Top 10 Data Mining Mistakes
-- and how to avoid them

Salford Systems Data Mining Conference
New York, New York
March 29, 2005

John F. Elder IV, Ph.D.
elder@datamininglab.com

Elder Research, Inc.
635 Berkmar Circle
Charlottesville, Virginia 22901
434-973-7673
www.datamininglab.com

You've made a mistake if you…

0. Lack Data
1. Focus on Training
2. Rely on One Technique
3. Ask the Wrong Question
4. Listen (only) to the Data
5. Accept Leaks from the Future
6. Discount Pesky Cases
7. Extrapolate
8. Answer Every Inquiry
9. Sample Casually
10. Believe the Best Model

0. Lack Data

- Need labeled cases for best gains (to classify or estimate; clustering is much less effective). Interesting known cases may be exceedingly rare. Some projects probably should not proceed until enough critical data is gathered to make it worthwhile.
- Ex: Fraud Detection (Government contracting): Millions of transactions, a handful of known fraud cases; likely that large proportion of fraud cases are, by default, mislabeled clean. Only modest results (initially) after strenuous effort.
- Ex: Fraud Detection (Taxes; collusion): Surprisingly many known cases -> stronger, immediate results.
- Ex: Credit Scoring: Company (randomly) gave credit to thousands of applicants who were risky by conventional scoring method, and monitored them for two years. Then, estimated risk using what was known at start. This large investment in *creating* relevant data paid off.

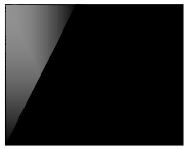
1. Focus on Training

- Only out-of-sample results matter. (Otherwise, use a lookup table!)
- Cancer detection Ex: MD Anderson doctors and researchers (1993), using neural networks, surprised to find that longer training (week vs. day) led to only slightly improved training results, and much worse evaluation results.
- (ML/CS often sought models with exact results on known data -> overfit.)
- **Re-sampling** (bootstrap, cross-validation, jackknife, leave-one-out...) is an *essential* tool. (Traditional significance tests are a weak defense when structure is part of search, though stronger penalty-based metrics are useful.) However, note that resampling no longer tests a single model, but a model class, or a modeling process (Whatever is held constant throughout the sampling passes.)

2. Rely on One Technique

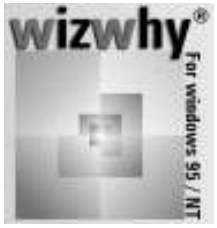
- "To a little boy with a hammer, all the world's a nail."
  For best work, need a whole toolkit.
- At very least, compare your method to a conventional one (linear regression say, or linear discriminant analysis).
- Study: In refereed Neural Network journals, over 3 year period, only 1/6 articles did both ~1 & ~2; that is, test on unseen data and compare to a widely-used technique.
- Not checking other methods leads to blaming the *algorithm* for the results. But, it's somewhat unusual for the particular modeling technique to make a big difference, and when it will is hard to predict.
- Best: use a handful of good tools. (Each adds only 5-10% effort.)

Data Mining Products



ANGOSS

Model 1



DeltaMiner 5

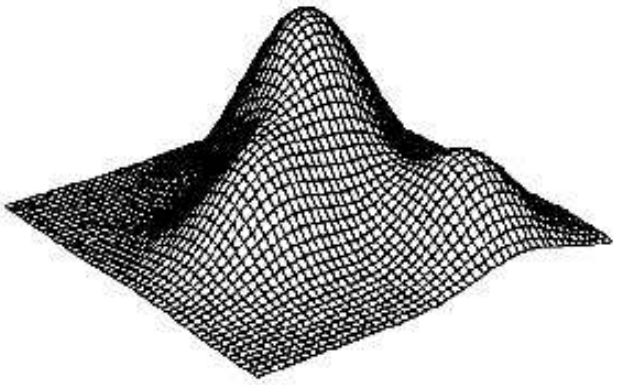
PolyAnalyst 4.5

AnswerTree

NeuroShell 2

Decision Tree

Nearest Neighbor



Delaunay Triangles

Kernel

Neural Network (or Polynomial Network)

Relative Performance Examples: 5 Algorithms on 6 Datasets

(John Elder, Elder Research & Stephen Lee, U. Idaho, 1997)

Error Relative to Peer Techniques (lower is better)

Essentially every Bundling method improves performance

Error Relative to Peer Techniques (lower is better)

3. Ask the Wrong Question

a) Project Goal: Aim at the right target

- Fraud Detection (Positive example!) (Shannon Labs work on Int'l calls): Didn't attempt to classify fraud/nonfraud for general call, but characterized normal behavior for each account, then flagged outliers. -> A brilliant success.

b) Model Goal: Get the computer to "feel" like you do.
[e.g., employee stock grants vs. options]

- Most researchers are drawn into the realm of squared error by its convenience (mathematical beauty). But ask the computer to do what's most helpful for the system, not what's easiest for it. [Stock price ex.]

4. Listen (only) to the Data

4a. Opportunistic data:

- [School funding ex.] *Self-selection*. Nothing inside the data protects analyst from significant, but wrong result.

4b. Designed experiment:

- [Tanks vs. Background with Neural networks]: Great results on out-of-sample portion of database. But found to depend on random pixels (Tanks photographed on *sunny* day, Background only on *cloudy*).
- [Tanks & Networks 2]: Tanks and Trucks on rotating platforms, to train to discriminate at different angles. Used radar, Fourier transforms, principle components, polynomial networks. But, source of the key signal = *platform corner*. And, discriminated between the two classes primarily using *bushes*.

5. Accept Leaks from the Future

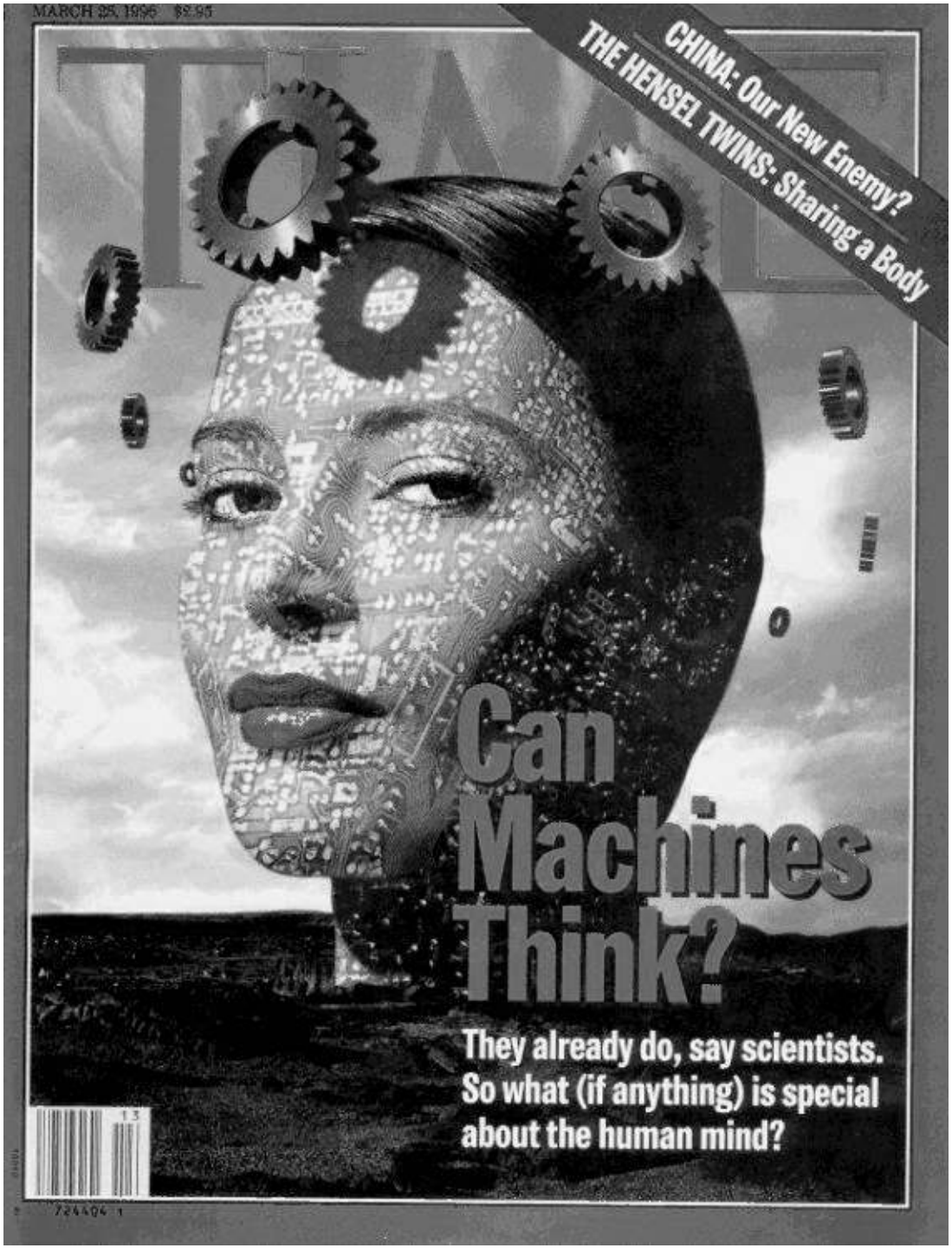
- Forecasting ex.: Interest rate at Chicago Bank.
  N.net 95% accurate, but *output was a candidate input*.
- Financial ex. 2: moving average of 3 days, but centered on *today*.
- Many passes may be needed to expel anachronistic *leakers* - D. Pyle.
- Look for variables which work (too) well.
  Insurance Ex: code associated with 25% of purchasers
  turned out to describe type of cancellation.
- Date-stamp records when storing in Data Warehouse, or
  Don't overwrite old value unless archived.
- Survivor Bias [financial ex.]

6. Discount Pesky Cases

- Outliers may be killing results (ex: decimal point error on price), or be the whole answer (ex: Ozone hole), so examine carefully.
- The most exciting phrase in research isn't "Aha!", but "That's odd…"
- Internal inconsistencies in the data may be clues to problems with the process flow of information within the company; a larger business problem may be revealed.
- Direct Mail example: persisting in hunting down oddities found errors by Merge/Purge house, and was a major contributor to doubling sales per catalog.
- *Visualization* can cover a multitude of assumptions.

7. Extrapolate

- Tend to learn too much from first few experiences.
- Hard to "erase" factoids after an upstream error is discovered.
- *Curse of Dimensionality*: low-dimensional intuition is useless in high-*d*.
- *Philosophical*: Evolutionary Paradigm:
  Believe we can start with pond scum (pre-biotic soup of raw materials)
  + *zap* + time + chance + differential reinforcement -> a critter.
  (e.g., daily stock prices + MARS -> purchase actions,
  or pixel values + neural network -> image classification)
  Better paradigm is selective breeding:
  mutts + time + directed reinforcement -> greyhound
  Higher-order features of data + domain expertise essential



"Of course machines can think. After all, humans are just machines made of meat."
- MIT CS professor

Human and computer strengths are more complementary than alike.

8. Answer Every Inquiry

- "Don't Know" is a useful model output state.
- Could estimate the *uncertainty* for each output (a function of the number and spread of samples near *X*). Few algorithms provide a conditional σ with their conditional μ.

Global R^d Optimization when Probes are Expensive (GROPE)

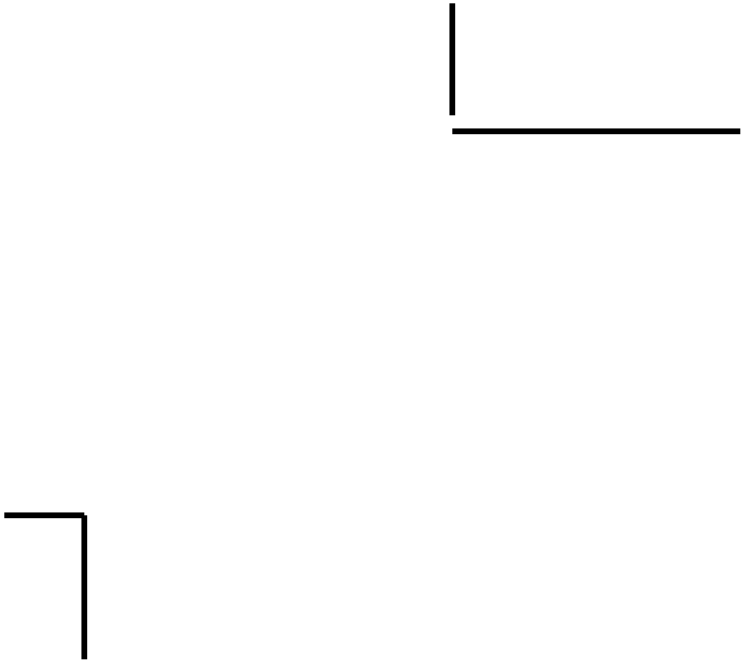
9. Sample without Care

- **9a *Down-sample*** Ex: MD Direct Mailing firm had too many non-responders (NR) for model (about 99% of >1M cases). So took all responders, and every 10th NR to create a more balanced database of 100K cases. Model predicted that *everyone* in Ketchikan, Wrangell, and Ward Cove Alaska would respond. (*Sorted* data, by *zip code*, and 100Kth case drawn before bottom of file (999**) reached.)
- "Shake before baking". Also, add case number (and other random variables) to candidate list; use as "canaries in the mine" to signal trouble when chosen.
- **9b *Up-sampling*** Ex: Credit Scoring - paucity of interesting (Default) cases led to quintupling them. Cross-validation employed with many techniques and modeling cycles. Results tended to improve with the complexity of the models. Oddly, this didn't reverse. Noticed that Default (rare) cases were better estimated by complex models but others were worse. (Had *duplicated* Defaults in each set by *up-sampling* before splitting.)
  -> Split first.
- It's hard to beat a stratified sample; that is, proportional sample from each group. [even with *Data Squashing* - W. DeMouchel]

10. Believe the Best Model

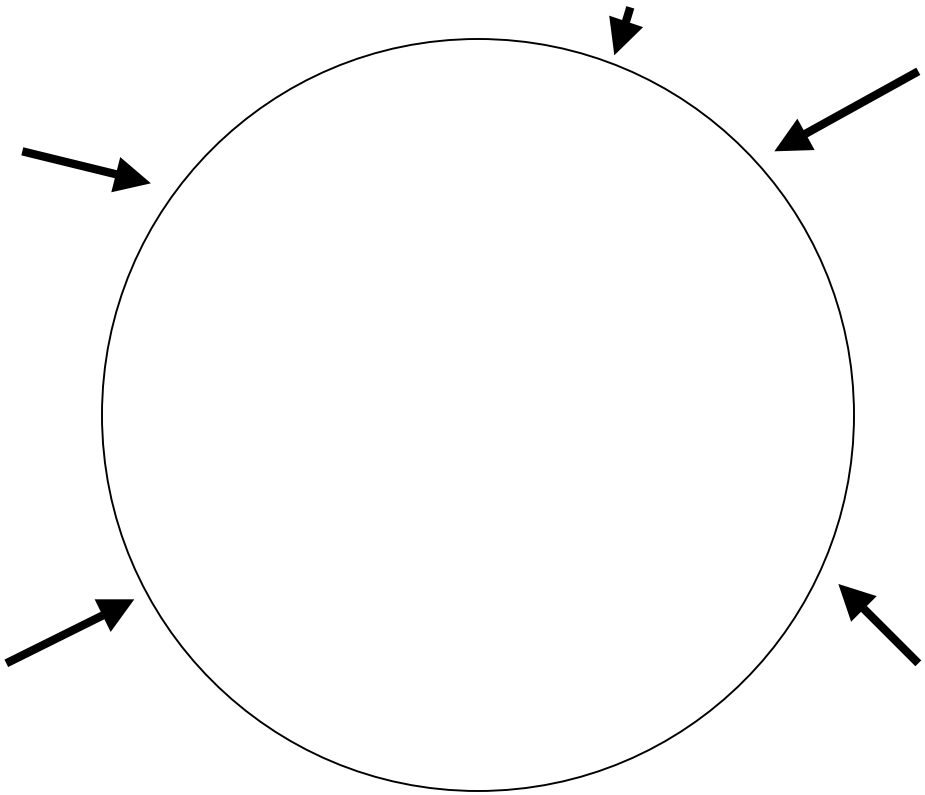
- Interpretability not always necessary.
  Model can be useful without being "correct" or explanatory.
- Often, particular variables used by "best" model (which barely won out over hundreds of others of the millions (to billions) tried, using a score function only approximating one's goals, and on finite data) have too much attention paid to them. (*Un-interpretability* could be a virtue!).
- Usually, many very similar variables are available, and the particular structure of the best model can vary chaotically. [Polynomial Network Ex.] But, structural similarity is different from functional similarity. (Competing models often look different, but act the same.)
- Best estimator is likely to be *bundle* of models.
  [Direct Marketing trees] [5x6 table] [Credit Scoring ex.]

Lift Chart: %purchasers vs. %prospects

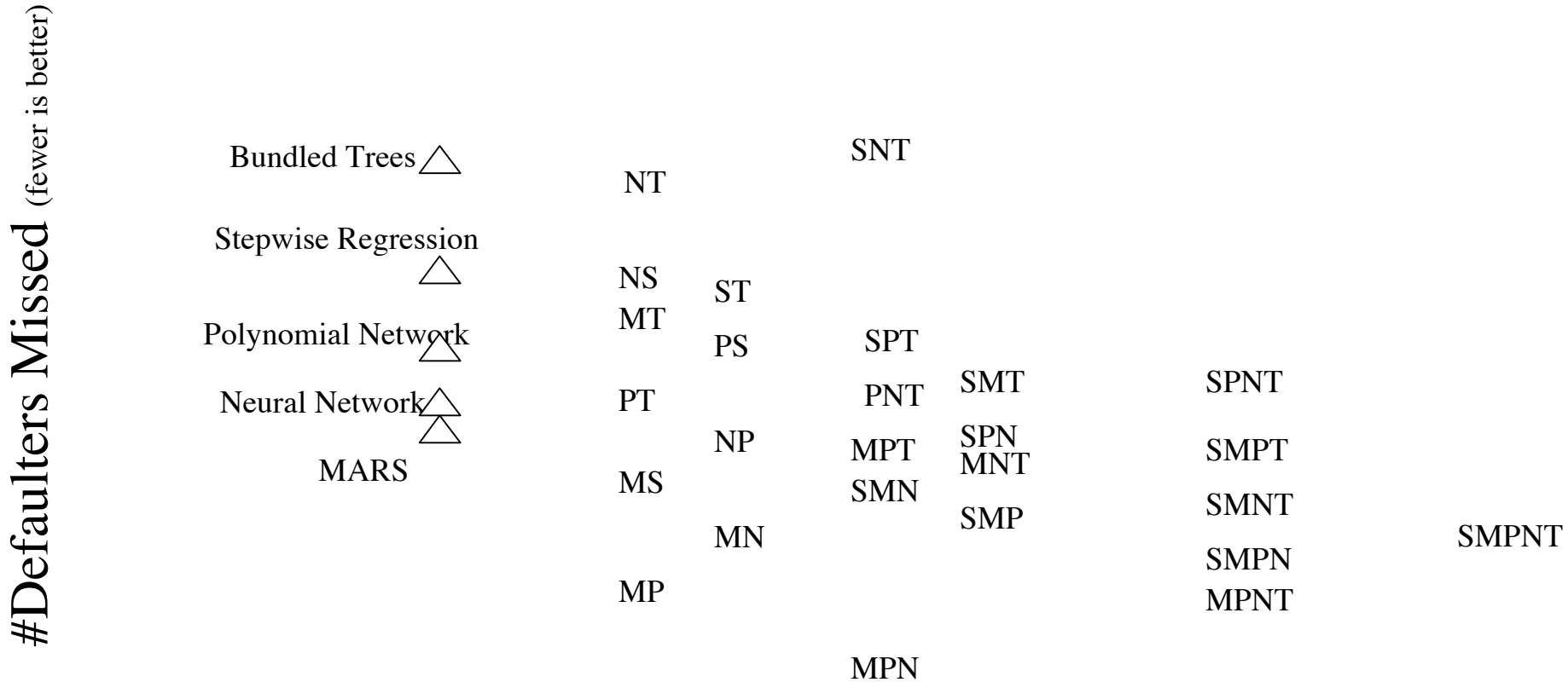


- Ex: Last quintile of customers are 4 times more expensive to obtain than first quintile (10% vs. 40% to gain 20%)
- Decision Tree provides relatively few decision points.

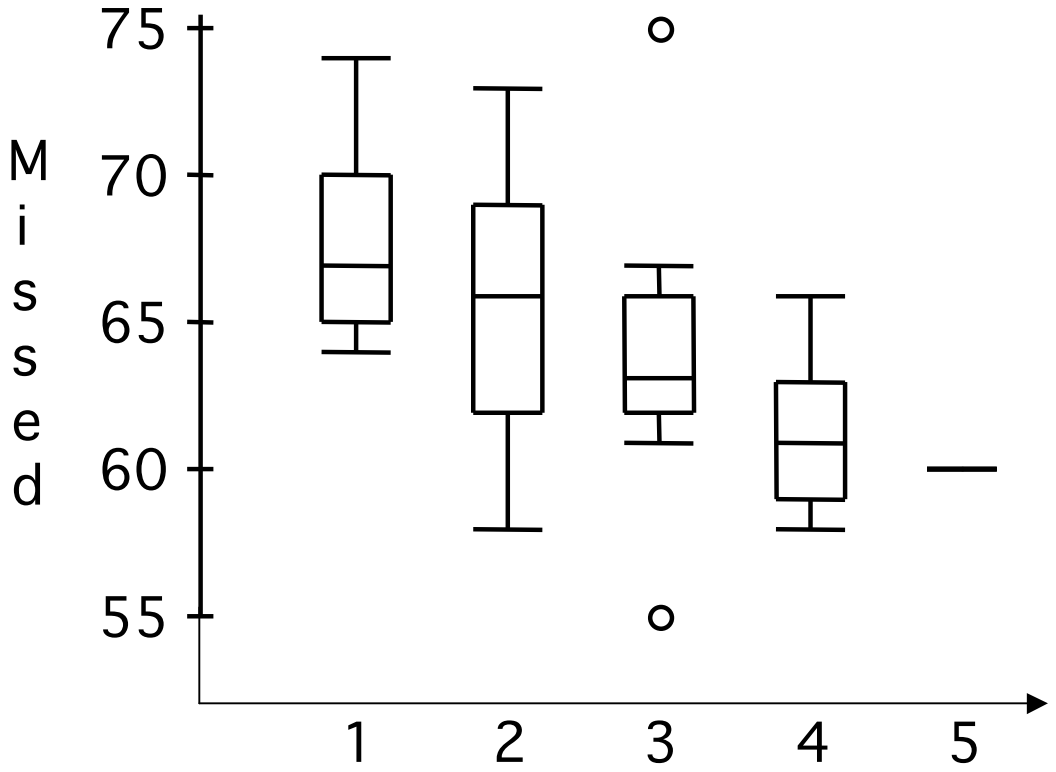
Bundling 5 Trees
improves accuracy and smoothness



Credit Scoring Model Performance



Median (and Mean) Error Reduced with each Stage of Combination



Fancier tools and harder problems –> more ways to mess up.

How then can we succeed?

Success <- Learning <- Experience <- Mistakes

(so go out and make some good ones!)

PATH to success:

- **Persistence** - Attack repeatedly, from different angles.
  Automate essential steps. Externally check work.
- **Attitude** - Optimistic, can-do.
- **Teamwork** - Business and statistical experts must cooperate.
  Does everyone *want* the project to succeed?
- **Humility** - Learning from others requires vulnerability.
  Don't expect too much of technology.

John F. Elder IV
Chief Scientist, ERI

Dr. John Elder heads a data mining consulting team with offices in Charlottesville, Virginia and Washington DC, and close affiliates in Boston, New York, San Diego, and San Francisco (*www.datamininglab.com*). Founded in 1995, Elder Research, Inc. focuses on investment and commercial applications of pattern discovery and optimization, including stock selection, image recognition, process optimization, cross-selling, biometrics, drug efficacy, credit scoring, market timing, and fraud detection.

John obtained a BS and MEE in Electrical Engineering from Rice University, and a PhD in Systems Engineering from the University of Virginia, where he's an adjunct professor, teaching Optimization. Prior to a decade leading ERI, he spent 5 years in aerospace defense consulting, 4 heading research at an investment management firm, and 2 in Rice's *Computational & Applied Mathematics* department.

Dr. Elder has authored innovative data mining tools, is active on Statistics, Engineering, and Finance conferences and boards, is a frequent keynote conference speaker, and was a Program co-chair of the 2004 *Knowledge Discovery and Data Mining* conference. John's courses on data analysis techniques – taught at dozens of universities, companies, and government labs -- are noted for their clarity and effectiveness. Dr. Elder holds a top secret clearance, and since the Fall of 2001, has been honored to serve on a panel appointed by Congress to guide technology for the National Security Agency.

John is a follower of Christ and the proud father of 5.