

Decision Trees

Praphul Chandra



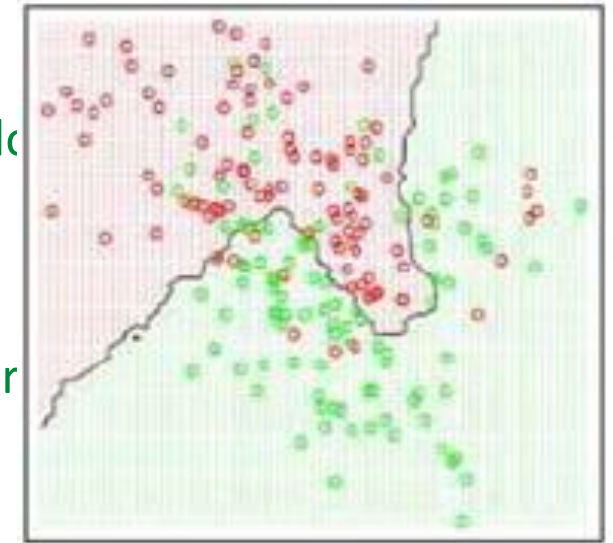
Decision Trees | Statistical Decision Theory | K-Nearest Neighbor

- Statistical Decision Theory

- The best prediction of Y at an point $X=x$ is the conditional mean. (L2 loss)

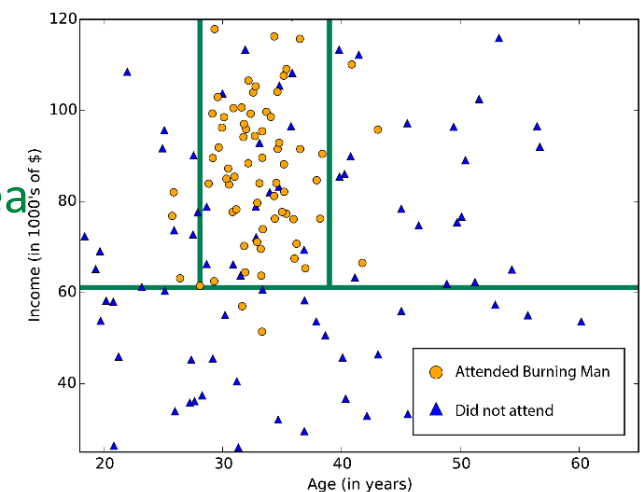
- knn

- At each point x , approximate y by averaging all y_i with input x_i near x
 - Near $x = k$ nearest neighbors
 - Locally constant approximation



- Decision Tree

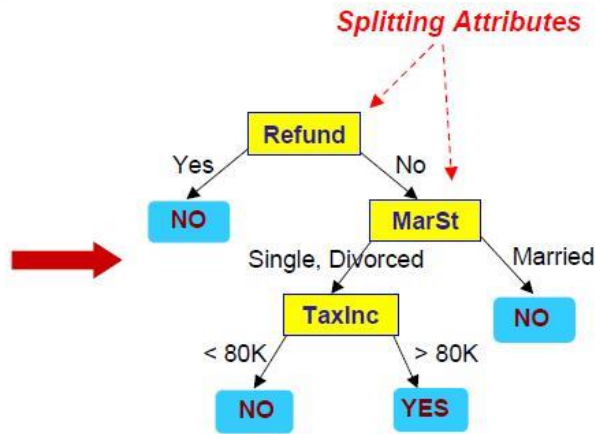
- At each point x , approximate y by averaging all y_i with input x_i near x
 - Near $x =$ Region in which x lies | Find the region optimally
 - Locally constant approximation
 - M5 variant of decision tree embeds linear regression in each leaf



Building & Using Trees

Tid	categorical		categorical		continuous		class
	Refund	Marital Status	Taxable Income	Cheat			
1	Yes	Single	125K	No			
2	No	Married	100K	No			
3	No	Single	70K	No			
4	Yes	Married	120K	No			
5	No	Divorced	95K	Yes			
6	No	Married	60K	No			
7	Yes	Divorced	220K	No			
8	No	Single	85K	Yes			
9	No	Married	75K	No			
10	No	Single	90K	Yes			

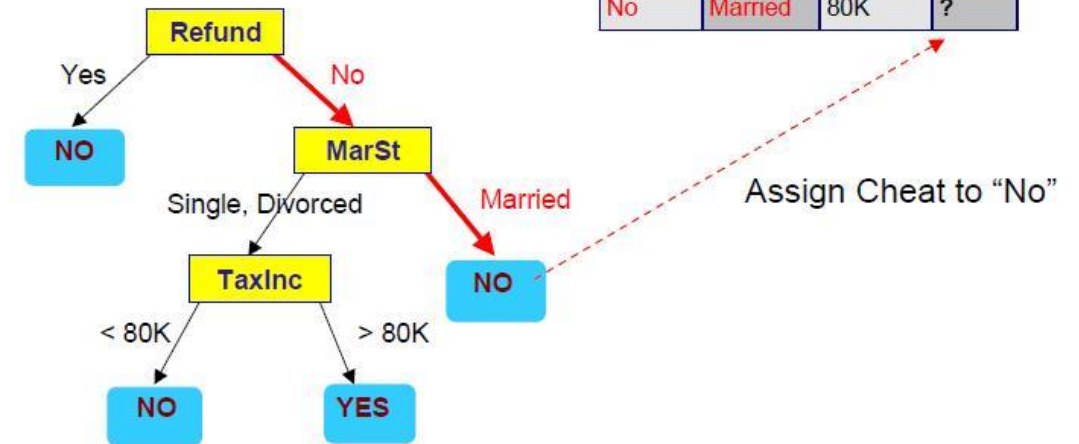
Training Data



Model: Decision Tree

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Build

- Think : "If, Then" rules specified in the feature space.
- Greedily divide (binary split) the feature space into distinct, non-overlapping regions

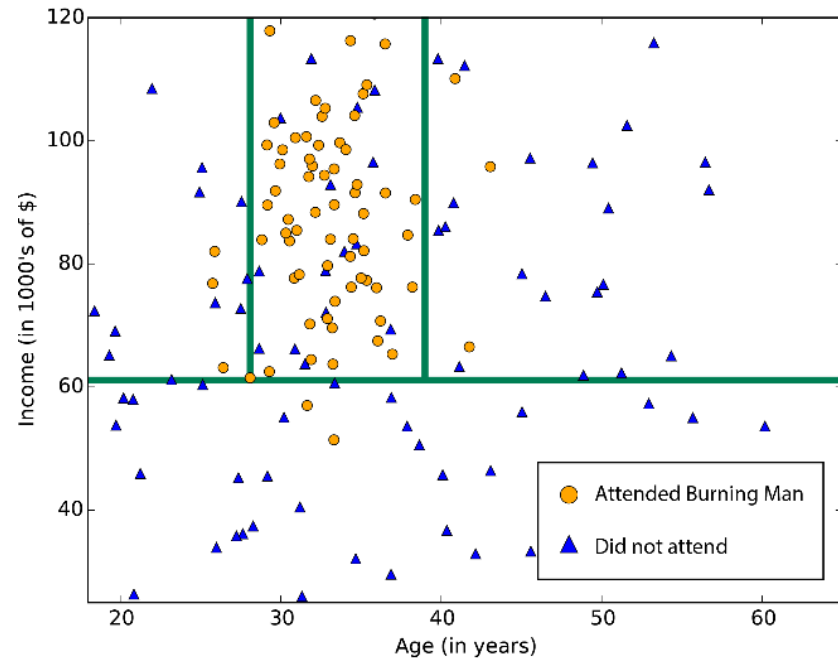
Use

- Every observation mapped to a leaf node assigned the label most commonly occurring in that leaf (Classification)
- Every observation mapped to a leaf node assigned the mean of the samples in the leaf (Regression)
- "Natural" clustering given the target variable.



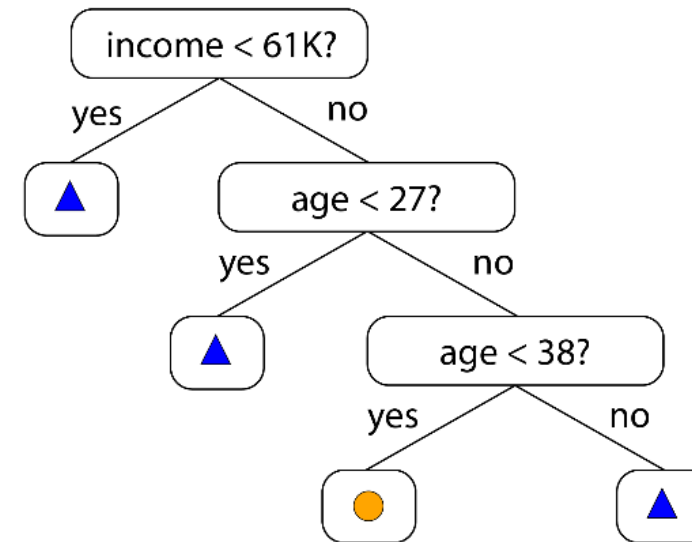
Decision Trees : Continuous splitting of the feature spaces

- In Feature Space



- The feature space contains all data
- Divided regions contain “homogeneous” data subsets
- Region boundaries define regions (*homogeneous data*)

- As a Tree



- Root contains all data
- Leaves contain “homogeneous” data subsets
- Paths along branches define leaves (*homogeneous data*)

Theme & Key Variations

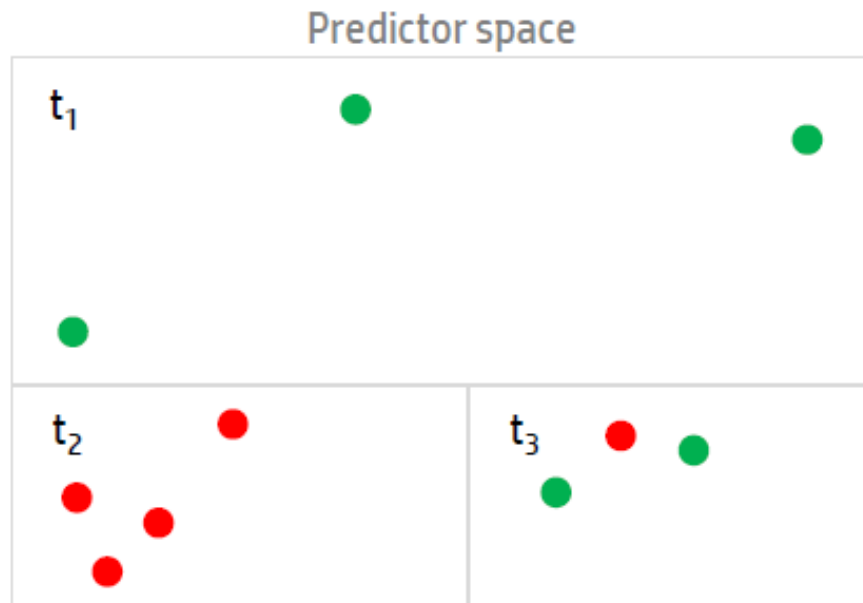
- Decision Trees :
 - Continuous splitting of the feature space
 - Recursive Partitioning of the feature space
- Split
 - Divide the data such that all data in subset-1 is “different” from the data in subset-2 in a certain dimension.
- How to split?
 - Which variable to use to split the data?
 - Which value of the variable to split on?
- What criteria should be used to evaluate a split?
 - What is the split trying to achieve?
 - How do you measure the homogeneity of a subset?
 - In Classification / Regression
 - Supervised Clustering
- Algorithm names
 - CART
 - C4.5
 - C5.0
 - CHAID
 - ID3
 - ...
- Other Variations
 - Handling missing values
 - Different category, surrogate splits etc.
 - More than two child nodes
 - One variable appears only once in the tree



Choosing the Split - Classification

What is a good split?

- Among all possible splits (*all features, all split points*)
- Which split maximizes gain / minimizes error (*Greedy*)
- Information Gain / Impurity reduction.



Choosing feature, split-point

- Cluster “homogeneous” data (subset of data)
- What is a good split measure?
 - Classification Error $1 - \max_j p_j$
 - Gini Index $p_1(1 - p_2) + p_2(1 - p_1)$
 - Entropy $p_1 \log(p_1) + p_2 \log(p_2)$

$$i(t_1) = 1 - \max\{p_g, p_r\} = 1 - \max\left\{\frac{3}{3}, \frac{0}{3}\right\} = 0$$

$$i(t_2) = 1 - \max\{p_g, p_r\} = 1 - \max\left\{\frac{0}{4}, \frac{4}{4}\right\} = 0$$

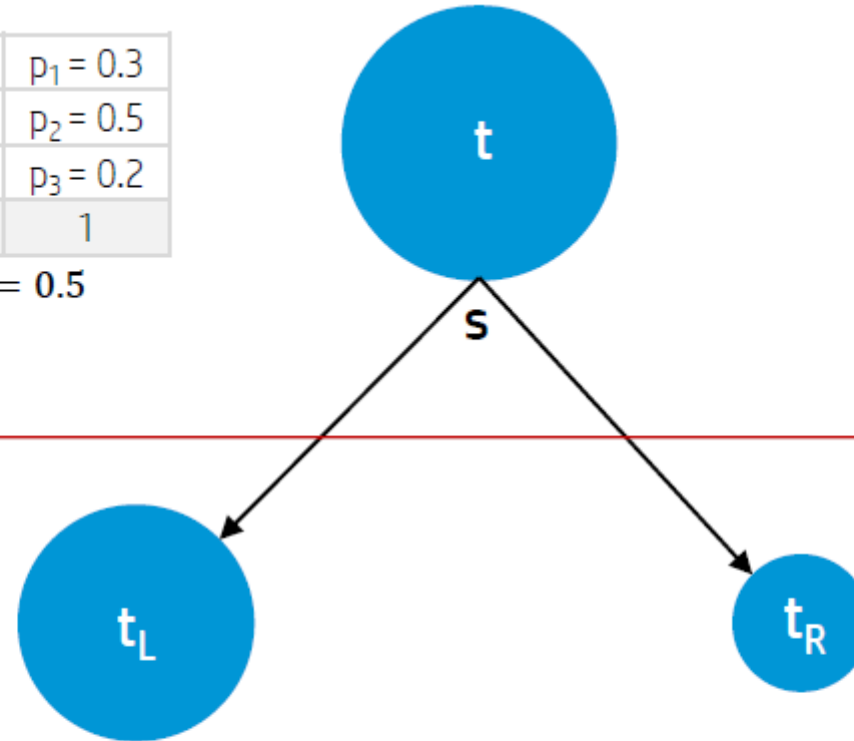
$$i(t_3) = 1 - \max\{p_g, p_r\} = 1 - \max\left\{\frac{2}{3}, \frac{1}{3}\right\} = 0.33$$



Impurity = Classification Error Rate

Class 1	$n(t_1) = 60$	$p_1 = 0.3$
Class 2	$n(t_2) = 100$	$p_2 = 0.5$
Class 3	$n(t_3) = 40$	$p_3 = 0.2$
Total	$n(t) = 200$	1

$$i(t) = 1 - (0.5) = 0.5$$



Class 1	$n(t_1) = 10$	$p_1 = 0.07$
Class 2	$n(t_2) = 100$	$p_2 = 0.66$
Class 3	$n(t_3) = 40$	$p_3 = 0.27$
Total	$n(t) = 150$	1

$$i(t_L) = 1 - 0.66 = 0.33$$

Class 1	$n(t_1) = 50$	$p_1 = 1.0$
Class 2	$n(t_2) = 0$	$p_2 = 0.0$
Class 3	$n(t_3) = 0$	$p_3 = 0.0$
Total	$n(t) = 50$	1

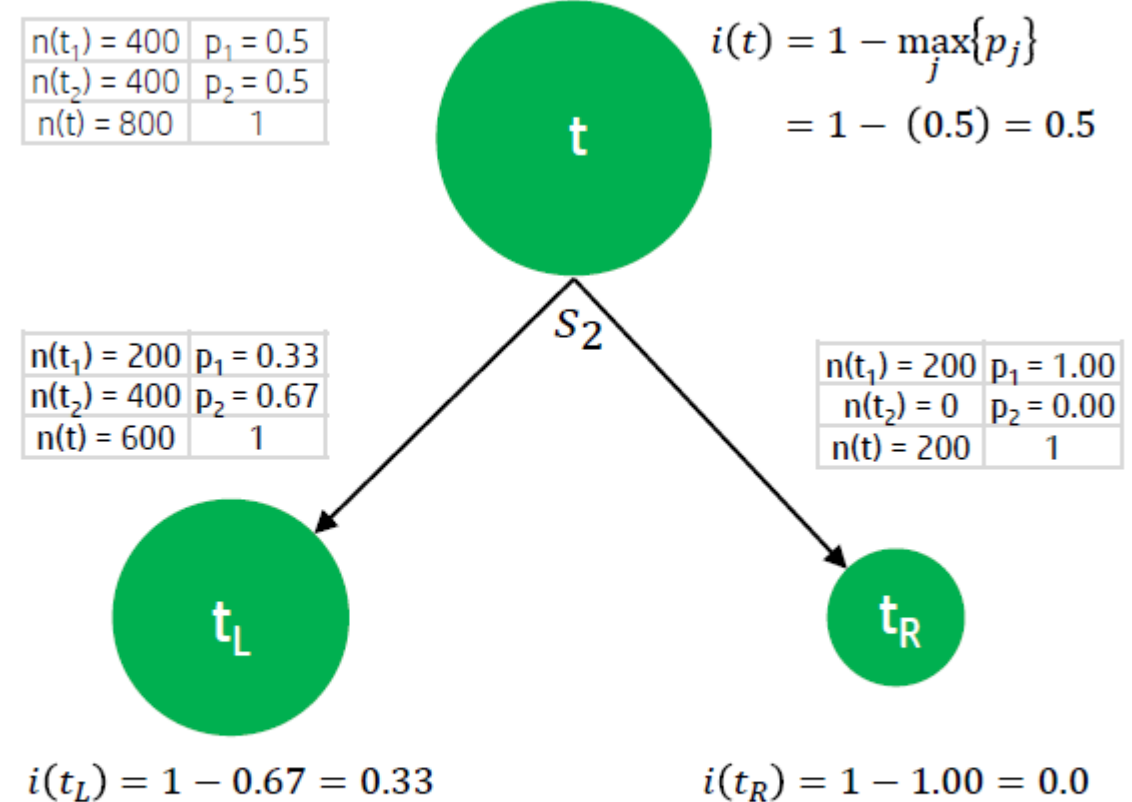
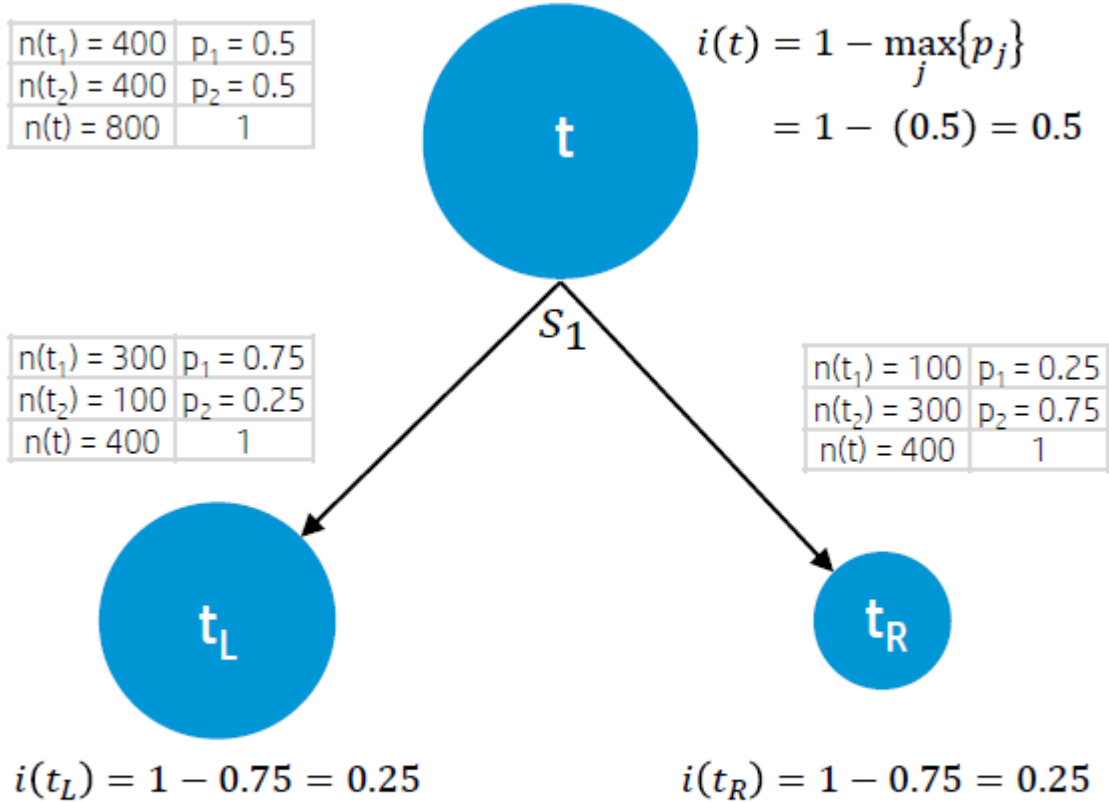
$$i(t_R) = 1 - 1.0 = 0$$

$$\frac{150}{200} \times 0.33 + \frac{50}{200} \times 0 = 0.25$$

$$\Delta i(s, t) = 0.5 - 0.25 = 0.25$$

maximize { Information Gain }

Impurity = Classification Error Rate (cont'd)

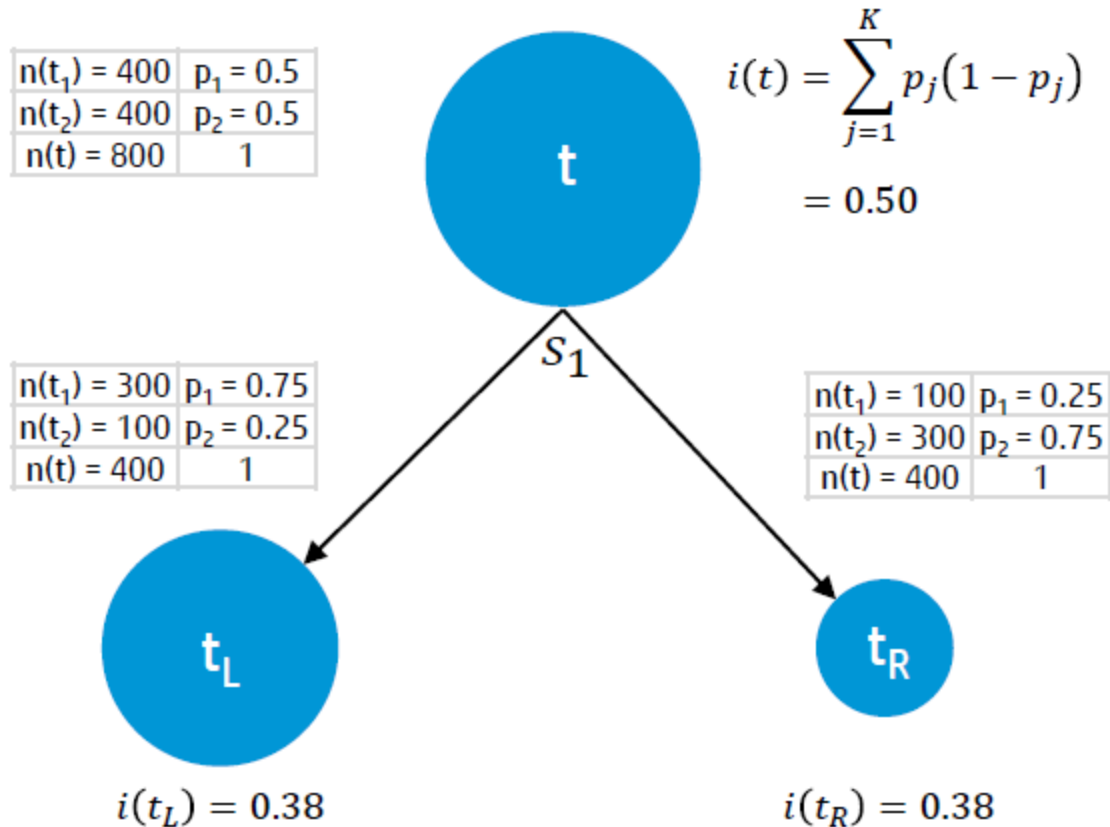


$$\Delta i(s, t) = 0.5 - \left[\frac{400}{800} \times 0.25 + \frac{400}{800} \times 0.25 \right] = 0.25$$

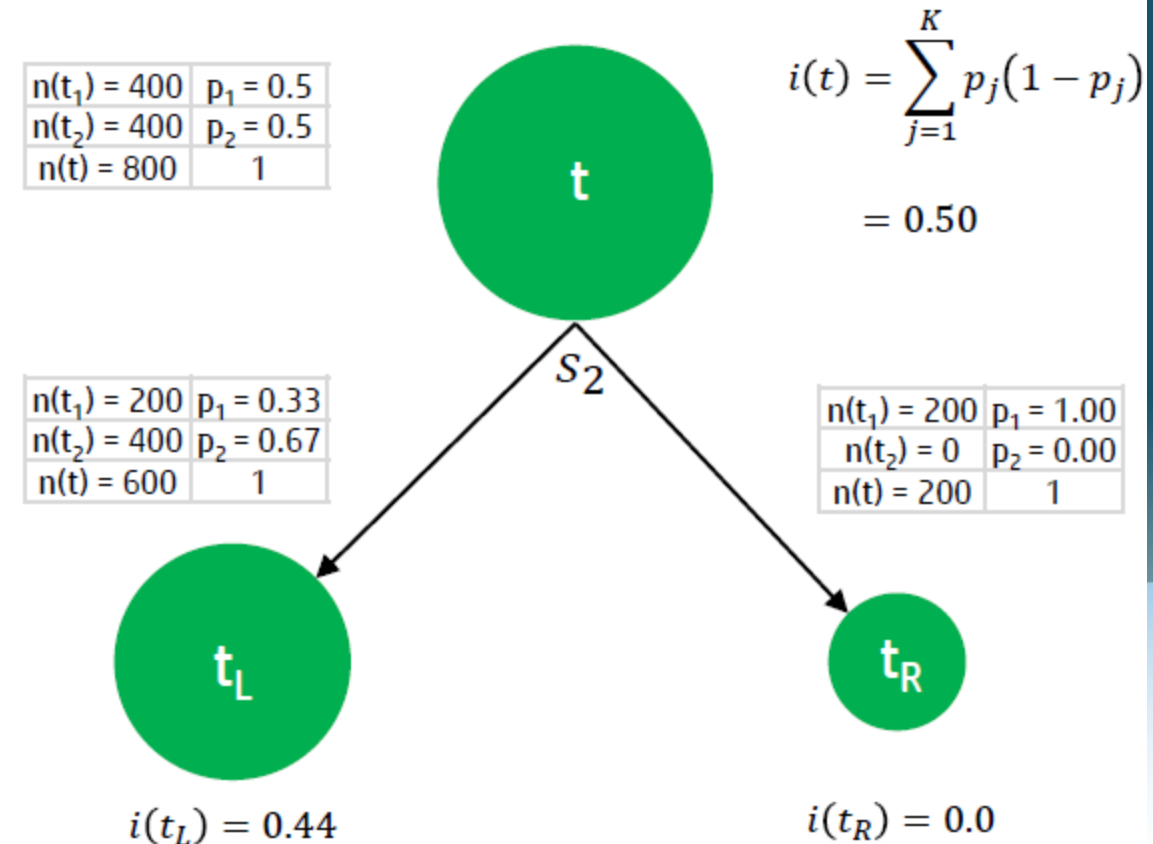
$$\Delta i(s, t) = 0.5 - \left[\frac{600}{800} \times 0.33 + \frac{200}{800} \times 0.0 \right] = 0.25$$



Impurity = Gini Index



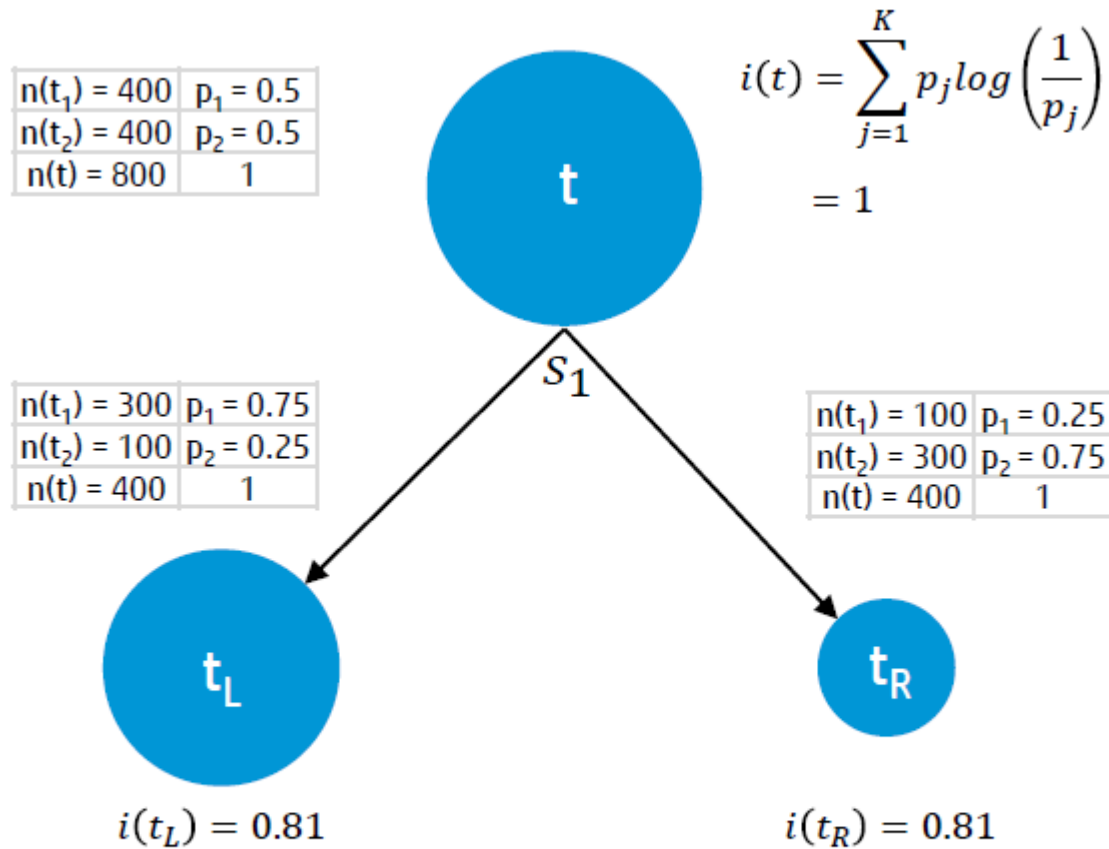
$$\Delta i(s, t) = 0.50 - \left[\frac{400}{800} \times 0.38 + \frac{400}{800} \times 0.38 \right] = 0.12$$



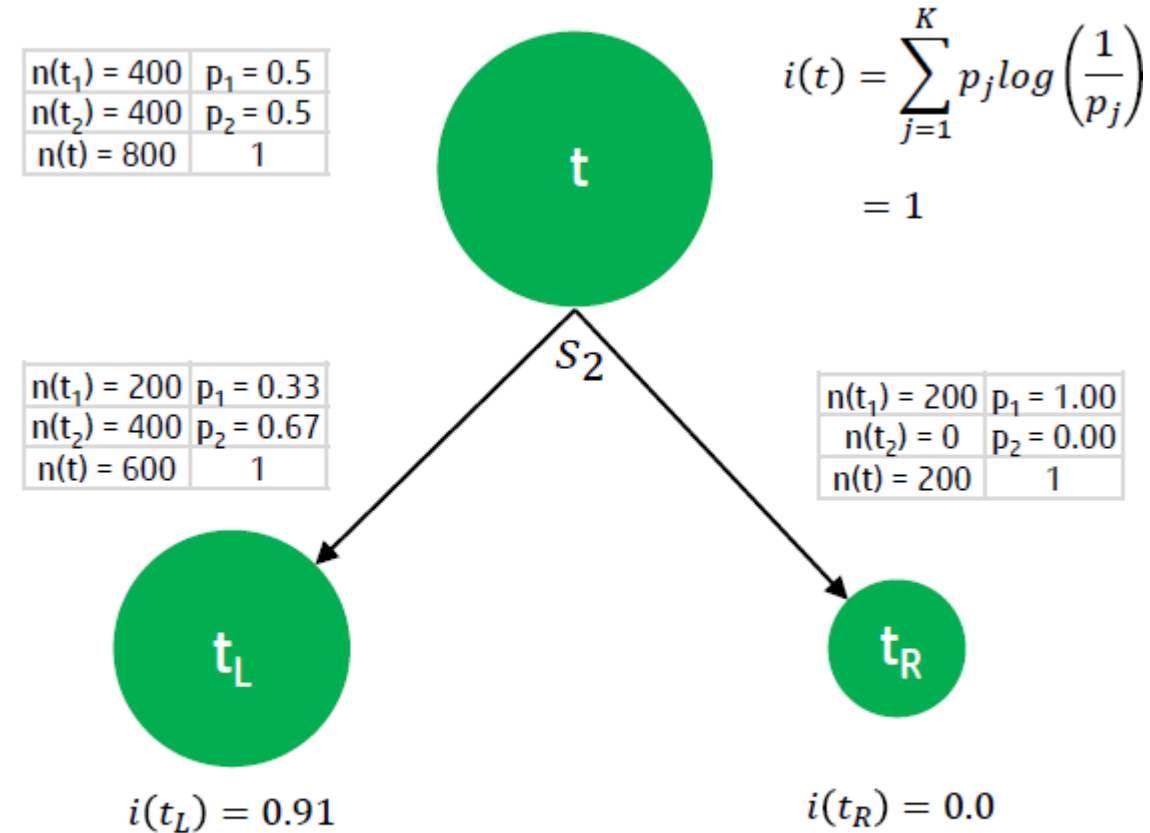
$$\Delta i(s, t) = 0.50 - \left[\frac{600}{800} \times 0.44 + \frac{200}{800} \times 0.0 \right] = 0.17$$



Impurity = Cross Entropy



$$\Delta i(s, t) = 1 - \left[\frac{400}{800} \times 0.81 + \frac{400}{800} \times 0.81 \right] = 0.19$$

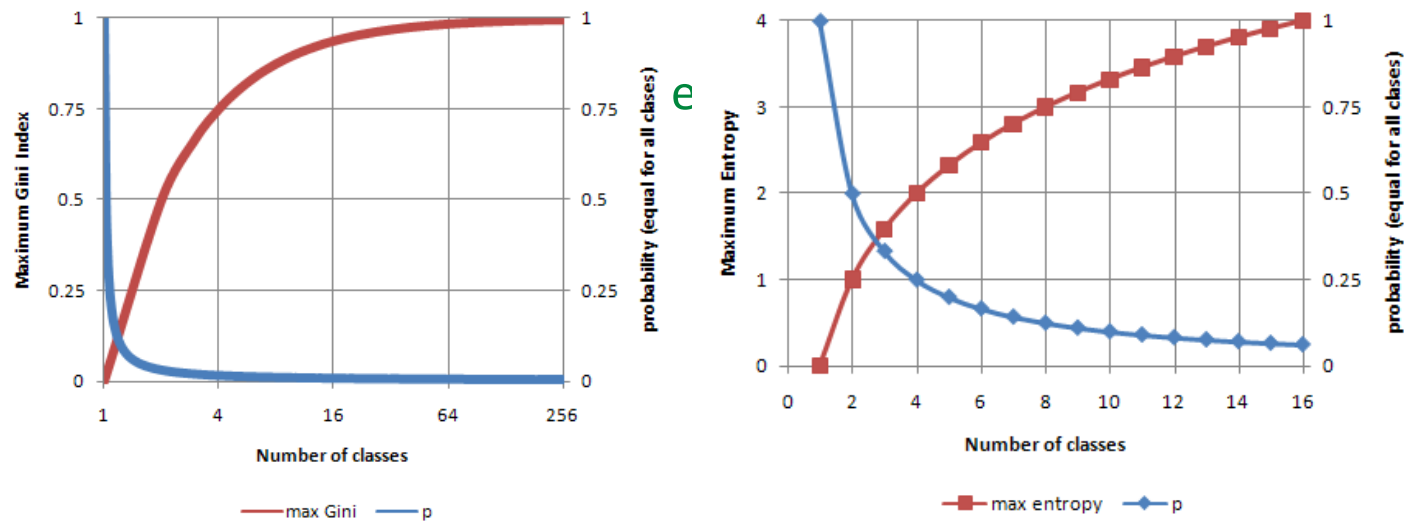


$$\Delta i(s, t) = 1 - \left[\frac{600}{800} \times 0.91 + \frac{200}{800} \times 0.0 \right] = 0.3175$$

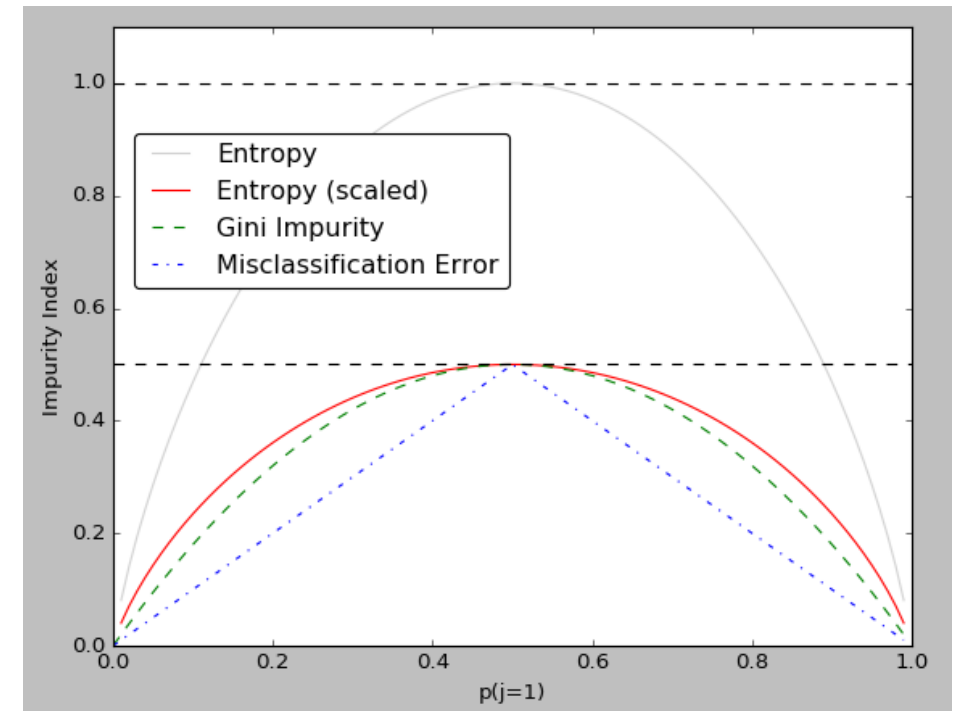


Classification error vs. Gini vs. Entropy

- Measures of Impurity
 - Determine Information Gain
 - Determine split choice
- For binary classification
 - All measures reach a maxima at (0.5,0.5)
 - All measure are symmetrical around the



<http://people.revoledu.com/kardi/tutorial/DecisionTree/how-to-measure-impurity.htm>



When to stop splitting?

Decision Tree

- Continuous splitting of the feature space
- Recursive Partitioning of the feature space

When to stop splitting (Avoiding overfitting)

When will we be “forced” to stop?

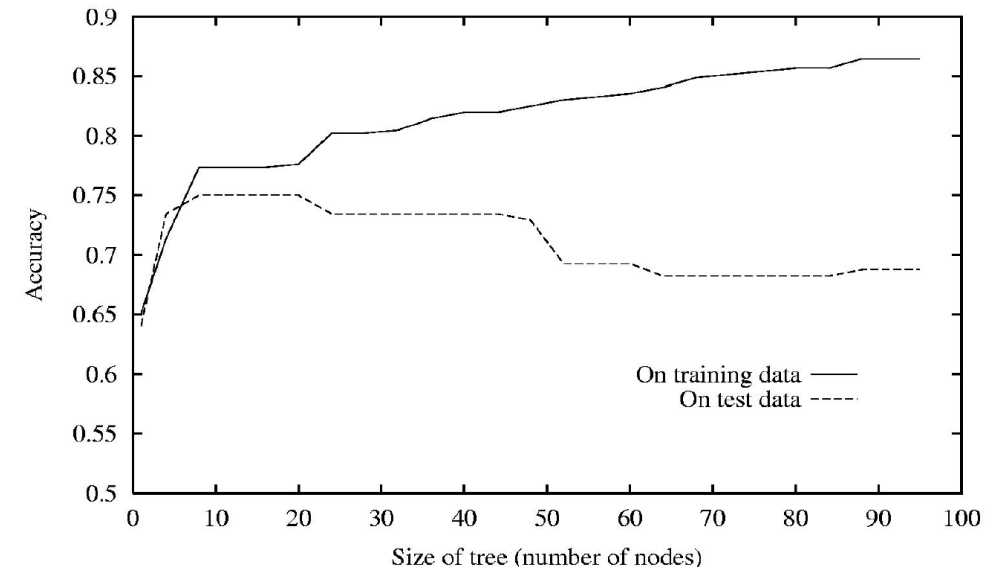
- When all nodes are pure (homogeneous leaves)
- These trees can be very deep : Overfitting
- Good trees don't over-fit !
- All models must guard against overfitting

Early Stopping

- Information Gain < Threshold
- Minimum Instances per Node
- Maximum Tree Depth

Grow & Prune

- Tree building is greedy!
- Current split gain < Future split gain (Gotcha !)



Split & Merge : Grow & Prune

Key Idea

- Grow deep trees first (Greedy workaround)
- Prune low gain branches.

What is a good tree?

- When to stop pruning?
- Overfitting measure: number of leaves, depth of tree

Cost Complexity Tradeoff

- Cost of pruning : Increase in Impurity
- Reduction in Complexity : Shorter trees, Fewer leaves

Optimal Tradeoff

- Parameter trading off cost complexity
- Try different values: choose one based on performance on test data



Decision Tree

- Function Approximation formulation
- Choosing feature, split-point
 - Cluster “homogeneous” data (subset of data)
 - What is a good split measure?
 - Classification Error
 - Gini Index $1 - \max_j p_j$
 - Entropy $p_1 \log(p_1) + p_2 \log(p_2)$
 - CART, C4.5, CHAID, ID3 variants
- When to stop splitting (Avoiding overfitting)
 - Grow & Prune
 - Complexity (Hyper)-Parameter : Penalty for # nodes
 - Optimal Cp : Grid search + (k-fold)-Cross Validation
 - Metric: TreeMisclassificationError/RootNodeError

$$f(X) = \sum_{m=1}^{|T|} c_m \cdot 1_{(X \in R_m)} \quad \text{Decision Tree}$$

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j \quad \text{Linear Regression}$$

$$N_m = \#\{x_i \in R_m\}$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$$



(Additional) Advantages of splitting

Splits : Branches for homogenizing data

- Alternative splits evaluated at build-time
- If an alternative split \sim actual split, use the alternative split at prediction time if variable missing.

Feature Importance

- Reduction in Optimization Criteria due to splits containing feature.
- Features which appear higher and more often more important.



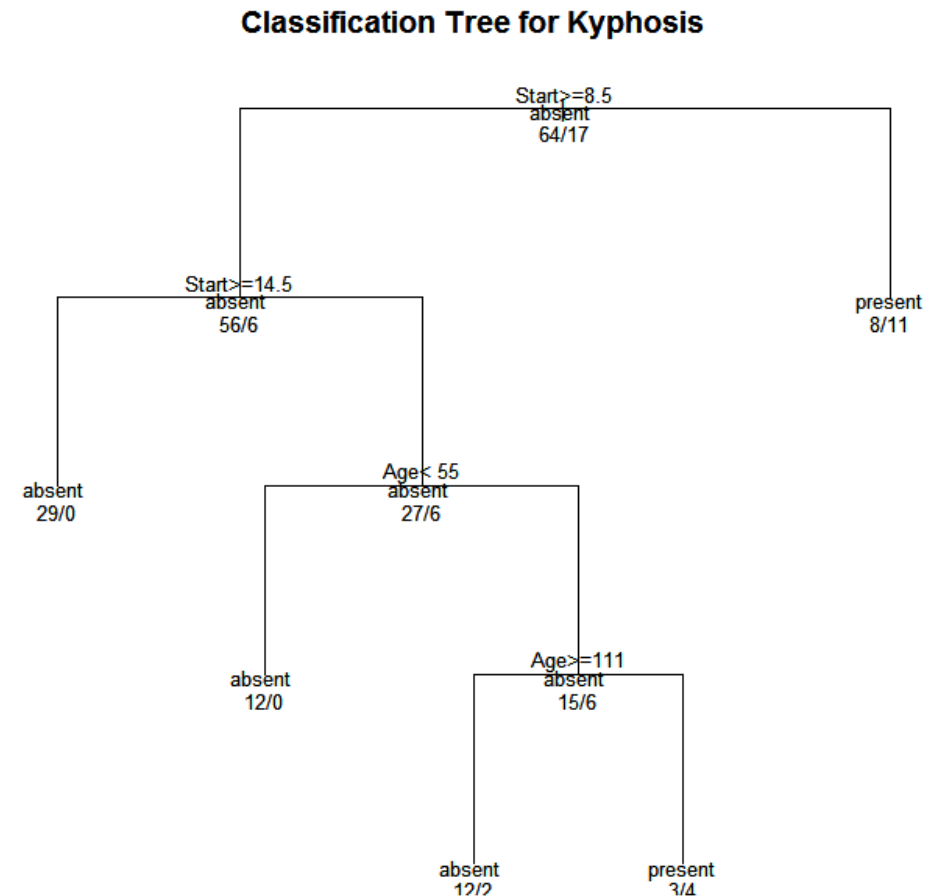
Example

- predict a type of deformation (kyphosis) after surgery, from
 - age in months (Age),
 - number of vertebrae involved (Number), h
 - highest vertebrae operated on (Start).

```
fit <- rpart(Kyphosis ~ Age + Number + Start, method="class", data=kyphosis)
```

```
plot(fit, uniform=TRUE, main="Classification Tree for Kyphosis")
```

```
text(fit, use.n=TRUE, all=TRUE, cex=.8)
```



<http://www.statmethods.net/advstats/cart.html>



Example

- Classify radars
 - “good” (evidence of some structure in the ionosphere based on reflections received from transmitted rays)
 - “bad” : signals pass through the ionosphere.

#split into training and test sets

```
Ionosphere[,"train"] <- ifelse(runif(nrow(Ionosphere))<0.8,1,0)
```

#separate training and test sets

```
trainset <- Ionosphere[Ionosphere$train==1,]
```

```
testset <- Ionosphere[Ionosphere$train==0,]
```

#build model, plot tree

```
rpart_model <- rpart(Class~.,data = trainset, method="class")
```

```
plot(rpart_model);text(rpart_model)
```

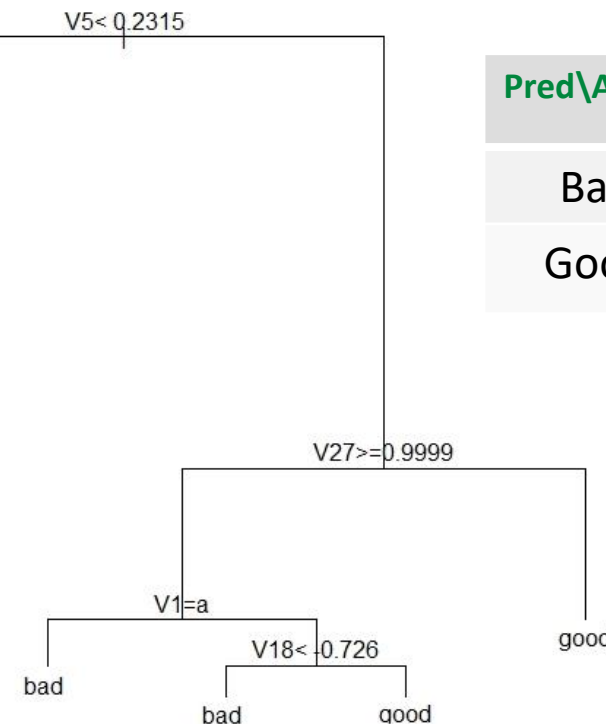
#predict on test data

```
rpart_predict <- predict(rpart_model,testset[, -typeColNum],type="class")
```

#confusion matrix

```
table(pred=rpart_predict,true=testset$Class)
```

Pred\Actual	Bad	Good
Bad	17	2
Good	9	43



Example (cont'd)

```
#cost-complexity pruning
printcp(rpart_model)
```

- Best Tree \longleftrightarrow Best Cp
 - Lowest cross-validate relative error OR
 - the smallest (simplest) tree within one standard error of the best tree.

```
# get index of CP with lowest xerror
```

```
opt <- which.min(rpart_model$cptable[, "xerror"])
```

```
cp <- rpart_model$cptable[opt, "CP"]
```

```
#prune tree, plot tree
```

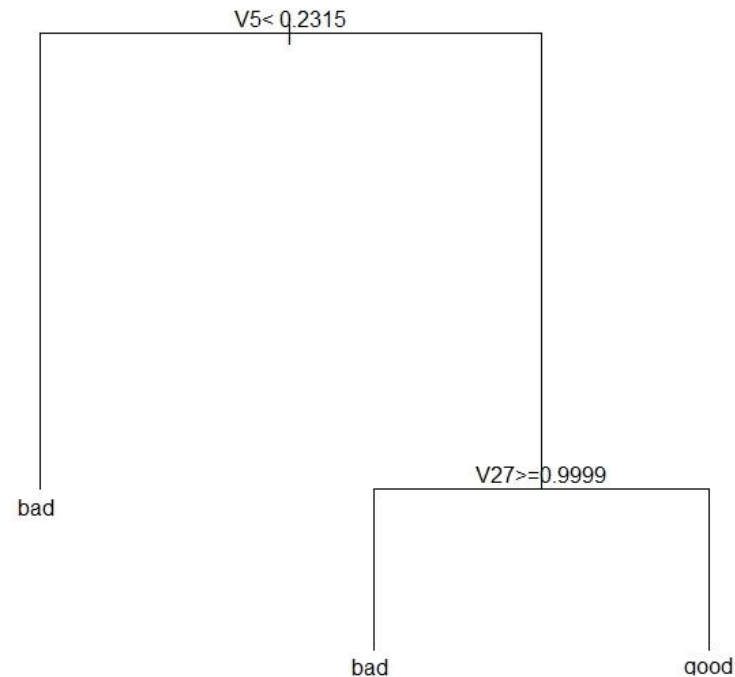
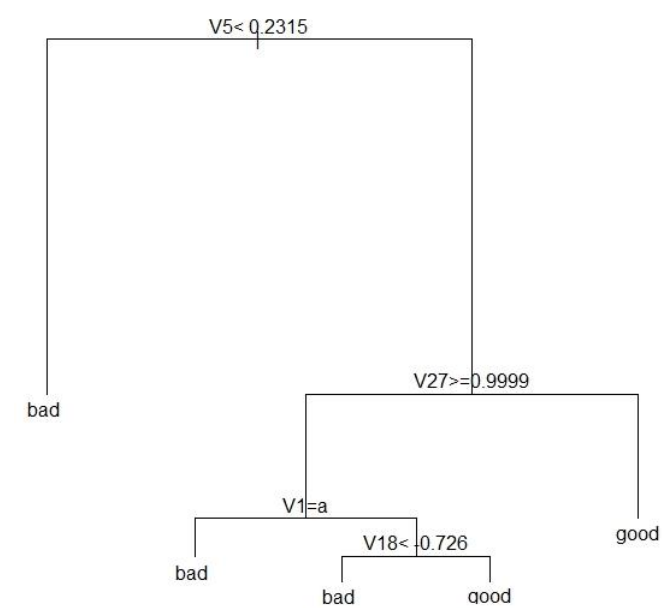
```
pruned_model <- prune(rpart_model, cp)
```

```
plot(pruned_model); text(pruned_model)
```

Cross validation Error Rate (10-fold)
Scaled (Relative) w.r.t. Root Node

Cp	nsplit	rel. error	xerror	xstd
0.57	0	1.00	1.00	0.080178
0.20	1	0.43	0.46	0.062002
0.02	2	0.23	0.26	0.048565
0.01	4	0.19	0.35	

Training Error Rate Standard Error



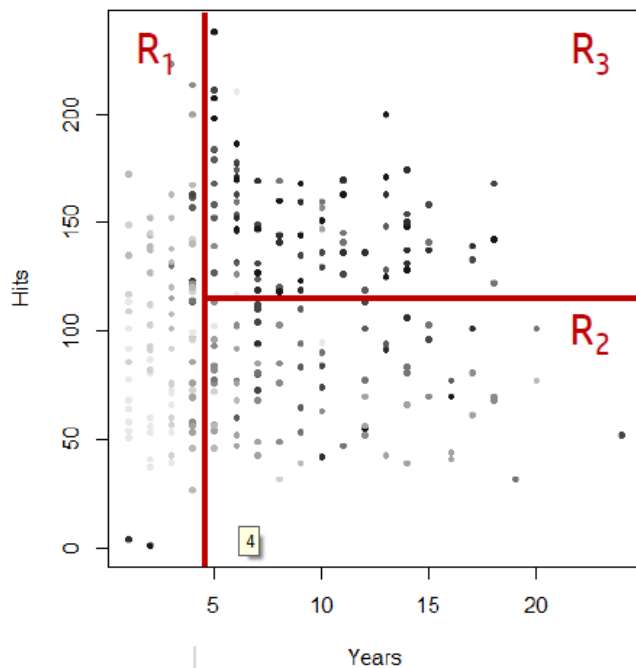
<https://eight2late.wordpress.com/2016/02/16/a-gentle-introduction-to-decision-trees-using-r/>



Decision Trees for Regression

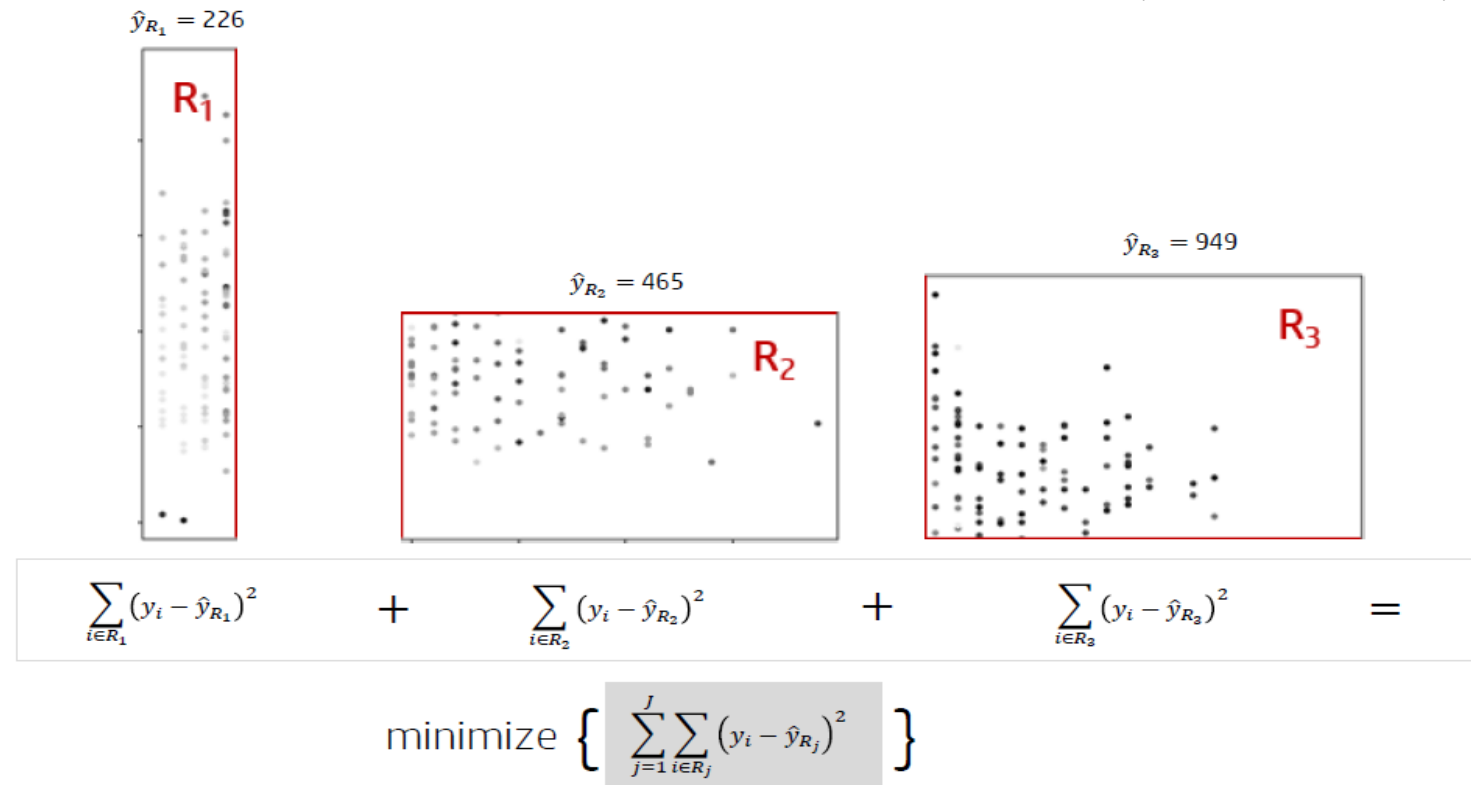
What is a good split?

- Among all possible splits (*all features, all split points*)
- Which split maximizes gain / minimizes error (*Greedy*)
- Information Gain / Impurity reduction



Choosing feature, split-point

- Contain “homogeneous” data (*subset of data*)
- What is a good split measure?
- Squared Sum of Errors $\sum_{i \in L} (\hat{y}_L - y_{i,L})^2 + \sum_{i \in R} (\hat{y}_R - y_{i,R})^2$



Decision Trees : Visualization

Splits = Branching

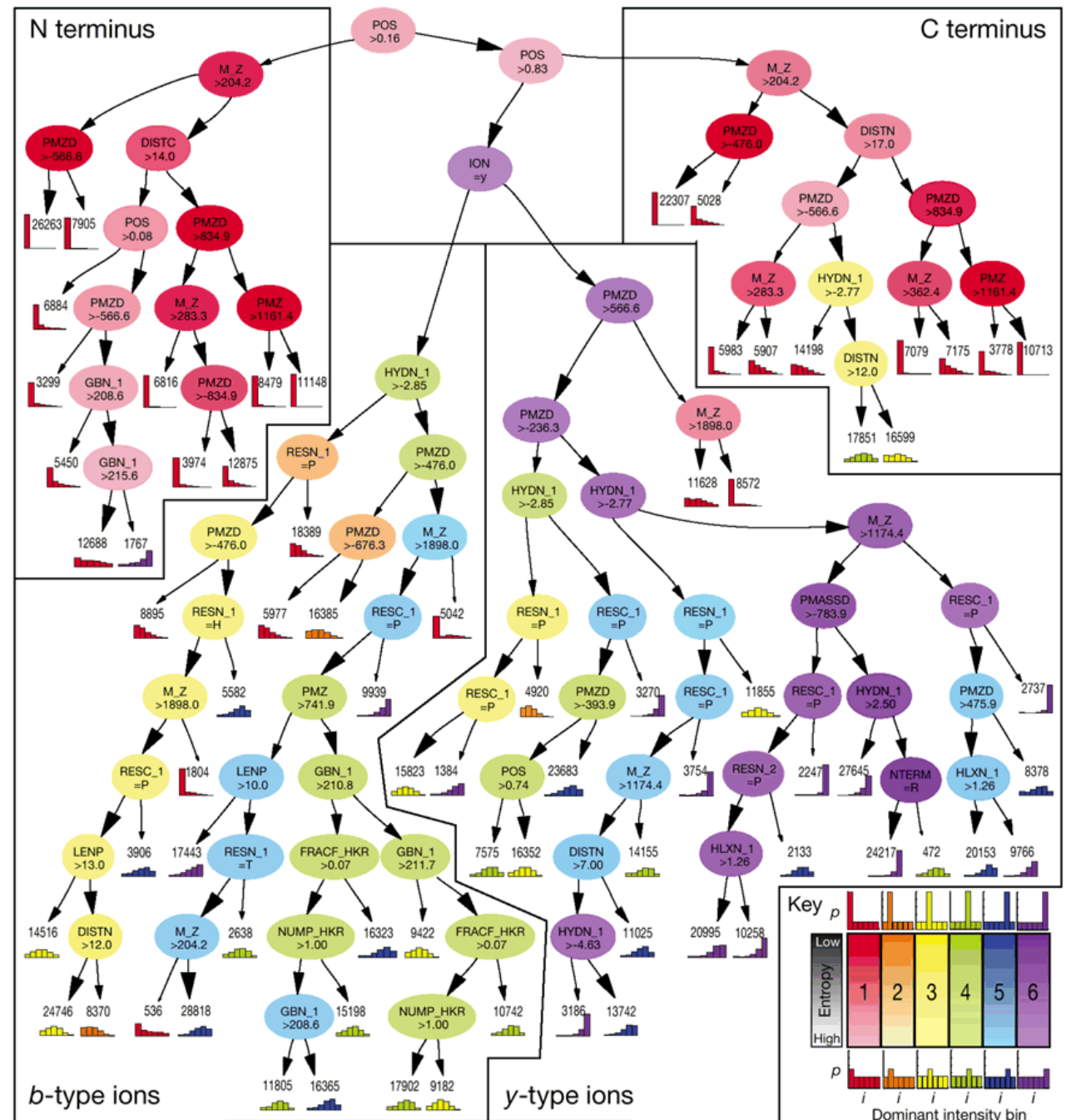
- Split = Feature, Split point

Information gain (Entropy) = Colour

- In the dominant intensity bin

Leaf Distribution = Data Homogeneity

- Some leaves are better than others



Decision Trees : Summary

Versatility

- Can be used for classification, regression & clustering
- Effectively handle missing values.
- Can be adapted to streaming data.

Predictive Accuracy

- Not so great.
- But : Bagging, Boosting, Random Forests

Interpretability

- Easy to understand / present / visualize
- Human interpretable rules
- Allow post processing: Rules systems

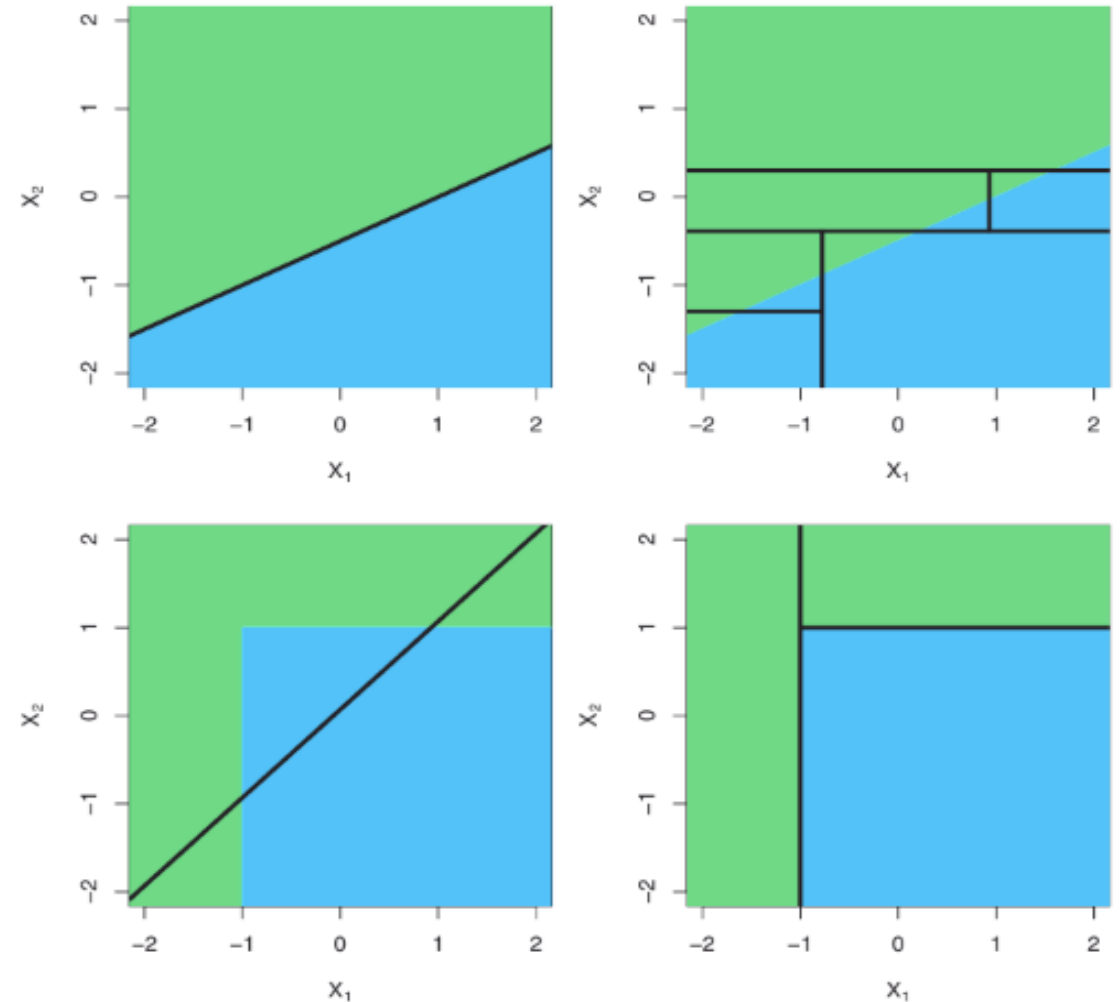
Model Stability

- High Variance: Strong dependence on training set.
- But : Bagging, Boosting, Random Forests



Decision Trees vs. Linear Regression (Separating Hyperplane)

- Linear Regression
 - Linear: y is a linear combination of its features
 - The separating boundary is a hyperplane
- Decision Tree
 - The separating boundary is piecewise linear along one of the features
 - Keep splitting the feature spaces till variance in the dependent variable is low enough
- $Y = f(X)$



Q?

Praphul Chandra

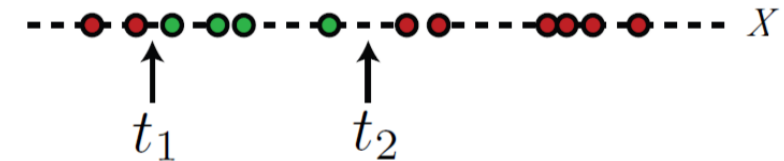
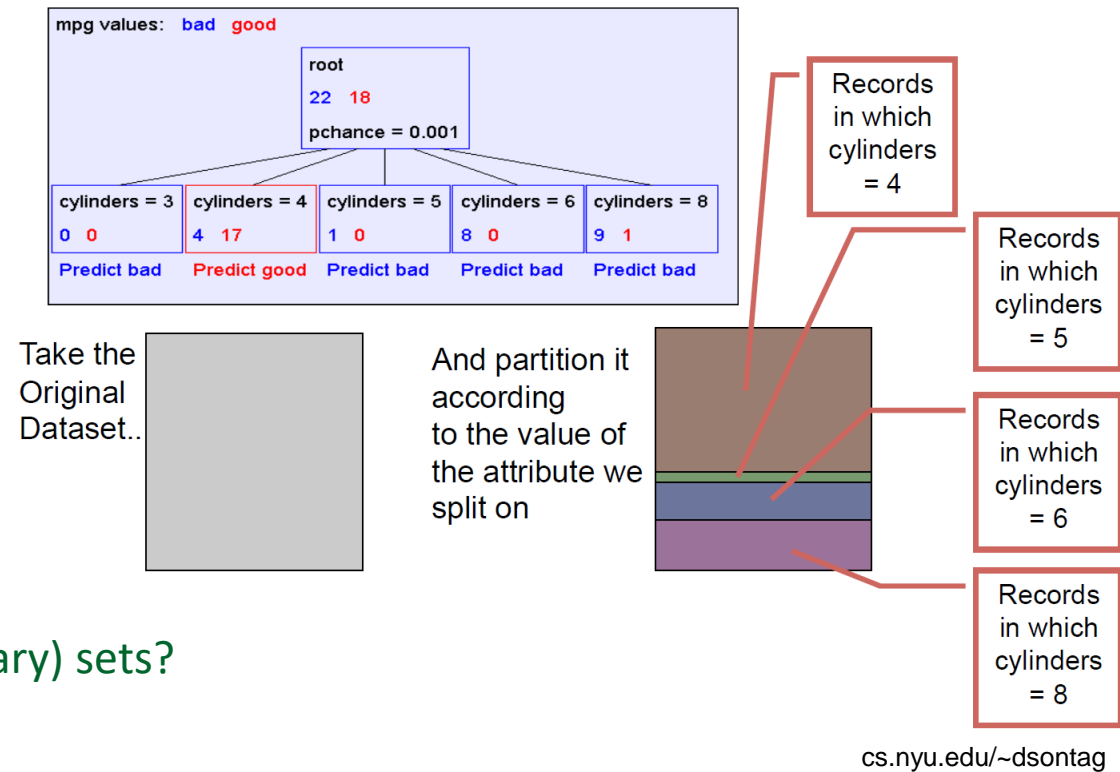
Insofe

Bangalore, Hyderabad



Splitting based on Categorical variable

- How many potential split points possible?
 - Determines the computational complexity
- Categorical variable with k factors
 - In how many ways can you split the data into two (binary) sets?
 - kC_2 (k if we restrict split points to be “k = t and k ≠ t”)
- Can do better
 - Sort the attributes that you can split on.
 - Find all the "breakpoints" where the class labels associated with them change.
 - Consider the split points where the labels change.



Splitting based on Numeric variable

- How many potential split points possible?
 - Infinite??
- What is the range of values of the numeric variable
 - Consider split points of the form $t = x_i + (x_{i+1} - x_i)/2$
 - One branch: $< t$; Other branch $\geq t$
- Can do better
 - Sort the attributes that you can split on.
 - Find all the "breakpoints" where the class labels associated with them change.
 - Consider the split points where the labels change.

