VIRGINIA TECH

# Decision Trees

Lifu Huang

Computer Science, Virginia Tech

January 21/26, 2021

Slides adapted from Luke Zettlemoyer, David Sontag, Bert Huang

# A learning problem: predict fuel efficiency

- Fuel efficiency annotation (mpg) records
  - good
  - bad
- Each row denotes a record
- Each column except mpg denotes one attribute
- Goal: given the values of all attributes (X), predict MPG (Y)

$f: X \rightarrow Y$

| 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 |

$Y$ $X$

Data from UCI repository

# Hypothesis space

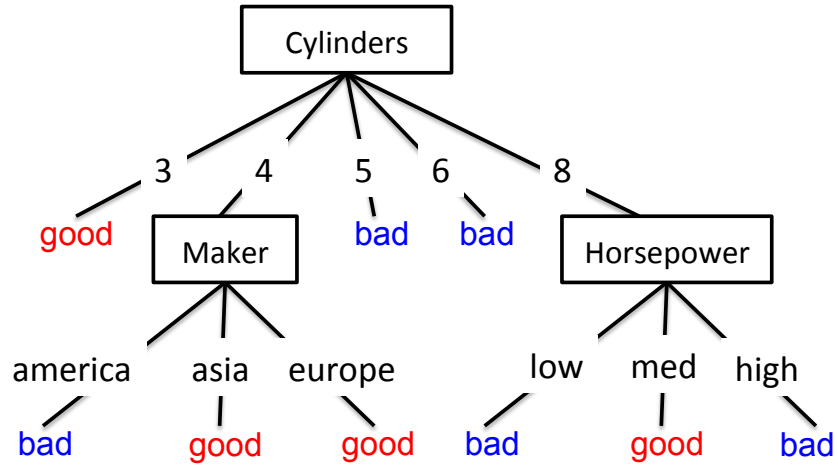
- Many possible hypothesizes
- Natural choice: conjunction of attribute constraints
- For each attribute:
  - Constraint to a specific value, e.g., maker=asia
  - Don’t care ?
- For example
  - maker=asia ∧ weight=low

| ***maker*** | ***cyl*** | ***displace*** | ***weight*** | ***accel ….*** |
|---|---|---|---|---|
| **asia** | **?** | **?** | **low** | **?** |

# 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, and output the labeled class $y$
- How many possible hypotheses (decision trees)?
- How can we choose the best one?

| 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 |



# What is the simplest tree?

- Just one node:  predict all records as mpg=bad
- Is this a good tree?

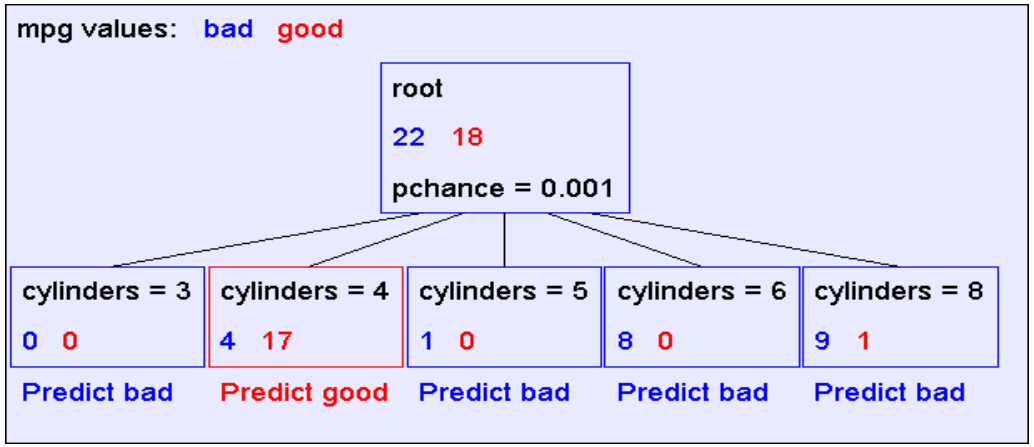
[18+, 22-]

correct on 22 examples

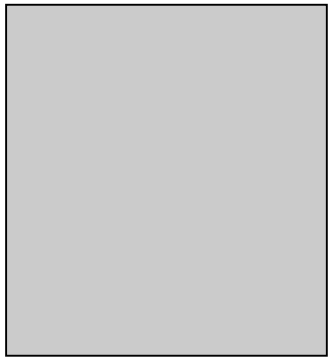
incorrect on 18 examples

| 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 |

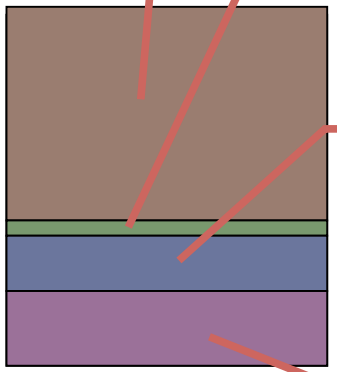
# Greedily learn trees using recursion




Take the Original Dataset..



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



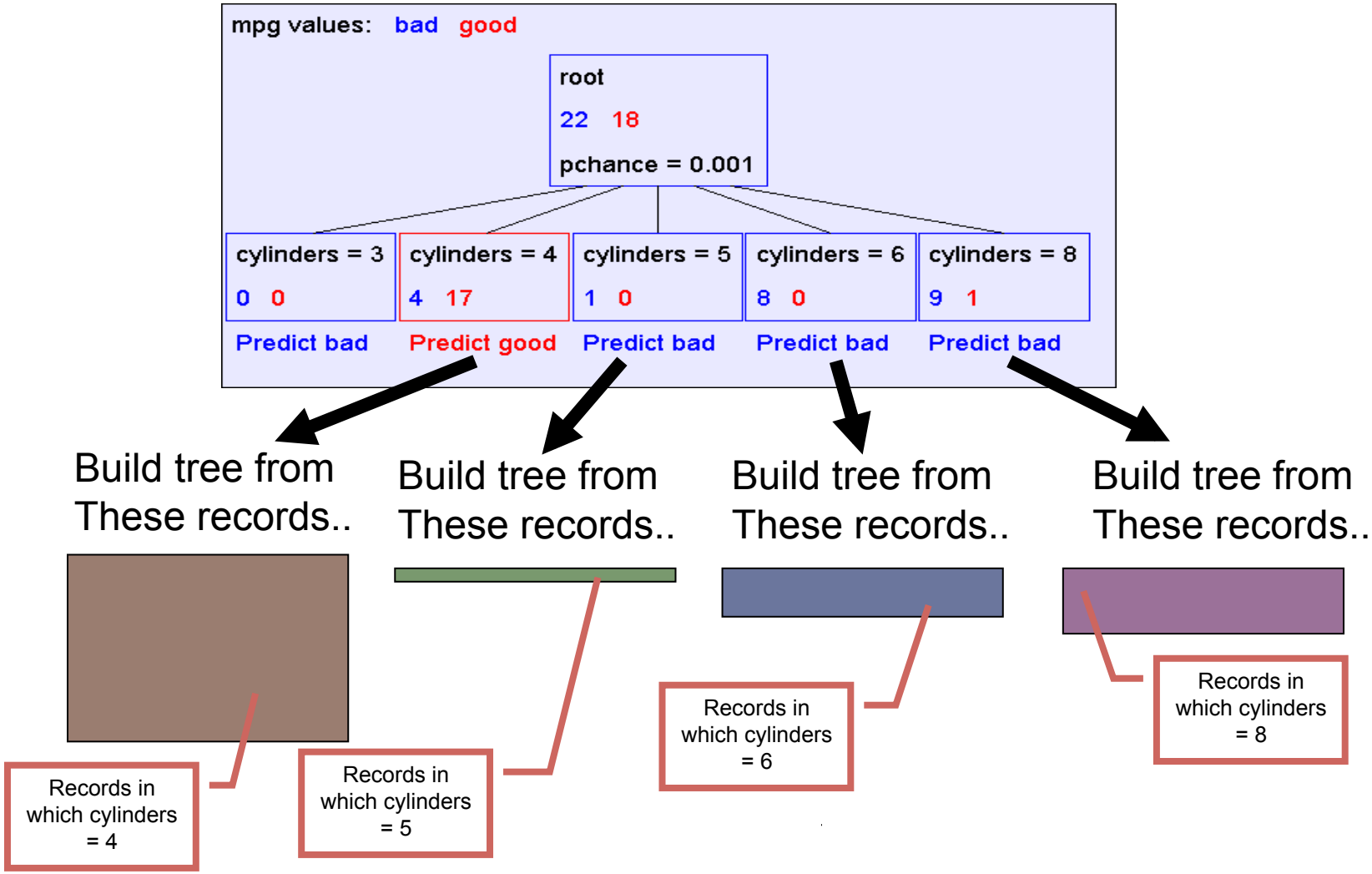
Records in which cylinders = 4

Records in which cylinders = 5

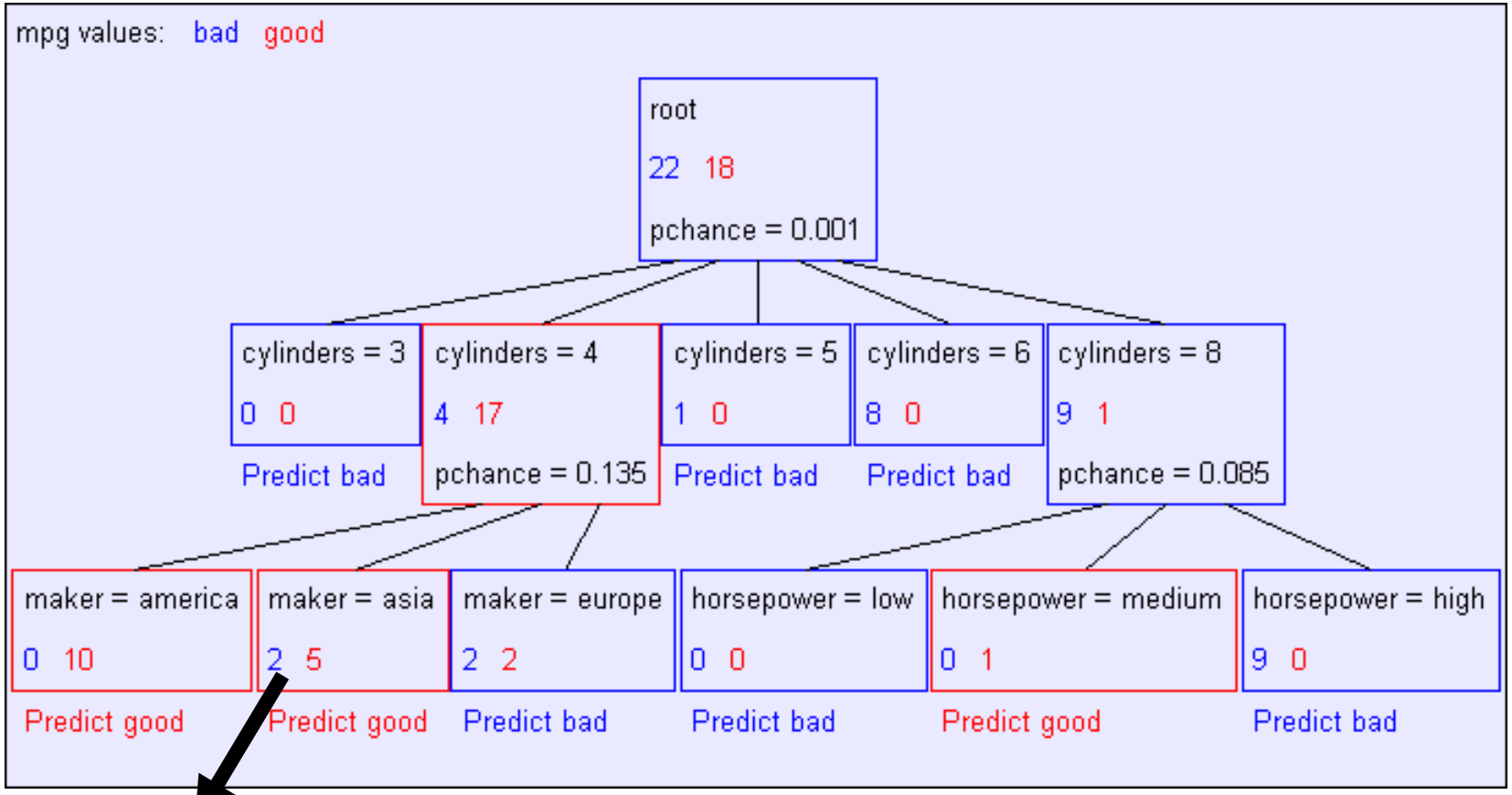
Records in which cylinders = 6

Records in which cylinders = 8

# Recursive Step



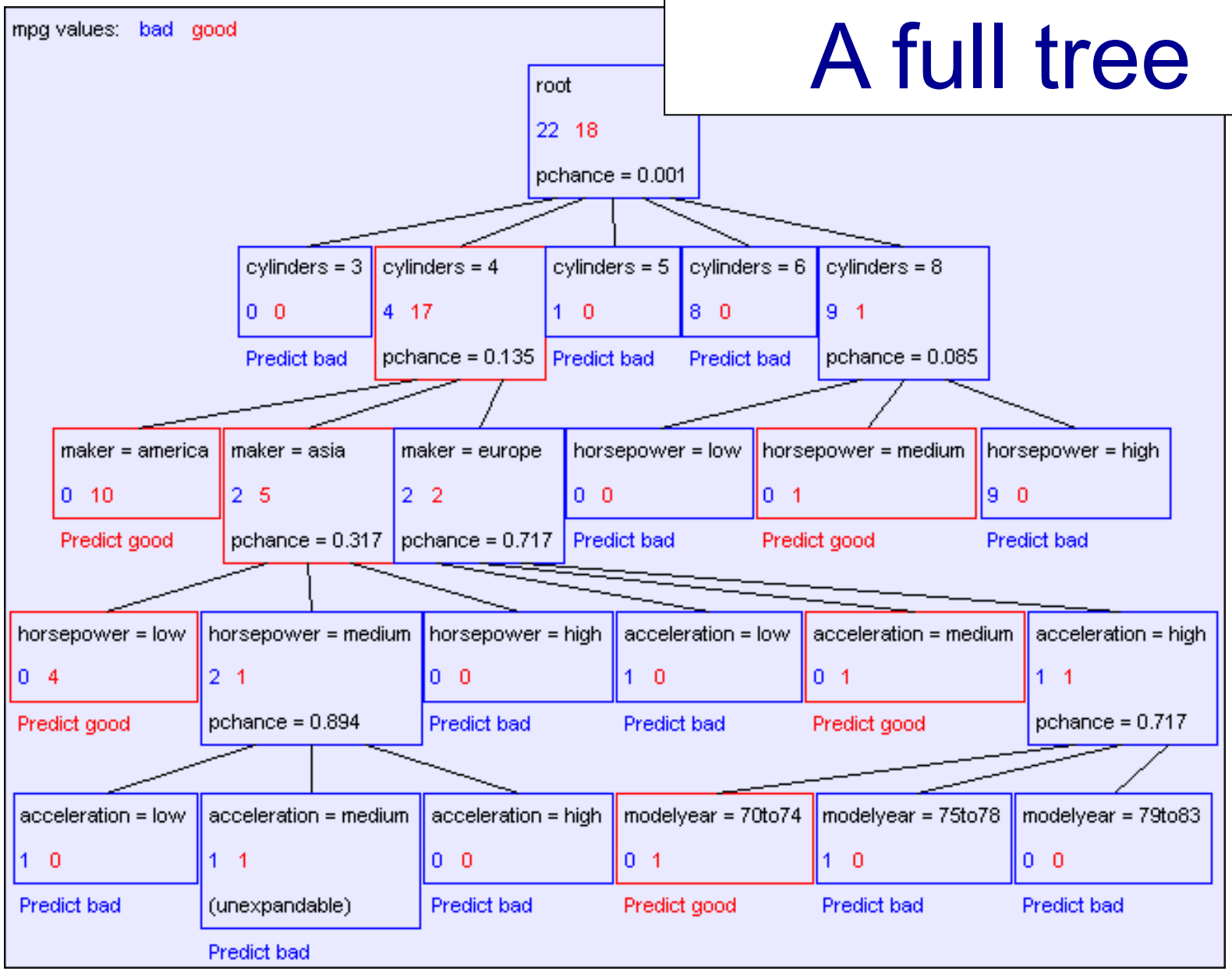
# Second level of tree




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

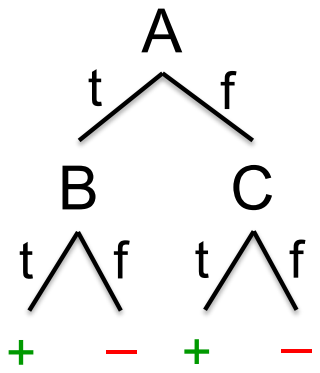
(Similar recursion in the other cases)

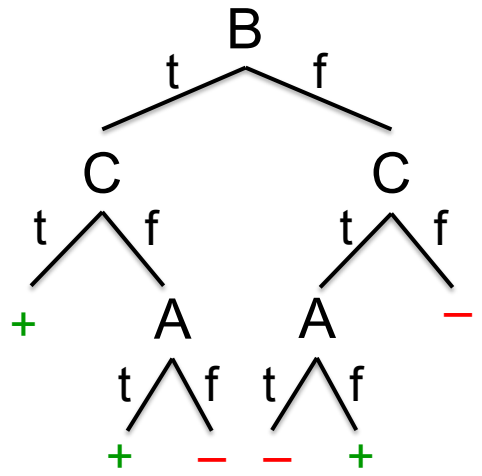
# A full tree


mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = medium
0 1
Predict good
horsepower = high
9 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
Predict bad
acceleration = low
1 0
Predict bad
acceleration = medium
0 1
Predict good
acceleration = high
1 1
pchance = 0.717
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
acceleration = high
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad

# Are all decision trees equal?

- Many trees can represent the same concept
- However, not all trees will have the same size!!
  - e.g., $\phi = (A \wedge B) \vee (\neg A \wedge C)$ -- ((A and B) or (not A and C))



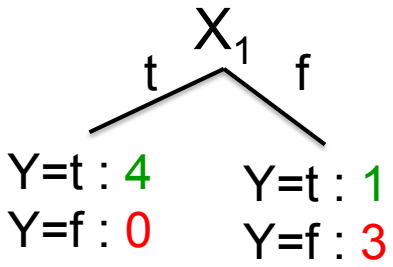



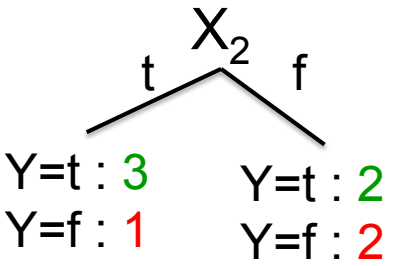

- Which tree do we prefer?
  - Smaller tree has more examples at each leaf!!

# Learning decision tree is hard!!

- Learning the 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

| $X_1$ | $X_2$ | 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 |




**Idea**: use counts at leaves to define probability distributions, so we can measure uncertainty.

# Measure 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 |

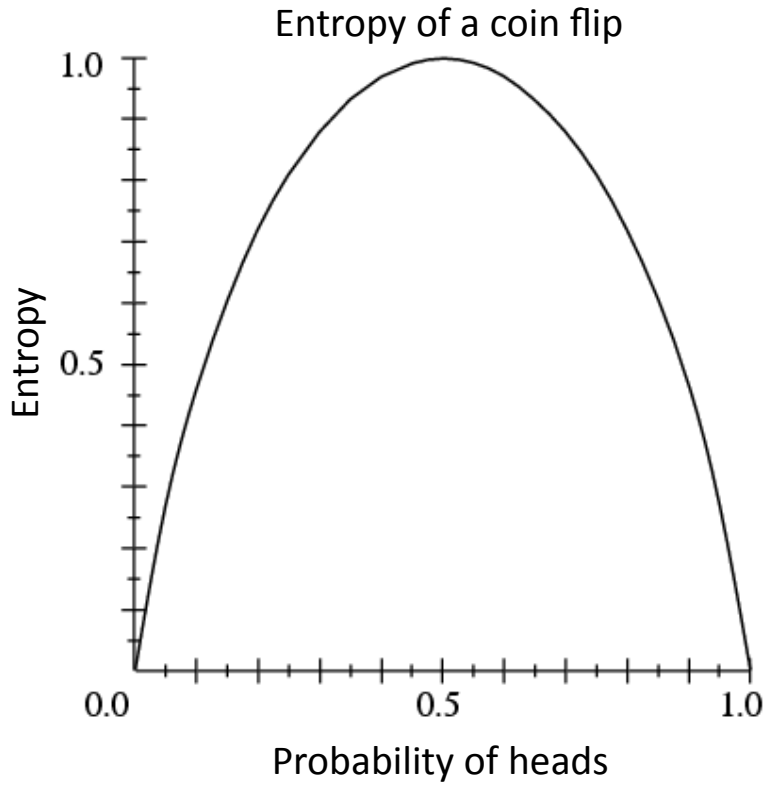
# Entropy

- Entropy $H(Y)$ of a random variable $Y$

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

- More uncertainty, more entropy!

Information Theory interpretation: $H(Y)$ is the expected number of bits needed to encode a randomly drawn value of $Y$ (under most efficient code)



# 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 Example

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

P(Y=t) = 5/6

P(Y=f) = 1/6

H(Y) = - 5/6 $\log_2$ 5/6 - 1/6 $\log_2$ 1/6

= 0.65

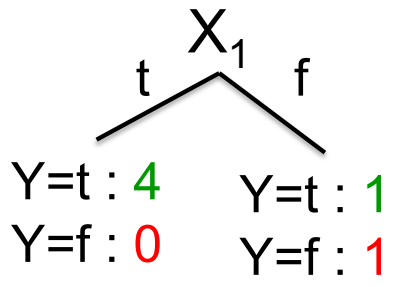
| $X_1$ | $X_2$ | Y |
|---|---|---|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

# 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:




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

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

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

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

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

$= 2/6$

# Information gain

- Decrease in entropy (uncertainty) after splitting

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

- In our running example:

$IG(X_1) = H(Y) - H(Y|X_1)$
$= 0.65 - 0.33$

$IG(X_1) > 0$ → we prefer the split!

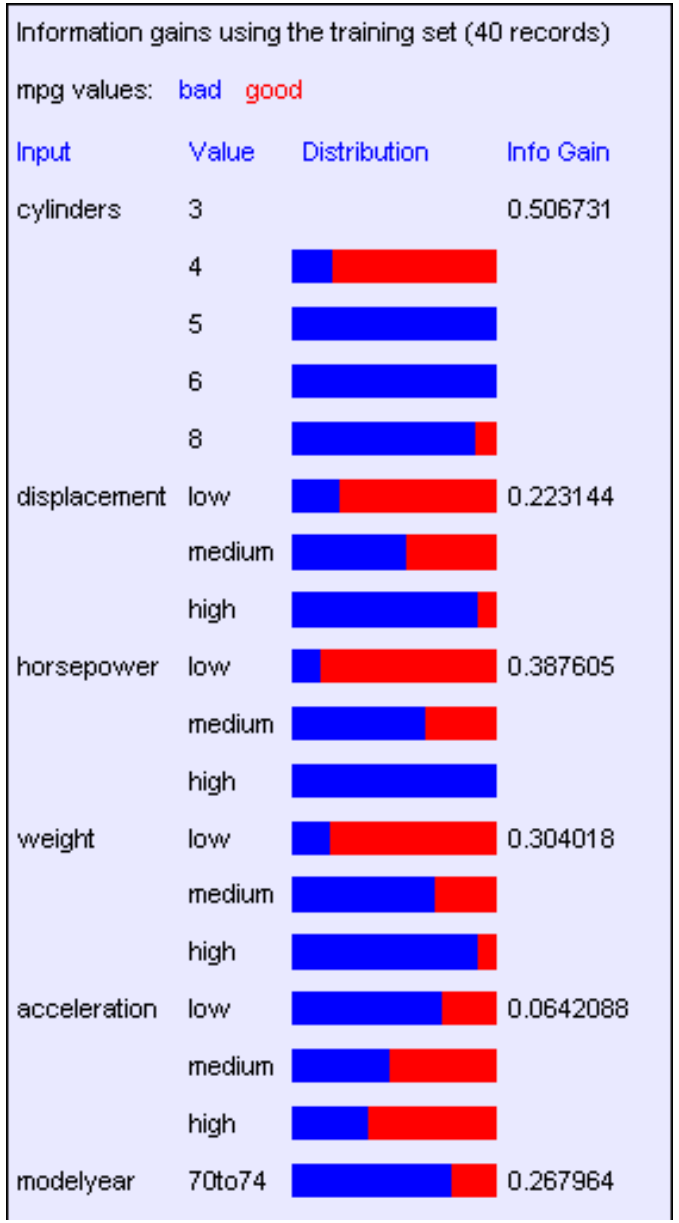
| $X_1$ | $X_2$ | Y |
|---|---|---|
| T | T | T |
| T | F | T |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

# Learning decision trees

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

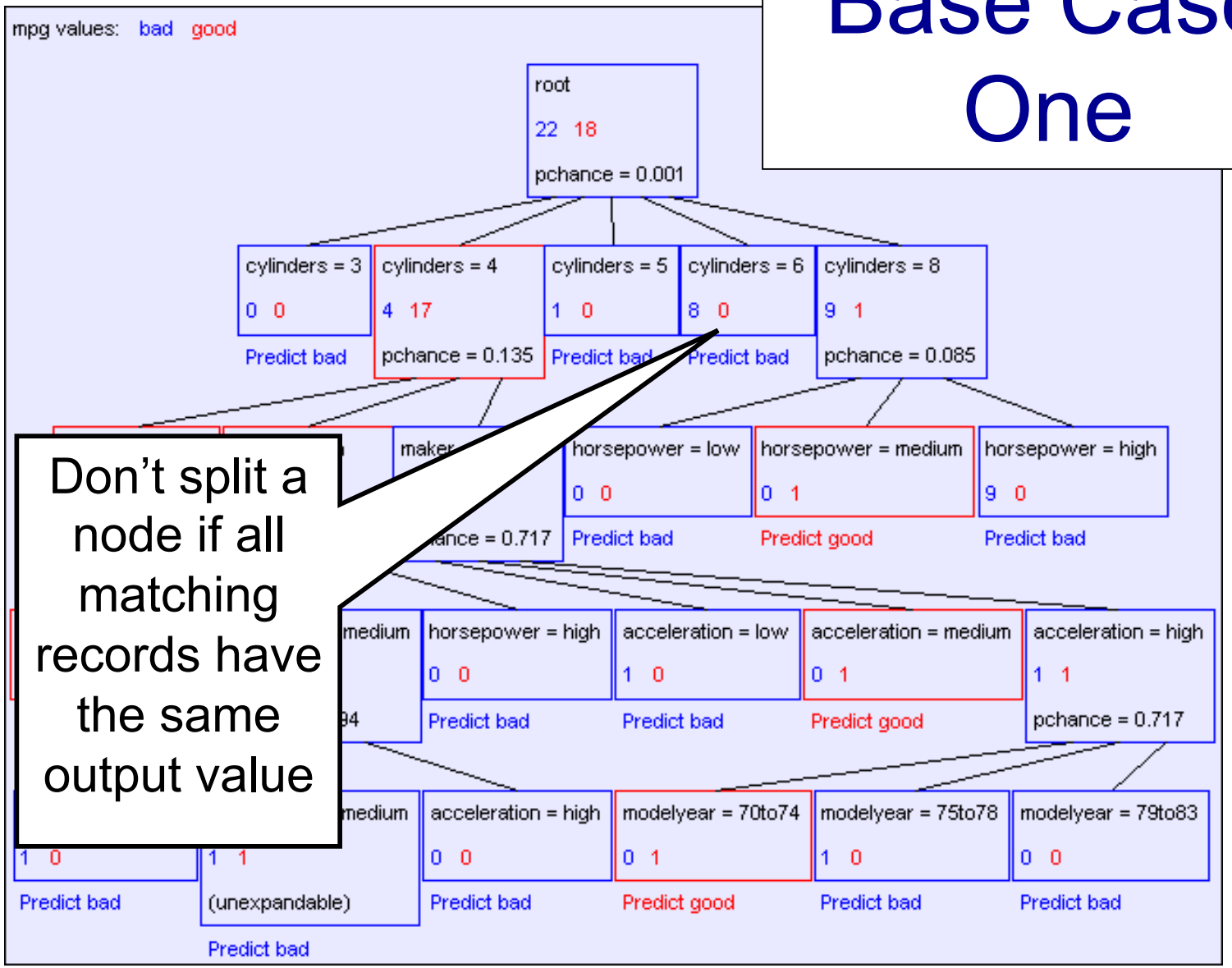
$$\arg\max_i IG(X_i) = \arg\max_i H(Y) - H(Y \mid X_i)$$

- Recurse



# When to stop?

- **Base Case One:** If all records in current data subset have the same output, then don't recurse

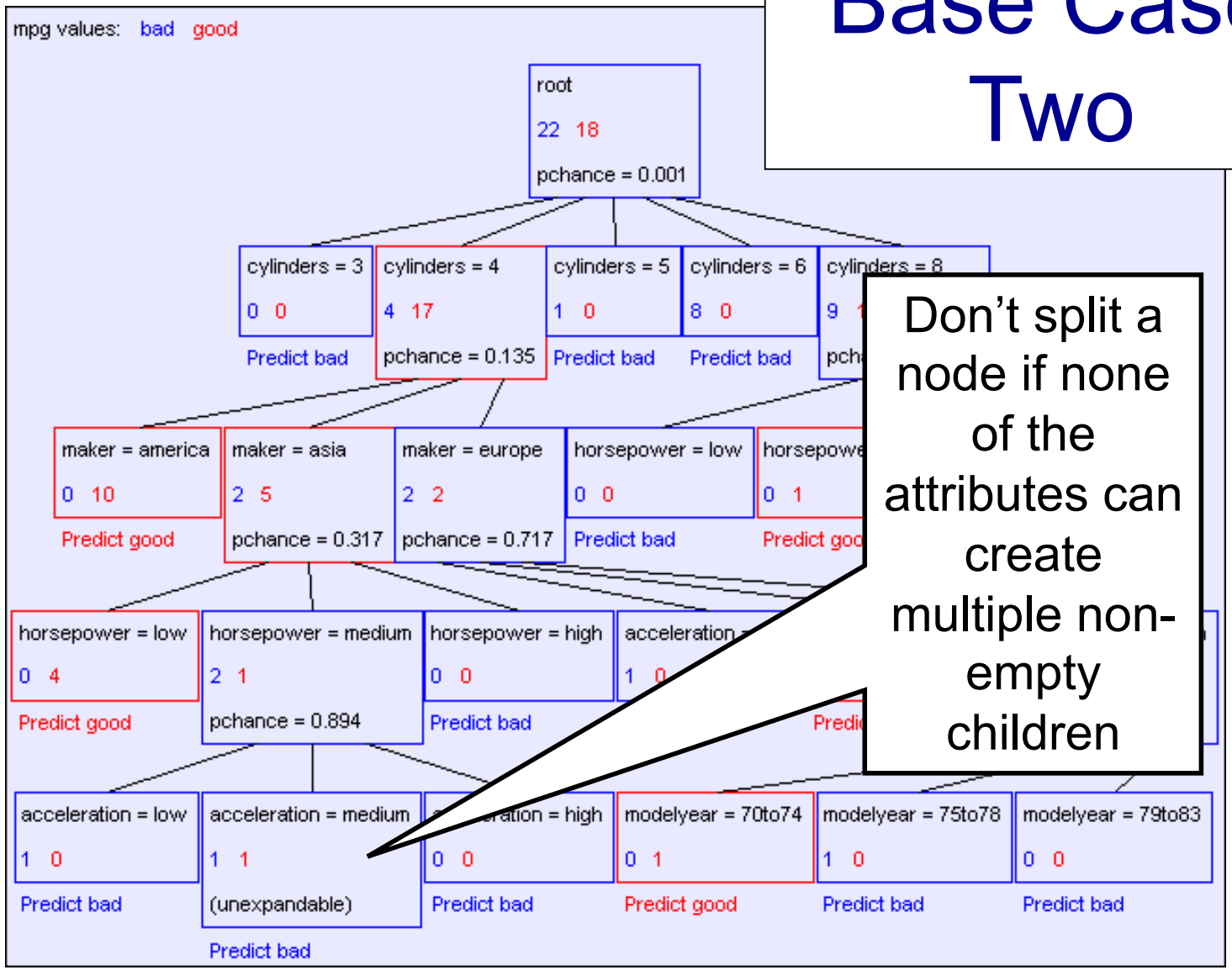



Don't split a node if all matching records have the same output value

Base Case One

# When to stop?

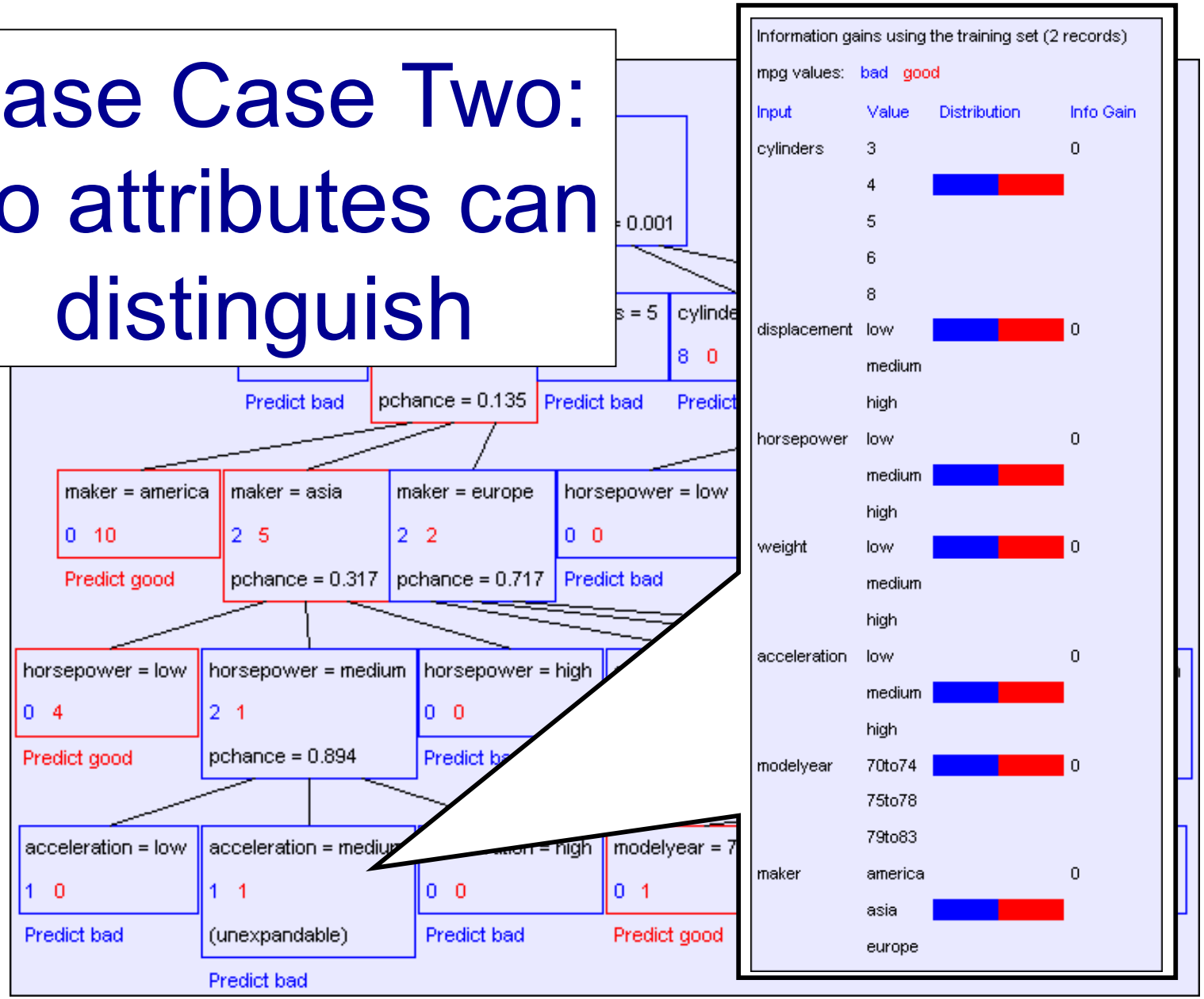
- **Base Case One:** If all records in current data subset have the same output, then don't recurse
- **Base Case Two:** If all records have exactly the same set of input attribute values, then don't recurse



# When to stop?

- **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 attribute values, then don't recurse

**Base Case Two: No attributes can distinguish**



# Summary: Building Decision Trees

BuildTree(*DataSet,Output*)

- If all output values are the same in *DataSet*, return a leaf node that says "predict this unique output"
- If all input values are the same, return a leaf node that says "predict the majority output"
- Else find attribute *X* with highest Info Gain
- Suppose *X* has $n_X$ distinct values (i.e. X has arity $n_X$).
  - Create a non-leaf node with $n_X$ children.
  - The *i*'th child should be built by calling

    BuildTree($DS_i$,*Output*)

    Where $DS_i$ contains the records in DataSet where X = *i*th value of X.

# Gain Ratio

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

- Limitation of Information Gain
  - biased towards choosing attributes with a large number of values
  - won't work for new data
- Information Gain Ratio
  - adjust Information Gain by the entropy of the partitioning (SplitINFO)
  - Attributes with higher SplitINFO are less useful

$$GR(X) = \frac{IG(X)}{SplitINFO}$$

$$SplitINFO = -\sum_{k=1}^{K} \frac{n_k}{n} log \frac{n_k}{n}$$

Parent node is split into $K$ partitions

the number of records in partition $k$

total number of records in the current parent node

# Gain Ratio - Example

| ID code | Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|---|
| A | Sunny | Hot | High | False | No |
| B | Sunny | Hot | High | True | No |
| C | Overcast | Hot | High | False | Yes |
| D | Rainy | Mild | High | False | Yes |
| E | Rainy | Cool | Normal | False | Yes |
| F | Rainy | Cool | Normal | True | No |
| G | Overcast | Cool | Normal | True | Yes |
| H | Sunny | Mild | High | False | No |
| I | Sunny | Cool | Normal | False | Yes |
| J | Rainy | Mild | Normal | False | Yes |
| K | Sunny | Mild | Normal | True | Yes |
| L | Overcast | Mild | High | True | Yes |
| M | Overcast | Hot | Normal | False | Yes |
| N | Rainy | Mild | High | True | No |

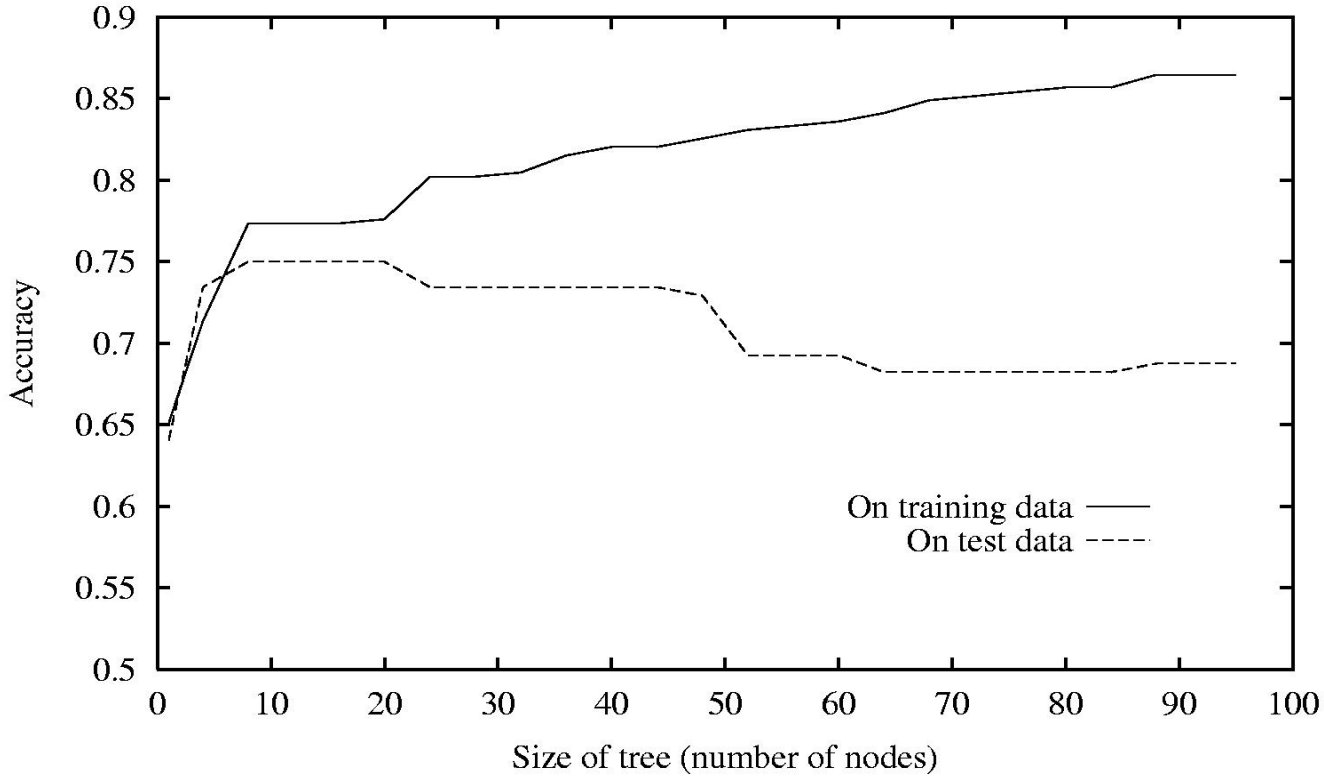
e.g., Split with Outlook, [5, 4, 5]

$$SplitINFO = -\frac{5}{14}\log_2\frac{5}{14} - \frac{4}{14}\log_2\frac{4}{14} - \frac{4}{14}\log_2\frac{4}{14}$$

| **Outlook** | | **Temp.** | |
|---|---|---|---|
| Entropy: | 0.940 | Entropy: | 0.940 |
| Info Gain: | 0.247 | Info Gain: | 0.029 |
| SplitINFO: | 1.577 | SplitINFO: | 1.557 |
| Gain Ratio: | 0.157 | Gain Ratio: | 0.019 |
| **Humidity** | | **Windy** | |
| Entropy: | 0.940 | Entropy: | 0.940 |
| Info Gain: | 0.152 | Info Gain: | 0.048 |
| SplitINFO: | 1.000 | SplitINFO: | 0.985 |
| Gain Ratio: | 0.152 | Gain Ratio: | 0.049 |

# Decision trees will overfit!!

- Training set error will be zero!
- Many strategies for picking simpler trees
  - Fixed depth
  - Early stopping
  - Fixed number of leaves
  - Or something smarter…



# One definition of overfitting

- Assume:
  - Data Generated from distribution $D(X, Y)$
  - A hypothesis space $H$
- Define errors for hypothesis $h \in H$
  - Training set error: $error_{train}(h)$
  - Data error: $error_D(h)$
- We say $h$ overfits the training data if there exists an $h' \in H$ such that:

$$error_{train}(h) < error_{train}(h')$$

and

$$error_D(h) > error_D(h')$$

# Real-Valued Inputs

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

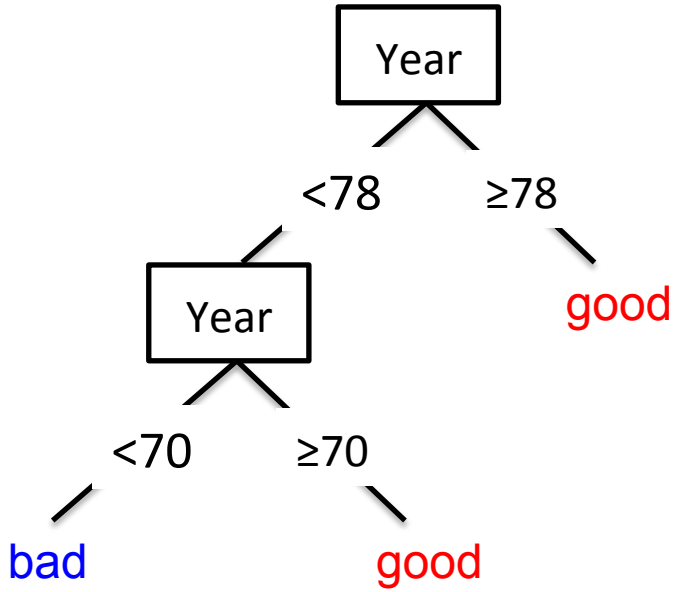
Infinite number of possible split values!!!

Finite dataset, only finite number of relevant splits!

| 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 |
| | | | | | | | |

# 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 the same variable
- Search through possible values of $t$
  - Seems hard !!!
  - But only finite number of t's are important



# Picking the best threshold

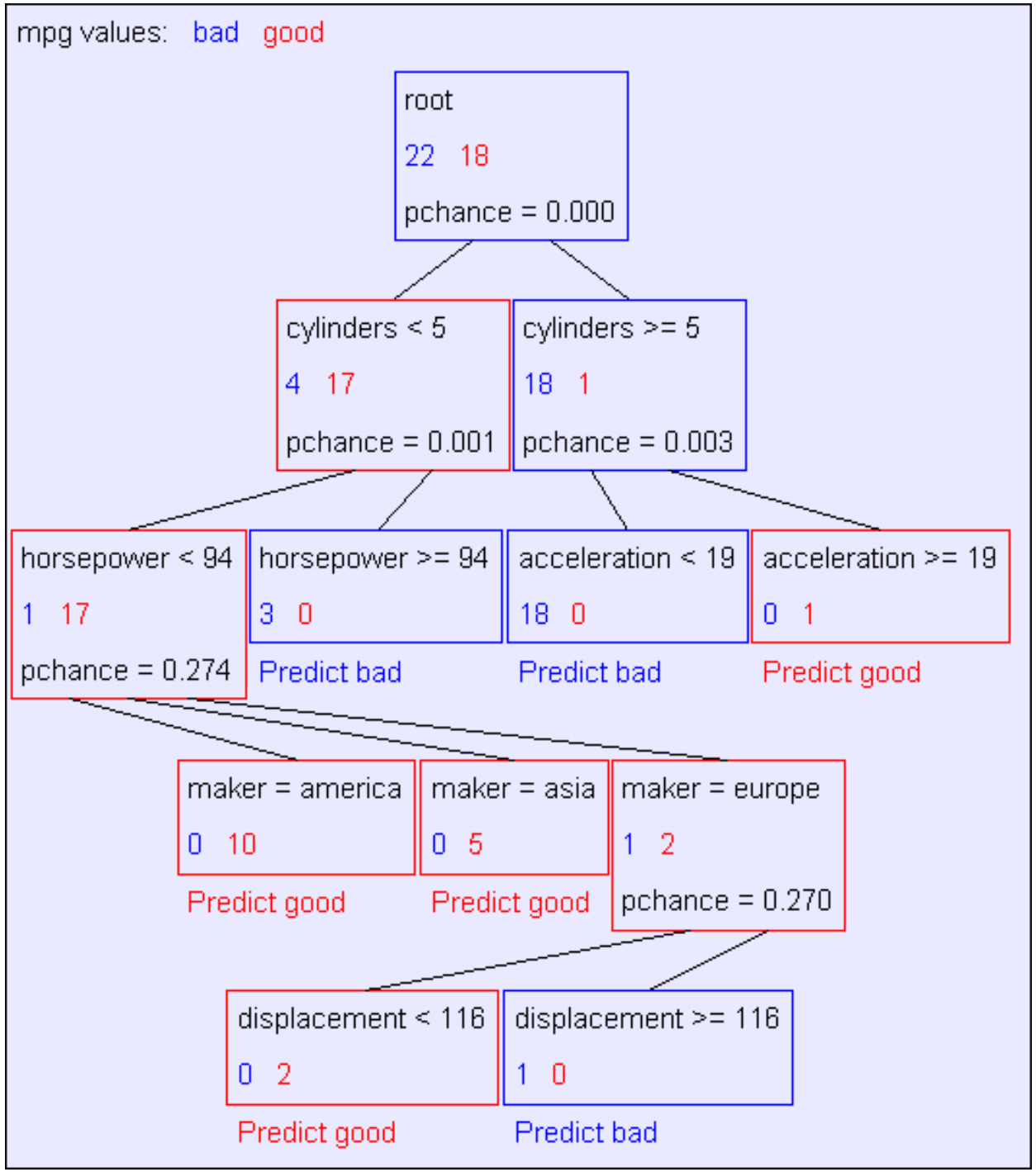
- Suppose $X$ is real valued with threshold $t$
- *Goal* $IG(Y|X:t)$: the information gain for $Y$ when testing if $X$ is greater than or less than $t$
- Define:
  - *H(Y|X:t) =*
    *H(Y|X < t) P(X < t) + H(Y|X >= t) P(X >= t)*
  - *IG(Y|X:t) = H(Y) - H(Y|X:t)*
  - *IG*(Y|X) = max$_t$ IG(Y|X:t)*
- Use: $IG(Y|X)$ for continuous variables

# Example with MPG

Information gains using the training set (40 records)

mpg values: bad good

| Input | Value | Distribution | Info Gain |
|---|---|---|---|
| cylinders | < 5 | | 0.48268 |
| | >= 5 | | |
| displacement | < 198 | | 0.428205 |
| | >= 198 | | |
| horsepower | < 94 | | 0.48268 |
| | >= 94 | | |
| weight | < 2789 | | 0.379471 |
| | >= 2789 | | |
| acceleration | < 18.2 | | 0.159982 |
| | >= 18.2 | | |
| modelyear | < 81 | | 0.319193 |
| | >= 81 | | |
| maker | america | | 0.0437265 |
| | asia | | |
| | europe | | |



# 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)
- Understand entropy, conditional entropy, information gain, SplitINFO, Gain Ratio
- Know how to use information gain to select attributes
- 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
    - Fixed number of leaves
    - …