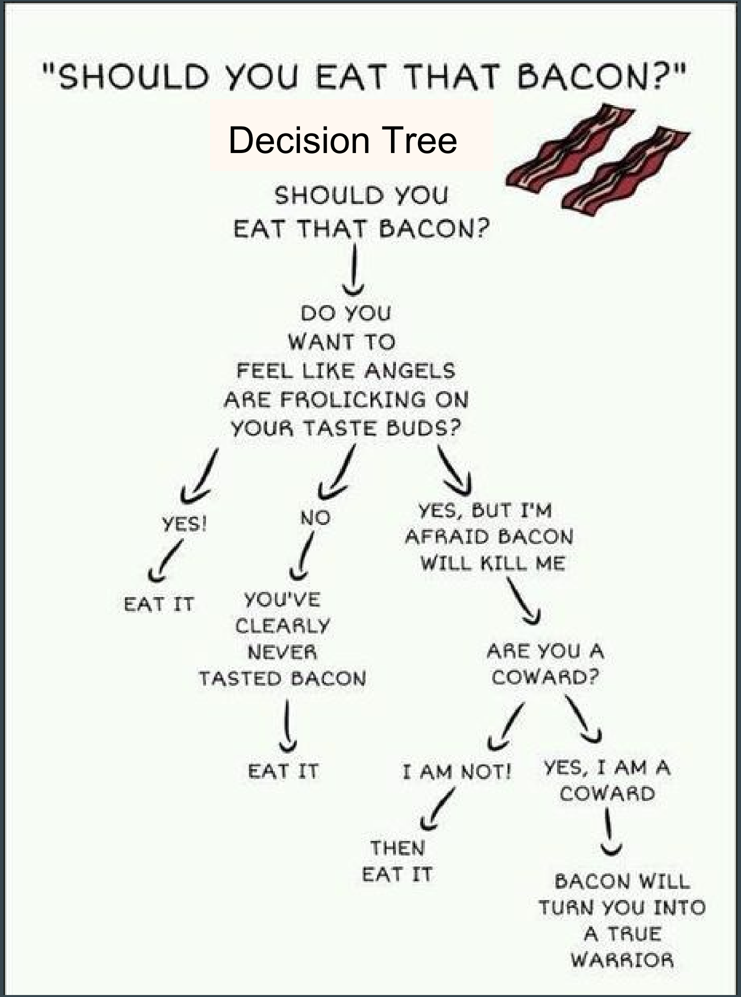
Tree Based Methods

•••

Machine Learning II (2017)
Team 3

Agenda

1. Basics of Trees
2. Regression Trees
3. Pruning
4. R Example


"SHOULD YOU EAT THAT BACON?"
SHOULD YOU EAT THAT BACON?
DO YOU WANT TO FEEL LIKE ANGELS ARE FROLICKING ON YOUR TASTE BUDS?
YES!
EAT IT
NO
YOU'VE CLEARLY NEVER TASTED BACON
EAT IT
YES, BUT I'M AFRAID BACON WILL KILL ME
ARE YOU A COWARD?
I AM NOT!
THEN EAT IT
YES, I AM A COWARD
BACON WILL TURN YOU INTO A TRUE WARRIOR

Decision Tree

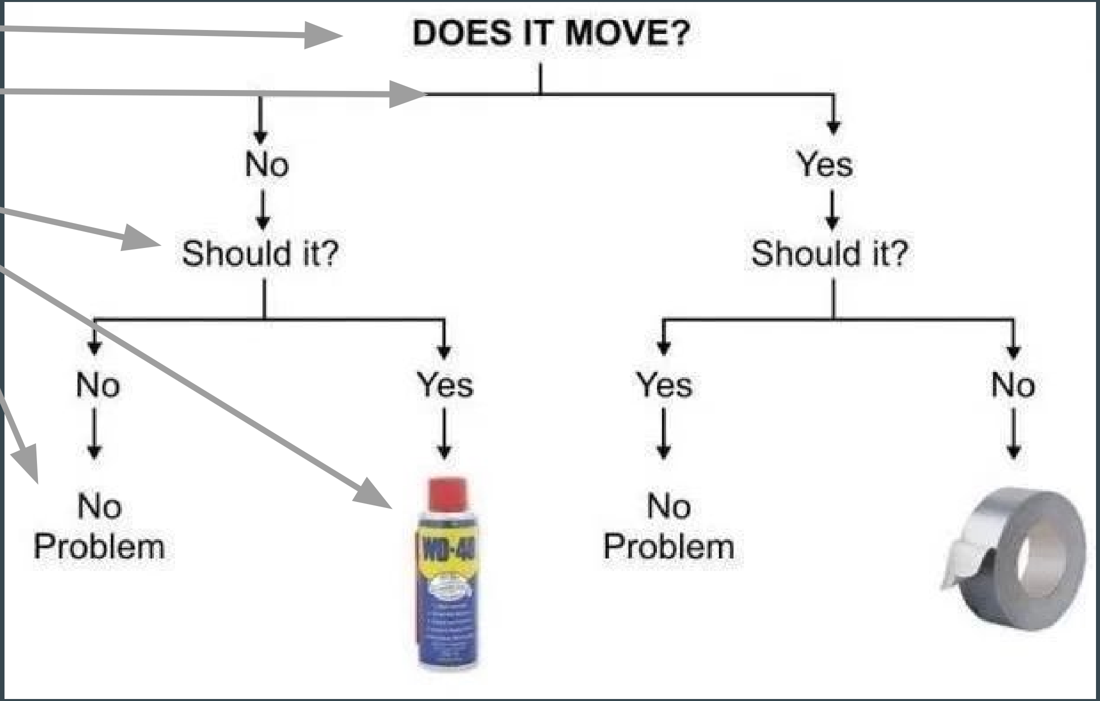
Source: http://data.iwastesomuchtime.com/April-10-2012-16-45-32-525662101507246461831181732071781179451664172746364 3n.jpg

Basics of Trees

- Classification and Regression Trees are the most widely used Machine Learning-based data mining model
- Trees use the supervised learning method
- Moving into a non-Parametric approach, no guess about the shape ahead of time
- Stratifying or segmenting the predictor space into a number of simple regions
- Simple, easy to use
- Humans understand and can easily interpret results vs. other regression based models

Anatomy of a Tree

- Root node
- Branches
- Internal nodes
- Terminal nodes (leaves)



Source: https://eight2late.files.wordpress.com/2016/02/7214525854_733237dd83_z1.jpg?w=700

Problems with Trees

- Sacrifice accuracy for ease of interpretation
- Only one mean prediction/response for each region (terminal node)
- Sometimes utilizing trees with many terminal nodes leads to overfitting

Example Solutions

- Fraud detection
- Gender classification based only on first name
- Titanic survival

Creating a Regression Tree

- There are infinite possible trees as each variable can be used at any node and branching can involve splitting the any x variable at any point. Variables can also be used repeatedly for multiple different splits
    - Therefore the goal is to find a model that is good enough, but perhaps not the best, in a reasonable amount of time
    - The goal for each root and internal node is to find a variable that can split the dataset into 2 Regions that are more homogeneous than the observations inputted into the node
    - The stopping point can be when all samples from a node are the same, when further splitting does not lead to a more homogeneous model or when a maximum number of nodes are reached

Creating a Tree

- Divide all the X values into distinct Regions, represented by R_j
- All observations in a region have the same prediction (the mean of the Y values for the training observations that fell in that region)
- To identify the regions, the X values are split into boxes - ideally boxes with the lowest possible RSS - given by:

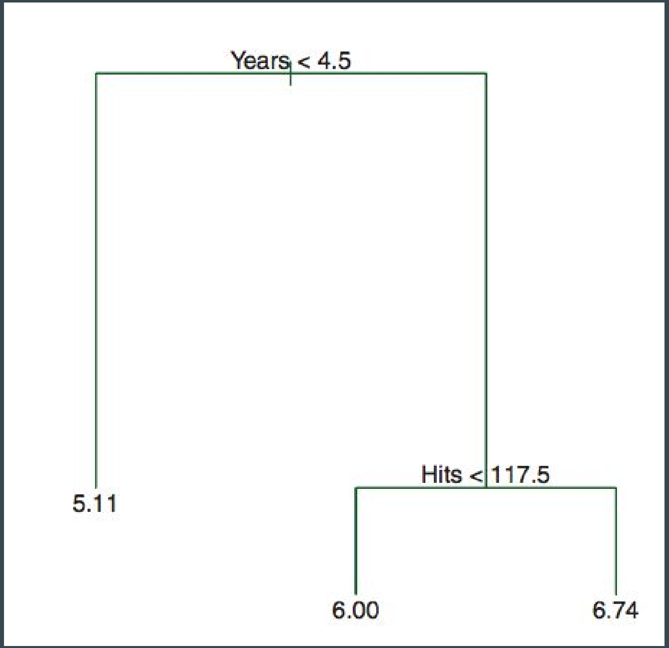
$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

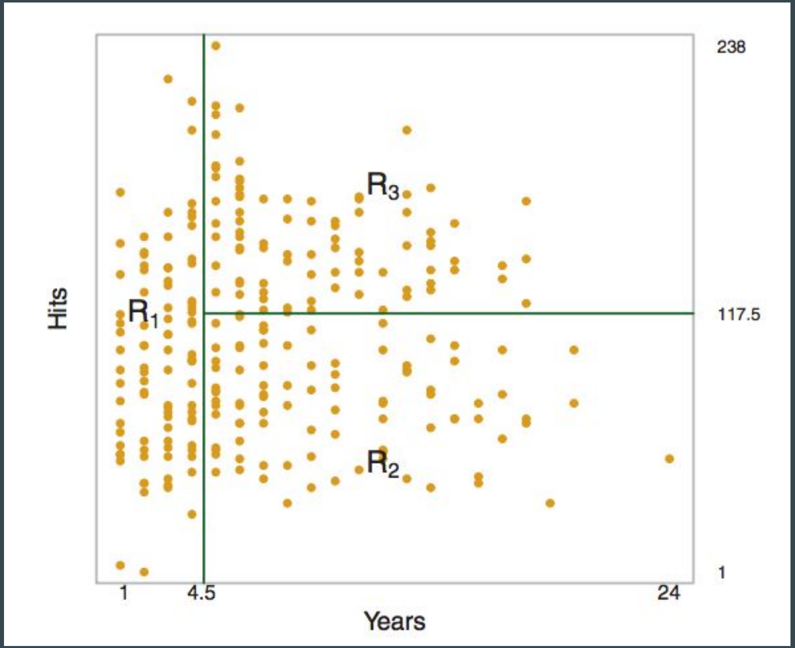
mean response for the training observations within the jth box

Shortcut

- As it is infeasible to consider every box configuration, Recursive Binary Splitting/Top Down Greedy approach is used
- This method starts at the top of the tree, then successively splits the tree into two new branches
- It is greedy as it makes the best split at each level rather than attempting to predict future steps
- Selects a predictor X and the cutoff s such that splitting the predictor space into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ leads to the greatest possible reduction in RSS
- Repeat selecting predictors and splits until end criterion (num of nodes)

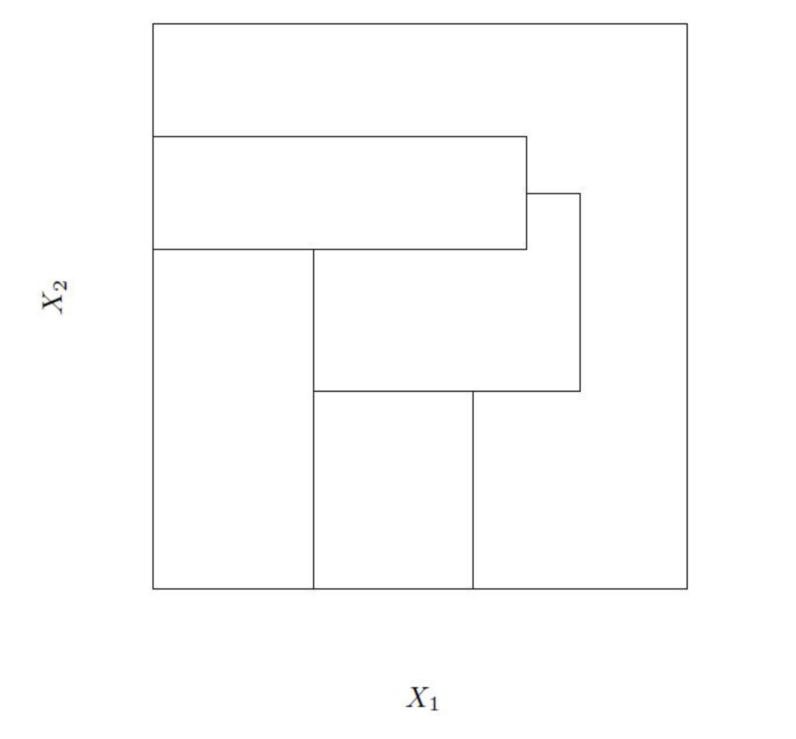
Basic Hitters Data Example





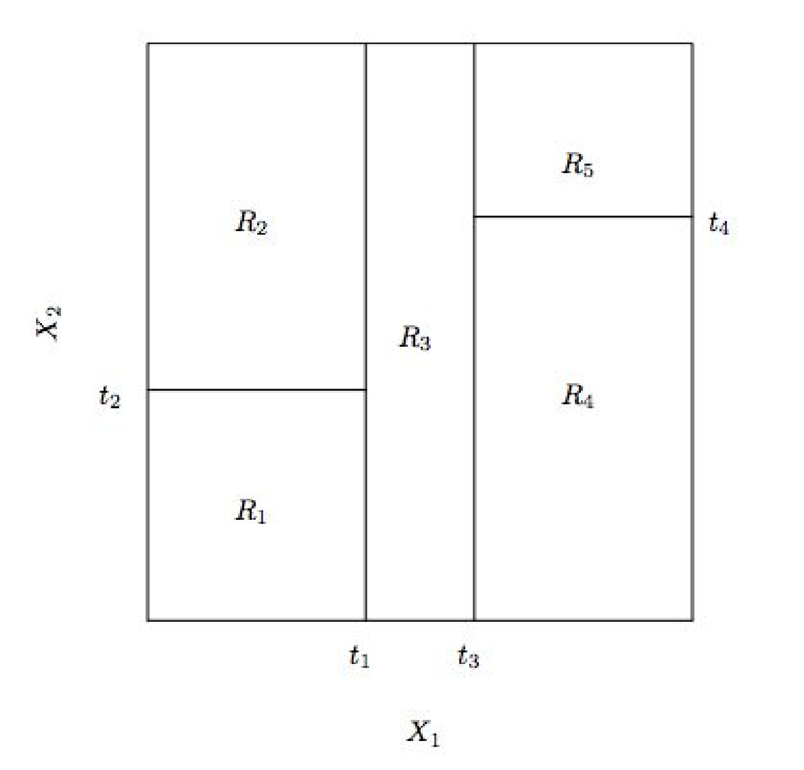
Recursive Binary Splitting Example

- Could NOT result from recursive binary splitting

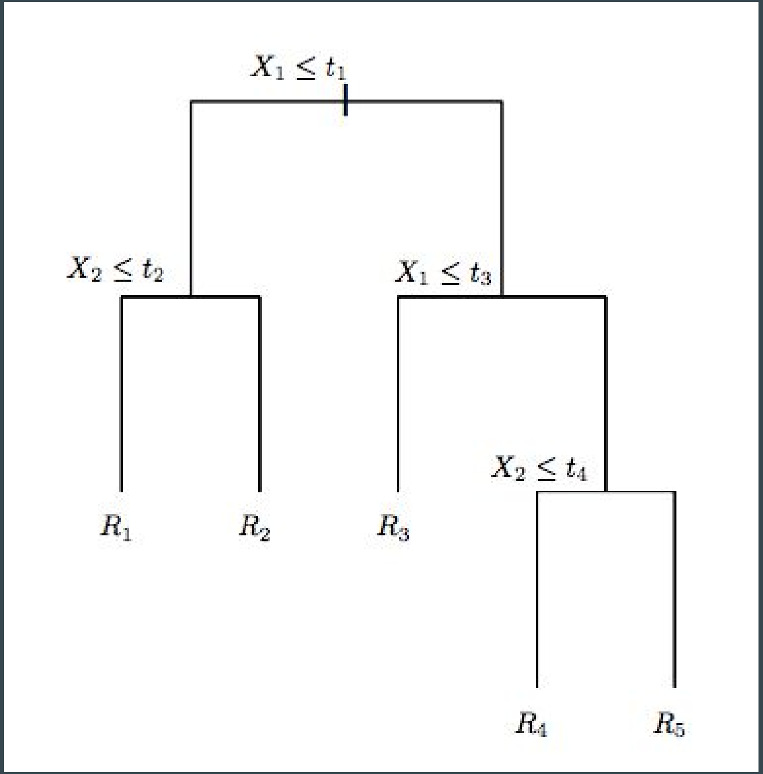


A partition of two-dimensional feature space

Recursive Binary Splitting Example

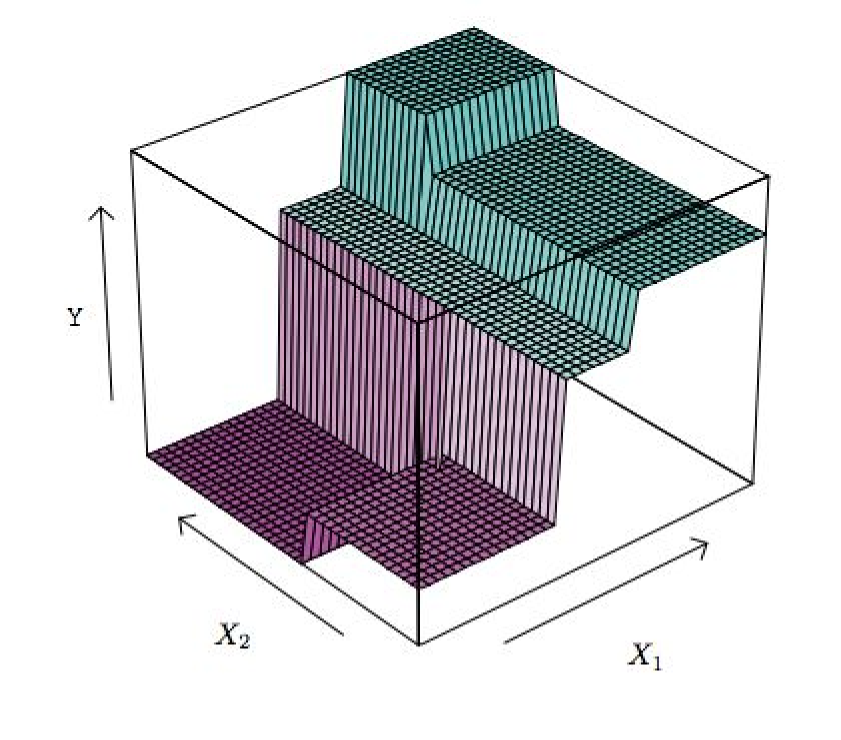


The output of recursive binary splitting



A tree corresponding to the partition

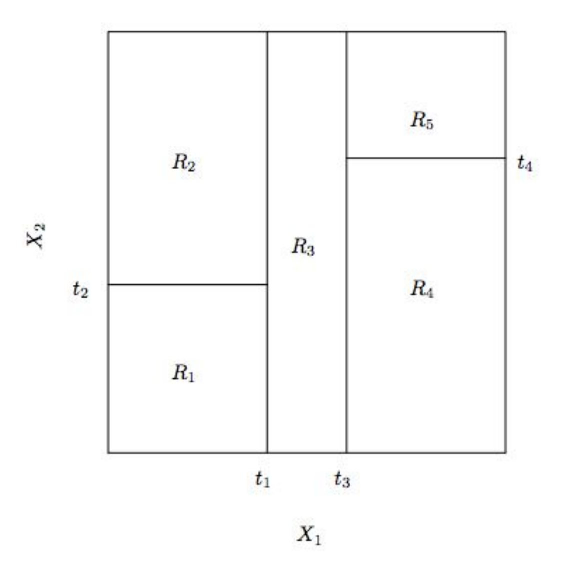
Recursive Binary Splitting Example



A perspective plot of the prediction surface corresponding

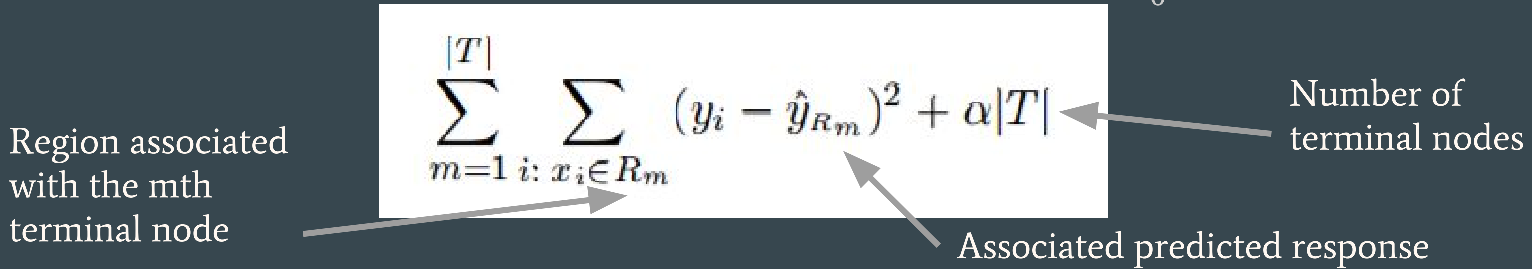
Problems with Recursive Binary Splitting

- Tree might be too complex (overfitting)
- High error for test set data
- Tradeoff of bias and variance



Cost Complexity/ Weakest Link Pruning

- Tradeoff between complexity and model fit (test error rate), pruning reduces variance at the cost of bias
- After creating a large tree (T_0), prune it back into a subtree (T)
- Sequence of trees indicated by nonnegative tuning parameter **α**
- For each value of α there corresponds a subtree $T \subset T_0$ such that



$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

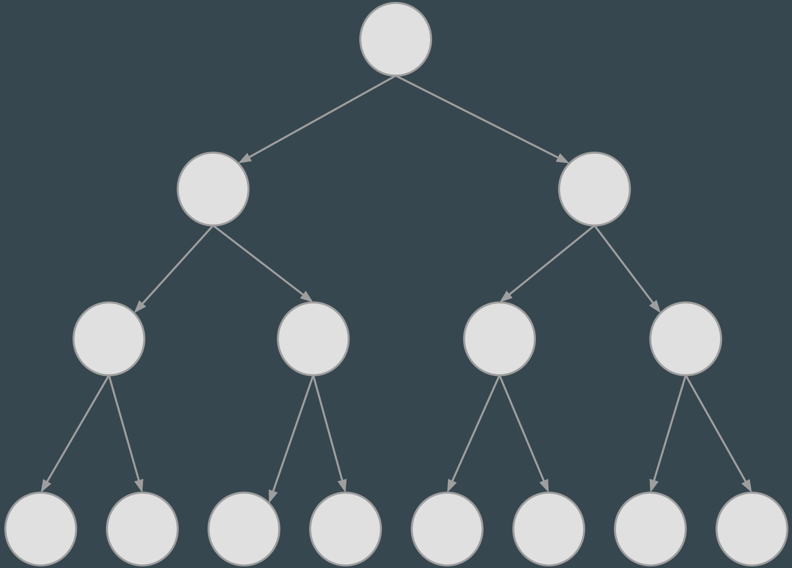
Region associated with the mth terminal node

Associated predicted response

Number of terminal nodes

- α controls a trade-off between complexity and training fit. When α = 0, then the subtree T will simply equal T_0, but as α increases, the tree will be pruned in a nested, predictable way

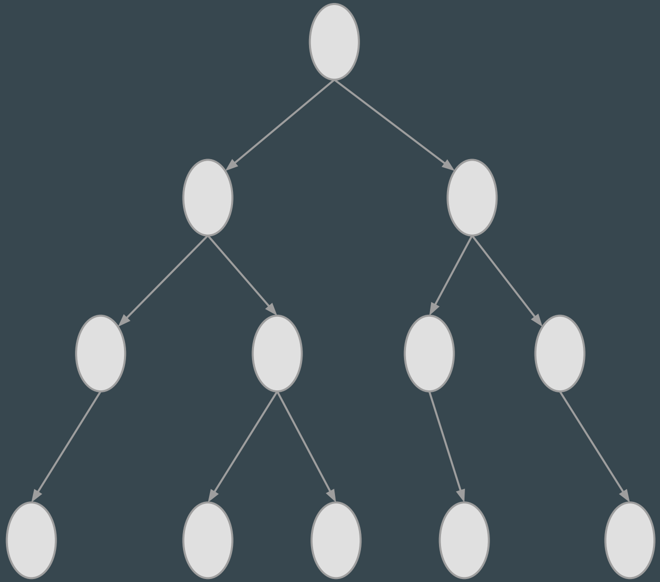
Cost Complexity



$\alpha = 0$

Cost Complexity

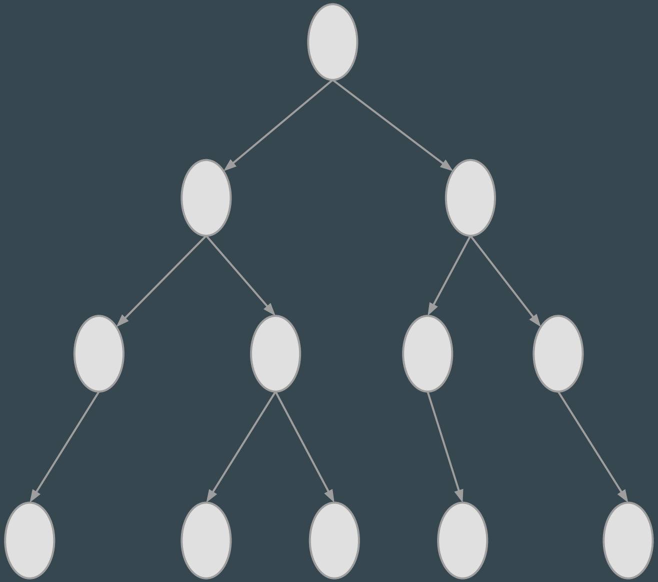
$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$



α = 3

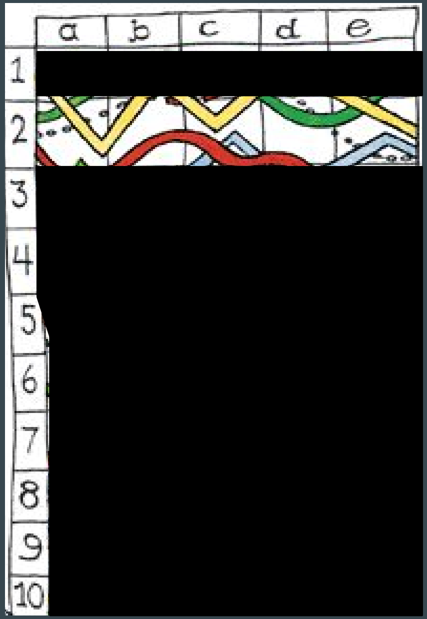
How the Cross Validation Process Works

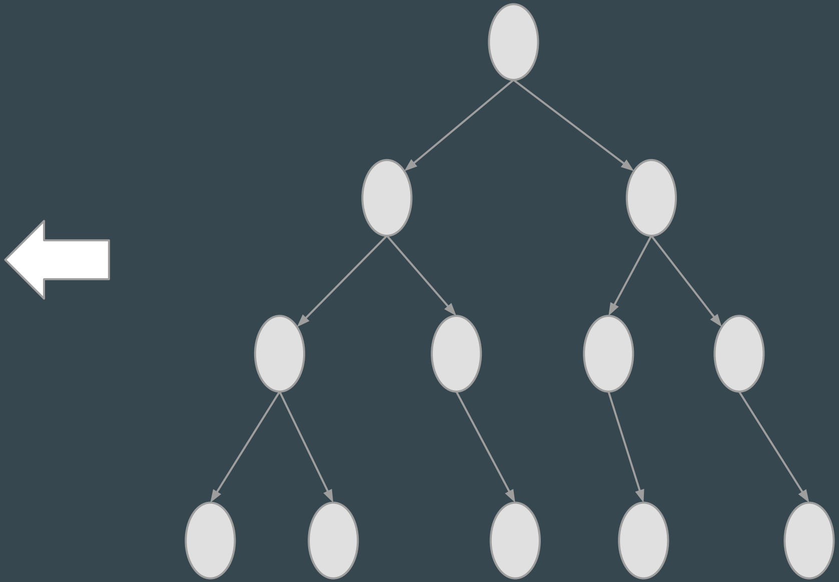
[redacted]



$\alpha = 3$

How the Cross Validation Process Works





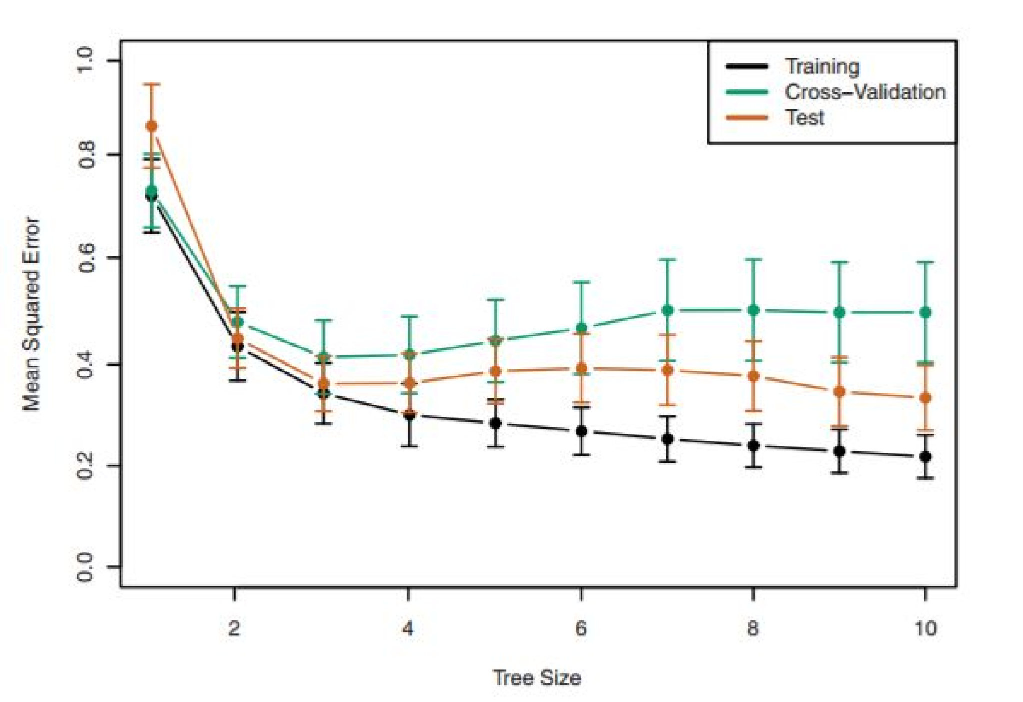
α = 3

How the Cross Validation Process Works

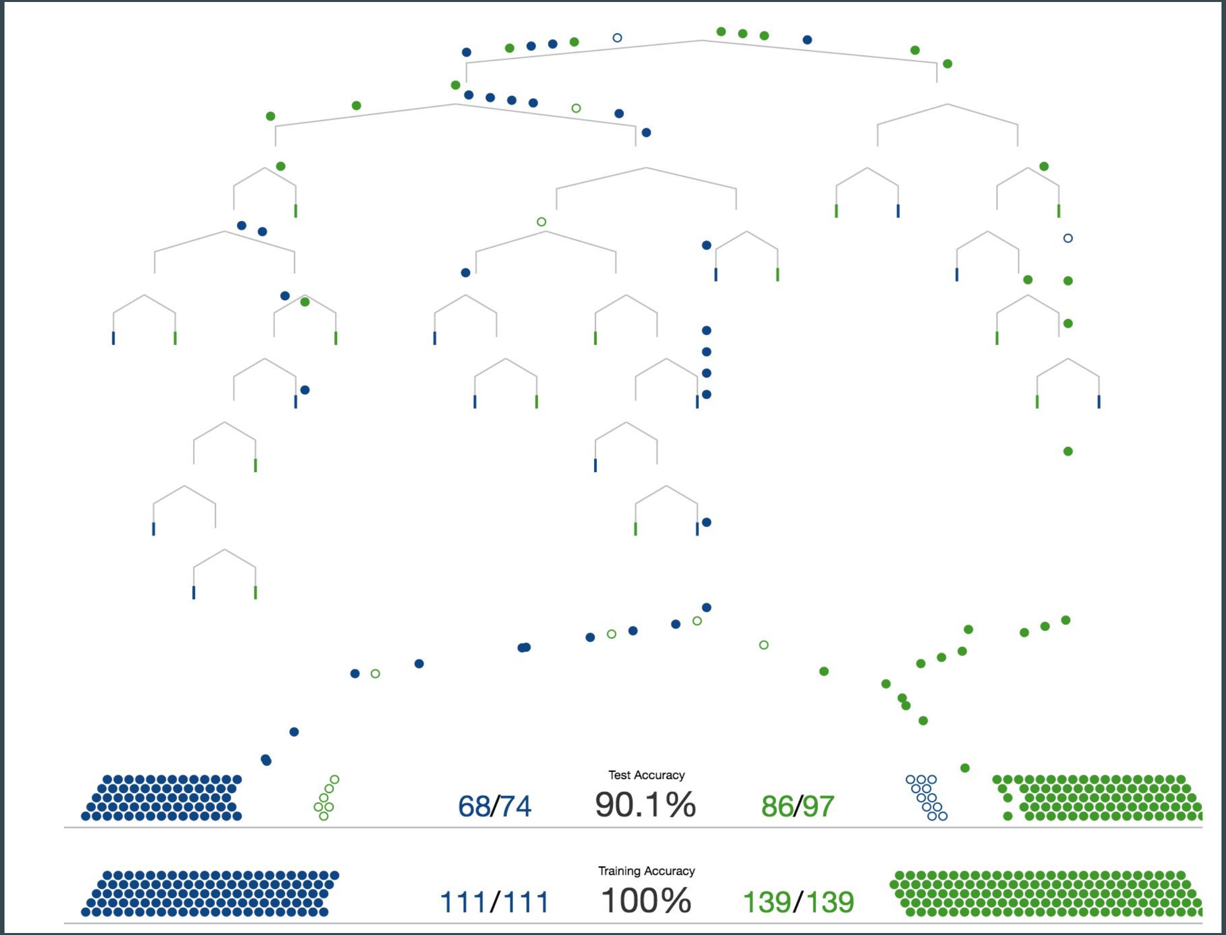
	α_0	α_1	α_2	α_3	α_n
F_1	$CV_{0, 1}$	$CV_{1, 1}$	$CV_{2, 1}$	$CV_{3, 1}$	$CV_{n, 1}$
F_2	$CV_{0, 2}$	$CV_{1, 2}$	$CV_{2, 2}$	$CV_{3, 2}$	$CV_{n, 2}$
F_3	$CV_{0, 3}$	$CV_{1, 3}$	$CV_{2, 3}$	$CV_{3, 3}$	$CV_{n, 3}$
F_k	$CV_{0, k}$	$CV_{1, k}$	$CV_{2, k}$	$CV_{3, k}$	$CV_{n, k}$
	$\bar{\alpha}_0$	$\bar{\alpha}_1$	$\bar{\alpha}_2$	$\bar{\alpha}_3$	$\bar{\alpha}_n$

Choosing the Best Subtree

- Use K-fold cross-validation or a validation set to choose which **α** value corresponds to the best fit model (dividing training observations into k-folds)
- Then return to the full data set and select the subtree corresponding to the **α** with the lowest cross-validated MSE, whick is approximates the test error



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/



Regression Trees in Review

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations or you reach a max number of nodes.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α .
3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each k = 1, 2, ... , K:
    a. Repeat Steps 1 and 2 on all but the kth fold of the training data.
    b. Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α. Average the results for each value of α, and pick α to minimize the average error.
4. Return the subtree from Step 2 that corresponds to the chosen value of α.

R Example

https://github.com/WilliamandMary-BUAD5082-Spring2017/Class-10-Tree-Based-Methods-Regression-Trees

bit.ly/BUAD5082

Questions?