



Inspire...Educate...Transform.

# Methods and Algorithms in Machine Learning

## Decision Trees

**Dr. Sridhar Pappu**  
Executive VP – Academics, INSOF

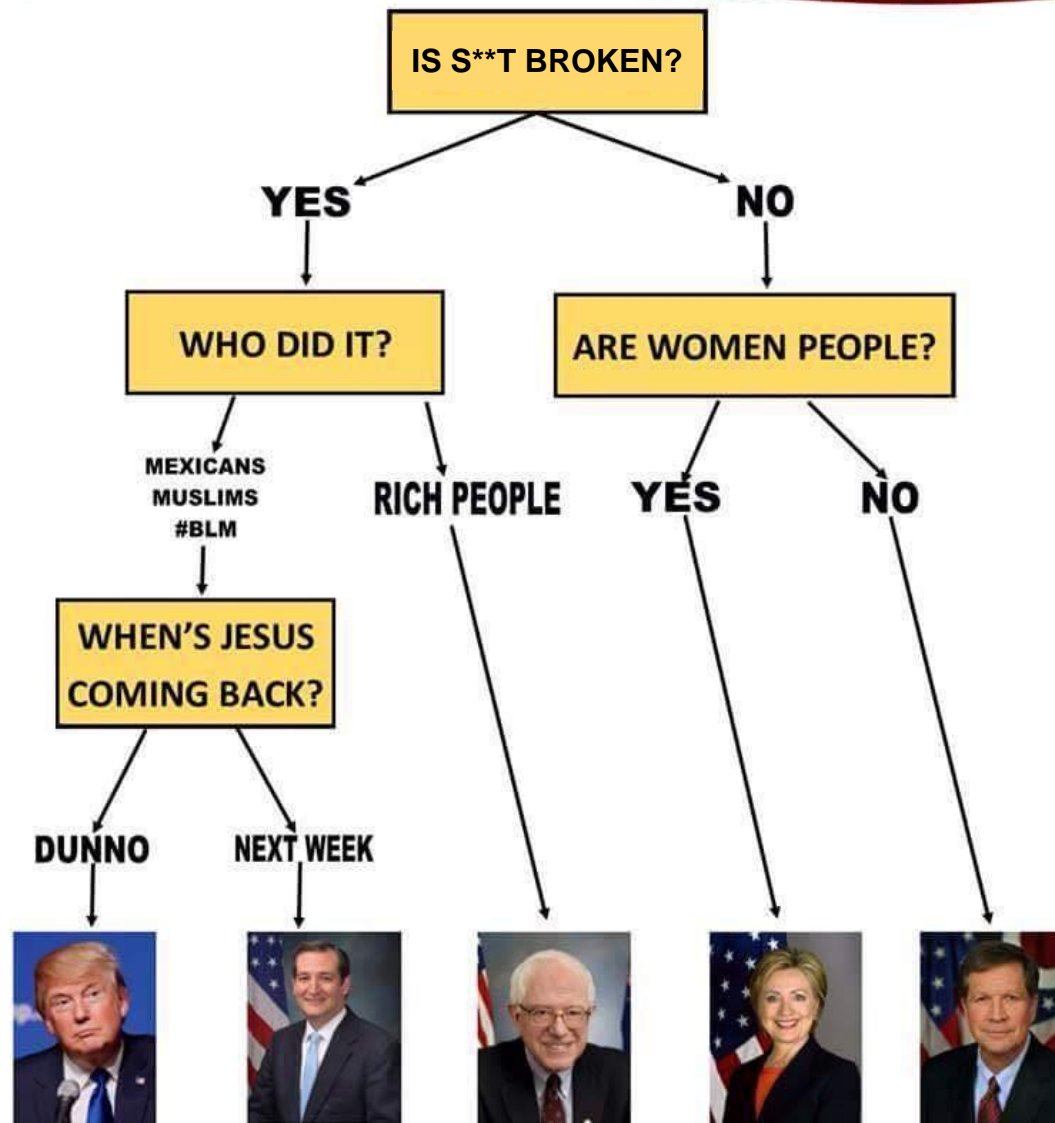
February 04, 2017

# And the awards go to...

- **#10: CART (34 votes)**
- **#7: Naive Bayes (45)**
- **#7: kNN (45)**
- **#7: AdaBoost (45)**
- **#6: PageRank (46)**
- **#5: EM (Expectation Maximization) (48)**
- **#4: Apriori (52)**
- **#3: SVM (58)**
- **#2: K-Means (60)**
- **#1: C4.5 (61)**

# DECISION TREES

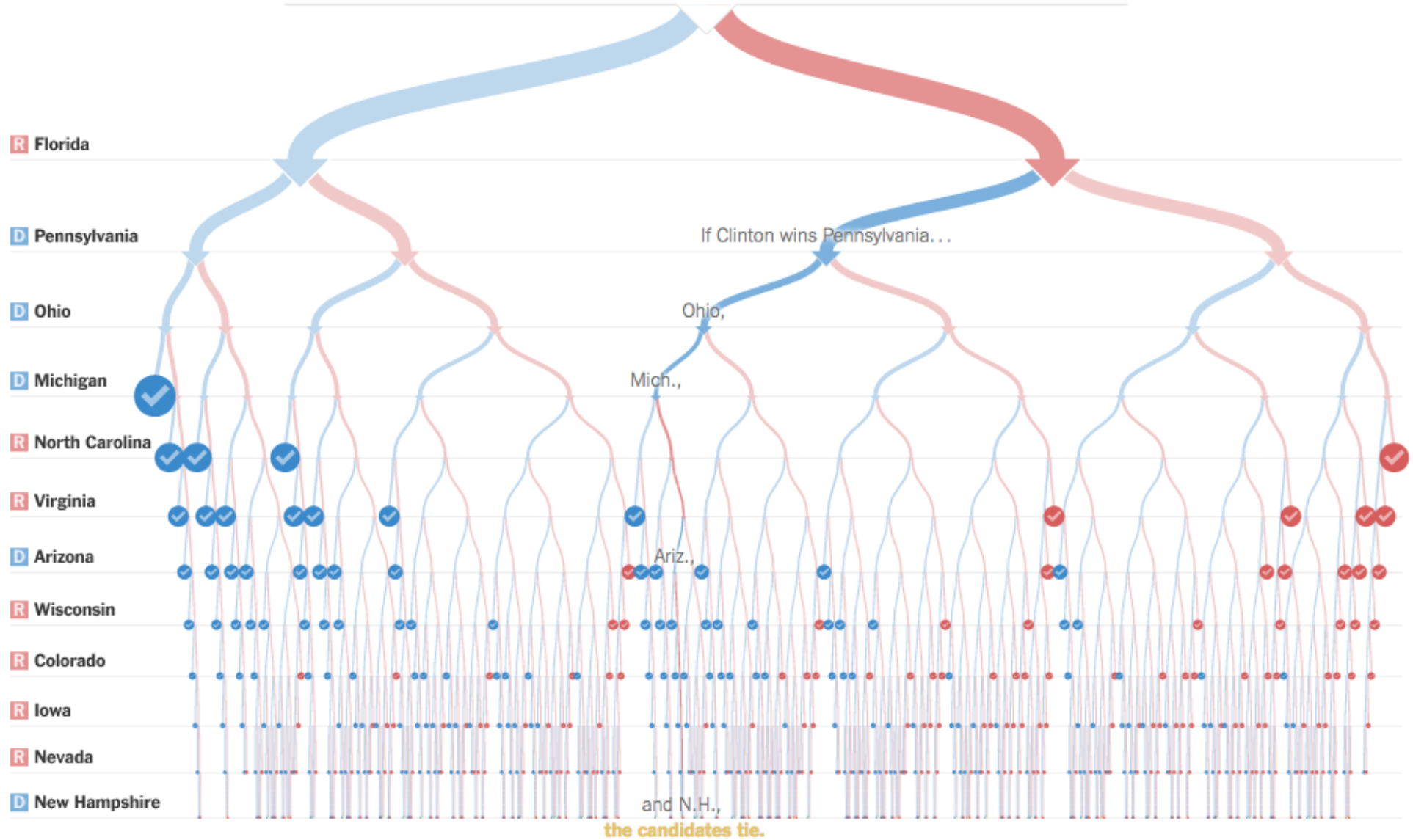
# Who should I vote for?



Clinton has **2,586** ways to win  
63% of paths

**59** ties  
1% of paths

Trump has **1,451** ways to win  
35% of paths



Clinton has **9** ways to win

14% of paths

**0** ties

0% of paths

Trump has **55** ways to win

86% of paths

✕ Reset

**D** Pennsylvania

If Clinton wins Pennsylvania...

**D** Michigan

Mich.,

**D** Wisconsin

Wis.,

**R** Iowa

**D** Nevada

and Nev.,

New Hampshire

Clinton wins.

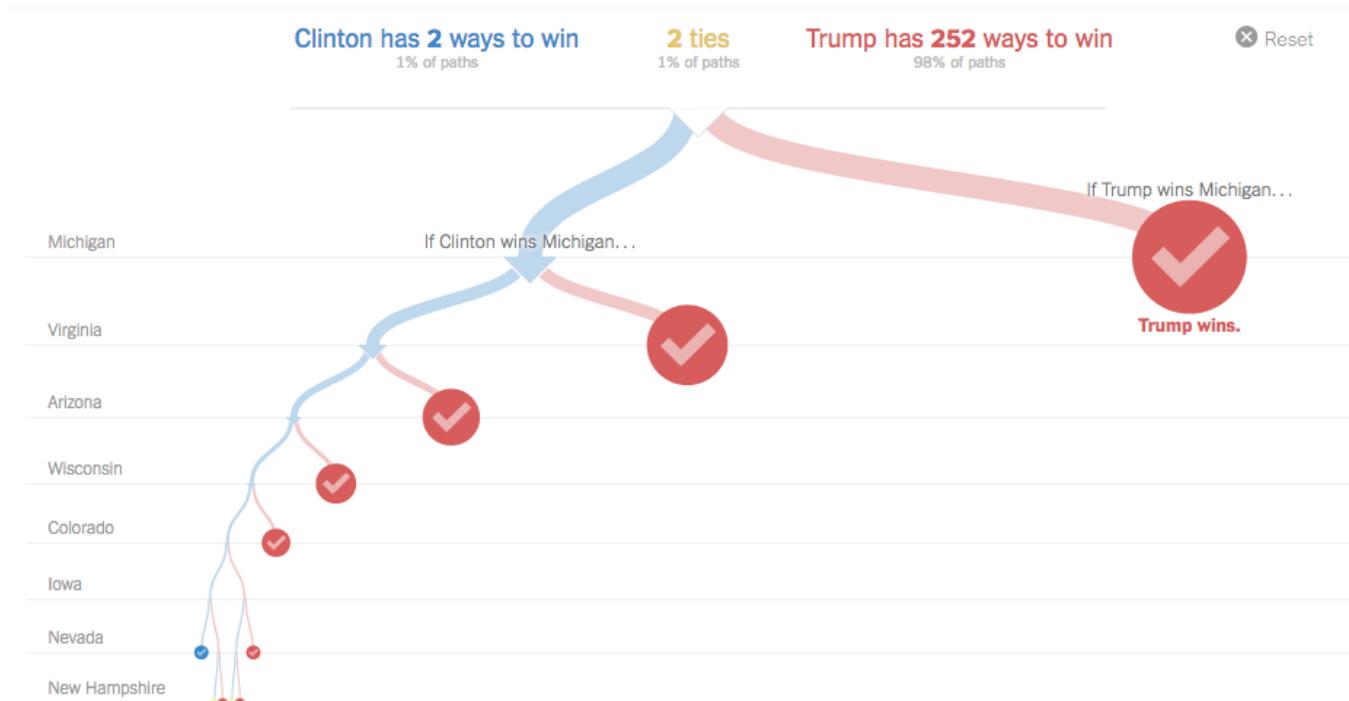
3 7 4 0 5 C



# The Ways Clinton or Trump Can Win the Election

Select a winner in the states below to see the paths to victory available for the candidates. **UPDATED** November 8, 2016

Fla. 95% Rep.	Pa. 65% Rep.	Ohio 95% Rep.	Mich. 68% Rep.	N.C. 95% Rep.
<input type="button" value="Dem"/> <input checked="" type="button" value="Rep"/>	<input type="button" value="Dem"/> <input checked="" type="button" value="Rep"/>	<input type="button" value="Dem"/> <input checked="" type="button" value="Rep"/>	<input type="button" value="Dem"/> <input type="button" value="Rep"/>	<input type="button" value="Dem"/> <input checked="" type="button" value="Rep"/>
Va. 95% Dem.	Ariz. 95% Rep.	Wis. 90% Rep.	Colo. 95% Dem.	Iowa 74% Rep.
<input type="button" value="Dem"/> <input type="button" value="Rep"/>	<input type="button" value="Dem"/> <input type="button" value="Rep"/>	<input type="button" value="Dem"/> <input type="button" value="Rep"/>	<input type="button" value="Dem"/> <input type="button" value="Rep"/>	<input type="button" value="Dem"/> <input type="button" value="Rep"/>
Nev. 64% Dem.		N.H. 64% Dem.		
<input type="button" value="Dem"/> <input type="button" value="Rep"/>		<input type="button" value="Dem"/> <input type="button" value="Rep"/>		



# Induction of Decision Trees

- Data Set (Learning Set)
  - Each example = Attributes + Class
- Induced description = Decision tree
- TDIDT
  - Top Down Induction of Decision Trees
- Recursive Partitioning



# Some TDIDT Systems

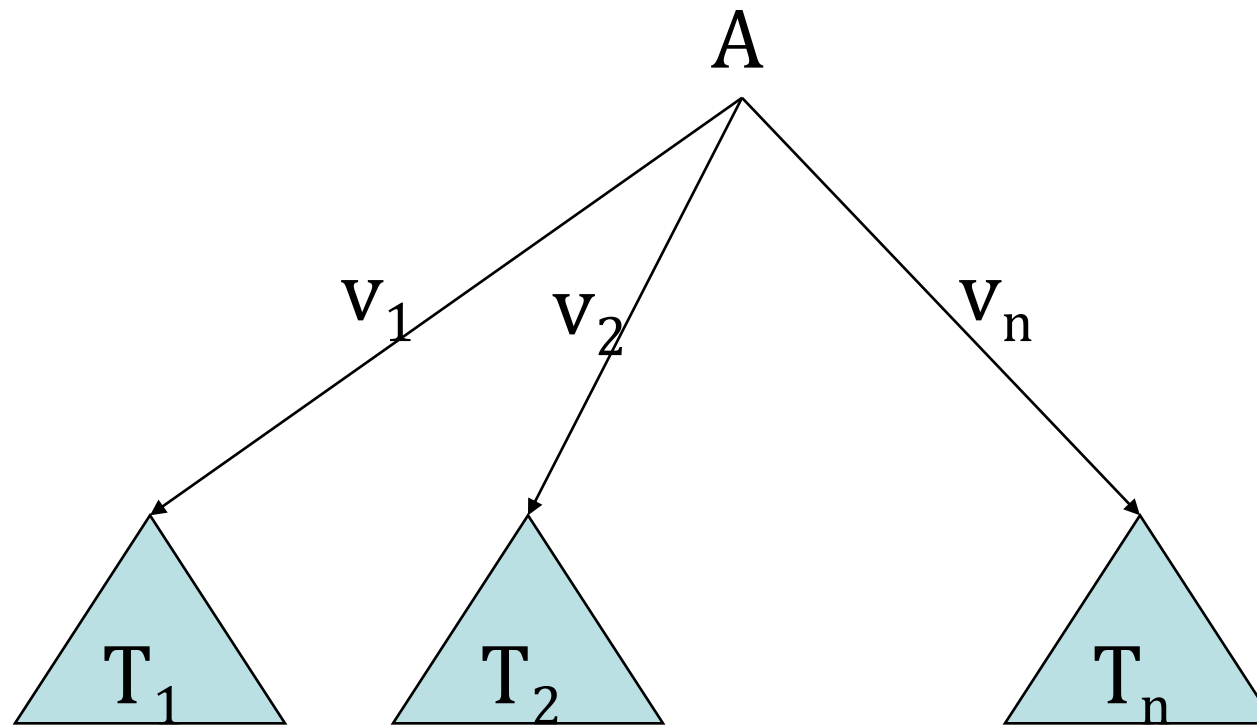
- ID3 (Iterative Dichotomizer) (Quinlan)
- CART (Breiman et al.)
- Assistant (Cestnik et al.)
- C4.5 (Quinlan)
- See5/C5.0 (Quinlan)
- ...

# TDIDT Algorithm

- Also known as ID3 (Quinlan)
- To construct decision tree  $T$  from learning set  $S$ :
  - **If** all examples in  $S$  belong to some class  $C$  **Then** make leaf labeled  $C$
  - **Otherwise**
    - select the “most informative” attribute  $A$
    - partition  $S$  according to  $A$ ’s values
    - recursively construct subtrees  $T_1, T_2, \dots$ , for the subsets of  $S$

# TDIDT Algorithm

Resulting tree  $T$  is:



Attribute  $A$

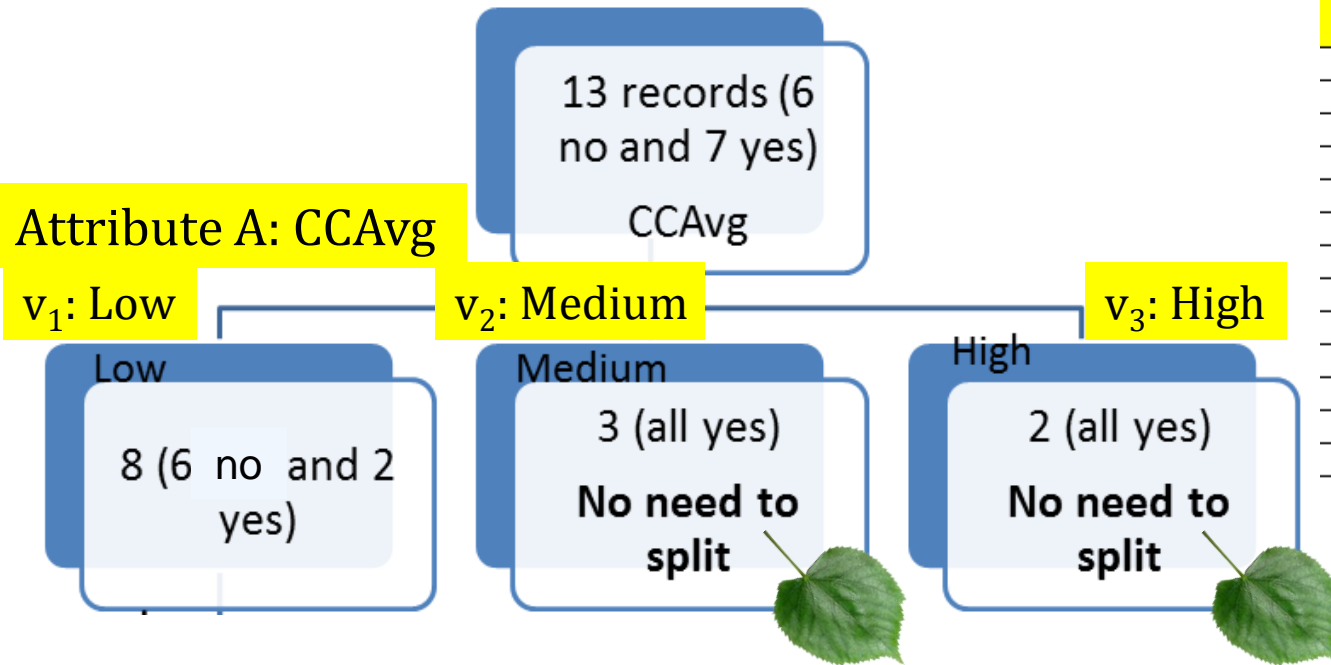
$A$ 's values

Subtrees

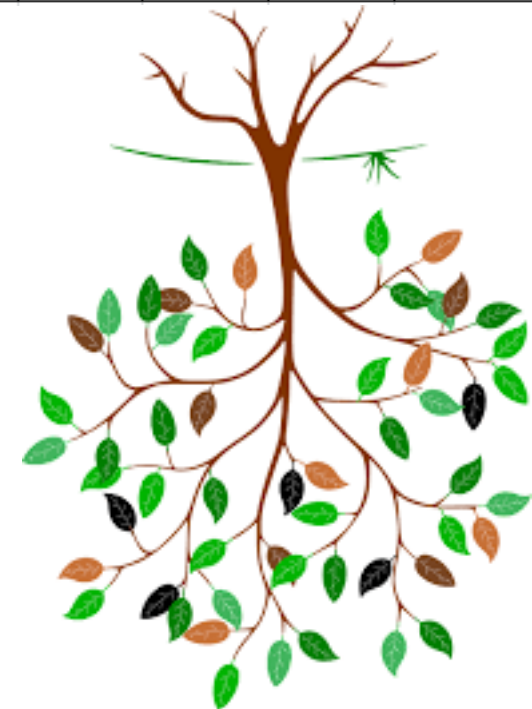
# Data

ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

# Constructing a Tree

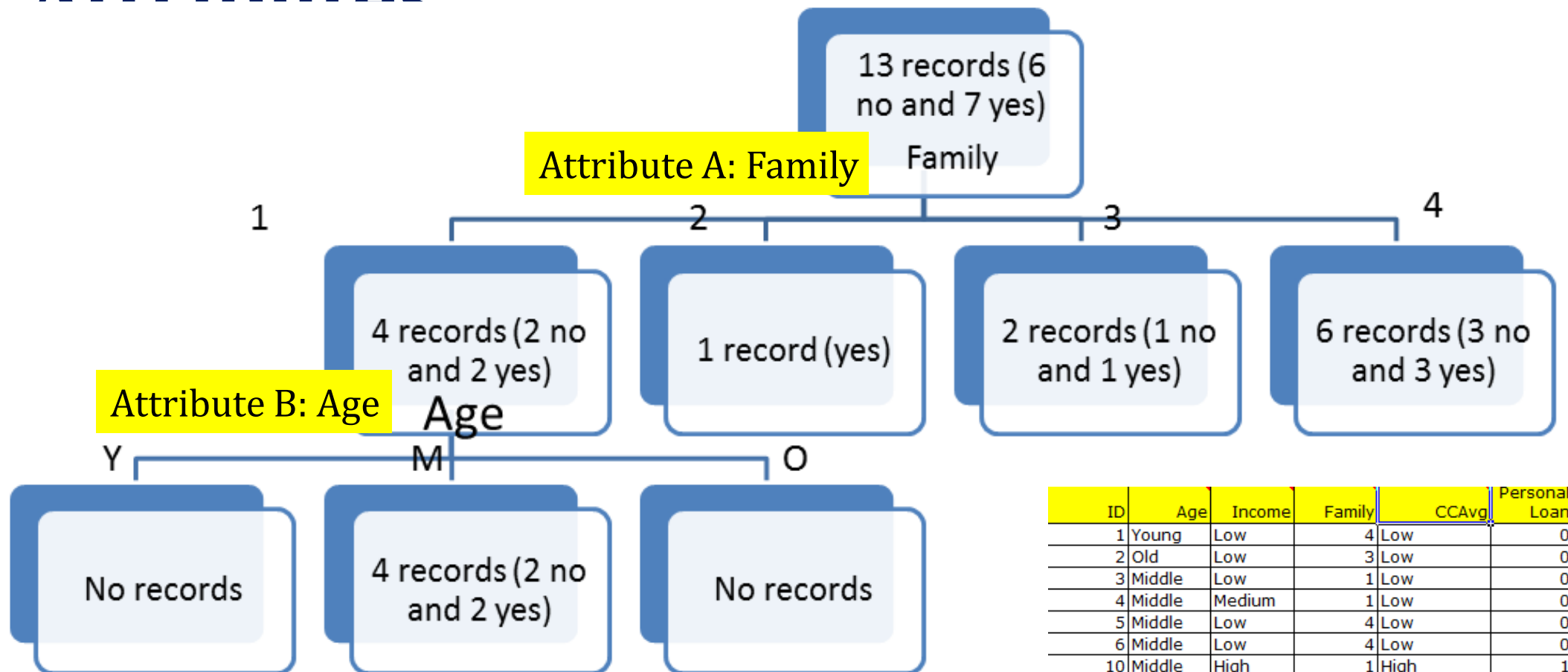


ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1



**Nodes (root node):** Test/Decision points  
**Leaves:** Final Decisions / Conclusion  
**Branch:** Collection of nodes and the leaf

# Decision Trees with Different Attributes



ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1



# Decision Trees with Different Attributes

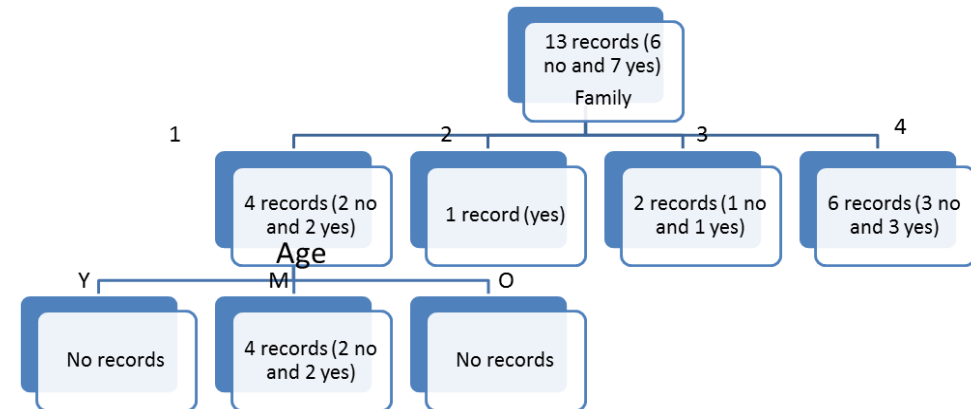
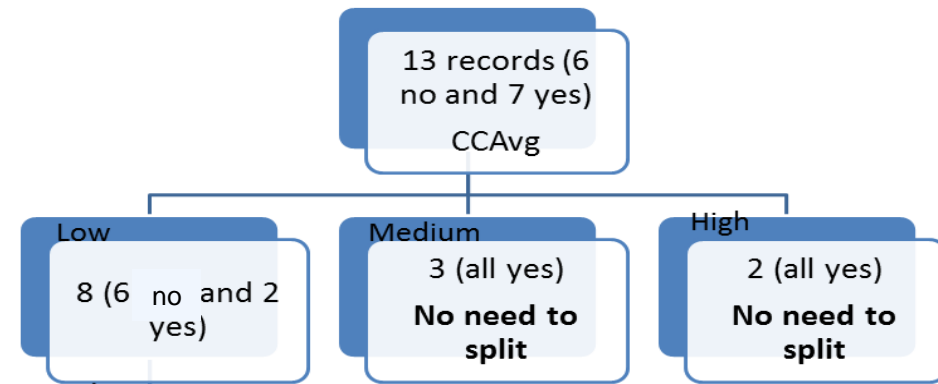
- Decision points (Divide-and-Conquer)
  - Deciding where to start (Selection of the root node)
  - Deciding when to stop (To avoid overfitting)

# UNDERSTANDING DECISION TREES

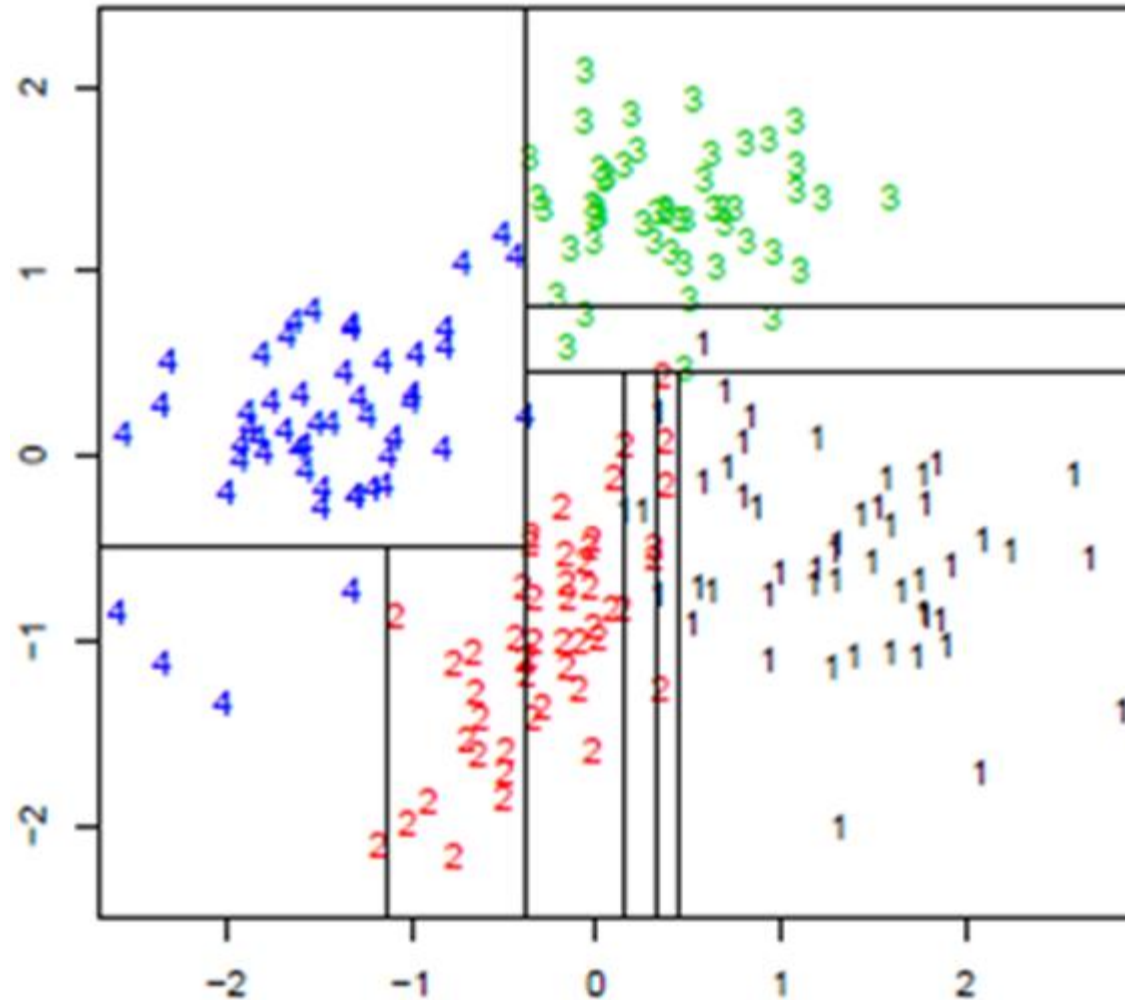


# Trees are Rules Expressed as Disjunctive Normal Form

- *If (ccAvg is Medium) or (CCAvg is High) then (Loan = Yes)*
  - Within branch nodes are connected with “and” and branches with similar outcome are connected with “or”
- Disjunctive Normal Form
  - Disjunction (*or*) of conjunction (*and*) clauses



# Geometry of Decision Trees: Axis Parallel Search



# Hypothesis Oriented Analysis

- There is a possible space of decision trees (hypotheses) with all combinations of attributes (e.g., selecting the root node as Family, CCAvg, etc.) (*a la* linear regression hypotheses with various possible line fits)
- We need an algorithm that searches for the best tree (*a la* least squares in linear regression to find the line of best fit)
- The best tree:
  - Occam's razor: Smallest tree (least number of nodes) with smallest error (least number of incorrectly classified

# Advantages of Trees

- Fast
- Robust
- Explicable
- Require very little experimentation
- You may also build some intuitions about your customer base, e.g. “Are customers with different family sizes truly different?”

# Regression Trees

- Can we use a decision tree only for classification or can we use them for forecasting or predicting a numeric attribute?

# Regression Trees

- It turns out that, we are collecting very similar records at each leaf. So, we can use median or mean of the records at a leaf as the predictor value for all the new records that obey similar conditions. Such trees are called Regression Trees.

# CONSTRUCTING A DECISION TREE

# Two Aspects

- Which attribute to choose?
- Where to stop?

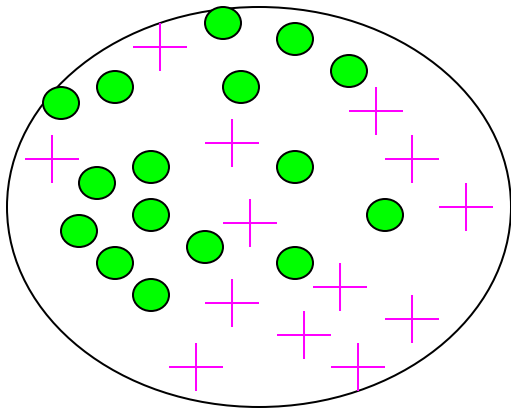


# Attribute Selection Criteria

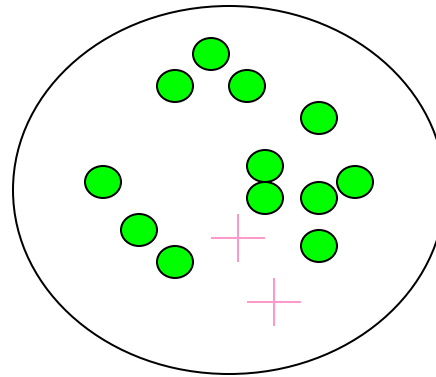
- Main principle
  - Select attribute which partitions the learning set into subsets as “pure” as possible
- Various measures of purity
  - Information-theoretic
  - Gini index
  - $\chi^2$
  - ...
- Various improvements
  - probability estimates
  - normalization
  - binarization, subsetting

# Impurity

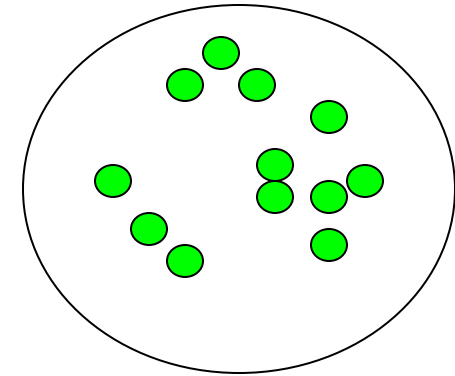
High impurity



Less impurity



Minimum impurity

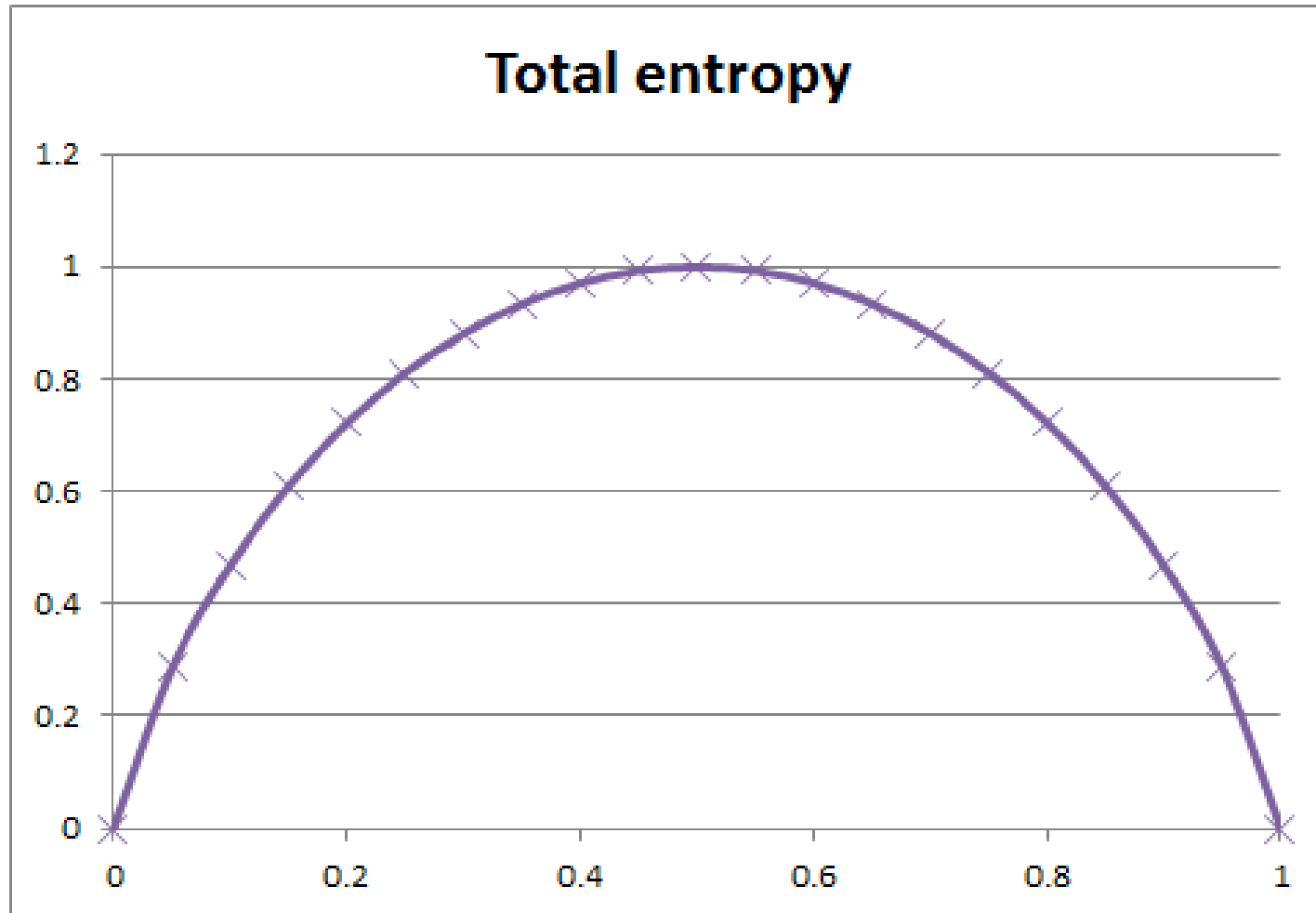


# Classification Trees:

## Entropy

- Entropy of information is a measure of the randomness or uncertainty or impurity of the outcome.
- Entropy
  - Let us say, I am considering an action like a coin toss. Say, I have five coins with **probabilities for heads** 0, 0.25, 0.5, 0.75 and 1. When I toss them, which one has highest uncertainty and which one has the least?

# Entropy: A measure of randomness

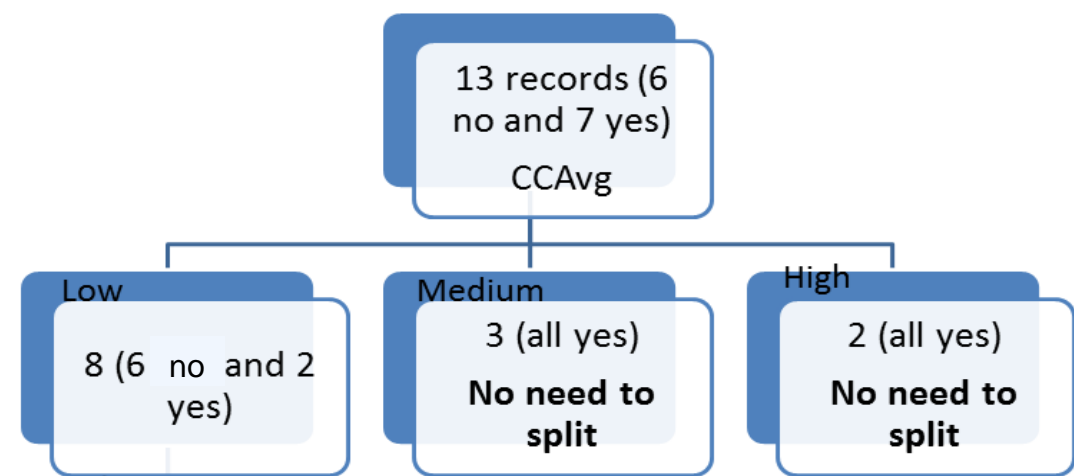


# Entropy and Information Gain (C5.0)

- Shannon's entropy,  $\mathbf{H} = -\sum_i p_i \log_2 p_i$   
 $\mathbf{H}$  is the Greek capital letter, eta.
- Information gain = Entropy of the system



*(Recall a posteriori or Frequentist approach to calculating probabilities)*



Entropy before split in our example

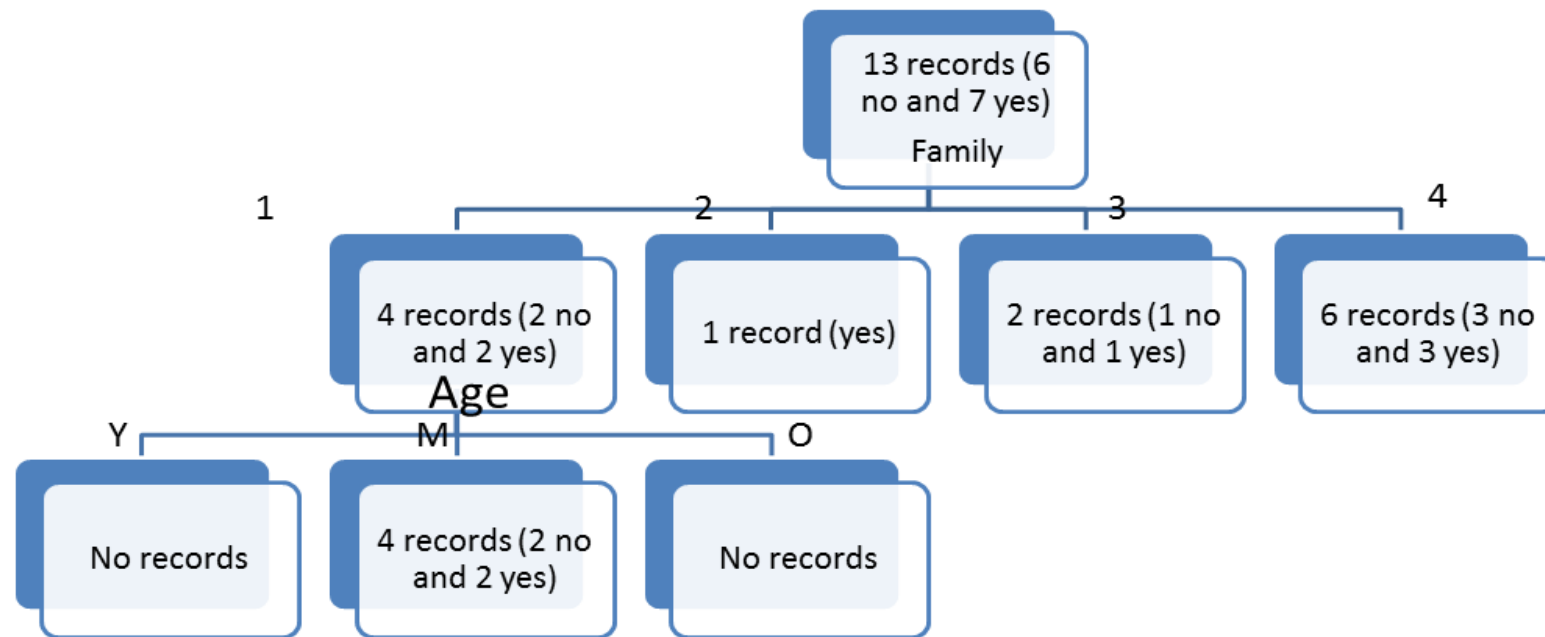
$$H = -\frac{6}{13} * \log_2 \frac{6}{13} - \frac{7}{13} * \log_2 \frac{7}{13} = 0.9957$$

Entropy (Weighted) after split on CCAvg

$$H = \frac{8}{13} \left( -\frac{6}{8} * \log_2 \frac{6}{8} - \frac{2}{8} * \log_2 \frac{2}{8} \right) + \frac{3}{13} \left( -\frac{3}{3} * \log_2 \frac{3}{3} \right) + \frac{2}{13} \left( -\frac{2}{2} * \log_2 \frac{2}{2} \right) = 0.4992$$

$$\text{Information Gain} = 0.9957 - 0.4992 = 0.4965$$





Similar calculation for information gain when splitting on Family gives  
 Information Gain = 0.0726



# Sometimes Information Gain Fails

Let us do information gain for split on ID

ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1



# Entropy after split

Now, the system will have 13 splits, one for each ID.

$$\text{Entropy} = -1 * \text{LOG}(1,2) = 0$$

Entropy of the total system after split is the weighted average of the individual parts= 0

Aha! Information gain is the highest (0.9957), compared to all other attributes.

# Is ID the root attribute?

- An attribute with many more states is likely to have less variation in each state. So, it will always give better information gain.
- So, we need to normalize it to get something like information gain per state.

# Information Content

- Information content is defined as  $= - \sum f_i \log f_i$ .  
We only want to know fraction of the members in a state (# of members divided by the total members).
- Information content of ID: It has 13 states. So, the information content

$$= - 1/13 * \text{LOG}(1/13, 2) * 13 = 3.7$$



# Gain Ratio – Used in C4.5/C5.0

- Information Gain is biased towards attributes with many values (levels)
- Gain Ratio normalizes Information Gain by dividing by the Information Content at the attribute

$$GainRatio(A) = \frac{InformationGain(A)}{InformationContent(A)}$$

- Attribute with the maximum Gain Ratio is selected as the splitting attribute



# Gain Ratio

- Gain Ratio for ID = 0.27
- Gain Ratio for ccAvg = 0.37



# Gini Index – Used in CART

$$1 - \sum_{i=1}^m p_i^2$$

It is computed on binary splits only.

So, if we take ccAvg (low, medium and high), it considers all binary options {low}, {medium, high} OR {medium}, {low, high}, etc.

Is a low or a high Gini preferred?

# Gini Index

$$1 - \sum_{i=1}^m p_i^2$$

ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

$$\text{Gini Index before split} = 1 - \left(\frac{6}{13}\right)^2 - \left(\frac{7}{13}\right)^2 = 0.497$$

$$\begin{aligned} &\text{Gini after split with } \{\text{Low}\} \text{ and } \{\text{Medium, High}\} \\ &= \frac{8}{13} \left( 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 \right) + \frac{5}{13} \left( 1 - \left(\frac{5}{5}\right)^2 \right) = 0.231 \end{aligned}$$

Calculated similarly for other binary splits

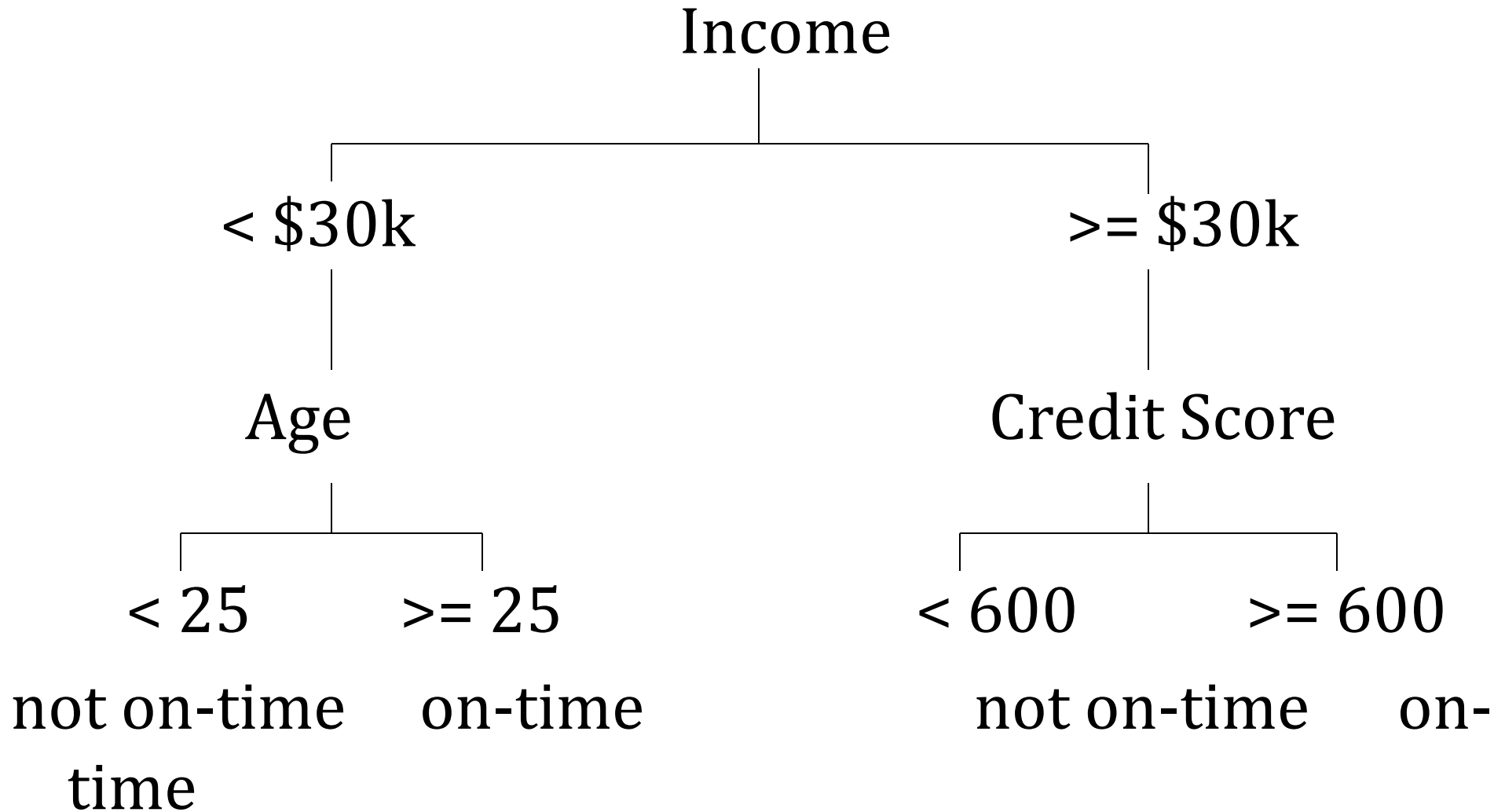
The one that gives the least Gini Index is picked

$$\begin{aligned} &\text{Gini Index before split} = 1 - \left(\frac{6}{13}\right)^2 - \left(\frac{7}{13}\right)^2 = 0.497 \\ &\text{Gini after split with } \{\text{Low}\} \text{ and } \{\text{Medium, High}\} \\ &= \frac{8}{13} \left( 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 \right) + \frac{5}{13} \left( 1 - \left(\frac{5}{5}\right)^2 \right) = 0.231 \end{aligned}$$

Calculated similarly for other binary splits

The one that gives the least Gini Index is picked

# Chi-Square – Loan Payment Example





# Evaluating the Partitions

- When the target is categorical, for each partition of an input variable a chi-square statistic is computed
- A contingency table is formed that maps responders and non-responders against the partitioned input variable

# Evaluating the Partitions

- For example, the null hypothesis might be that there is no difference between people with income  $< \$30k$  and those with income  $\geq \$30k$  in making an on-time loan payment
  - The lower the significance or  $p$ -value, the more likely that we reject this hypothesis, meaning that this income split is a discriminating factor

# Contingency Table

	$\$ < 30k$	$\$ \geq 30k$	Total
Payment on-time			
Payment not on-time			
Total			

# Chi-Square Statistic

- The chi-square statistic computes a measure of how different the number of observations is in each of the four cells as compared to the expected number
  - The  $p$ -value associated with the null hypothesis is computed
- The split that generates the lowest  $p$ -value for a given input variable is selected

# Regression Trees - Variance

- Minimizing the variance
  - Variance before split
  - Weighted average of Variance after split
  - The attribute that reduces the variance most is chosen for the node

# Based on Attribute Selection

- Are decision trees greedy?
  - Yes (at each point, what gives the maximum Information Gain is selected)
- Are they likely to find local minima?
  - Yes (may be selecting a split giving lesser Information Gain at a point may yield a better search down the line)
- They are hence unstable (change the data a bit, especially as you get closer to the leaf, and the split may change) and overfit

# A General Thought on Linear vs Non-Linear Models

- If the number of features/variables/attributes are very high and the number of records very low, linear models tend to outperform non-linear models
  - For example, if you have 100,000 features and 100 records, non-linear models will tend to overfit
- If the number of features/variables/attributes are very low and the number of records very high, non-linear models tend to outperform linear models
  - For example, if you have 100 features and 100,000 records, non-linear models will tend to do better
- **In any case, use various techniques to actually see which perform better**

We can grow until we exhaust the data. But is that the right time to stop?

## HOW TO MINIMIZE THE OVERFIT?



# Termination Criteria

- All the records at the node belong to one class
- A significant majority fraction of records belong to a single class (e.g., 99% of the records are buyers)
- The segment contains only one or very small number of records
- The improvement is not substantial enough to warrant making the split

# Why Prune

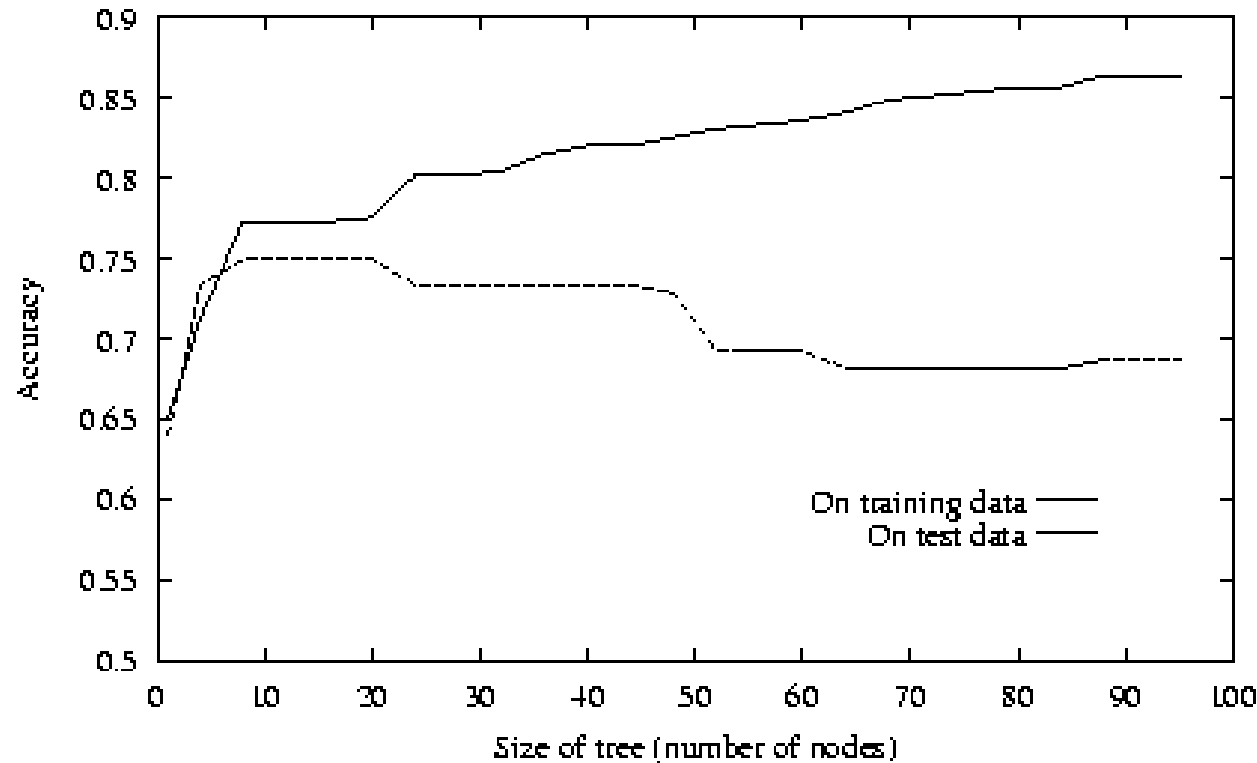
- Avoid overfitting
- The philosophy of **Occam's razor**
  - **Always choose a decision tree that has the optimum combination of size and error.**

# Approaches to Pruning Trees

- Three approaches (using training and testing data)
  - Stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data, and check on test data.
  - Allow the tree to overfit the data, and then post-prune the tree. Chop the bottom layer and check errors on training and testing data.
  - **Allow the tree to overfit the data, transform the tree to *rules* and then post-prune the rules.**

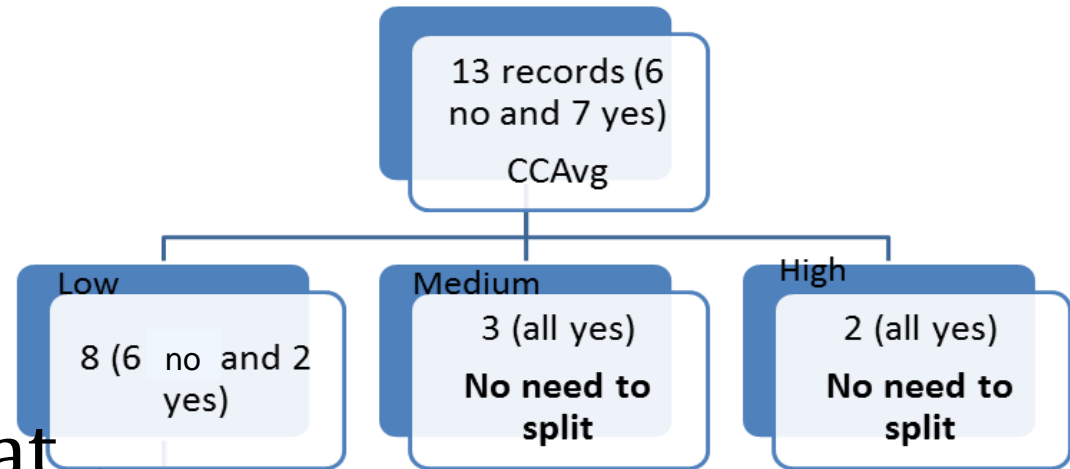
# Minimize Variance

- Build the tree on train data
- Test it on test data
- Plot test and train errors (or accuracy) at various pruning levels



# Reduced Error Pruning

- At each node, analyze the error (probability of making a mistake or fraction of mistakes made during classification) if that node is converted into a leaf of the **majority** class.



- If the error is less than the sum of the errors of its children, then prune the

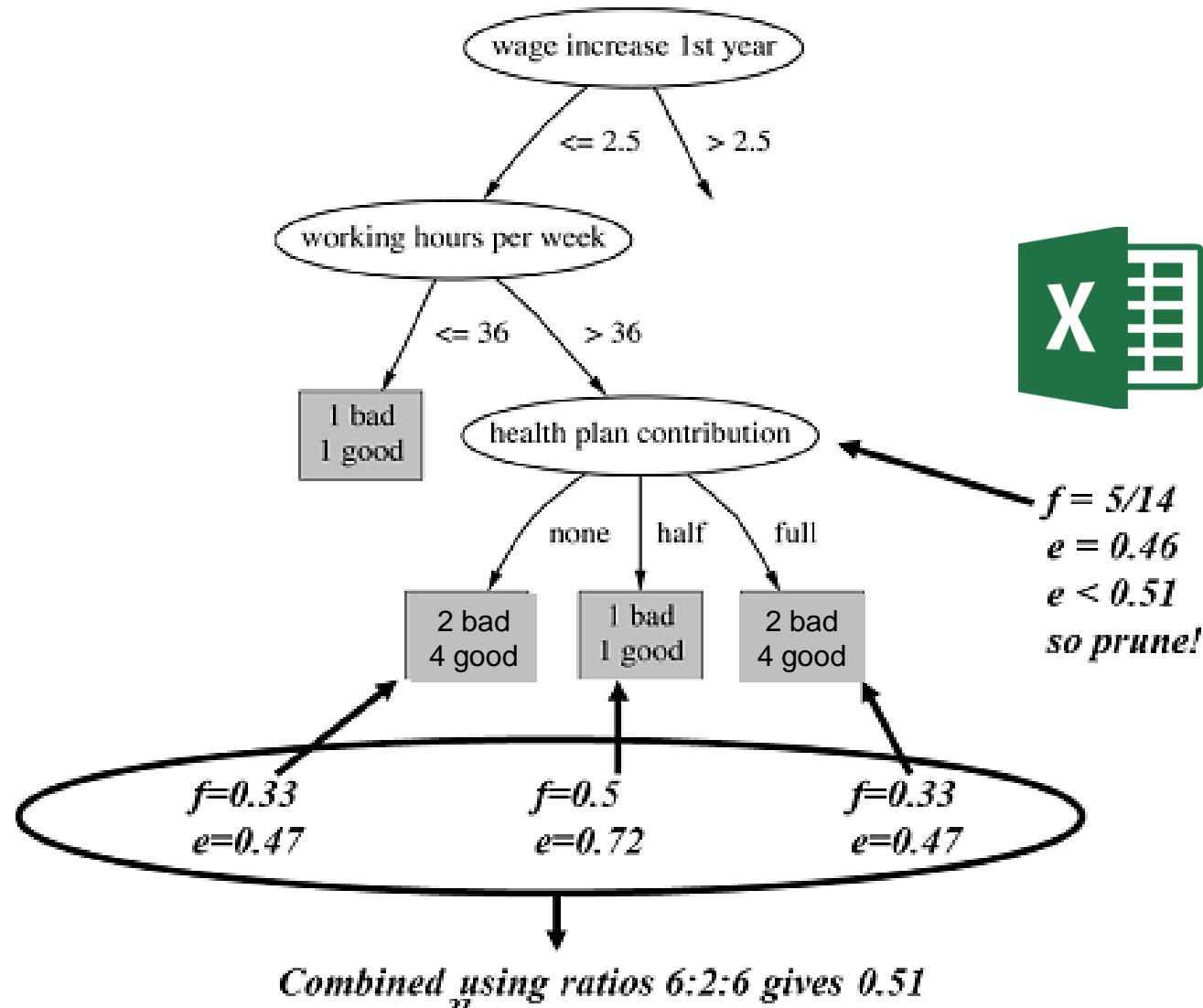
# Pessimistic Pruning

- C4.5 uses a statistical upper bound of error at each node and prunes the node if that upper bound is less than the total error at all the sub-trees generated at the node.

$$e = \left( f + \frac{z^2}{2N} + z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) / \left( 1 + \frac{z^2}{N} \right)$$

where  $f$  is the error on the training data and  $N$  is the number of instances covered by the leaf.  $z$ -score corresponds to the confidence. If confidence = 75%,  $z = 0.67$  (based on normal distribution).

# Pessimistic Pruning



# Cost Complexity Pruning

- $J(\text{Tree}, S) = \text{ErrorRate}(\text{Tree}, S) + \alpha |\text{Tree}|$
- $|\text{Tree}|$  is the number of leaf nodes in the tree and  $\alpha$  is a parameter (complexity parameter) that controls the tradeoff between the error rate and the penalty.
- Play with several values  $\alpha$  starting from 0.  $\alpha = 0$  gives the biggest tree possible.
- Do a K-fold cross-validation on all of them and find the best pruning  $\alpha$ .

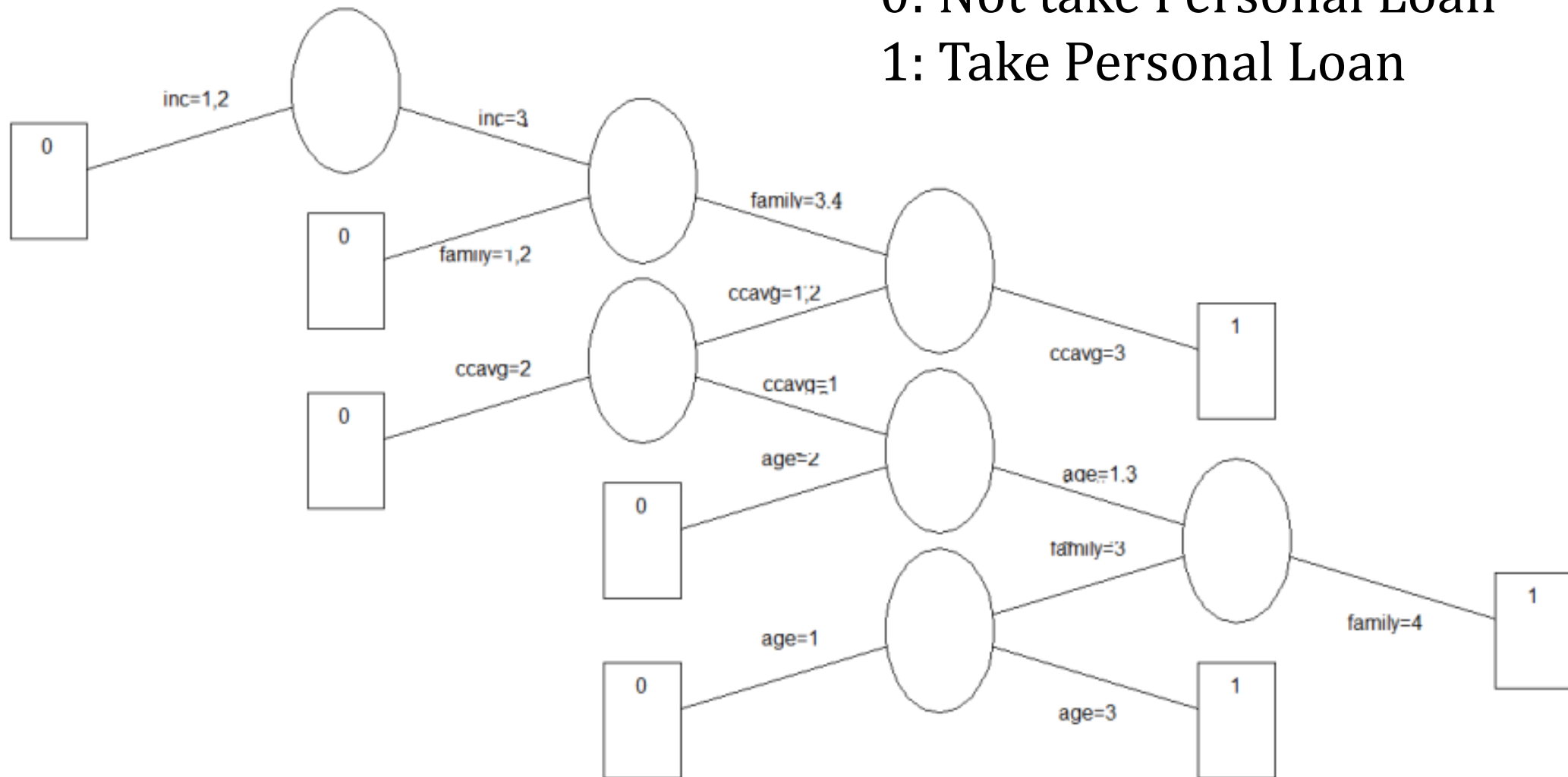


# Two Most Popular Decision Tree Algorithms

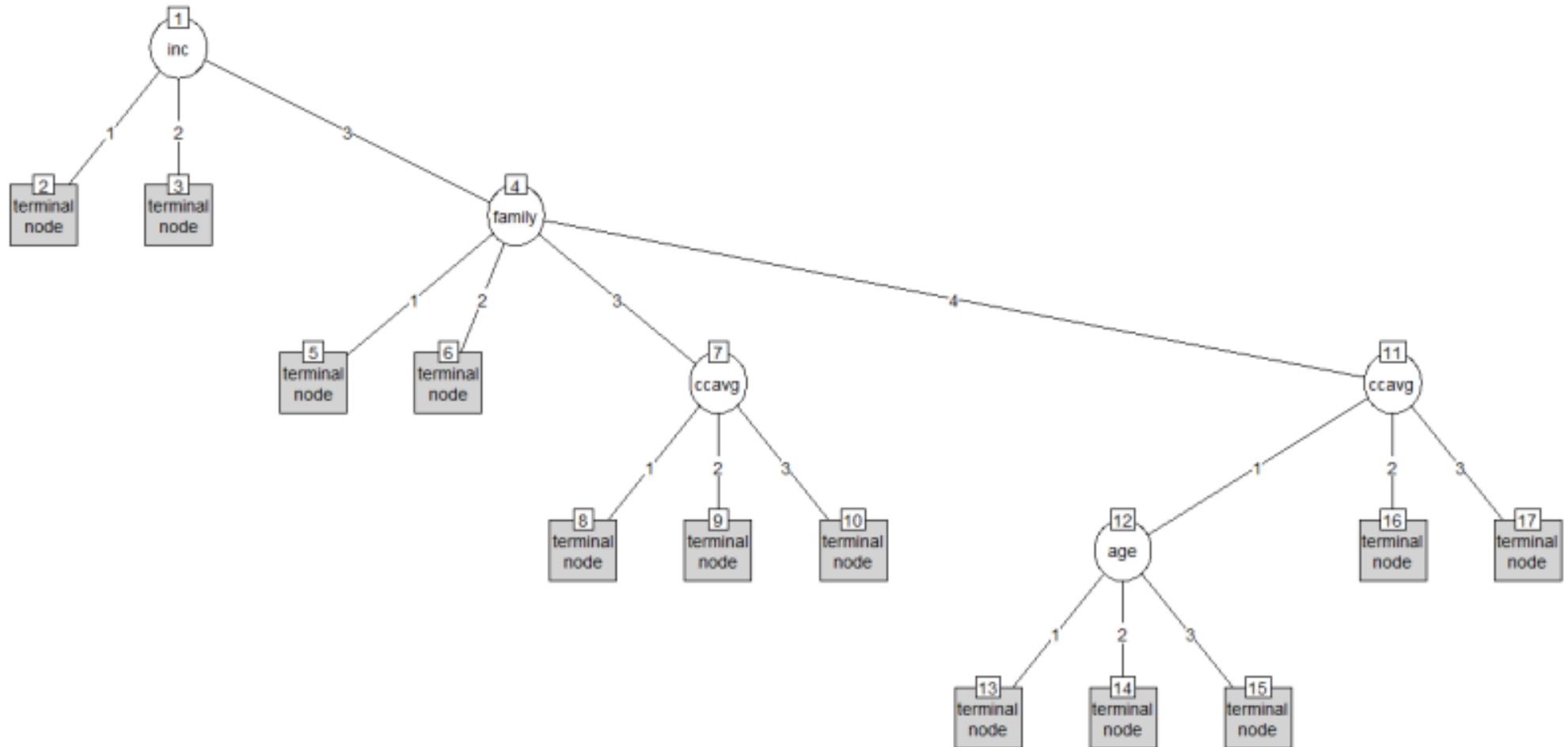
- CART
  - Binary split
  - Gini index
  - Cost complexity pruning
- C5.0
  - Multi split
  - Info gain / Gain ratio
  - Pessimistic pruning

# CART

0: Not take Personal Loan  
1: Take Personal Loan



# C4.5



# SPECIAL TREES

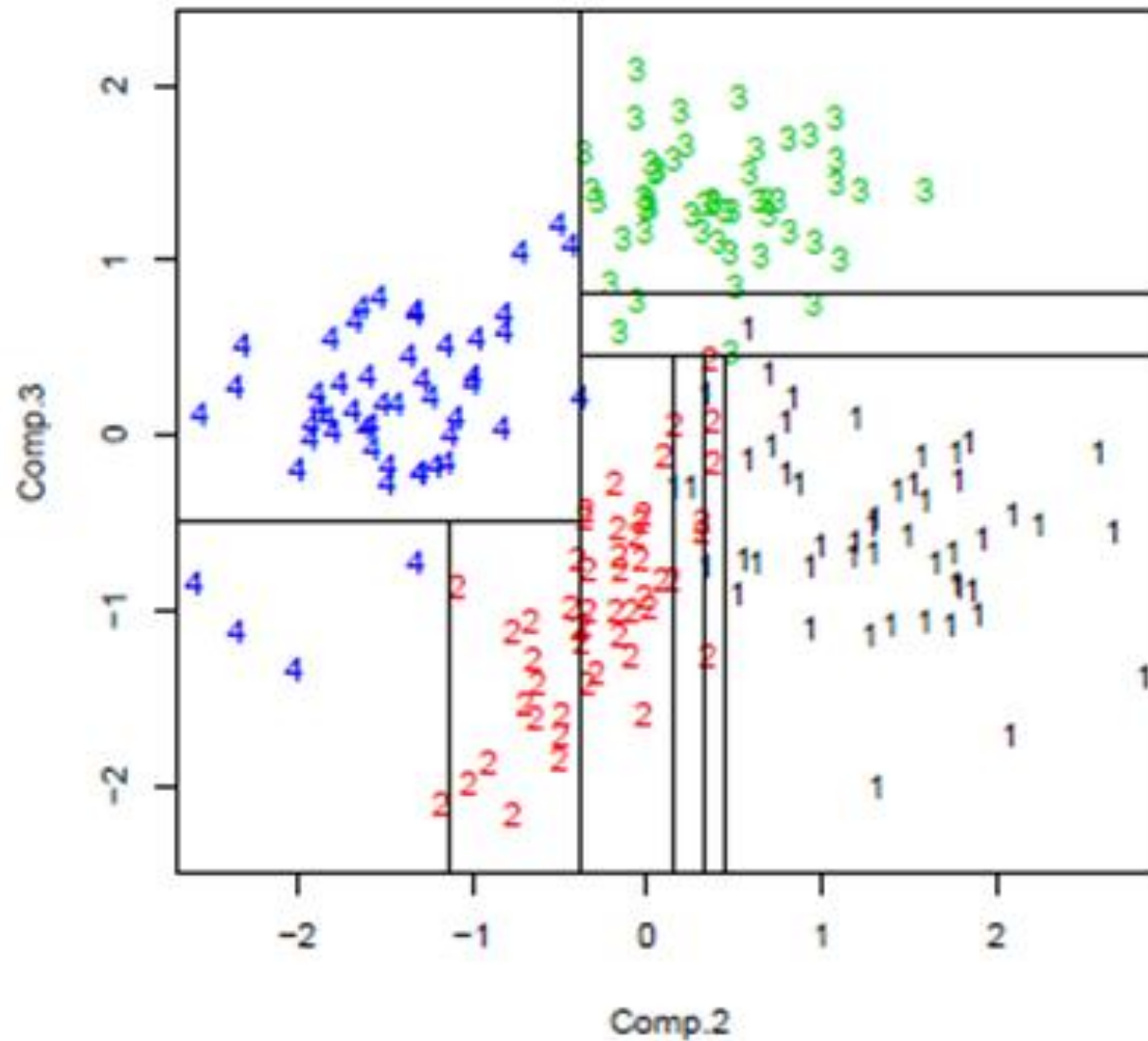
# Oblique Trees (OC1 Algorithm)

From  $x_i > K$  or  $< K$   
to

$$a_1x_1 + a_2x_2 + \dots + c > \text{or} < K$$

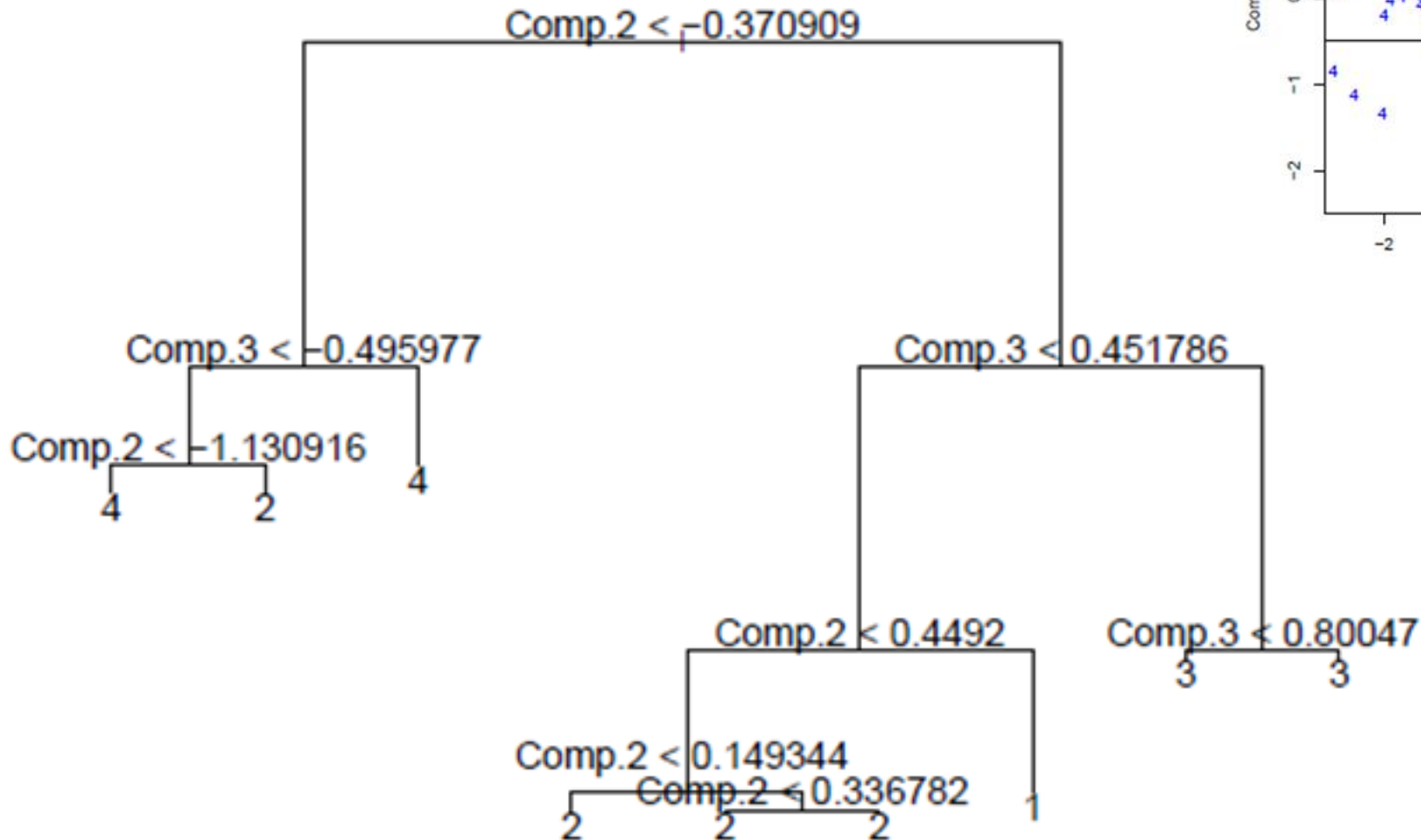
Possible only for numeric attributes

## Associated Decision Boundaries

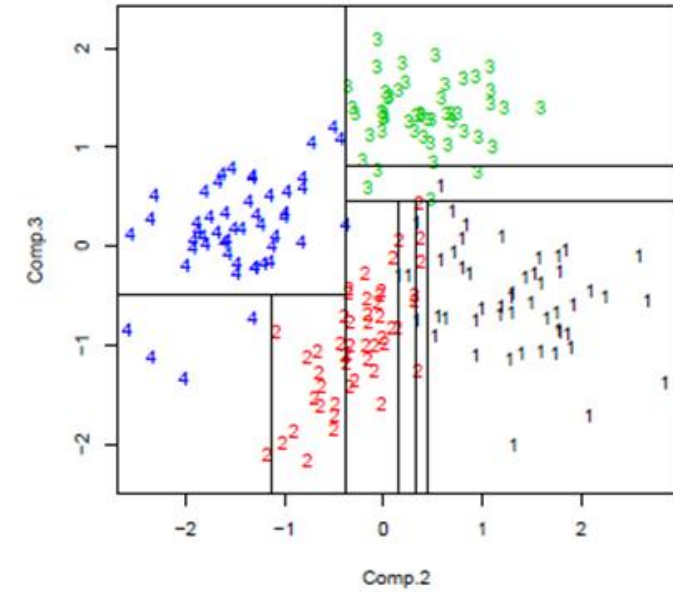


# CART

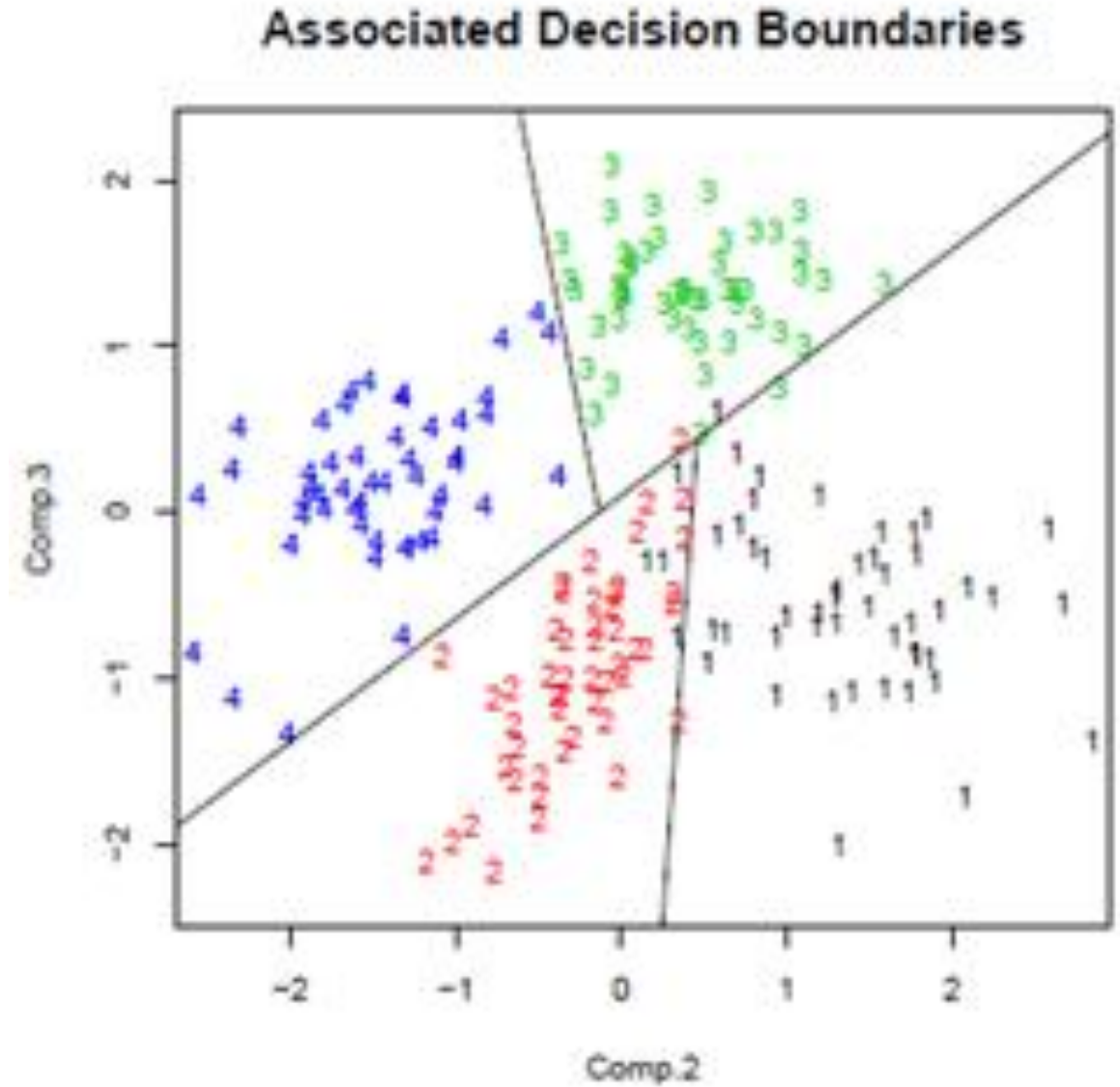
Axis-Parallel Tree



Associated Decision Boundaries

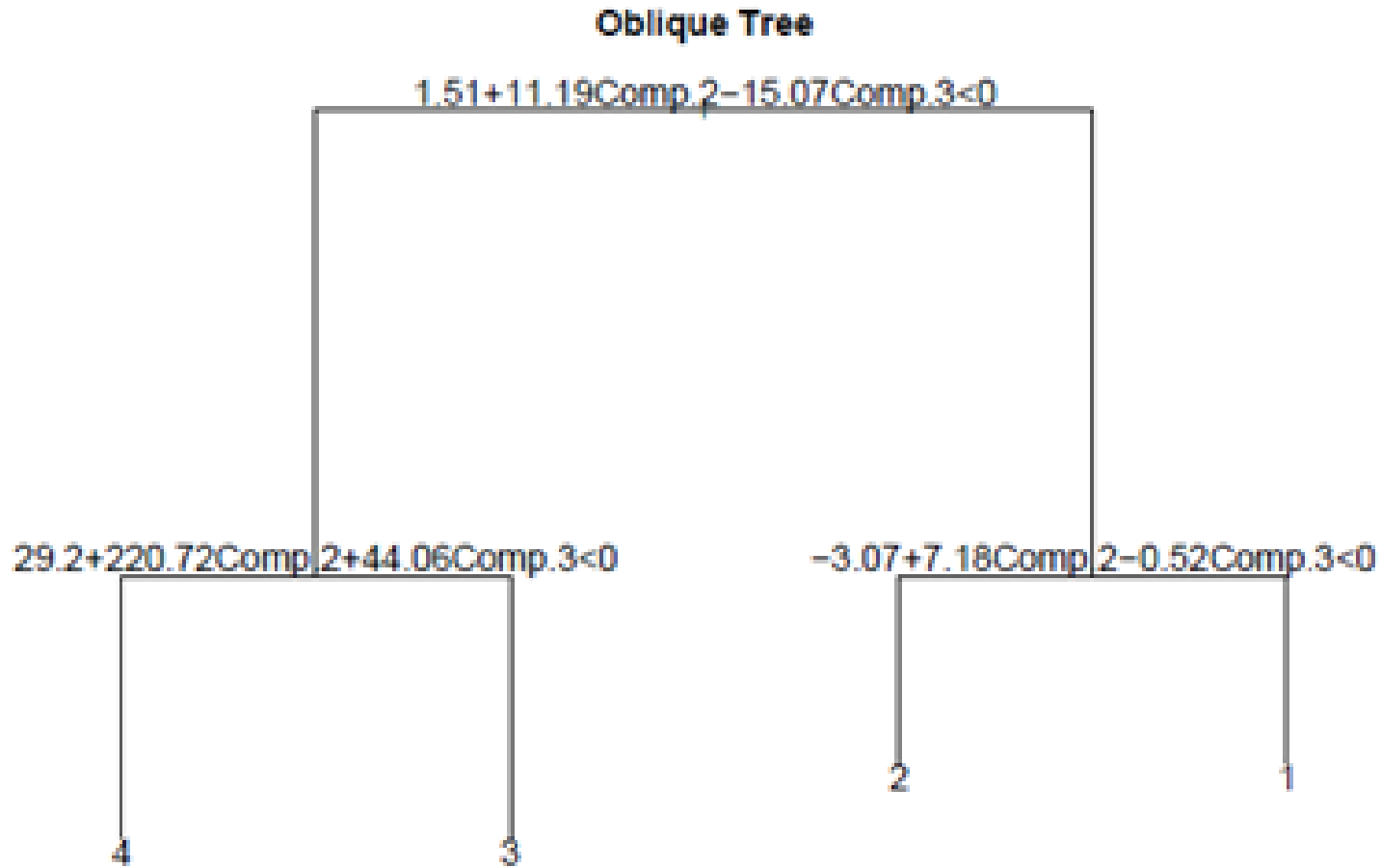


# Oblique Trees



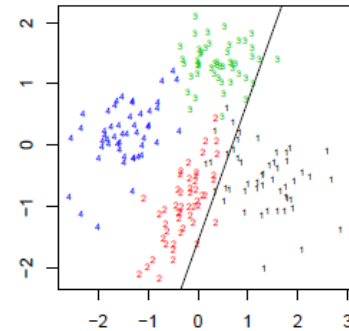


# Oblique Trees

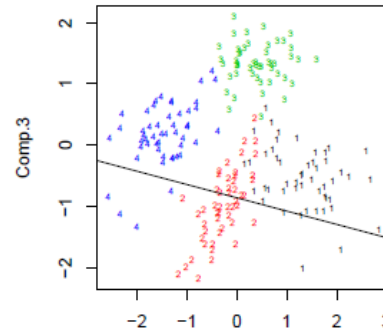


# The choices of oblique planes are infinite

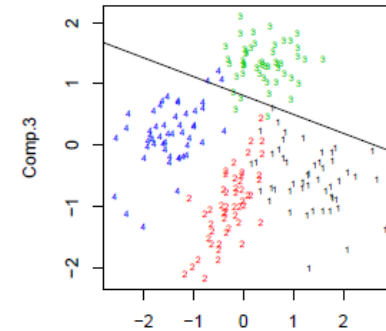
**{1}{2,3,4}**



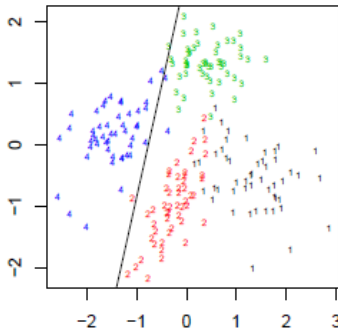
**{2}{1,3,4}**



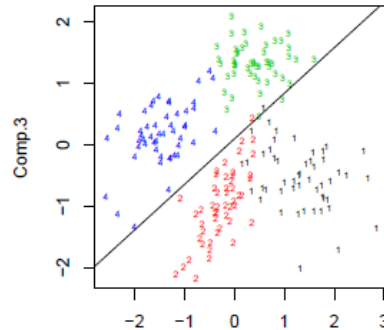
**{3}{1,2,4}**



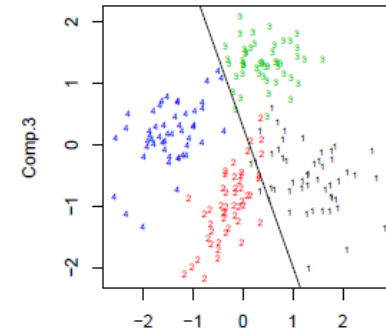
**{4}{1,2,3}**



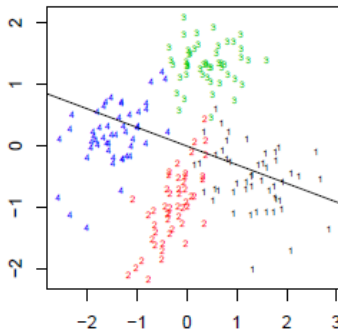
**{1,2}{3,4}**



**{1,3}{2,4}**



**{1,4}{2,3}**



# Oblique Trees

- If there are 4 classes, we have  $2^{(4-1)}-1$  ideal splits (7 splits)
- The impurity of each split is evaluated to identify the best ideal split
- Tree-growth proceeds as usual by applying the best ideal split (with the same stopping criteria as before)

Generalizing Trees

# INCREASING APPLICABILITY OF TREES

# Missing Values

- Missing values
  - if node  $n$  tests  $A$ , assign most common value of  $A$  among other examples routed to node  $n$
  - if node  $n$  tests  $A$ , assign most common value of  $A$  among other examples routed to node  $n$  that have the same class label as  $x$


ID	Age	Income	Family	COAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4		1
48	Middle	High	4	Low	1

7405G

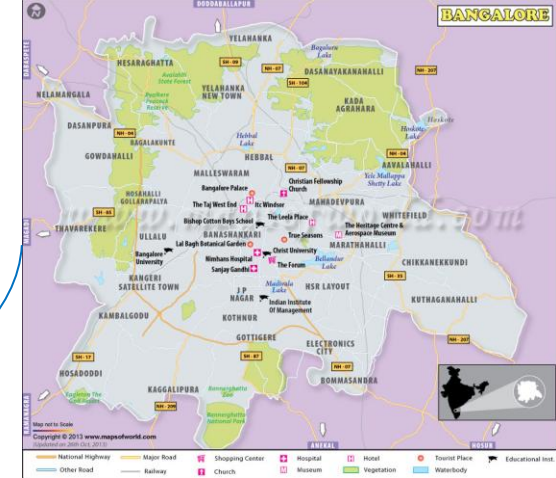


# Handling Numeric Attributes

Age	25	32	34	35	35	37	37	38
Loan	0	1	1	0	0	0	1	1



- Sort the data
- Create a bin wherever the class variable changes
- The value is the average of the nearest values



## HYDERABAD

Plot 63/A, Floors 1&2, Road # 13, Film Nagar,  
Jubilee Hills, Hyderabad - 500 033  
+91-9701685511 (Individuals)  
+91-9618483483 (Corporates)

## BENGALURU

Incubex, #728, Grace Platina, 4th Floor, CMH Road,  
Indira Nagar, 1st Stage, Bengaluru – 560038  
+91-9502334561 (Individuals)  
+91-9502799088 (Corporates)

## Social Media

Web: <http://www.insofe.edu.in>  
Facebook: <https://www.facebook.com/insofe>  
Twitter: <https://twitter.com/Insofeedu>  
YouTube: <http://www.youtube.com/InsofeVideos>  
SlideShare: <http://www.slideshare.net/INSOFE>  
LinkedIn: <http://www.linkedin.com/company/international-school-of-engineering>

*This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.*