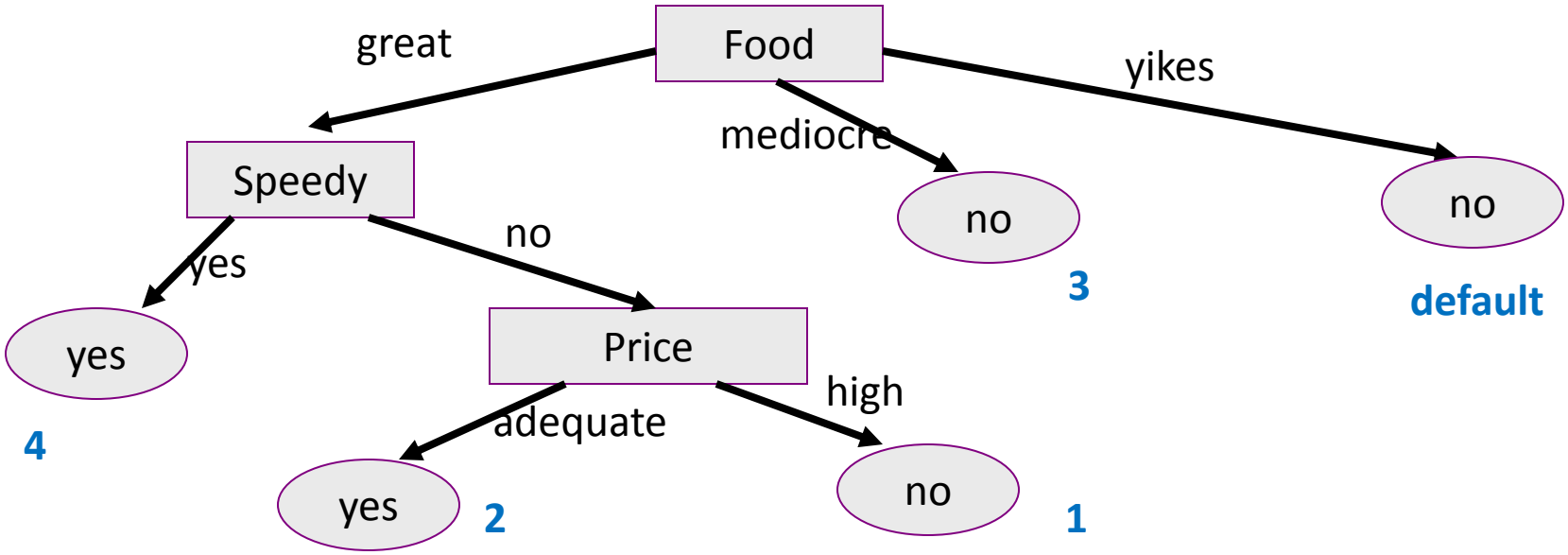
Decision Trees

Compacting Instances: Creating models

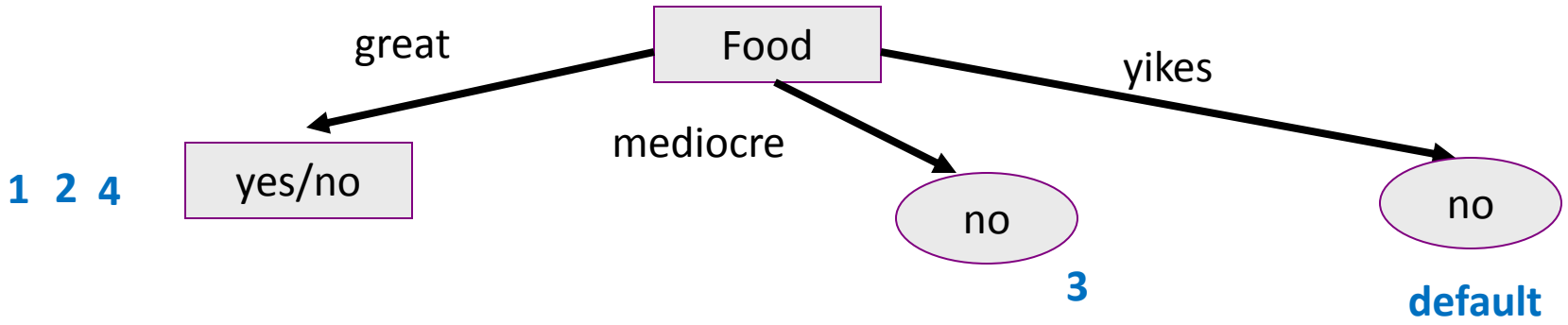
	Food (3)	Chat (2)	Speedy (2)	Price (2)	Bar (2)	BigTip
1	great	yes	yes	adequate	no	yes
2	great	no	yes	adequate	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	adequate	yes	yes

Decision Tree Example: BigTip



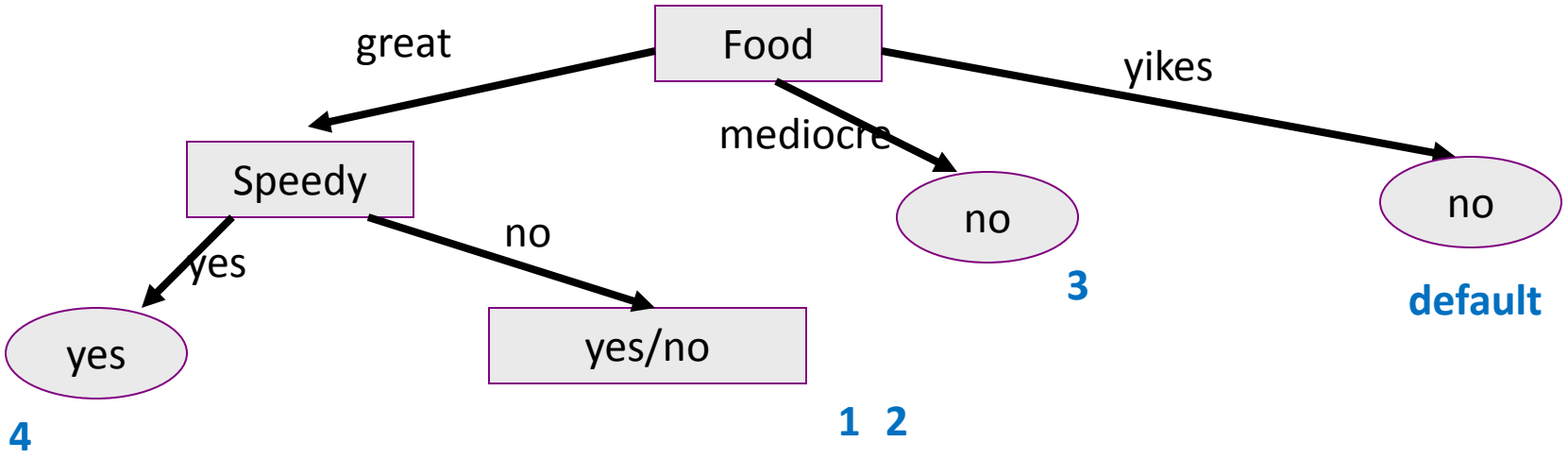
	Food (3)	Chat (2)	Speedy (2)	Price (2)	Bar (2)	BigTip
1	great	yes	no	high	no	no
2	great	no	no	adequate	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	adequate	yes	yes

Decision Tree Example: BigTip



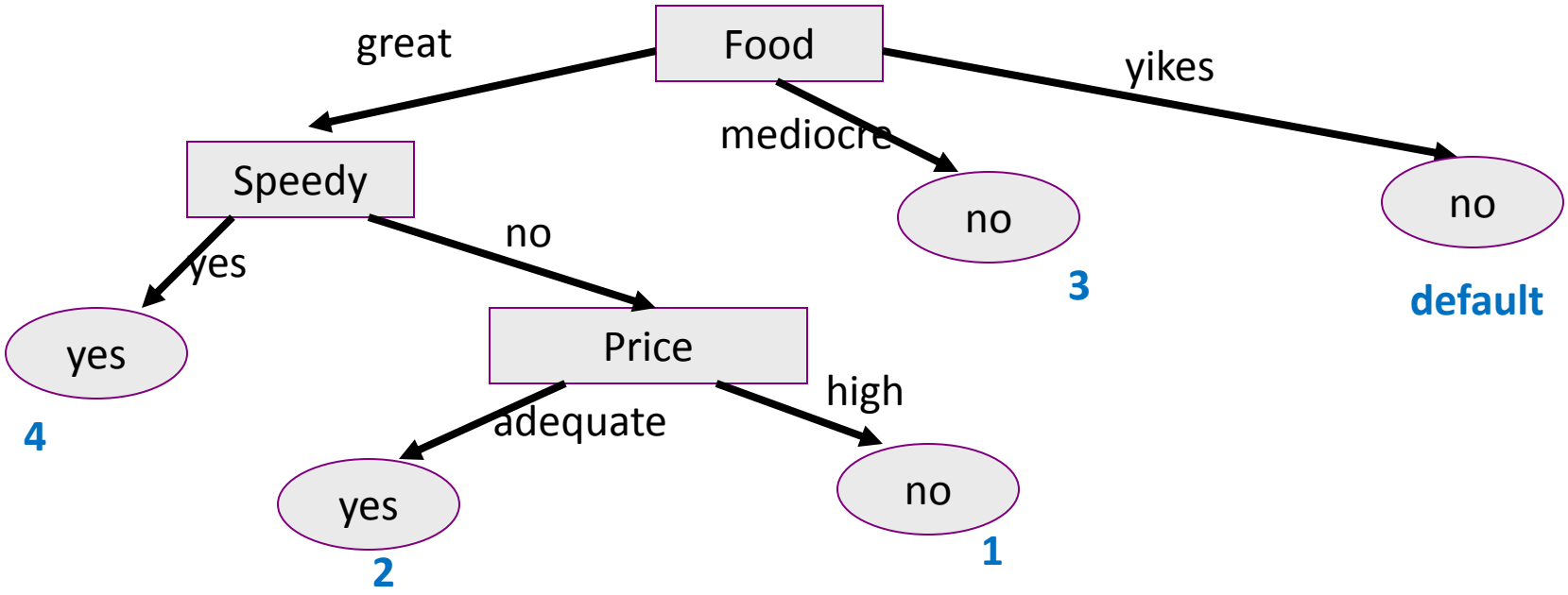
	Food (3)	Chat (2)	Speedy (2)	Price (2)	Bar (2)	BigTip
1	great	yes	no	high	no	no
2	great	no	no	adequate	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	adequate	yes	yes

Decision Tree Example: BigTip



	Food (3)	Chat (2)	Speedy (2)	Price (2)	Bar (2)	BigTip
1	great	yes	no	high	no	no
2	great	no	no	adequate	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	adequate	yes	yes

Decision Tree Example: BigTip



	Food (3)	Chat (2)	Speedy (2)	Price (2)	Bar (2)	BigTip
1	great	yes	no	high	no	no
2	great	no	no	adequate	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	adequate	yes	yes

Top-Down Induction of DT (simplified)

Training Data: $D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$

TDIDT(D,c_{def})

- IF(all examples in D have same class c)
  - Return leaf with class c (or class c_{def}, if D is empty)
- ELSE IF(no attributes left to test)
  - Return leaf with class c of majority in D
- ELSE
  - Pick A as the "best" decision attribute for next node
  - FOR each value v_i of A create a new descendent of node
    - $D_i = \{(\vec{x}, y) \in D : \text{attribute A of } \vec{x} \text{ has value } v_i\}$
    - Subtree t_i for v_i is TDIDT(D_i,c_{def})
  - RETURN tree with A as root and t_i as subtrees

Example: Text Classification

- Task: Learn rule that classifies Reuters Business News
  - Class +: “Corporate Acquisitions”
  - Class -: Other articles
  - 2000 training instances
- Representation:
  - Boolean attributes, indicating presence of a keyword in article
  - 9947 such keywords (more accurately, word “stems”)

LAROCHE STARTS BID FOR NECO SHARES +

Investor David F. La Roche of North Kingstown, R.I., said he is offering to purchase 170,000 common shares of NECO Enterprises Inc at 26 dlrs each. He said the successful completion of the offer, plus shares he already owns, would give him 50.5 pct of NECO's 962,016 common shares. La Roche said he may buy more, and possible all NECO shares. He said the offer and withdrawal rights will expire at 1630 EST/2130 gmt, March 30, 1987.

SALANT CORP 1ST QTR -
FEB 28 NET

Oper shr profit seven cts vs loss 12 cts. Oper net profit 216,000 vs loss 401,000. Sales 21.4 mln vs 24.9 mln. NOTE: Current year net excludes 142,000 dlr tax credit. Company operating in Chapter 11 bankruptcy.

Decision Tree for “Corporate Acq.”

```
vs = 1: -
vs = 0:
|   export = 1:
…
|   export = 0:
|   |   rate = 1:
|   |   |   stake = 1: +
|   |   |   stake = 0:
|   |   |   |   debenture = 1: +
|   |   |   |   debenture = 0:
|   |   |   |   |   takeover = 1: +
|   |   |   |   |   takeover = 0:
|   |   |   |   |   |   file = 0: -
|   |   |   |   |   |   file = 1:
|   |   |   |   |   |   |   share = 1: +
|   |   |   |   |   |   |   share = 0: -
… and many more
```

Total size of tree:

- 299 nodes

Note: word stems expanded for improved readability.

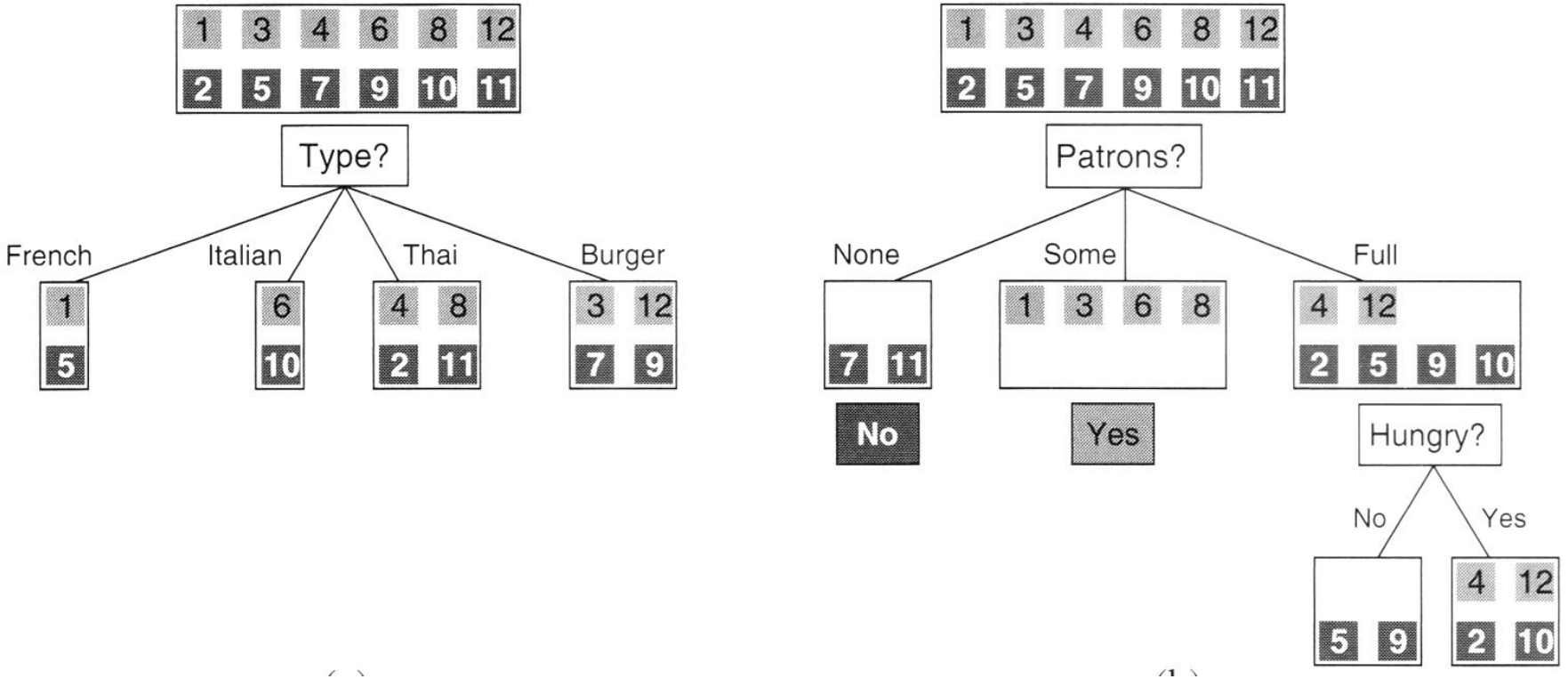
20 Questions

- I choose a number between 1 and 1000
- You try to find it using yes/no questions
- Which question is more informative?
  - Is the number 634?
  - Is the number a prime?
  - Is the number smaller than 500?

Should we wait?

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

Maximum Separation



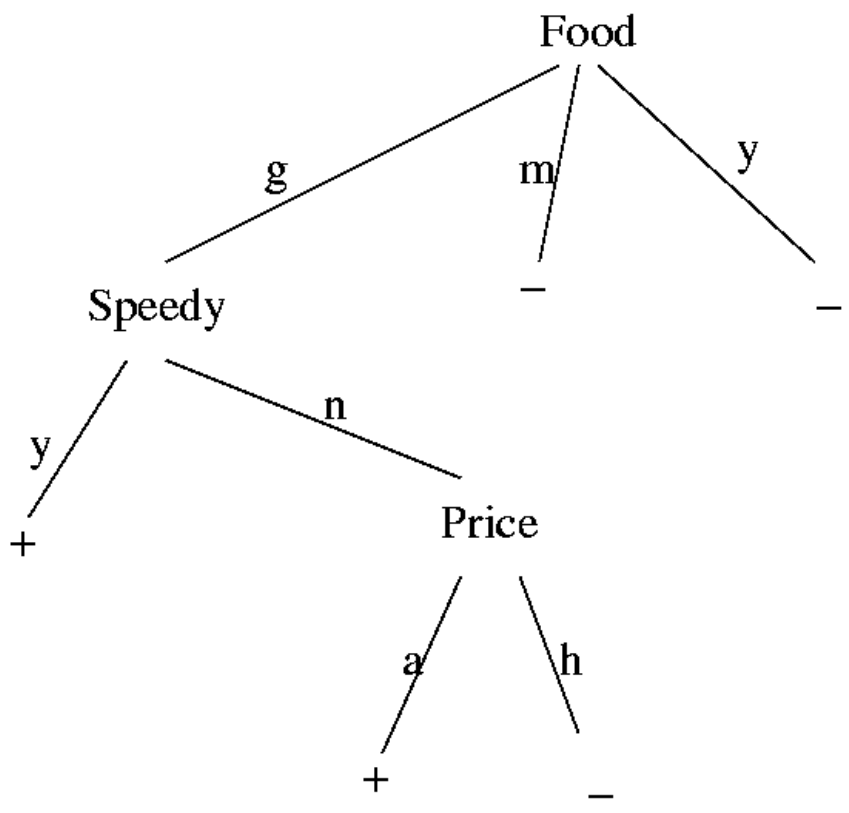
Example: TDIDT

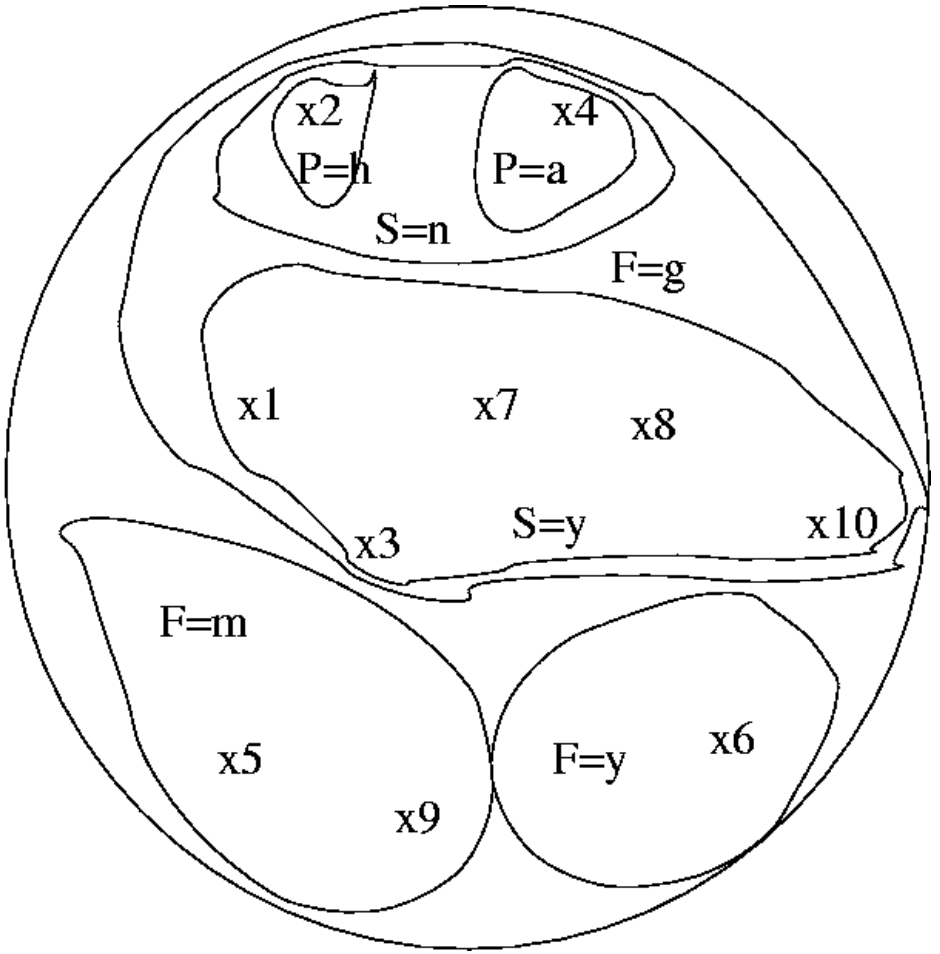
Training Data D:

	F S P	BigTip
$\vec{x}_1 =$	(g, y, a)	$f(\vec{x}_1) = 1$
$\vec{x}_2 =$	(g, n, h)	$f(\vec{x}_2) = 0$
$\vec{x}_3 =$	(g, y, h)	$f(\vec{x}_3) = 1$
$\vec{x}_4 =$	(g, n, a)	$f(\vec{x}_4) = 1$
$\vec{x}_5 =$	(m, y, a)	$f(\vec{x}_5) = 0$
$\vec{x}_6 =$	(y, y, a)	$f(\vec{x}_6) = 0$
$\vec{x}_7 =$	(g, y, a)	$f(\vec{x}_7) = 1$
$\vec{x}_8 =$	(g, y, h)	$f(\vec{x}_8) = 1$
$\vec{x}_9 =$	(m, y, a)	$f(\vec{x}_9) = 0$
$\vec{x}_{10} =$	(g, y, a)	$f(\vec{x}_{10}) = 1$

Which is the best decision variable? **A=F,** **B=S,** **C=P**

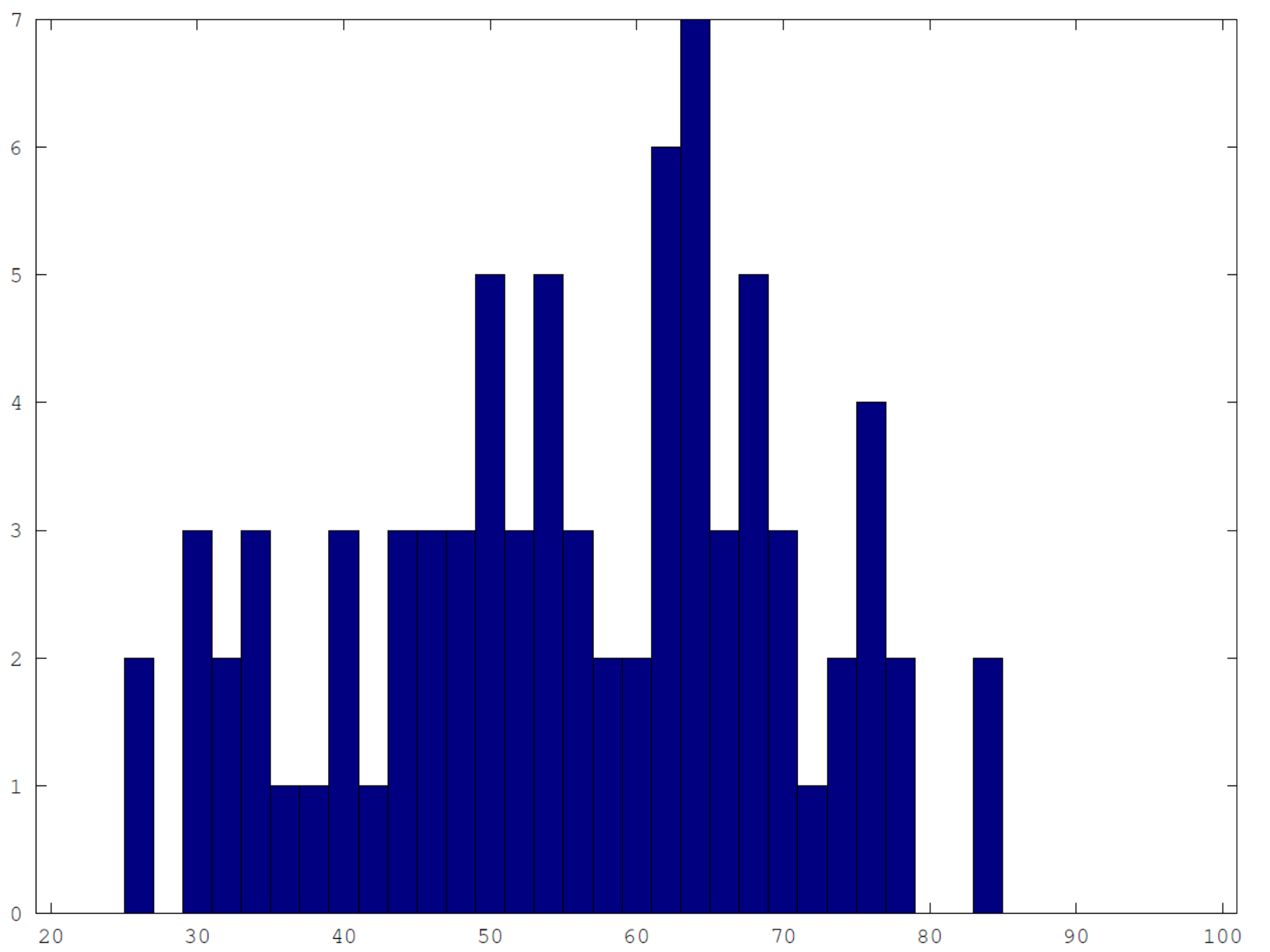
TDIDT Example





Picking the Best Attribute to Split

- Ockham's Razor:
  - All other things being equal, choose the simplest explanation
- Decision Tree Induction:
  - Find the smallest tree that classifies the training data correctly
- Problem
  - Finding the smallest tree is computationally hard
- Approach
  - Use heuristic search (greedy search)


0
1
2
3
4
5
6
7
20
30
40
50
60
70
80
90
100

Maximum information

- Information in a set of choices

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \log_2 P(v_i)$$

  - E.g. Information in a flip of a fair coin

$$I\left(\tfrac{1}{2}, \tfrac{1}{2}\right) = -\tfrac{1}{2}\log_2 \tfrac{1}{2} - \tfrac{1}{2}\log_2 \tfrac{1}{2} = 1 \text{ bit.}$$

  - Information in an unfair (99:1) coin:
    - I(1/100, 99/100) = 0.08
- Information in full classification of (p,n) samples

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2 \frac{p}{p+n} - \frac{n}{p+n}\log_2 \frac{n}{p+n}$$

Maximum information

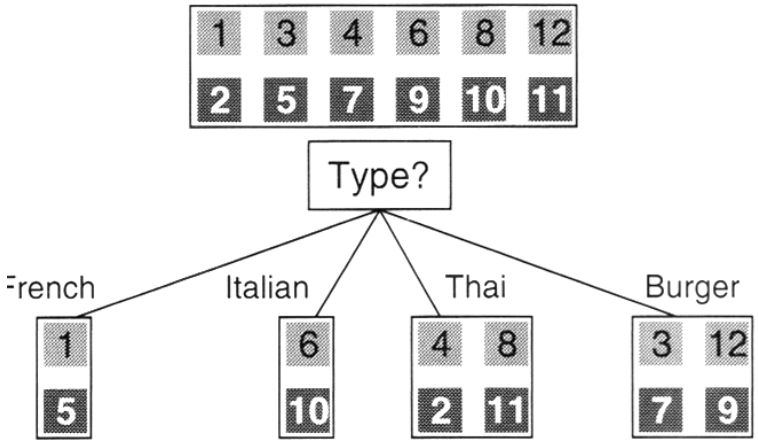
- After classification by attribute A

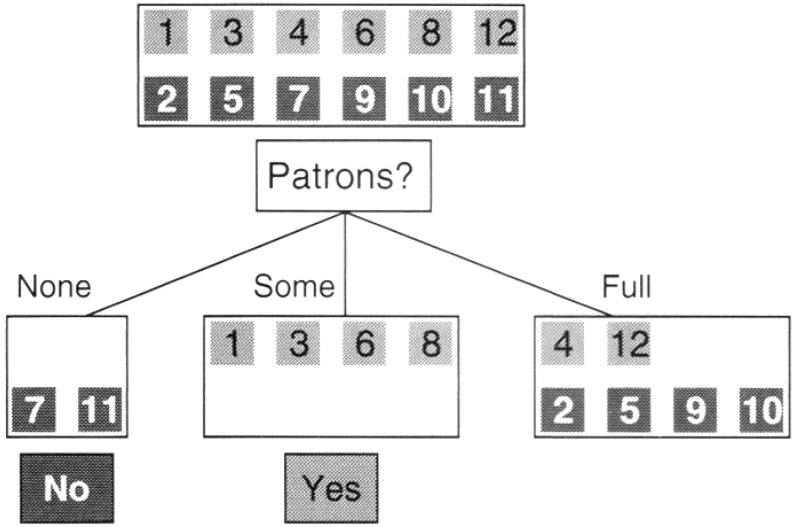
$$Remainder(A) = \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

- Information Gain by attribute A

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - Remainder(A)$$

Information gain





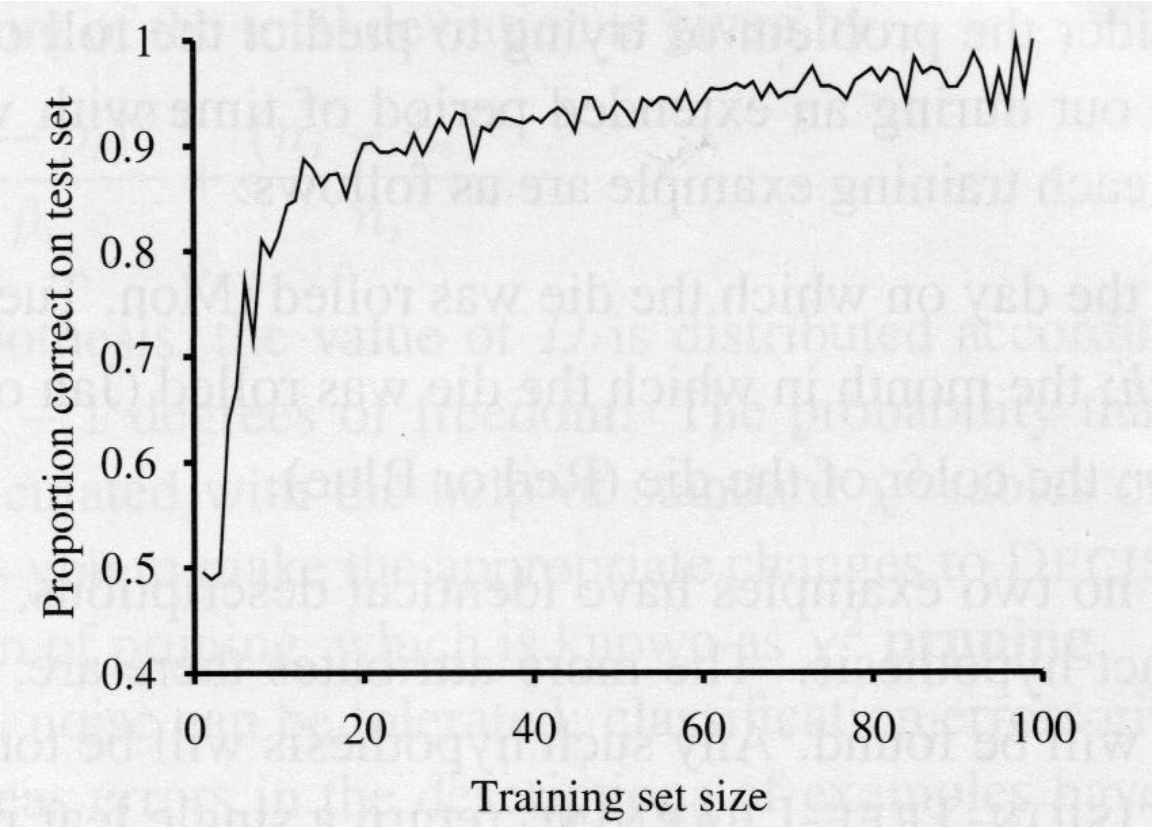
$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Which attribute has higher information gain?

A=Type B=Patrons C=Neither

Learning curve

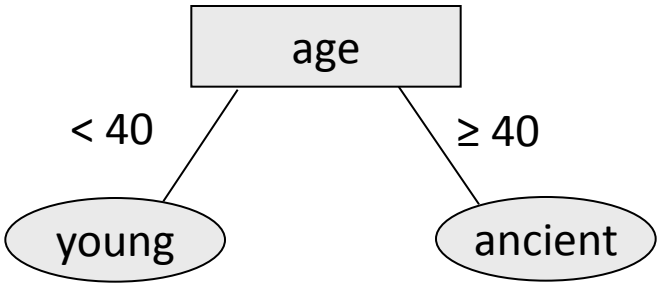
– Success as function of training set size

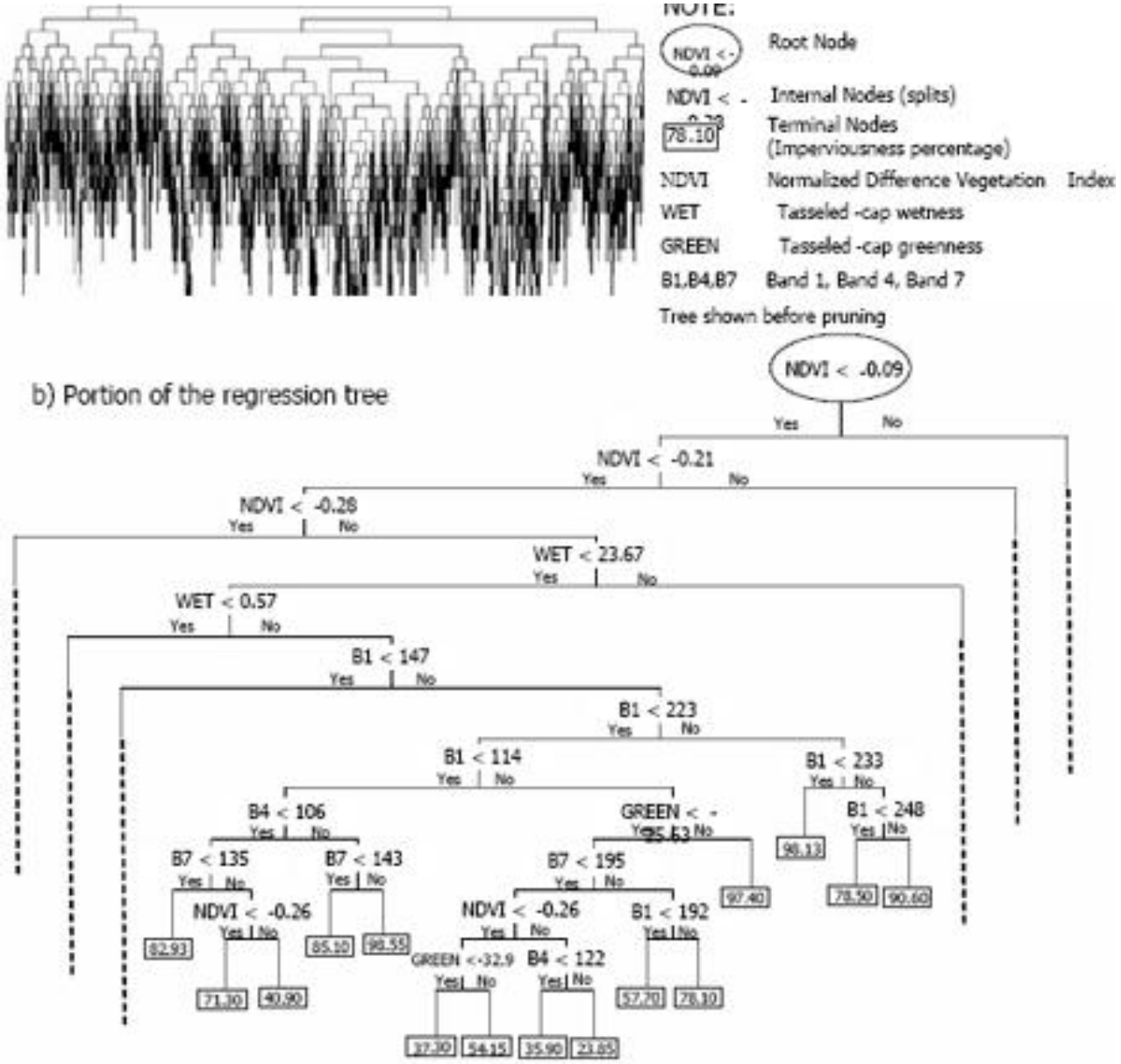


A hard problem will have a: A-Steep B-Shallow learning curve

Continuous variables?

- Look for optimal split point.

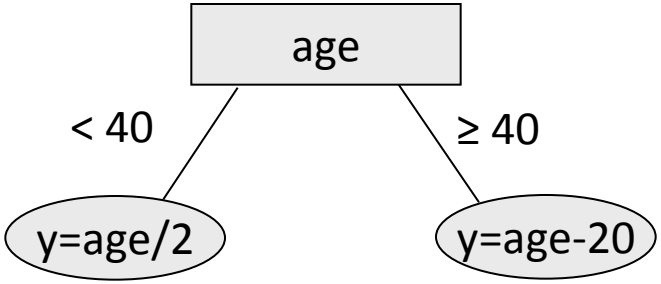




From: http://www.gisdevelopment.net/technology/rs/images/ma06110_8.jpg

Continuous output?

- Regression trees



Spurious attributes?

- Cross validation
- Information gain ratio
  - Normalize information gain

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - Remainder(A)$$

by the net information in the attribute itself

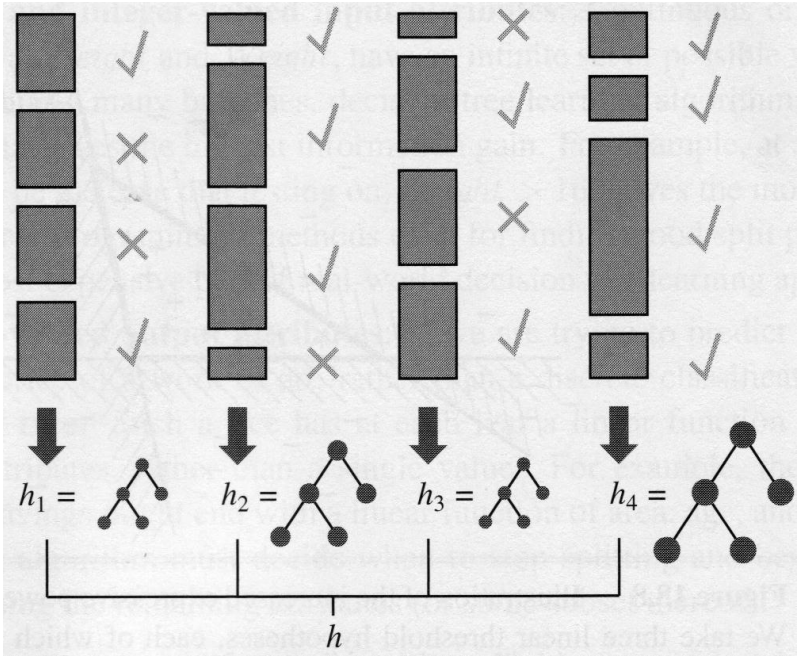
$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \log_2 P(v_i)$$

Datapoint Weighting

- How can we give certain datapoints more importance than others?
  - Introduce weight factors in kNN
  - What about decision trees?
    - Duplicate points
    - Give more weight when choosing attributes

Ensemble learning

- Boosting: Create multiple classifiers that vote
  - Give more weight to wrongly classified samples
  - E.g. sum of incorrectly classified weights equals sum of correctly classified



Ensemble learning

- If the input algorithm L is a weak algorithm (>50%), then AdaBoost will return a perfect algorithm for large enough M