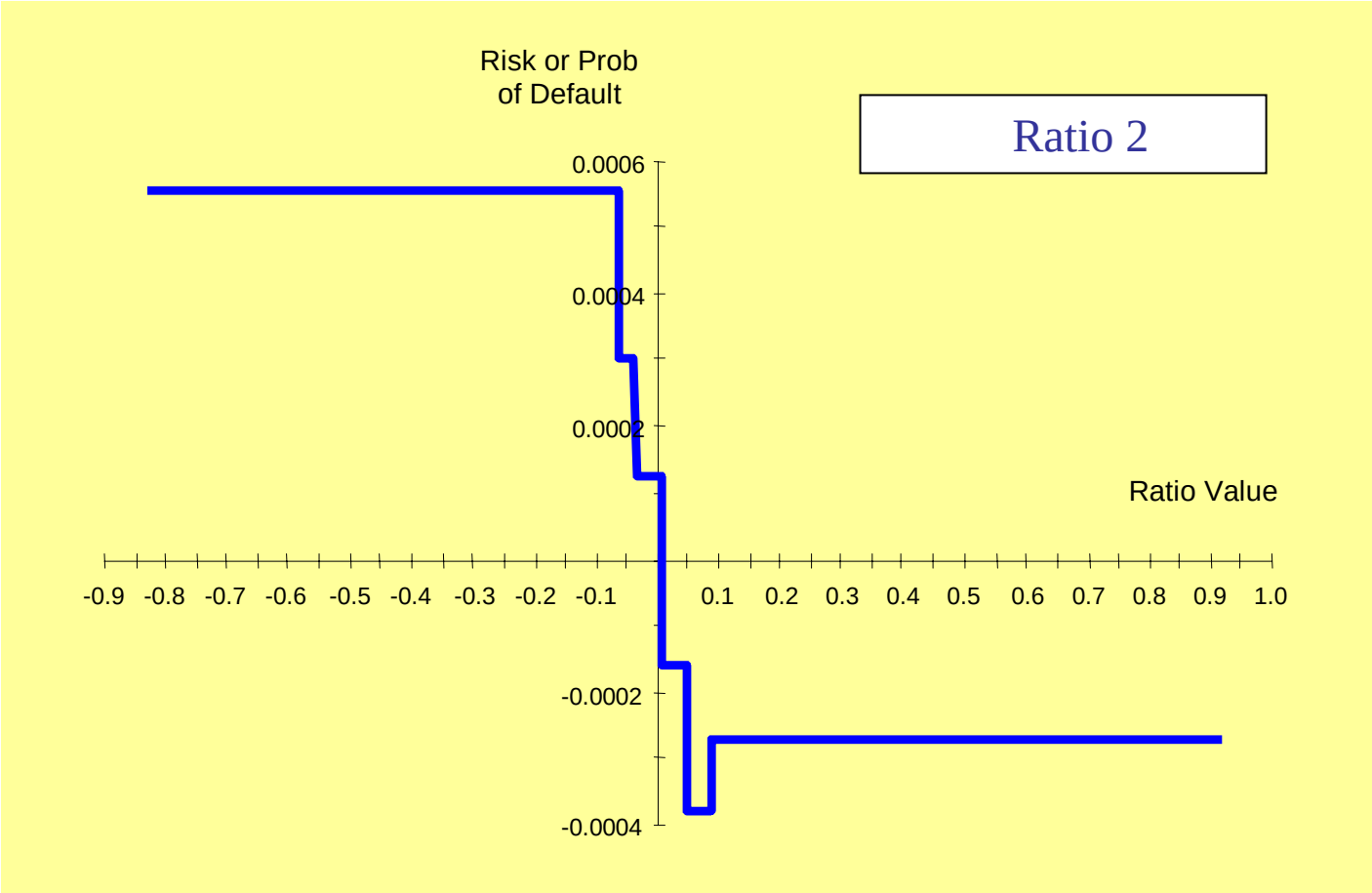
- TreeNet Cross-Validated ROC=0.71649 far better than single tree
- Able to make use of more variables due to the many trees
  - Single CART tree uses 6 predictors, TreeNet uses more than 20
- Some of these variables were missing in 35% or more of all accounts
- Can extract graphical displays to describe model

Predictive Variable – Financial Ratio 1



TreeNet can trace out impact of predictor on PD

Predictive Variable – Financial Ratio 2



Additive vs Interaction Model (2-node vs 6-node trees)

- 2 node trees model

	CXE	Class Error	ROC Area	Lift
Optimal N Trees:	2000	1812	1978	1763
Optimal Criterion	0.54538	0.39331	0.70749	1.84881

- 6 node trees model

Optimal N Trees:	2000	1930	1789	1911
Optimal Criterion	0.53491	0.34852	0.71649	1.92317

- Two node trees do not permit interactions of any kind as each term in the model involves a single predictor in isolation.2-node tree models are highly nonlinear but strictly additive
- Three- or more node trees allow progressively higher order interactions.
- Running model with different tree sizes allows simple discovery of the precise amount of interaction required for maximum performance models

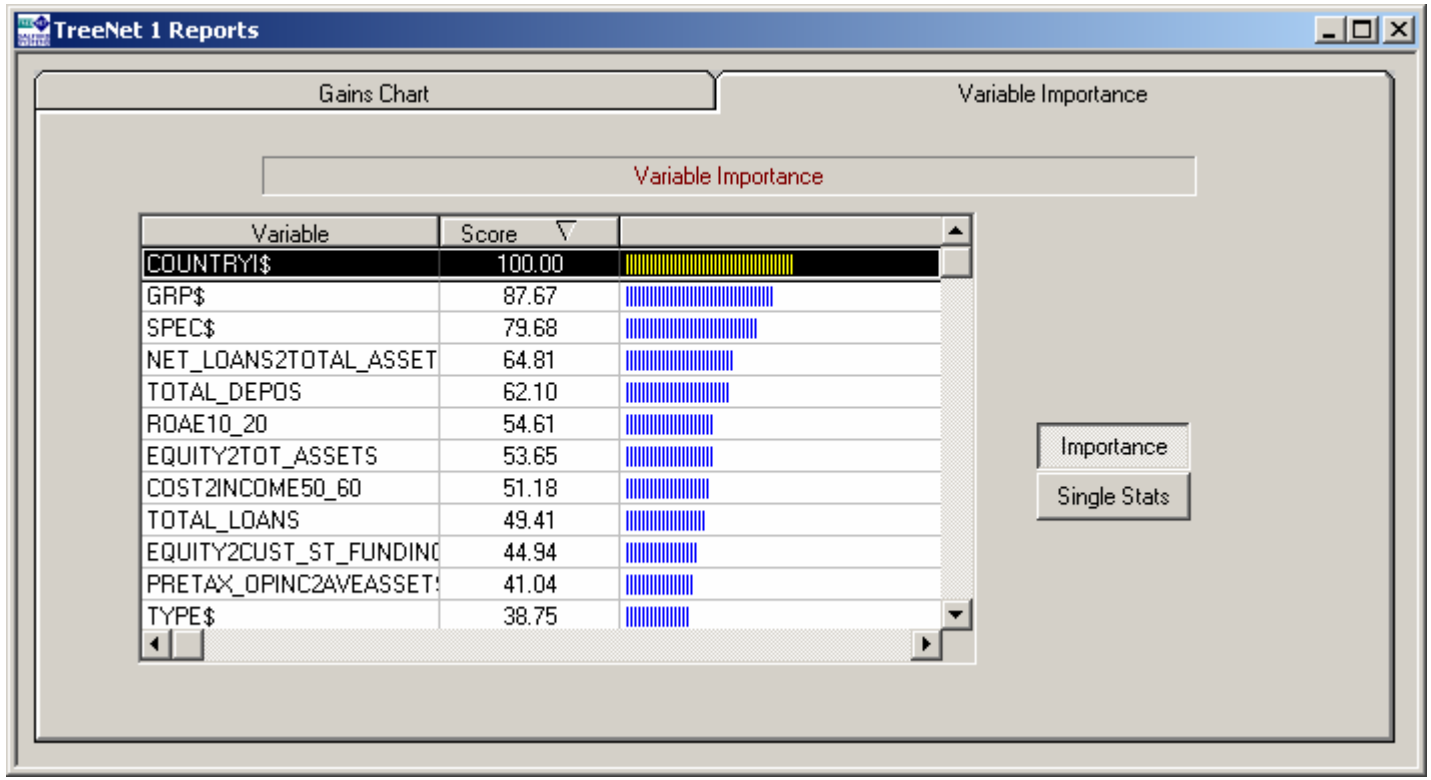
Bank Ratings: Regression Example

- Build a predictive model for average of major bank ratings
  - Scaled average of S&P. Moody's, Fitch Ratings
- Challenges include
  - Small data base (66 banks)
  - Missing values prevalent (up to 35 missing in any predictor)
  - Impossible to build linear regression model because of missings
  - Expect relationships to be nonlinear
- 66 banks
- 25 potential predictors

Cross-Validated Performance: Predicting Rating Score

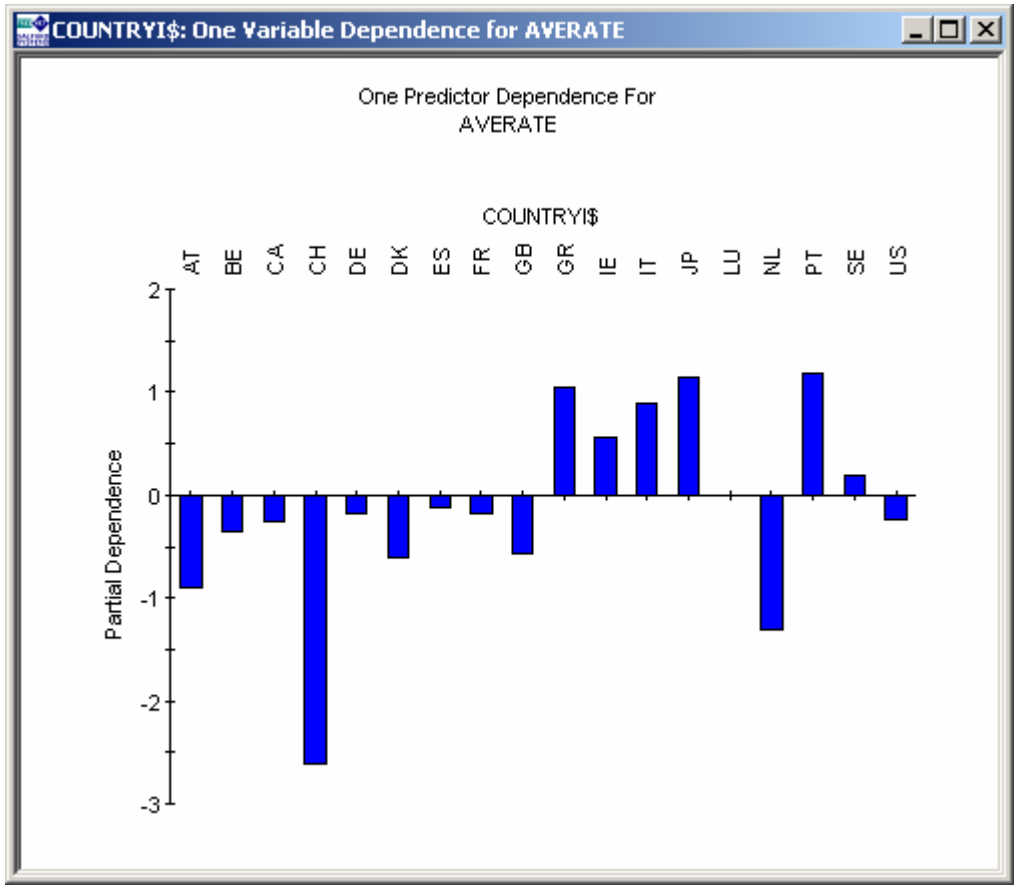
The optimal model achieved around 860 trees with cross-validated mean absolute error 0.87, target variable ranges from 1 to 10

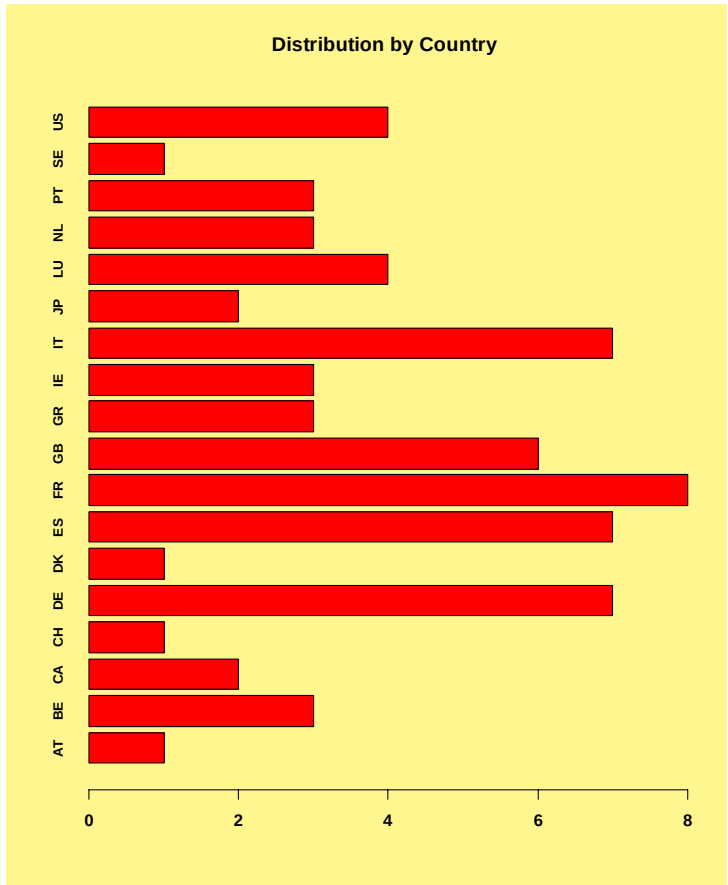
Variable Importance Ranking



Ranks variables according to their contributions to the overall variation in the target variable

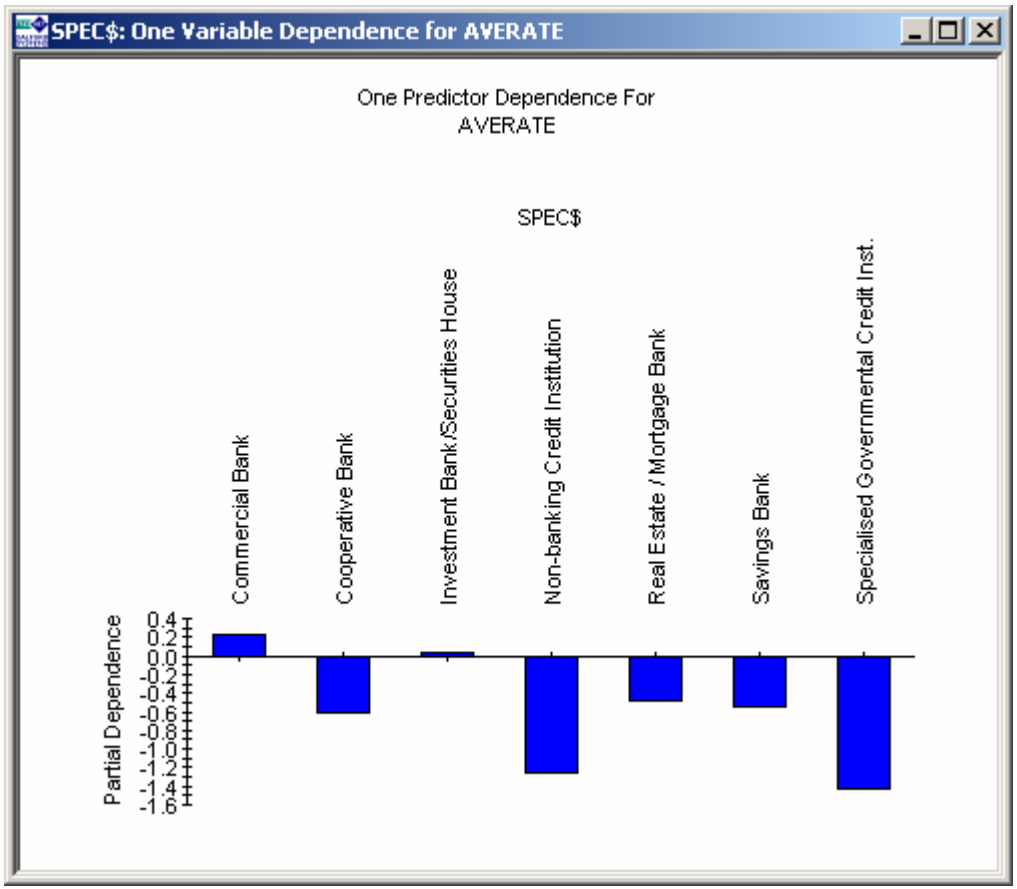
Country Contribution to Risk Score

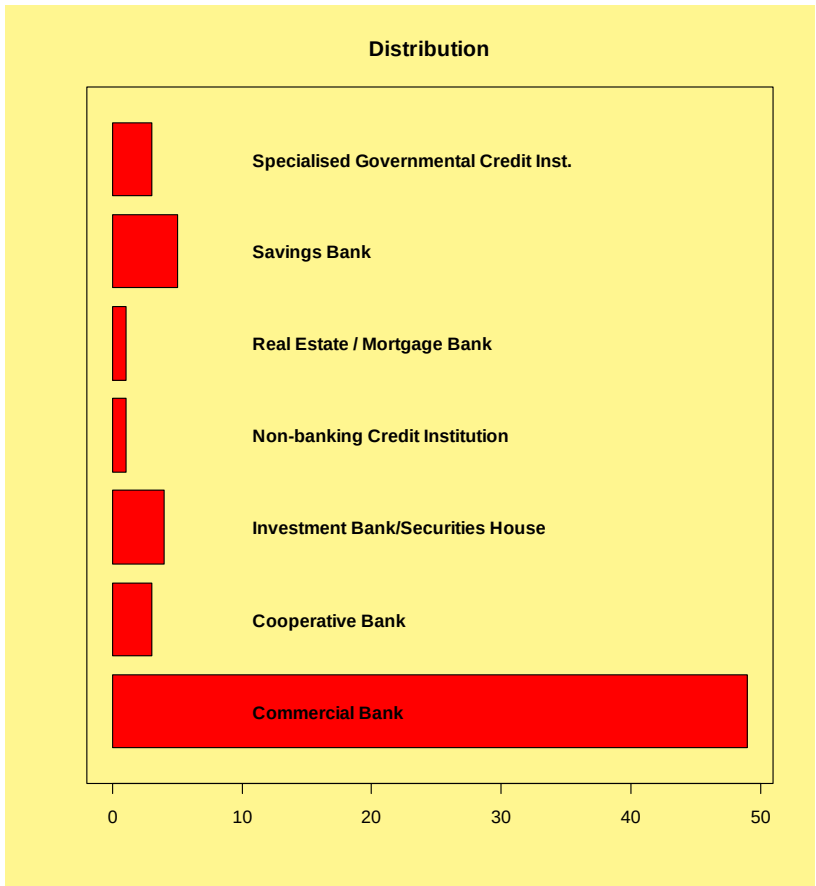




Swiss banks tend be rated low risk

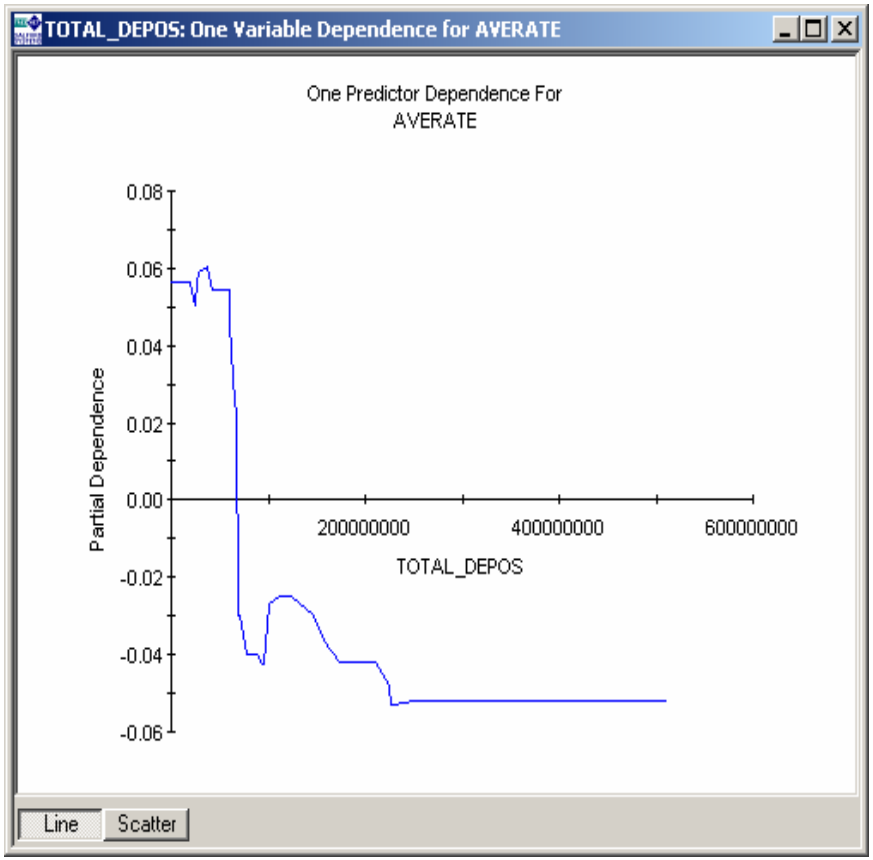
Bank Specialization

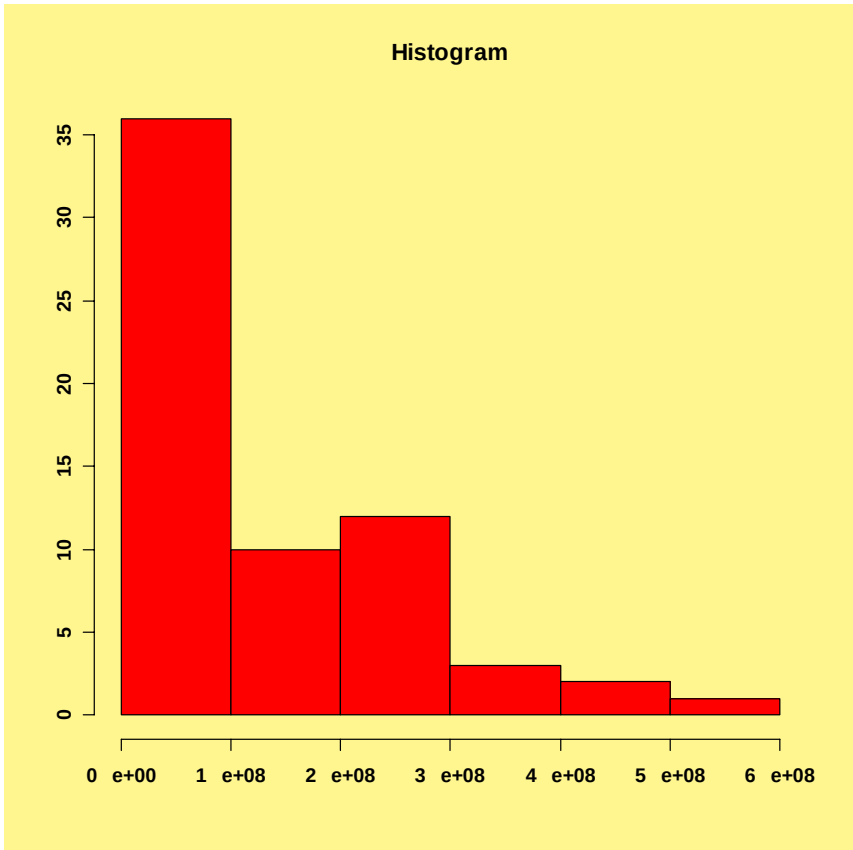




- Commercial banks and investment banks are rated higher risk

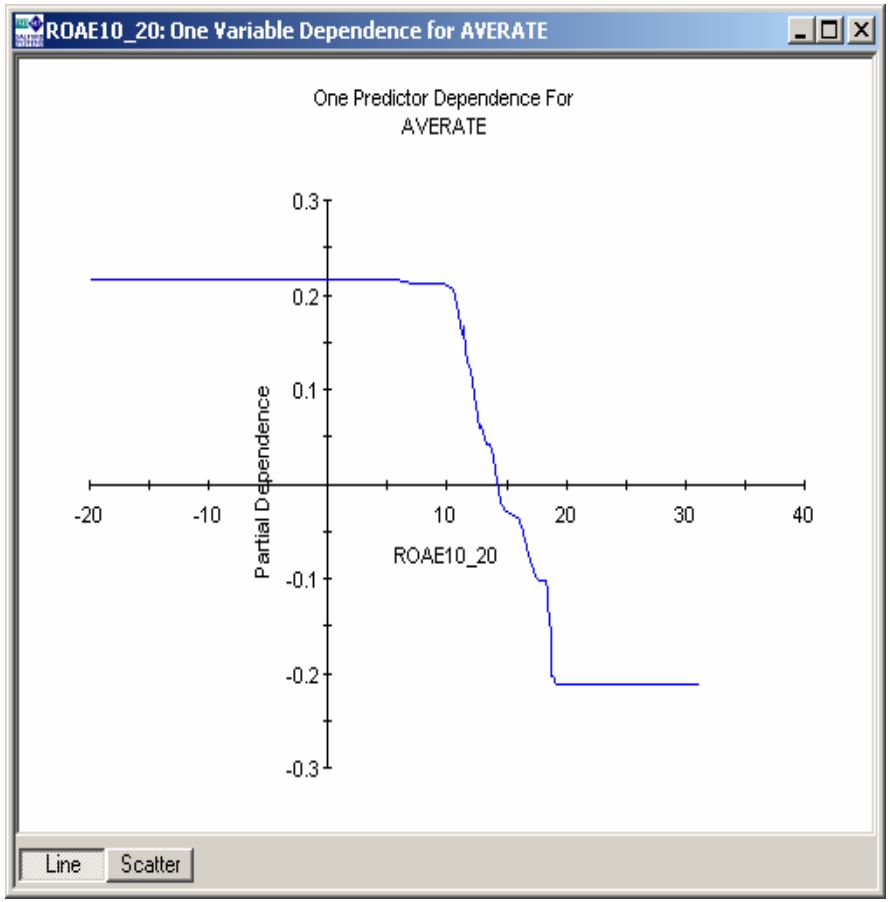
Scale: Total Deposits

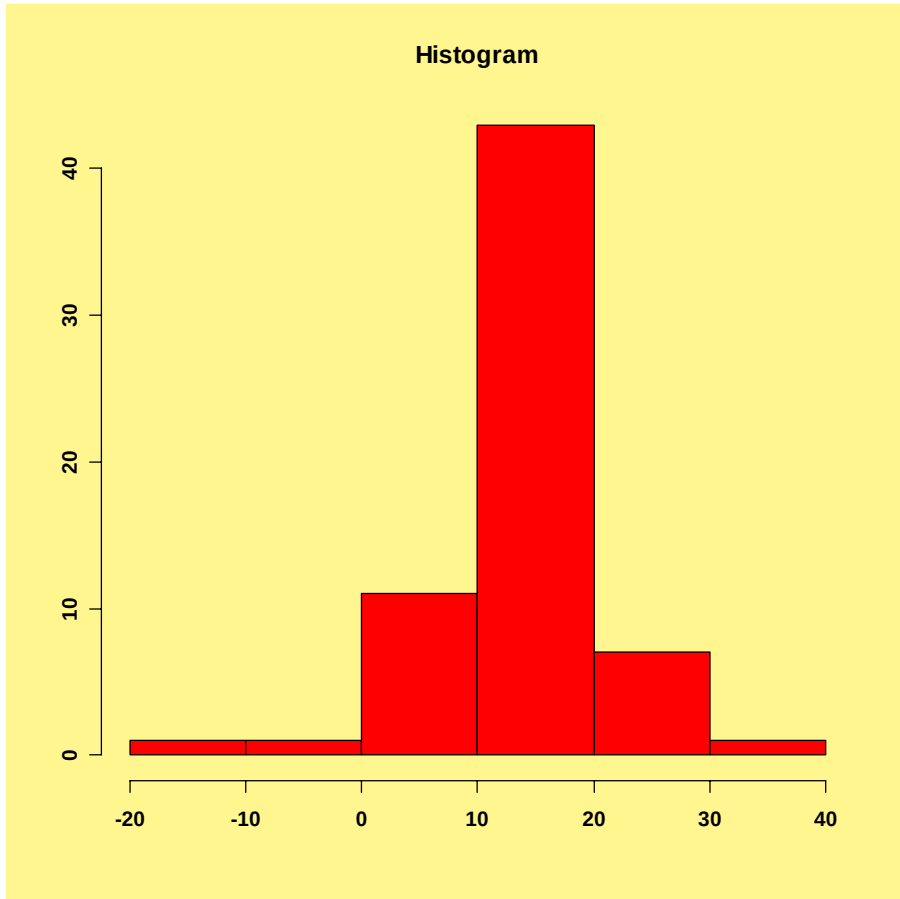




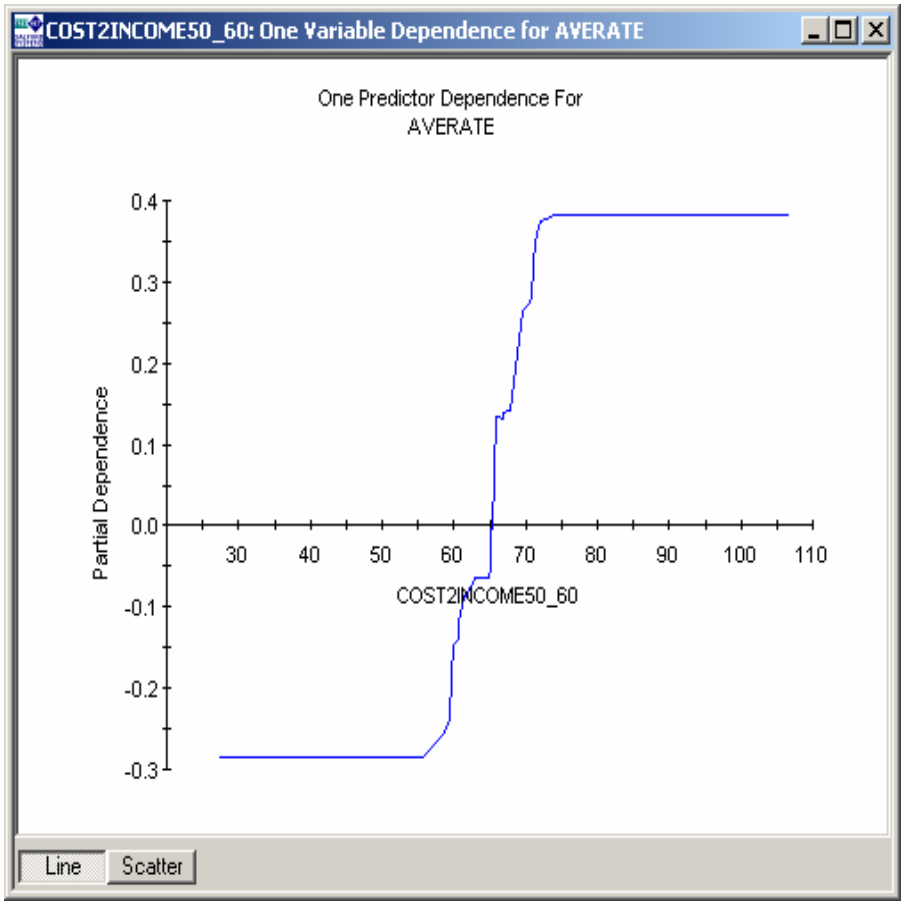
A step function in size of bank

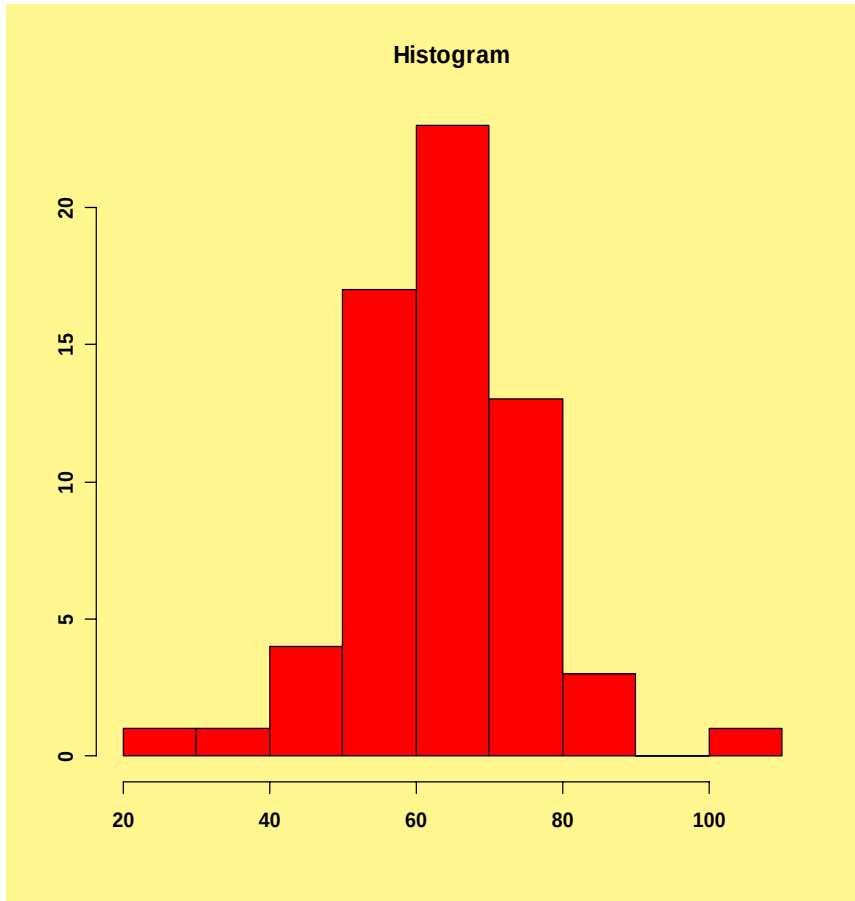
ROAE


ROAE10_20: One Variable Dependence for AVERATE
One Predictor Dependence For
AVERATE
Partial Dependence
0.3
0.2
0.1
-0.1
-0.2
-0.3
-20
-10
10
20
30
40
ROAE10_20
Line
Scatter


Histogram
0
10
20
30
40
-20
-10
0
10
20
30
40

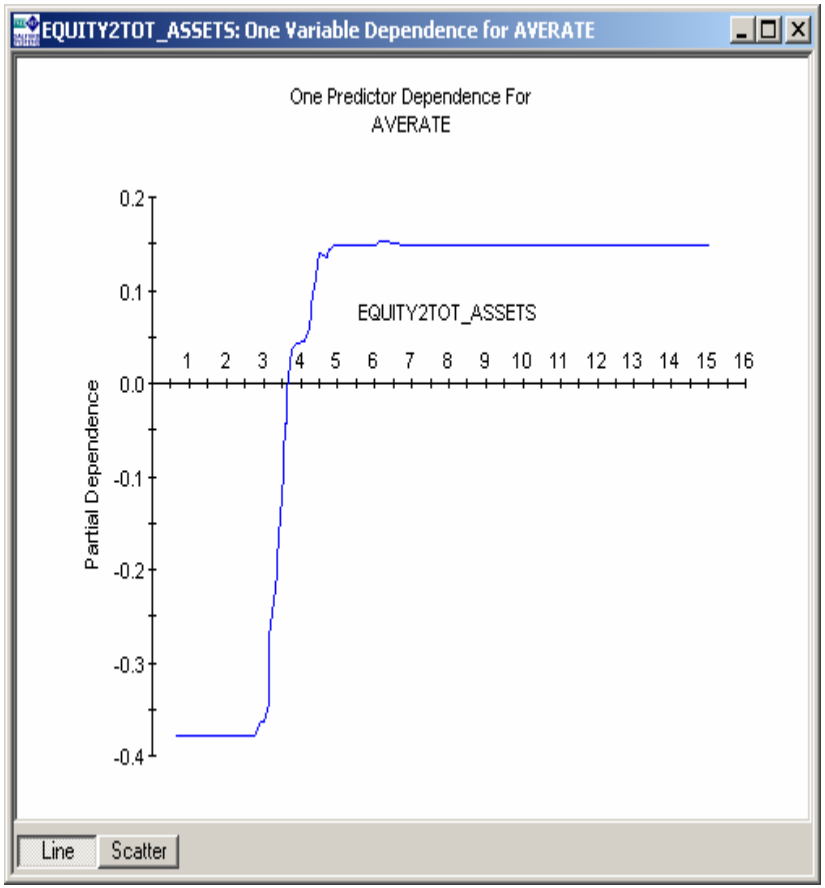
Cost to Income Ratio: Impact on Risk Score

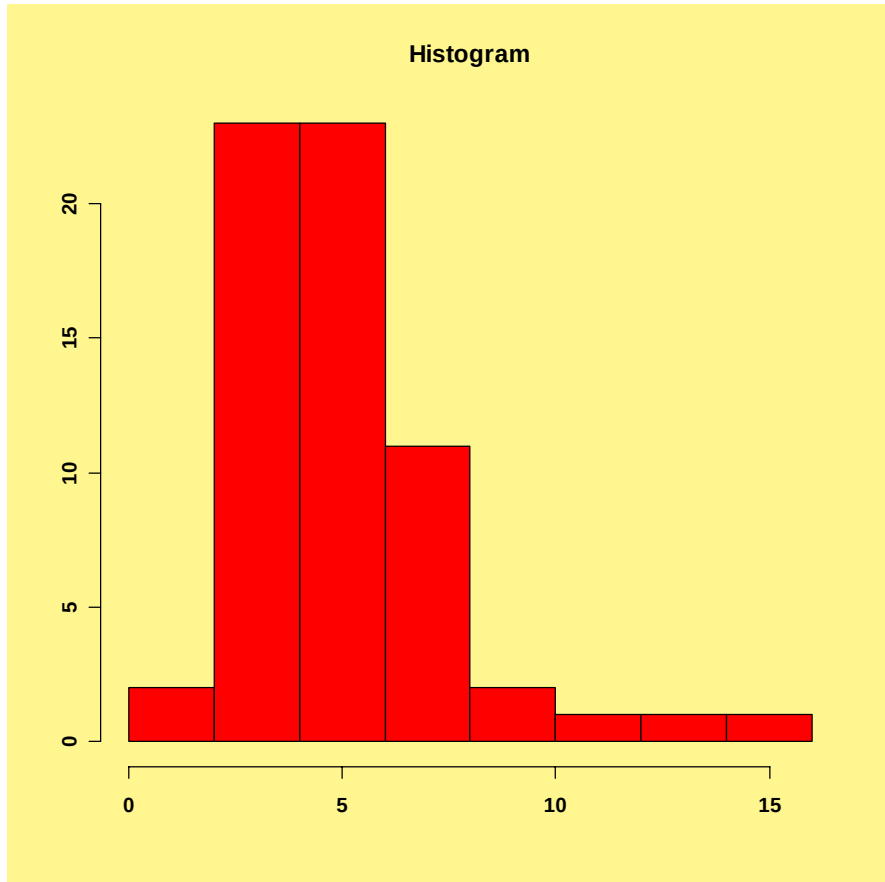




High cost to income increases risk score

Equity to Total Assets





Greater risk forecasted when equity is a large share of total assets

References

- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.
- Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In L. Saitta, ed., *Machine Learning: Proceedings of the Thirteenth National Conference*, Morgan Kaufmann, pp. 148-156.
- Friedman, J.H. (1999). Stochastic gradient boosting. Stanford: Statistics Department, Stanford University.
- Friedman, J.H. (1999). Greedy function approximation: a gradient boosting machine. Stanford: Statistics Department, Stanford University.
- Hastie, T., Tibshirani, R., and Friedman, J.H (2000). The Elements of Statistical Learning. Springer.