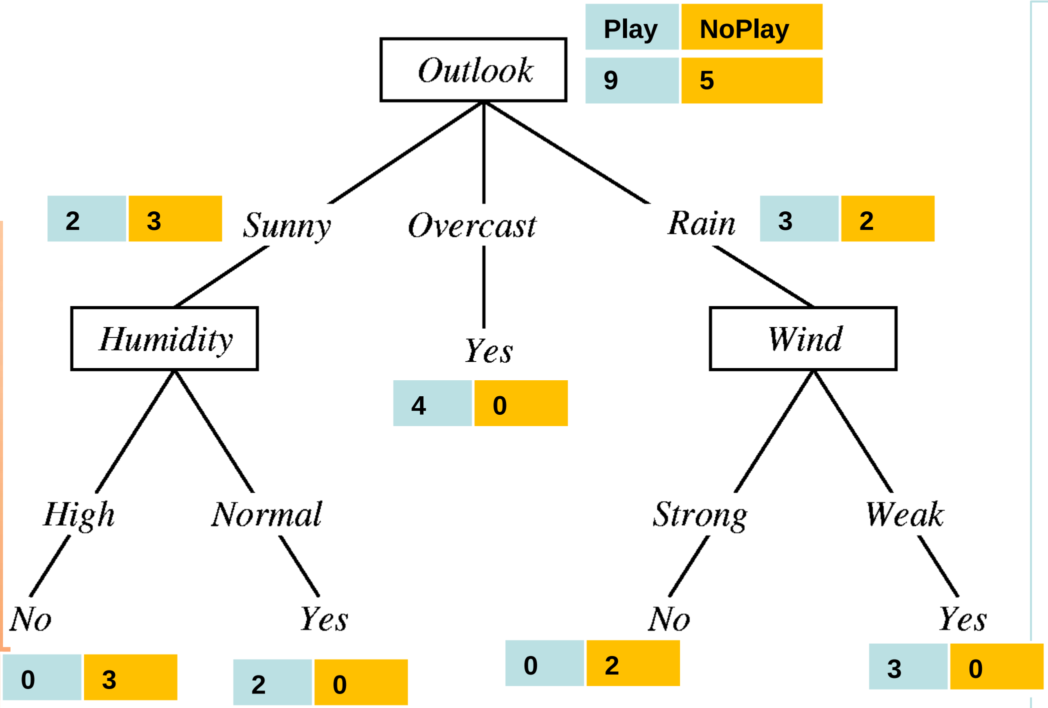
Inspire…Educate…Transform.

Decision Trees

Lt. Suryaprakash Kompalli
Senior Mentor, INSOFE

23rd Aug 2015



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision tree

- Extracts rules from data

- Example: Play happens when It is:

(Rainy **AND** NOT Humid) **OR** (Overcast) **OR** (Sunny **AND** NOT Windy)

Example Use Cases

- Detection of breast cancer relapse:
  - Local: Other parts of breast
  - Regional: Chest, underarms, collarbone
  - Distant: Liver, lungs
- Data (36 features):
  - Concentration of various proteins
  - Clinical demographics: Age, location of tumor, type
- What if you had 50 patients? What for 5K patients?

Example Use Cases

- Classifying noise vs star in hubble telescope images
  - 20 numerical attributes 2.2K images
- Estimating software development costs
- Predicting user actions, component failure, maintenance schedules
- Grouping related articles / books together
  - Legal documents, receipts, tax articles

See more here: http://booksite.elsevier.com/9780124438804/leondes_expert_vol1_ch3.pdf

CONSTRUCTING A DECISION TREE

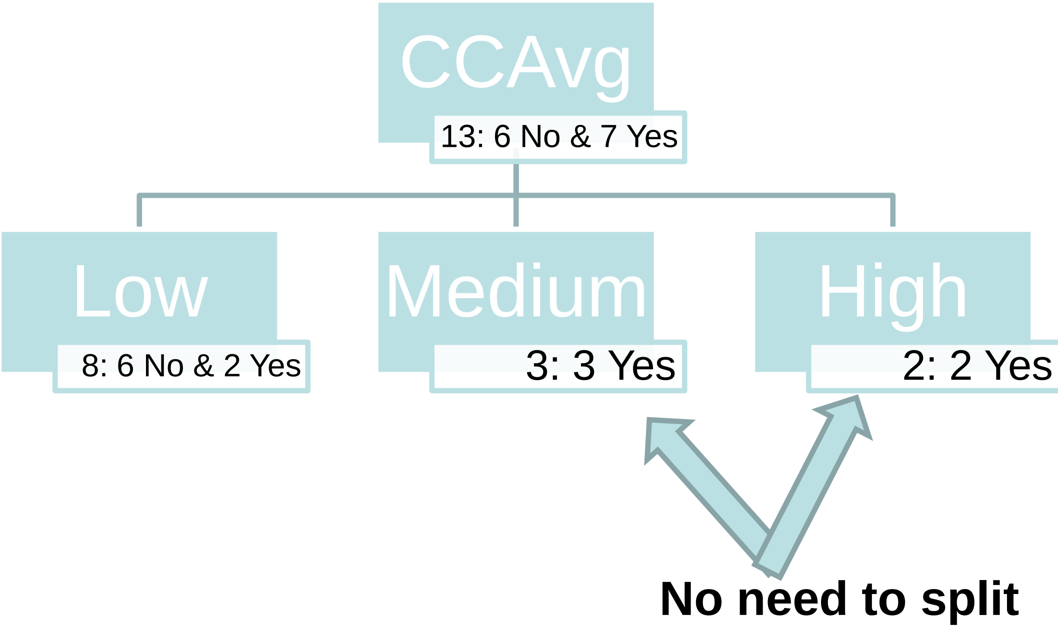
Induction of Decision Trees

- Data Set (Learning Set)
  - Each example = Attributes + Class
- TDIDT
  - Top Down Induction of Decision Trees or ID3
- Easy to grasp:
  - If data S has only one class, create leaf node
  - Else:
    - Split data S into two sets S1 and S2 using **"most informative attribute"** A
    - Create sub trees using S1 and S2

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

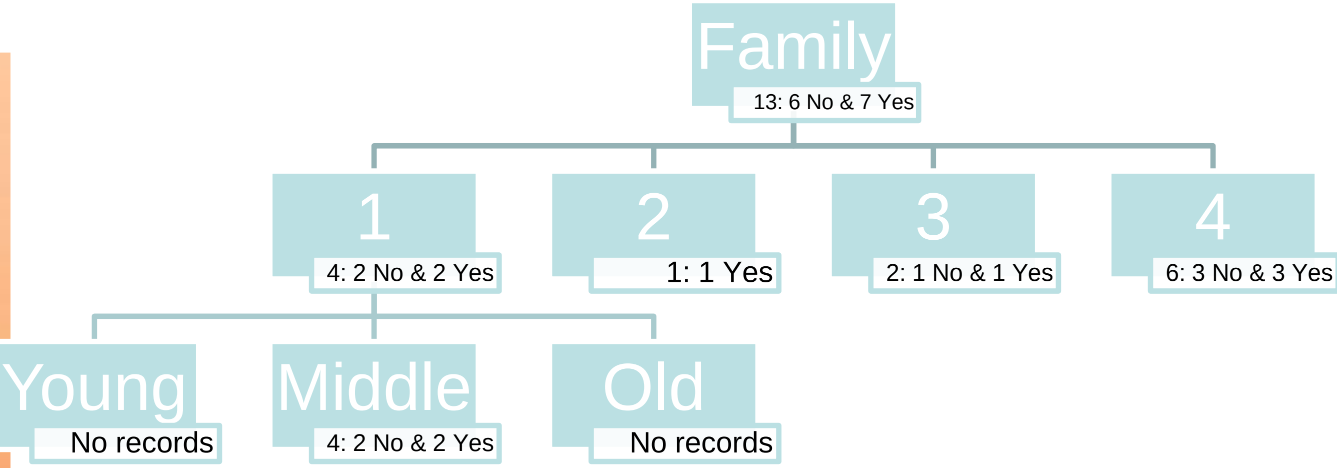
Constructing a Decision Tree

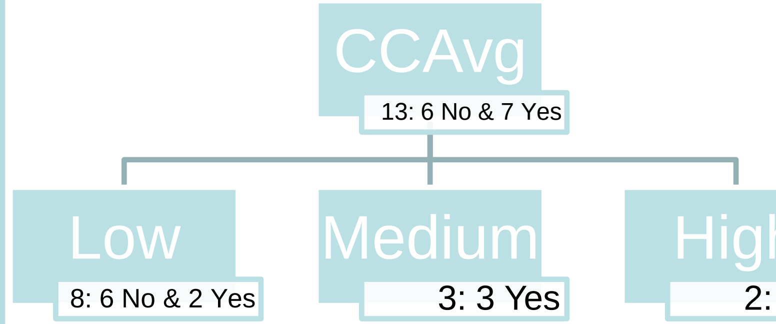


Nodes (root node): Test points
Leaves: Decision points
Branch: Collection of nodes and the leaf

Family	CCAvg	Personal Loan
4	Low	0
3	Low	0
1	Low	0
1	Low	0
4	Low	0
4	Low	0
1	High	1
4	Medium	1
2	High	1
1	Medium	1
3	Medium	1
4	Low	1
4	Low	1

Constructing a Decision Tree





Which DT is better? Using Family as attribute (DT on left) or CCAvg as attribute (DT on right)

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

Two aspects

- Which attribute to choose?
- Where to stop?

Entropy

- H $= -\sum_i p_i \log_2 p_i$

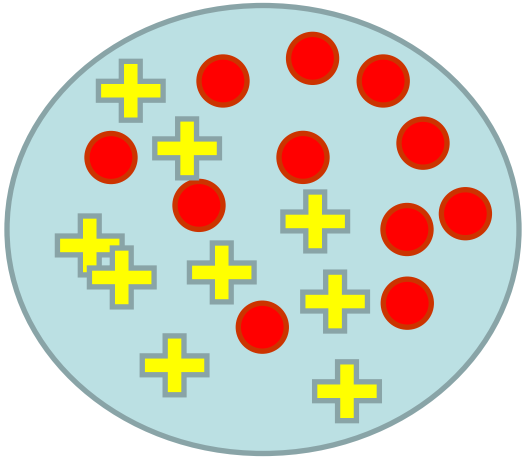
Probability of toss		
Heads	Tails	Entropy
0	1	NAN
0.0001	0.9999	0.0014
0.05	0.95	0.29
0.25	0.75	0.81
0.5	0.5	1
0.75	0.25	0.81
0.95	0.05	0.29
0.9999	0.0001	0.0014
1	0	NAN

- Compute Entropy for coin:
  - Let us say I have five biased coins, what is the uncertainty of tossing each coin?
  - Prob of heads: 0,0.05,0.25,0.5,0.75,0.95,1

Entropy: A measure of randomness

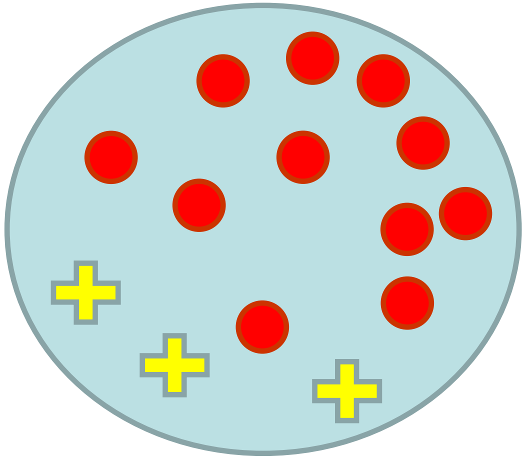
$$H = -\sum_i p_i \log_2 p_i$$

More confusion



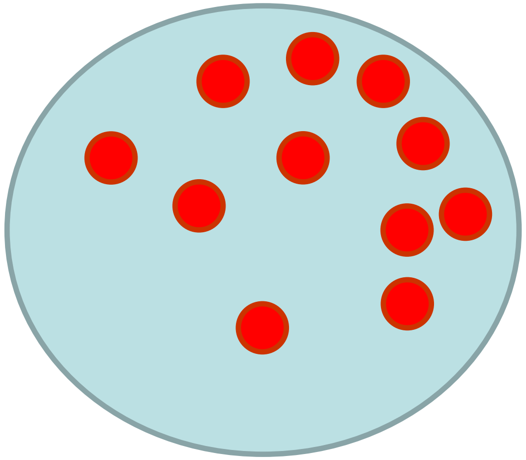
Entropy: $-1 * \left(\frac{11}{20} log \frac{11}{20} + \frac{9}{20} log \frac{9}{20}\right)$
$= 0.47 + .52 = 0.99$

Less confusion



Entropy: $-1 * \left(\frac{11}{14} log \frac{11}{14} + \frac{3}{14} log \frac{3}{14}\right)$
$= 0.27 + .48 = 0.75$

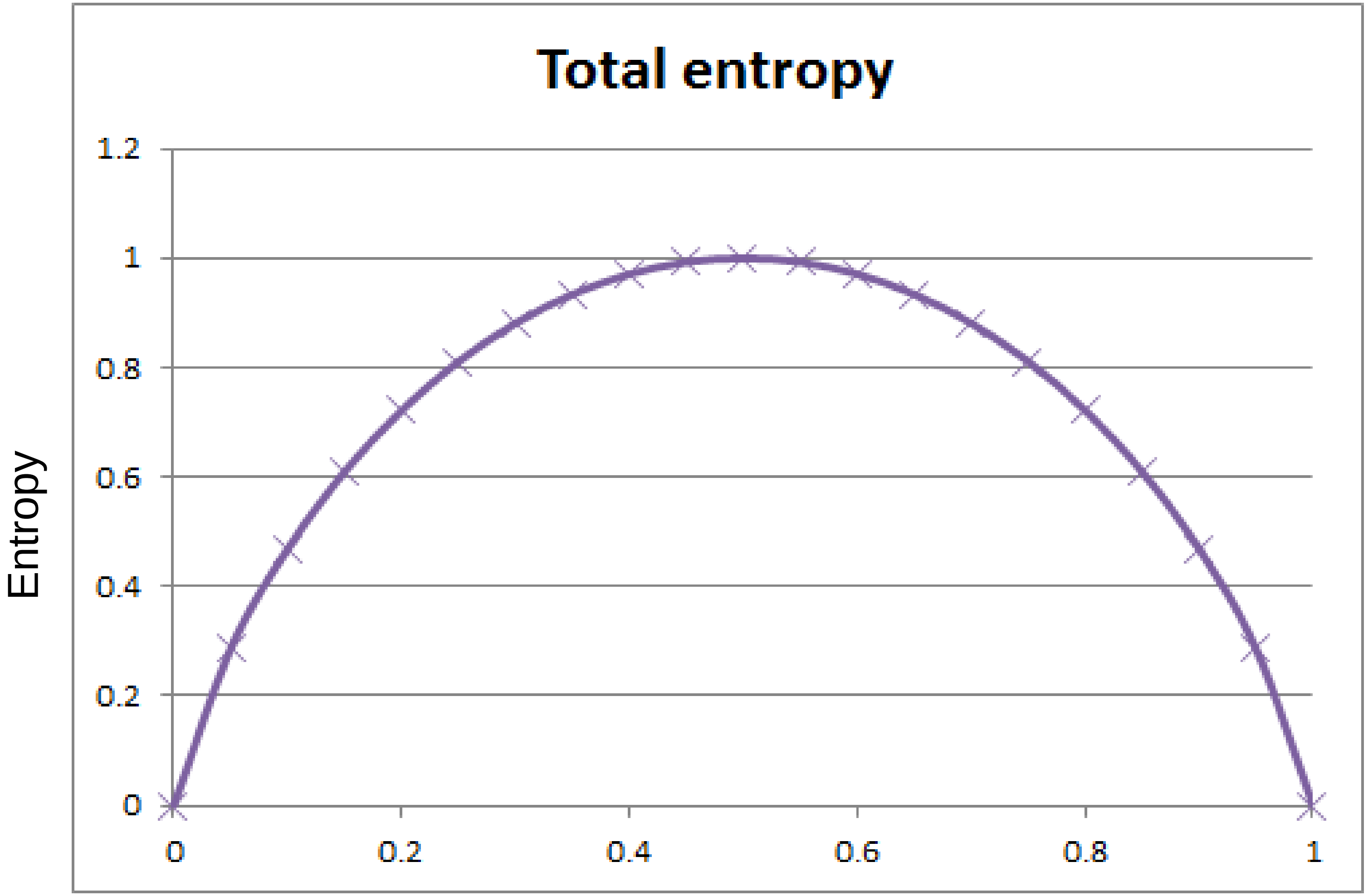
No confusion



Entropy: $-1 * \left(\frac{11}{11} log \frac{11}{11}\right)$
$= 0$

Entropy: A measure of randomness

EntropycalculationsManual.R



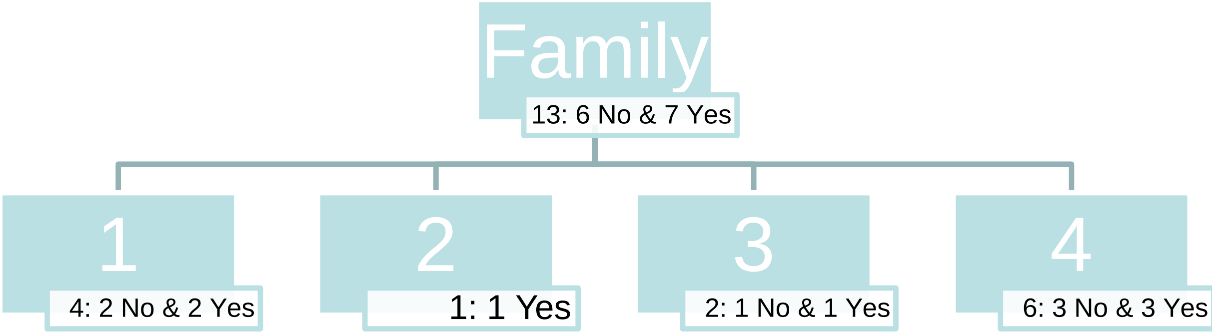
Prob of each class

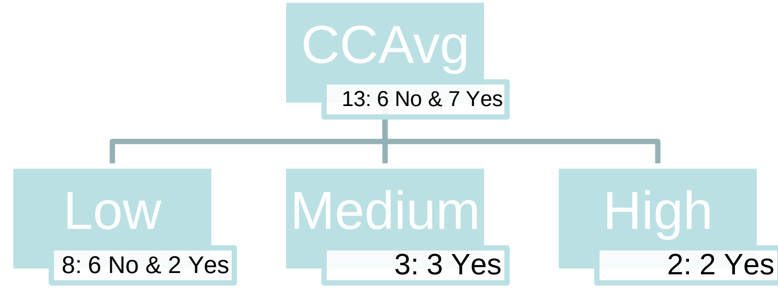
$$H = -\sum_i p_i \log_2 p_i$$

$$H = -\sum_{i=1\ to\ 3} 1/3 \log_2 1/3$$

This is for a 2-class problem. What could the max value be for 3-class? For 10 class?

Constructing a Decision Tree





Which DT is better? Using Family as attribute (DT on left) or CCAvg as attribute (DT on right)

Let us use Entropy to select attributes

Formula for entropy: $\mathrm{H} = -\sum_i p_i \log_2 p_i$

Entropy at a split from a particular attribute

$$H_{split} = \sum_{n=1}^{N} w_n * H_n$$

N is the number of nodes at the split
w_n Weight of each node at that split

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

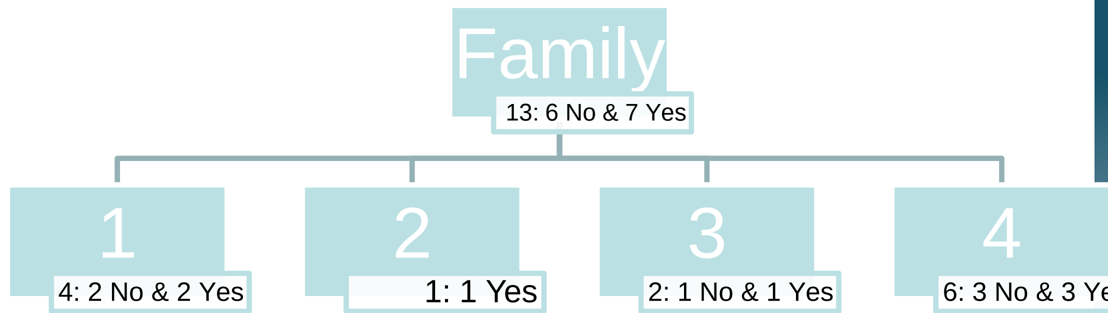
How to use Entropy to Select Attributes

EntropycalculationsManual.R

- Initial entropy: $-\frac{6}{13}log_2\left(\frac{6}{13}\right)-\frac{7}{13}log_2\left(\frac{7}{13}\right)=0.995$
- Entropy on split with family:

$$\frac{4}{13}\left(-\frac{2}{4}log_2\left(\frac{2}{4}\right)-\frac{2}{4}log_2\left(\frac{2}{4}\right)\right)+\frac{1}{13}\left(-\frac{1}{1}log_2\left(\frac{1}{1}\right)\right)+\frac{2}{13}(\ldots)+\frac{4}{13}(\ldots)$$

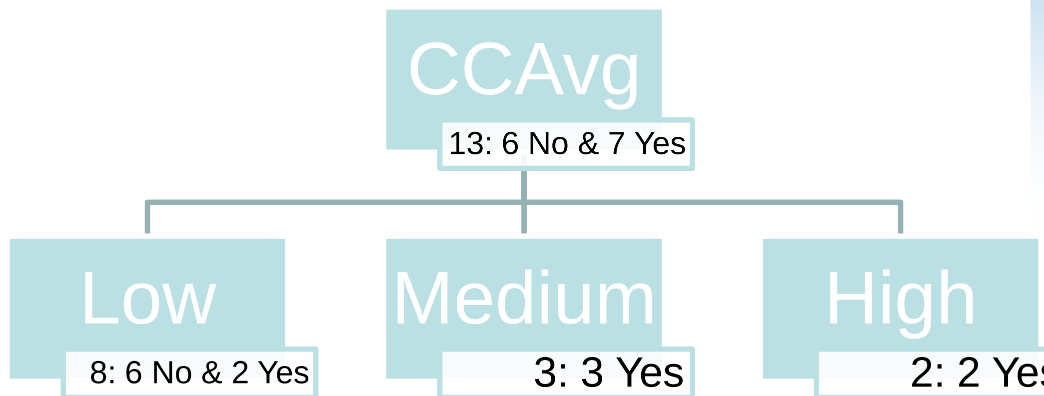
$$\frac{4}{13}(1)+\frac{1}{13}(0)+\frac{2}{13}(1)+\frac{4}{13}(1)=0.923$$



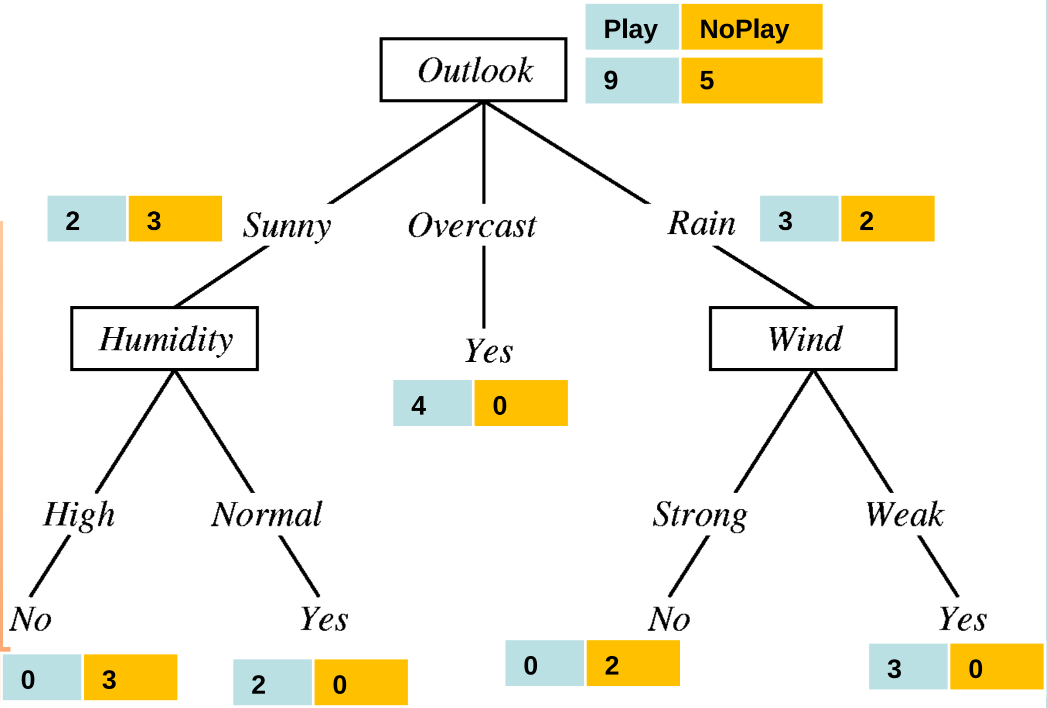
- Entropy on split with CCAvg:

$$\frac{8}{13}(0.81)+\frac{3}{13}(0)+\frac{2}{13}(0)=0.499$$

Information gain = Entropy of the system before split – Entropy of the system after split



Select attribute with largest information gain



Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Ok – so we build a DT by selecting best attribute at each split. How do we use the DT?

ID3 is very fast.

DT is easy to explain. One can justify classification result for a new sample. You can get insights like: “Customers with different family sizes are not very different”

Can you do regression?

Are we done?

Is information gain the best measure to select “A”?

- Data Set (Learning Set)
  - Each example = Attributes + Class
- TDIDT
  - Top Down Induction of Decision Trees or ID3
- Easy to grasp:
  - If data S has only one class, create leaf node
  - Else:
    - Split data S into two sets S1, S2, S3… using **“most informative attribute”** A
    - Create sub trees using S1,S2,S3

Stop only when leaf has one class!!! Will this overfit?

What if “A” is numeric?

UNDERSTANDING DECISION TREES

Sometimes information gain fails

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

Let us do information gain for split on ID

Other examples:
Date of oil change

Bill amount

Entropy after split

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

Entropy = -1*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 (0.995) 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 entropy gain.

- So, we need to normalize it to get something like entropy gain per state.

Information content

- Split Information is defined as = $-f_i \log_2 f_i$. We only want to know fraction of the members in a state divided by the total members.

- **Split Information of ID**: It has 13 states. So, the information content = -1/13*LOG(1/13,2)

Normalized Information gain: $\frac{\text{Overall Information gain at the node}}{\sum_{i=0}^{N} \text{Split Information}_i}$

- Go to R

EntropycalculationsManual.R

Other Measures

A: *Attribute on which split happens*
v: Different values of attribute
c, i, j: Class labels

- Information Gain Using Residual Information

$$Gain(A) = I - I_{res} \quad I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

This term is biased toward attributes with many values

- Information Gain Ratio

$$I(A) = -\sum_{v} p(v) \log_2(p(v))$$

$$GainRatio(A) = \frac{Gain(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

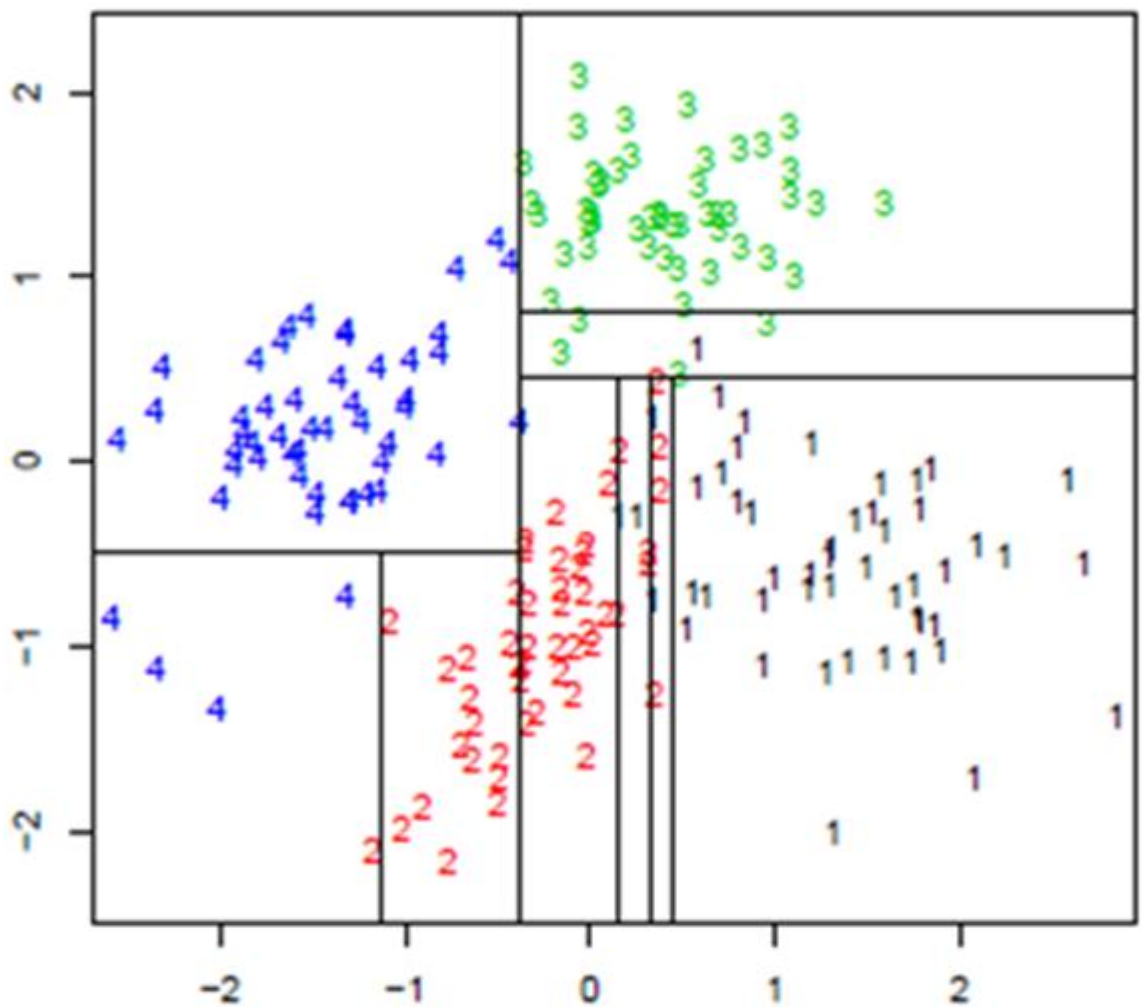
- Gini Index

$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v) p(j|v)$$

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

HOW TO MINIMIZE THE OVERFIT?

Geometry of Decision Trees: Axis Parallel Search



What happens in case of class imbalance?
Can a decision tree overfit?

Termination criteria

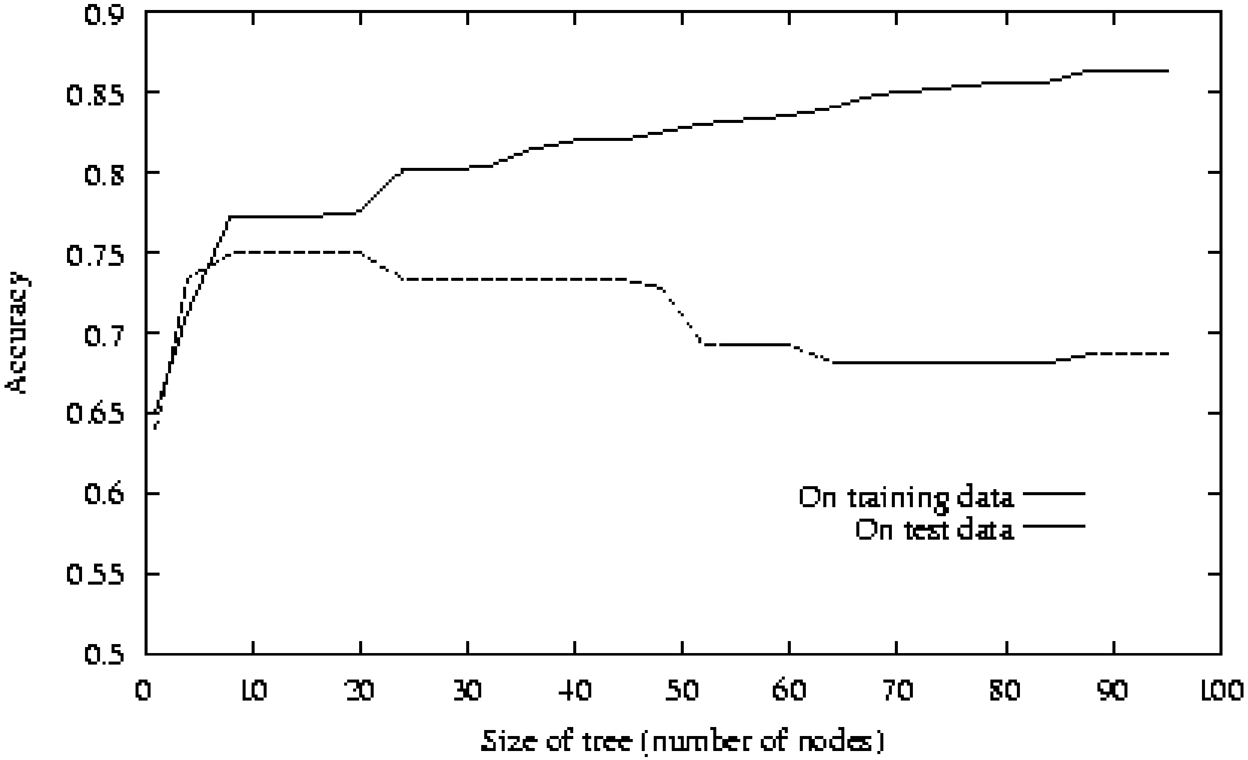
- All the records at the node belong to one class
- Changed to:
  - A significant majority fraction of records belong to a single class
  - The segment contains only one or very small number of records
  - The improvement is not substantial enough to warrant making the split.

Approaches to prune tree

- Three approaches
  - Stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data (Use test or Eval data to stop)
  - Allow the tree to overfit the data, and then post-prune the tree layer by layer
  - Allow the tree to overfit the data, transform the tree to rules and then post-prune the rules.

Minimize variance

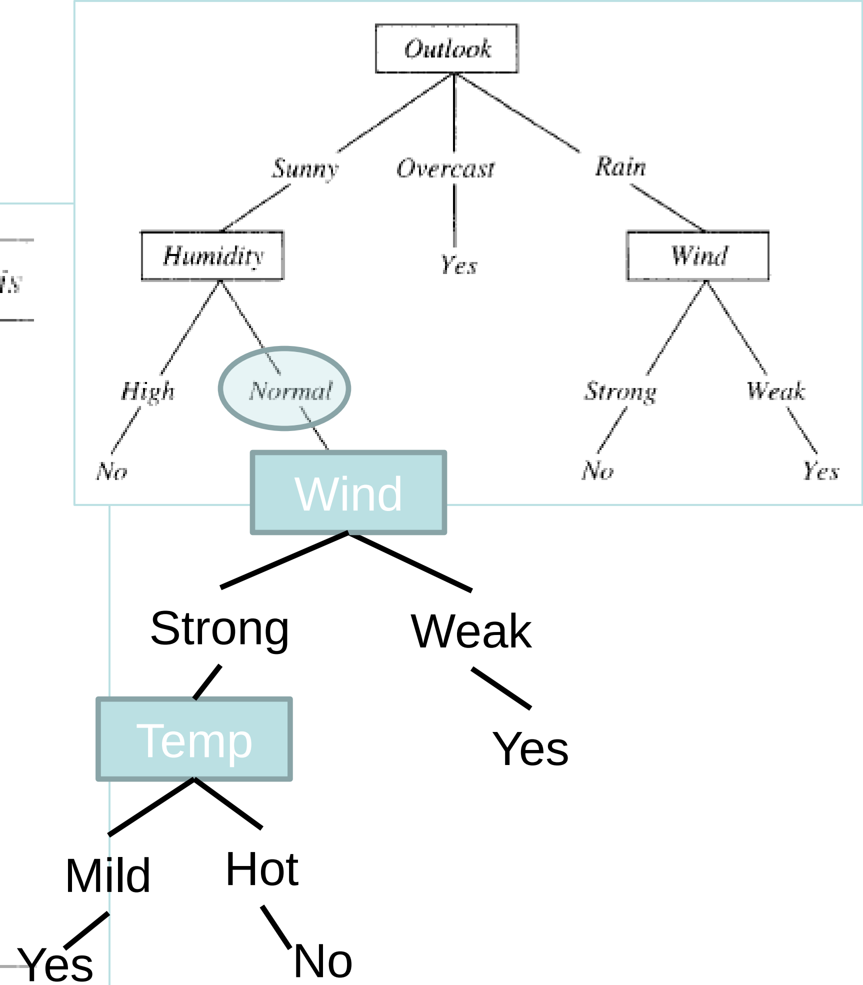
- Build the tree on train data
- Test it on test data
- Plot test and train errors at various pruning levels



Reference text, Tom Mitchell has an example on pages 67-68

Overfitting in DTs

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

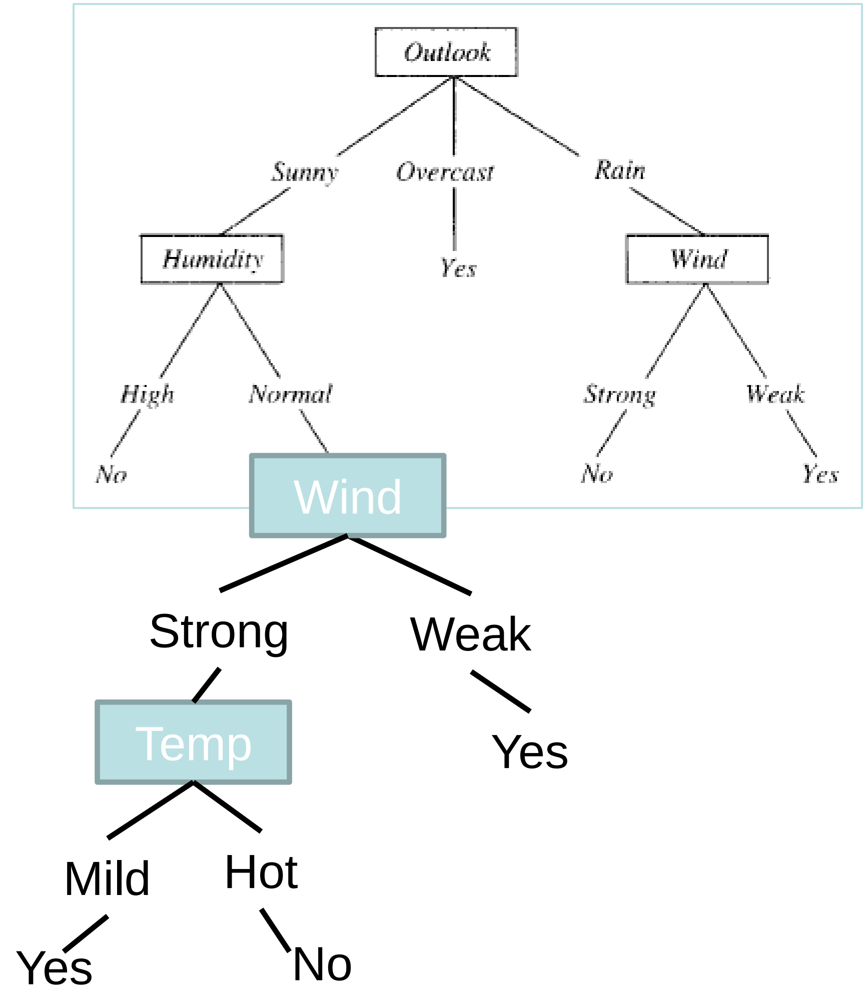


$\langle Outlook = Sunny, Temperature = Hot, Humidity = Normal,$

$Wind = Strong, PlayTennis = No\rangle$

Reduced Error Pruning

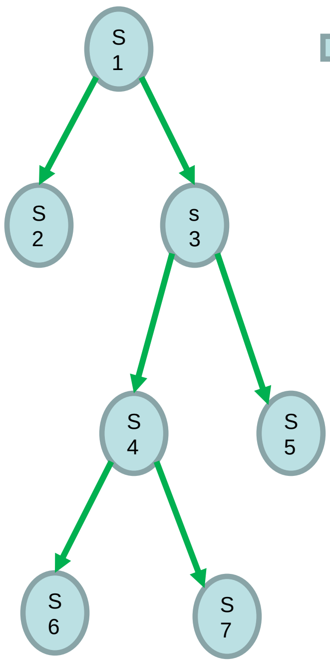
- Start from leaf nodes
- If removal of node does not change validation accuracy
  - Combine leaf elements of this node into previous node
- Continue till root node



Cost complexity pruning - CART

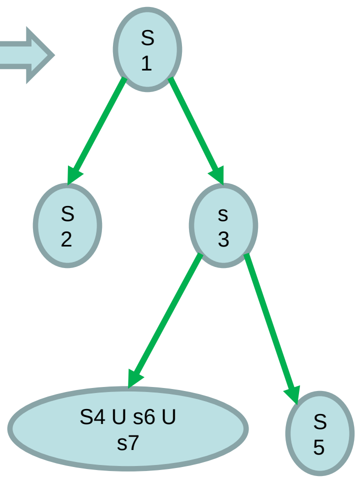
R: 07regression.R

- $J(T_t) = ErrorRate(T_t) + \alpha * |T_t|$
  - T_t: Different trees, pruned from original tree
  - α Cost-complexity parameter
  - Error rate can be sum of squared error



Since few nodes are removed:
$ErrorRate(T_{left}) < ErrorRate(T_{right})$
However,
$J(T_{left}) > J(T_{right})$
depending on α and $|T_t|$

We need to find a value of α that balances error rate with complexity (height) of a tree



http://web.cs.du.edu/3704DM/materials/Lecture10.pdf

Cost complexity pruning - CART

R: 07regression.R

- $J(T_t, S) = ErrorRate(T_t, S) + \alpha * |T_t|$
  - T_t:Different trees, pruned from original tree
  - S:Data set "S"
  - Uses GINI
- Increase α slowly, starting from 0
- Do a K-fold validation on all of them and find the best pruning α
- It is also possible set a threshold cost complexity

C4.5: Pessimistic Pruning

- A node is built using "N" samples.
  - Estimate true error if we had infinite samples
  - If true error after split is more, remove split

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

e – Estimated true error
N – Number of samples covered by leaf
f – error on training data
z- Inverse of Standard Normal Cumulative function

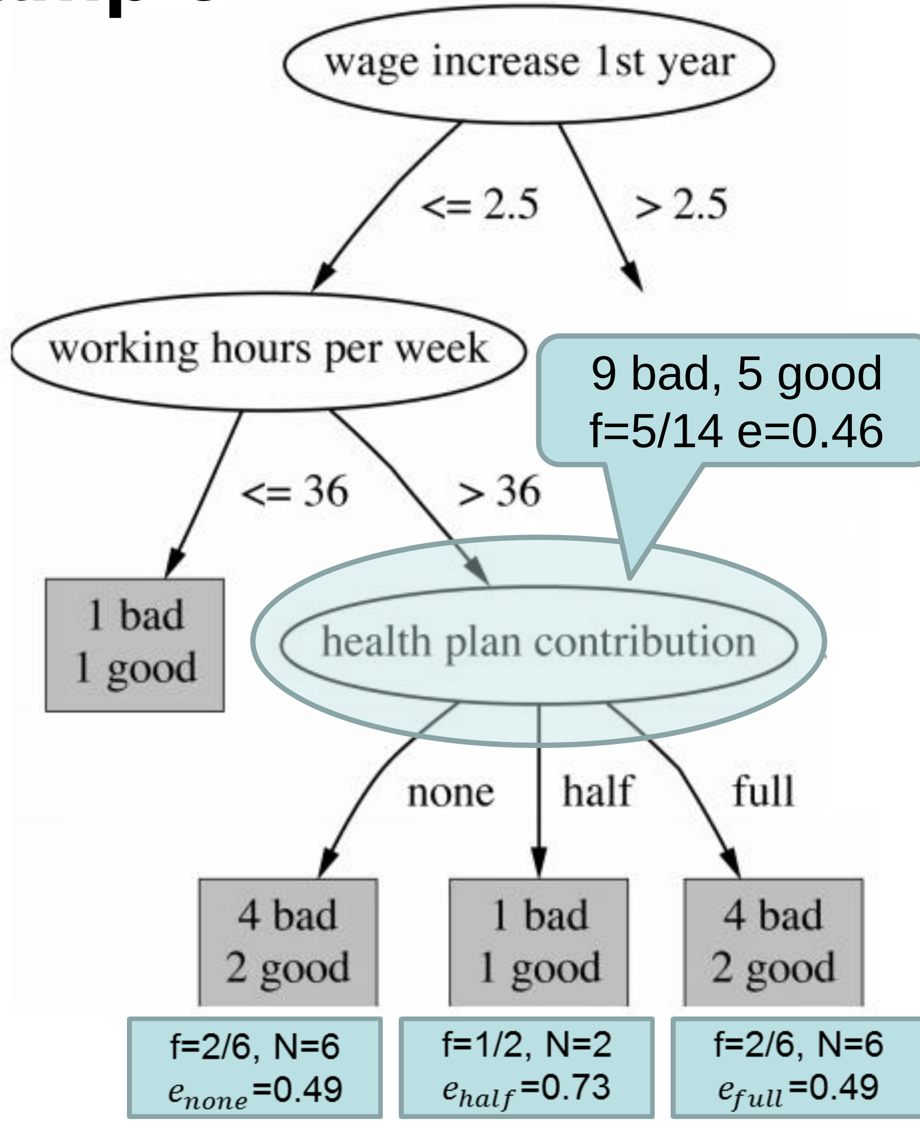
z	Confidence
0.67	50%
1	68%
1.64	90%
1.96	95%

- Handling continuous variables (Discussed later)

http://web.cs.du.edu/3704DM/materials/Lecture10.pdf
*www.kdnuggets.com/data_mining_course/dm7-decision-tree-**c45**.ppt*

C4.5 Pessimistic Pruning Example

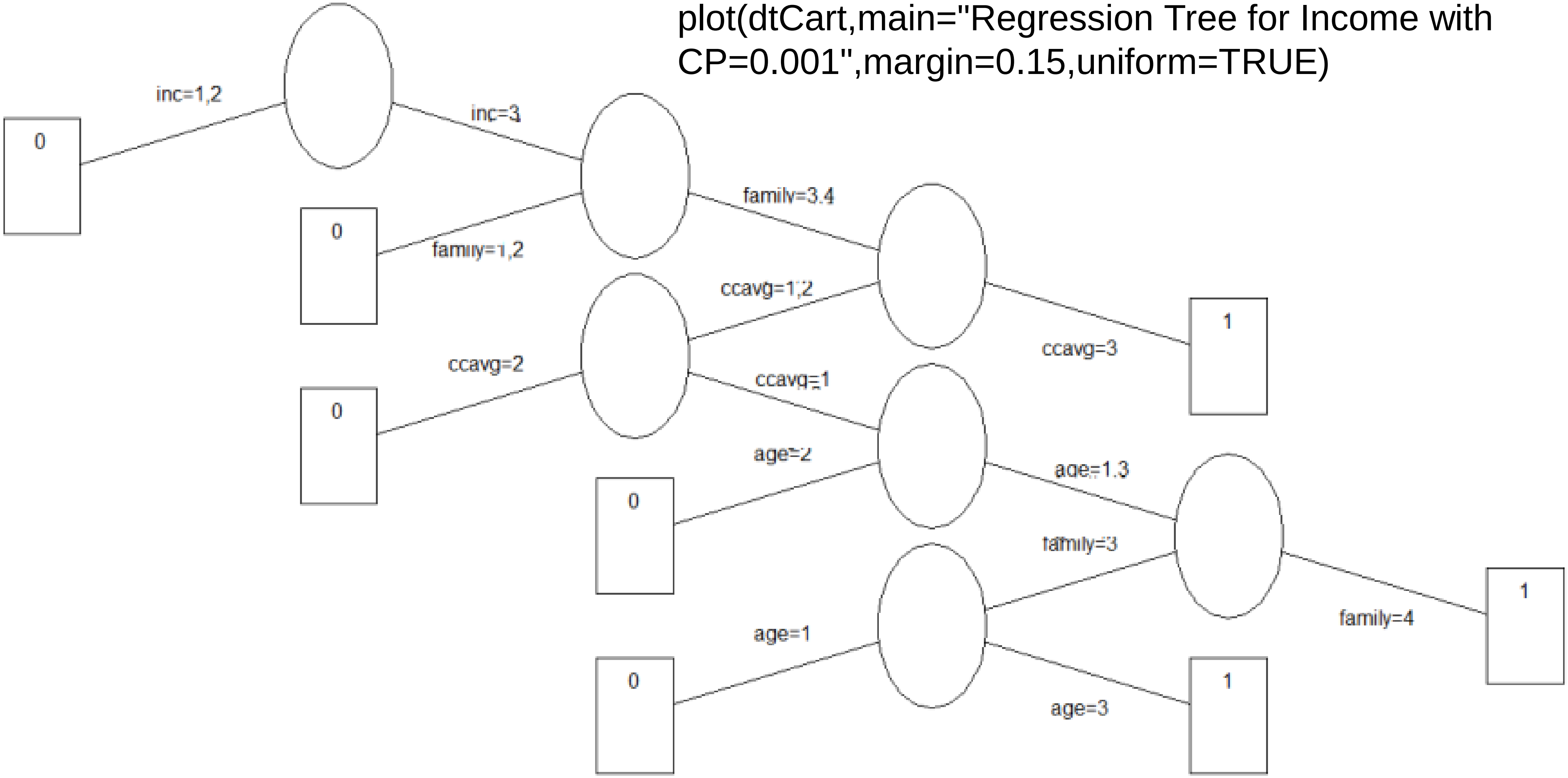
- At highlighted node:
  - Majority class is "Bad"
    - Error is f=5/14, e_{merge}=0.46
  - Cumulative nodes:
  - $e_{split} = \frac{6}{14} e_{none} + \frac{2}{14} e_{half} + \frac{6}{14} e_{full} = 0.524$
- $e_{merge} < e_{split}$, so prune highlighted node



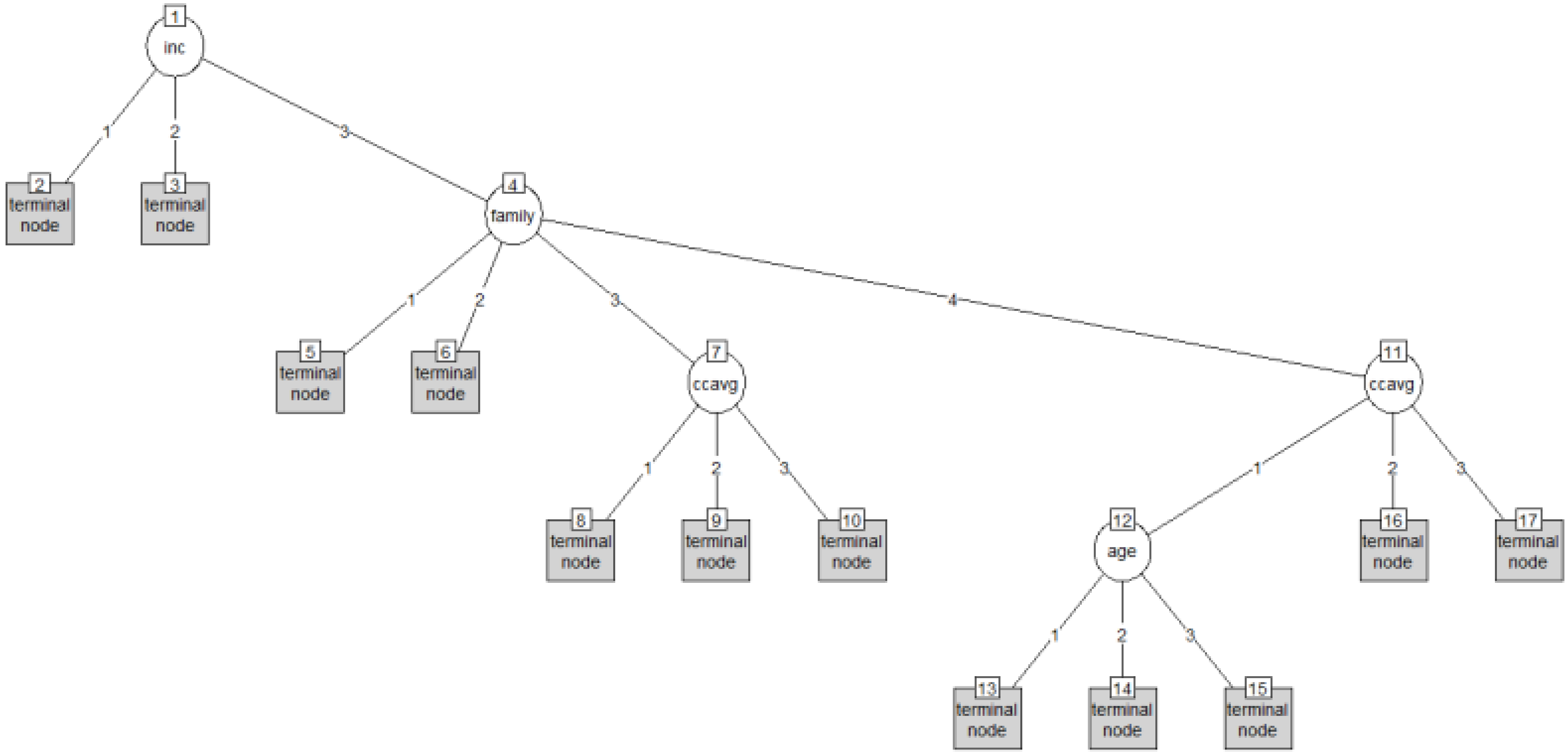
http://www.ke.tu-darmstadt.de/lehre/archiv/ws0809/mldm/dt.pdf

CART

```
dtCart=rpart(inc ~.,data=train,
             method="anova",
             cp=0.001)
plot(dtCart,main="Regression Tree for Income with CP=0.001",margin=0.15,uniform=TRUE)
```



C4.5


1
inc
1
2
3
2
terminal node
3
terminal node
4
family
1
2
3
4
5
terminal node
6
terminal node
7
ccavg
1
2
3
8
terminal node
9
terminal node
10
terminal node
11
ccavg
1
2
3
12
age
1
2
3
13
terminal node
14
terminal node
15
terminal node
16
terminal node
17
terminal node

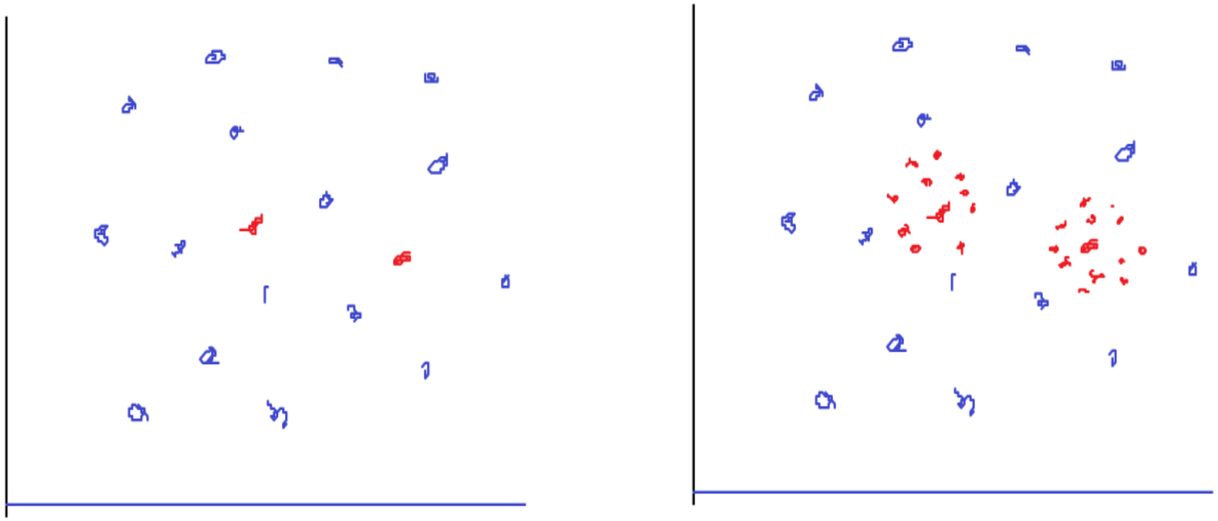
INCREASING APPLICABILITY OF TREES

Missing values

- Missing values
  - Given input "x" has missing value for attribute "A"
  - 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*

Smoting

- When there are few samples for a class
  - Use K nearest elements of other class to form a bound
  - Create additional samples within bound



06FinalTrees.R

Cost

Is all information easy to get?

Age of car	Maximum speed	Tyre install date	Air pressure	Tyre failure
3	76	Nov-1976	33	Y
7	56	Feb-1982	28	N

Quinman uses cost-normalized gain instead of information gain.

$$CostNormlizedGain(D, A) = \frac{Gain^2(D, A)}{Cost(D, A)}$$

An alternative measure: $\frac{2^{Gain(D,A)} - 1}{(Cost(D, A) + 1)^w}$

HANDLING NUMERIC ATTRIBUTES

Bucketing numeric attributes we can convert them to categorical

How to bucket? Equal width buckets, Equal population buckets

For numeric attributes, perform binary split; use Information gain to identify one threshold for the numeric attribute among many possible thresholds.

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

Sort attribute in increasing order. See where the class attribute changes value. Use mid-point as the threshold. In above dataset, three thresholds will be tried: 28.5, 34.5, 37

Can the same numeric attribute be split at different locations at different heights?

Yes !!! In above example, first split may be at 28.5 or 37 and next split may be at 34.5

*www.kdnuggets.com/data_mining_course/dm7-decision-tree-**c45**.ppt*

SPECIAL TREES

Oblique tree

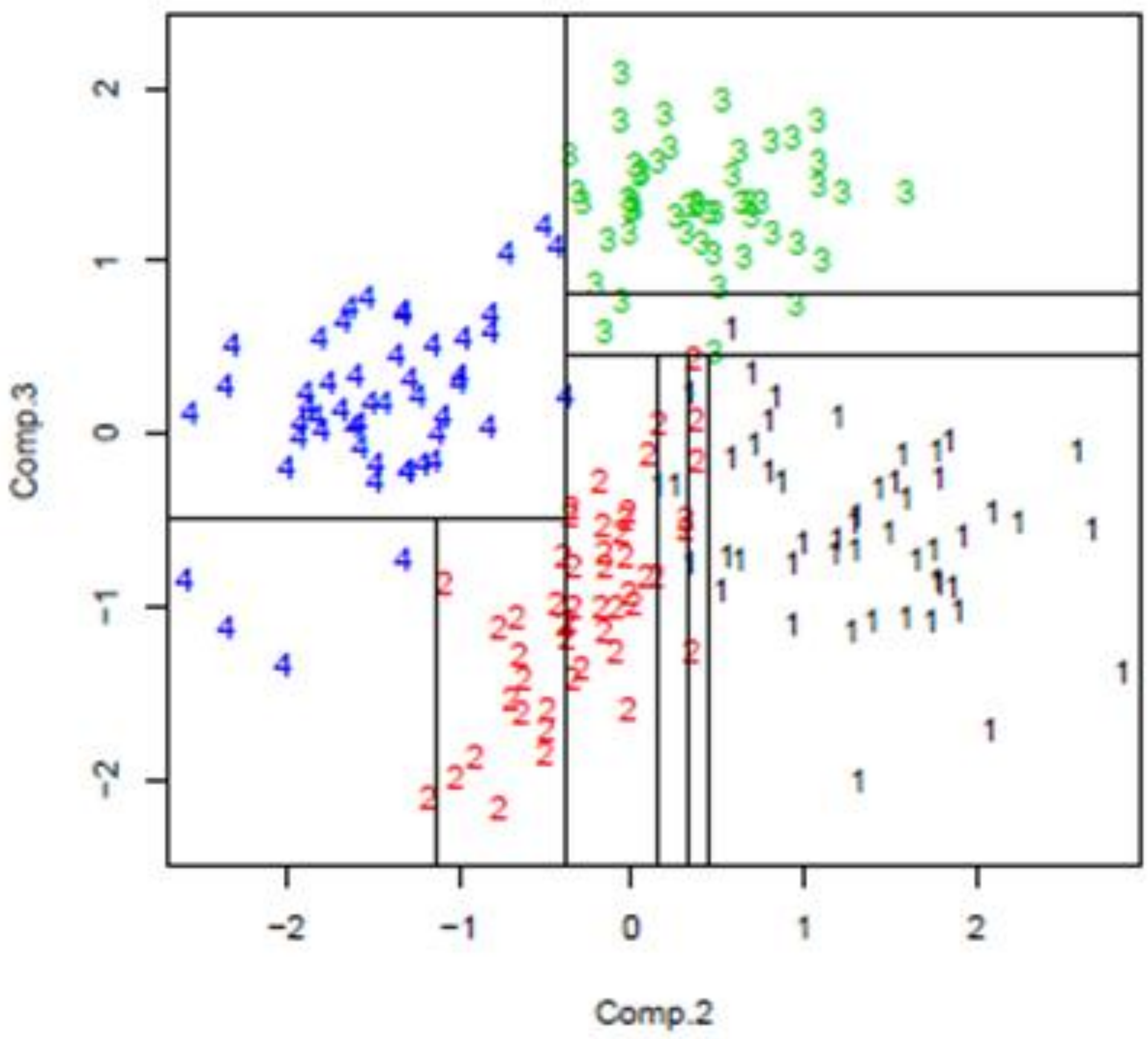
$$x_i > K \ or < K$$

To

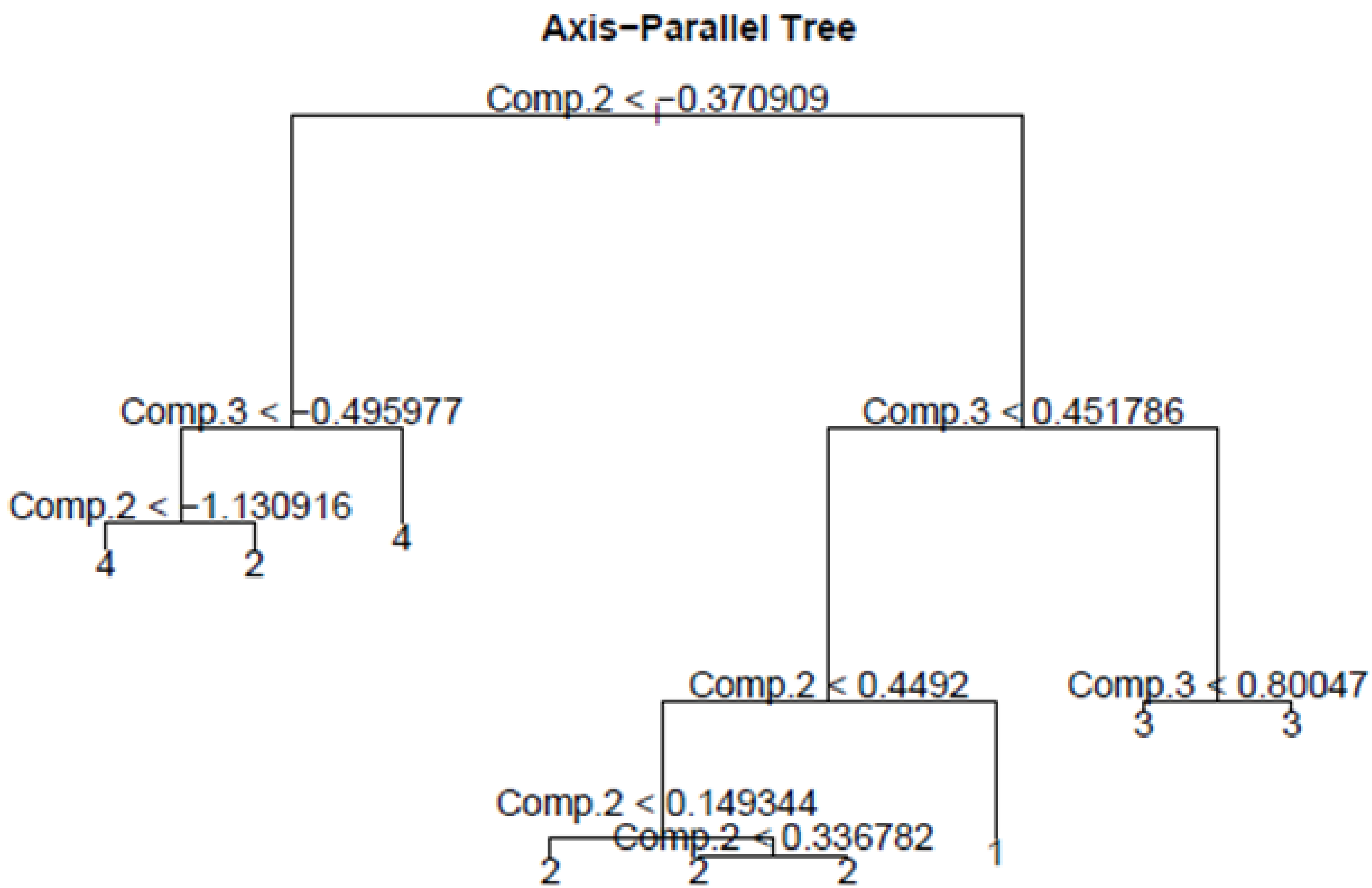
$$a_1x_1 + a_2x_2 + \cdots + c > or < K$$

Possible only for numeric attributes

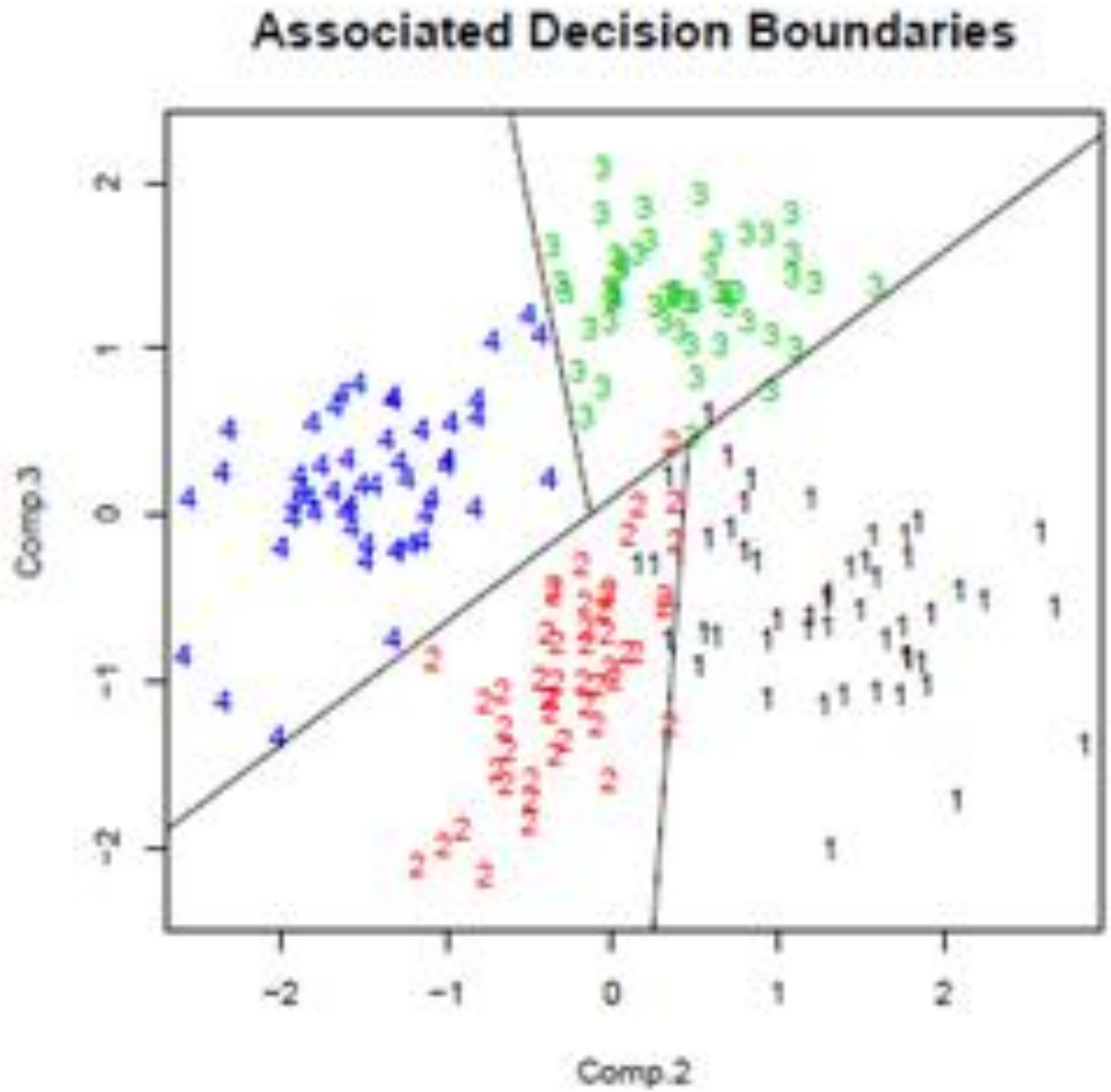
Associated Decision Boundaries



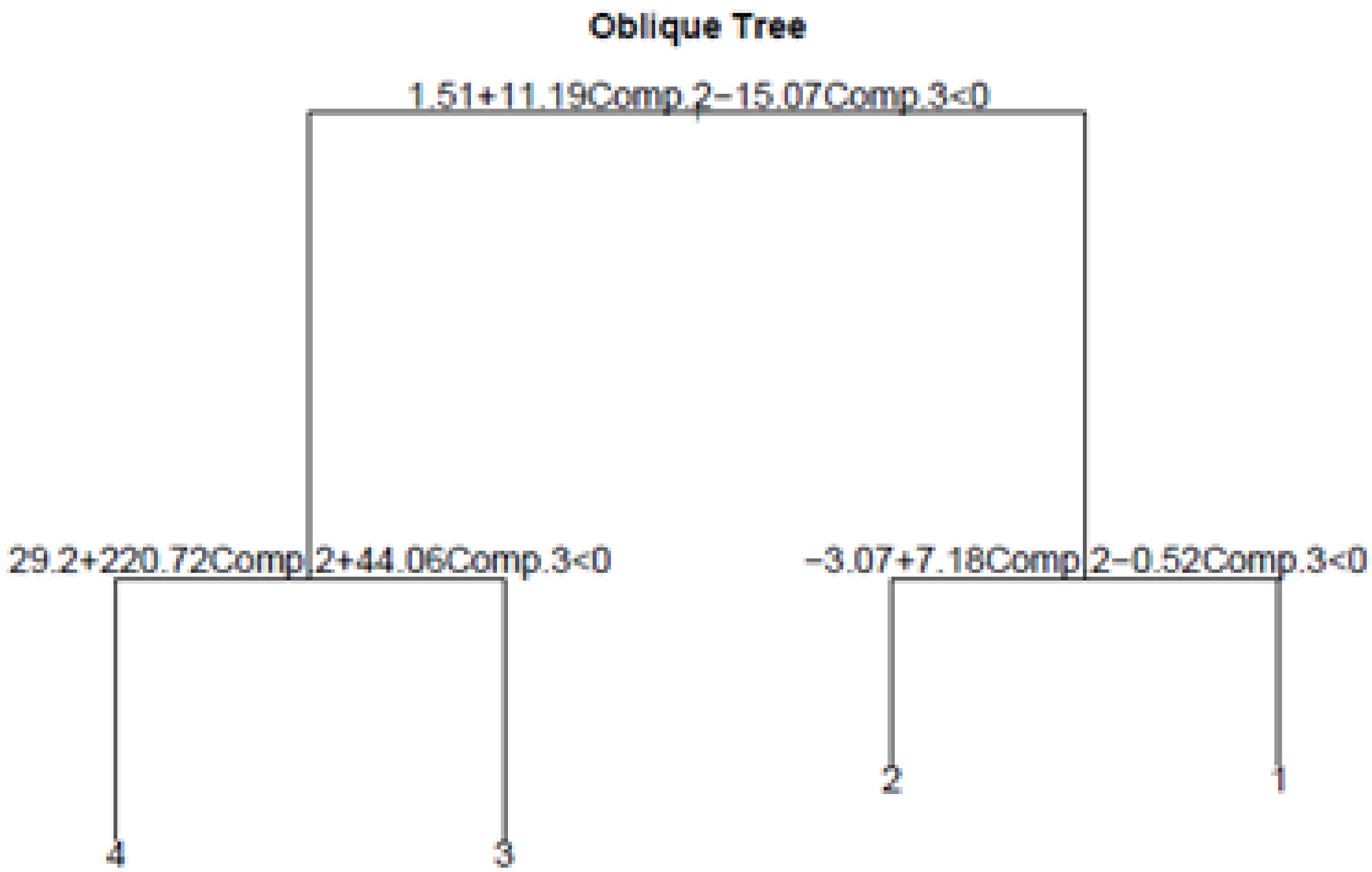
CART


Axis−Parallel Tree
Comp.2 < −0.370909
Comp.3 < −0.495977
Comp.3 < 0.451786
Comp.2 < −1.130916
4
4
2
Comp.2 < 0.4492
Comp.3 < 0.80047
3
3
Comp.2 < 0.149344
Comp.2 < 0.336782
2
2
2
1

Oblique

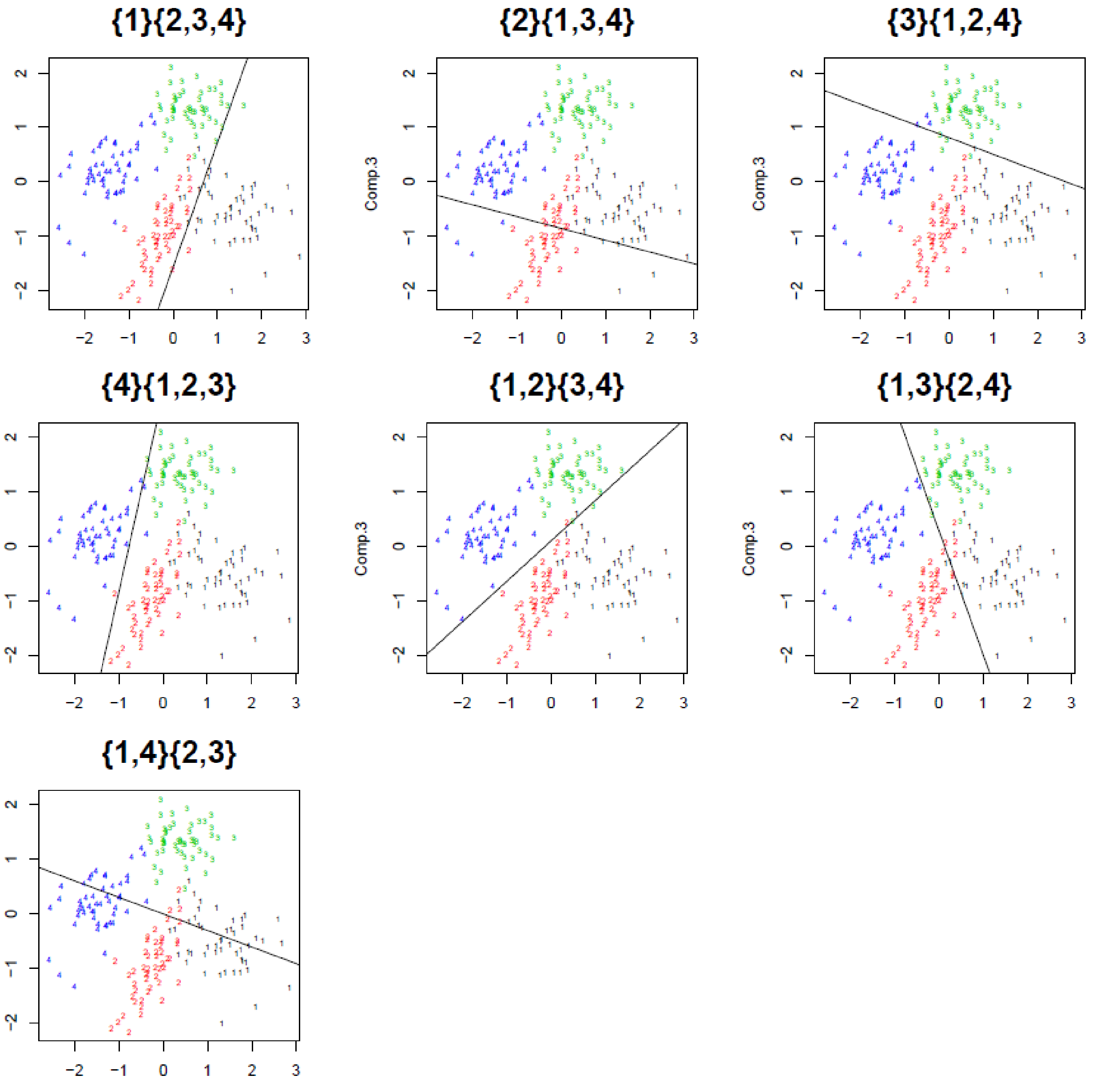


Oblique



Oblique Tree
1.51+11.19Comp.2−15.07Comp.3<0
29.2+220.72Comp.2+44.06Comp.3<0
−3.07+7.18Comp.2−0.52Comp.3<0
4
3
2
1

The choices of oblique planes are infinite



Oblique Tree

- Data set "S":
  - If there are 4 classes, we have 2^(4-1)-1 = 7 ways in which classification can happen
  - For each classification perform:
    - Attribute selection using ID3 method and logistic regression
    - Retain the attribute or the logistic regression based on information gain or impurity measure (m)
  - We will be left with 7 choices; select best among these
  - You will have two datasets S1 and S2
- Tree-growth proceeds with S1 and S2

Oblique Tree

Class	Attribute/Logistic	Information gain
{1}{2,3,4}	A1	0.67
	A2	0.12
	Logistic Regression	0.85
{2}{1,3,4}	A1	0.35
	A2	0.6
	Logistic Regression	0.17
…		
{1,2}{3,4}	A1	0.74
	A2	0.34
	Logistic Regression	0.82
…		

Let us say we had 2 attributes and 4 classes

At each step, we compute best split: axis parallel on these two attributes, logistic regression

We will have 7 * 3 = 21 ways in which to split data.

Select best method using a suitable metric

For example: We may pick {1}{2,3,4} and logistic regression

GOODNESS OF RULES

An if-then propositional rule

- If (x) and (y) and (z) then A

- x, y, z: Antecedent
- A: Consequent

- Length of a rule: Number of antecedents

"If CCAvg is medium then loan = accept"

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

This rule covers 3 in 13 samples, or 23% of data

This is called support

"If CCAvg is medium then loan = accept"

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

Of the three occasions left is present, right too is present.

This rule has 100% confidence

23% support

Lift

- Is confidence and support always good?
  - Example: Leak in oil gasket causes Engine rebuild
    - Prob of Engine rebuild will be very small, hence support for this rule is small
    - Confidence will be high

Age of car	Maximum speed	Tyre install date	Air pressure	Tyre failure
3	76	Nov-1976	33	Y
7	56	Feb-1982	28	N

  - Hence you use lift:
    - Confidence of rule / Overall prob of the event
    - i.e. Confidence of oil gasket rule / prob of engine rebuild

How do we define minimum support

- Rules with a certain minimum support alone are important. How do we know that?
  - Domain expertise
  - But, this is subjective

Montecarlo simulation

- Generate rules with all supports (do not prune any)
- Keep the independent part of the data as it is and randomly shuffle the class variable
- Measure the support of all the rules in the random data
- Iterate the process and compute mean support and SD of rules in the random data
- Only those rules whose support is more than 3 sds away are good rules

How do we know whether a rule is trivial?

- Ask the business user
- A short rule with high support and confidence

$$\frac{Support+Confidence}{Length}$$

Actionability

- If (the mother has B positive) and (smoked during pregnancy) and (the kid is eating a lot of carbohydrates) then the he/she is likely to get Asthma

- All three attributes have different actionabilities

Analysis of attributes in universal banks

- **Non-actionable**: Acts of God (weather), external factors (price of gold, rupee value etc.)
- **Actionable**: Age, experience, income, family, education
- **Actionable and changeable**: Mortgage, mortgage status, average credit card spending and other statistics, usage of other accounts (cc, cd, online & securities), infoReq

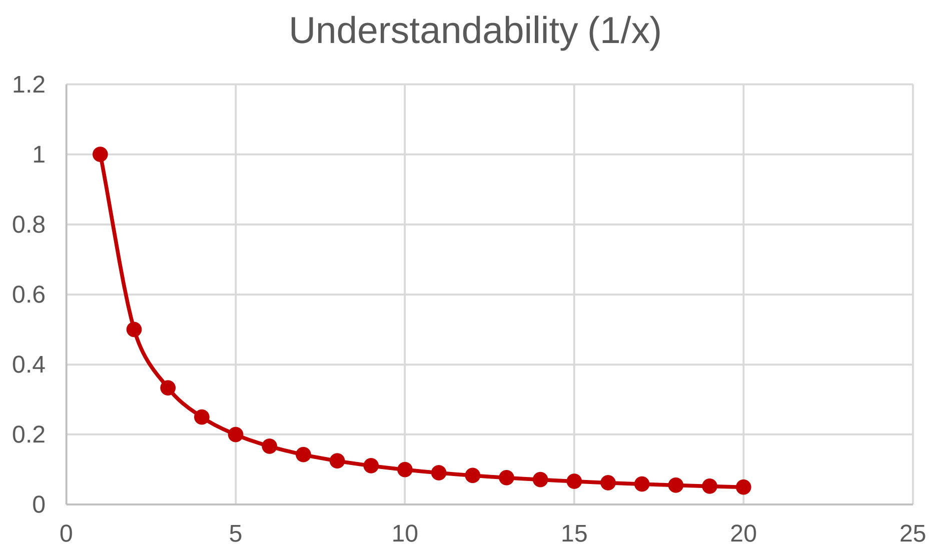
Actionability of a rule

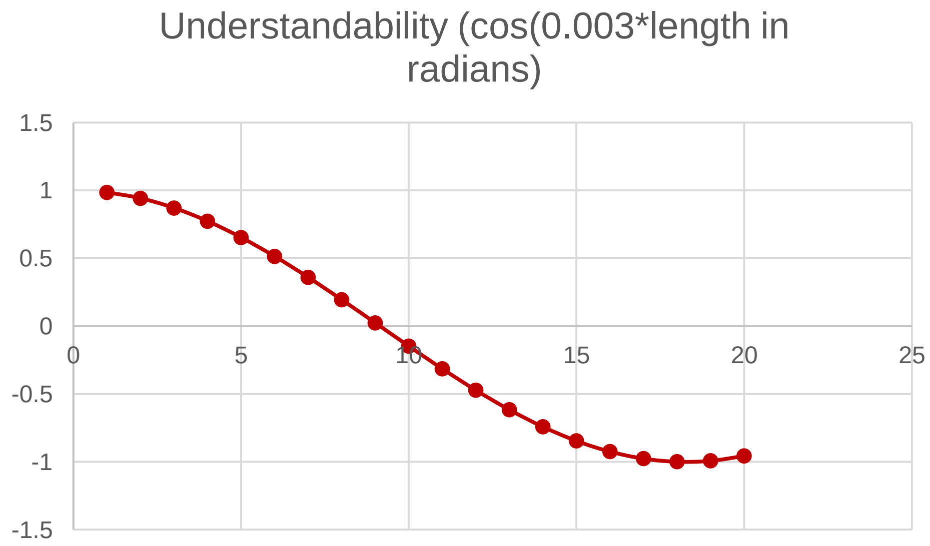
- $norm\left(\frac{\sum Actionability\ of\ antecedents}{Total\ number\ of\ attributes\ in\ the\ antecedent}\right)$

- If we take the numerator alone, a long rule and short rule with same actionability come out as equals

Understandability

- More precedents, less understandable





International School of Engineering

2-56/2/19, Khanamet, Madhapur, Hyderabad - 500 081

For Individuals:	+91-9177585755 or 040-65743991
For Corporates:	+91-9618483483
Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/intl.school.engineering
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