# AIN311
## Fundamentals of Machine Learning

### Lecture 18:
#### Decision Trees

Erkut Erdem // Hacettepe University // Fall 2024

HACETTEPE UNIVERSITY COMPUTER VISION LAB

# Today

- Decision Trees
- Tree construction
- Overfitting
- Pruning
- Real-valued inputs
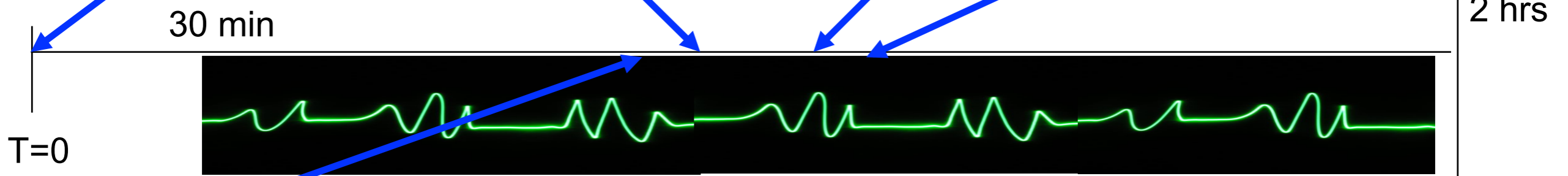
# Machine Learning in the ER

Physician documentation

Triage Information (Free text)

MD comments (free text)

Specialist consults

30 min

T=0

2 hrs

Disposition

Repeated vital signs (continuous values)
Measured every 30 s

Lab results (Continuous valued)

3

# Can we predict infection?

Triage Information
(Free text)

Specialist consults

MD comments
(free text)

Many crucial decisions about a patient's care are made here!

Lab results
(Continuous valued)

Repeated vital signs
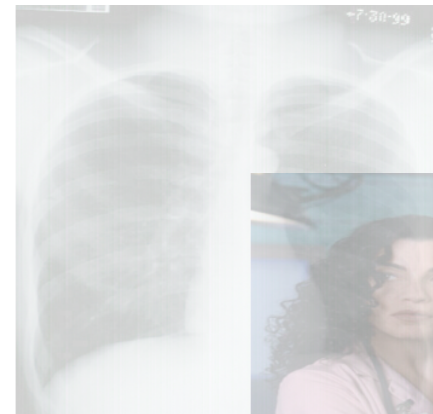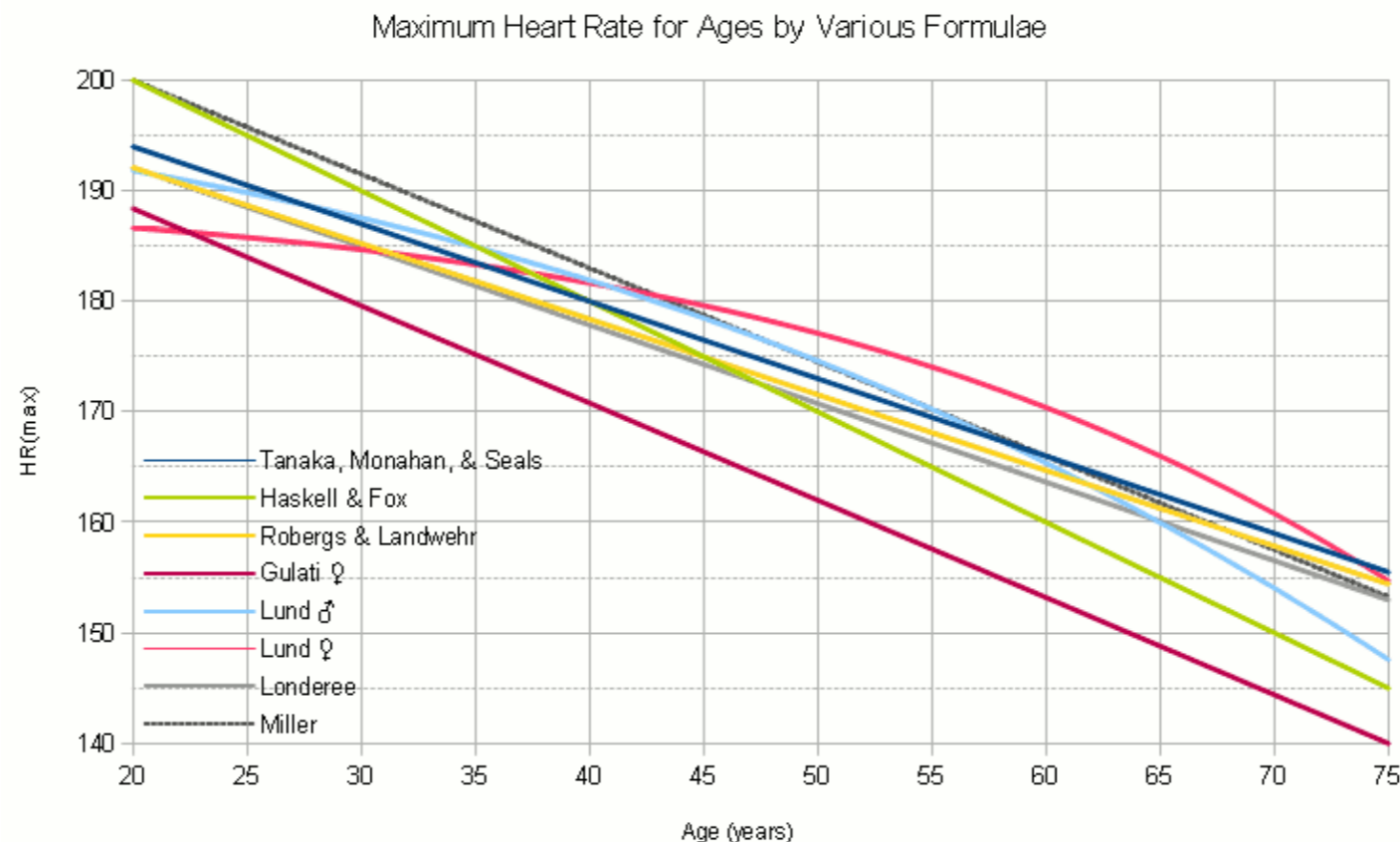(continuous values)
Measured every 30 s
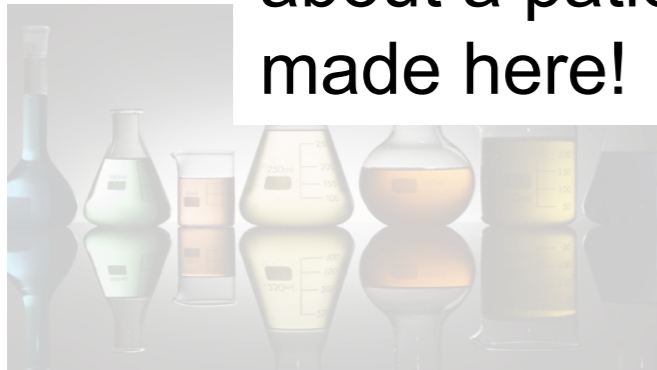
# Can we predict infection

- Previous automatic approaches based on simple criteria:
  - Temperature < 96.8 °F or > 100.4 °F
  - Heart rate > 90 beats/min
  - Respiratory rate > 20 breaths/min

- Too simplified... e.g., heart rate depends on age!



Maximum Heart Rate for Ages by Various Formulae

5

# Can we predict infection?

- These are the attributes we have for each patient:
  - Temperature
  - Heart rate (HR)
  - Respiratory rate (RR)
  - Age
  - Acuity and pain level
  - Diastolic and systolic blood pressure (DBP, SBP)
  - Oxygen Saturation (SaO2)
- We have these attributes + label (infection) for 200,000 patients!
- Let's **learn** to classify infection

slide by David Sontag

# Predicting infection using decision trees
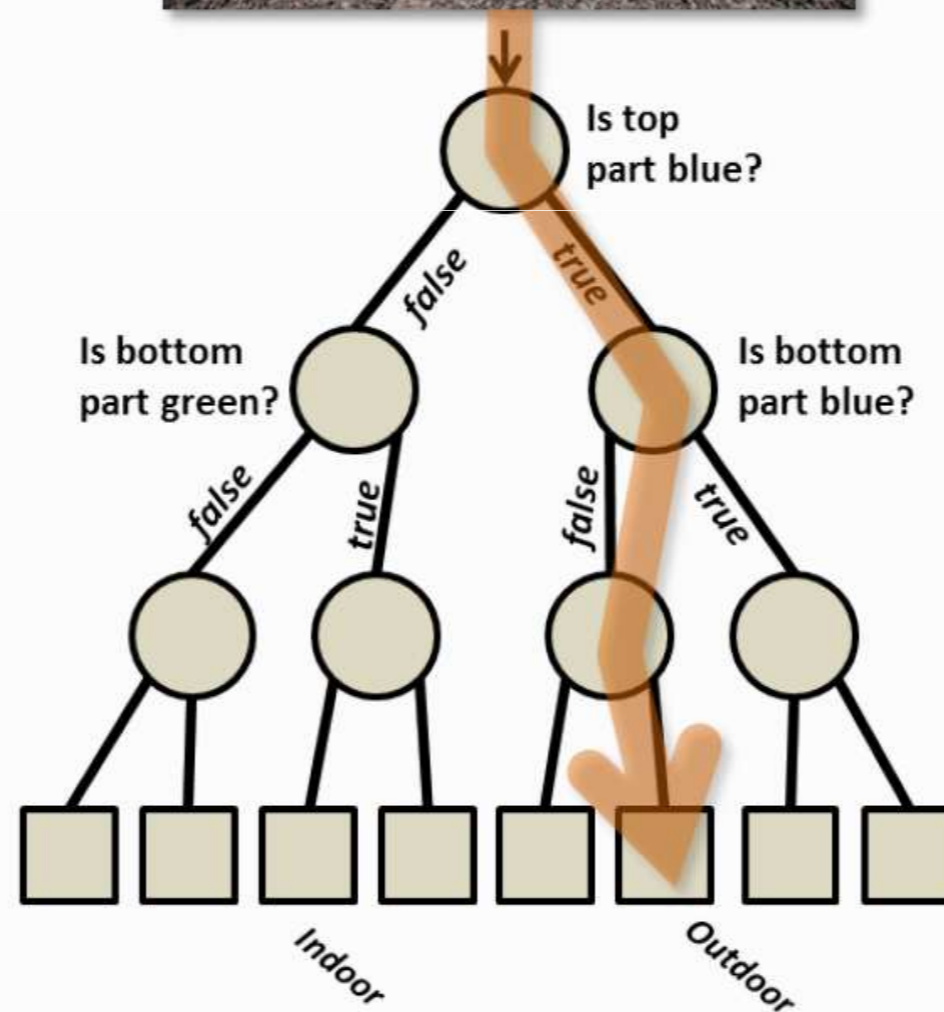


TriageTemp <= 100.9500
error = 0.381403449603
samples = 1657
value = [ 425. 1232.]

Age <= 41.5000
error = 0.465677281203
samples = 729
value = [ 269. 460.]

TriageSBP <= 112.5000
error = 0.279689357907
samples = 928
value = [ 156. 772.]

TriageHR <= 108.5000
error = 0.496029801
samples = 303
value = [ 138. 165.]

TriagePain <= 0.5000
error = 0.425896537283
samples = 426
value = [ 131. 295.]

error = 0.1413
samples = 183
value = [ 14. 169.]

TriageTemp <= 101.4500
error = 0.308548263592
samples = 745
value = [ 142. 603.]

error = 0.4970
samples = 169
value = [ 91. 78.]

error = 0.4554
samples = 134
value = [ 47. 87.]

error = 0.3471
samples = 170
value = [ 38. 132.]

TriageHR <= 98.5000
error = 0.462615966797
samples = 256
value = [ 93. 163.]

Age <= 51.5000
error = 0.366668906208
samples = 244
value = [ 59. 185.]

TriageRR <= 18.5000
error = 0.276445113764
samples = 501
value = [ 83. 418.]

error = 0.4861
samples = 132
value = [ 55. 77.]

error = 0.4251
samples = 124
value = [ 38. 86.]

error = 0.4064
samples = 141
value = [ 40. 101.]

error = 0.3009
samples = 103
value = [ 19. 84.]

error = 0.3118
samples = 326
value = [ 63. 263.]

error = 0.2024
samples = 175
value = [ 20. 155.]

7

# Example: Image Classification



**A decision tree**

Is top part blue?

Is bottom part blue?

*true*

Outdoor

i et al, 2011]

# Example: Mushrooms



**Mushroom cap shapes**

convex    bell-shaped    conical    knobbed    flat    sunken

**Mushroom cap surfaces**

smooth    velvety    hairy or fibrous    raised scales    flat scales    patches

**Annular rings**

pendant    flaring    sheathing    double    cobwebby    ring zone

http://www.usask.ca/biology/fungi/
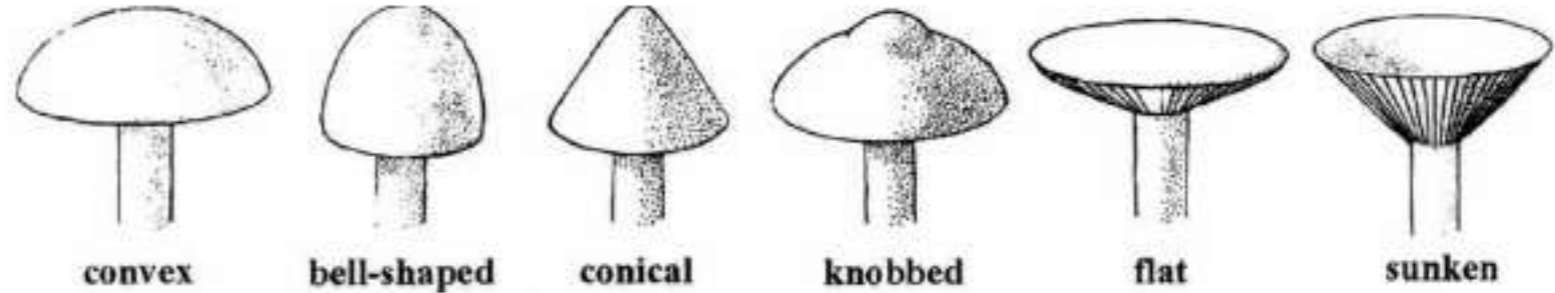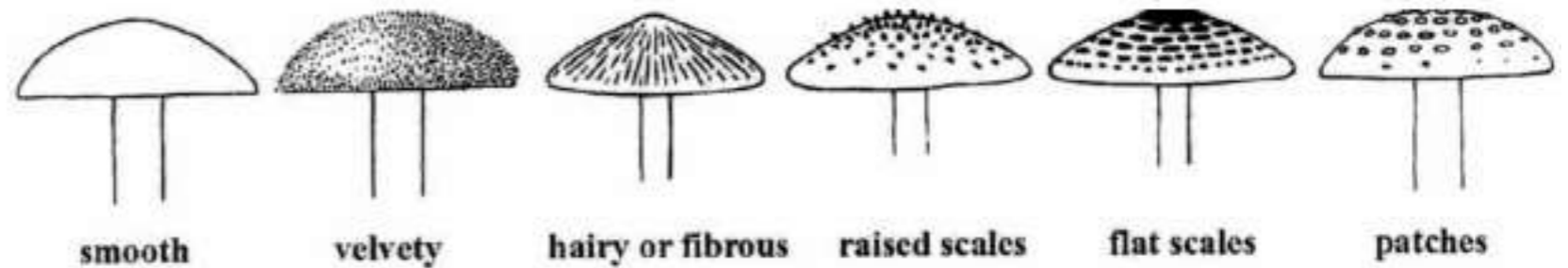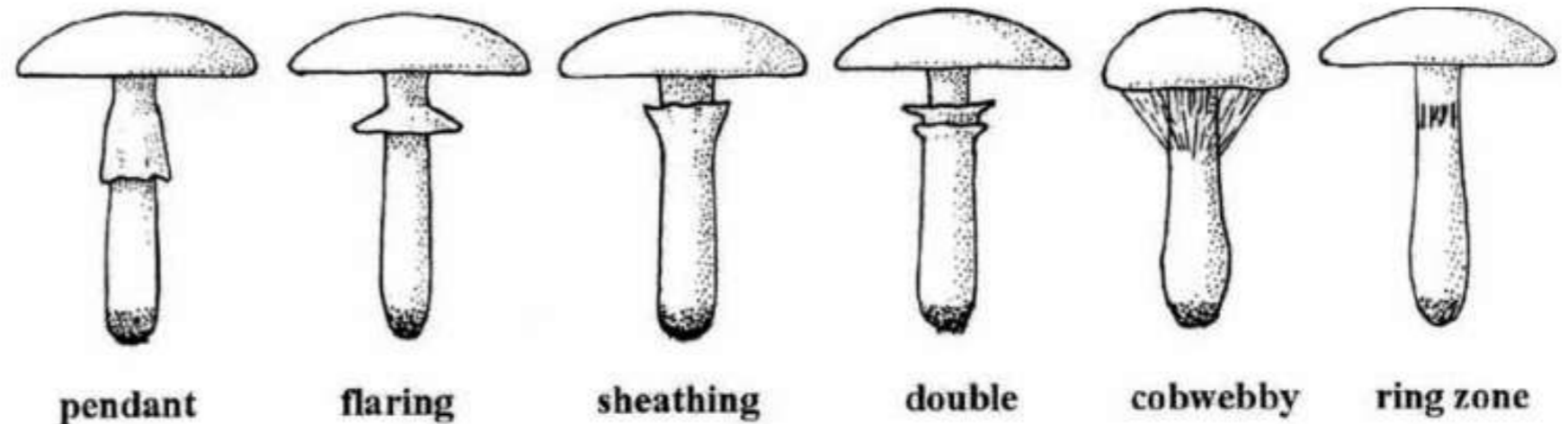
# Mushroom features

1. **cap-shape:** bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s

2. **cap-surface:** fibrous=f, grooves=g, scaly=y, smooth=s

3. **cap-color:** brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p,purple=u, red=e, white=w, yellow=y

4. **bruises?:** bruises=t,no=f

5. **odor:** almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s

6. **gill-attachment:** attached=a, descending=d, free=f, notched=n

7. ...

# Two mushrooms

$x_1$=x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u

$y_1$=p

$x_2$=x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g

$y_2$=e

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. …

# Example: Automobile Miles-per-gallon prediction



| mpg | cylinders | displacement | horsepower | weight | acceleration | modelyear | maker |
|------|-----------|--------------|------------|--------|--------------|-----------|---------|
| good | 4 | low | low | low | high | 75to78 | asia |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | medium | medium | medium | low | 75to78 | europe |
| bad | 8 | high | high | high | low | 70to74 | america |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | low | medium | low | medium | 70to74 | asia |
| bad | 4 | low | medium | low | low | 70to74 | asia |
| bad | 8 | high | high | high | low | 75to78 | america |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 8 | high | medium | high | high | 79to83 | america |
| bad | 8 | high | high | high | low | 75to78 | america |
| good | 4 | low | low | low | low | 79to83 | america |
| bad | 6 | medium | medium | medium | high | 75to78 | america |
| good | 4 | medium | low | low | low | 79to83 | america |
| good | 4 | low | low | medium | high | 79to83 | america |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 4 | low | medium | low | medium | 75to78 | europe |
| bad | 5 | medium | medium | medium | medium | 75to78 | europe |

# Hypotheses: decision trees $f : X \rightarrow Y$

- Each internal node tests an attribute $x_i$

- Each branch assigns an attribute value $x_i = v$

- Each leaf assigns a class $y$

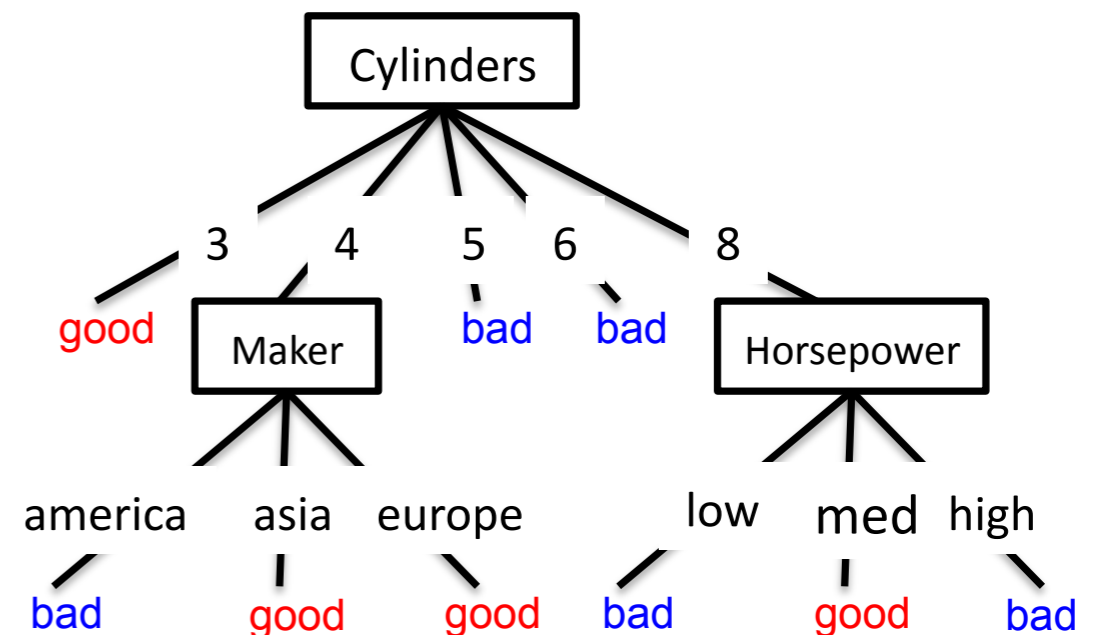- To classify input $x$: traverse the tree from root to leaf, output the labeled $y$
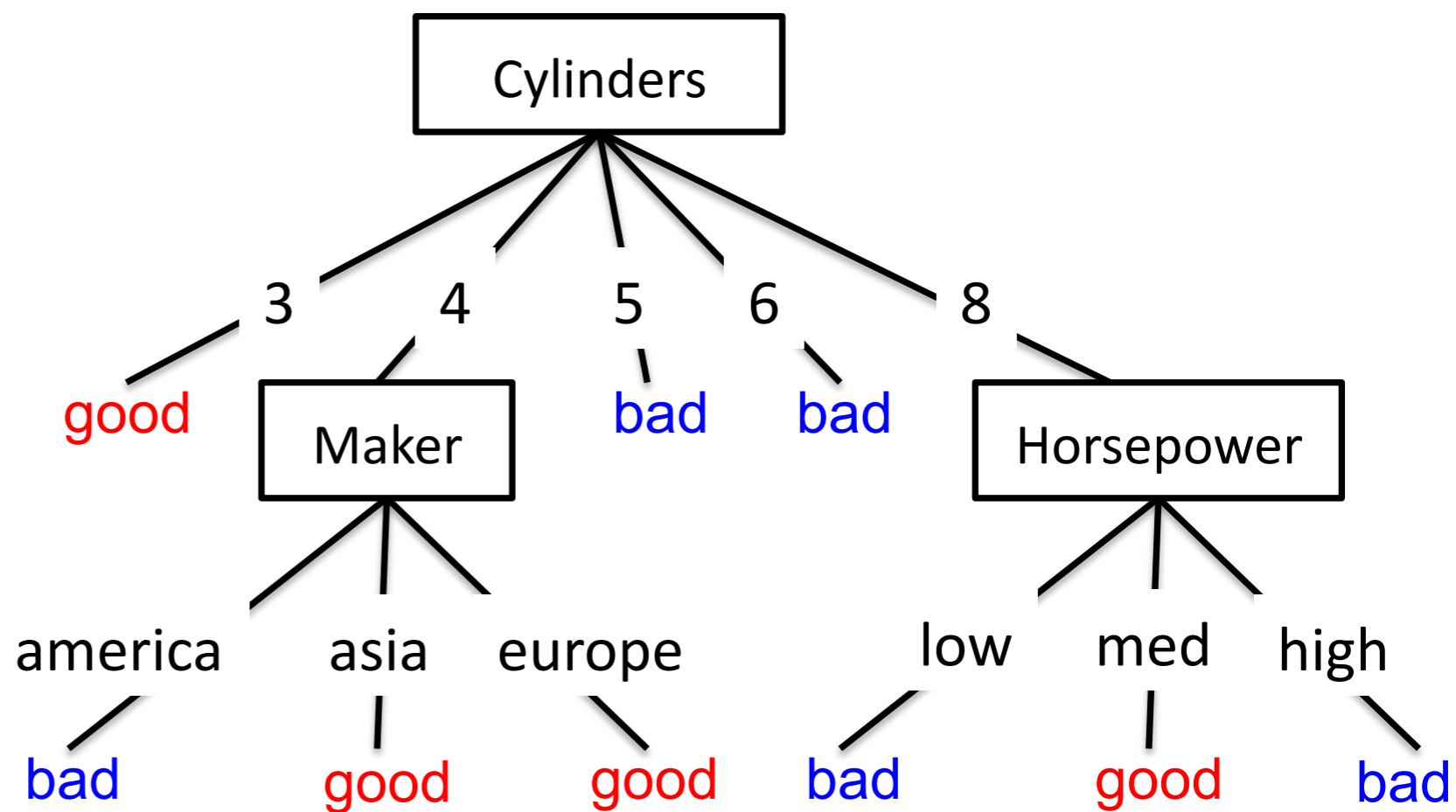


**Human interpretable!**

# Hypothesis space

- How many possible hypotheses?
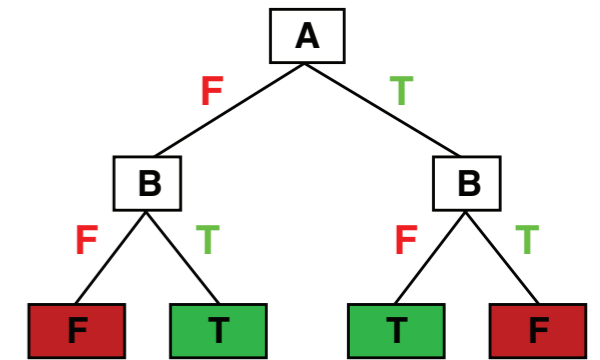
- What functions can be represented?

| mpg | cylinders | displacement | horsepower | weight | acceleration | modelyear | maker |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| good | 4 | low | low | low | high | 75to78 | asia |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | medium | medium | medium | low | 75to78 | europe |
| bad | 8 | high | high | high | low | 70to74 | america |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | low | medium | low | medium | 70to74 | asia |
| bad | 4 | low | medium | low | low | 70to74 | asia |
| bad | 8 | high | high | high | low | 75to78 | america |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 8 | high | medium | high | high | 79to83 | america |
| bad | 8 | high | high | high | low | 75to78 | america |
| good | 4 | low | low | low | low | 79to83 | america |
| bad | 6 | medium | medium | medium | high | 75to78 | america |
| good | 4 | medium | low | low | low | 79to83 | america |
| good | 4 | low | low | medium | high | 79to83 | america |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 4 | low | medium | low | medium | 75to78 | europe |
| bad | 5 | medium | medium | medium | medium | 75to78 | europe |

Cylinders

3 → good
4 → Maker
5 → bad
6 → bad
8 → Horsepower

Maker:
america → bad
asia → good
europe → good

Horsepower:
low → bad
med → good
high → bad

14

# What functions can be represented?

Discrete-input, discrete-output case:
– Decision trees can express any function of the input attrib
– E.g., for Boolean functions, truth table row → path to lea
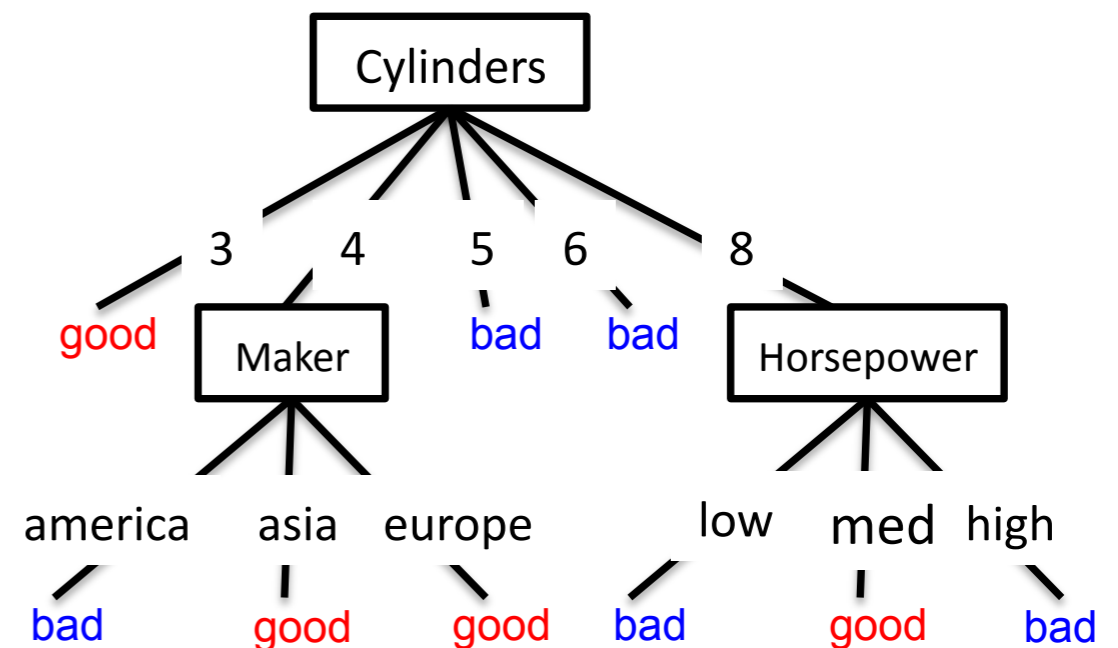
- Decision trees can represent any function of the input attributes!

- For Boolean functions, path to leaf gives truth table row

- But, could require exponentially many nodes…

| A | B | A xor B |
|---|---|---------|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |

(Figure from Stuart Russell)

Continuous-input, continuous-output case:
– Can approximate any function arbitrarily closely

Trivially, there is a consistent decision tree for any training set
w/ one path to leaf for each example (unless $f$ nondeterministi
but it probably won't generalize to new examples

Need some kind of regularization to ensure more **compact** de

Cylinders

3   4   5   6   8

good          bad   bad

Maker                    Horsepower

america   asia   europe        low   med   high

bad    good    good    bad    good    bad

CS194-1

cyl=3 ∨ (cyl=4 ∧ (maker=asia ∨ maker=europe)) ∨ …

slide by David Sontag

15

# Are all decision trees equal?

- Many trees can represent the same concept
- But, not all trees will have the same size
  - e.g., φ=(A∧B)∨(¬A∧ C) — ((A and B) or (not A and C))
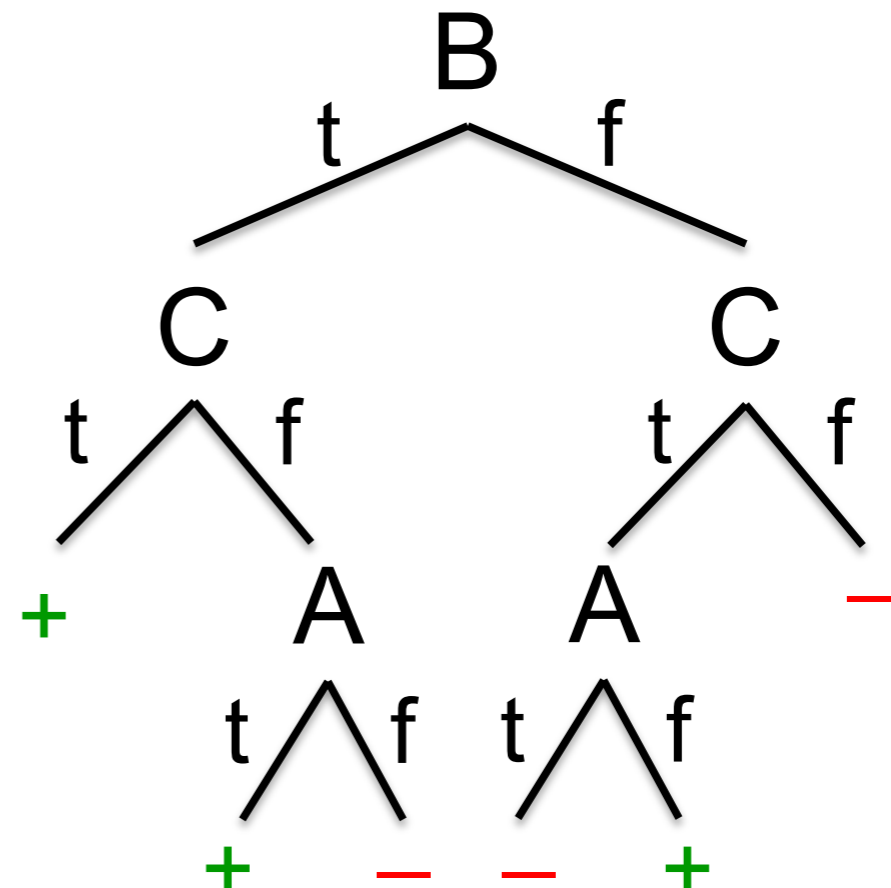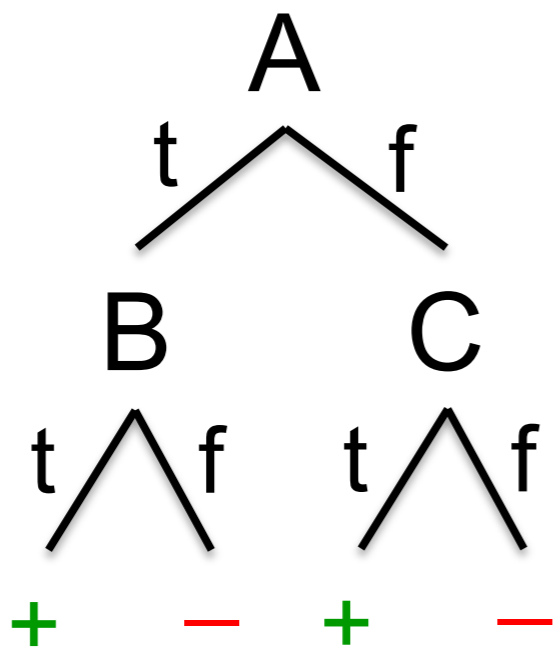


- Which tree do we prefer?

# Learning decision trees is hard!!!

- Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest '76]

- Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on next best attribute (feature)
  - Recurse

# A Decision Stump

mpg values: bad good

root

22 18

pchance = 0.001

Internal node question: "what is the number of cylinders"?

| cylinders = 3 | cylinders = 4 | cylinders = 5 | cylinders = 6 | cylinders = 8 |
|---|---|---|---|---|
| 0 0 | 4 17 | 1 0 | 8 0 | 9 1 |
| Predict bad | Predict good | Predict bad | Predict bad | Predict bad |
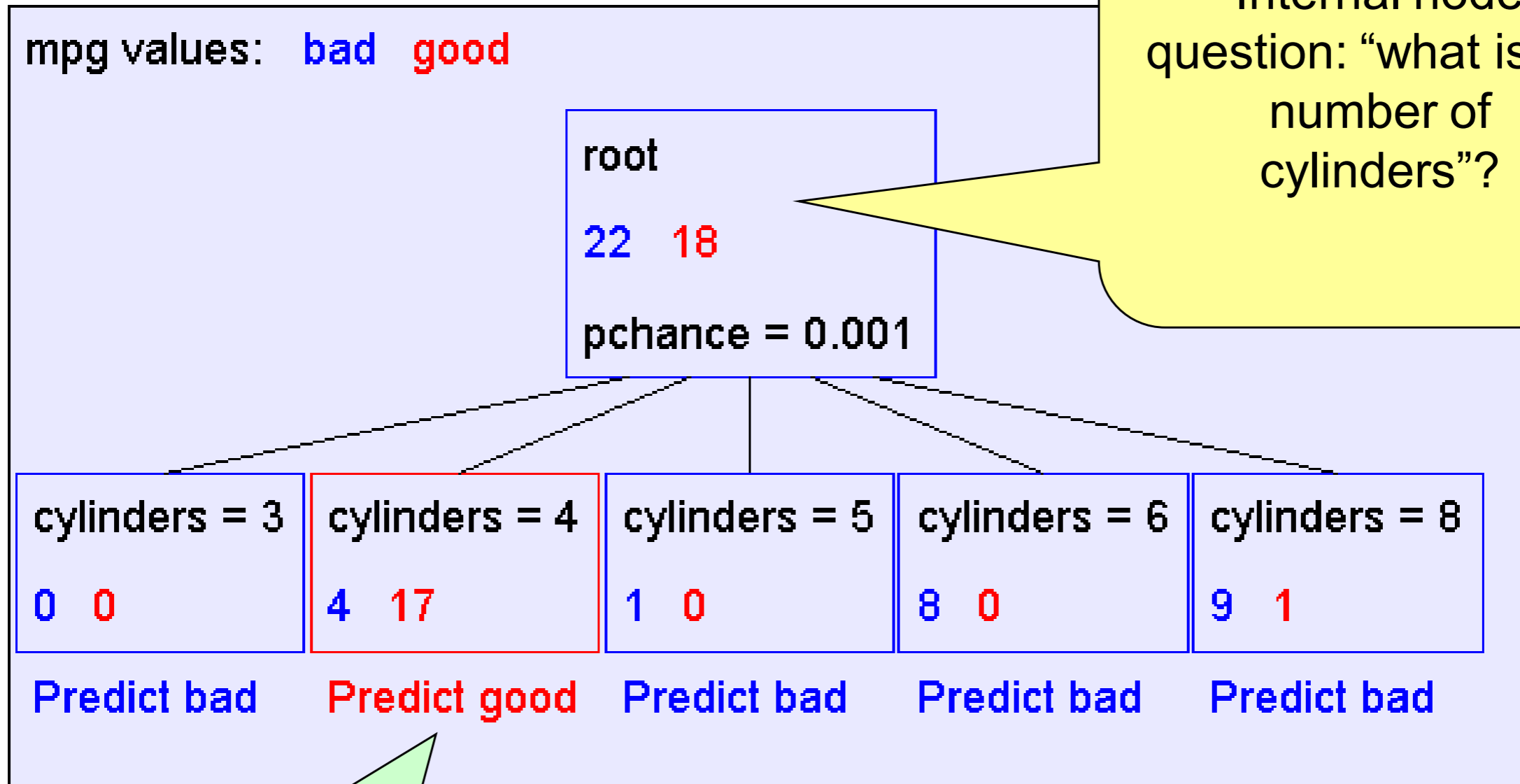
Leaves: classify by majority vote

18

# Key idea: Greedily learn trees using **recursion**
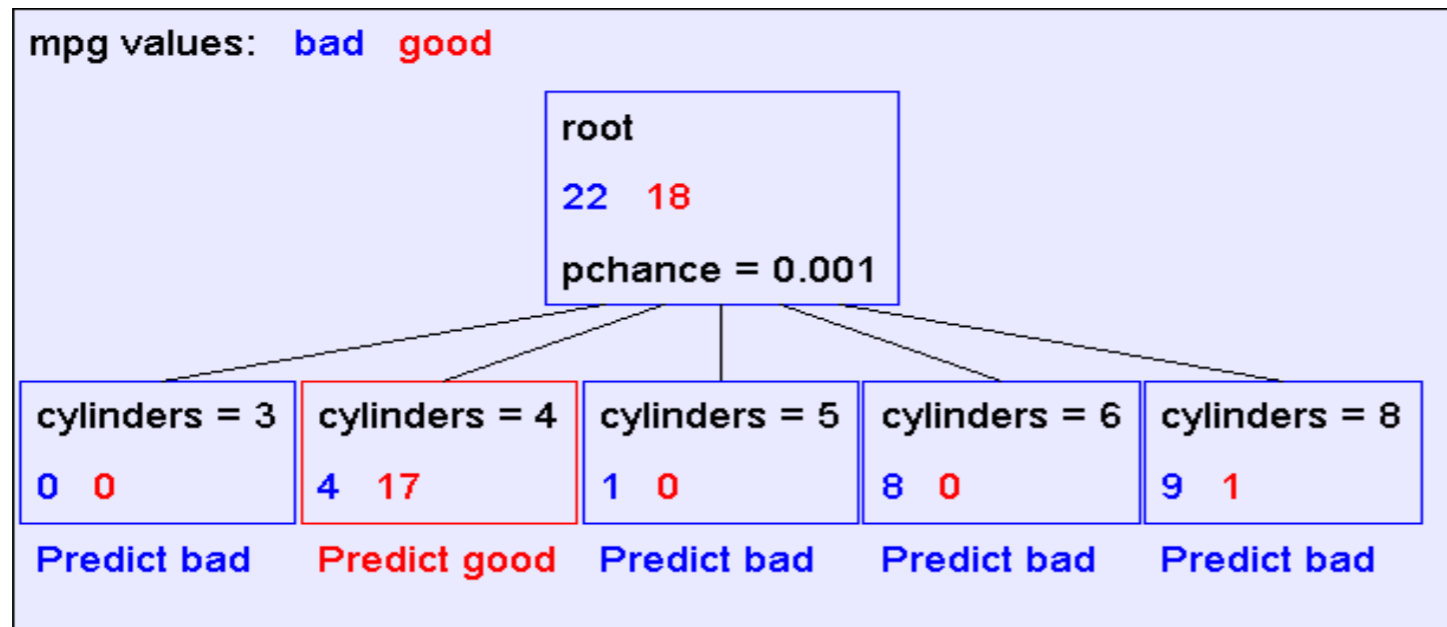
mpg values:  bad  good

root
22   18
pchance = 0.001

| cylinders = 3 | cylinders = 4 | cylinders = 5 | cylinders = 6 | cylinders = 8 |
| 0  0 | 4  17 | 1  0 | 8  0 | 9  1 |
| Predict bad | Predict good | Predict bad | Predict bad | Predict bad |

Take the Original Dataset..

And partition it according to the value of the attribute we split on

Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

Records in which cylinders = 8

19

# Recursive Step



mpg values:  bad  good

root
22  18
pchance = 0.001

| cylinders = 3 | cylinders = 4 | cylinders = 5 | cylinders = 6 | cylinders = 8 |
| 0  0 | 4  17 | 1  0 | 8  0 | 9  1 |
| Predict bad | Predict good | Predict bad | Predict bad | Predict bad |

Build tree from These records..

Build tree from These records..
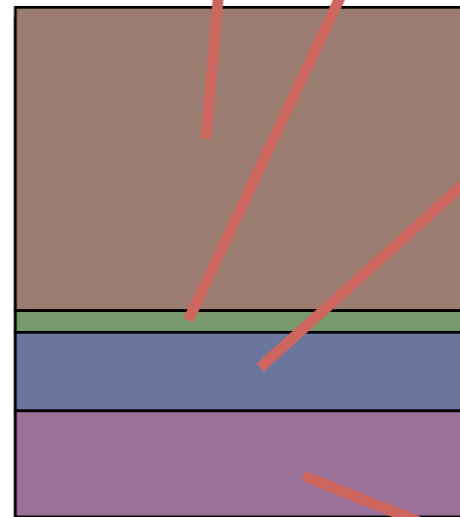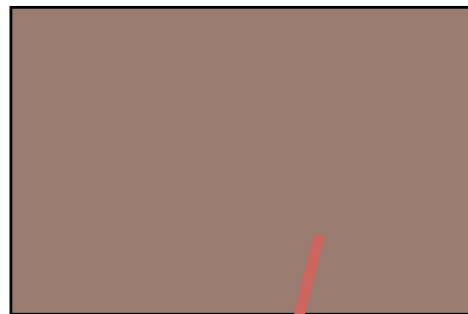
Build tree from These records..

Build tree from These records..

Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

Records in which cylinders = 8

20

# Second level of tree
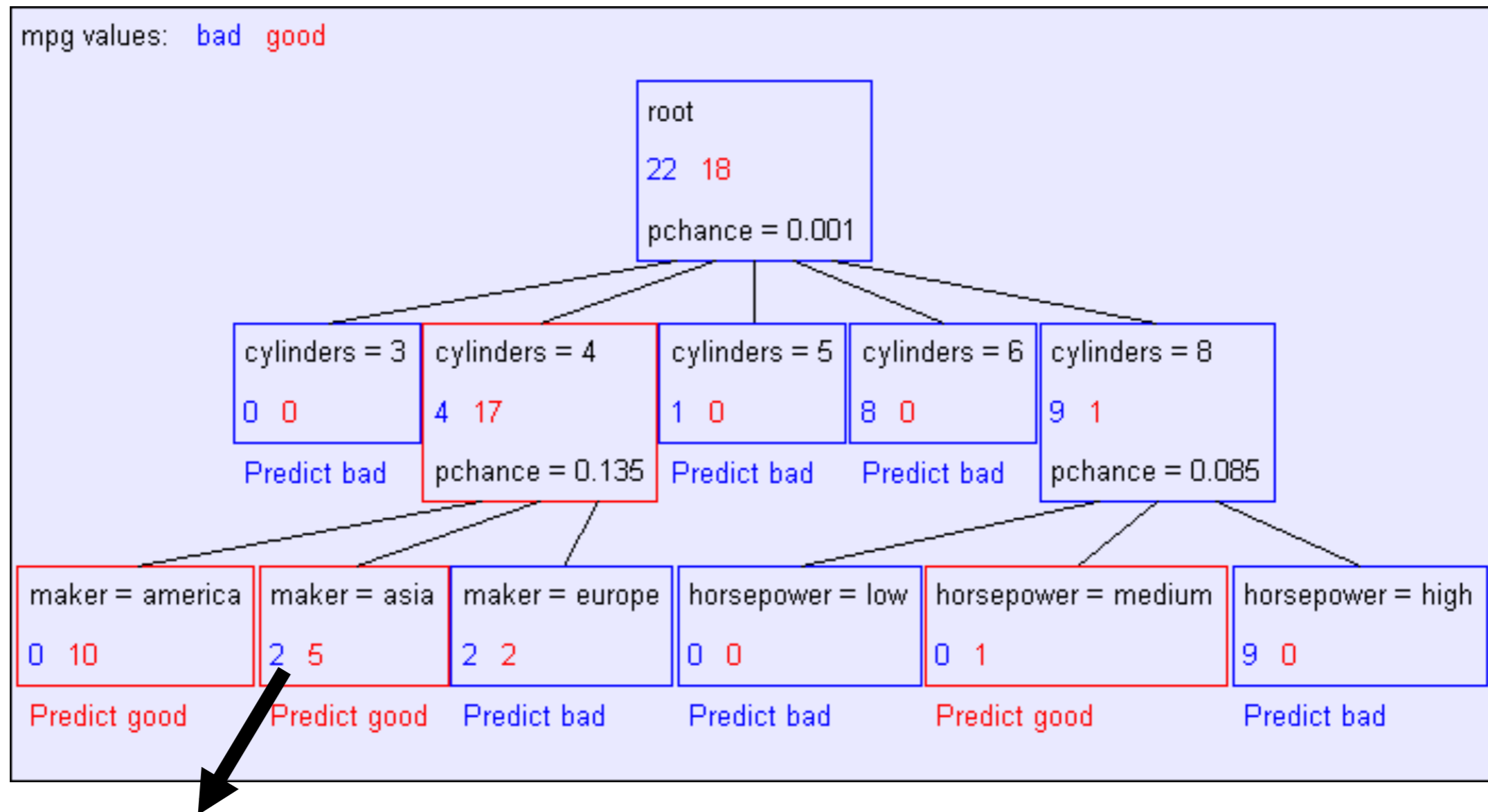


Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

21

The full decision tree
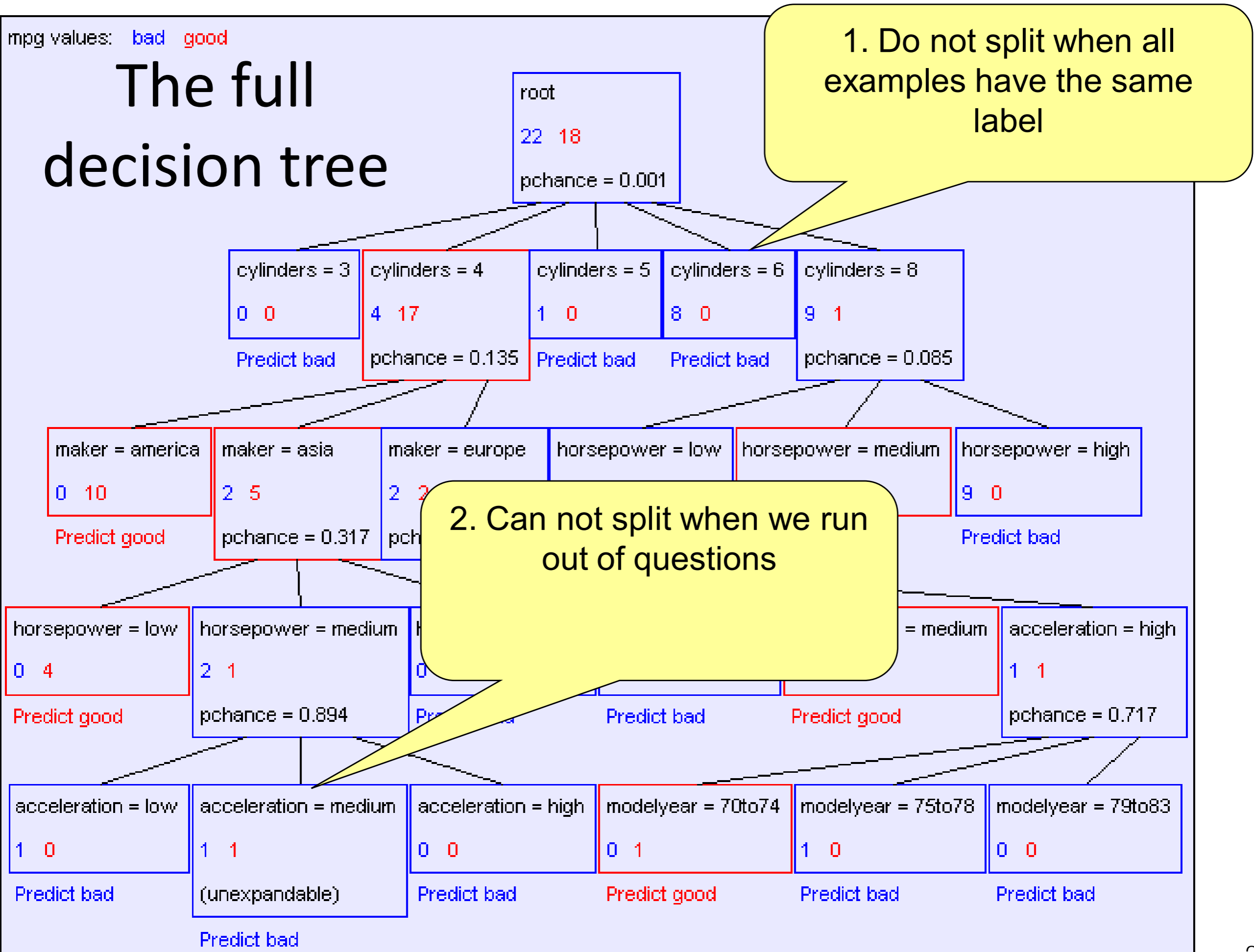
# Splitting: Choosing a good attribute

- Would we prefer to split on $X_1$ or $X_2$?

$X_1$
t    f

Y=t : 4    Y=t : 1
Y=f : 0    Y=f : 3

$X_2$
t    f

Y=t : 3    Y=t : 2
Y=f : 1    Y=f : 2
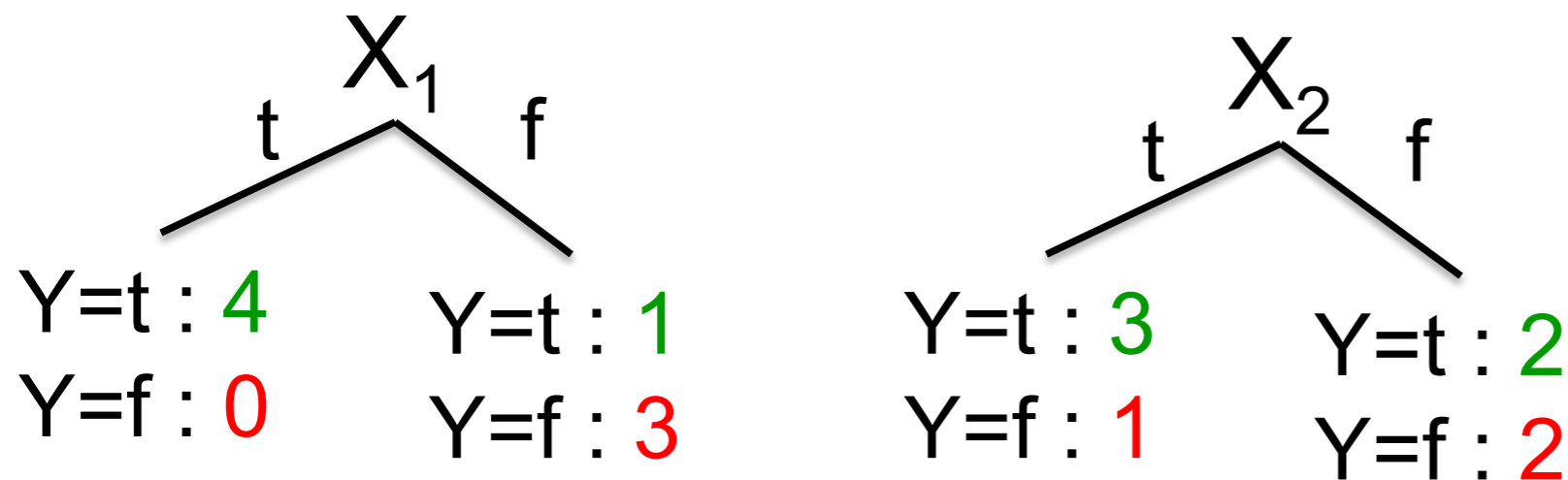
**Idea:** use counts at leaves to define probability distributions, so we can measure uncertainty!

| $X_1$ | $X_2$ | Y |
|-------|-------|---|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |
| F | T | F |
| F | F | F |

# Measuring uncertainty

- Good split if we are more certain about classification after split
  - Deterministic good (all true or all false)
  - Uniform distribution bad
  - What about distributions in between?

| P(Y=A) = 1/2 | P(Y=B) = 1/4 | P(Y=C) = 1/8 | P(Y=D) = 1/8 |
|---|---|---|---|

| P(Y=A) = 1/4 | P(Y=B) = 1/4 | P(Y=C) = 1/4 | P(Y=D) = 1/4 |
|---|---|---|---|

# Entropy

- Entropy $H(Y)$ of

$$H(Y) =$$

- **More uncertai**

- Information The
$H(Y)$ is the exp
bits needed to
drawn value of
efficient code)

Entropy

Entropy of a coin flip

1.0

1.0

Entropy(S)

Entropy

0.5

0.5

0.0                    0.5                    1.0

0.

Probability of heads

$p_\oplus$
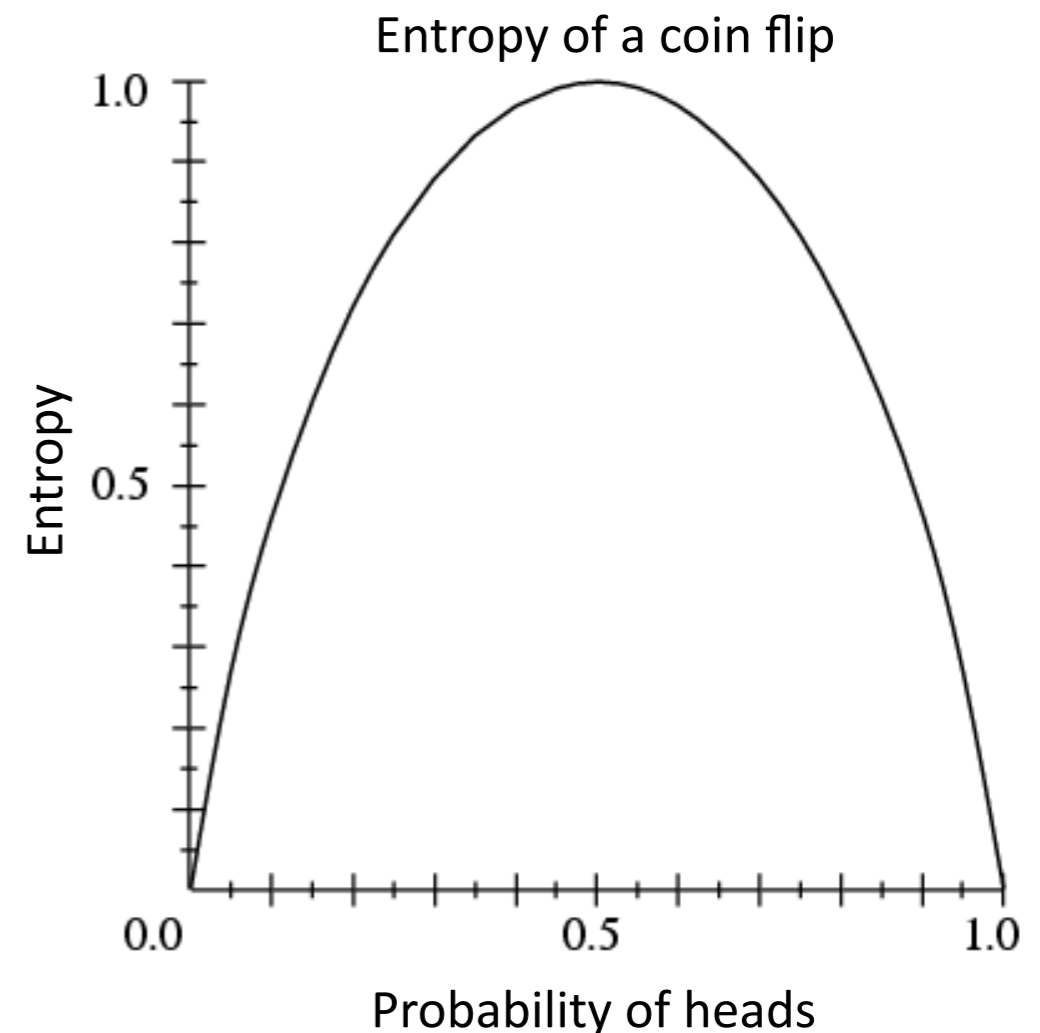
# High, Low Entropy

- **"High Entropy"**
  - Y is from a uniform like distribution
  - Flat histogram
  - Values sampled from it are less predictable

- **"Low Entropy"**
  - Y is from a varied (peaks and valleys) distribution
  - Histogram has many lows and highs
  - Values sampled from it are more predictable

# Entropy Exar

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

### Entropy of a coin flip

Entropy

Probability of heads

P(Y=t) = 5/6

P(Y=f) = 1/6

- $S$ is a sa
- $p_\oplus$ is the
- $p_\ominus$ is the
- Entropy

$En$

H(Y) = - 5/6 log$_2$ 5/6 - 1/6 l

= 0.65

| $X_1$ | $X_2$ | $Y$ |
|-------|-------|-----|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

File: ch3.ps
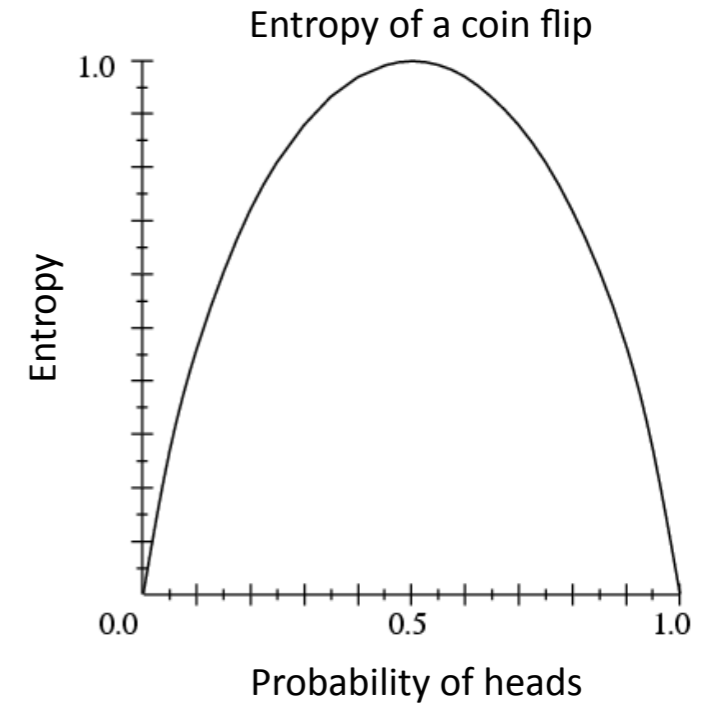
# Conditional Entropy

Conditional Entropy $H(Y|X)$ of a random variable $Y$ conditioned on a random variable $X$

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

Example:

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$

$X_1$

t            f

Y=t : 4       Y=t : 1

Y=f : 0       Y=f : 1
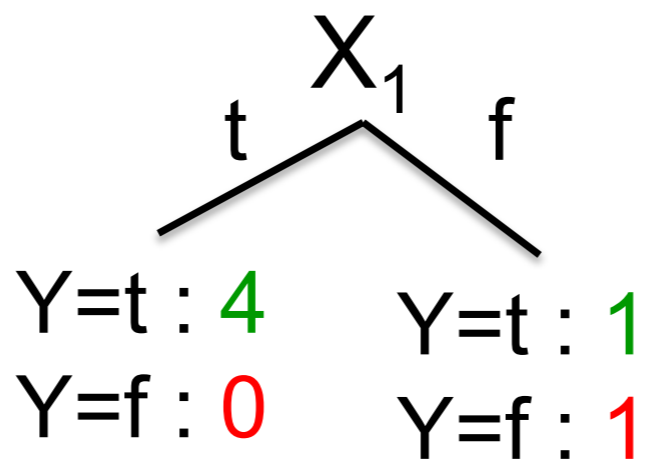
$H(Y|X_1) = -$ 4/6 (1 $\log_2$ 1 + 0 $\log_2$ 0)

$-$ 2/6 (1/2 $\log_2$ 1/2 + 1/2 $\log_2$ 1/2)

= 2/6

| $X_1$ | $X_2$ | Y |
|---|---|---|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

# Information gain

- Decrease in entropy (uncertainty) after splitting

$$IG(X) = H(Y) - H(Y \mid X)$$

In our running example:

IG($X_1$) = H(Y) – H(Y|$X_1$)

$\qquad$ = 0.65 – 0.33

IG($X_1$) > 0 → we prefer the split!

| $X_1$ | $X_2$ | Y |
|---|---|---|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

29

# Learning decision trees

- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute:

$$\arg \max_i IG(X_i) = \arg \max_i H(Y) - H(Y \mid X_i)$$

- Recurse

# When to stop?



mpg values: bad good

root
22 18
pchance = 0.001

cylinders = 3 | cylinders = 4 | cylinders = 5 | cylinders = 6 | cylinders = 8
0 0 | 4 17 | 1 0 | 8 0 | 9 1
Predict bad | Predict good | Predict bad | Predict bad | Predict bad
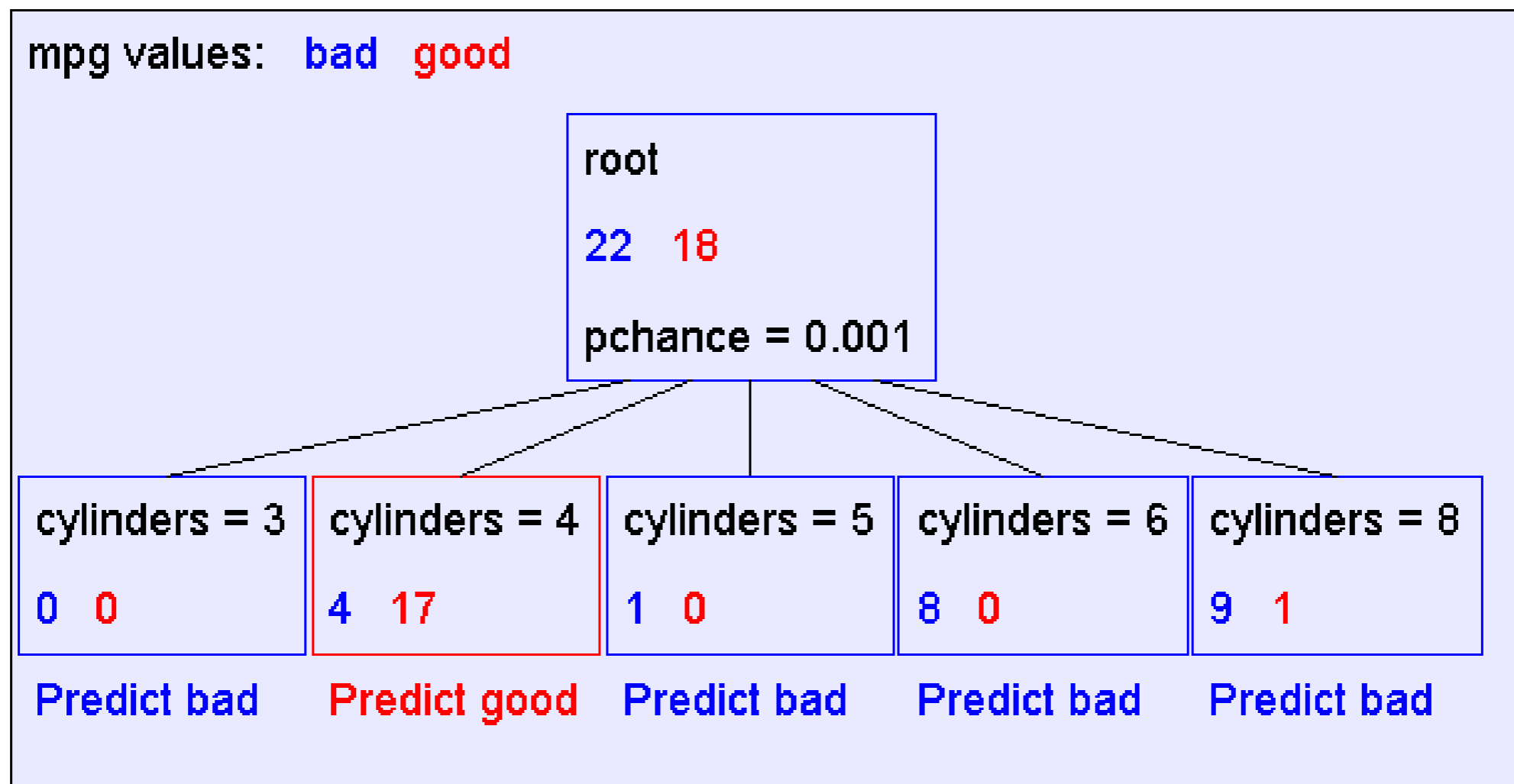
- First split looks good! But, when do we stop?

Base Case One

32
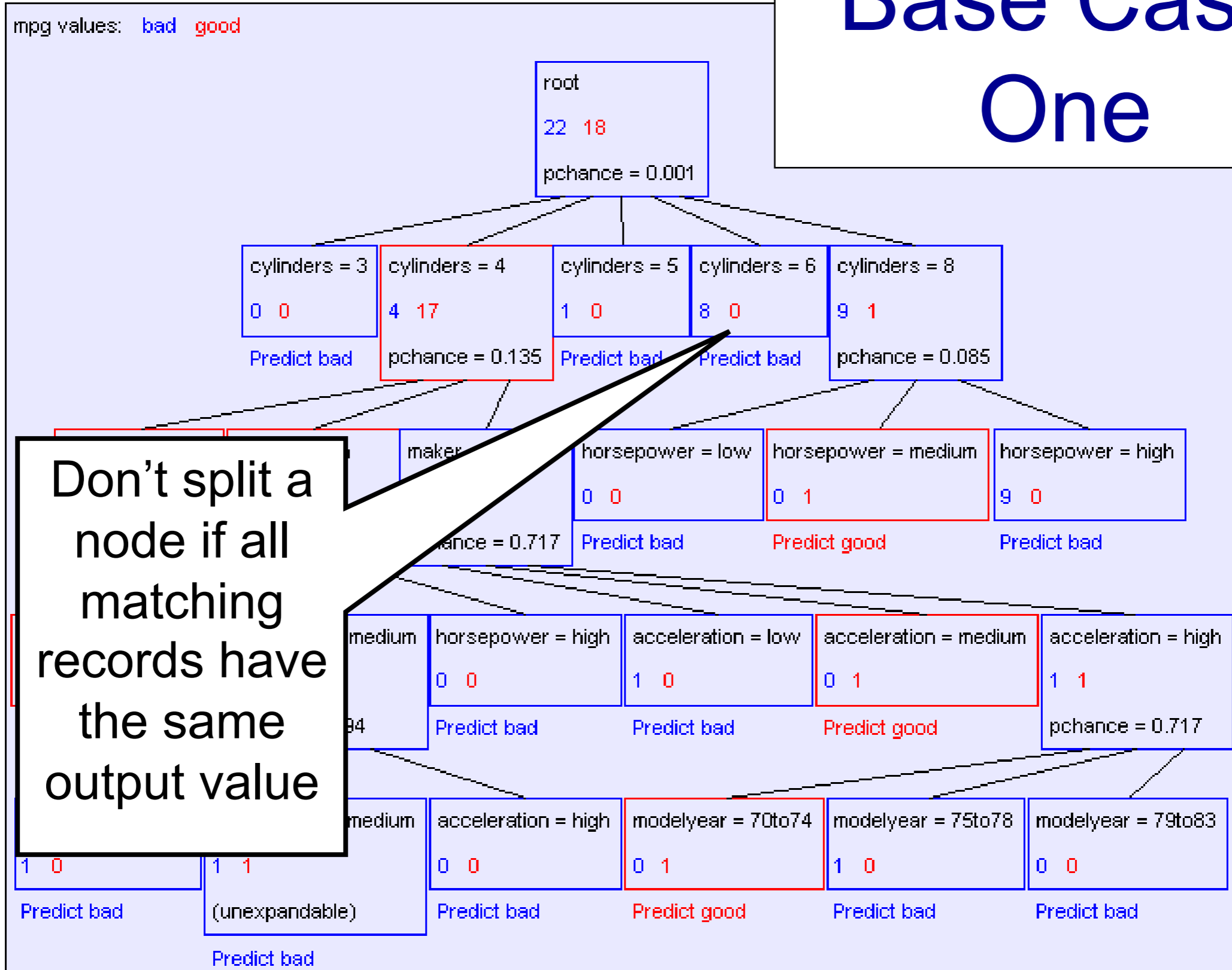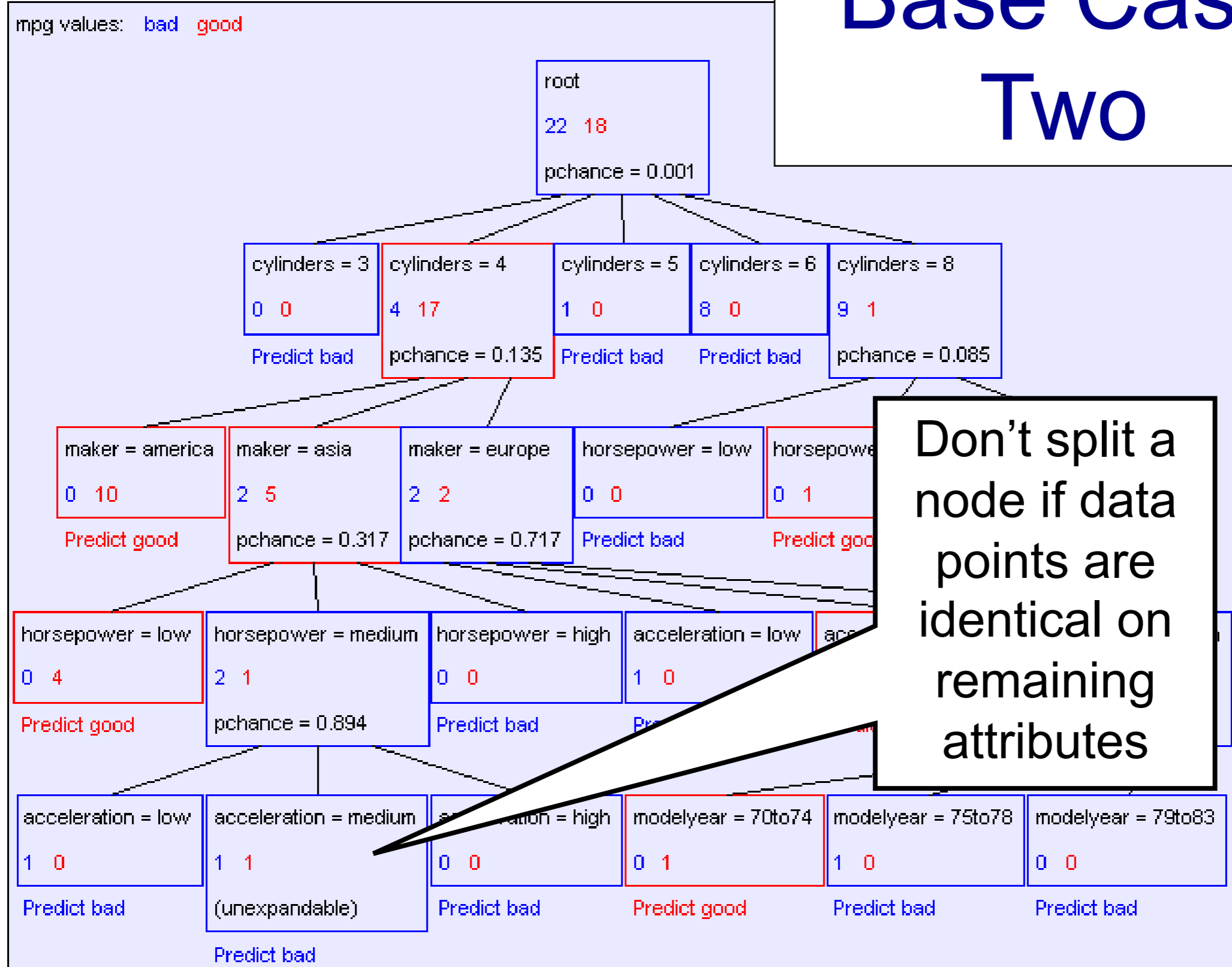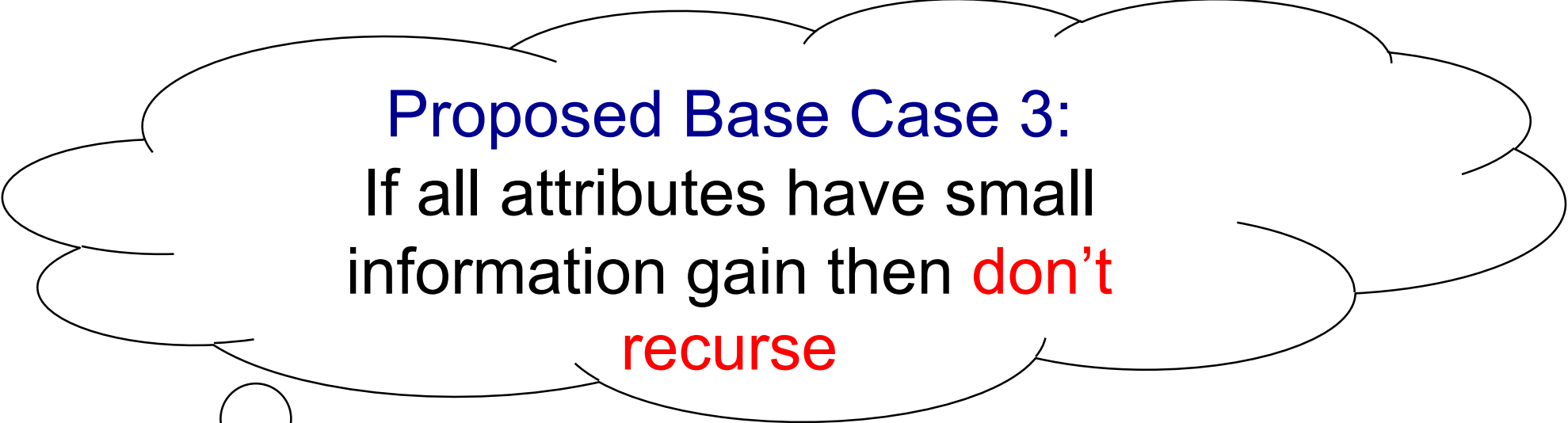
Base Case Two

# Base Cases: An idea

- **Base Case One:** If all records in current data subset have the same output then don't recurse
- **Base Case Two:** If all records have exactly the same set of input attributes then don't recurse

Proposed Base Case 3:
If all attributes have small information gain then don't recurse

- *This is not a good idea*

# The problem with proposed case 3

y = a XOR b

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

The information gains:
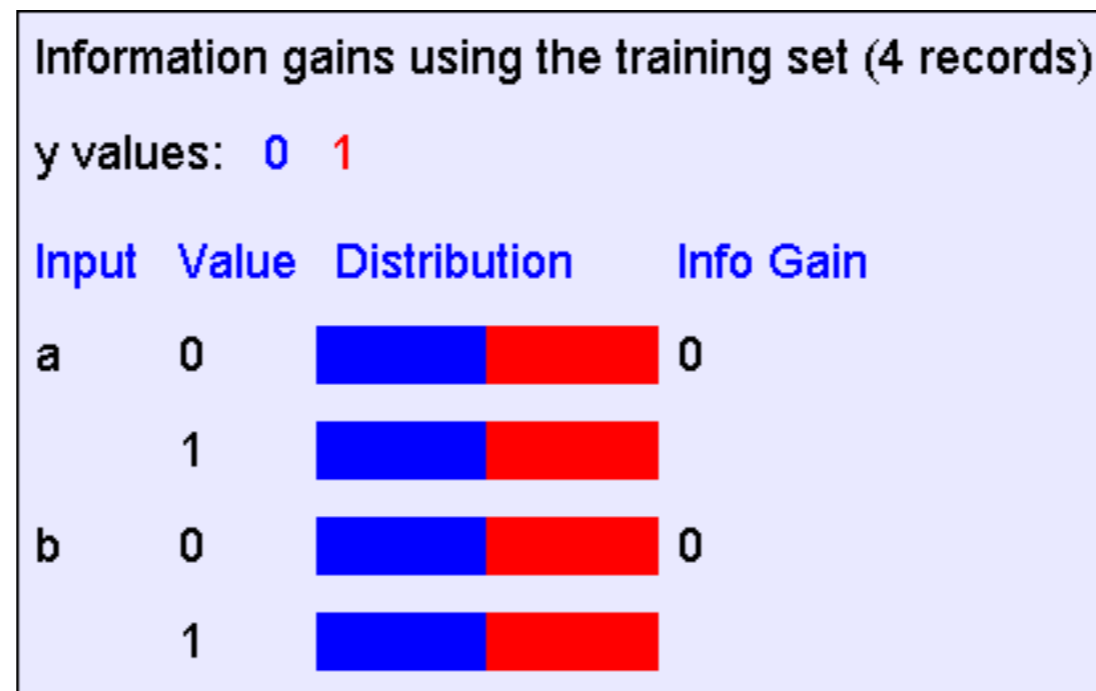
# If we omit proposed case 3:

The resulting decision tree:

y = a XOR b

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Instead, perform **pruning** after building a tree



y values:  0  1

root

2  2

pchance = 1.000

a = 0

1  1

pchance = 0.414

a = 1

1  1

pchance = 0.414

b = 0

1  0

Predict 0

b = 1

0  1

Predict 1

b = 0

0  1

Predict 1

b = 1

1  0

Predict 0

# Decision trees will overfit

37
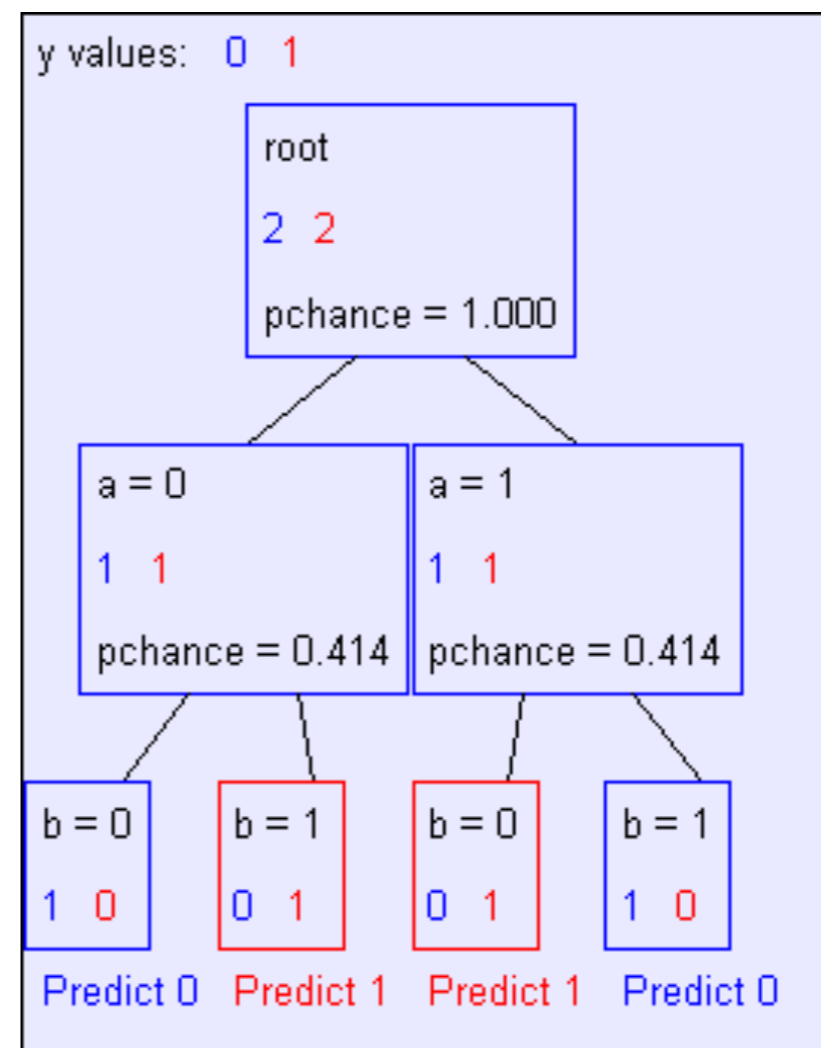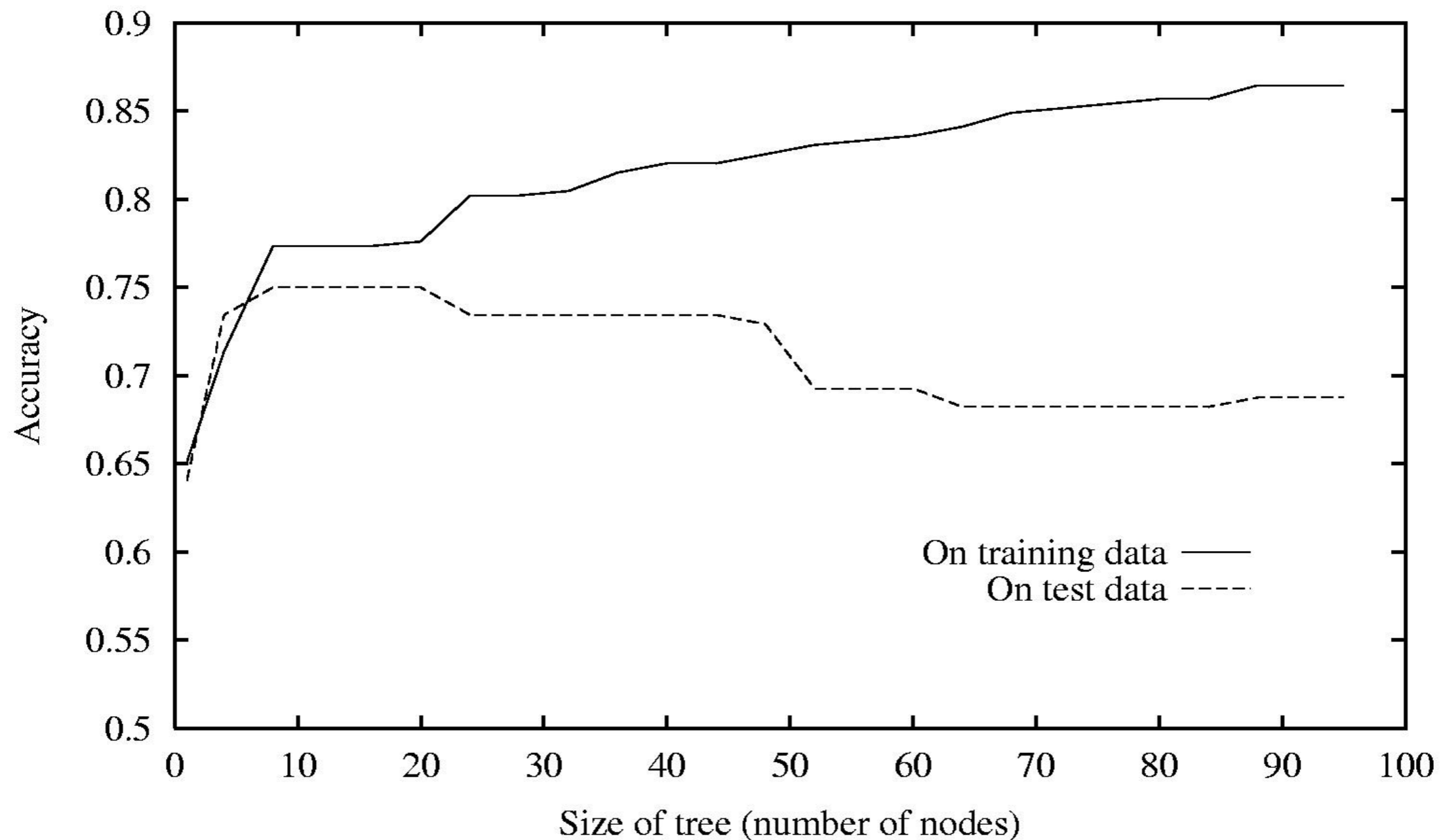
# Decision trees will overfit

- **Standard decision trees have no learning bias**
  - Training set error is always zero!
    - (If there is no label noise)
  - Lots of variance
  - Must introduce some bias towards simpler trees

- **Many strategies for picking simpler trees**
  - Fixed depth
  - Fixed number of leaves

- **Random forests**

# Real-valued inputs

- What should we do if some of the inputs are real-valued?

<span style="color:red">Infinite number of possible split values!!!</span>

| mpg | cylinders | displacemen | horsepower | weight | acceleration | modelyear | maker |
|-----|-----------|-------------|------------|--------|--------------|-----------|-------|
| | | | | | | | |
| good | 4 | 97 | 75 | 2265 | 18.2 | 77 | asia |
| bad | 6 | 199 | 90 | 2648 | 15 | 70 | america |
| bad | 4 | 121 | 110 | 2600 | 12.8 | 77 | europe |
| bad | 8 | 350 | 175 | 4100 | 13 | 73 | america |
| bad | 6 | 198 | 95 | 3102 | 16.5 | 74 | america |
| bad | 4 | 108 | 94 | 2379 | 16.5 | 73 | asia |
| bad | 4 | 113 | 95 | 2228 | 14 | 71 | asia |
| bad | 8 | 302 | 139 | 3570 | 12.8 | 78 | america |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| good | 4 | 120 | 79 | 2625 | 18.6 | 82 | america |
| bad | 8 | 455 | 225 | 4425 | 10 | 70 | america |
| good | 4 | 107 | 86 | 2464 | 15.5 | 76 | europe |
| bad | 5 | 131 | 103 | 2830 | 15.9 | 78 | europe |
| | | | | | | | |

# "One branch for each numeric value" idea:
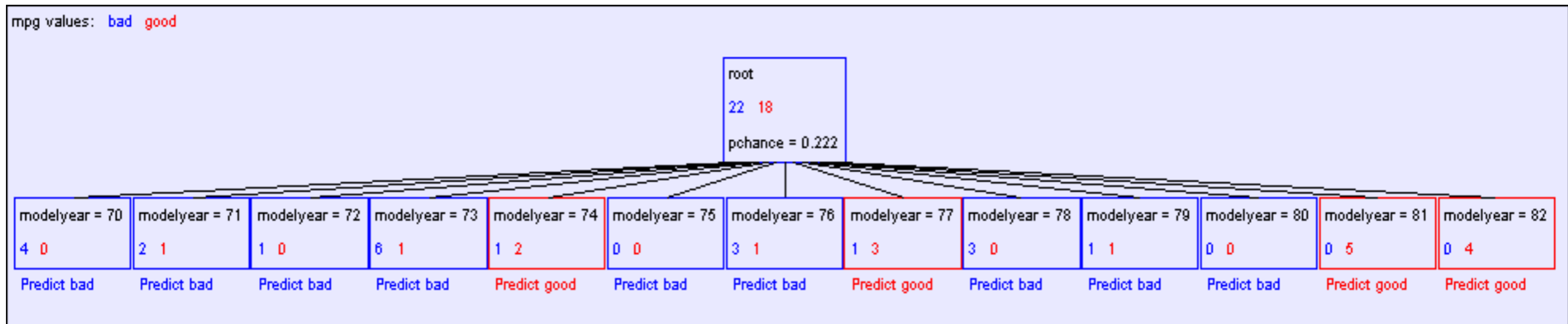


**Hopeless:** hypothesis with such a high branching factor will shatter **any** dataset and overfit

# Threshold splits

- Binary tree: split on attribute X at value t
  - One branch: $X < t$
  - Other branch: $X \geq t$

- Requires small change
  - Allow repeated splits on same variable **along a path**

41
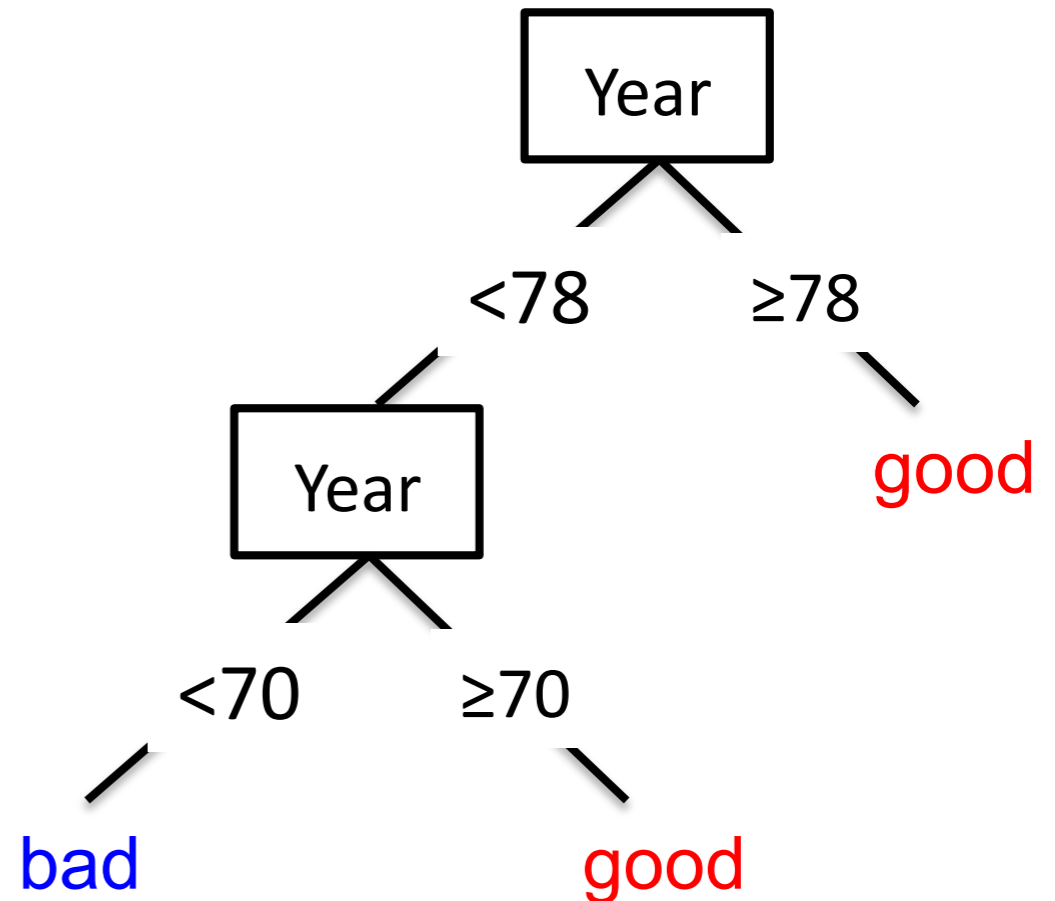
# The set of possible thresholds

- Binary tree, split on attribute $X$

  - One branch: $X < t$

  - Other branch: $X \geq t$

- Search through possible values of $t$

  - Seems hard!!!

- But only a finite number of $t$'s are important.

Infinitely many possible split points $c$ to define node test $X_j > c$ ?

$X_j\ X$

No! Moving split point along the empty space between two observed values

has no effect on information gain or empirical loss; so just use midpoint

$t_1 \quad t_2$
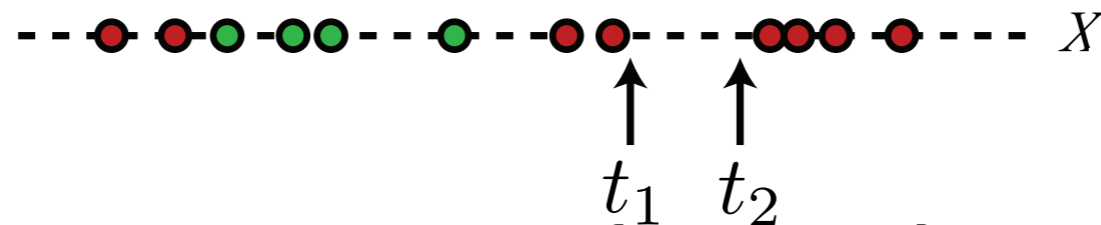
- Sort data according to $X$ into $\{x_1, \ldots, x_m\}$

  $X_j\ X_j$

- Consider split points of the form $x_i + (x_{i+1} - x_i)/2$

  $c_1 \quad c_2$

- Moreover, only splits between examples from different classes matter!

  Moreover, only splits between examples from different classes

  can be optimal for information gain or empirical loss reduction

  $X_j$

  $c_1 \qquad c_2$

  $X_j$

  $X$

$t_1 \qquad t_2$
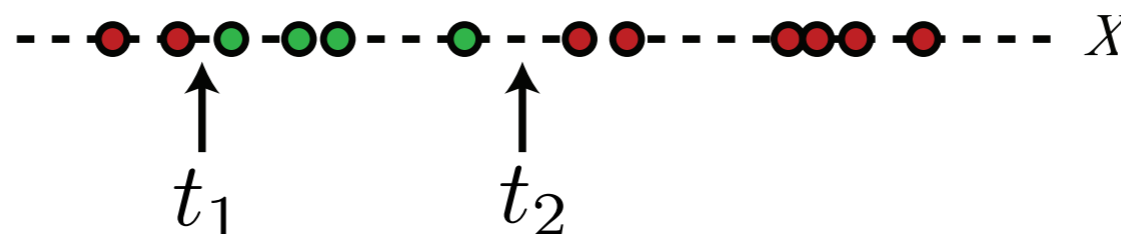
# Picking the best threshold

- Suppose $X$ is real valued with threshold $t$

- Want $IG(Y|X{:}t)$, the information gain for $Y$ when testing if $X$ is greater than or less than $t$

- Define:
  - $H(Y|X{:}t) = p(X<t)H(Y|X<t)+p(X>=t)H(Y|X>=t)$
  - $IG(Y|X{:}t) = H(Y) - H(Y|X{:}t)$
  - $IG^*(Y|X) = \max_t IG(Y|X{:}t)$

- Use: $IG^*(Y|X)$ for continuous variables
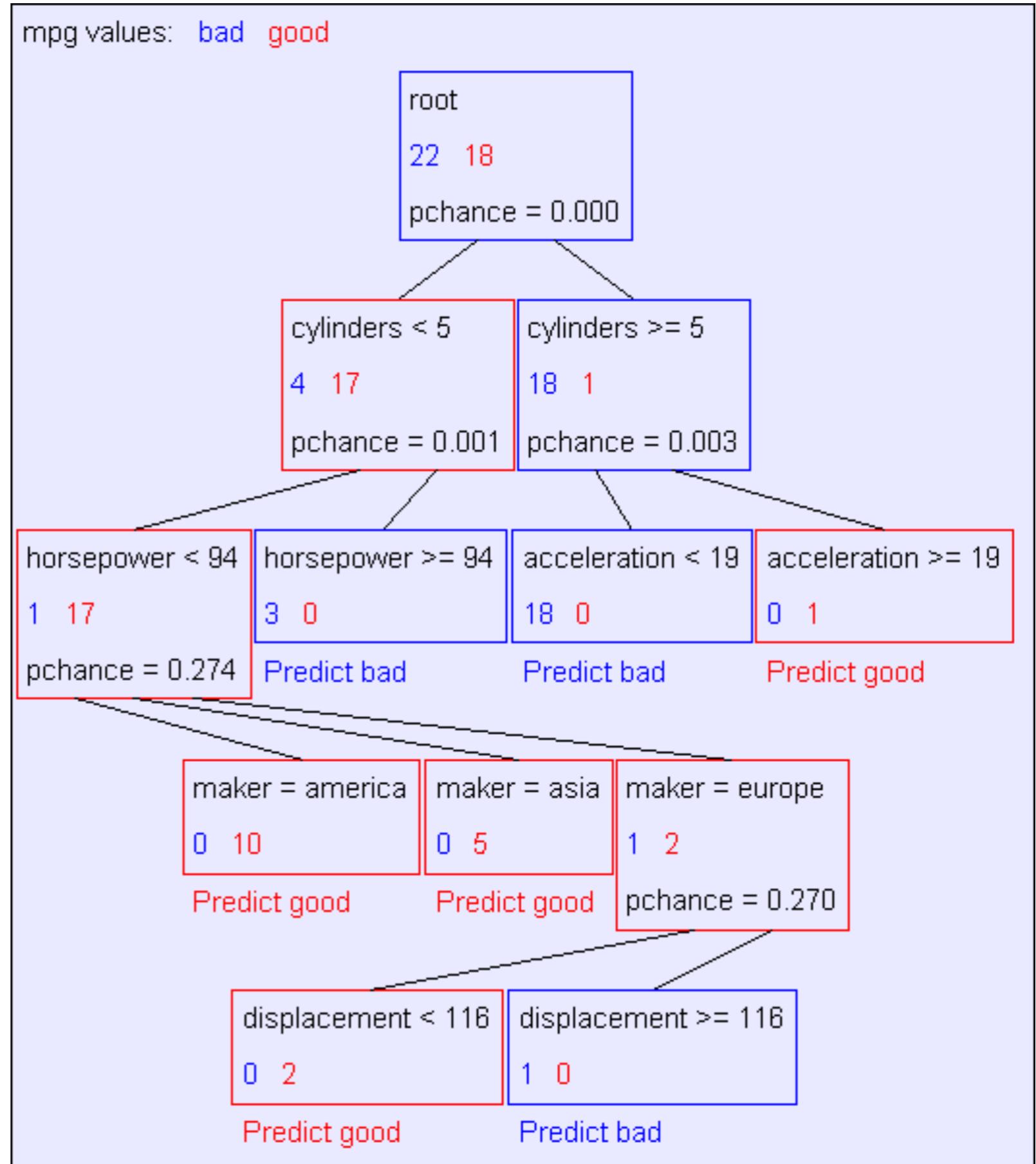
# Example with MPG

44

# Example tree for our continuous dataset

# Demo time…

# What you need to know about decision trees

- Decision trees are one of the most popular ML tools
  - Easy to understand, implement, and use
  - Computationally cheap (to solve heuristically)

- Information gain to select attributes (ID3, C4.5,...)

- Presented for classification, can be used for regression and density estimation too

- Decision trees will overfit!!!
  - Must use tricks to find "simple trees", e.g.,
    - Fixed depth/Early stopping
    - Pruning
  - Or, use ensembles of different trees (random forests)

# Decision Trees vs SVM

| Characteristic | SVM | Trees |
|---|---|---|
| Natural handling of data of "mixed" type | ▼ | ▲ |
| Handling of missing values | ▼ | ▲ |
| Robustness to outliers in input space | ▼ | ▲ |
| Insensitive to monotone transformations of inputs | ▼ | ▲ |
| Computational scalability (large $N$) | ▼ | ▲ |
| Ability to deal with irrelevant inputs | ▼ | ▲ |
| Ability to extract linear combinations of features | ▲ | ▼ |
| Interpretability | ▼ | ◆ |
| Predictive power | ▲ | ▼ |

Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

slide by Vibhav Gogate

# **Next Lecture:**
# Ensemble Methods