Lecture 20:
Trees

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

October 21, 2021

Agenda

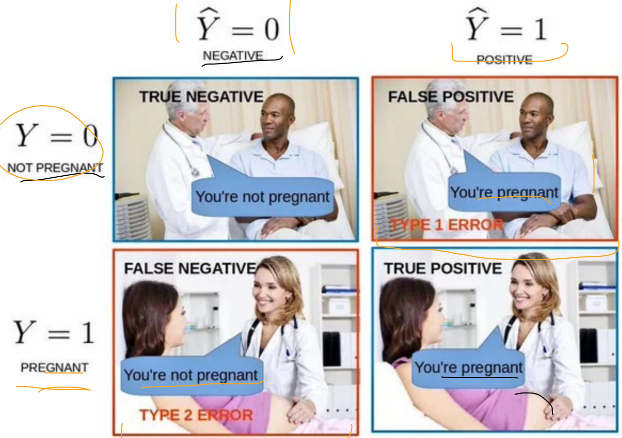
Classification

Classification: Motivation

- Admit a student to *PEG* based on their grades and LoR
- Give a credit, based on credit history, demographics?
- Classifying emails: spam, personal, social based on email contents
- Aim is to classify y based on $X's$
- y can be
  - qualitative (e.g., spam, personal, social)
  - Not necessarily ordered
  - Not necessarily two categories, but will start with the binary case

Motivation

- Two states of nature $y \to n \in \{0, 1\}$
- Two actions $(\hat{y}) \to a \in \{0, 1\}$



→ matriz de confusion

→ missclassifying

Source: https://dzone.com/articles/understanding-the-confusion-matrix

Probability, Cost, and Classification

- Two states of nature $y \to n \in \{0, 1\}$
- Two actions $(\hat{y}) \to a \in \{0, 1\}$
- Probabilities
  - $p = Pr(y = 1|X)$
  - $1 - p = Pr(y = 0|X)$
- Loss: $L(a, n)$, penalizes being in bin (a, n)
- Risk: expected loss of taking action a

Probability, Cost, and Classification

- Risk: expected loss of taking action a

$$E[L(a,n)] = \sum_n p_n L(a,n) \tag{1}$$

$$R(a) = (1-p)L(a,0) + pL(a,1)$$

- The objective is the same as before: minimize the risk
- We have to define $L(a,n)$:

Probability, Cost, and Classification

- Risk: expected loss of taking action a

$$E[L(a,n)] = \sum_n p_n L(a,n) \tag{1}$$
$$R(a) = (1-p)L(a,0) + pL(a,1)$$

- The objective is the same as before: minimize the risk
- We have to define $L(a,n)$:

$$L(n,a) = \begin{cases} 1 & \text{if } a \neq n \\ 0 & \text{if } \underbrace{a = n} \end{cases} \tag{2}$$

Probability, Cost, and Classification

- Which action do we choose?

Probability, Cost, and Classification

- Which action do we choose?

- We can compare the risk of each action

- We are going to choose to take action 1 when the risk is lower:

$$
\begin{aligned}
R(1) &< R(0) \\
1 - p &< p \\
p &> \frac{1}{2}
\end{aligned}
\tag{3}
$$

$\rightarrow$ $R(1) = (1-p)$
$R(0) = p$

$p = Pr(y=1|X)$

- This is known as the Bayes Classifier, choose the estate that minimizes the risk

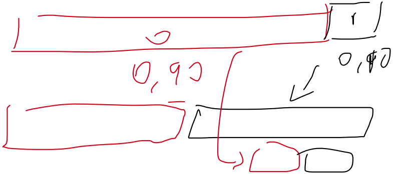
$\rightarrow$ nolul

Probability, Cost, and Classification

- Under a 0-1 penalty the problem boils down to finding $p = Pr(y = 1|X)$
- We then predict 1 if $p > 0.5$ and 0 otherwise (Bayes classifier)
- We can think 4 ways of finding this probability in binary cases
  - K-Nearest Neighbors
  - Logistic } Regularización
  - Probit
  - LDA → Bayes
- But there's a trade off each time we choose a classification rule

$$p > \frac{1}{2}$$

ROC



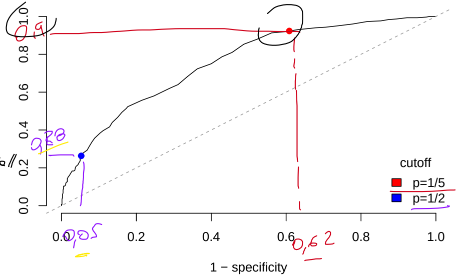
Oversampling

undersampling

- ROC curve illustrates the trade-off of the classification rule
- Gives us the ability
  - Measure the predictive capacity of our model
  -

TPR

Frac. correctamente identificado

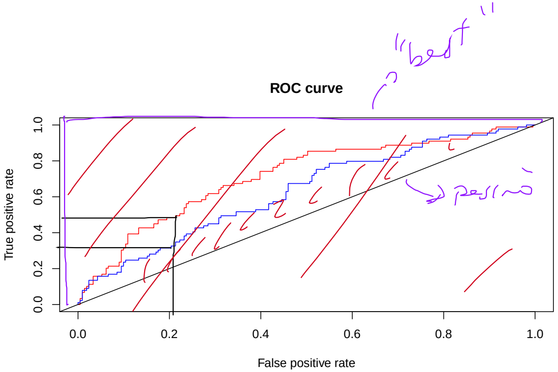


FPR

Fraction of 0 q' clasificamos incorrectamente como 1

ROC

- ROC curve illustrates the trade-off of the classification rule
- Gives us the ability
  - Measure the predictive capacity of our model
  - Compare between models like an R^2



Trees

Motivation

- I'm going to change slightly the approach
- Inspired by Leo Breiman:

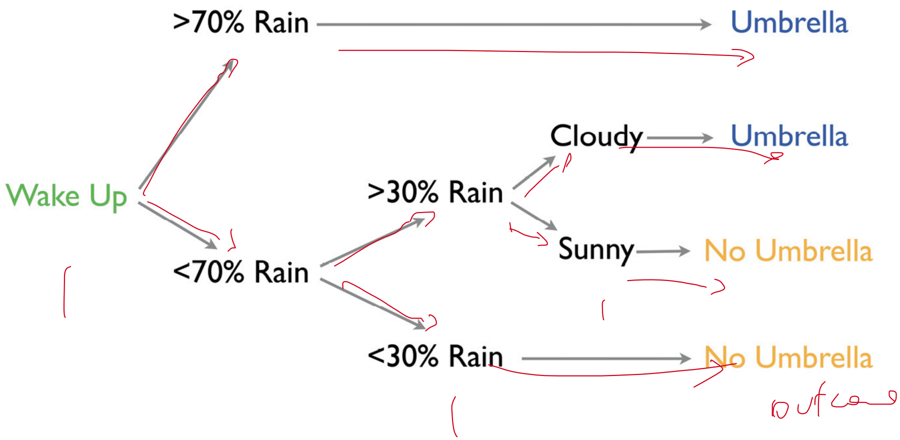
  "There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown." Breiman [2001b], p199.

  "The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools." Breiman [2001b], p199.

Motivation

- End goal is to model $y = f(x) + \epsilon$ for predictive power
  - Thus far we have imposed a lot of structure to the problem
    - Linear
    - Spatial
    - Logit
- Regression trees, and their extension random forests are very popular and effective methods
- They are very flexibly at regression functions in settings where out-of-sample predictive power is important.

Motivation

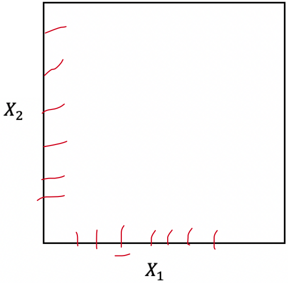


>70% Rain
Umbrella
Wake Up
Cloudy
Umbrella
>30% Rain
Sunny
No Umbrella
<70% Rain
<30% Rain
No Umbrella
outcome

Trees: Background

1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

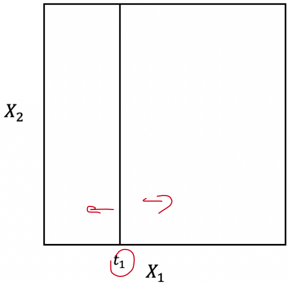
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (4)$$



Trees: Background

1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

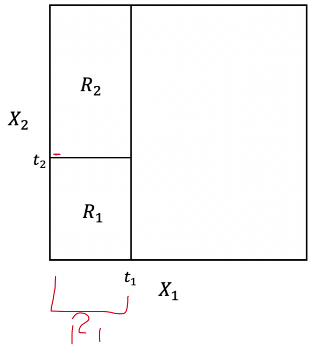
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (5)$$



Trees: Background

1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

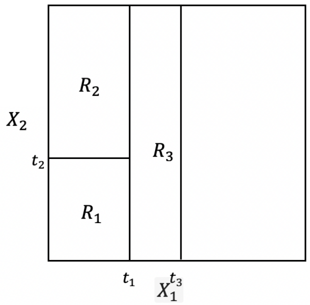
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (6)$$



Trees: Background

1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

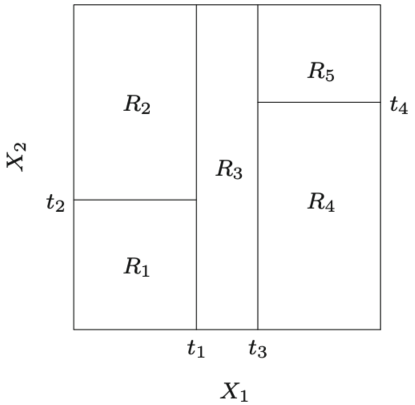
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (7)$$



Trees: Background

1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

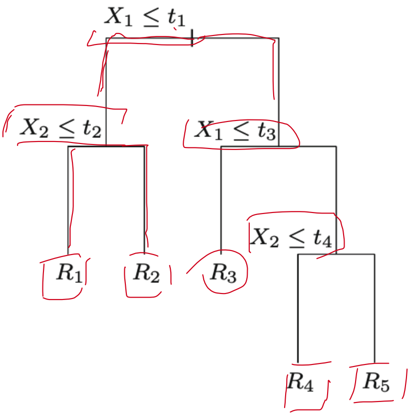
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (8)$$



Trees: Background

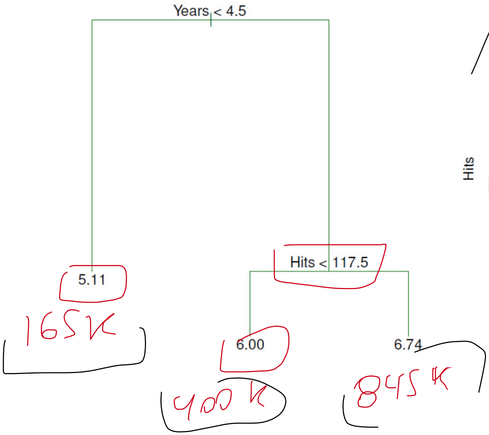
1. Tree-based methods partition the feature space into a set of rectangles,
2. fit a simple model (like a constant) in each one.

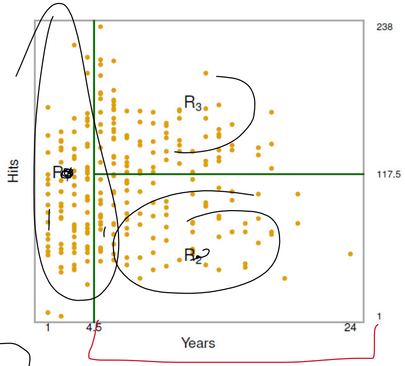
$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \qquad (9)$$



Trees: Background

$W = \beta_0 + \beta_1 Hits + \beta_2 Years + \varepsilon$

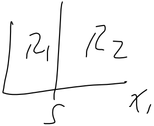




Regression Trees

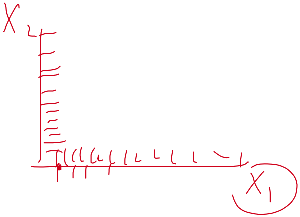
Problema
¿Cuál variable?
y
¿Dónde?

- We have data y $n \times 1$ (outcome) and X $n \times p$ (predictors)
- Some definitions
  - j is the partition variable and s is the partition point
  - Define the following half-planes



$$R_1(j,s) = \{X|X_j \leq s\} \;\&\; R_2(j,s) = \{X|X_j > s\} \tag{10}$$

  - Problem then boils down to searching the partition variable X_j and the partition point s such that



$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y-c_2)^2\right] \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial c_1} = -2\sum y_i - \sum c_1 = 0 \rightarrow \sum y_i - n c_1 = 0$$

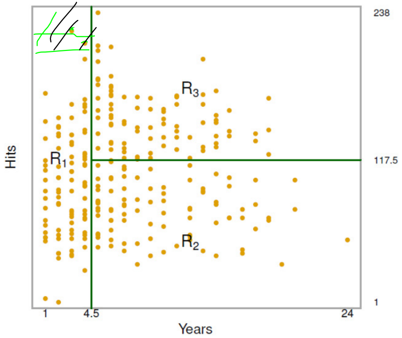
Regression Trees

- For each partition variable, and partition point, the internal minimization is the mean of each region

$$\hat{c}_m = \frac{1}{n_m} \sum (y_i | x_i \in R_m) \tag{12}$$

- Process is repeated inside each region.

Regression Trees



Regression Trees

- For each partition variable, and partition point, the internal minimization is the mean of each region

$$\hat{c}_m = \frac{1}{n_m} \sum (y_i | x_i \in R_m) \tag{12}$$

- Process is repeated inside each region.
- If the final tree has M regions then

$$\hat{f}(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m) \tag{13}$$

Regression Trees

- We grew our tree, now how do we stop?
- A tree too big, overfits the data (like a dummy for each observation)
- A smaller tree, with fewer splits (fewer regions $R_1, \ldots, R_j$) might lead to lower variance and better interpretation at the cost of a little bias
- Solution: Pruning
  - Grow a very large tree T_0
  - Prune it to get a *subtree*
  - How do we determine the best way to prune the tree? $\rightarrow$ lowest test error using cross-validation

Regression Trees

- Draw back, estimate the CV error for every possible subtree would be too much (two many possible subtrees)
- Solution: *Cost complexity pruning (weakest link pruning)*
  - We index the trees with T.
  - A subtree $T \in T_0$ is a tree obtained by collapsing the terminal nodes of another tree (cutting branches).
  - $[T]$ = number of terminal nodes of tree T

Regression Trees

- Cost complexity of tree T

$$C_\alpha(T) = \overbrace{\sum_{m=1}^{[T]} n_m Q_m(T)}^{\text{Ajuste}} + \overbrace{\alpha}^{\text{Parámetro de penalidad (Lasso } \lambda)} \overbrace{[T]}^{\text{Cuán grande}} \tag{14}$$

  - where $Q_m(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$ for regression trees
  - $Q_m(T)$ penalizes heterogeneity (impurity) within each region, and the second term the number of regions.
  - Objective: for a given α, find the optimal pruning that minimizes $C_\alpha(T)$

Regression Trees

- ~~Search mechanism for T_α (optimal pruning given α).~~
  - Result: for each α there is a unique subtree T_α that minimizes $C\alpha$ (T).
  - Weakest link: successively eliminate the branches that produce the minimum increase in $\sum_{m=1}^{[T]} n_m Q_m(T)$
  - Idea: to remove branches is to collapse, this increases impurity, ergo, we collapse the least necessary partition.
  - This eventually collapses at the initial node, but goes through a succession of trees, from the largest to the smallest, through the weakest link pruning process.
  - Breiman et al. (1984): T_α belongs to this sequence.
  - Narrow your search to this succession of subtrees.
  - Choice of α: cross validation.

Classification Trees

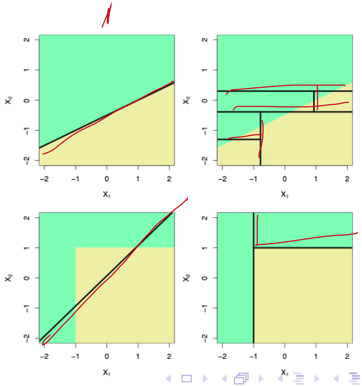
- A classification tree is very similar to a regression tree except that we try to make a prediction for a categorical rather than continuous Y.
- For each region (or node) we predict the most common category among the training data within that region.
- The tree is grown (i.e. the splits are chosen) in exactly the same way as with a regression tree except that minimizing MSE no longer makes sense.
- There are several possible different criteria to use
  - Misclasification error: $\frac{1}{n_m}\sum_{i\in R_m} I(y_i \neq k(m))$
  - Gini Index: $\sum_{k\neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$
  - Cross entropy or deviance: $-\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk})$

Trees vs. Linear Models

- Which model is better?
  - If the relationship between the predictors and response is linear, then classical linear models such as linear regression would outperform regression trees
  - On the other hand, if the relationship between the predictors is non-linear, then decision trees would outperform classical approaches

ISLR

- Top row: the true decision boundary is linear
  - Left: linear model (good)
  - Right: decision tree
- Bottom row: the true decision boundary is non-linear
  - Left: linear model
  - Right: decision tree (good)



Advantages and disadvantages of trees

- Pros:
  - Trees are very easy to explain to people (probably even easier than linear regression)
  - Trees can be plotted graphically, and are easily interpreted even by non-expert. More important variables at the top
  - They work fine on both classification and regression problems
- Cons:
  - Trees are not very accurate or robust (bagging, random forests and boosting to the rescue)
  - If the structure is lineal, CART doesn't work well

Review & Next Steps

- Trees
- Regression Trees
- Classification Trees
- Advantages and disadvantages of trees
- CART Demo

- Next class: more on trees

- Questions? Questions about software?

Further Readings

- Athey, S., & Imbens, G. W. (2019). Machine learning methods that economists should know about. Annual Review of Economics, 11, 685-725.
- Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science, 16(3):199–231, 2001b.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.

CART Demo: Regression

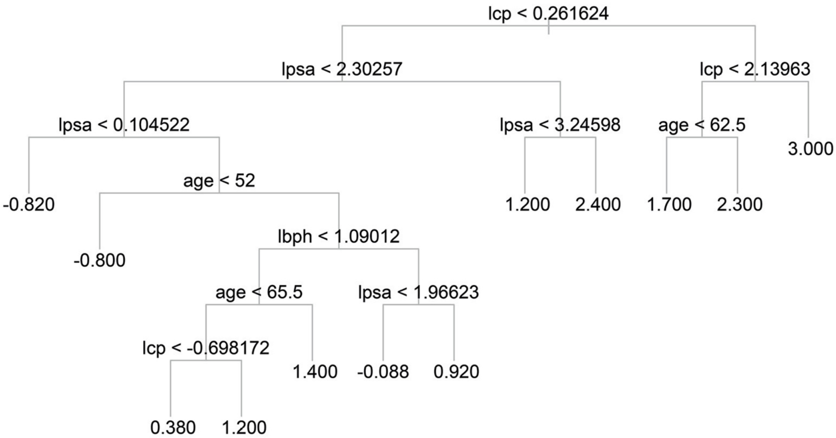
```r
library("tree")

prostate <- read.csv("prostate.csv")
str(prostate)
```

```
## 'data.frame':    97 obs. of  6 variables:
##  $ lcavol : num  -0.58 -0.994 -0.511 -1.204 0.751 ...
##  $ age    : int  50 58 74 58 62 50 64 58 47 63 ...
##  $ lbph   : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ lcp    : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ gleason: int  6 6 7 6 6 6 6 6 6 6 ...
##  $ lpsa   : num  -0.431 -0.163 -0.163 -0.163 0.372 ...
```

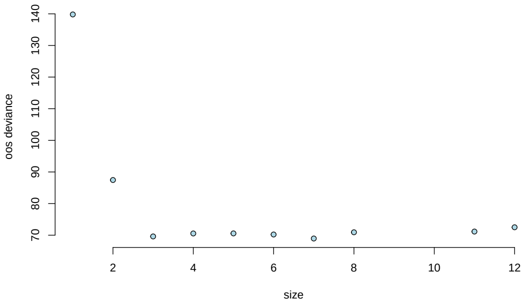
CART Demo: Regression

```
pstree <- tree(lcavol ~., data=prostate, mincut=1)
par(mfrow=c(1,1))
plot(pstree, col=8)
text(pstree, digits=2)
```



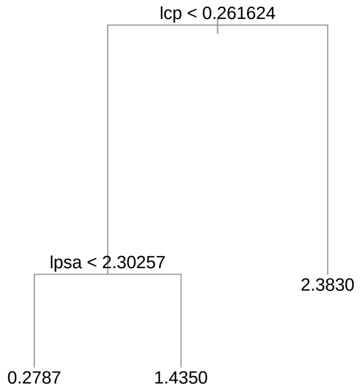
CART Demo: Regression

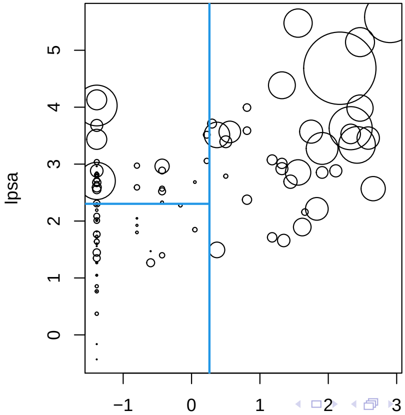
```
## Use cross-validation to prune the tree
cvpst <- cv.tree(pstree, K=10)
par(mai=c(.8,.8,0.1,0.1))
plot(cvpst$size, cvpst$dev, xlab="size", ylab="oos deviance", pch=21, bg="lightblue", bty="n")
```



CART Demo: Regression

```
par(mfrow=c(1,2))
## note across the top is 'average number of observations per leaf';
plot(cvpst, pch=21, bg=8, type="p", cex=1.5, ylim=c(65,100))
pstcut <- prune.tree(pstree, best=3)
```





CART Demo: Classification

```r
## read in the NBC show characteristics
nbc <- read.csv("nbc_showdetails.csv")
## lets look at the show demographics for predicting genre
demos <- read.csv("nbc_demographics.csv", row.names=1)
demos$genre <- as.factor(nbc$Genre)
head(demos[,c(11:17)])
```

```
##                                    WIRED.CABLE.W.PAY WIRED.CABLE.W.O.PAY
## Living with Ed                               36.4929             43.6019
## Monarch Cove                                 31.2500             39.5395
## Top Chef                                     42.8806             34.1528
## Iron Chef America                            44.3794             29.9661
## Trading Spaces: All Stars                    46.4945             34.5018
## Lisa Williams: Life Among the Dead           36.7206             35.3349
##                                    DBS.OWNER BROADCAST.ONLY VIDEO.GAME.OWNER
## Living with Ed                       20.2607          0.000          66.4692
## Monarch Cove                         29.0132          0.000          54.7368
## Top Chef                             23.2329          0.041          50.5019
## Iron Chef America                    25.7776          0.000          56.9295
## Trading Spaces: All Stars            19.1882          0.000          49.4465
## Lisa Williams: Life Among the Dead   28.6374          0.000          51.7321
##                                    DVD.OWNER VCR.OWNER
## Living with Ed                       98.4597   90.4028
## Monarch Cove                         94.2105   74.1447
## Top Chef                             92.2557   78.0783
## Iron Chef America                    94.2408   83.6464
## Trading Spaces: All Stars            90.2214   81.1808
## Lisa Williams: Life Among the Dead   94.2263   84.9885
```

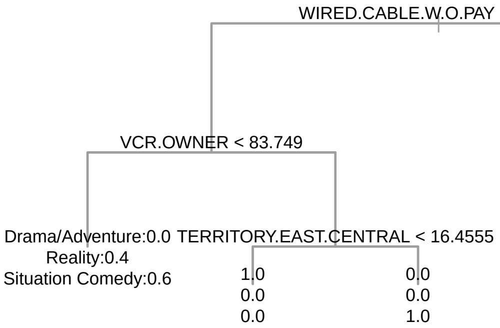
CART Demo: Classification

```r
## tree fit; it knows to fit a classification tree since genre is a factor.
genretree <- tree(genre ~ . , data=demos, mincut=1)
genretree
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 40 75.800 Drama/Adventure ( 0.47500 0.42500 0.10000 )
##    2) WIRED.CABLE.W.O.PAY < 28.6651 22 33.420 Drama/Adventure ( 0.72727 0.09091 0.18182 )
##      4) VCR.OWNER < 83.749 5  6.730 Situation Comedy ( 0.00000 0.40000 0.60000 ) *
##      5) VCR.OWNER > 83.749 17  7.606 Drama/Adventure ( 0.94118 0.00000 0.05882 )
##       10) TERRITORY.EAST.CENTRAL < 16.4555 16  0.000 Drama/Adventure ( 1.00000 0.00000 0.00000 ) *
##       11) TERRITORY.EAST.CENTRAL > 16.4555 1  0.000 Situation Comedy ( 0.00000 0.00000 1.00000 ) *
##    3) WIRED.CABLE.W.O.PAY > 28.6651 18 16.220 Reality ( 0.16667 0.83333 0.00000 )
##      6) BLACK < 17.2017 15  0.000 Reality ( 0.00000 1.00000 0.00000 ) *
##      7) BLACK > 17.2017 3  0.000 Drama/Adventure ( 1.00000 0.00000 0.00000 ) *
```

CART Demo: Classification

```r
## tree plot
plot(genretree, col=8, lwd=2)
## print the predictive probabilities
text(genretree, label="yprob")
```



CART Demo: Classification

```r
## example of prediction (type="class"  to get max prob classifications back)
genrepred <- predict(genretree, newdata=demos, type="class")
genrepred
```

```
##  [1] Reality         Drama/Adventure  Reality         Reality
##  [5] Reality         Reality          Reality         Reality
##  [9] Reality         Reality          Reality         Reality
## [13] Reality         Drama/Adventure  Drama/Adventure  Drama/Adventure
## [17] Drama/Adventure  Drama/Adventure  Situation Comedy Drama/Adventure
## [21] Drama/Adventure  Drama/Adventure  Situation Comedy Situation Comedy
## [25] Situation Comedy Drama/Adventure  Reality         Drama/Adventure
## [29] Drama/Adventure  Drama/Adventure  Reality         Drama/Adventure
## [33] Situation Comedy Drama/Adventure  Situation Comedy Drama/Adventure
## [37] Reality         Drama/Adventure  Drama/Adventure  Drama/Adventure
## Levels: Drama/Adventure Reality Situation Comedy
```