COS 324: Lecture 13

Beyond linear classifiers: decision trees

Elad Hazan   Yoram Singer

PRINCETON UNIVERSITY

This lecture contains material from the
T. Michel text "Machine Learning", and slides adapted from
David Sontag, Luke Zettlemoyer, Carlos Guestrin, and Andrew Moore

Admin

- New exercise – theory – due in two weeks
  (formal announcement next week, but out now for your convenience)

Agenda

Thus far:

- Rigorous definition of (PAC) learnability
- Efficient algorithms for learning based on convex optimization
- linear classifiers (perceptron, SGD, multiclass,…)

Today:

- Decision trees
- Build up for other non-linear machines (& neural networks)

Classification

Goal: Find *best* mapping from domain (features) to output (labels)

- Given a document (email), classify spam or ham.
  Features = words , labels = {spam, ham}

- Given a picture, classify if it contains a chair or not
  features = bits in a bitmap image, labels = {chair, no chair}

GOAL: automatic machine that learns from examples

Terminology for learning from examples:
- Set aside a ”training set” of examples, train a classification machine
- Test on a “test set”, to see how well machine performs on unseen examples

Classifying fuel efficiency

- 40 data points
- Goal: predict MPG
- Need to find:
  $f : X \rightarrow Y$
- Discrete data (for now)

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

$\underbrace{\text{mpg}}_{Y}\ \underbrace{\text{cylinders} \ldots \text{maker}}_{X}$

Decision trees for classification

- Why use decision trees?
- What is their expressive power?
- Can they be constructed automatically?
- How accurate can they classify?
- How well do decision trees generalize? (sample complexity)
- Computational complexity of finding the best tree

Decision trees for classification

Some real examples (from Russell & Norvig, Mitchell)

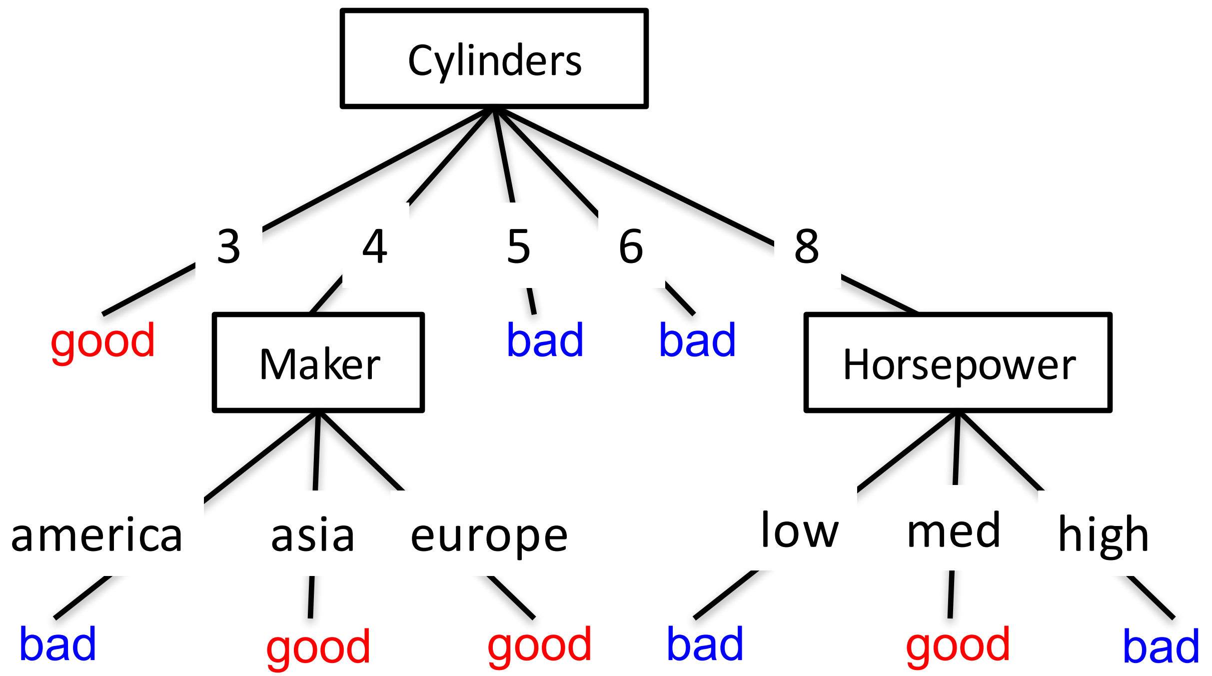
- BP’s GasOIL system for separating gas and oil on offshore platforms - decision trees replaced a hand-designed rules system with 2500 rules. C4.5-based system outperformed human experts and saved BP millions. (1986)
- learning to fly a Cessna on a flight simulator by watching human experts fly the simulator (1992)
- can also learn to play tennis, analyze C-section risk, etc.

Decision trees for classification

- interpretable/intuitive, popular in medical applications because they mimic the way a doctor thinks
- model discrete outcomes nicely
- C4.5 and CART - from “top 10 data mining methods” - very popular
- very expressive

decision trees $f : X \rightarrow Y$

- Each internal node tests an attribute x_i
- One branch for each possible 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

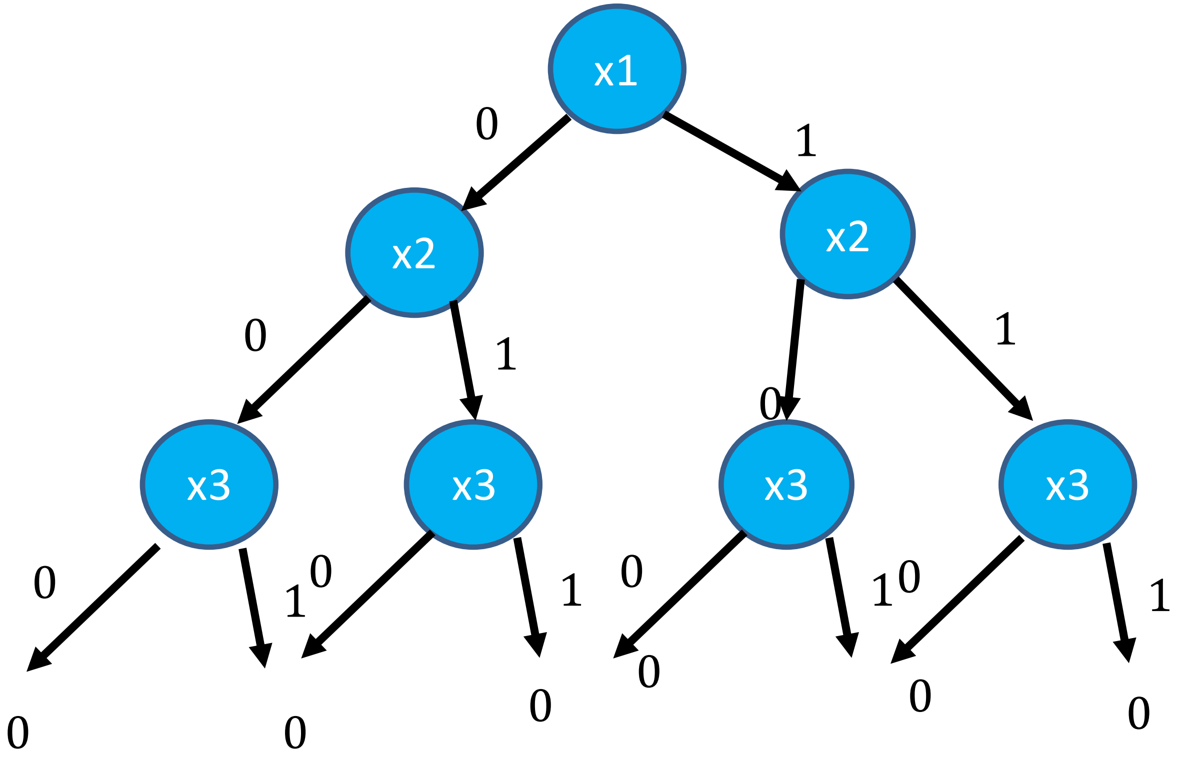


Human interpretable!

Expressive power of DT

Consider Boolean functions
F = $\{0,1\}^n \mapsto \{0,1\}$

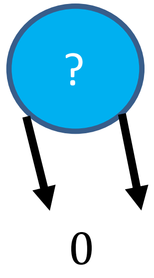
- How many functions can DT express?



X1	X2	X3	F(X1,X2,X3)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Sample complexity of DT

- Sample complexity of all decision trees?
- Smaller trees? (bound their size)



X1	X2	X3	F(X1,X2,X3)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

What is the Simplest Tree?

predict

mpg=bad

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

Is this a good tree?

[22+, 18-]

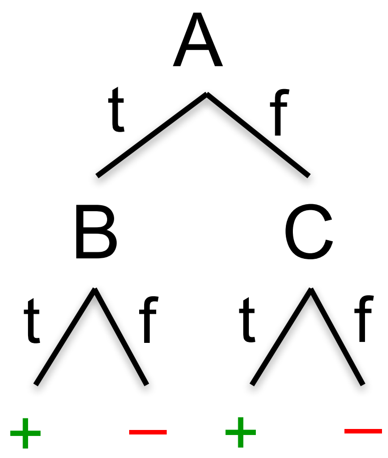
Means:

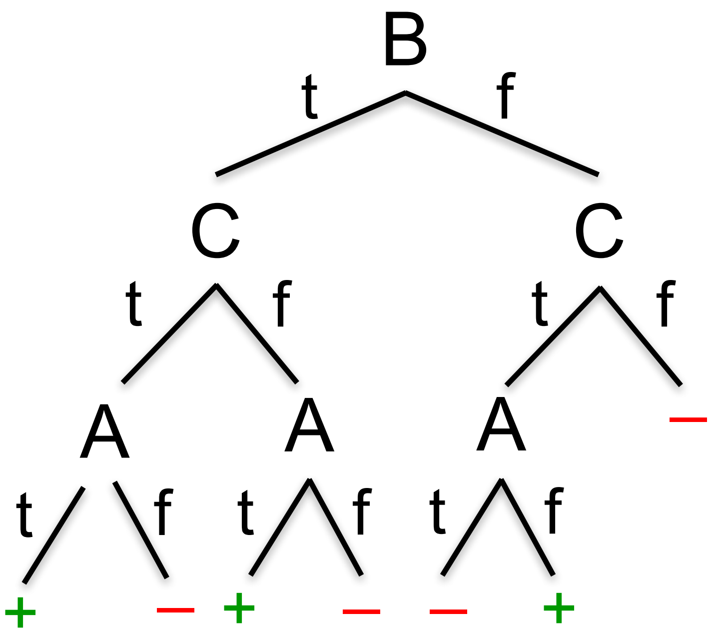
correct on 22 examples

incorrect on 18 examples

Are all decision trees equal?

- Many trees can represent the same concept
- But, not all trees will have the same size!
  - e.g., ((A and B) or (not A and C))





- Which tree do we prefer?

Sample complexity of DT

- How many trees over d Boolean variables with k nodes?

$$\leq d^k \times (2k+1)!$$

Choose k variables from d, with repetition

A binary tree over k variables has k+1 leaves.
To bound how many such trees: add k+1 “leaves”, write down the tree in DFS order.

Sample complexity of DT

- How many trees over d Boolean variables with k nodes?

$$d^k \times (2k+1)!$$

Choose k variables from d, with repetition

A binary tree over k variables has k+1 leaves.
To bound how many such trees: add k+1 “leaves”, write down the tree in DFS order.

Thus, by fundamental theorem of statistical learning, sample complexity is:

$$O\left(\frac{\log|H| + \log\frac{1}{\delta}}{\epsilon}\right) = O\left(\frac{\mathrm{k}\log(\mathrm{d}) + \log\frac{1}{\delta}}{\epsilon}\right)$$

Computational complexity of DT

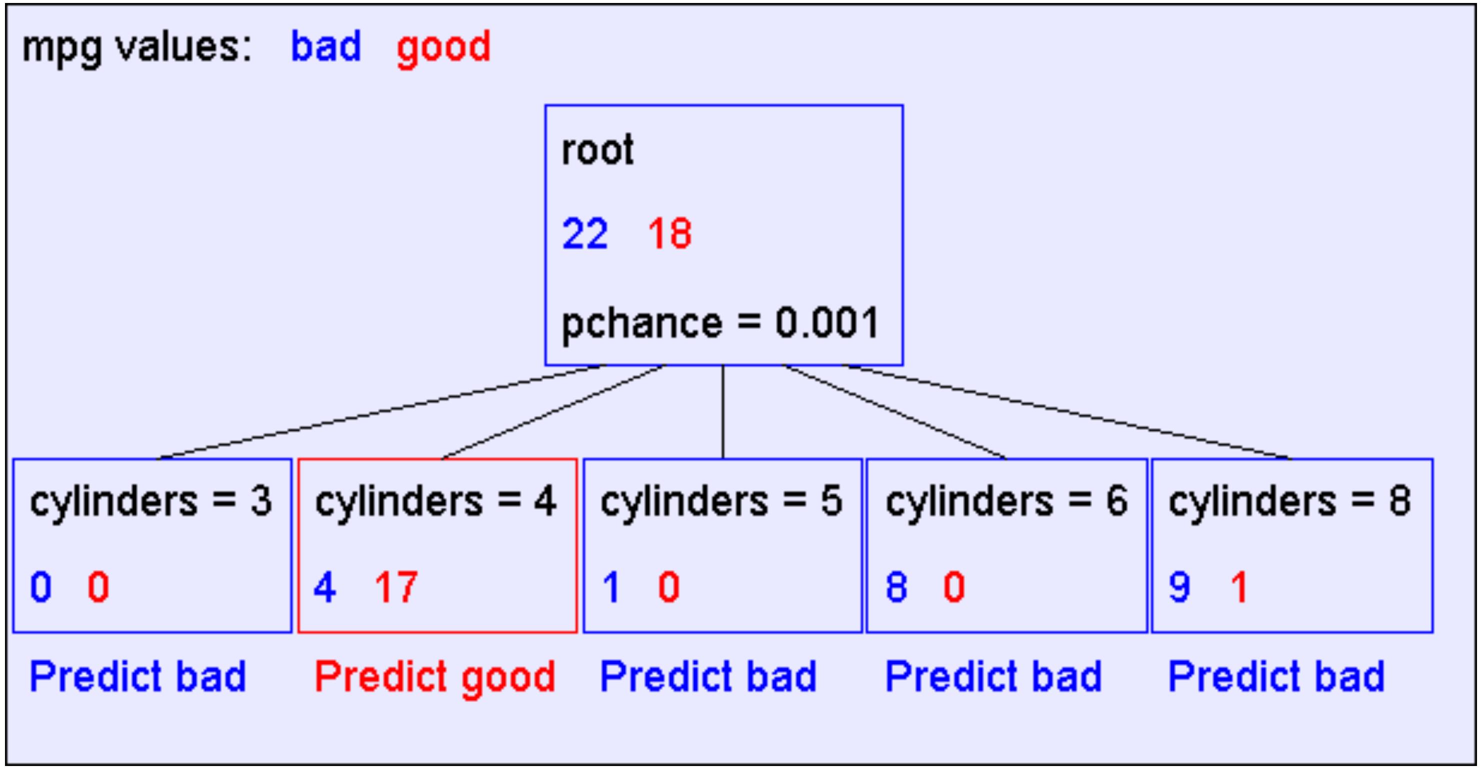
- Good news: sample complexity is descent.
  Fundamental theorem says we can learn with ERM rule!

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

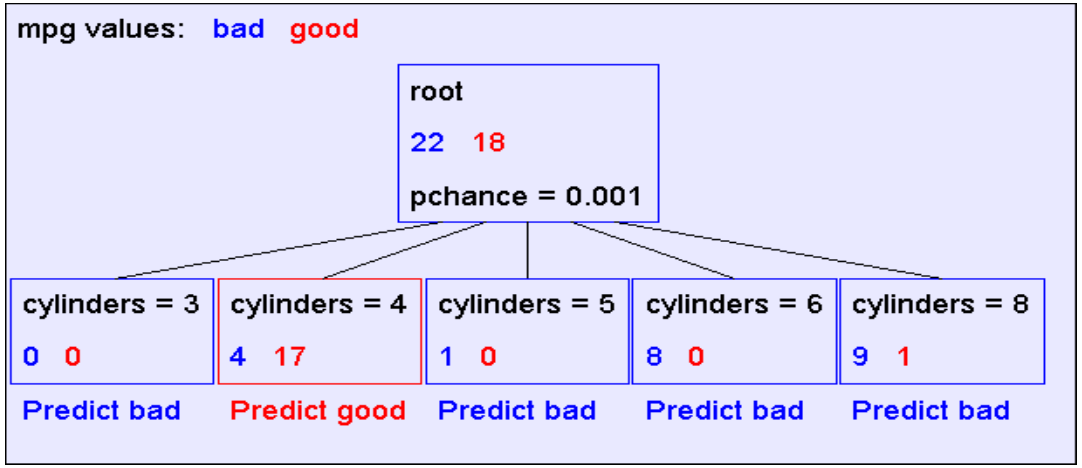
- Solution 1: Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on **next best attribute (feature)**
  - Recurs

- Next week: more rigorous theoretical solution – Boosting!

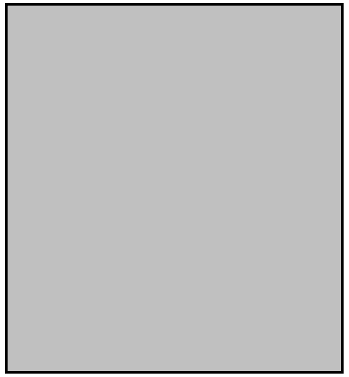
A Decision Stump



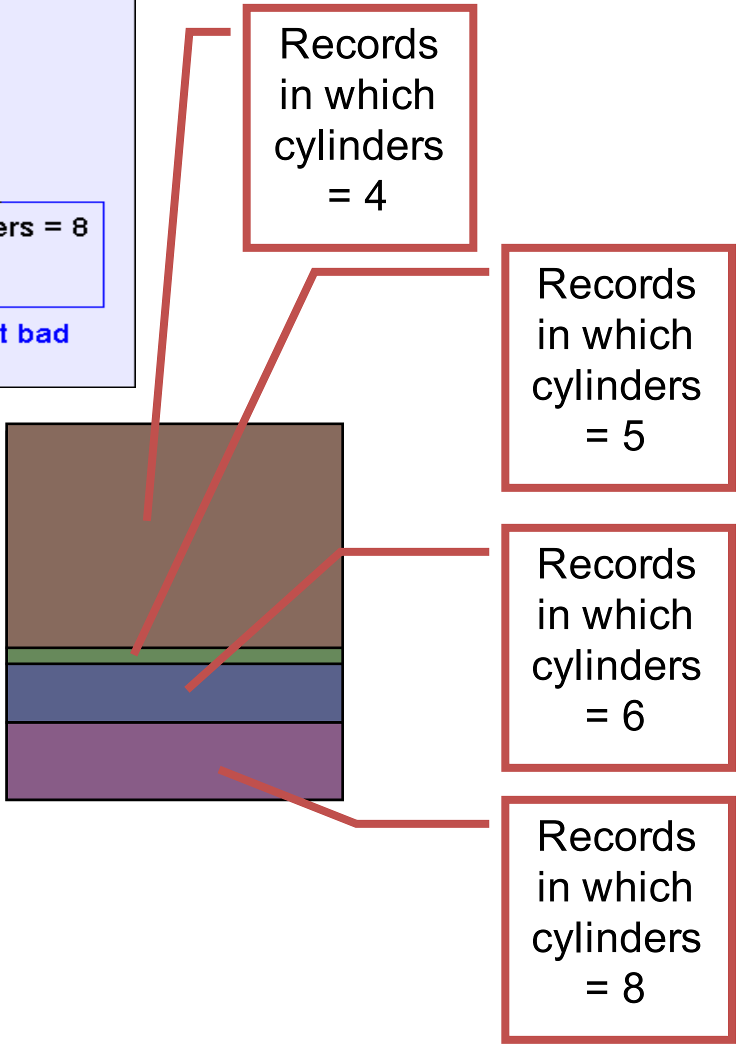
Key idea: Greedily learn trees using **recursion**



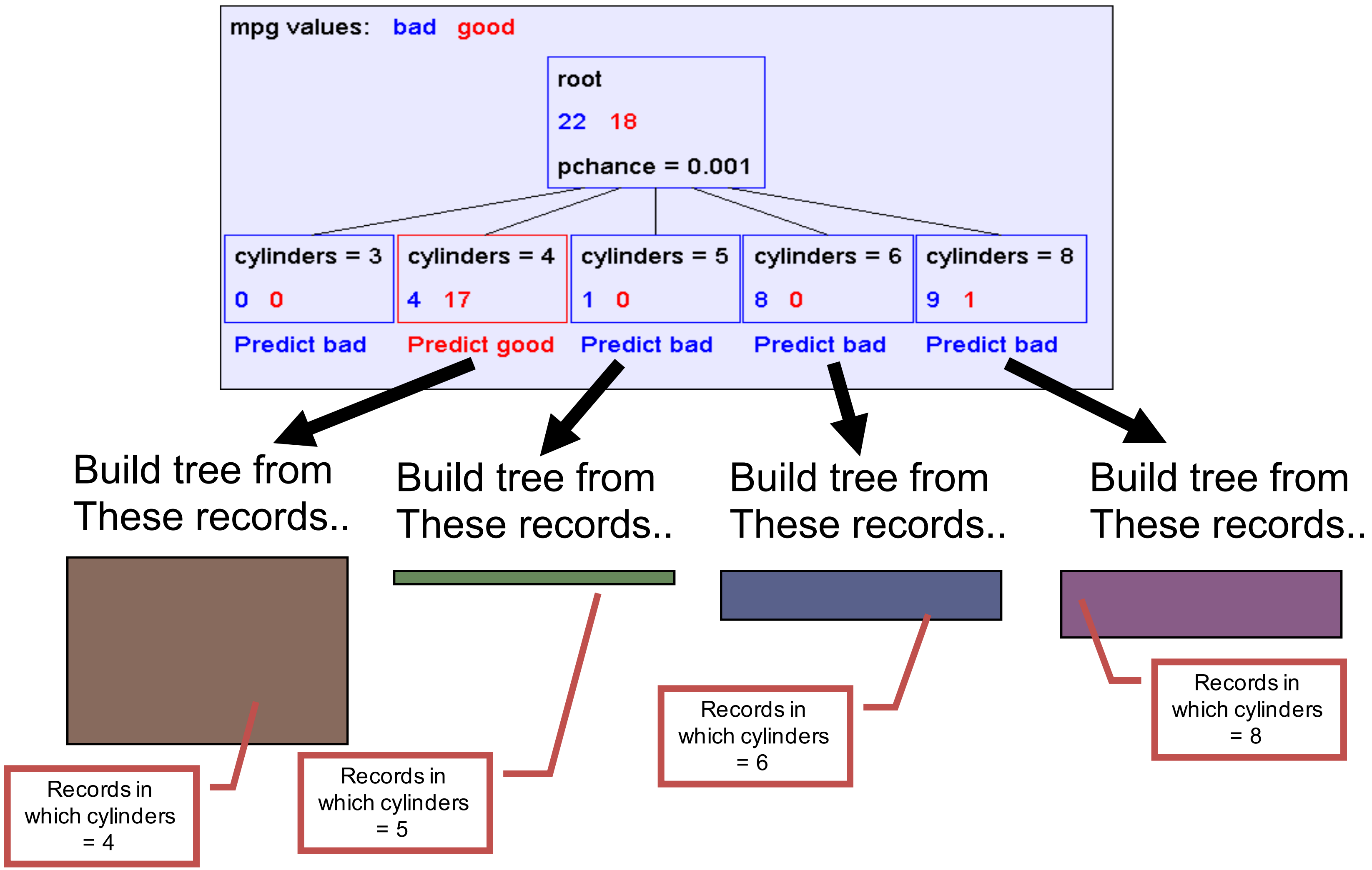
Take the Original Dataset..



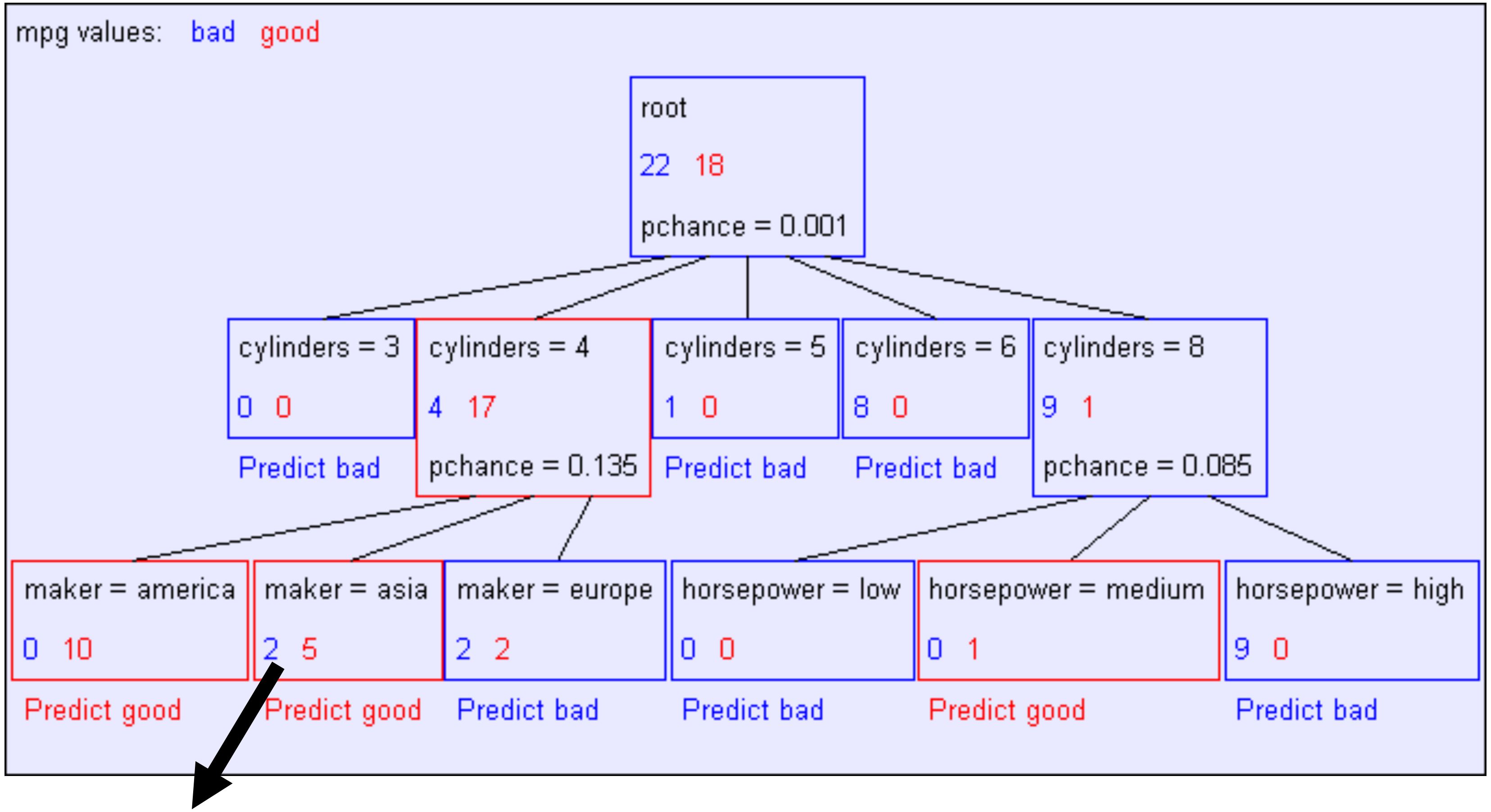
And partition it according to the value of the attribute we split on



Recursive Step



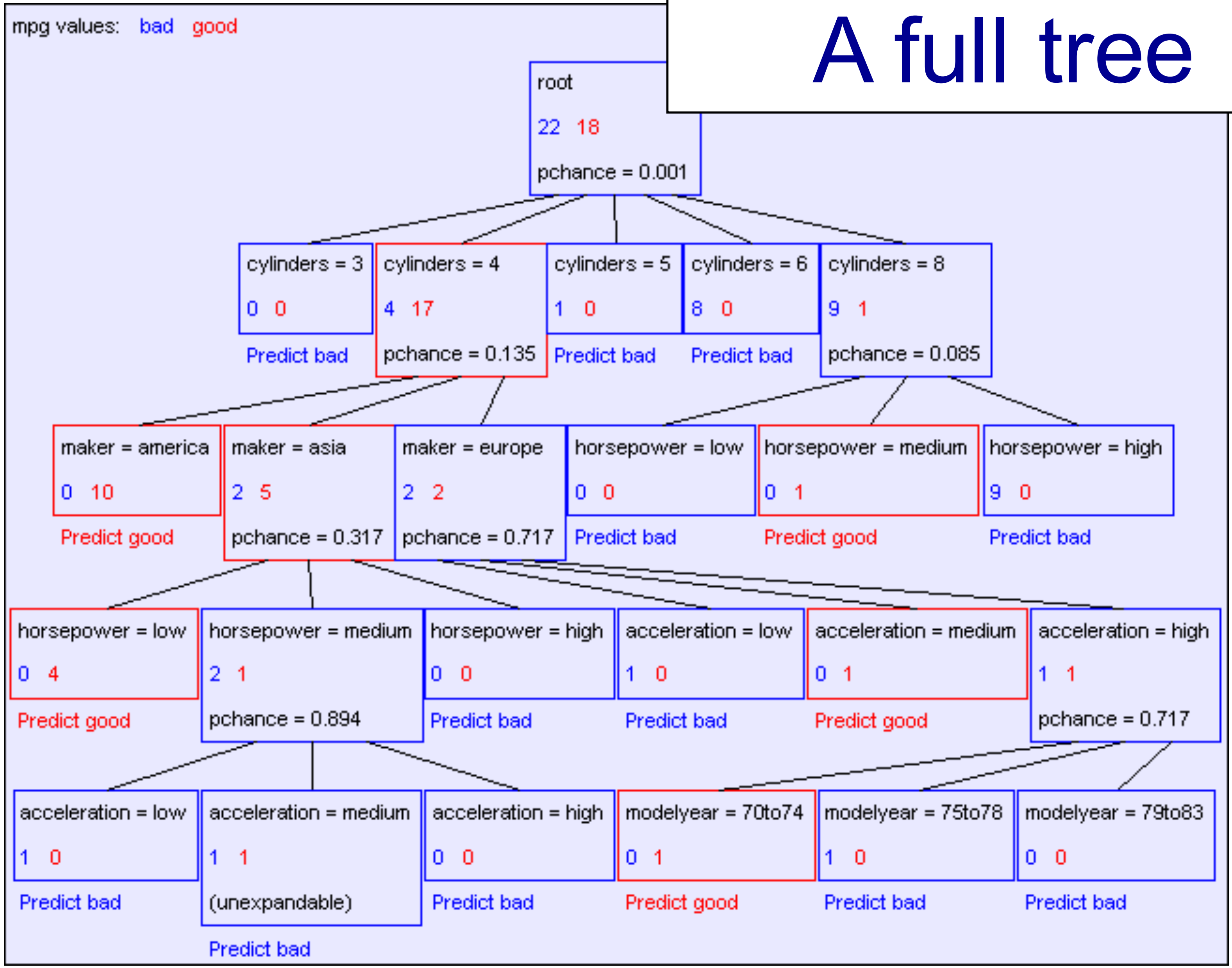
Second level of tree



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)

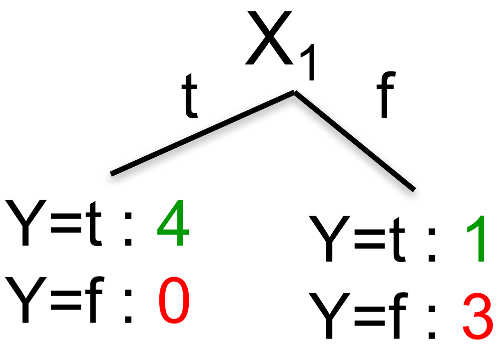
A full tree

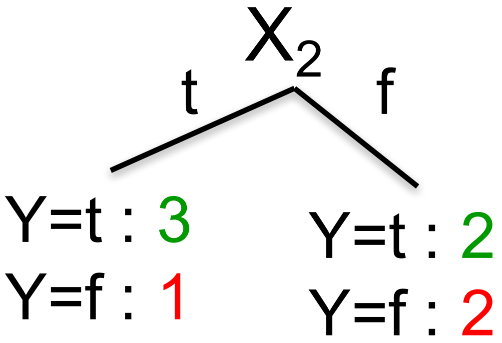


mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 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

Splitting: choosing a good attribute

Would we prefer to split on X_1 or X_2?





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!

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

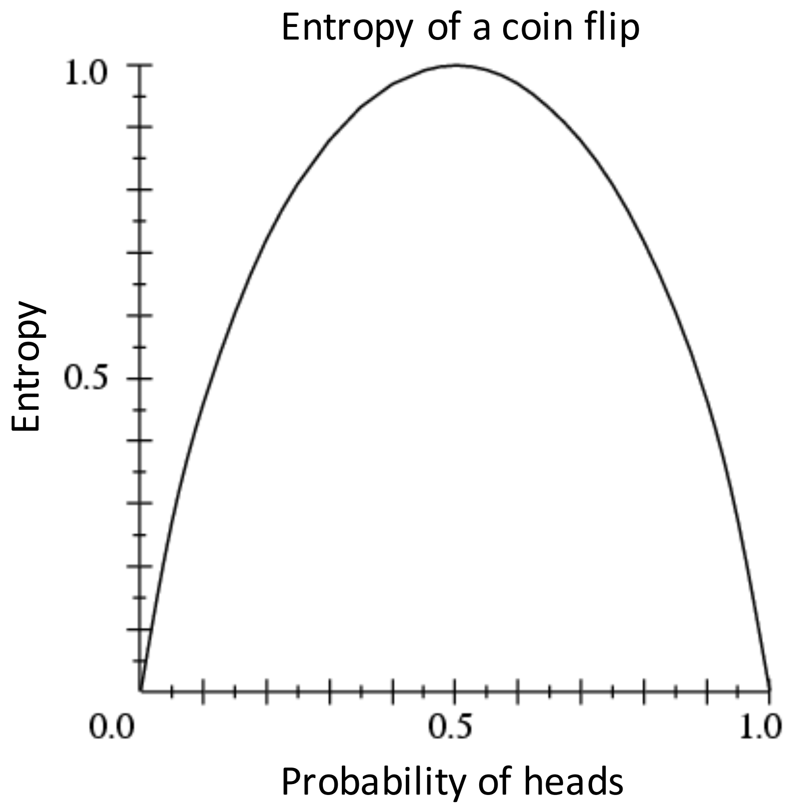
Entropy

Entropy $H(Y)$ of 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)

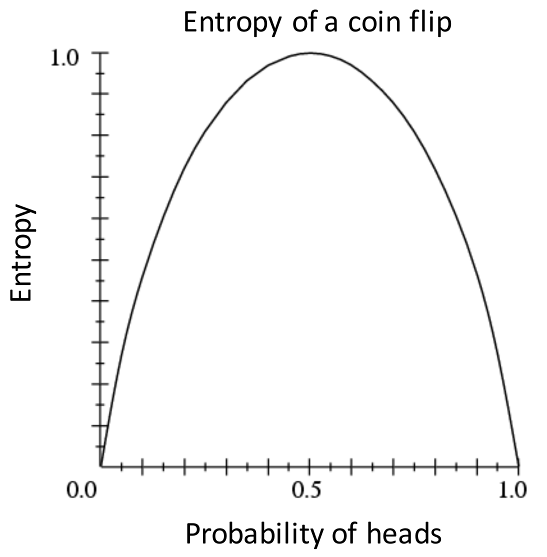


High, Low Entropy

- "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

(Slide from Vibhav Gogate)

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

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

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

$$= 0.65$$

Conditional Entropy

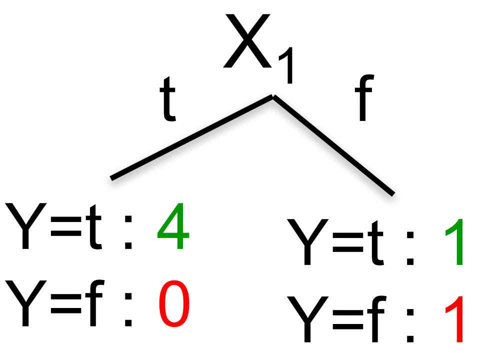
Conditional Entropy $H(Y\,|X)$ of a random variable Y conditioned on a random variable X

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

Example:

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

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



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

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

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

$= 2/6$

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

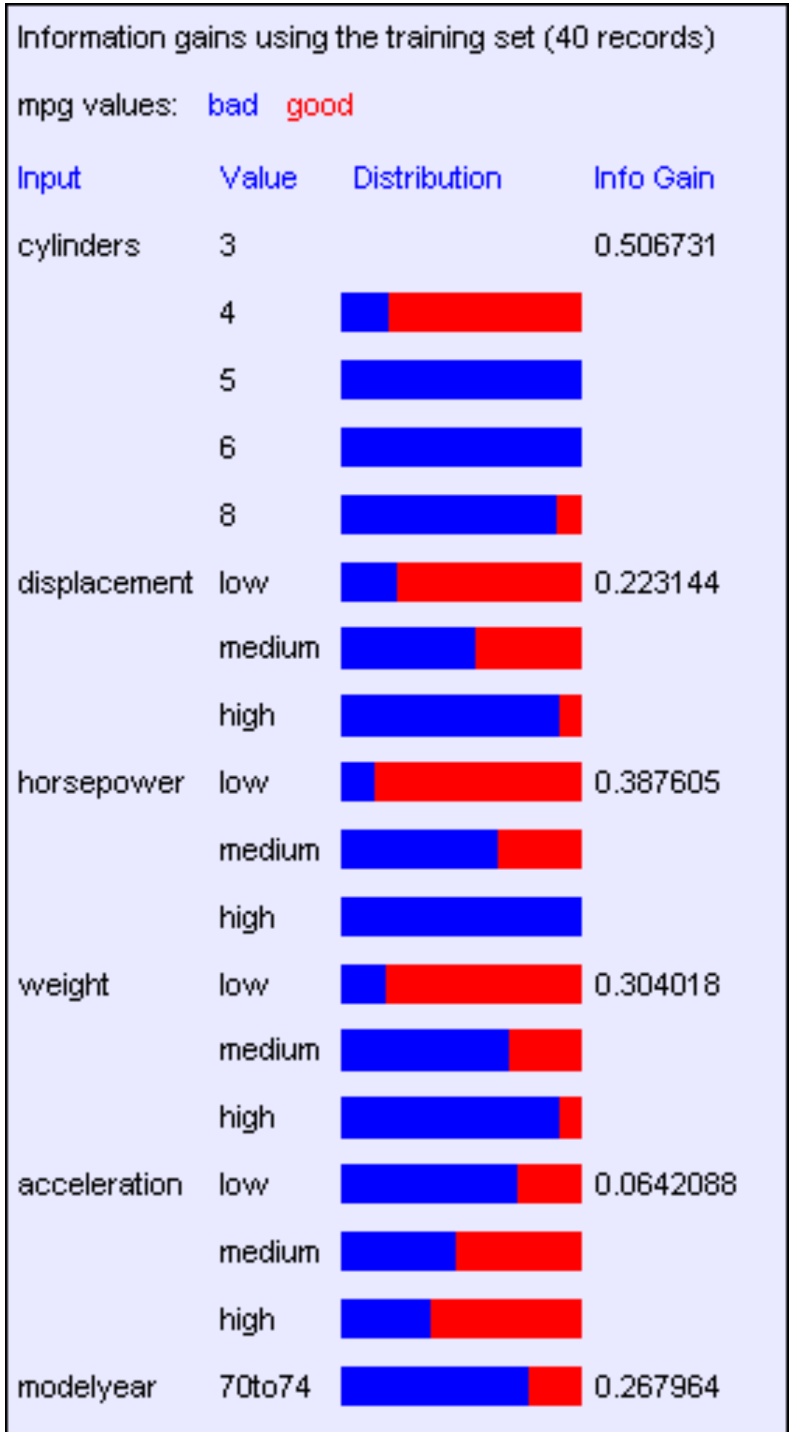
- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute:

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

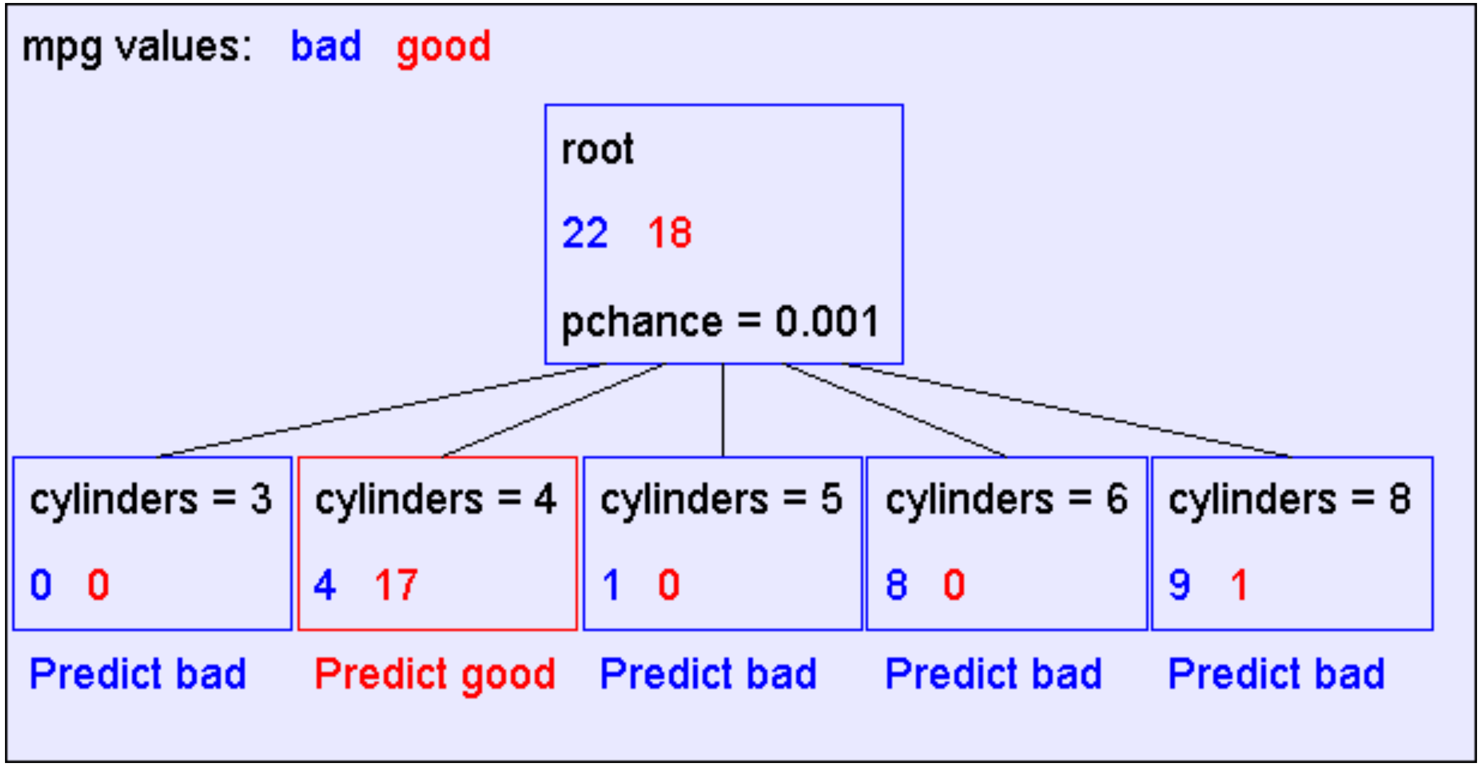
- Recurs

Suppose we want to predict MPG

Look at all the information gains…

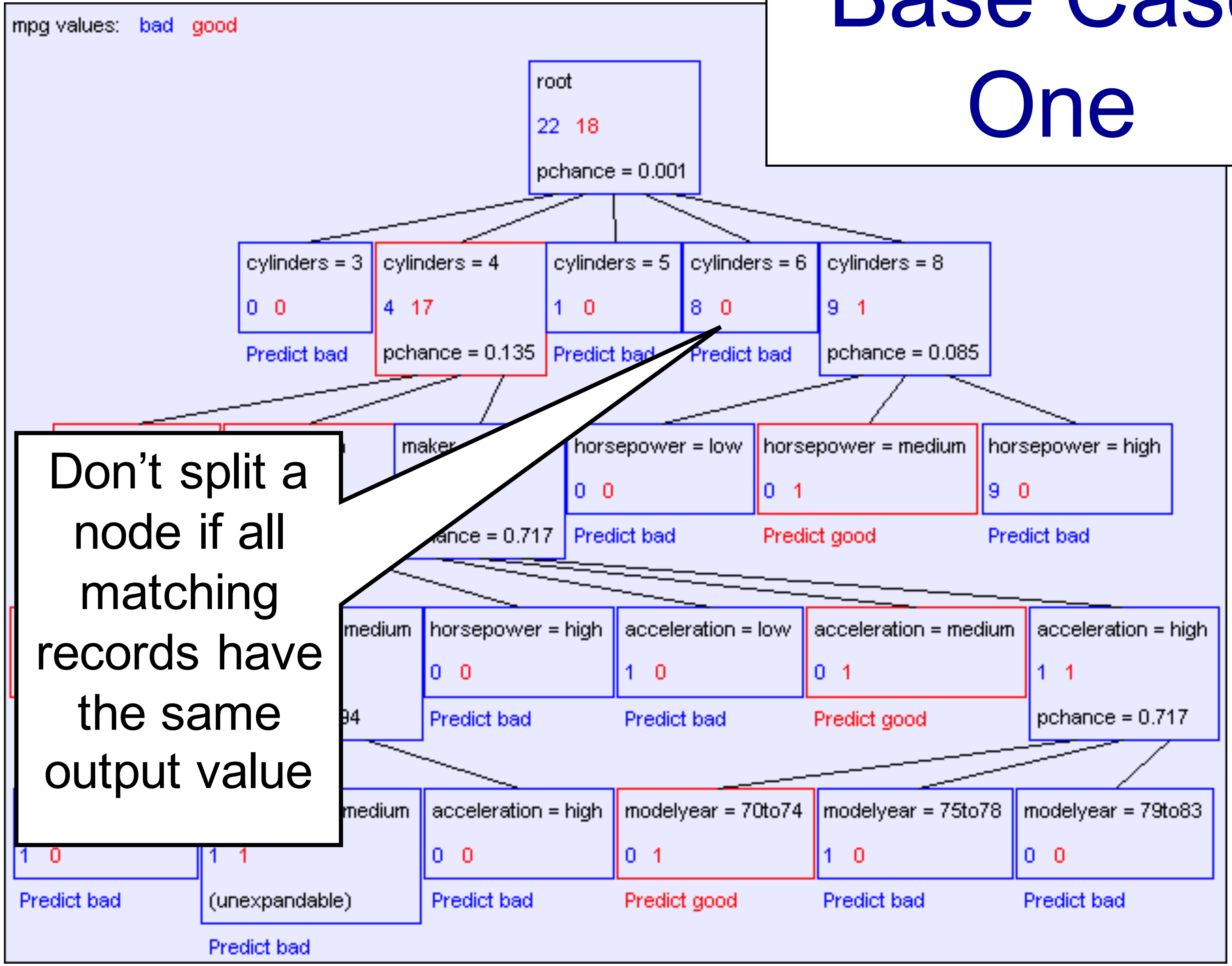


When to stop?

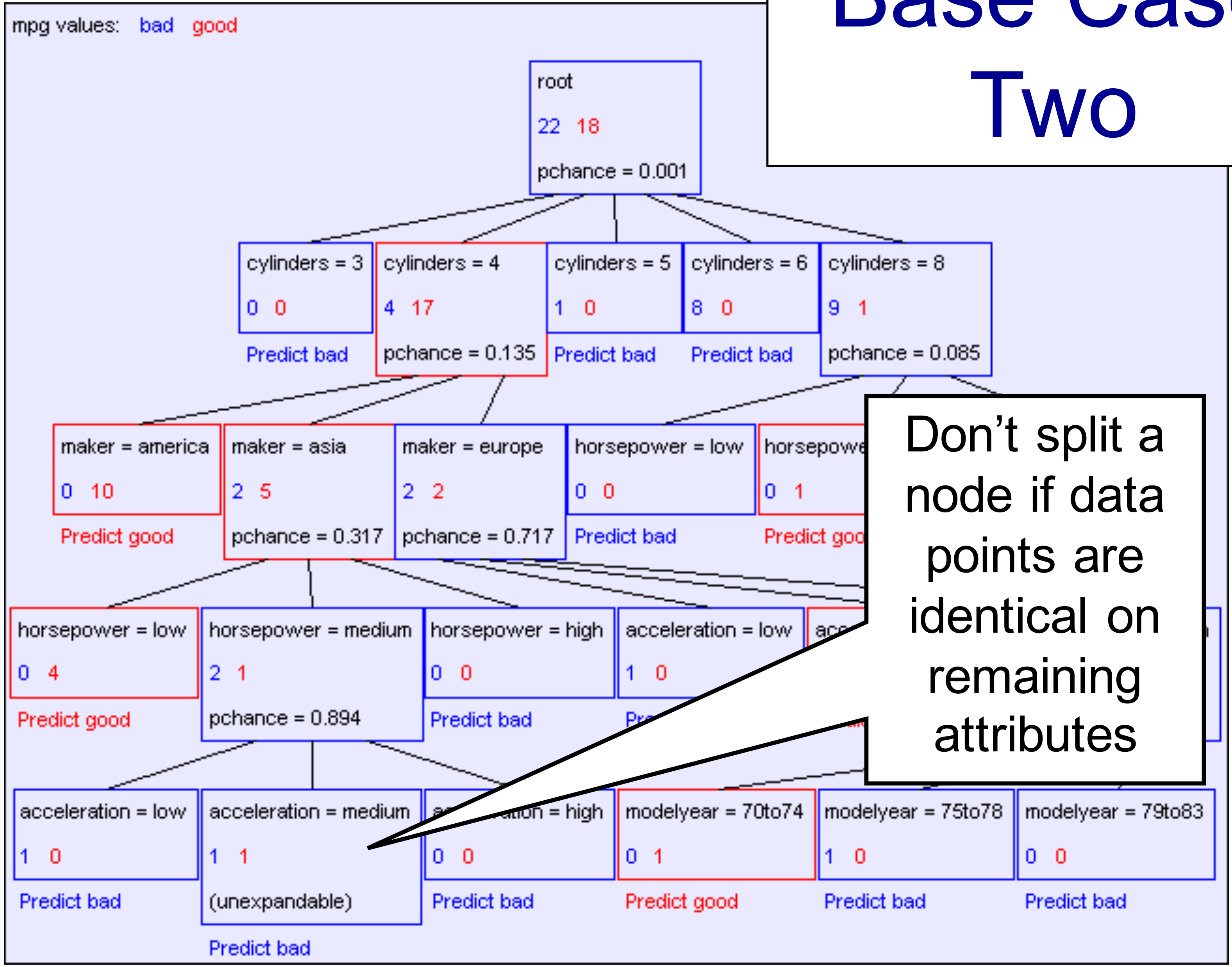


First split looks good! But, when do we stop?

Base Case One



Base Case Two



Don’t split a node if data points are identical on remaining attributes

Base Cases: An idea

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

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

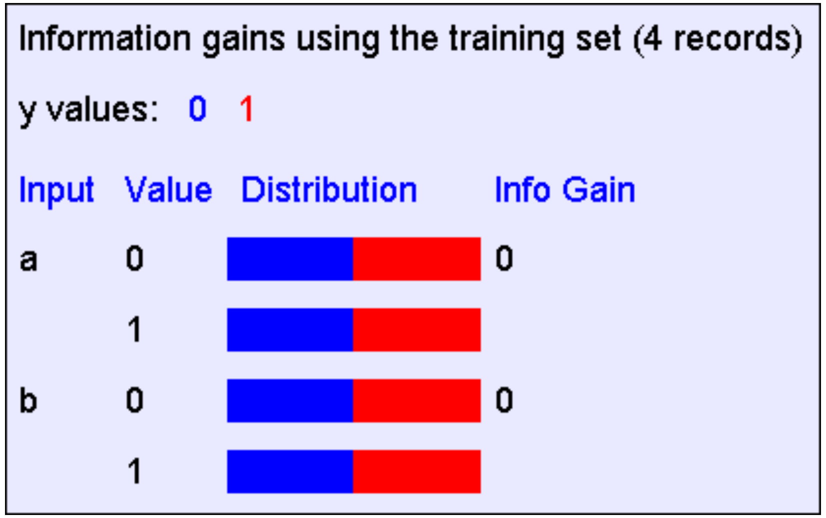
- *This is not a good idea*

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:



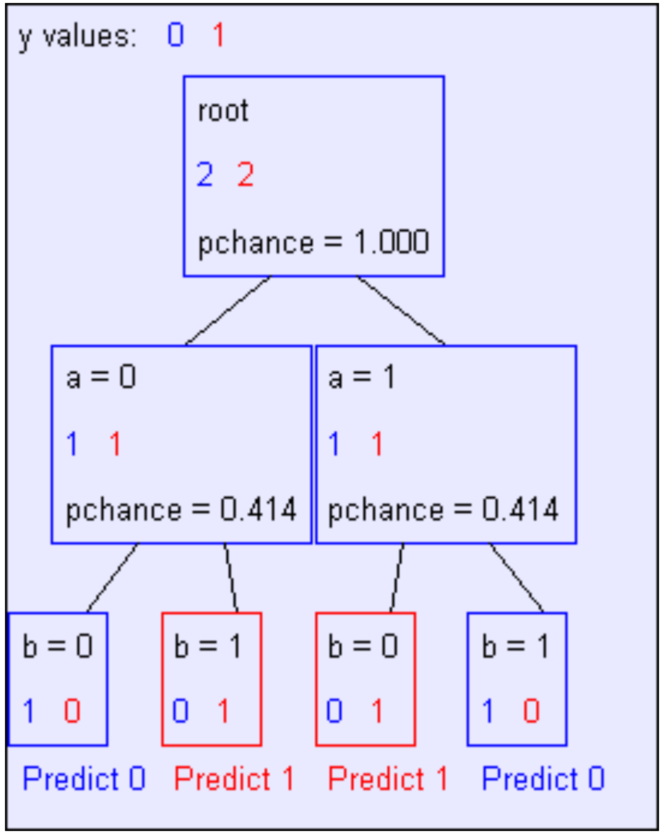
If we omit proposed case 3:

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

The resulting decision tree:



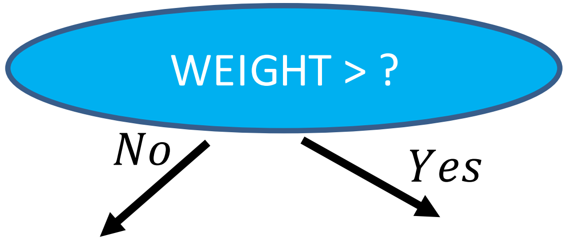
Non-Boolean Features

- Real-valued features?

Real-> threshold

- Number of thresholds <= # of different values in dataset

- Can choose threshold based on information gain



Summary: Building Decision Trees

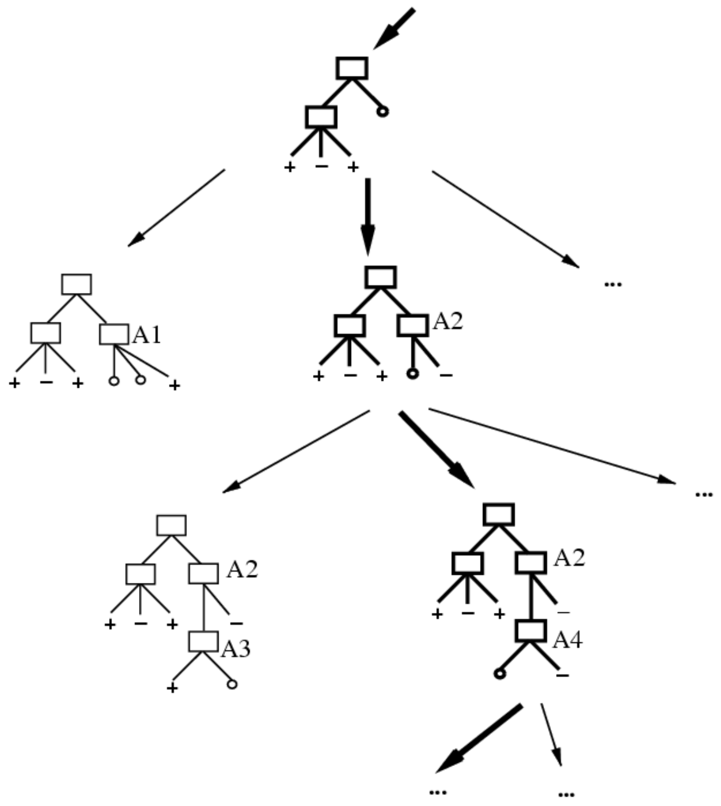
BuildTree(*DataSet*,*Output*)

- 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*)

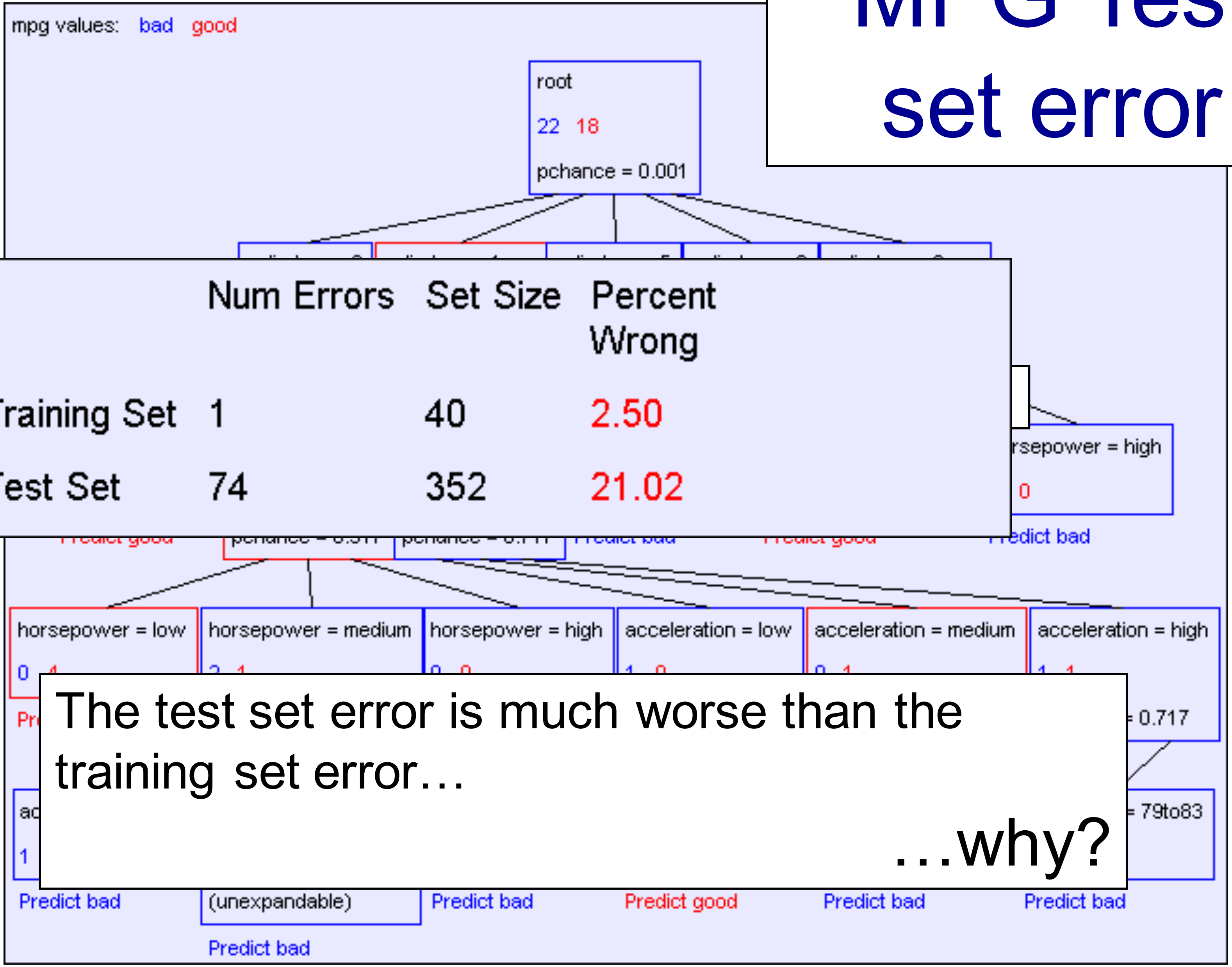
    Where DS_i contains the records in DataSet where X = *i*th value of X.

Machine Space Search



- ID3 / C4.5 / CART search for a succinct tree that perfectly fits the data.
- They are not going to find it in general (NP-hard)
- Entropy-guided splitting – well-performing heuristic. Exists others.

MPG Test set error

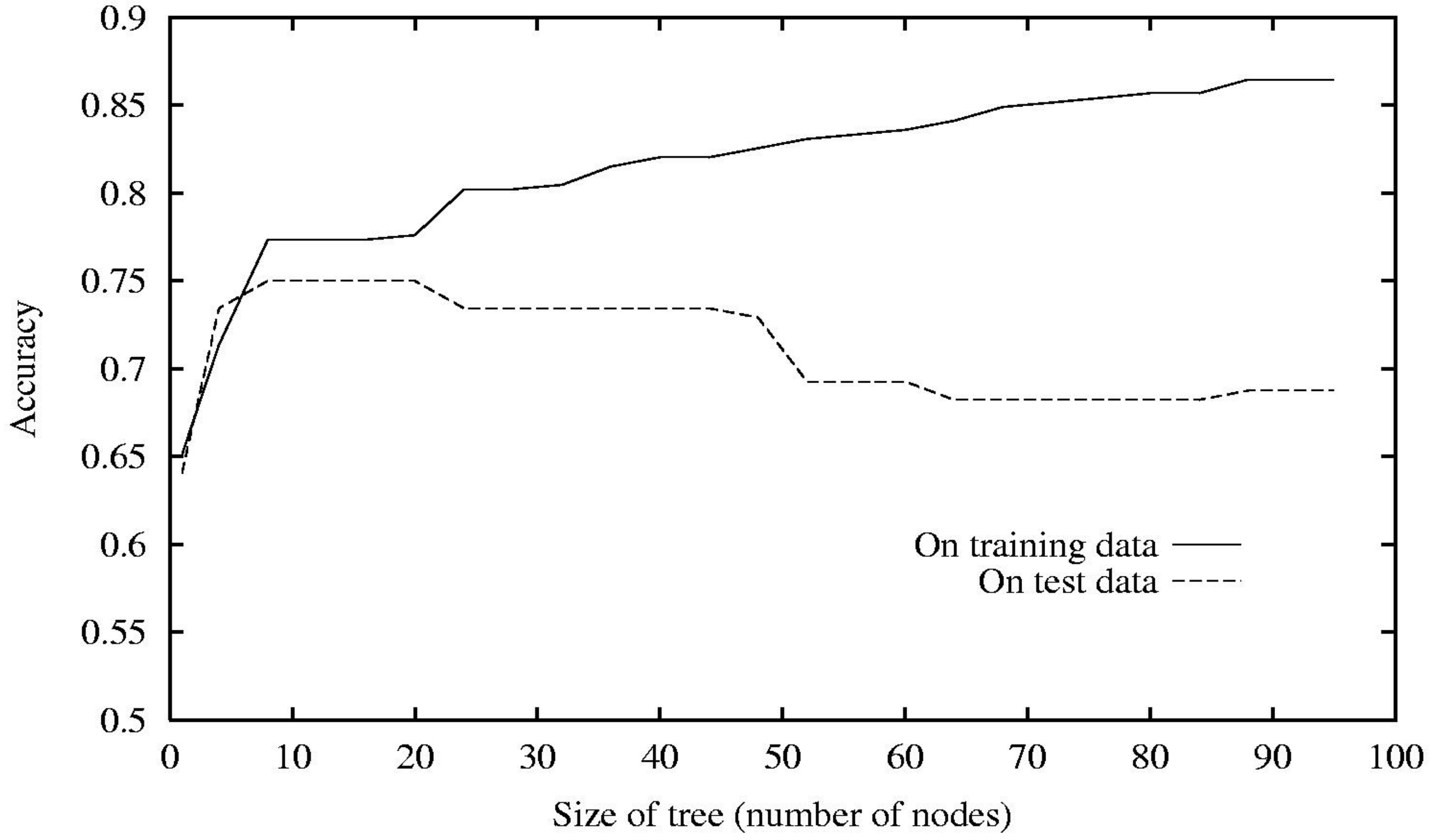


	Num Errors	Set Size	Percent Wrong
Training Set	1	40	2.50
Test Set	74	352	21.02

The test set error is much worse than the training set error…

…why?

Decision trees will overfit



Overfitting

- Precise characterization – statistical learning theory
- Special technics to prevent overfitting in DT learning
  - Pruning the tree, e.g. “reduced error” pruning:
    Do until further pruning is harmful:
    1. Evaluate the impact on validation (test) set of the data of pruning each possible node (and it’s subtree)
    2. Greedily remove one that most improves validation (test) error

- Next lecture: a theoretically sound way to make use of tree heuristics: BOOSTING!