CS639:
Data Management for Data Science

Lecture 16: Intro to ML and Decision Trees

Theodoros Rekatsinas
(lecture by Ankur Goswami many slides from David Sontag)

Today’s Lecture

1. Intro to Machine Learning

2. Types of Machine Learning

3. Decision Trees

1. Intro to Machine Learning

What is Machine Learning?

- “Learning is any process by which a system improves performance from experience” – Herbert Simon

- Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

- Improve their performance *P*
- at some task *T*
- with experience *E*

A well-defined learning task is given by <*P, T, E*>.

What is Machine Learning?

Machine Learning is the study of algorithms that

- Improve their performance *P*
- at some task *T*
- with experience *E*

A well-defined learning task is given by <*P, T, E*>*.*

Experience: data-driven task, thus statistics, probability

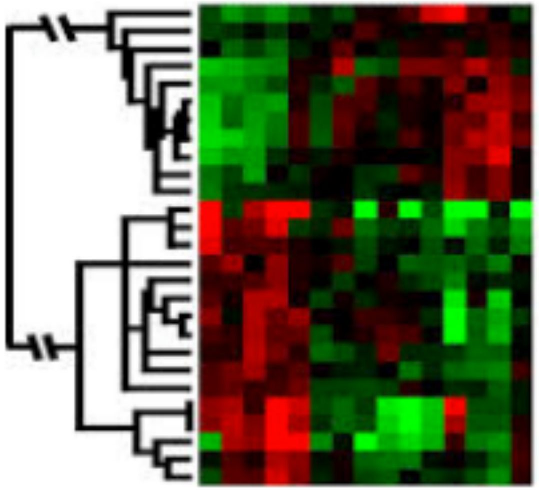
Example: use height and weight to predict gender

When do we use machine learning?

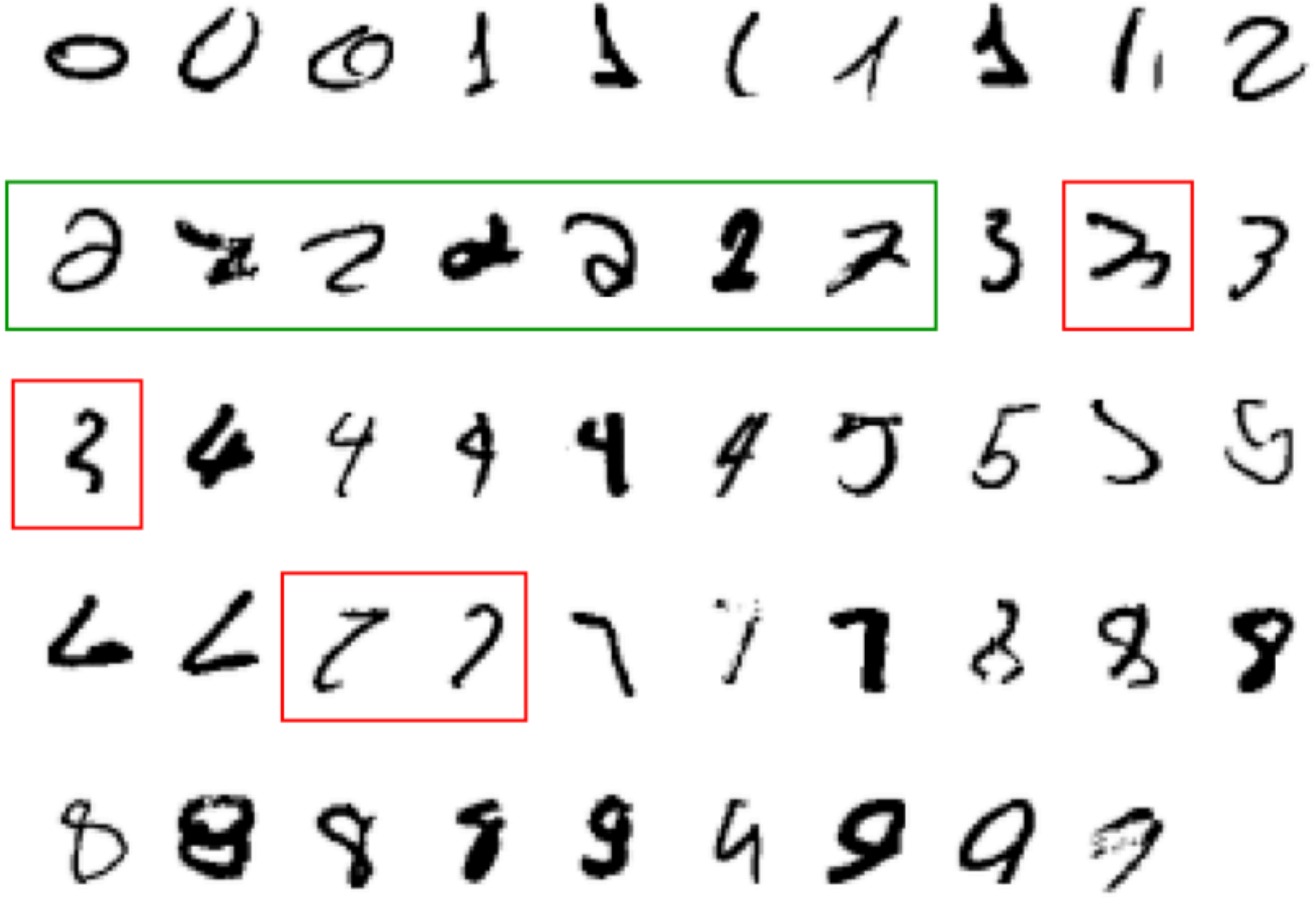
ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



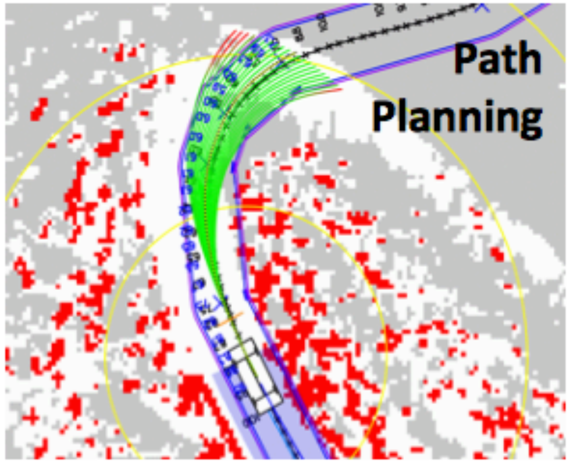
A task that requires machine learning

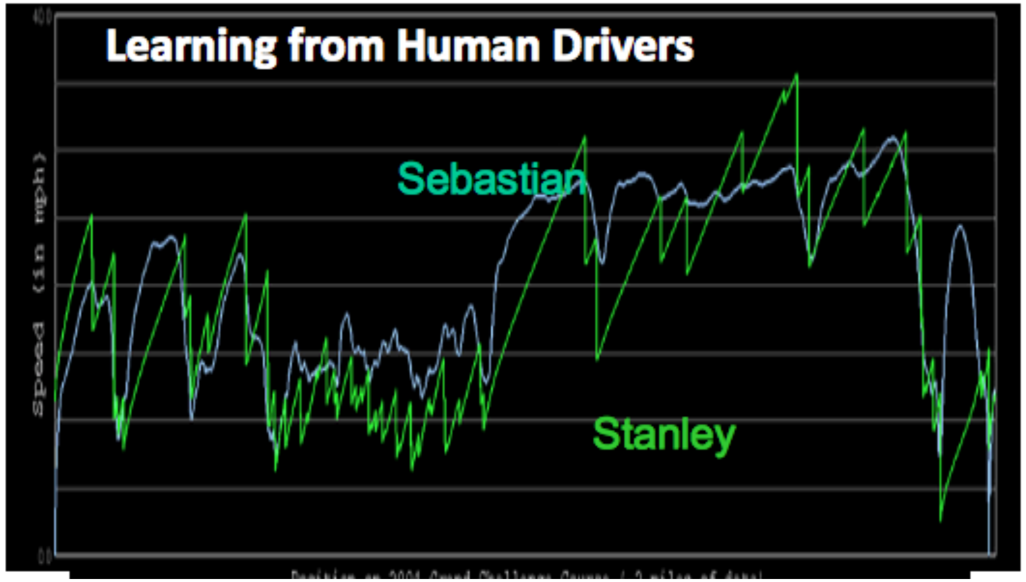


What makes a hand drawing be 2?

Slide credit: Geoffrey Hinton

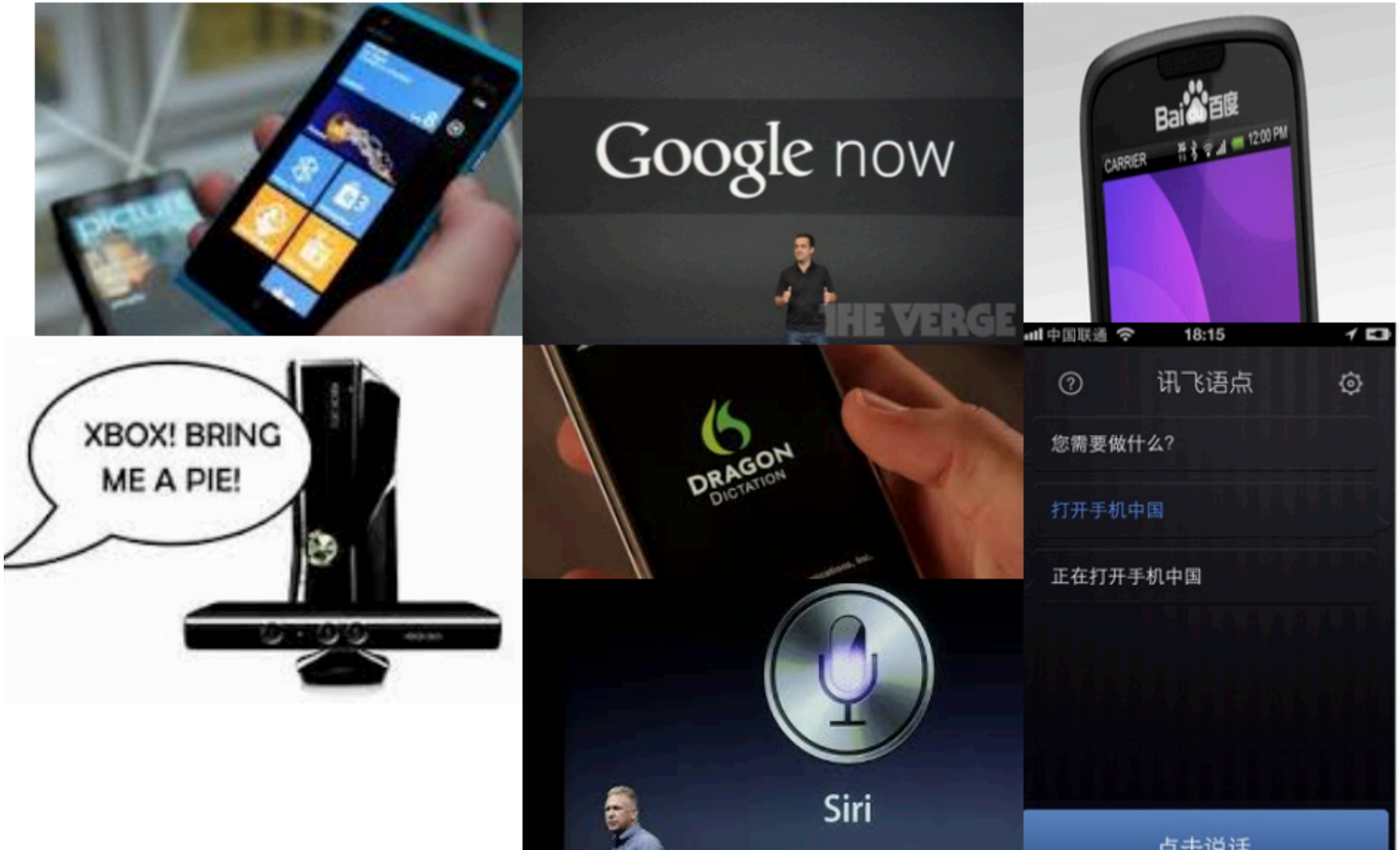
Modern machine learning: Autonomous cars





Modern machine learning: Scene Labeling

Modern machine learning: Speech Recognition



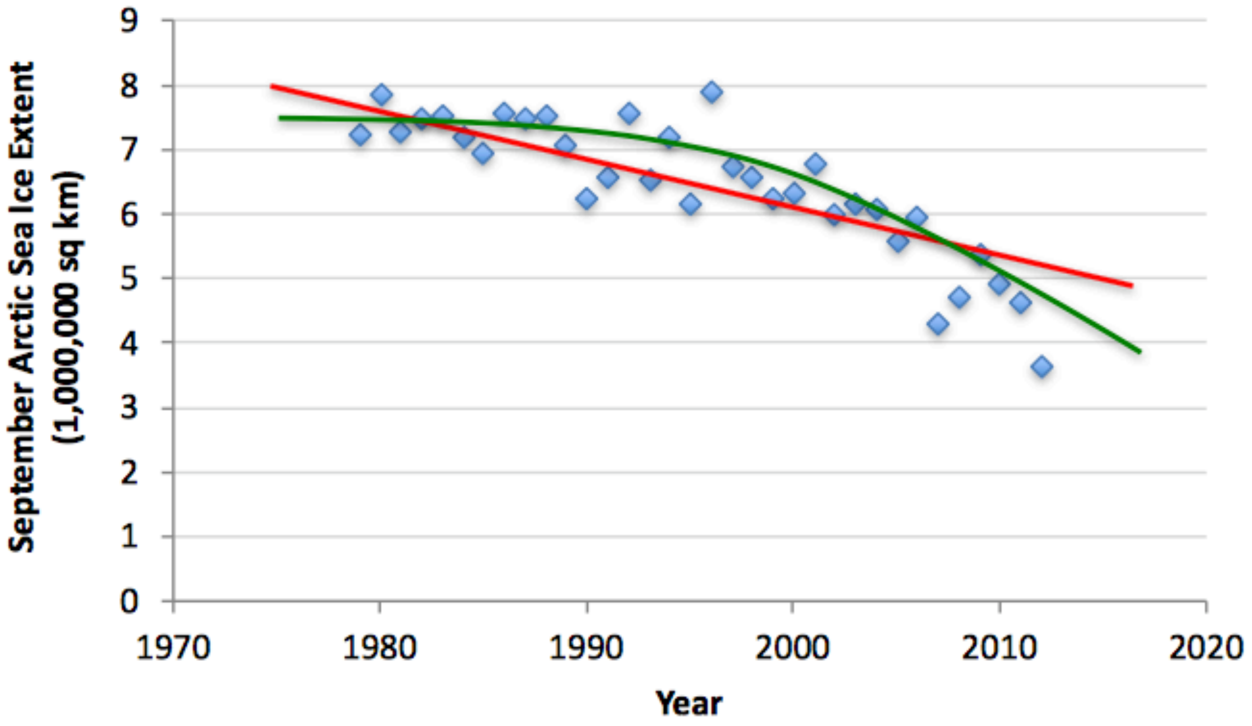
2. Types of Machine Learning

Types of Learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

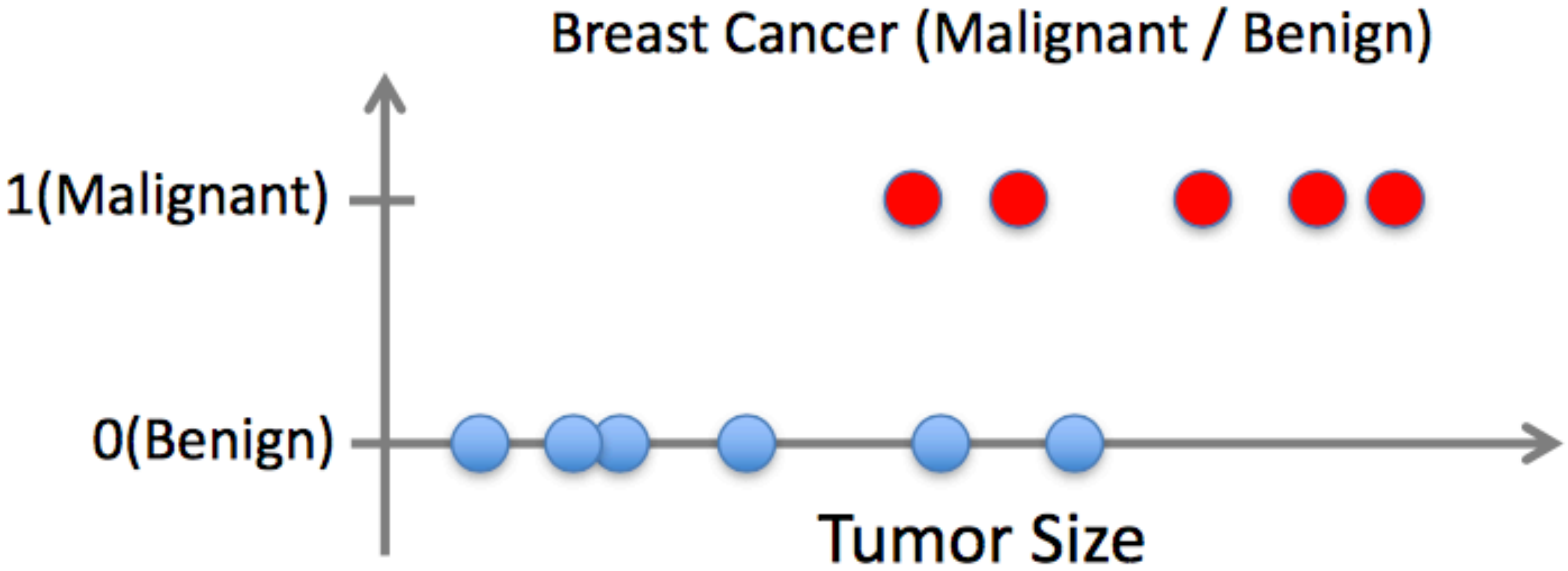
Supervised Learning: Regression

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function *f(x)* to predict *y* given *x*
  - *y* is real-valued == regression



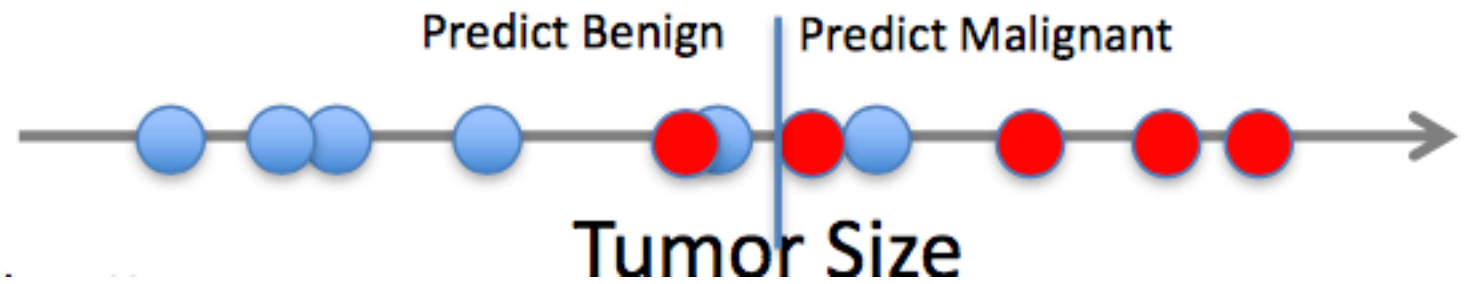
Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function *f(x)* to predict *y* given *x*
    - *y* is categorical == regression



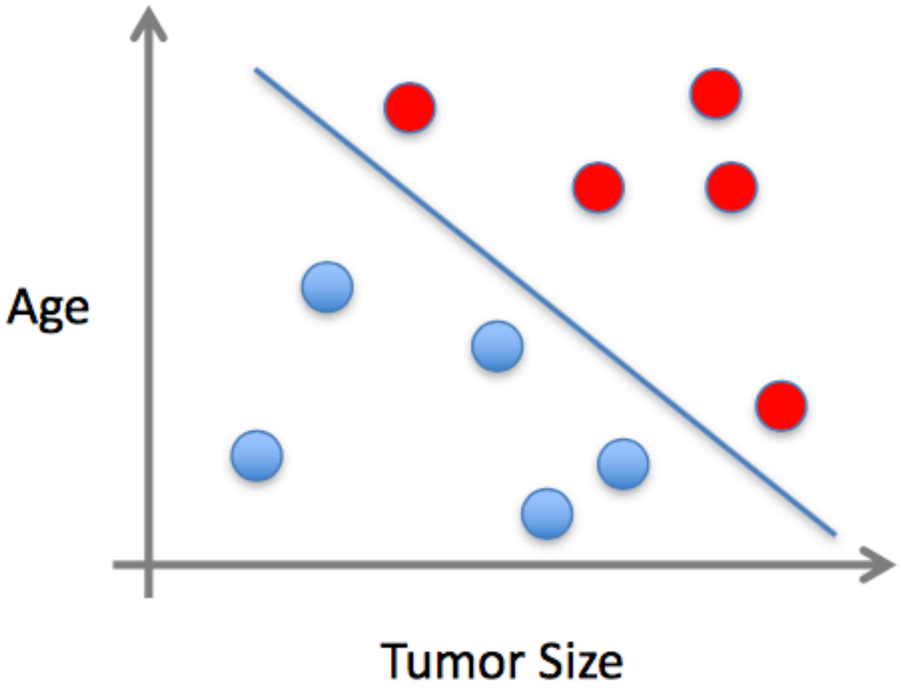
Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$
- Learn a function *f(x)* to predict *y* given *x*
  - *y* is categorical == regression



Supervised Learning

- Value *x* can be multi-dimensional.
    - Each dimension corresponds to an attribute



- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

Types of Learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

We will cover later in the class

3. Decision Trees

A learning problem: predict fuel efficiency

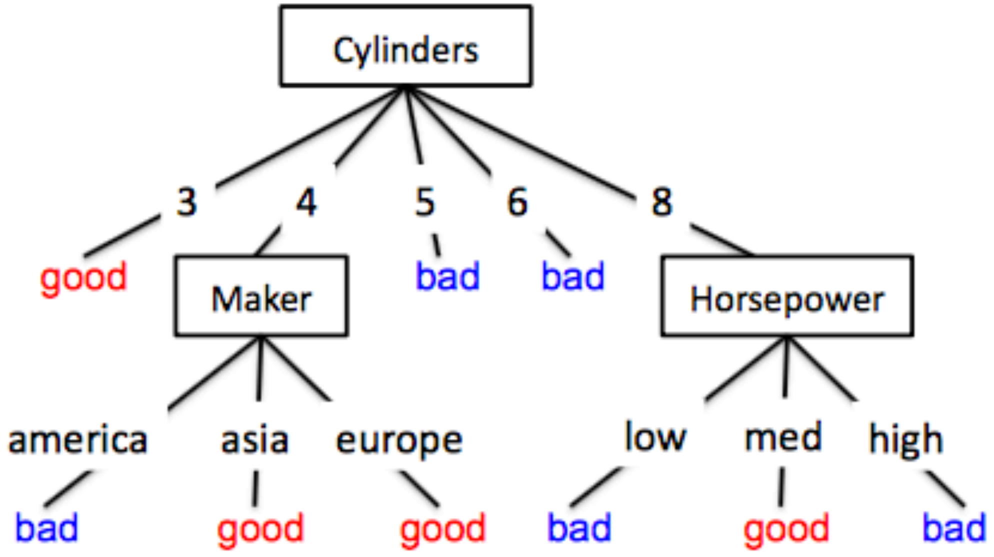
- 40 data points
- Goal: predict MPG
- Need to find: $f : X \rightarrow Y$
- Discrete data (for now)

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

Y X

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



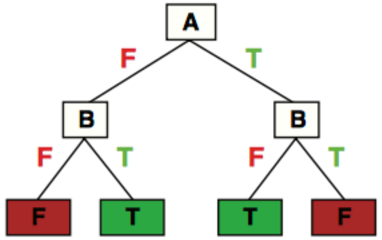
Human interpretable!

Informal
A hypothesis is a certain function that we believe (or hope) is similar to the true function, the *target function* that we want to model.

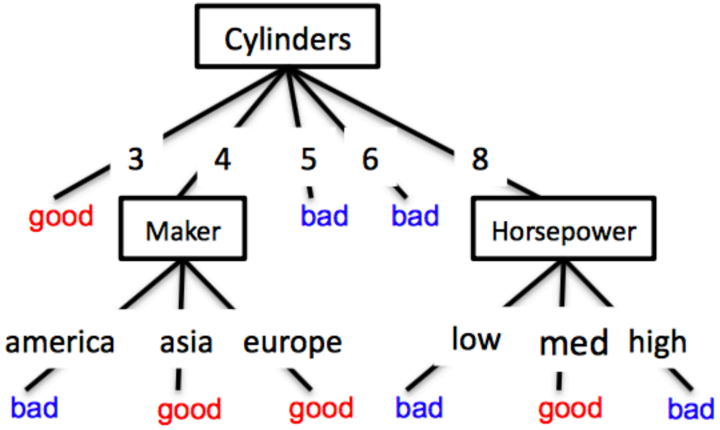
What functions can Decision Trees represent?

- 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)

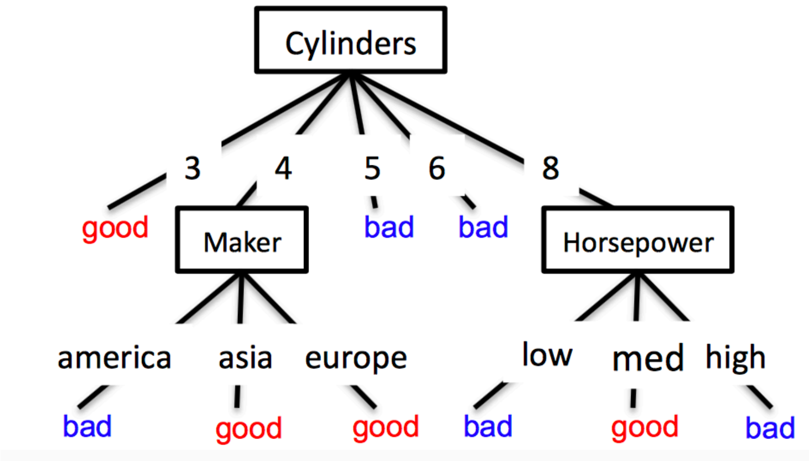


cyl=3 ∨ (cyl=4 ∧ (maker=asia ∨ maker=europe)) ∨ …

Space of possible decision trees

- How will we choose the best one?
- Lets first look at how to split nodes, then consider how to find the best tree

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?

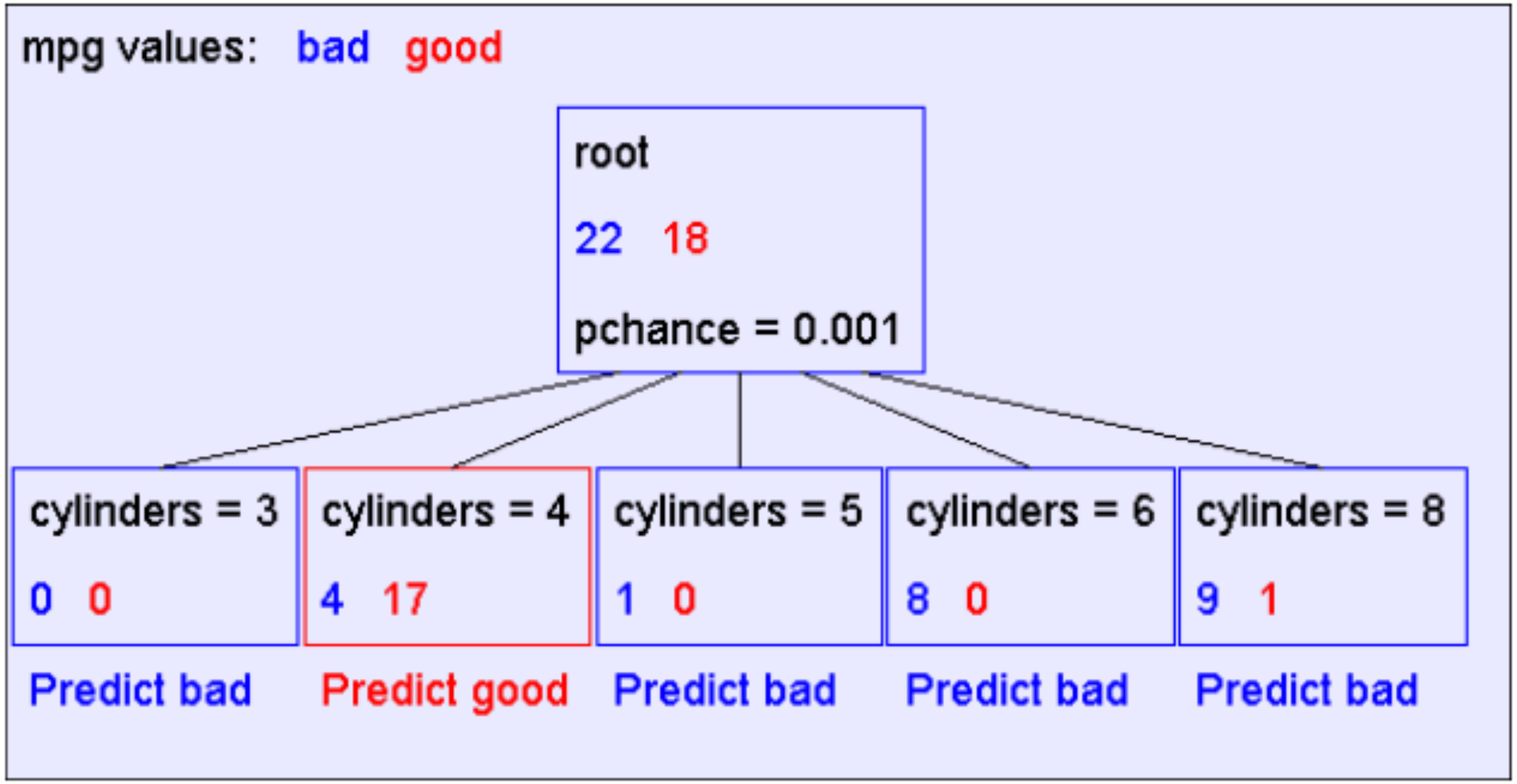
- Always predict mpg = bad
  - We just take the majority class

- Is this a good tree?
  - We need to evaluate its performance

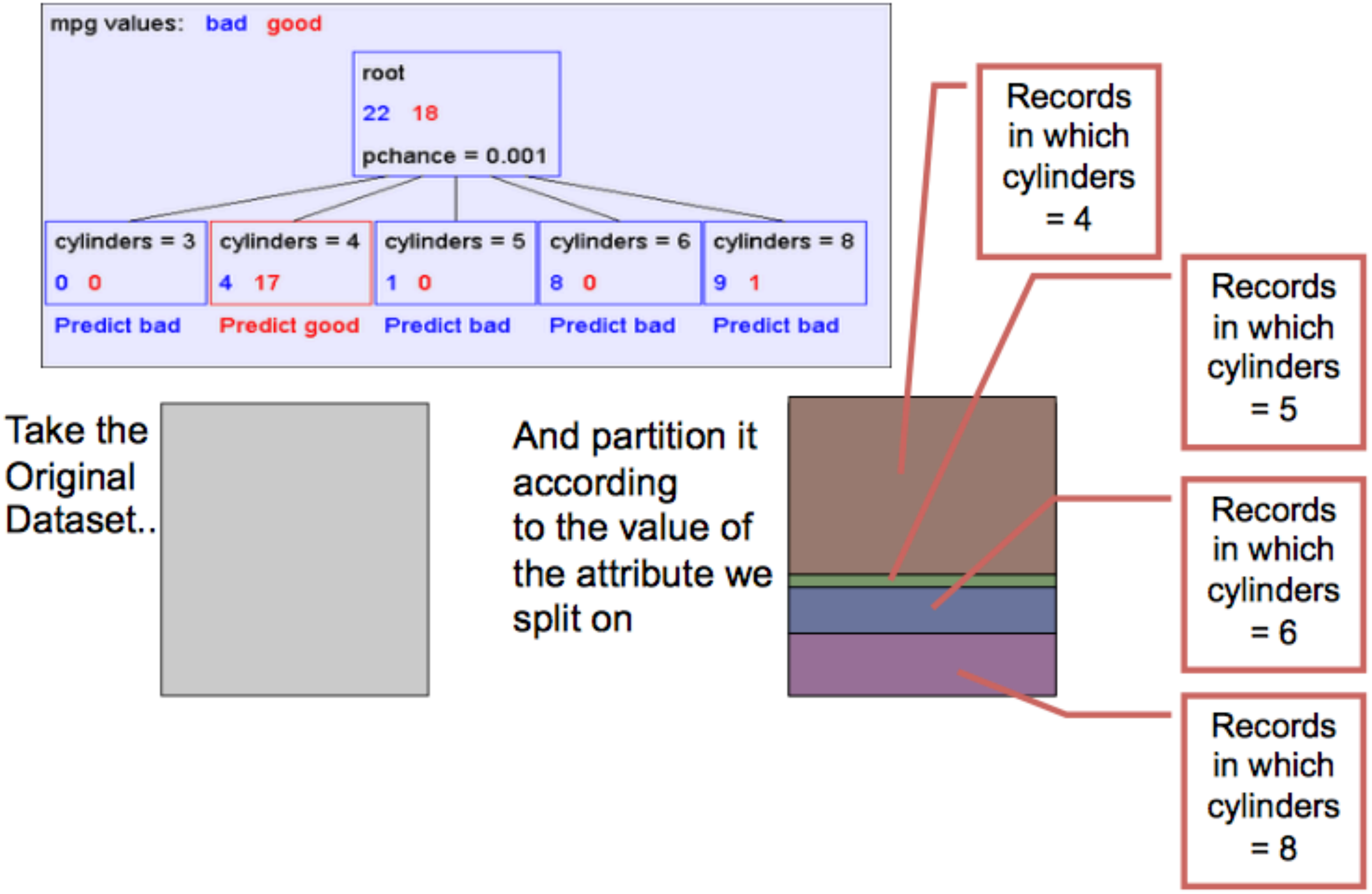
- Performance: We are correct on 22 examples and incorrect on 18 examples

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

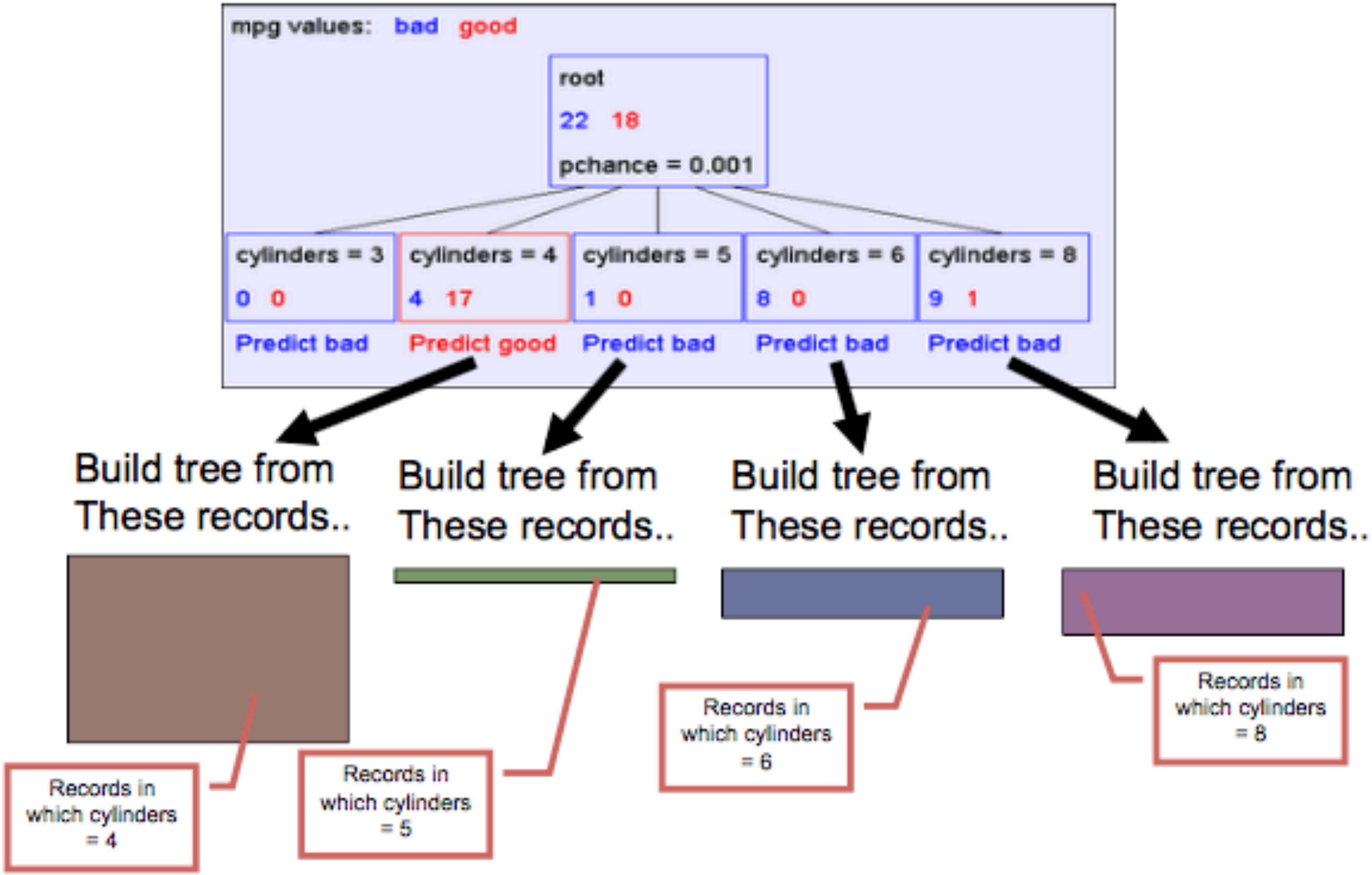
A decision stump



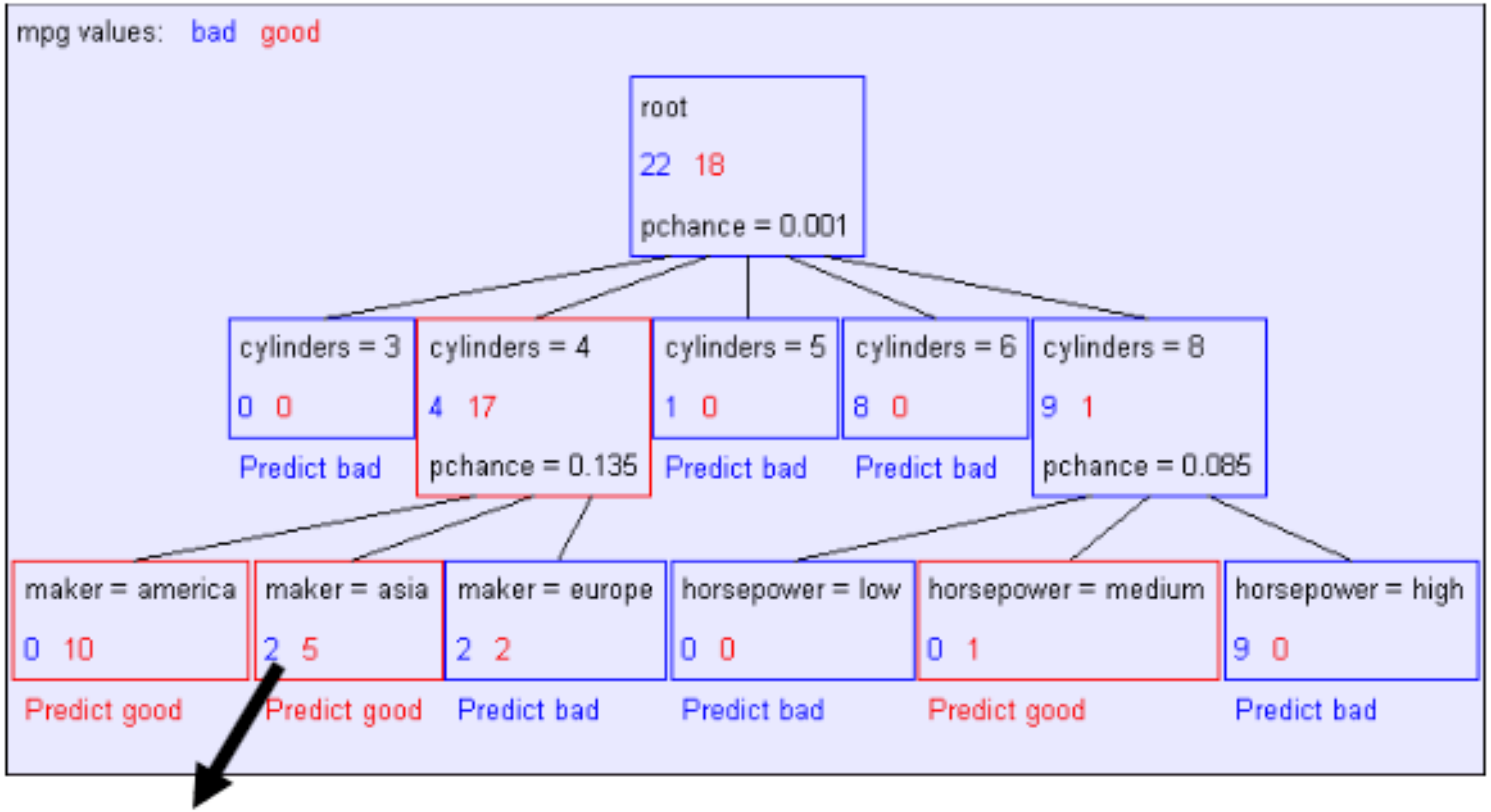
Recursive step



Recursive step



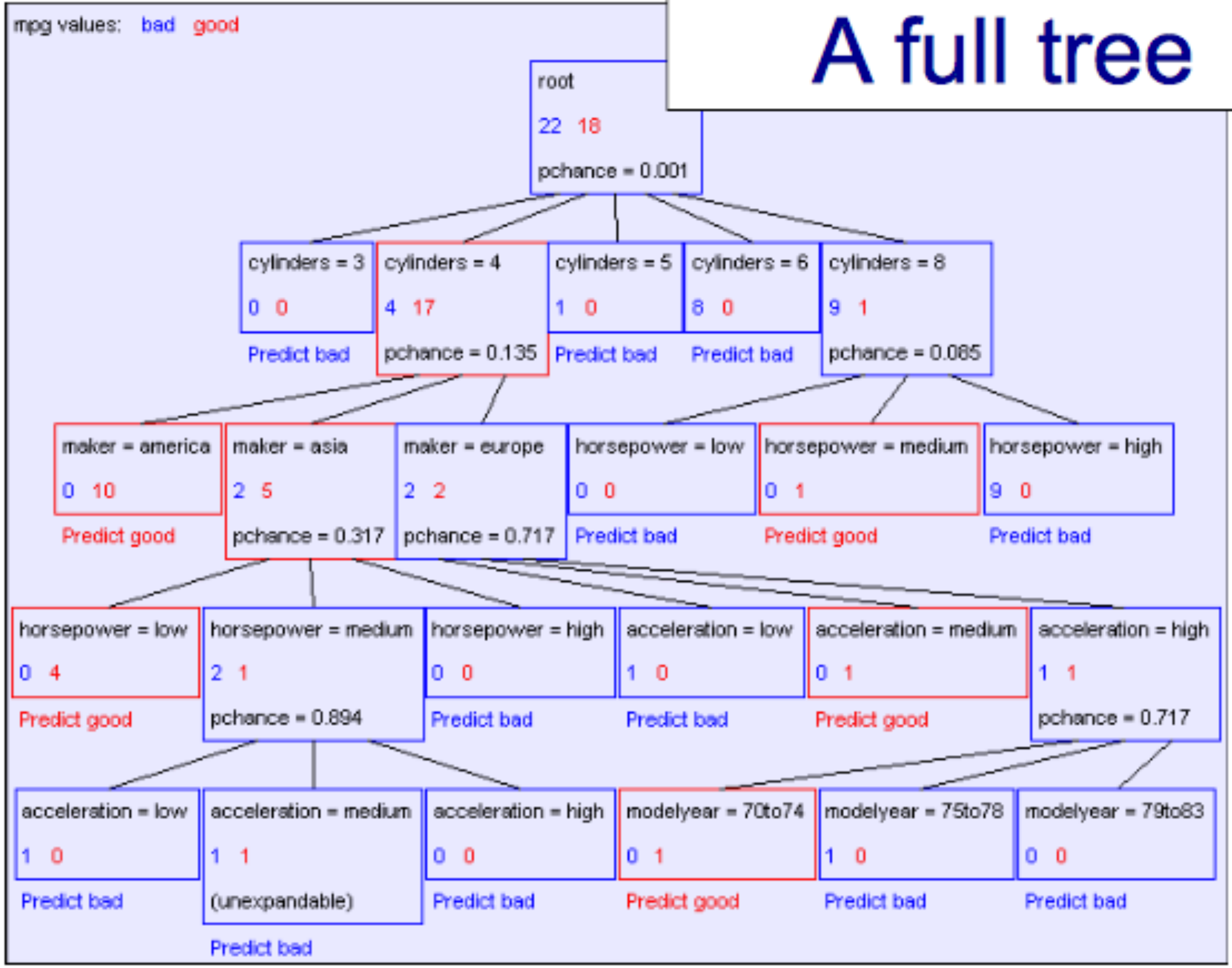
Second level of tree



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)

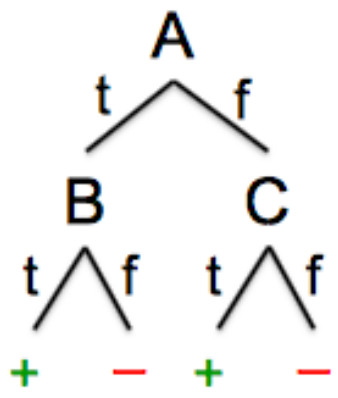
A full tree

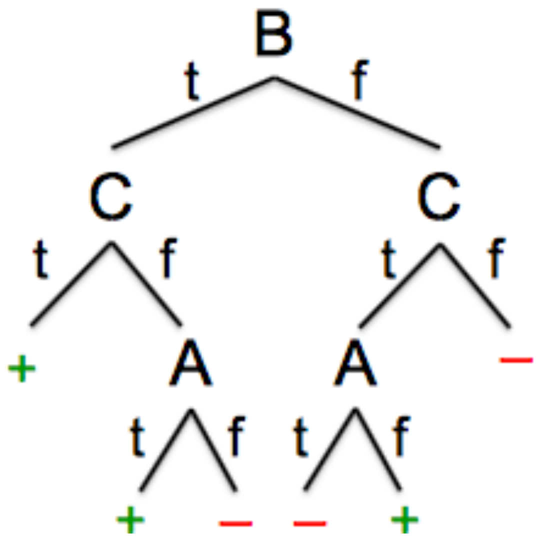


mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance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





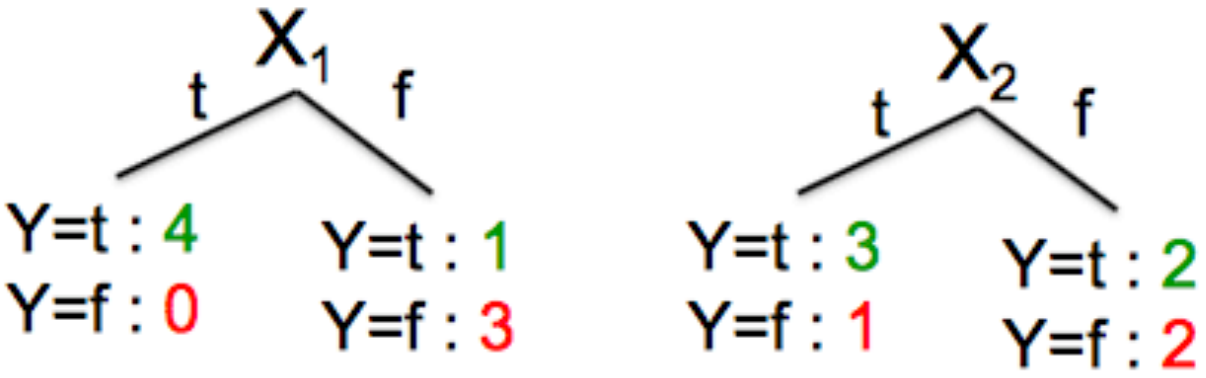
- Which tree do we prefer?

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

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

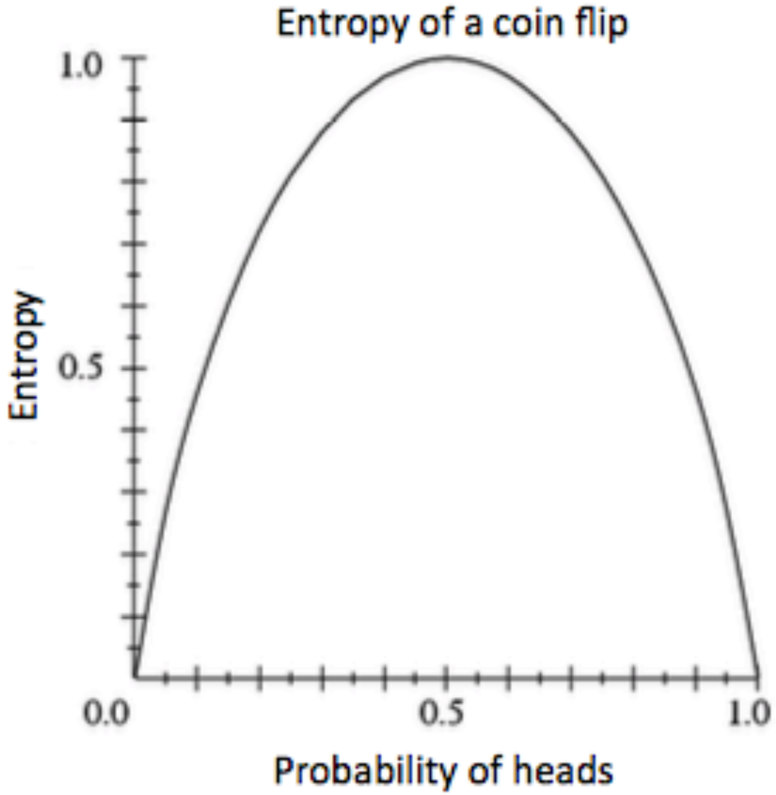
Entropy

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!

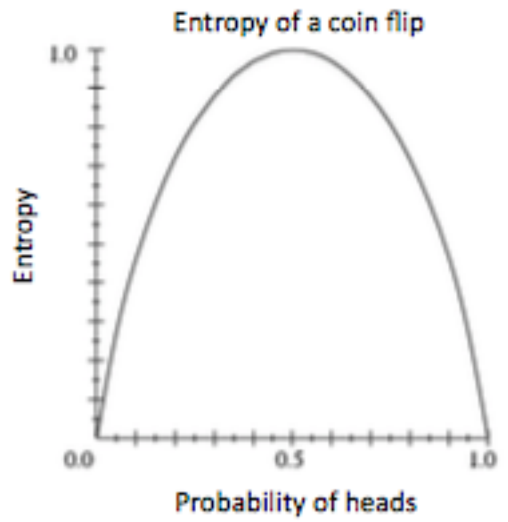
Information Theory interpretation:
H(Y) is the expected number of bits needed to encode a randomly drawn value of *Y* (under most efficient code)



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

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

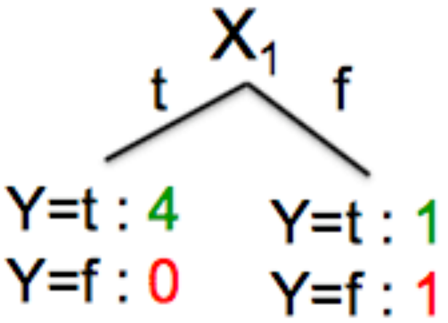
Conditional Entropy $H(Y|X)$ of a random variable Y conditioned on a random variable X

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

Example:

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

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



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

$$H(Y|X_1) = -\ 4/6\ (1 \log_2 1 + 0 \log_2 0)$$
$$-\ 2/6\ (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$$
$$= 2/6$$

Information gain

- Decrease in entropy (uncertainty) after splitting

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

In our running example:

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

$IG(X_1) > 0 \rightarrow$ 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

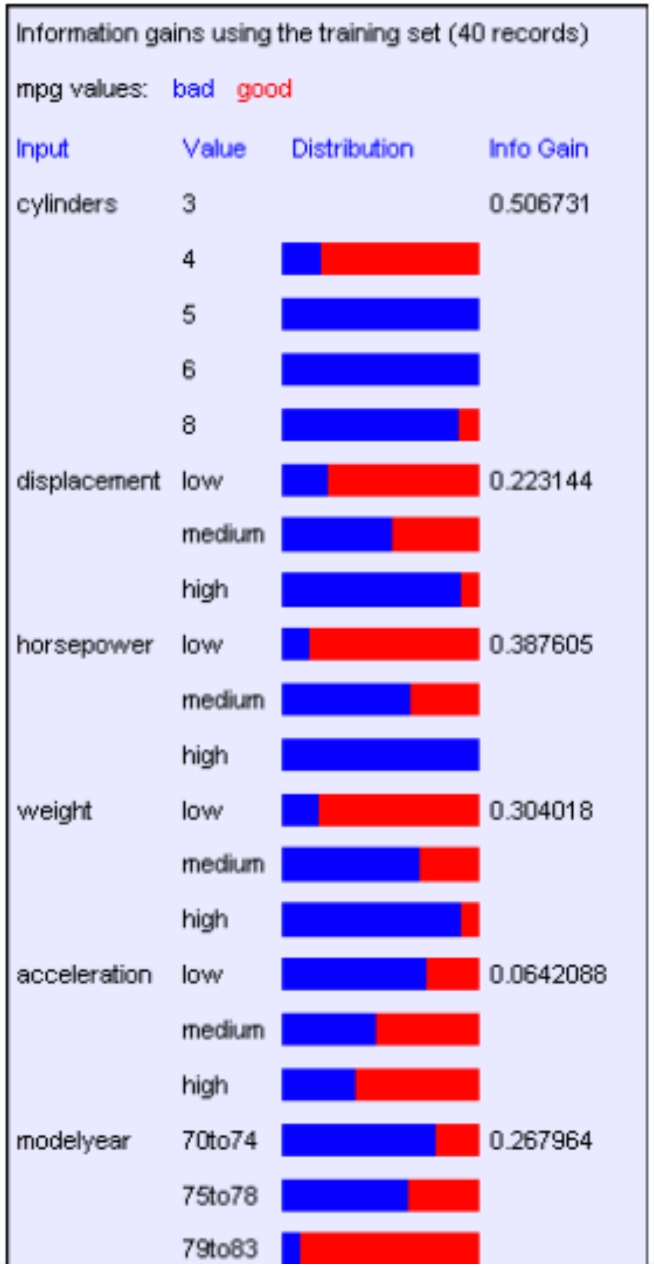
- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute:

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

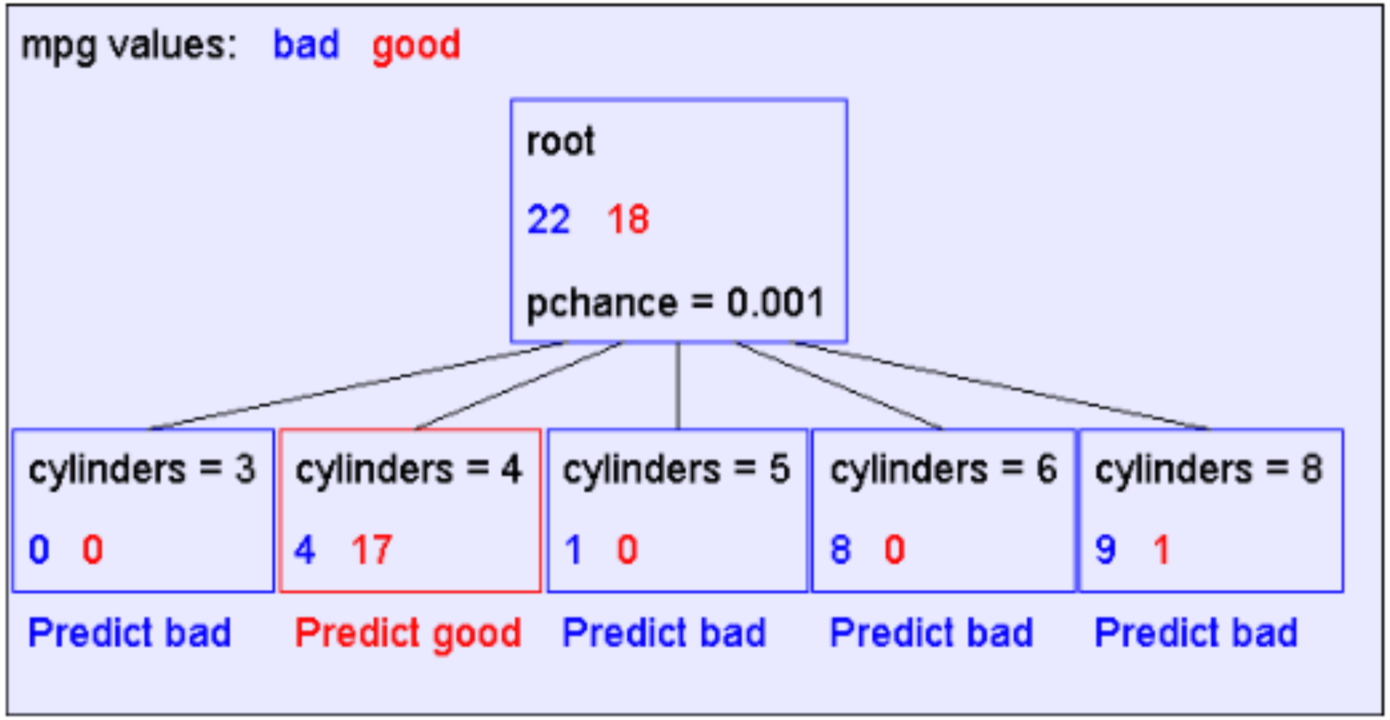
- Recurse

Suppose we want to predict MPG

Look at all the information gains…

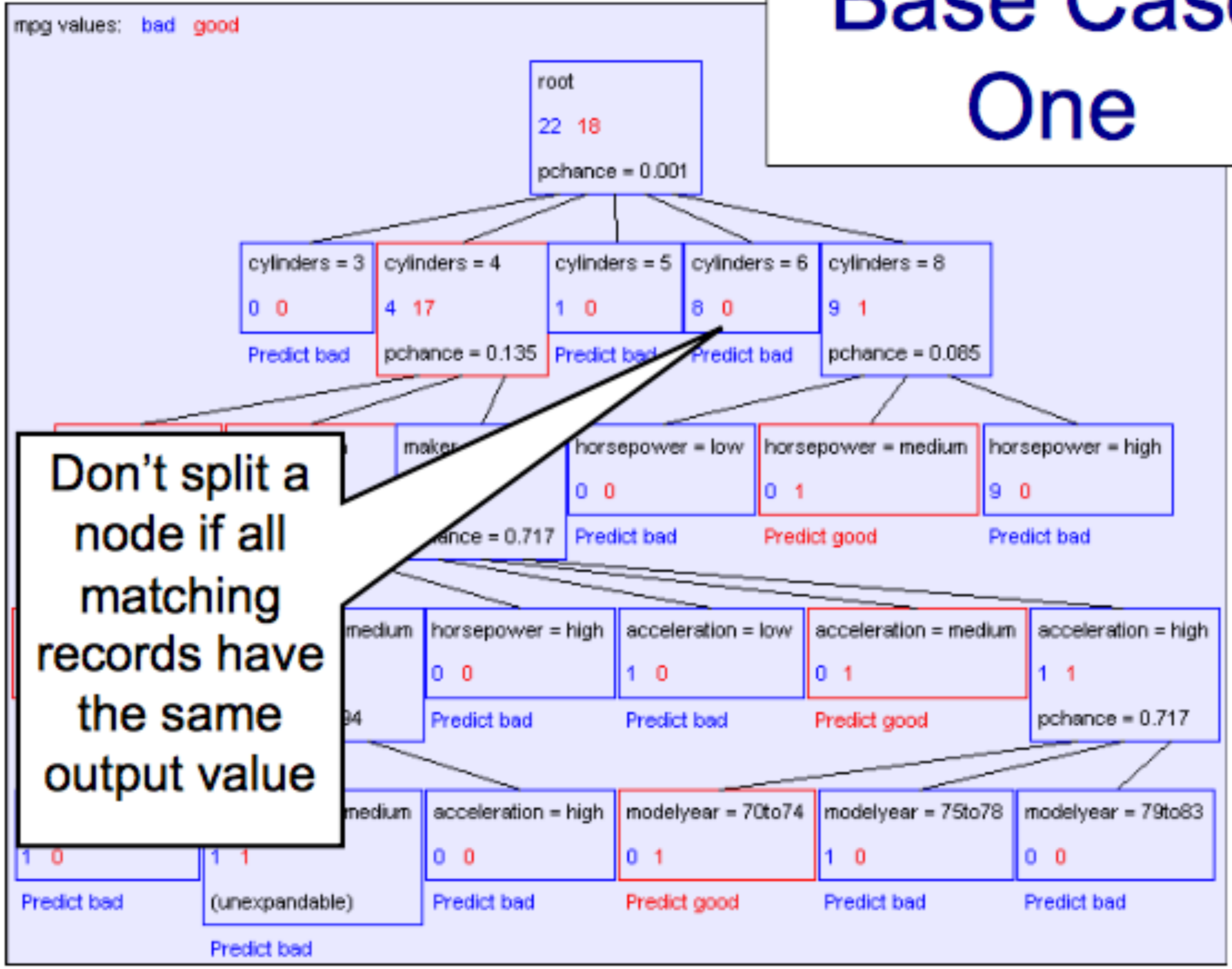


A decision stump

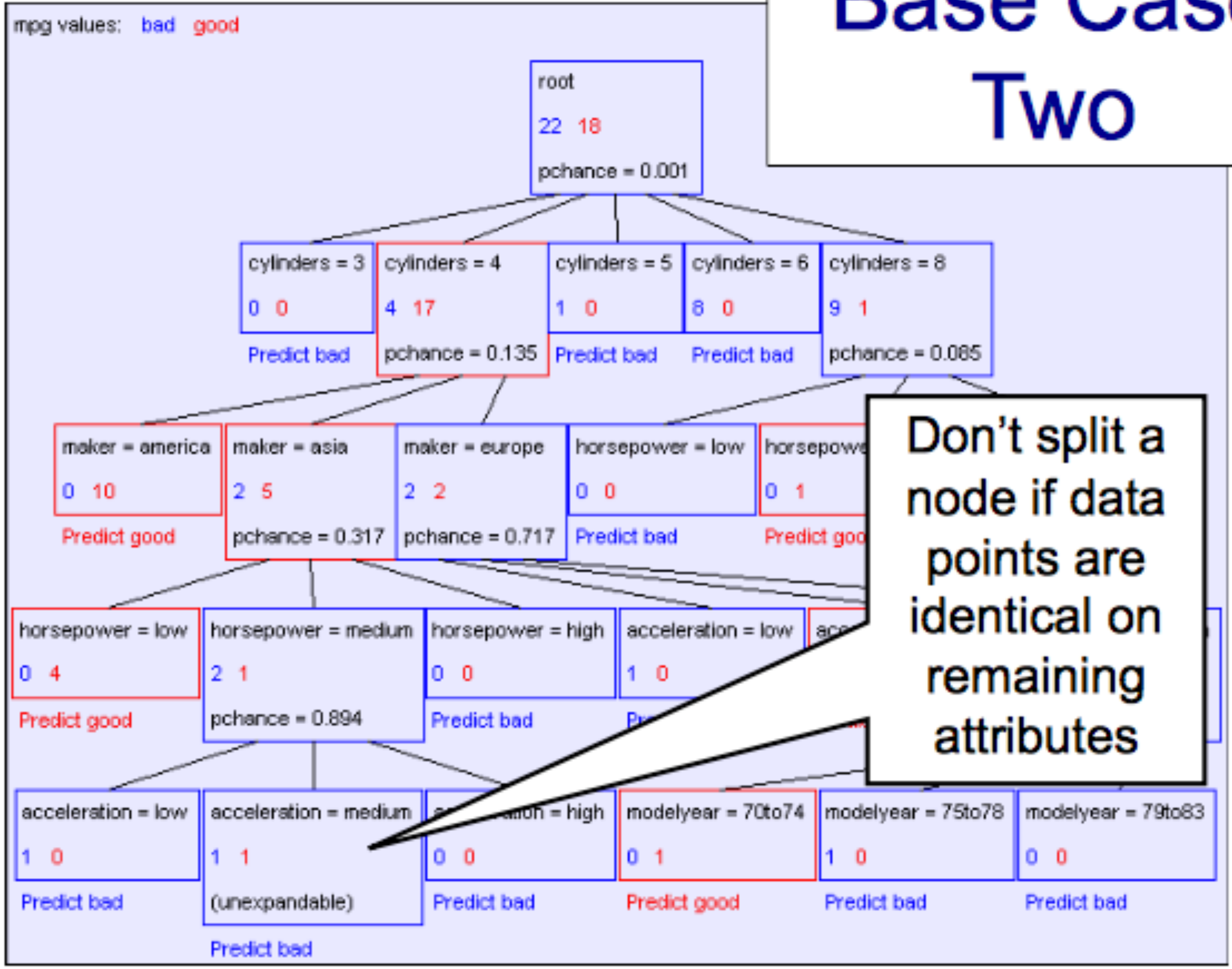


First split looks good! But, when do we stop?

Base Case One

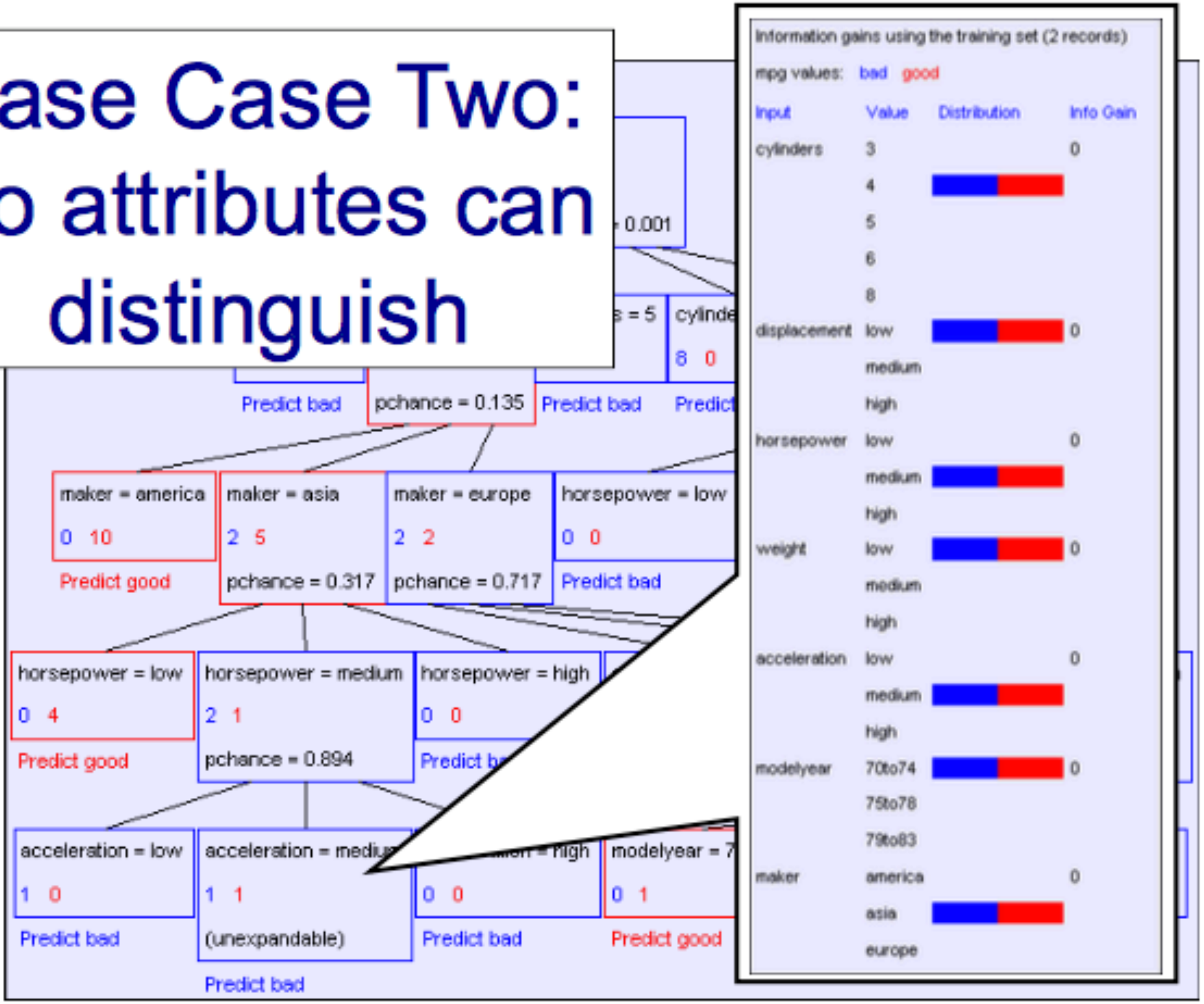


Base Case Two



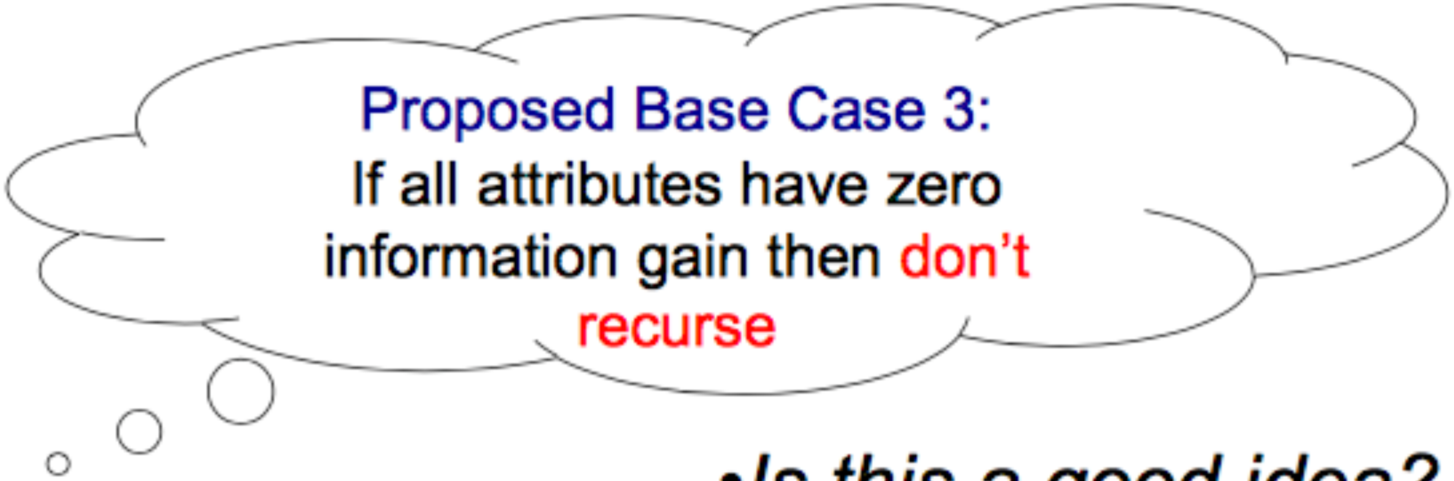
Don't split a node if data points are identical on remaining attributes

Base Case Two: No attributes can distinguish



Base Cases: An Idea

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



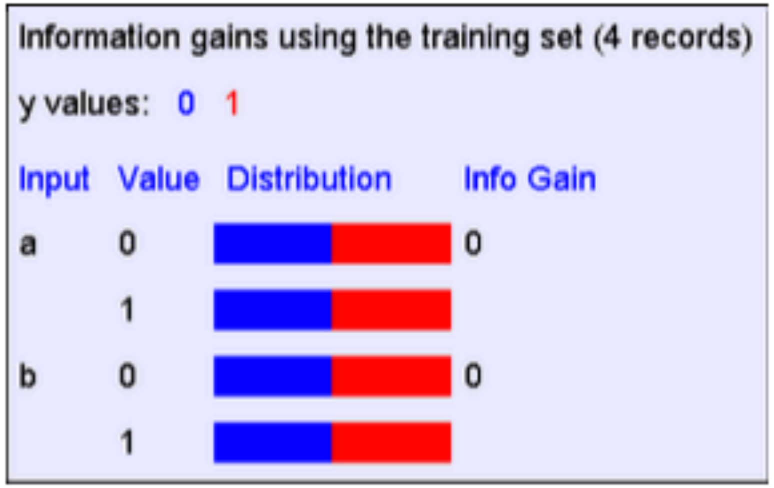
- *Is this a good idea?*

The problem with Base Case 3

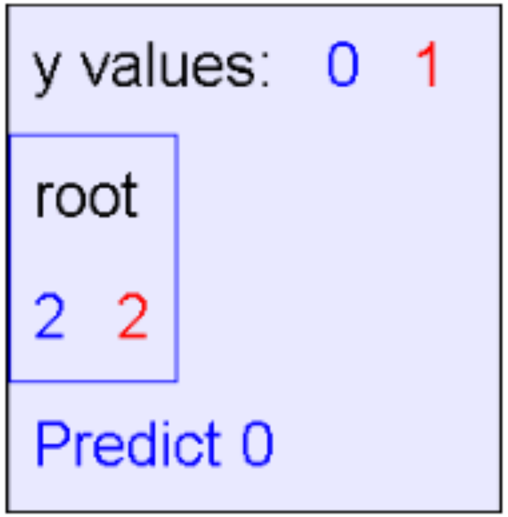
y = a XOR b

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

The information gains:



The resulting decision tree:



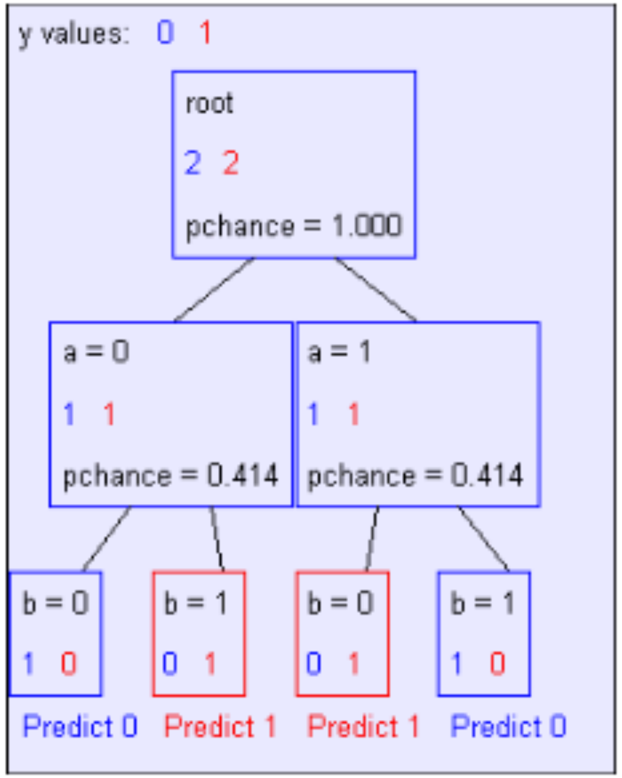
If we omit Base Case 3

y = a XOR b

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Is it OK to omit Base Case 3?

The resulting decision tree:



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

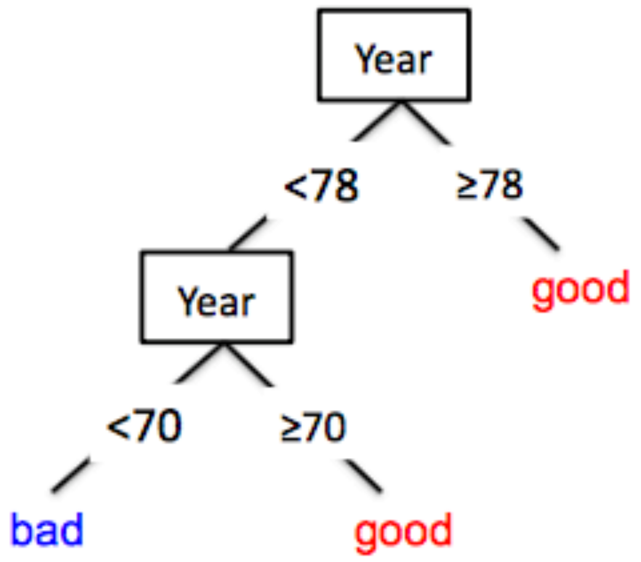
    BuildTree(DS_i,*Output*)

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

From categorical to real-valued attributes

- **Binary tree:** split on attribute X at value t
  - One branch: X < t
  - Other branch: X ≥ t

- Requires small change
  - Allow repeated splits on same variable
  - How does this compare to "branch on each value" approach?



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
- Presented for classification but can be used for regression and density estimation too
- Decision trees will overfit!!!
  - We will see the definition of overfitting and related concepts later in class.