Institute of Actuaries of Australia

INSIGHTS

Predicative Modeling and Data Mining for Actuaries

Dan Steinberg, Ph.D.

Mikhail Golovnya

http://www.salford-systems.com

Salford Systems

Institute of Actuaries of Australia

What Everyone Hears

Predictive Analytics

Machine Learning

Pattern Recognition

Artificial Intelligence

Business Intelligence

Data Warehousing

OLAP, CART, SVM, NN, etc.

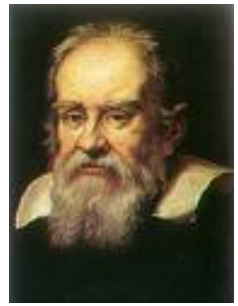
CRISP-DM, CRM, KDD, etc.

DATA MINING

- Statistics
- Computer science
- Database Management
- Insurance
- Finance
- Marketing
- Electrical Engineering
- Robotics
- Biotech and more

Institute of Actuaries of Australia

Data Mining Defined



- Data mining is the search for patterns in data using modern highly automated, computer intensive methods
  - Data mining may be best defined as the use of a specific class of tools (data mining methods) in the analysis of data
  - The term “search” is key to this definition, as is “automated”
- The literature often refers to finding hidden information in data

The Three Sisters

Science

Study the phenomenon
Understand its nature
Try to discover a law
The laws usually hold for a long time

Statistics

Collect some data
Guess the model (perhaps, using science)
Use the data to clarify and/or validate the model
If looks “fishy”, pick another model and do it again

Data Mining

Access to lots of data
No clue what the model might be
No long term law is even possible
Let the machine build a model
And let’s use this model while we can

Data Mining Mythology

- **Quest for the Holy Grail** – build an algorithm that will always find 100% accurate models
- **Absolute Powers** – data mining will finally find and explain everything
- **Gold Rush** – with the right tool one can rip the stock-market and become obscenely rich
- **Magic Wand** – getting a complete solution from start to finish with a single button push
- **Doomsday Scenario** – all conventional analysts will eventually be replaced by smart computer chips

Patterns Searched For

- We will focus on patterns that allow us to accomplish two tasks:
  - Classification
  - Regression

This is known as "supervised learning"

- We will briefly touch on a third common task
  - Finding groups in data (clustering, density estimation)

This is known as "unsupervised learning"

- There are other patterns we will not discuss today including
  - Patterns in sequences
  - Connections in networks (the web, social networks, link analysis)

Major Data Mining Tools

- **CART®** (Decision Trees, C4.5, CHAID among others)
- **MARS®** (Multivariate Adaptive Regression Splines)
- Artificial Neural Networks (ANNs, many commercial)
- Association Rules (Clustering, market basket analysis)
- **TreeNet®** (Stochastic Gradient Tree Boosting)
- **RandomForests®** (Ensembles of trees w/ random splits)
- Genetic Algorithms (evolutionary model development)
- Self Organizing Maps (SOM, like k-means clustering)
- Support Vector Machine (SVM wrapped in many patents)
- Nearest Neighbor Classifiers

CART® Overview

- **Classification and Regression Trees** (CART®) – original approach based on the "let the data decide local regions" concept developed by Breiman, Friedman, Olshen, and Stone:
  - For each current data region, consider all possible orthogonal splits (based on one variable) into 2 sub-regions
  - The best split is then defined as the one having the smallest MSE after fitting a constant in each sub-region (regression) or the smallest resulting impurity (classification)
  - Proceed sequentially until all structure in the training set has been completely exhausted ➔ largest tree is produced
  - Create a sequence of nested sub-trees with different amount of localization
- Pick the best tree based on the Test set performance

CART® – Major Strengths

- Relatively fast
- Requires minimal supervision by analyst
- Produces easy to understand models
- Conducts automatic variable selection
- Handles missing values via surrogate splits
- Invariant to monotonic transformations of predictors
- Impervious to outliers

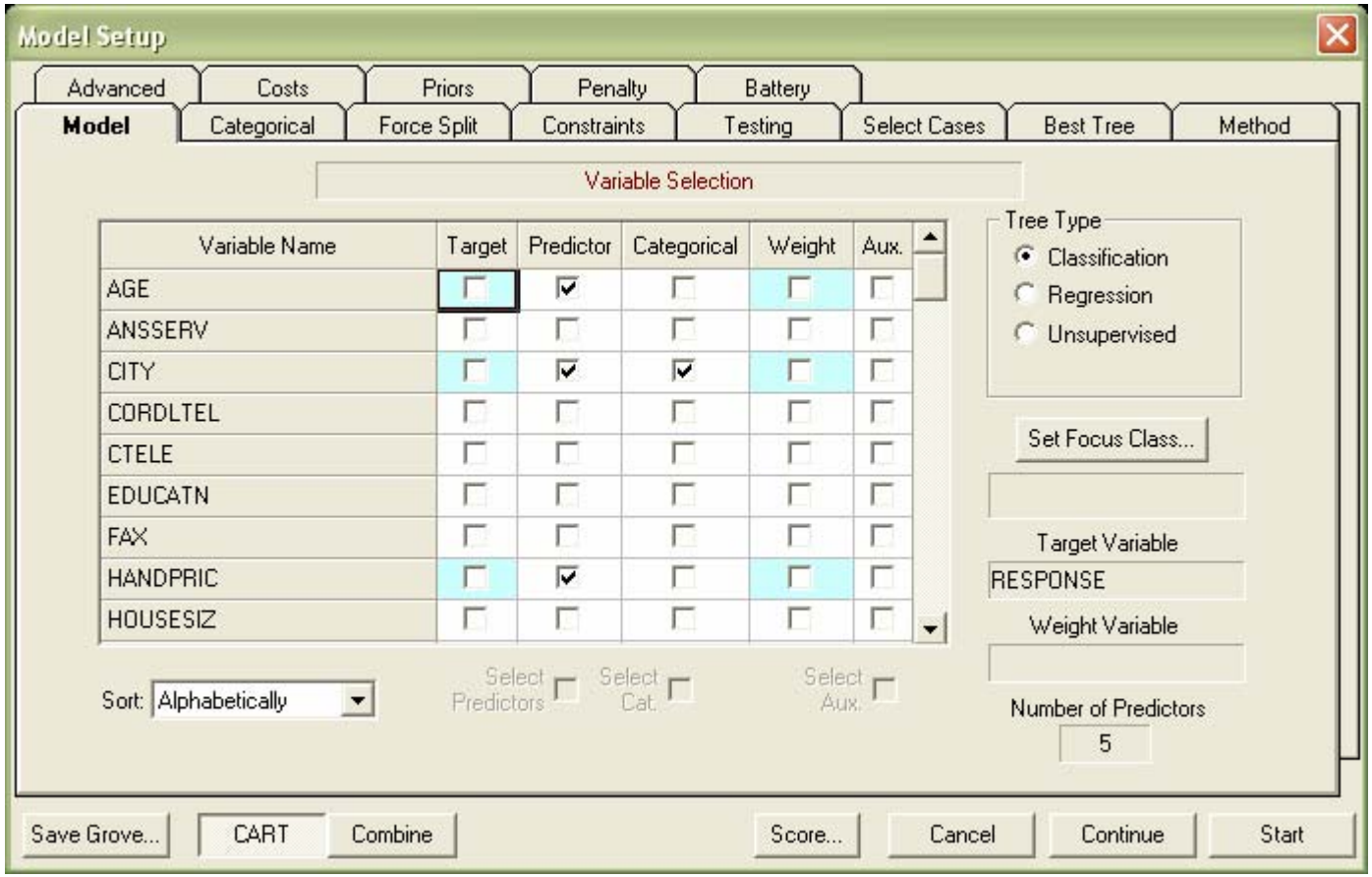
Classification with CART®
(real world study early 1990s)

- Fixed line service provider offering a new mobile phone service
- Wants to identify customers most likely to accept new mobile offer
- Data set based on limited market trial

Real World Trial

- 830 households offered a mobile phone package
- All offered identical package but pricing was varied at random
  - Handset prices ranged from low to high values
  - Per minute prices ranged separately from low to high rate rates
- Household asked to make yes or no decision on offer
- 15.2% of the households accepted the offer
- Our goal was to answer two key questions
  - Who to make offers to?
  - How to price?

Setting Up the Model



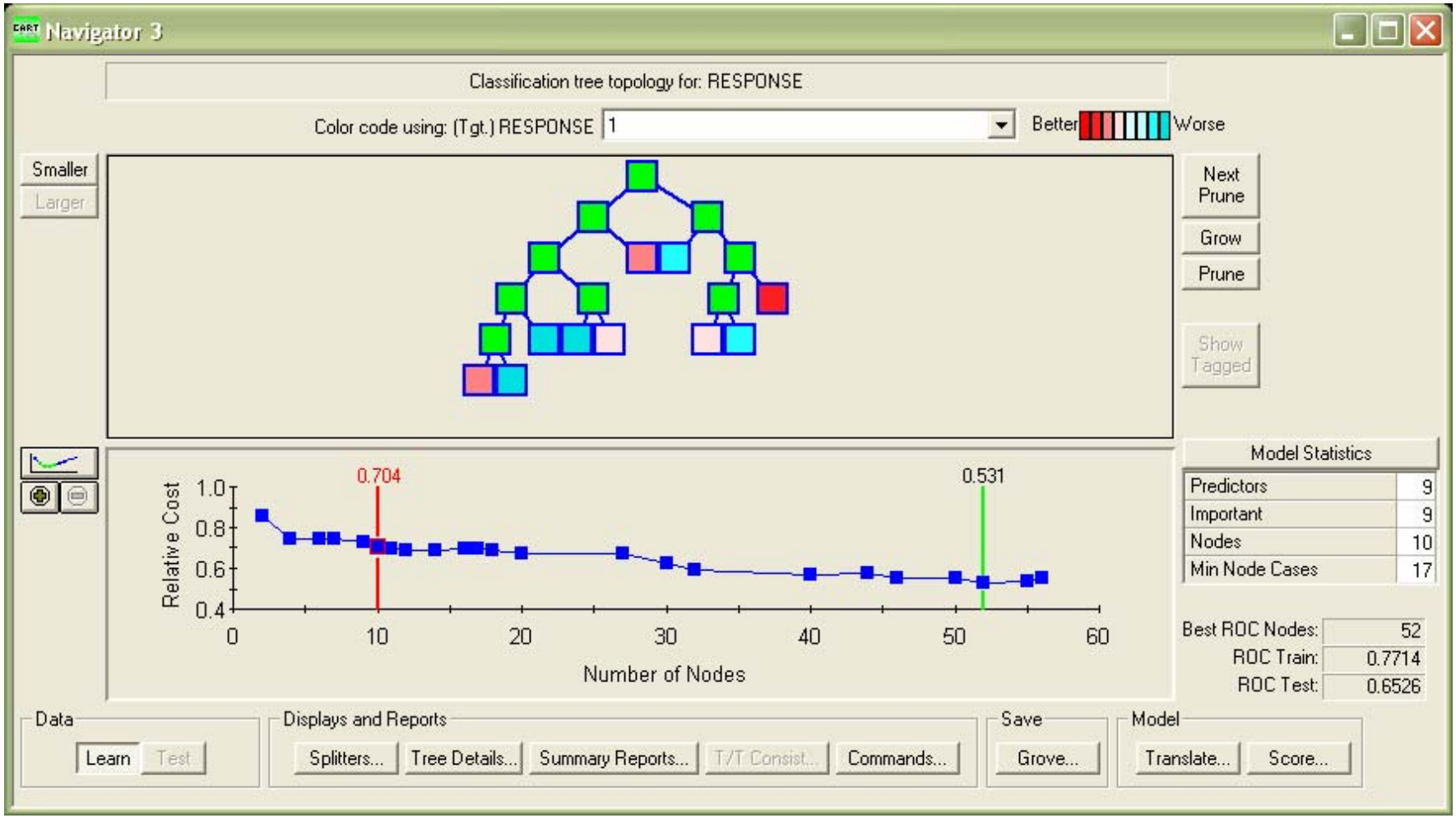
- Only requirement is to select TARGET (dependent) variable. CART will do everything else automatically

Institute of Actuaries of Australia

Model Results

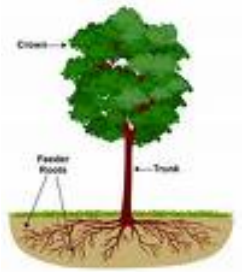
- CART completes analysis and gives access to all results from the NAVIGATOR
  - Shown on the next slide
- Upper section displays tree of a selected size
- Lower section displays error rate for trees of all possible sizes
- Green bar marks most accurate tree
- We display a compact 10 node tree for further scrutiny

Navigator Display



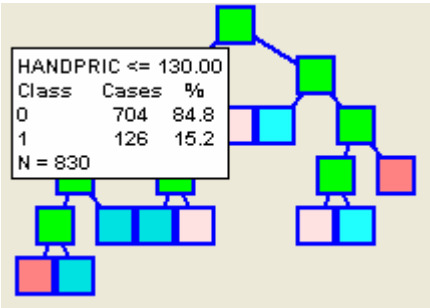
- Most accurate tree is marked with the green bar. We select the 10 node tree for convenience of a more compact display.

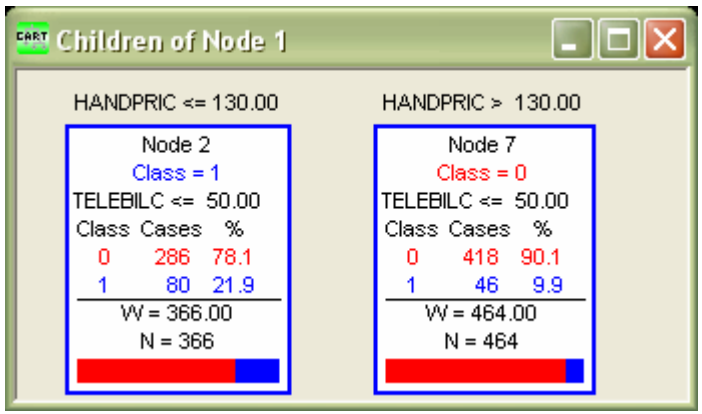
Institute of Actuaries of Australia



Root Node

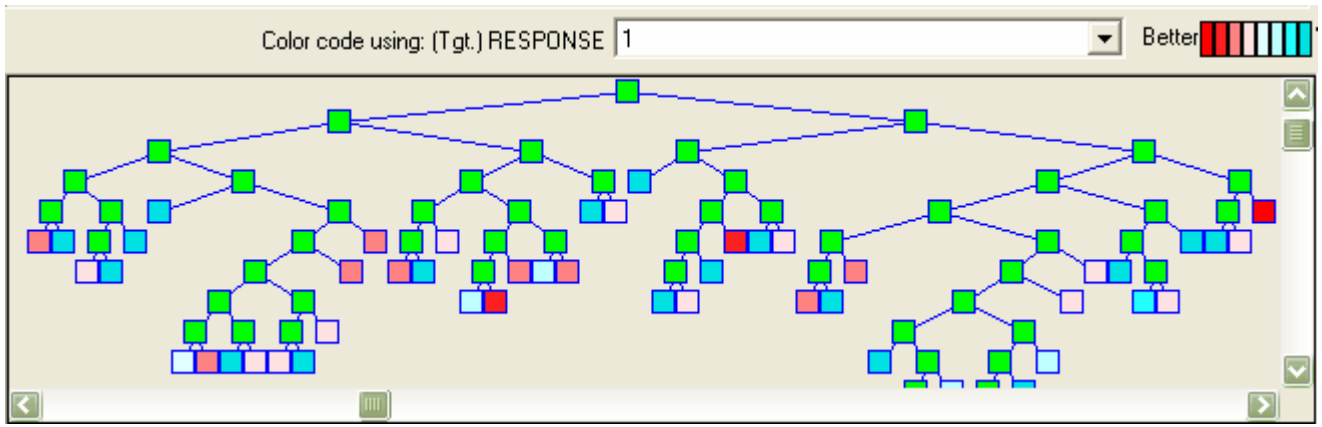
- Root node displays details of TARGET variable in overall training data
- We see that 15.2% of the 830 households accepted the offer
- Goal of the analysis is now to extract patterns characteristic of responders





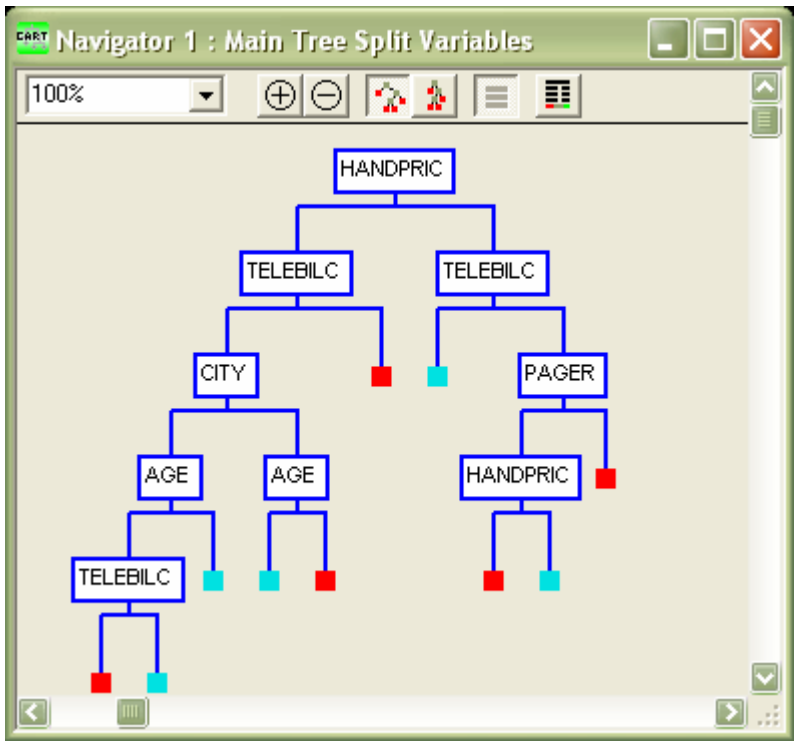
- If we could only use a single piece of information to separate responders from non-responders CART chooses the HANDSET PRICE
- Those offered the phone with a price > 130 contain only 9.9% responders
- Those offered a lower price respond at 21.9%

Maximal Tree



- Maximal tree is the raw material for the best model
- Goal is to find optimal tree embedded inside maximal tree
- Will find optimal tree via “pruning”
- Like backwards stepwise regression

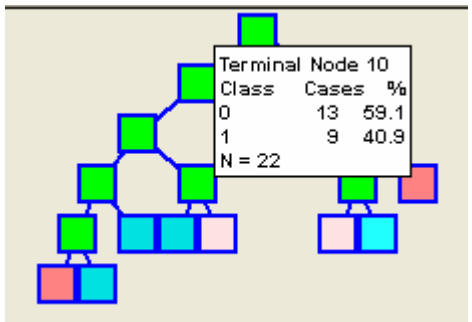
Model Overview: Main Drivers



- Red= Good Response Blue=Poor Response)
- High values of a split variable always go to the right; low values go left

Institute of Actuaries of Australia

Examine Individual Node



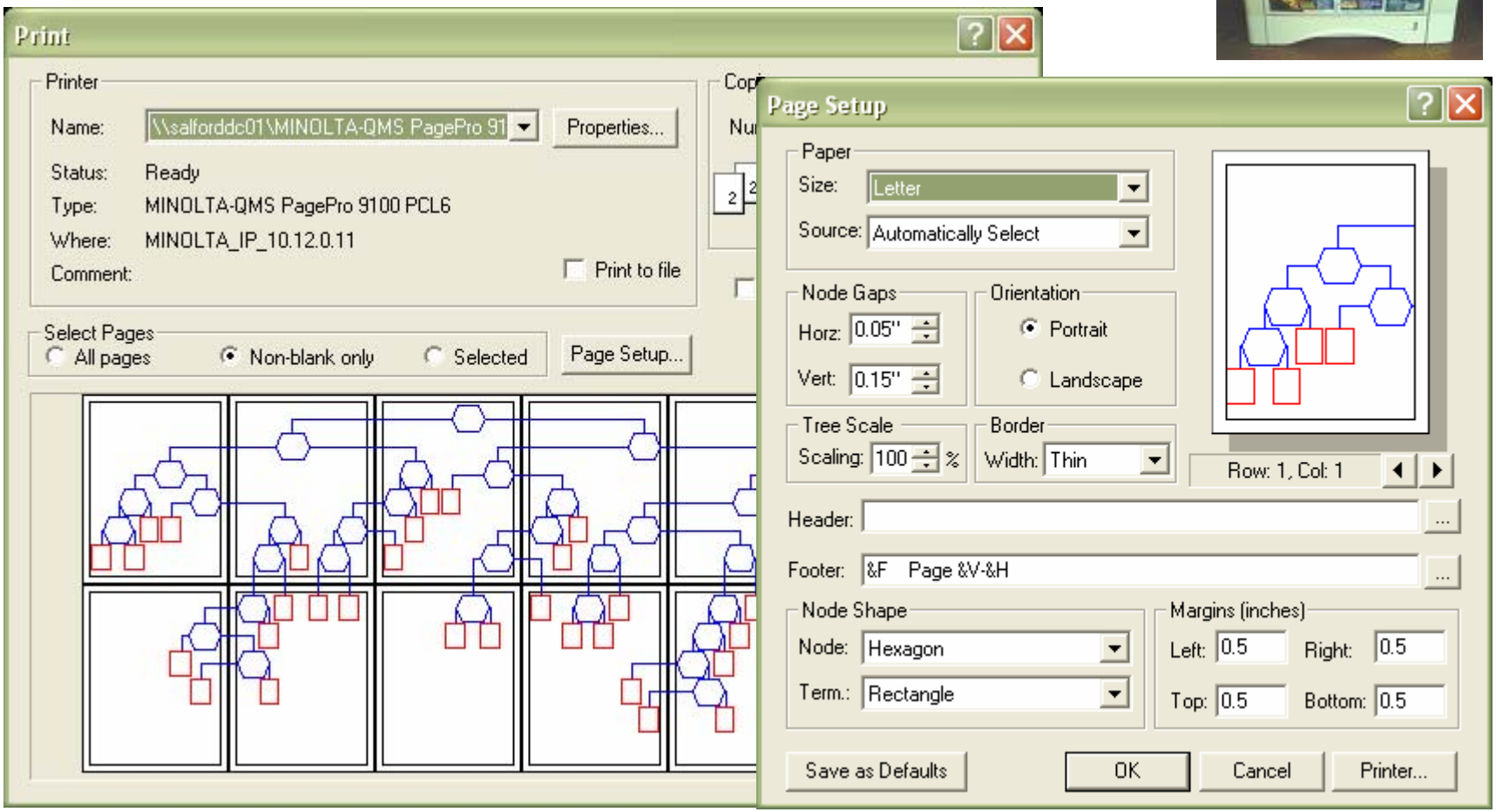
Terminal Node 10

Classification

```
/*Rules for terminal node 10*/

if
(
  (
     PAGER == 1
  ) &&
    HANDPRIC > 130 &&
    TELEBILC > 50
)

{
    terminalNode = 10;
    class = 1;
    probClass1 = 0.590909;
    probClass2 = 0.409091;
}
```

- Hover mouse over node to see inside
- Even though this node is on the “high price” of the tree it still exhibits the strongest response across all terminal node segments (43.5% response)
- Here we select rules expressed in C
- Entire tree can also be rendered in Java, XML/PMML, or SAS

Configure Print Image



- Variety of fine controls to position and scale image nicely

Institute of Actuaries of Australia

Variable Importance Ranking

Navigator 4 (33): Tree Summary Reports

Misclassification | Prediction Success | Gains Chart | Profit | Root Splits | Terminal Nodes | **Variable Importance**

Variable Importance

Variable	Score
TELEBILC	100.00
TRAVTIME	87.22
AGE	86.83
CITY	63.59
HANDPRIC	54.36
SEX	47.15
RENTHOUS	29.03
PAGER	19.01
MARITAL	17.34

Consider Only Primary Splitters

Show zero importance variables

Discount Surrogates

By association

Geometric 1.0

Use only top 0

New Keep List | New Keep & Build

- CART offers multiple ways to compute variable importance

Predictive Accuracy

Navigator 3 (52): Tree Summary Reports

Gains Chart | Profit | Root Splits | Terminal Nodes | Variable Importance

Misclassification | **Prediction Success**

Test Sample Prediction Success Table

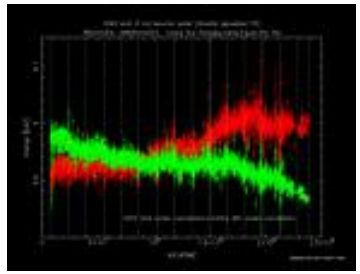
Actual Class	Total Cases	Percent Correct	Predicted Class 0 N=567	Predicted Class 1 N=263
0	704	75.43	531	173
1	126	71.43	36	90
Total:	**830.00**			
Average:		73.43		
Overall % Correct:		74.82		

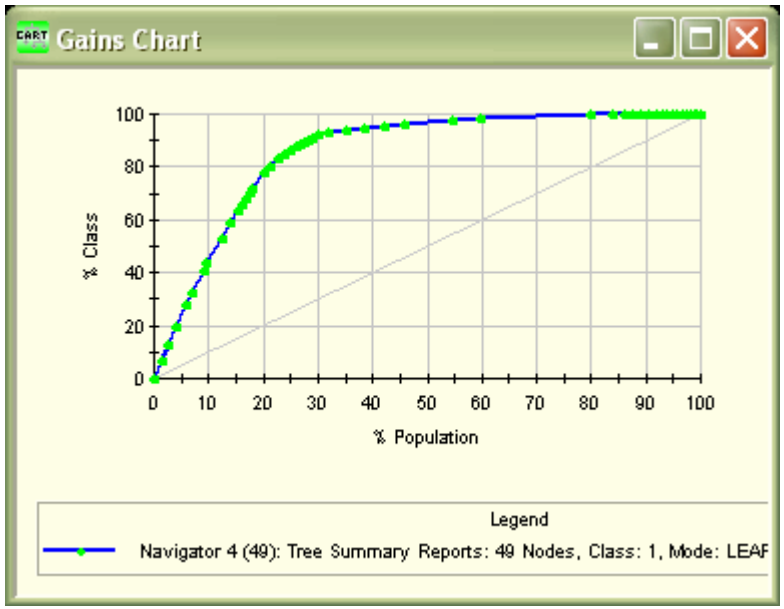
Learn | Test | Class: None | Count | Row % | Column %

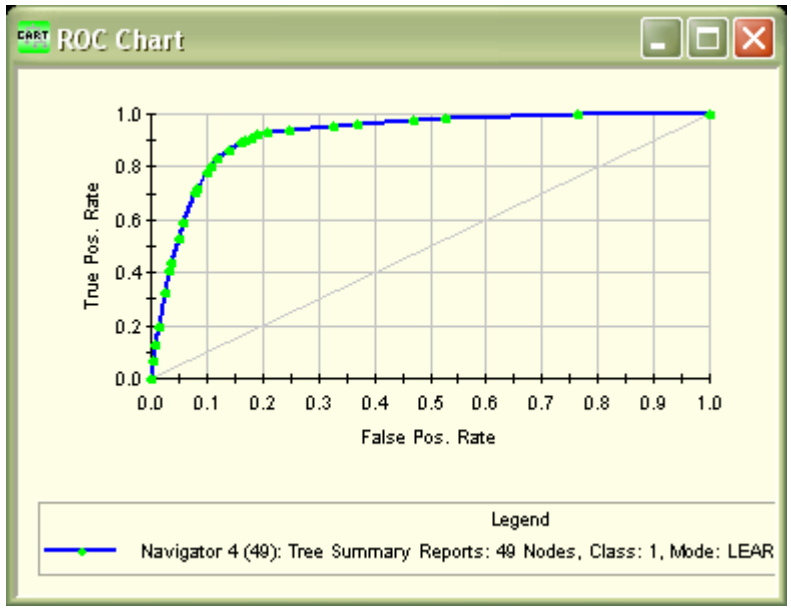
- This model is not very accurate but ranks responders well

Institute of Actuaries of Australia

Gains and ROC Curves

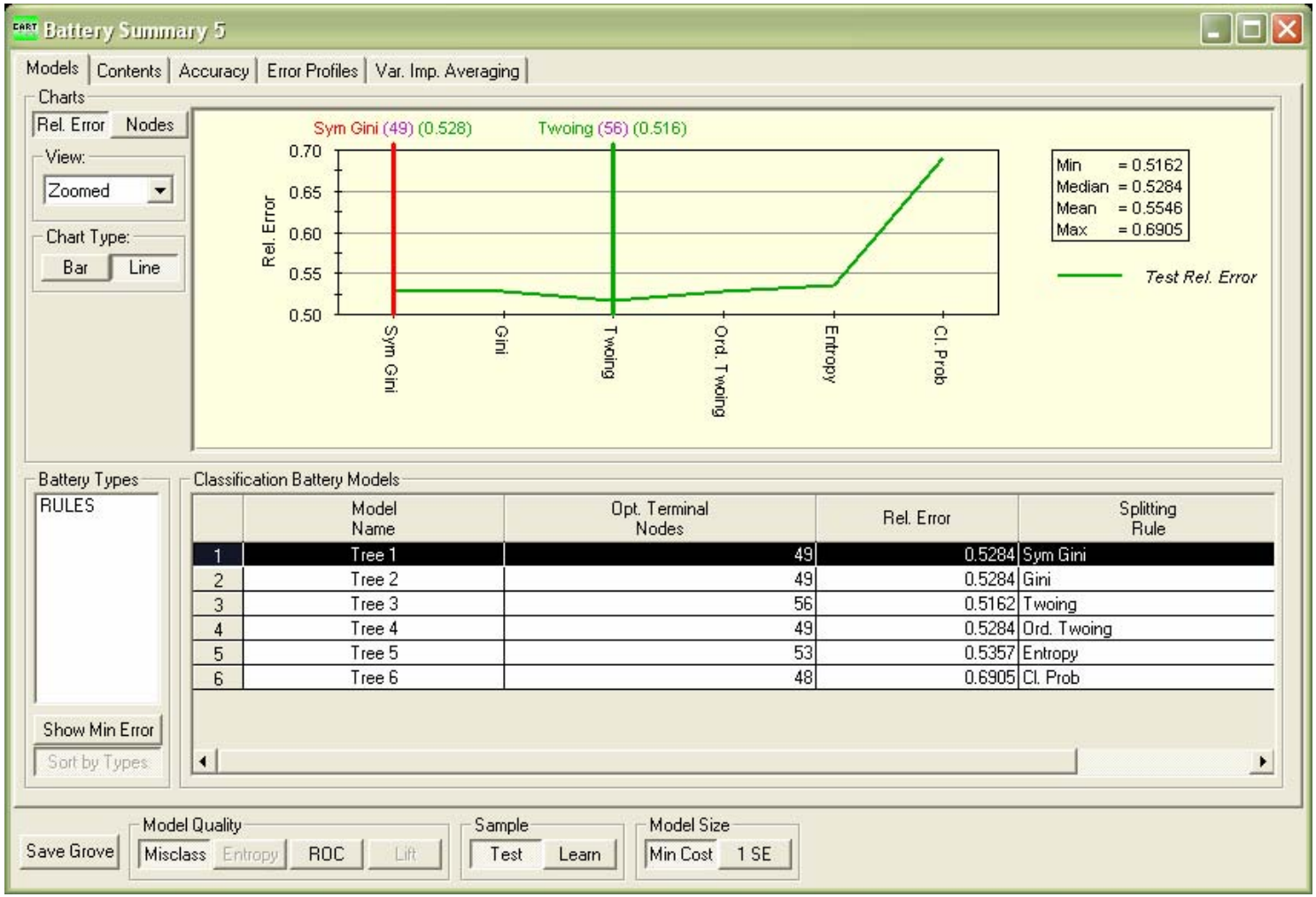






- In top decile model captures about 40% of responders

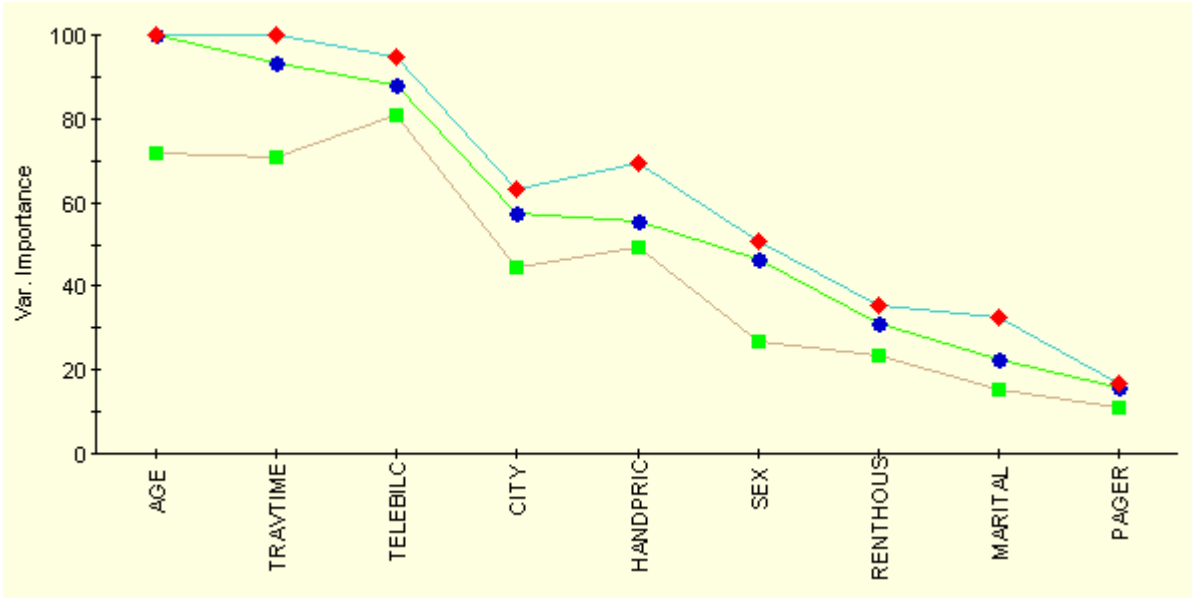
Modeling Automation

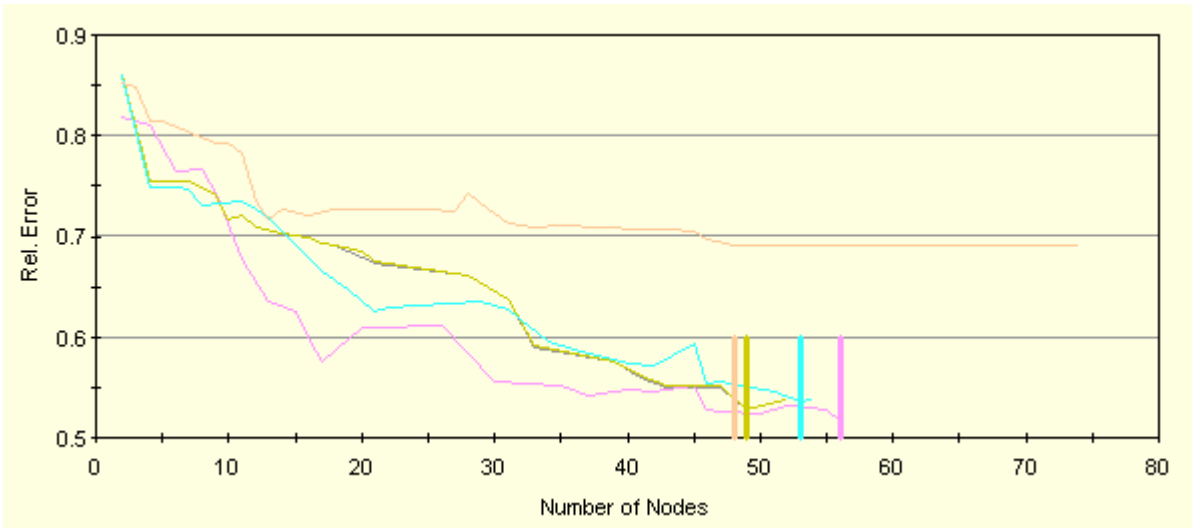


- Here we display results for each of the 6 major tree growing methods. Twoing yields best performance here. This is one of 18 different automation schemes

Institute of Actuaries of Australia

Battery Summaries

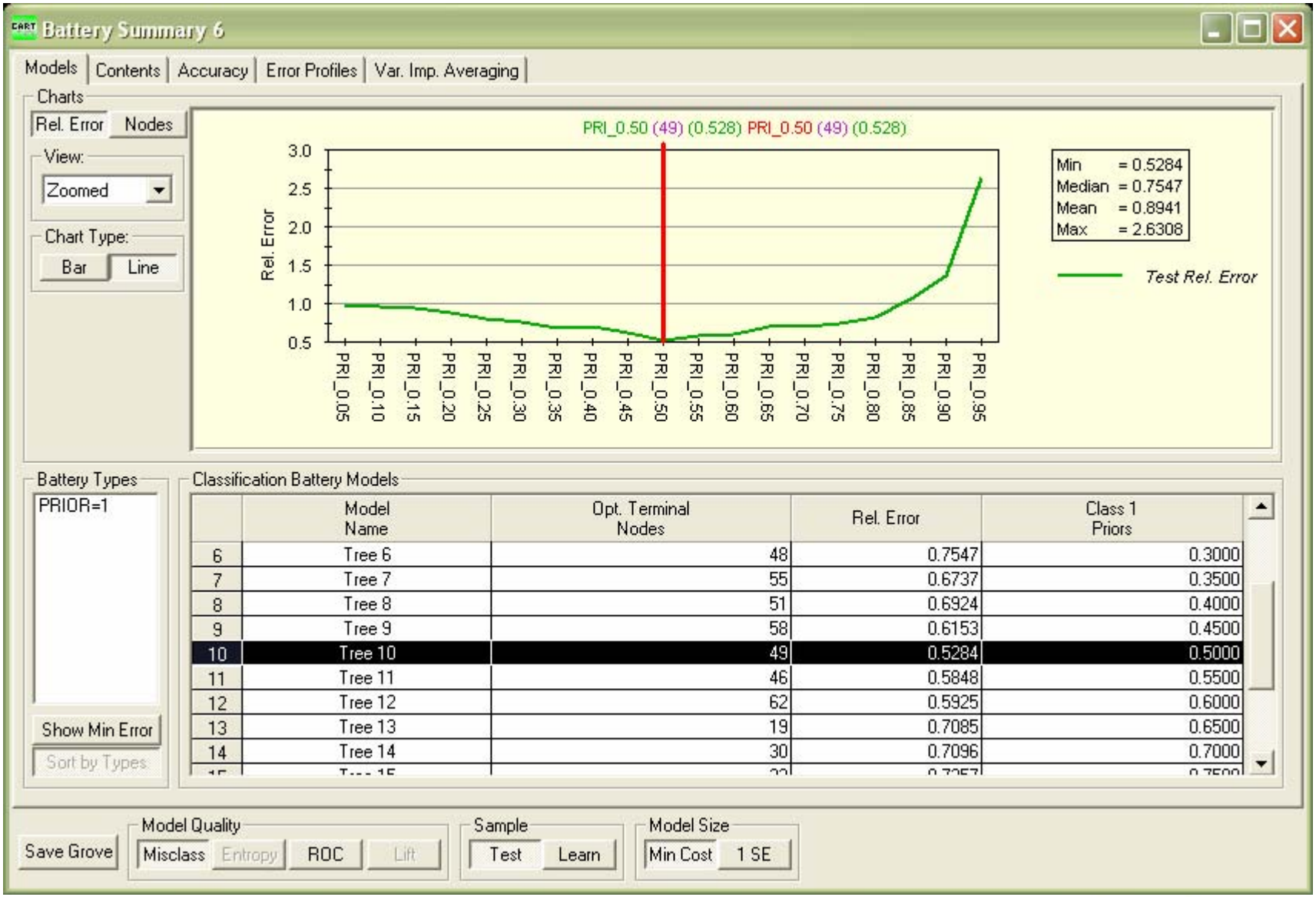




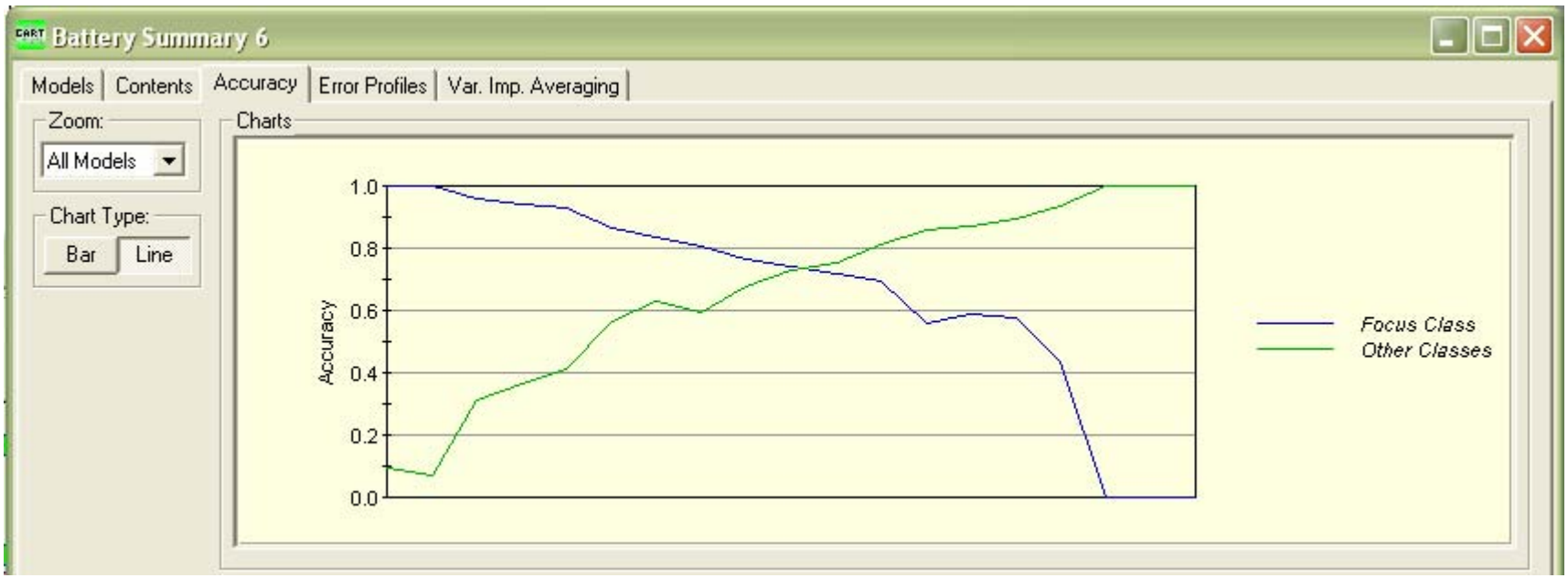
- Variable importance across alternative strategies
- Error profiles for each alternative strategy
- A number of other summaries also available

Institute of Actuaries of Australia

Vary Penalty on False Positives



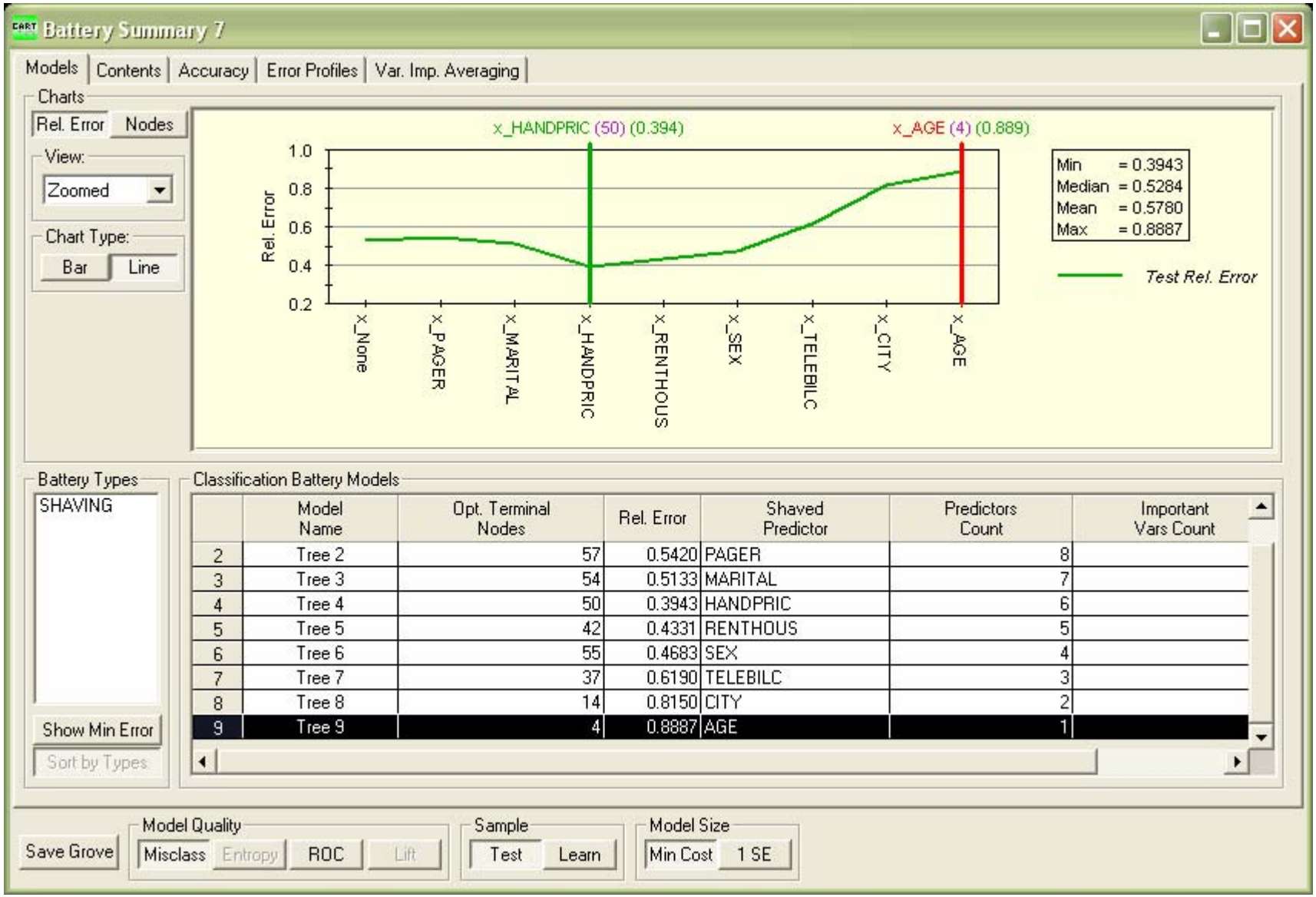
Class Accuracy Tradeoff



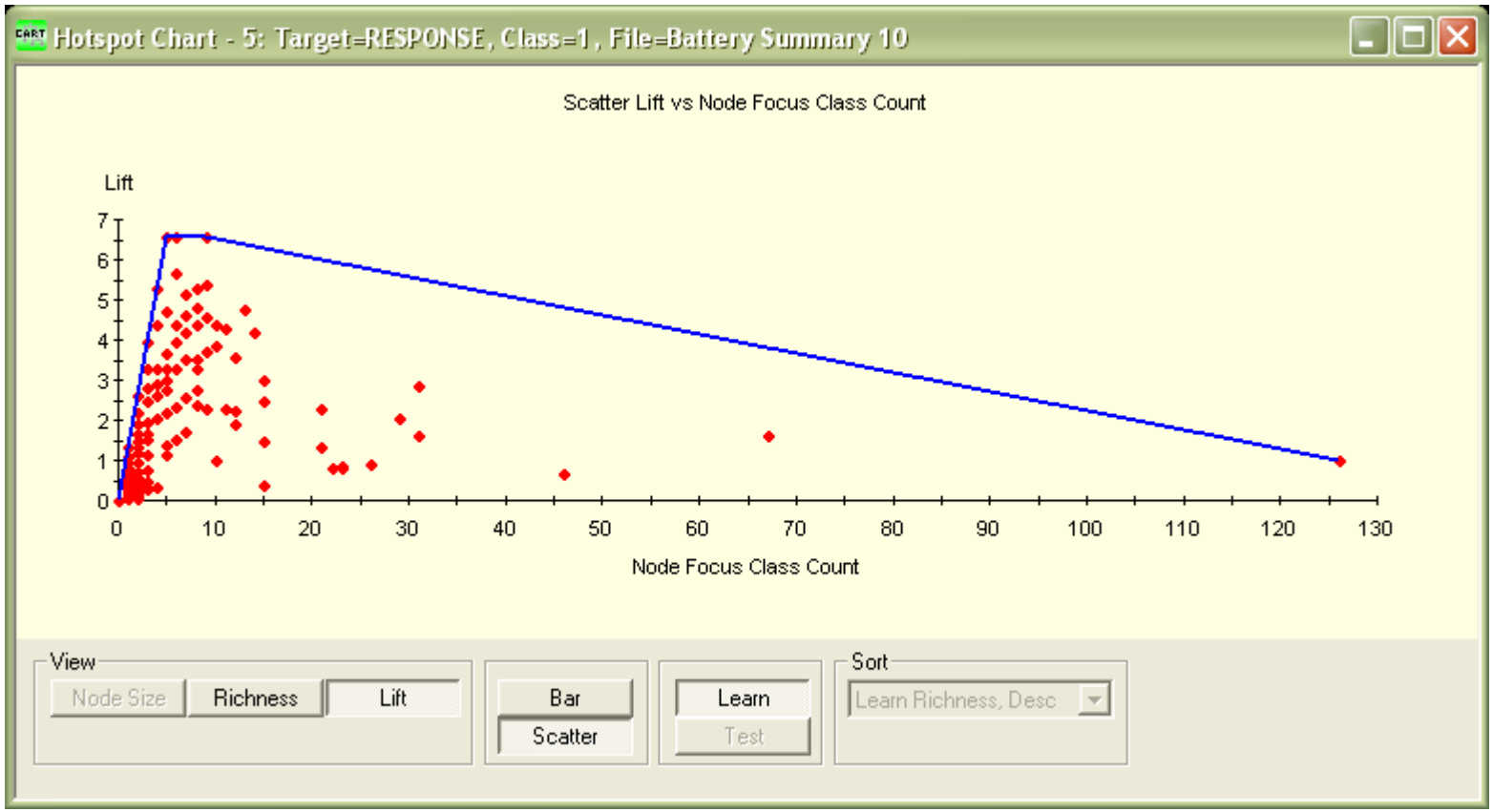
- The battery includes a variety of models with different emphasis on individual classes
- Analyst can choose a model most appropriate for the current set of requirements

Institute of Actuaries of Australia

Backwards Elimination of Features



Hotspot Detection

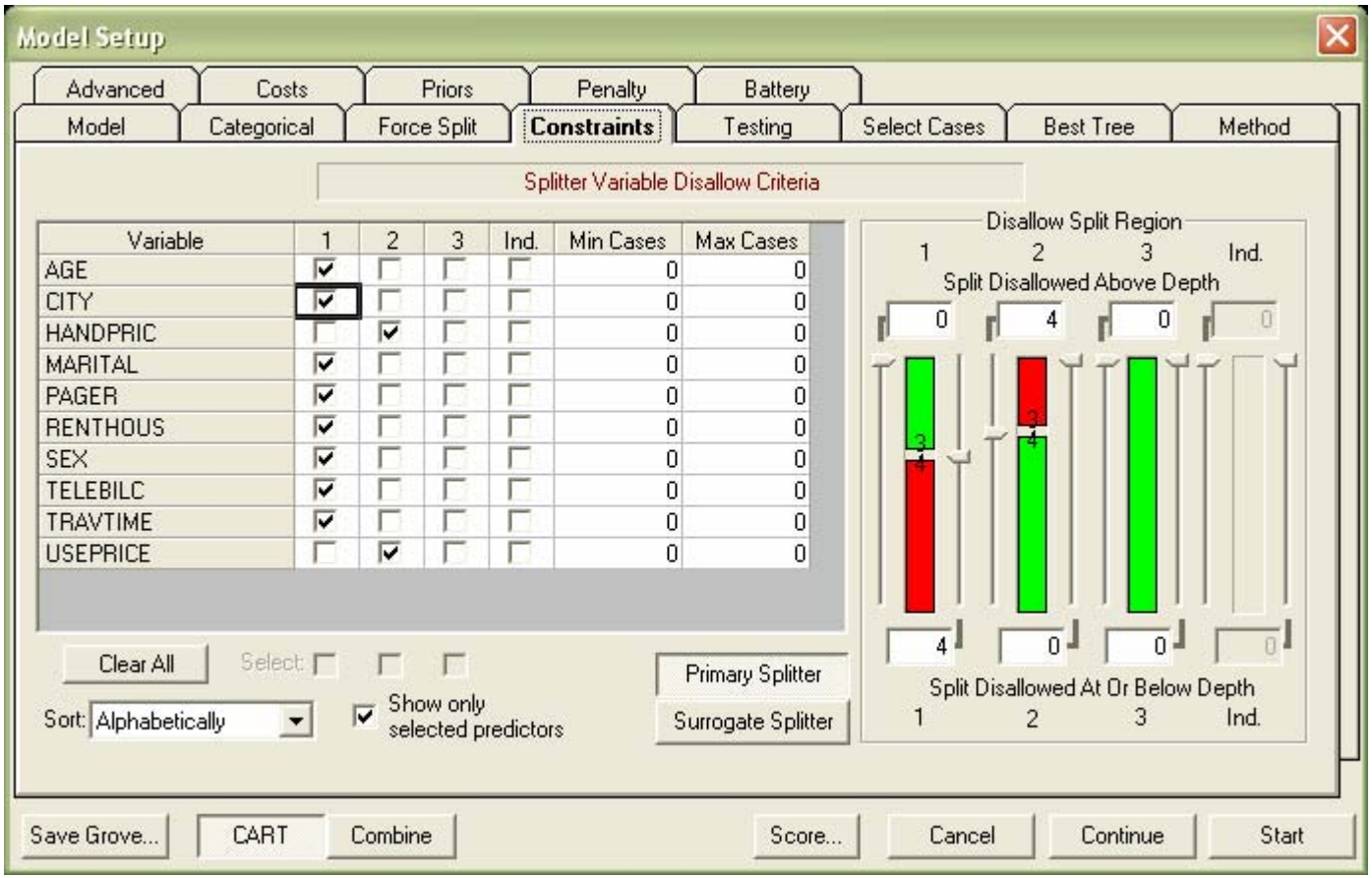


- Lift in node plotted against sample size: Examination of individual nodes from many different trees to find best segments

Constrained Trees

- Many predictive models can benefit from Salford's patent pending "Structured Trees"
- Trees constrained in how they are grown to reflect decision support requirements
- In mobile phone example: want tree to first segment on customer characteristics and then complete using price variables
- Price variables are under the control of the company
- Customer characteristics are not under company control

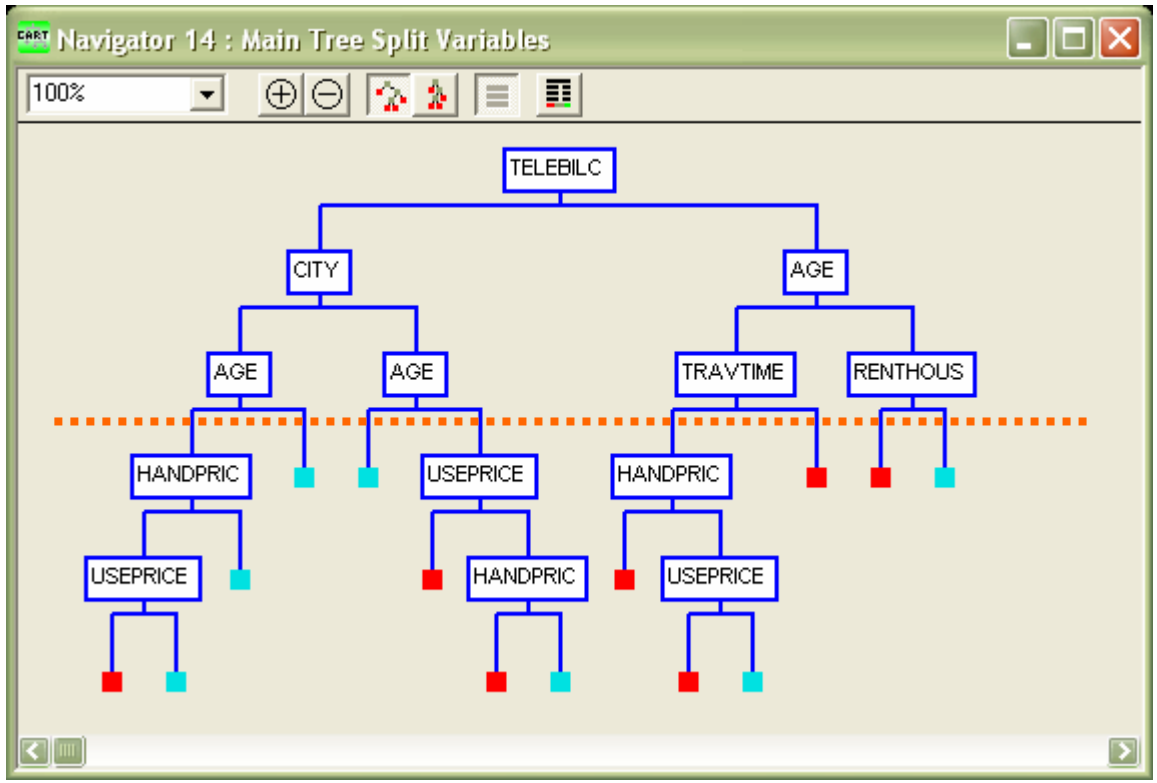
Setting Up Constraints



- Green indicates where in the tree variables of group are allowed to appear

Institute of Actuaries of Australia

Constrained Tree



- Demographic and spend information at top of tree
- Handset (HANDPRIC) and per minute pricing (USEPRICE) at bottom

Translating Tree into Code

Classic Output (Ctrl+Alt+C)

Target Frequency Table
Missing Value Prevalence
Tree Sequence
Node Information
Terminal Nodes
Misclassification
Importance
Option Settings
Competitor List

Tree 14

Target Frequency Table
Missing Value Prevalence
Tree Sequence
Node Information
Terminal Nodes
Misclassification
Importance
Option Settings
Competitor List

Translate Report

SAS Compatible Translation

```
/*************************************************************
 * The following SAS-compatible code was automatically generated
 * by the TRANSLATE feature in the Salford Systems CART(tm)
 * program, version: 6.2.0.142
 *************************************************************/

/* Data Dictionary, Number Of Variables = 8 */
/*    Name = SEX, Type = categorical. */
/*    Name = CITY, Type = categorical. */
/*    Name = TRAVTIME, Type = continuous. */
/*    Name = RENTHOUS, Type = continuous. */
/*    Name = MARITAL, Type = continuous. */
/*    Name = AGE, Type = continuous. */
/*    Name = HANDPRIC, Type = continuous. */
/*    Name = TELEBILC, Type = continuous. */

MODELBEGIN:

/* CART version: 6.2.0.142 */
/* Tree: Tree_1 */
/* Timestamp: 20070418153931010 */
/* Grove: C:\DOCUME~1\Mikhail\LOCALS~1\Temp\s3uc210 */
/* CART Optimal tree, Complexity threshold = 0.00778305 */
/* Target variable: RESPONSE, integer discrete with 2 levels. */
/* N terminal nodes = 5, Depth = 4 */

%let target = predicted_response;
%let node = node;
%let prob = prob;

/* Correspondence between probabilities and     */
/* target class levels.  Probabilities are      */
/* based on weighted learn sample class counts. */
/*    &prob.1: 0 */
/*    &prob.2: 1 */
```

Further Development of CART

- Hybrid models
  - Combining CART with Logistic Regression
  - Combining CART with Neural Nets
- Linear combination splits
- MARS – a better way to handle regression
- Committees of trees
  - Bagging
  - Arcing
  - RF
- Stochastic Gradient Boosting (MART/TreeNet)
- Random Forests

Multivariate Adaptive Regression Splines

- MARS is a highly-automated tool for regression
- Developed by Jerome H. Friedman of Stanford Univ.
  - Annals of Statistics, 1991 dense 65 page article
  - Takes some inspiration from its ancestor CART®
  - Produces smooth curves and surfaces, not the step-functions of CART
- Appropriate target variables are continuous
- Can also perform well on binary dependent variables
  - censored survival model (waiting time models as in churn)

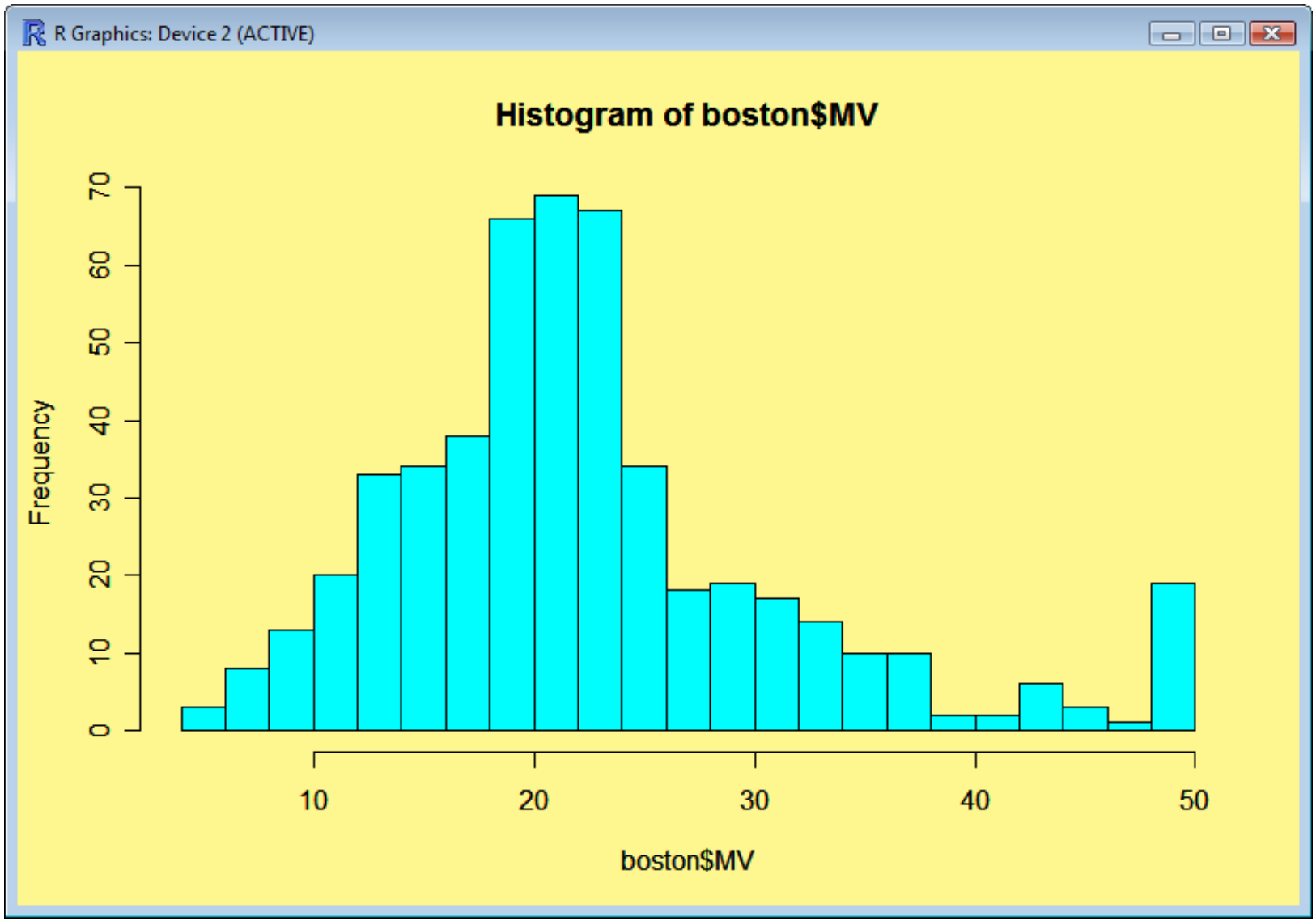
Key Features of MARS

- End result of a MARS run is a regression model
  - MARS automatically chooses which variables to use
  - variables are optimally transformed
  - interactions are detected
  - model is self-tested to protect against over-fitting
- Some formal studies find MARS can outperform Neural Nets

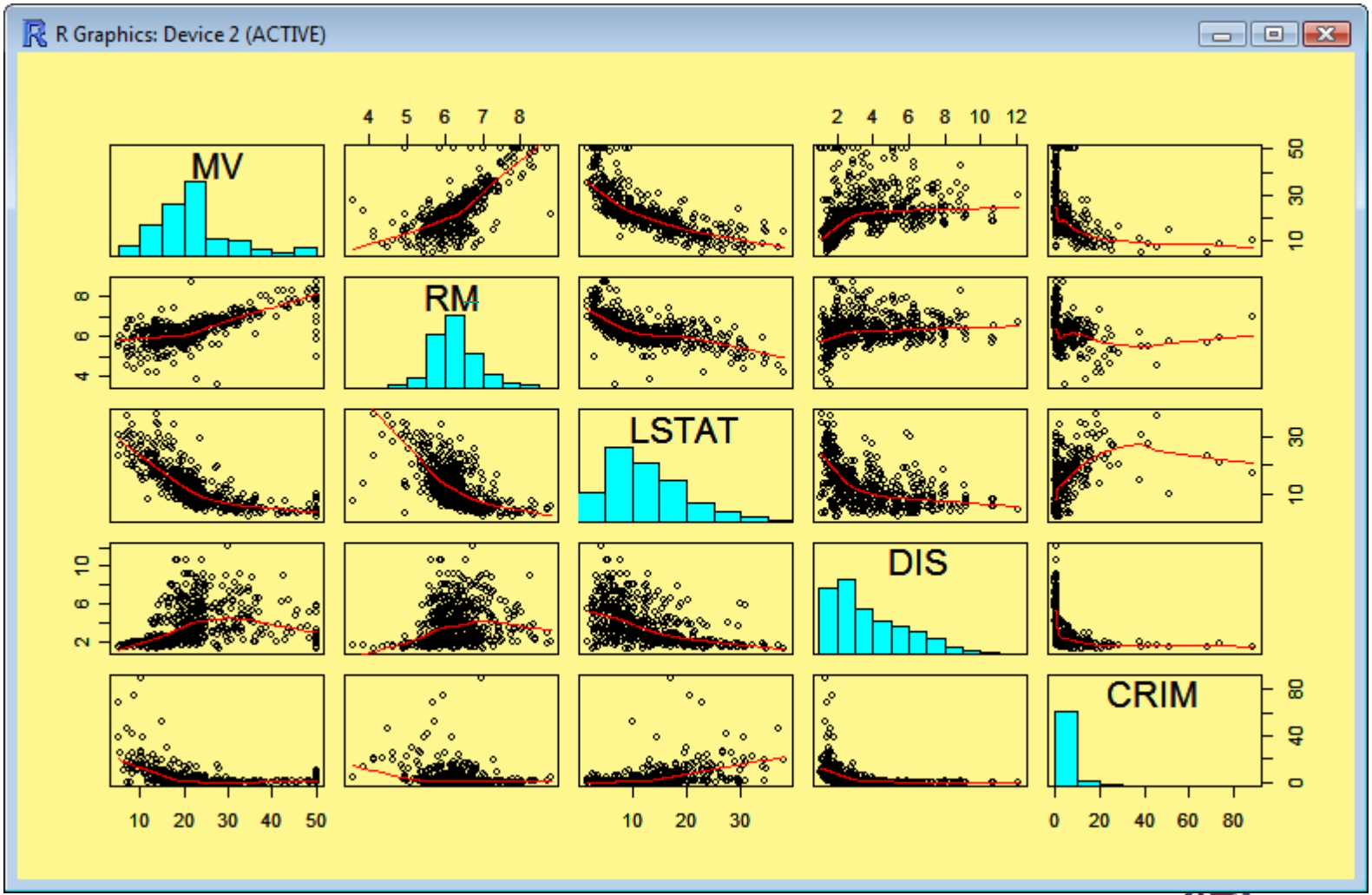
Boston Housing Data Set

- Harrison, D. and D. Rubinfeld.
  Hedonic Housing Prices & Demand For Clean Air. *Journal of Environmental Economics and Management,* v5, 81-102 , 1978
  - 506 census tracts in City of Boston for the year 1970
  - Goal: study relationship between quality of life variables and property values
  - MV median value of owner-occupied homes in tract ('000s)
  - CRIM per capita crime rates
  - NOX concentration of nitrogen oxides (pphm)
  - AGE percent built before 1940
  - DIS weighted distance to centers of employment
  - RM average number of rooms per house
  - LSTAT percent neighborhood 'lower SES'
  - RAD accessibility to radial highways
  - CHAS borders Charles River (0/1)
  - INDUS percent non-retail business
  - TAX tax rate
  - PT pupil teacher ratio

Target: Median House Value (MV)



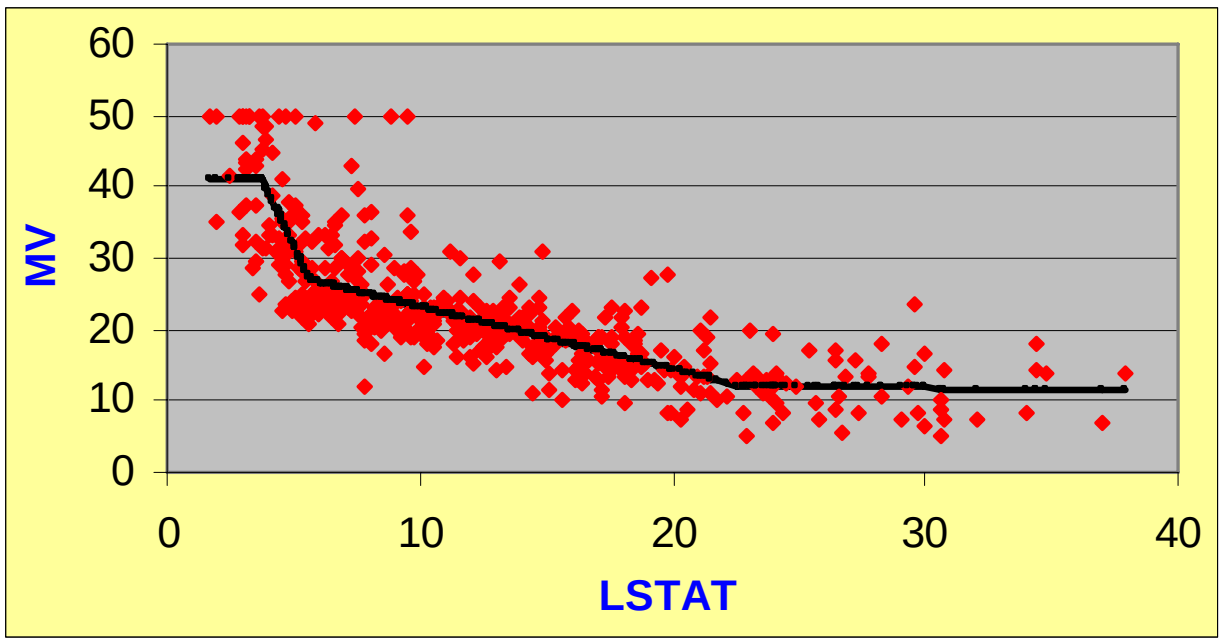
Scatter Matrix



- Clearly some non-normal distributions and non-linear relationships

MARS Model

- Piece-Wise Linear Regression
  - Simplest version of splines
  - Example: MARS spline with 3 knots superimposed on the actual data

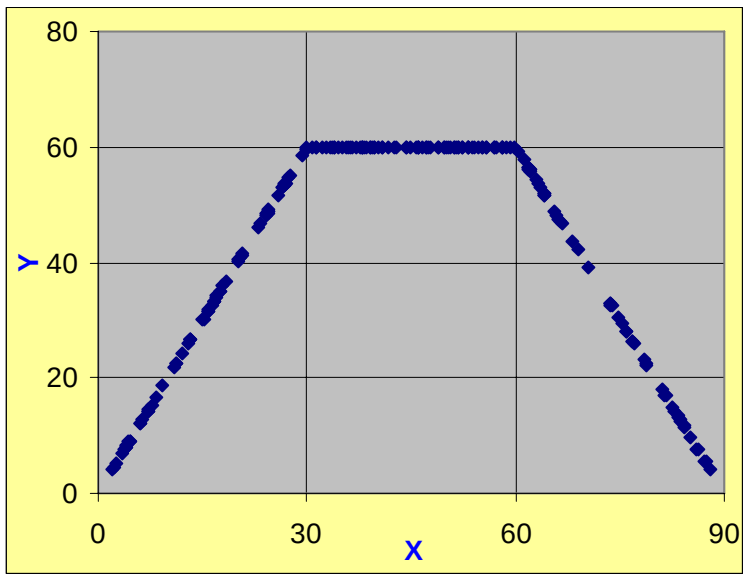


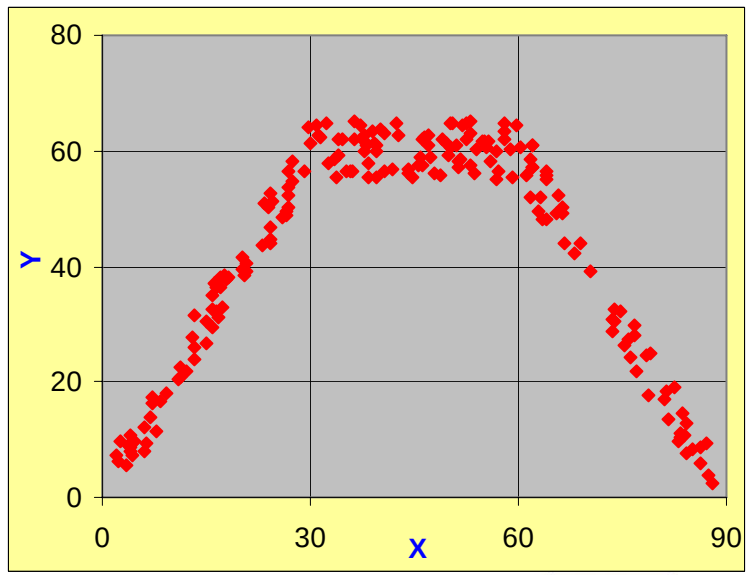
Challenge: Searching for Multiple Knots

- Finding the one best knot in a simple regression is a straightforward search problem
  - try a large number of potential knots and choose one with best R-squared
  - computation can be implemented efficiently using update algorithms; entire regression does not have to be rerun for every possible knot (just update X'X matrices)
- Finding more than two knots simultaneously will require far more computations
- To preserve linear problem complexity, one has to add or remove knots sequentially one at a time

Example: Flat-Top Function

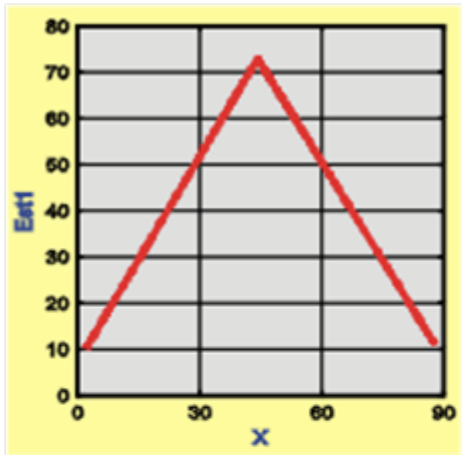
- True function (in graph on left) has two knots at X=30 and X=60
- Observed data at right contains random error
- Best single knot will be at X=45 and MARS finds this first

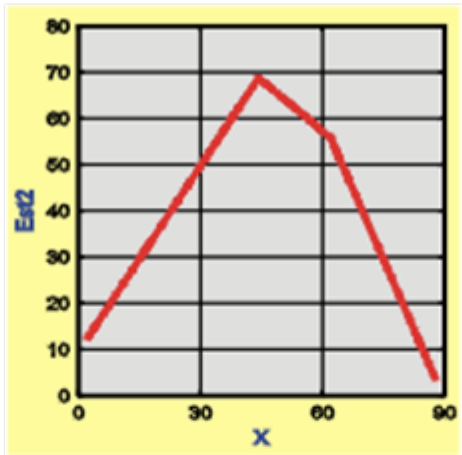


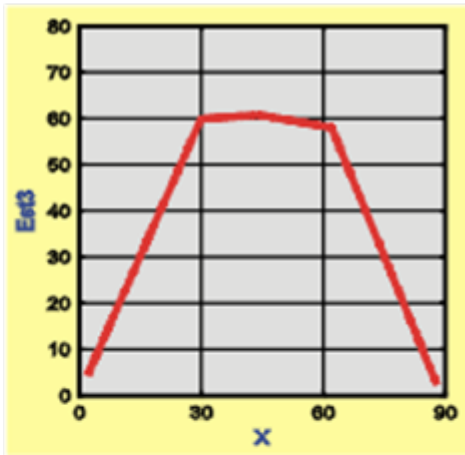


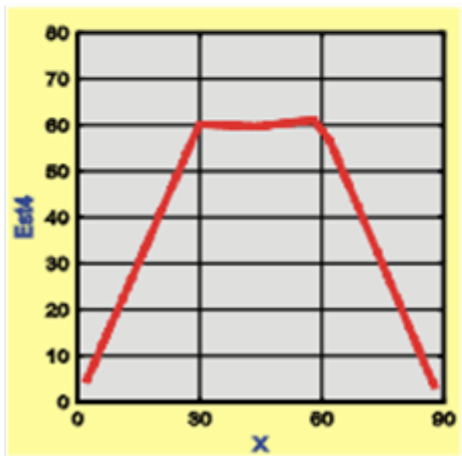
Adding Knots Sequentially

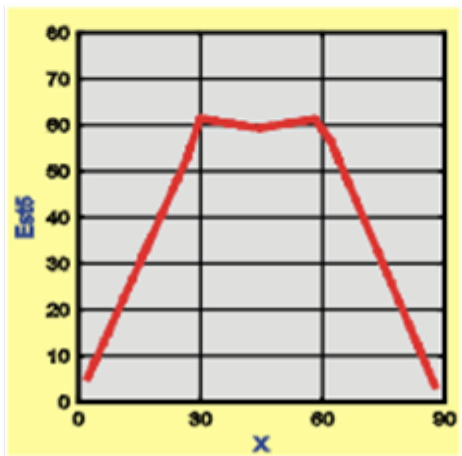
- Start with one knot - then steadily increase number of allowed knots

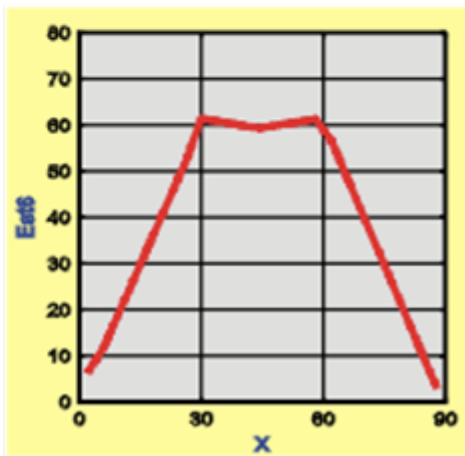












Computational Strategy

- Multiple knot placement is implemented in a step-wise manner
- Need a forward/backward procedure as used in CART
- The forward procedure results to a model that is clearly overfit with too many knots
- The backward procedure removes least contributing knots sequentially
  - Using appropriate statistical criterion remove all knots that add sufficiently little to model quality
- Resulting model will have approximately correct knot locations

Basis Functions

- Thinking in terms of knot selection works very well to illustrate splines in one dimension
- Thinking in terms of knot locations is unwieldy for working with a large number of variables simultaneously
  - need a concise notation and programming expressions that are easy to manipulate
  - Not clear how to construct or represent interactions using knot locations
- Basis Functions (BF) provide analytical machinery to express the knot placement strategy
- MARS creates sets of basis functions to decompose the information in each variable individually

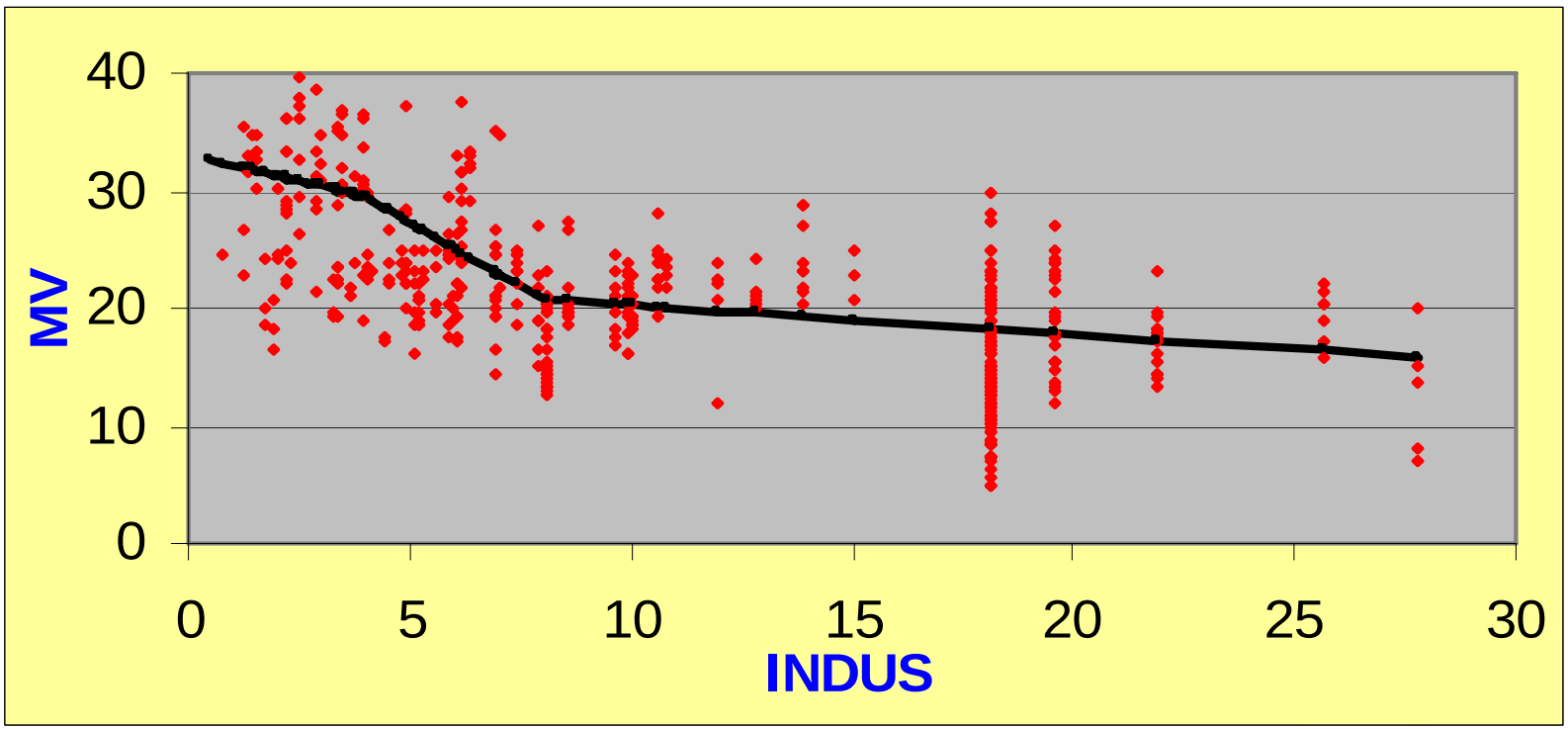
Spline with 3 Basis Functions

- We now have the following basis functions in INDUS

  BF1 = max (0, INDUS - 4) BF2 = max (0, INDUS - 8) BF3=max(0, 4 - INDUS)

  MV= 29.433 + 0.925*$(4 - INDUS)_+$ -2.180*$(INDUS-4)_+$ +1.939*$(INDUS-8)_+$

- All 3 line segments have negative slope even though 2 coefficients above >0



MARS Process

- MARS technology is similar to CART's
  - grow a deliberately overfit model and then prune back
  - core notion is that good model cannot be built from a forward stepping plus stopping rule
  - must overfit generously and then remove unneeded basis functions
- The forward step always adds basis functions in pairs
- The backward step always removes basis functions one at a time
- In practice user specifies an upper limit for number of knots to be generated in forward stage
  - limit should be large enough to ensure that true model can be captured
  - will have to be set by trial and error

Multi Tree Methods

- 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.
  - Eg. relate PD to fraction of trees predicting default for a given

Prediction Via Voting

- *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 (before sampling)*

Sampling with Replacement

- Have a training set of size *N*
- Create a new data set of size *N* by doing sampling with replacement from the training set
- The new set will be different from the original!
  - About a third of the original data will be left out
  - Another third will enter exactly once
  - The other third will enter more than once but unlikely more than 6 times
- May do this repeatedly to generate multiple bootstrap samples

Bootstrap Resampling

- Probability of being omitted in a single draw is (1 - 1/n)
- Probability of being omitted in all n draws is $(1 - 1/n)^n$
- Limit of series as n increases is (1/e) = 0.368
  - approximately 36.5% sample excluded — 0 % of resample
  - 37.5% sample included once — 37.5% of resample
  - 18% sample included twice thus represent ... — 36 % of resample
  - 6% sample included three times ... — 18 % of resample
  - 2% sample included four or more times ... — 8 % of resample

    100 %
  - Example: distribution of weights in a 2,000 record resample:

0.3680	0.3740	0.1840	0.060	0.0140	0.003	0.002
737	749	368	123	28	6	4
0	**1**	**2**	**3**	**4**	**5**	**6**

Institute of Actuaries of Australia

Combining Trees by Voting

- Trees combined via voting (classification) or averaging (regression)
- Classification trees "vote"
  - Recall that classification trees classify
    - assign each case to ONE class only
  - With 50 trees, 50 class assignments for each case
  - Winner is the class with the most votes
  - Votes could be weighted – say by accuracy of individual trees
- Regression trees assign a real predicted value for each case
  - Predictions are combined via averaging
  - Results will be much smoother than from a single tree

Bootstrap Aggregation Performance Gains

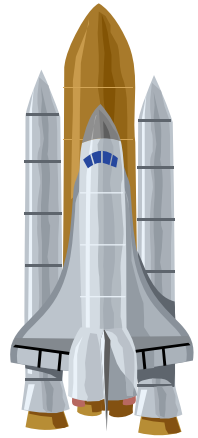
Statlog Data Set Summary

Data Set	# Training	# Variables	# Classes	# Test Set
Letters	15,000	16	26	5,000
Satellite	4,435	36	6	2,000
Shuttle	43,500	9	7	14,500
DNA	2,000	60	3	6,186

Test Set Misclassification Rate (%)

Data Set	1 Tree	Bag	Decrease
Letters	12.6	6.4	49%
Satellite	14.8	10.3	30%
Shuttle	0.062	0.014	77%
DNA	6.2	5.0	19%

Adaptive Resampling and Combining
(ARCing, a Variant of Boosting)



- Bagging proceeds by independent, identically-distributed resampling draws
- Adaptive resampling: probability that a case is sampled varies dynamically
- Starts with all cases having equal probability
- After first tree is grown, weight is increased on all *misclassified* cases
- For regression, weight increases with prediction error for that case
- Idea is to focus tree on those cases most difficult to predict correctly

ARCing Reweights Training Data

- Similar procedure first introduced by Freund & Schapire (1996)
- Breiman variant (ARC-*x*4) is easier to understand:
  - Suppose we have already grown K trees:
    let m_i = # times case i was misclassified ($0 \le m_i \le K$)

    Define $w_i = (1 + m_i^4)$

    Prob (sample inclusion) $\propto w_i$

  - Weight = 1 for cases with zero occurrences of misclassification
  - Weight = $1 + K^4$ for cases with K misclassifications
    - Rapidly becomes large if case is difficult to classify
  - Samples will tend to be increasingly dominated by misclassified cases

Boston Housing Data

Total SUM of Squares => 5,628

	SUM(RES^2)	SUM(ABS(DEV))	R-squared
FIRST TREE ONLY*	2078.620	272.000	63%
BAGGING	837.814	215.580	85%
ARCING**	744.090	214.423	87%

* Single tree now performs worse than stand alone CART run (R-squared=72%) because in bagging we always work with exploratory trees only

** Arcing performance beats MARS additive model but is still inferior to the MARS interactions model

Institute of Actuaries of Australia

Problems with Boosting

- Boosting (and Bagging) are very slow and consume a lot of memory, the final models tend to be awkwardly large and unwieldy
- Boosting in general is vulnerable to overtraining
  - Much better fit on training than on test data
  - Tendency to perform poorly on future data
- Boosting highly vulnerable to errors in the data
  - Technique designed to obsess over errors
  - Will keep trying to “learn” patterns to predict miscoded data
- Documented in study by Dietterich (1998)
    - An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization

Introduction to Random Forests

- New approach for many data analytical tasks developed by Leo Breiman of University of California, Berkeley
  - Co-author of CART® with Friedman, Olshen and Stone
  - Author of Bagging and Arcing approaches to combining trees
- Good for classification and regression problems
  - Also for clustering, density estimation
  - Outlier and anomaly detection
  - Explicit missing value imputation
- Builds on the notions of committees of experts but is substantially different in key implementation details

Random Forest

- A random forest is a collection of single trees grown in a special way
- The overall prediction is determined by voting (in classification) or averaging (in regression)
- The Law of Large Numbers ensures convergence
- The key to accuracy is low correlation and bias
- To keep bias low, trees are grown to maximum depth

The Algorithm

- Each tree is grown on a bootstrap sample from the learning set
- A number *R* is specified (square root by default) such that it is noticeably smaller than the total number of available predictors
- During tree growing phase, at each node only *R* predictors are randomly selected and tried

Performance

- All major advantages of a single tree are automatically preserved
- Since each tree is grown on a bootstrap sample, one can
  - Use out of bag samples to compute an unbiased estimate of the accuracy
  - Use out of bag samples to determine variable importances
- There is no overfitting as the number of trees increases

Further Derivatives

- It is possible to compute generalized proximity between any pair of cases
- Based on proximities one can
  - Proceed with a well-defined clustering solution
  - Detect outliers
  - Generate informative data views/projections using scaling coordinates
  - Do missing value imputation
- Easy expansion into the unsupervised learning domain

Introduction to TreeNet

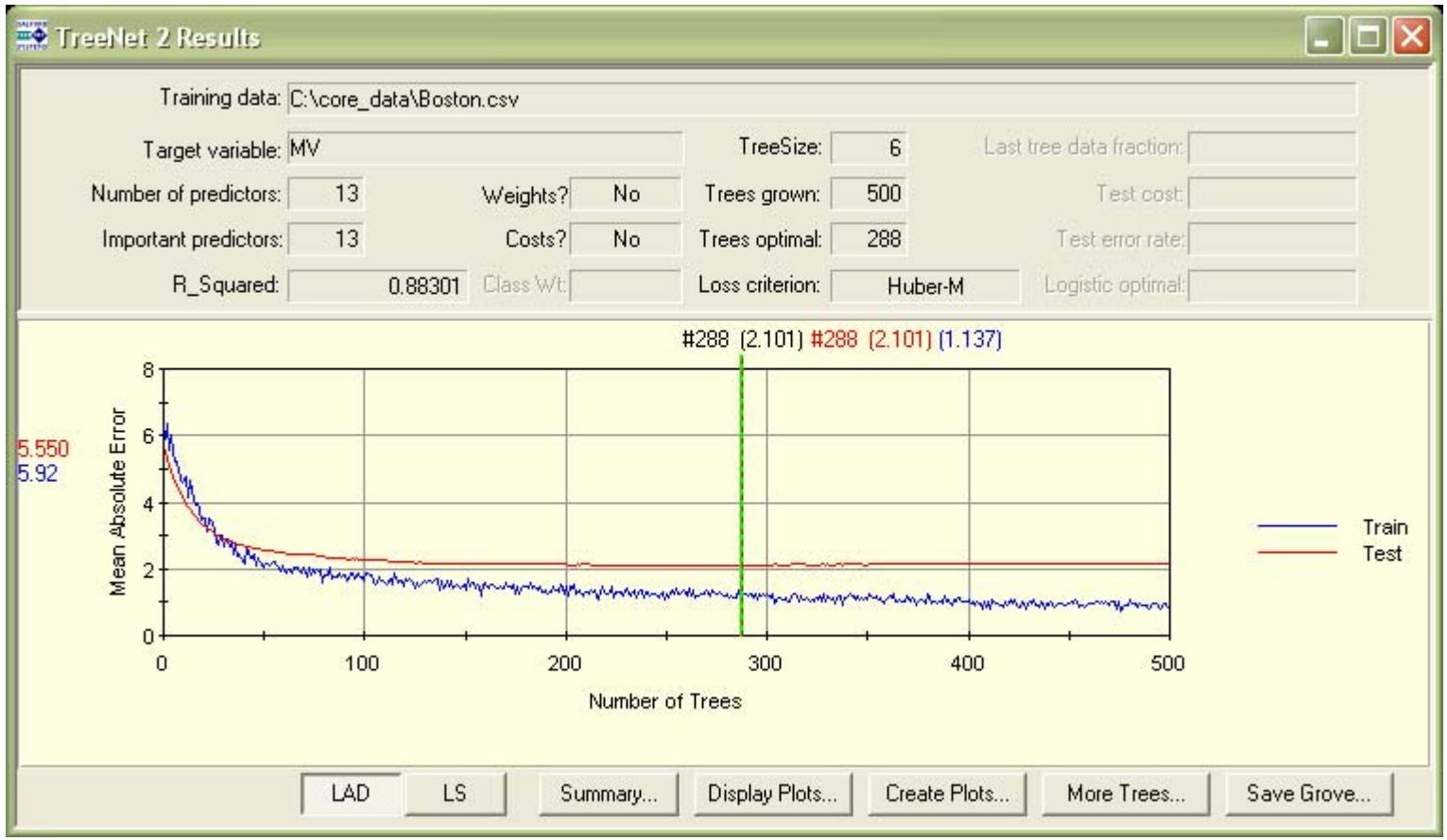
- New approach to machine learning and function approximation developed by Jerome H. Friedman at Stanford University
  - Co-author of CART® with Breiman, Olshen and Stone
  - Author of MARS®, PRIM, Projection Pursuit, COSA, RuleFit™ and more
- Also known as Stochastic Gradient Boosting
- Very strong for classification and regression problems
- Builds on the notions of committees of experts and boosting but is substantially different in key implementation details

Some TreeNet Successes

- 2006 PAKDD competition: customer type discrimination 3rd place
  - Model built in one day. 1st place accuracy 81.9% TreeNet Accuracy 81.2%
- 2005 BI-CUP Sponsored by University of Chile attracted 60 competitors. ‘Most Accurate”
- 2004 KDDCup “Most Accurate”
- 2003 “Duke University/NCR Teradata CRM modeling competition
  - Most Accurate” and “Best Top Decile Lift” on both in and out of time samples
- In use in major financial services and web operations

Institute of Actuaries of Australia

TreeNet Results Display



- Performance on train (blue) and test (red) data as model evolves

Institute of Actuaries of Australia

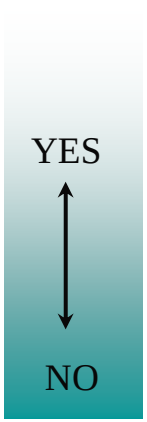
Benefits of TreeNet

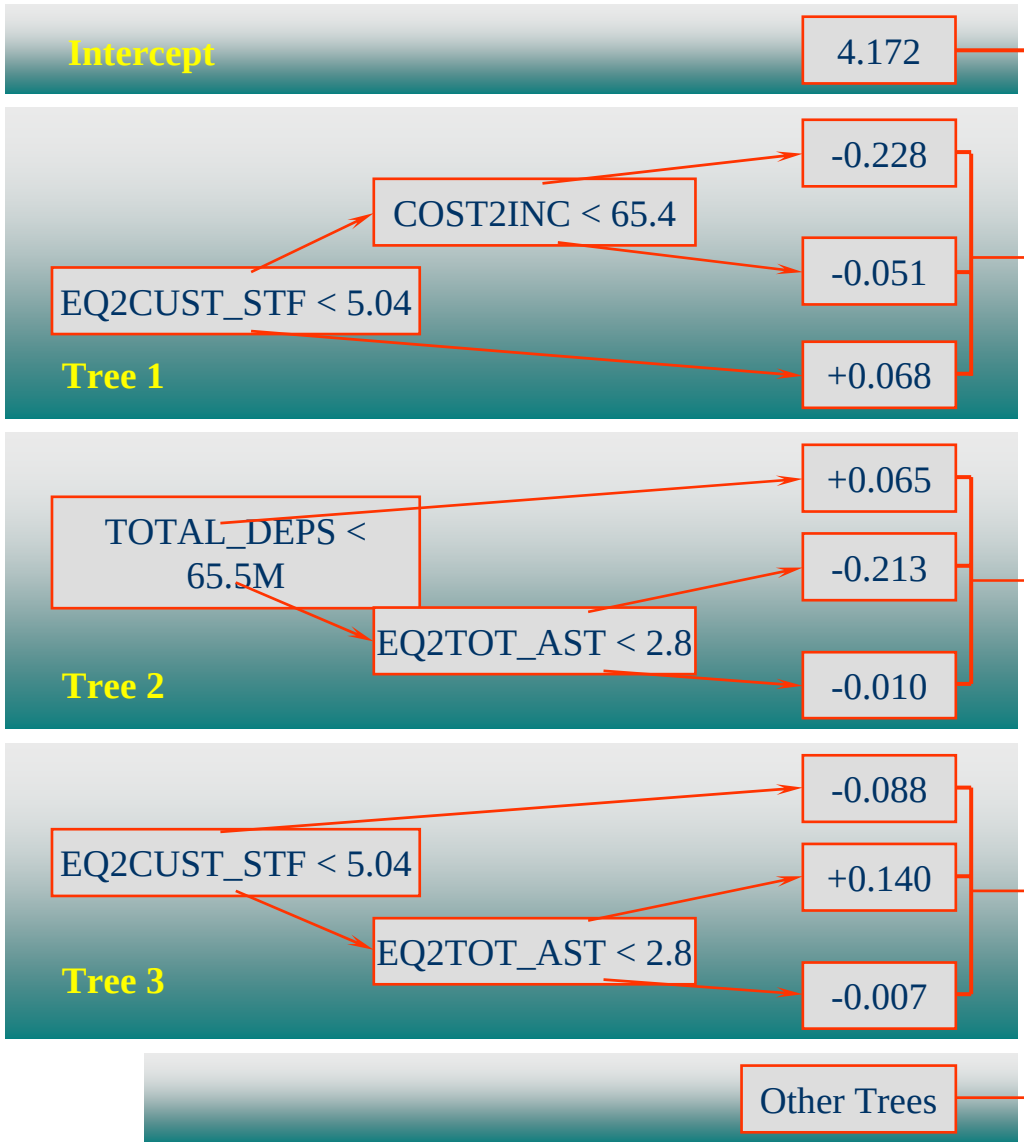
- Built on CART trees and thus
  - immune to outliers
  - handles missing values automatically
  - selects variables,
  - results invariant with monotone transformations of variables
- Resistant to mislabeled target data
  - In medicine cases are commonly misdiagnosed
  - In business, occasionally non-responders flagged as “responders”
- Resistant to over training – generalizes very well
- Can be remarkably accurate with little effort
- Trains very rapidly; comparable to CART

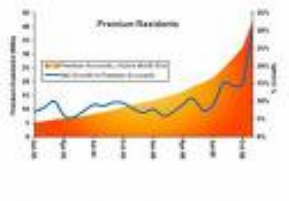
TreeNet Process

- Begin with one 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)
  - Model is intentionally “weak”
- Compute “residuals” for this simple model (prediction error) for every record in data (even for classification model)
- Grow second small tree to predict the residuals from first tree

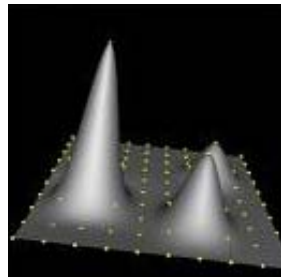
TreeNet: Sample Individual Tree







Key Points



- Trees are kept small (2-6 nodes common)
- Updates are small (downweighted). Update factors can be as small as .01, .001, .0001.
- 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

TreeNet Modeling Options

- Strictly Additive Model (no interactions allowed)
- Low level interactions allowed
- High level interactions allowed
- Constraints: only specific interactions allowed (TN PRO)
- Standard Classification
- Binary non-parametric logistic regression
- Regression: Least Squares, LAD, Huber-M hybrid of LS/LAD
- In development:
  - Survival models, matched sample case control logistic, Poisson

Interpreting TN Models

- As TN models consist of hundreds or even thousands of trees there is no useful way to represent the model via a display of one or two trees
- However, the model can be summarized in a variety of ways
  - Partial dependency plots: These exhibit the relationship between the target and any predictor – as captured by the model.
  - Variable Importance Rankings: These stable rankings give an excellent assessment of the relative importance of predictors
  - ROC curves: TN models produce scores that are typically unique fore ach scored record
  - Confusion Matrix: Using an adjustable score threshold this matrix displays the model false positive and false negative rates.

Partial Dependency Plots

TreeNet 2: Trees = 288 (Optimal): One Variable ...

```
if RM < 3.863 then
   RM_S1 = 0;
   RM_S2 = 0;
   RM_S3 = 0;
else if RM < 5.786 then
   RM_S1 = (RM - 3.863)**2 / 6.93626;
   RM_S2 = 0;
   RM_S3 = 0;
else if RM < 7.47 then
   RM_S1 = 1 - (RM - 7.47)**2 / 6.07419;
   RM_S2 = (RM - 5.786)**2 / 3.51788;
   RM_S3 = 0;
else if RM < 7.875 then
   RM_S1 = 1;
   RM_S2 = 1 - (RM - 7.875)**2 / 0.846046;
   RM_S3 = (RM - 7.47)**2 / 0.53055;
else if RM < 8.78 then
   RM_S1 = 1;
   RM_S2 = 1;
   RM_S3 = 1 - (RM - 8.78)**2 / 1.18555;
else
   RM_S1 = 1;
   RM_S2 = 1;
   RM_S3 = 1;

RM_T = -1.93585 + 0.00696936 * RM_S1 + 11.6229 * RM_S2 + 1.42805 * RM_S3;
```

Save | Submit | Code Format: CART Basic | SAS

le Dependence for MV: RM

One Predictor Dependence For
MV, Penalized SSE = 0.161235

(5.927)

Partial D

0

-5

4 5 6 7 8 9

RM

(5.786) (7.470) (7.875)

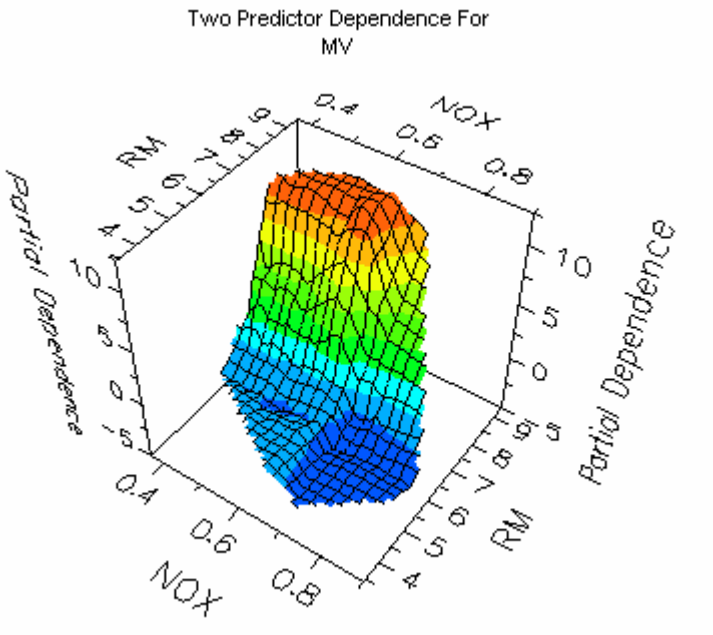
View As: Line | Scatter | Index Plot | Markers

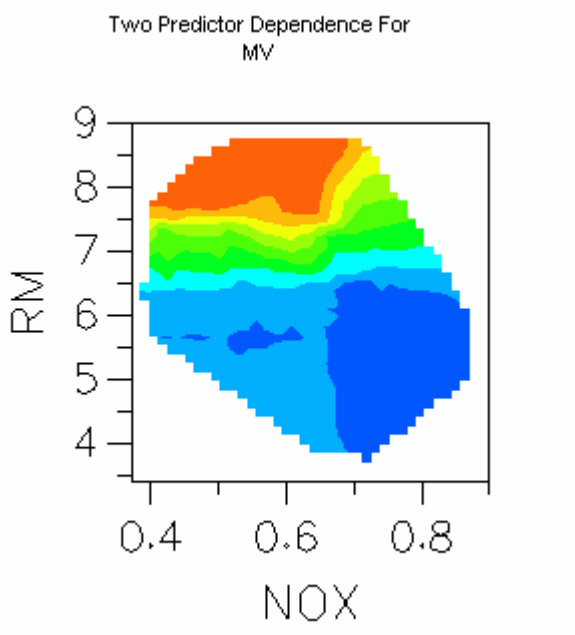
Knots: Select Current | Unselect Current | Reset All | 3 | Optimize

Splines: Monotonic: Up | Resolution: 1 | Smooth rate: 0 | Type: Integral | Standard | First Order | View Model

Truncate: Left, % 0 | Right, % 100 | Apply

Interaction Plots



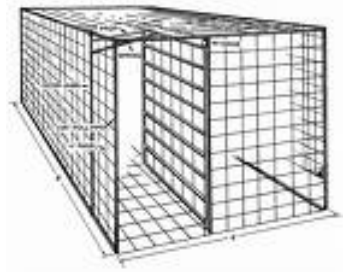


Institute of Actuaries of Australia

Interaction Detection

- TreeNet models based on 2-node trees by definition EXCLUDE interactions
  - Model may be highly nonlinear but is by definition strictly additive
  - Every term in the model is based on a single variable (single split)
- Build TreeNet on larger tree (default is 6 nodes)
  - Permits up to 5-way interaction but in practice is more like 3-way interaction
- Can conduct informal likelihood ratio test TN(2-node) versus TN(6-node)
- Large differences signal important interactions

Case I: Customer Retention

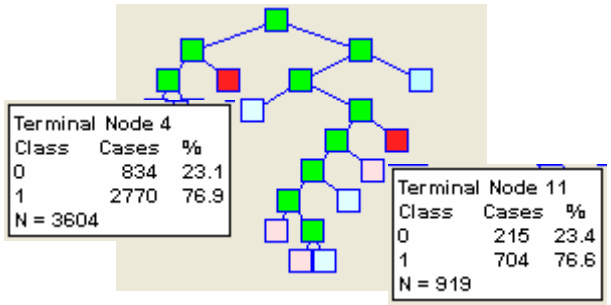


- Have a dataset containing information about last year renewals of insurance policies for the existing customers
  - About 100,000 observations randomly split into 50% learn and 50% test samples
  - Overall renewal rate set at about 50%
- Want to build a segmentation model to identify segments likely to renew

Key Segmentation Variables

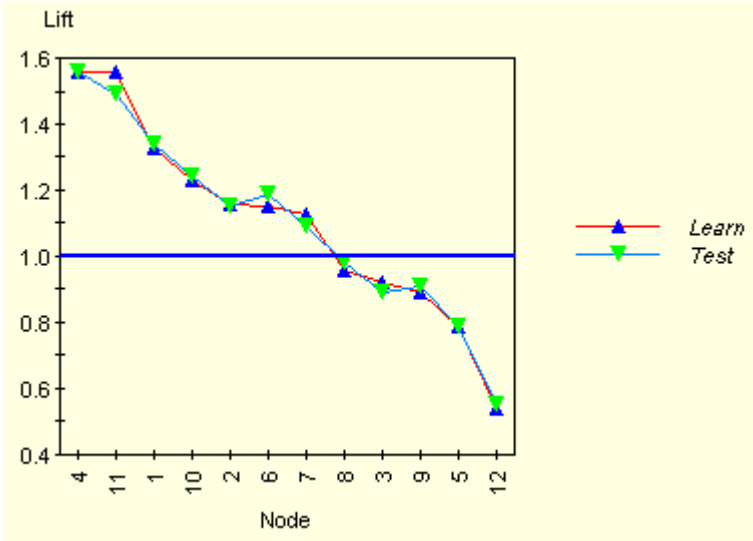
- GENDER – customer's gender
- AREA – customer's area
- POLICY_TYPE – type of policy
- POLICY_AGE – years with the customer
- RESTRICTION – policy restrictions
- AGE – customer age
- POLICYCHANGE – recent change in premium
- MARKETINTENSITY –how competitive the current market is

CART Model Results



Test Sample Prediction Success Table

Actual Class	Total Cases	Percent Correct	Predicted Class 0 N=25220	Predicted Class 1 N=23397
0	24,640	65.64	16,173	8,467
1	23,977	62.27	9,047	14,930
Total:	**48,617.00**			
Average:		63.95		
Overall % Correct:		63.98		



- CART has identified a 12-node tree with good agreement on renewal rates between the learn and the test partitions
- Segments 4 and 11 have the highest renewal rates

Institute of Actuaries of Australia

```
/*Rules for terminal node 4*/

if
(
   MARKETINTENSITY <= 21.5 &&
   POLICY_AGE > 7.5
)

{
   terminalNode = 4;
   class = 1;
   probClass1 = 0.23141;
   probClass2 = 0.76859;
}
```

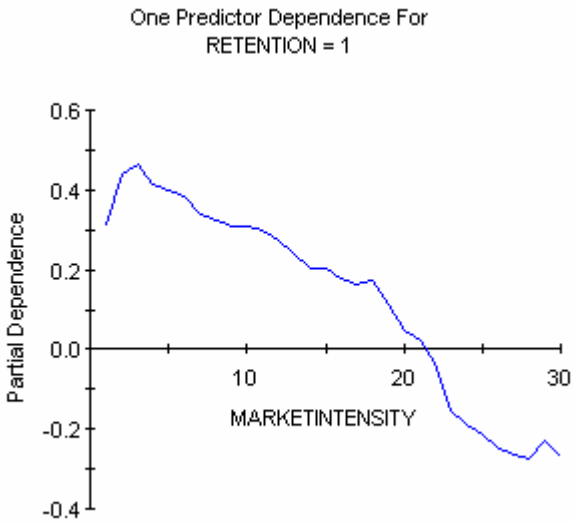
```
/*Rules for terminal node 11*/

if
(
   MARKETINTENSITY > 21.5 &&
   POLICYCHANGE <= 21.5 &&
   POLICY_AGE > 10.5
)

{
   terminalNode = 11;
   class = 1;
   probClass1 = 0.23395;
   probClass2 = 0.76605;
}
```

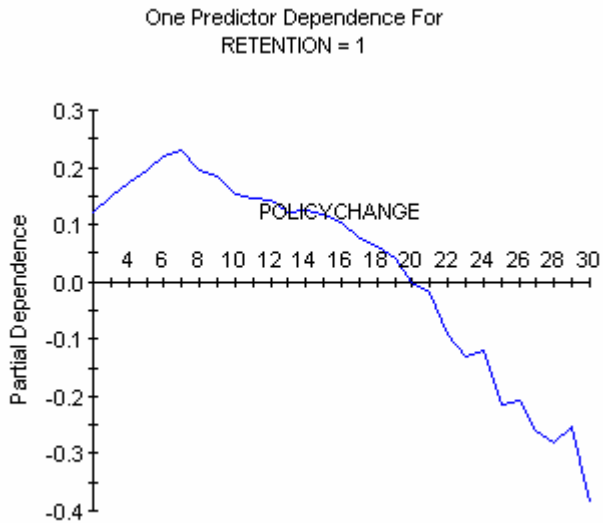
```
/*Rules for terminal node 12*/

if
(
   MARKETINTENSITY > 21.5 &&
   POLICYCHANGE > 21.5
)

{
   terminalNode = 12;
   class = 0;
   probClass1 = 0.732477;
   probClass2 = 0.267523;
}
```

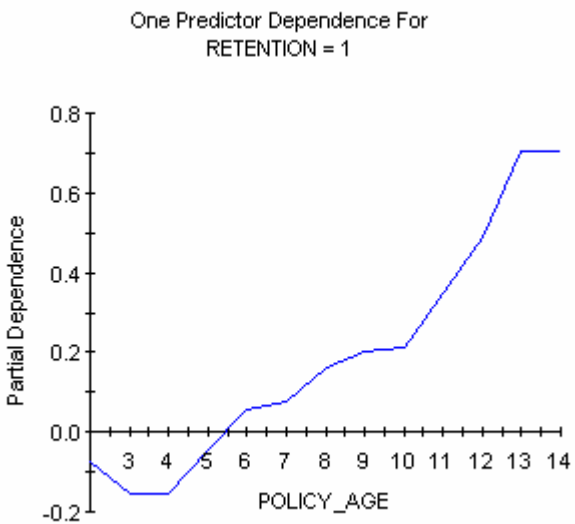
Extreme Segments

- Segment 4: customers with at least 8-year history are likely to renew when market intensity is low
- Segment 11: when market intensity is high only the most loyal customers (11 or more years history) are likely to renew provided that there were no significant premium increase
- Segment 12: when there is a significant premium increase we are likely to loose our customers in the intense market conditions

Refining Models with TreeNet





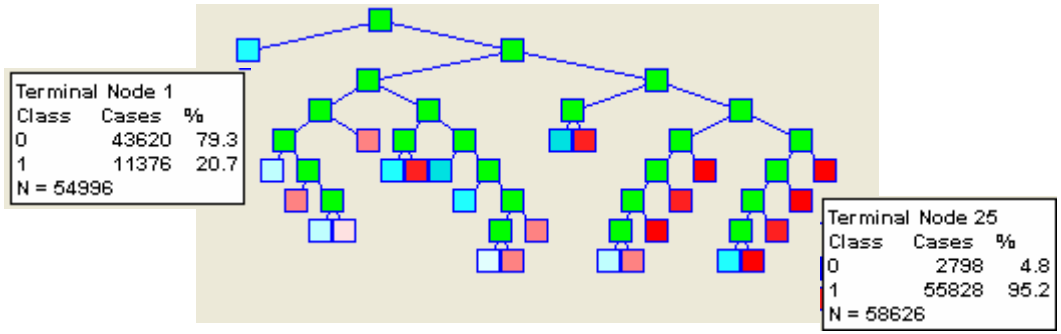


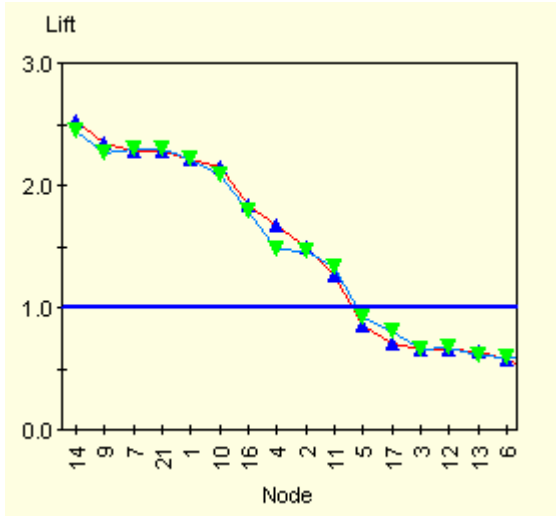
- TreeNet model further confirms CART findings by providing smooth curvilinear contributions for the top three predictors

Case II: Modeling Medical Claims

- Have a history of medical claims on approximately 340,000 customers
- Initial claim rate at 64%
- Use 50% as TEST data
- Limit node sizes to 100

CART Results



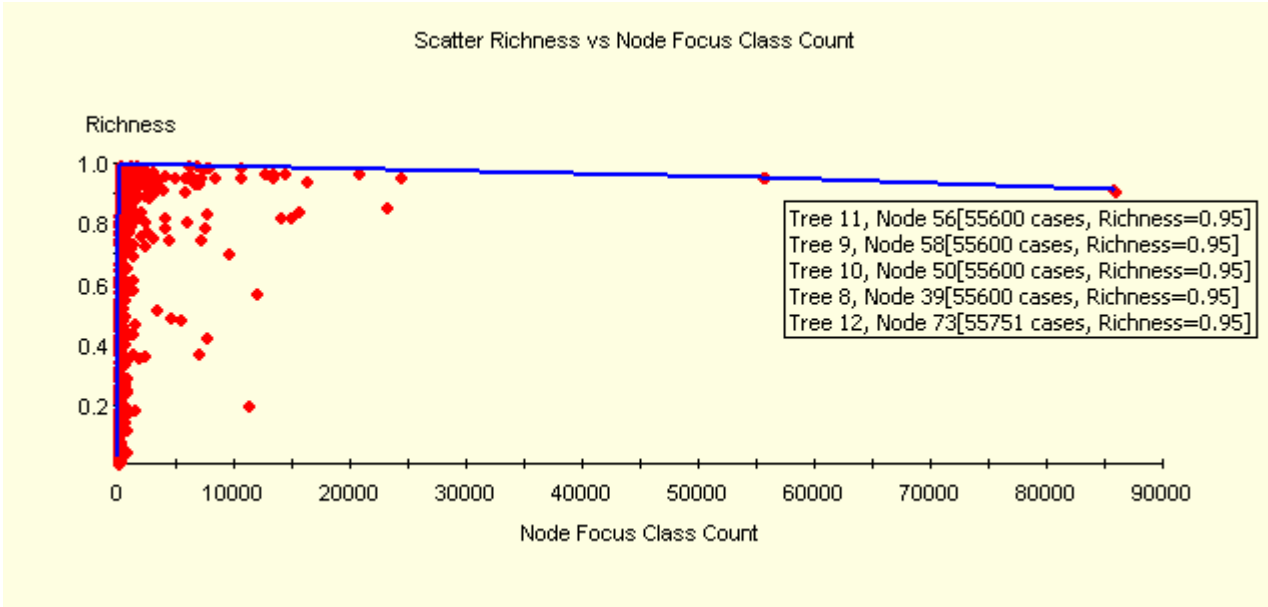


Test Sample Prediction Success Table

Actual Class	Total Cases	Percent Correct	Pred 0 N=66269	Pred 1 N=103585
0	61,223	84.80	51,918	9,305
1	108,631	86.79	14,351	94,280
Total:	**169,854.00**			
Average:		85.80		
Overall % Correct:		86.07		

- CART has identified that most of the data can be partitioned into two segments: the one with low claims rate and the one with high claims rate

Using Battery PRIOR



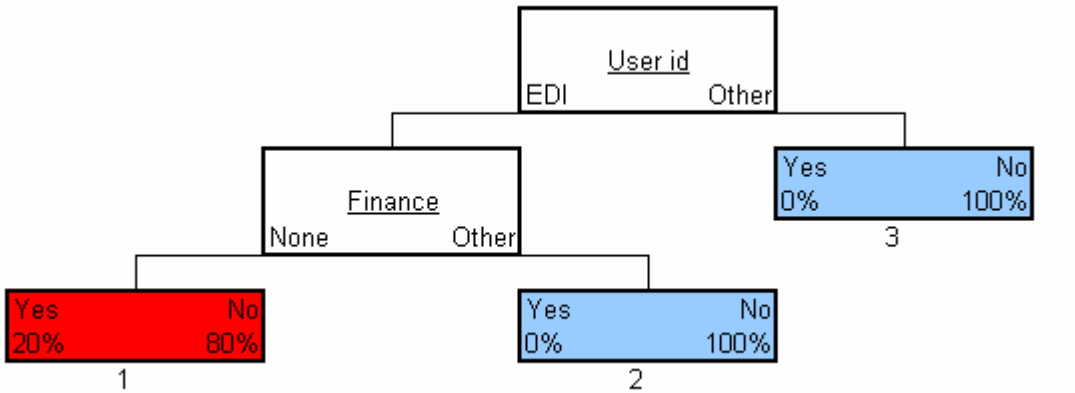
- Battery PRIOR results to a sequence of CART runs with varying degree of richness of the constructed segments
- We can now quickly identify hot-spots and the corresponding rules that lead there

Case III: Dodgy Data

- Try to replicate premiums calculated by a mainframe system – only 95% accuracy is achieved
- What's wrong with the remaining 5%?
- Want to find possible error in the program responsible for computing premiums
- Will run CART on a dataset where the error records are marked as "Yes" class

Dodgy Data Example - Solved

- CART has produced the following solution within minutes



- Some EDI transactions are erroneously being set to Finance Code “N” at the stage of loading into the mainframe
- The entire problem is solved in 30 minutes!

Case IV: Price Change

- When looking at the distribution of prices in the portfolio, a small number of extreme outliers getting large increases is detected
- Assigning all outliers into “Yes” group and running CART results to a single-split tree
- Per CART’s finding, one of the rate tables still contains added dummy premium that by pure accident was not removed
- The whole process takes 5 minutes!

Case V: Who Gets Big Decrease?

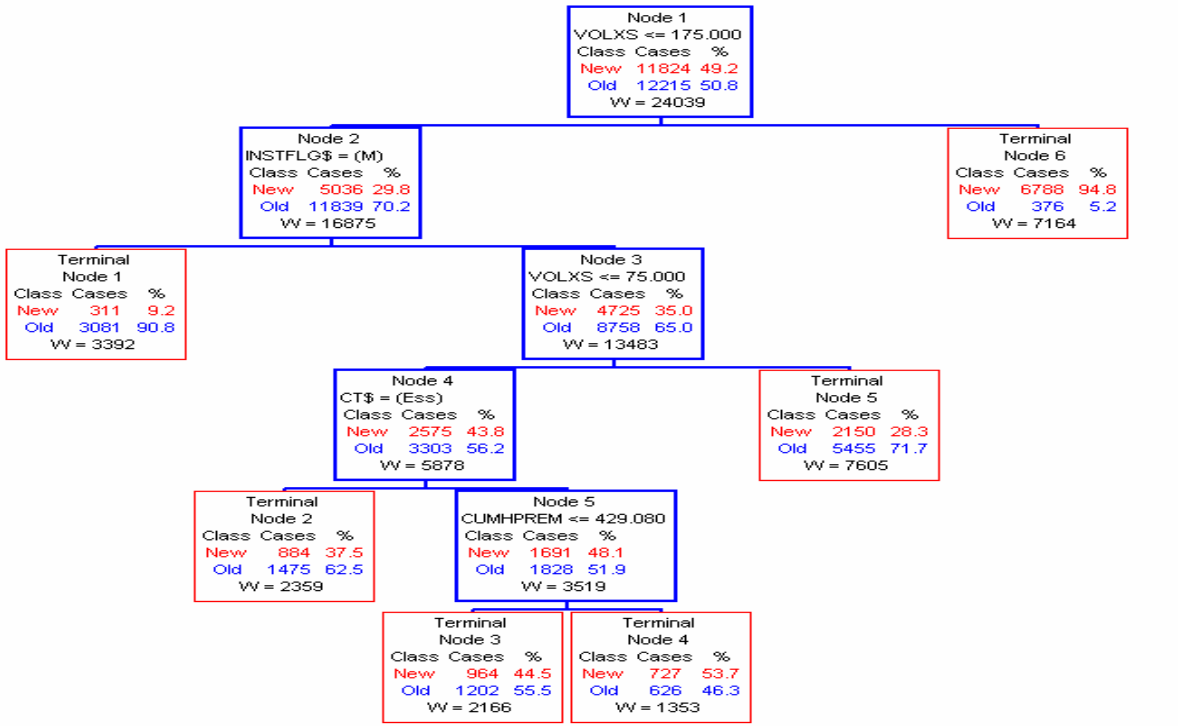
- A group of customers is getting a large decrease on the proposed rates
- The actual rating system is very complicated and hard to track – it is not easy to tell who belongs to this group
- CART builds a simple tree to present a story that could be used to present things to management

- All of these seemingly trivial examples would have taken a lot more time using conventional analysis tools

Case VI: Tracking Portfolio Change

- Interested how portfolio changed before and after a price change
- Extracted several months of new businesses before and after the price change, two separate groups were classified as “Yes” and “No”
- Run CART to detect major differences between the two groups

CART Solution (Home Insurance)

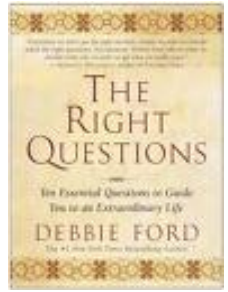


- The tree reflects the fact that the quoted by default excess rate has changed
- A perfect example of the actuarial quality control cycle

Case VII: Missing Value Analysis

- CART uses powerful mechanism of surrogates to deal with missing values
- May also introduce missing value indicators that mark all cases where a given variable is missing
- Now run sequence of analyses declaring one missing value indicator as a target variable – this often reveals very interesting structure behind missingness
- This process a generally much cheaper than direct inquiry into operations and data collection process
- CART short-circuits the path to understanding the data

Case VIII: Checking Assumptions



- We often make or have assumptions about the data, for example:
- When excluding $0 claims from analysis one might assume that these claims are randomly spread – CART can quickly verify this
- Assign $0 claims as “Yes” and the rest as “No”
- When the assumption is valid, expect to have no tree
- Otherwise, the tree will pinpoint exactly how and where the assumption is violated
- This process oftentimes results to unexpected and striking discoveries

References

- Breiman, L. (1996). Bagging predictors. Machine Learning, 24, 123-140.
- Breiman, L., Friedman, J., Olshen, L. and Stone, C. (1984)
  Classification and Regression Trees. Wadsworth. Reprinted by CRC press.
- Friedman, J. H. (1991a). Multivariate adaptive regression splines (with discussion). Annals of Statistics, 19, 1-141 (March).
- 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.