

# DS 4400

## Machine Learning and Data Mining I Spring 2021

Alina Oprea

Associate Professor

Khoury College of Computer Science

Northeastern University

March 4 2021

# Announcements

- Project proposal is due today
  - One page
  - Team of 2
  - Problem statement
  - Dataset description
  - ML Models and metrics for evaluation
  - Related work
- Homework 3 is due on Monday, March 8

# Outline

- Decision Trees
  - Entropy, information gain
  - Learning decision trees
  - Stopping conditions
- Ensemble models
  - Majority vote over multiple models

# Sample Dataset

- Columns denote features  $X_i$
- Rows denote labeled instances  $\langle x_i, y_i \rangle$
- Class label denotes whether a tennis game was played

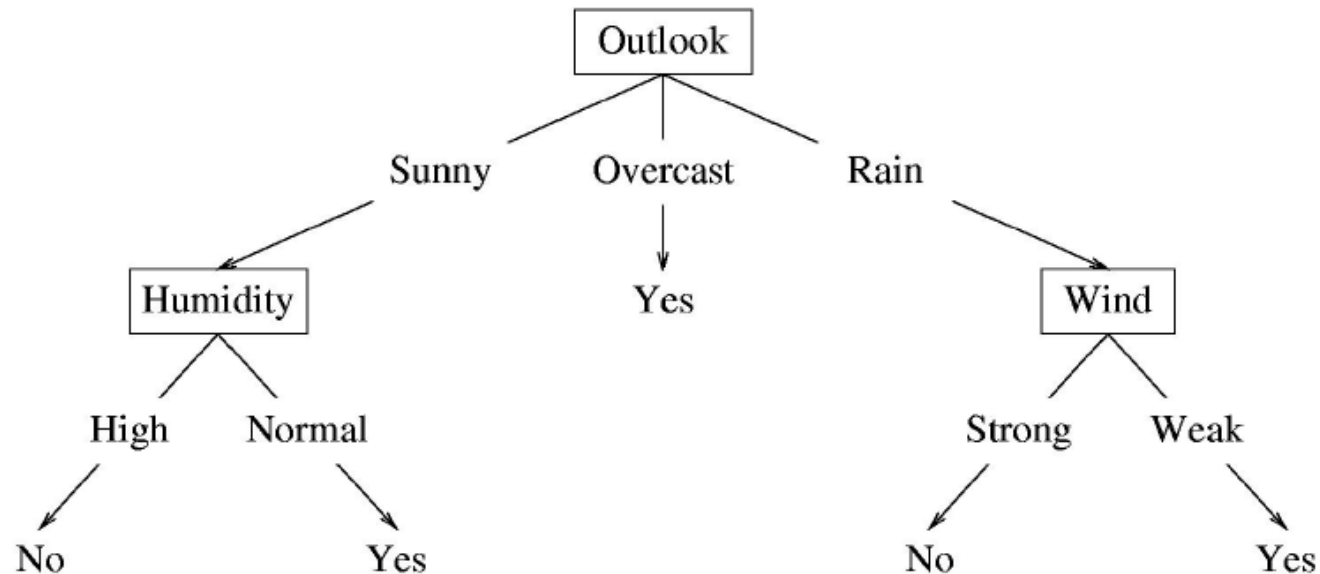
Predictors				Response
Outlook	Temperature	Humidity	Wind	Class
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

$\langle x_i, y_i \rangle$

Categorical  
data

# Decision Tree

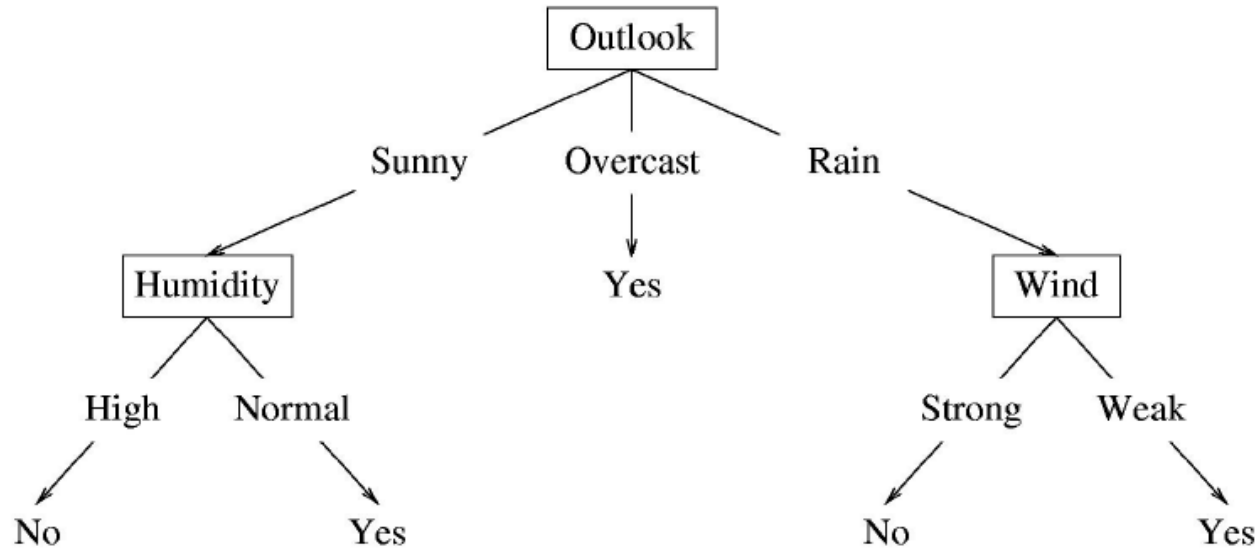
- A possible decision tree for the data:



- Each internal node: test one attribute  $X_i$
- Each branch from a node: selects one value for  $X_i$
- Each leaf node: predict  $Y$  (or  $p(Y \mid x \in \text{leaf})$  )

# Decision Tree

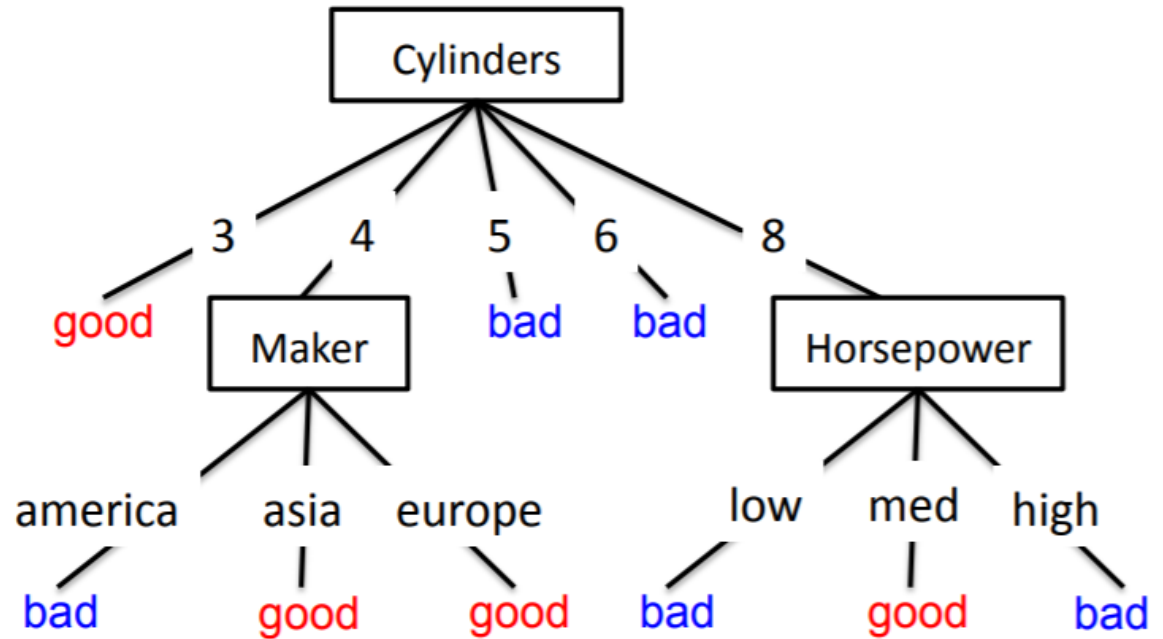
- A possible decision tree for the data:



- What prediction would we make for  
<outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

# Interpretability

- 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  $y$



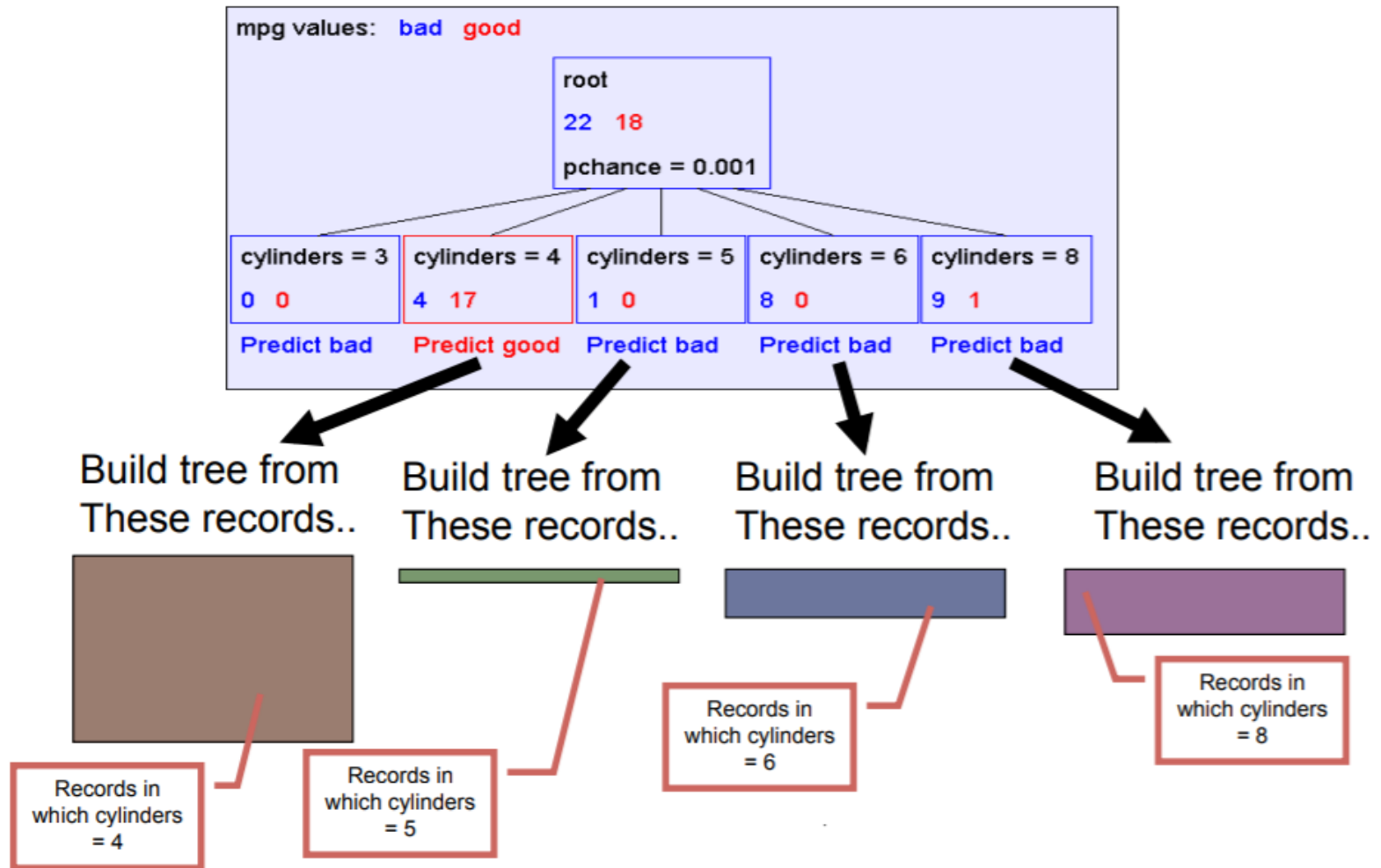
Human interpretable!

# Learning Decision Trees

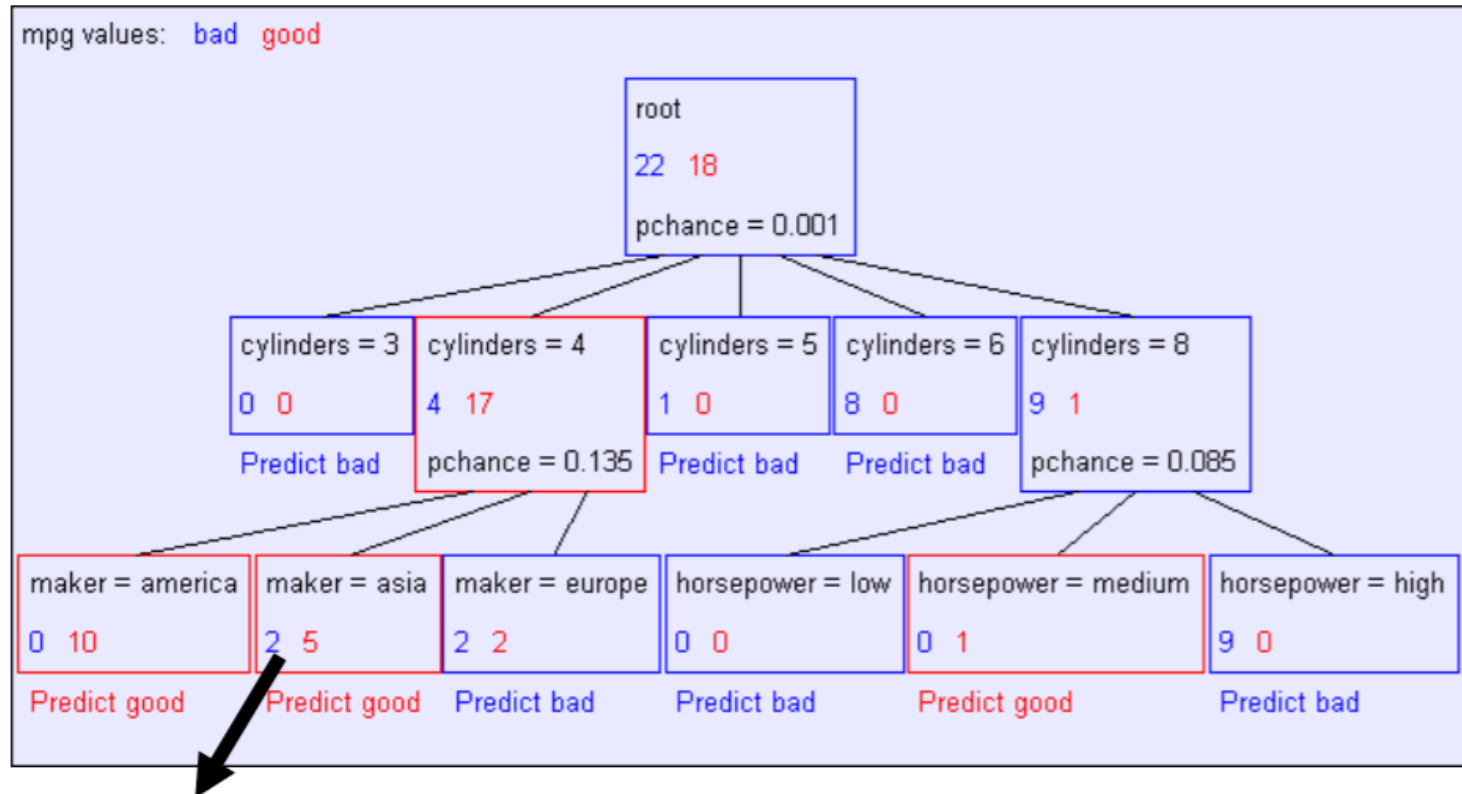
- Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest '76]
- Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on **next best attribute (feature)**
  - Recurse



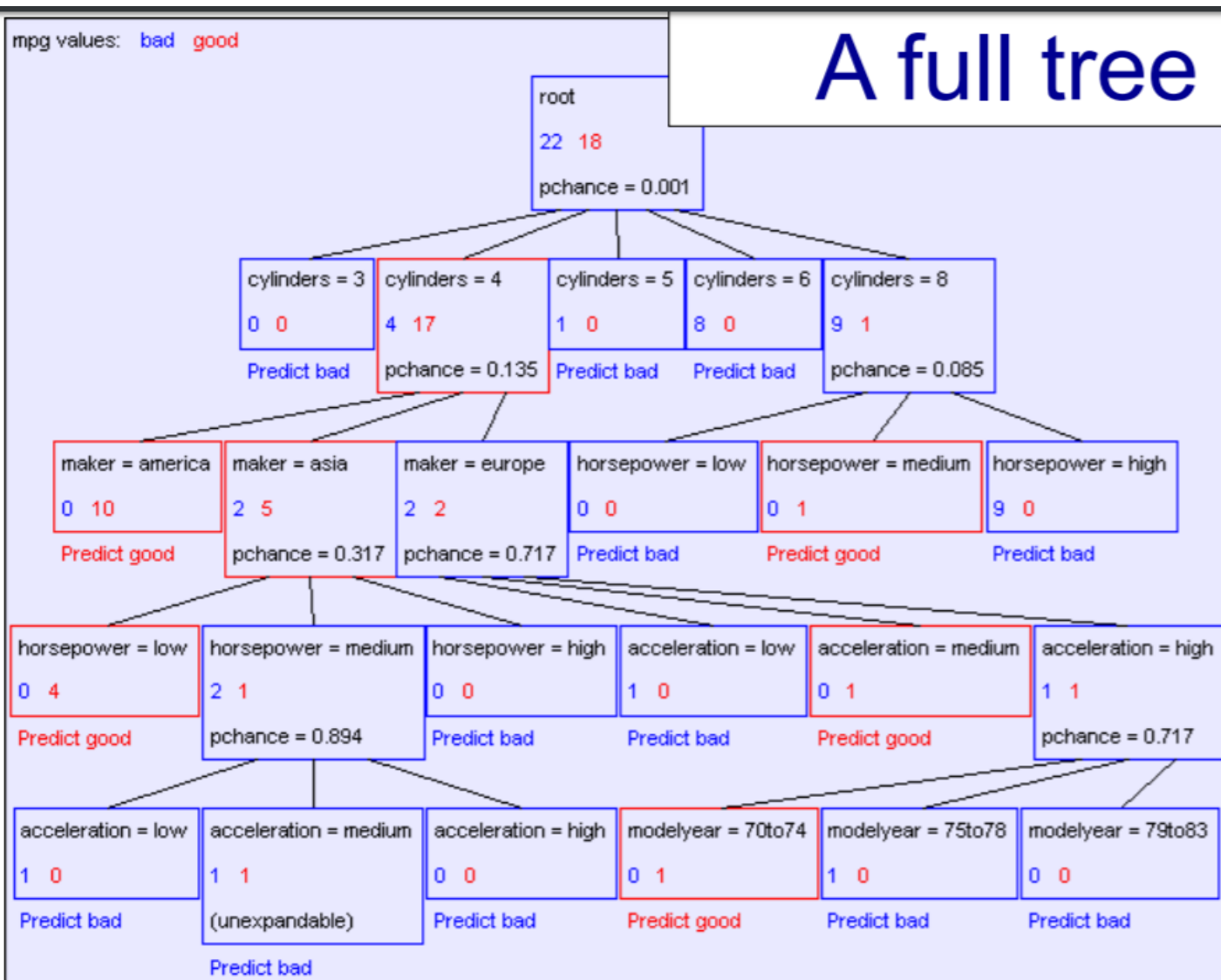
# Key Idea: Use Recursion Greedily



# Second Level

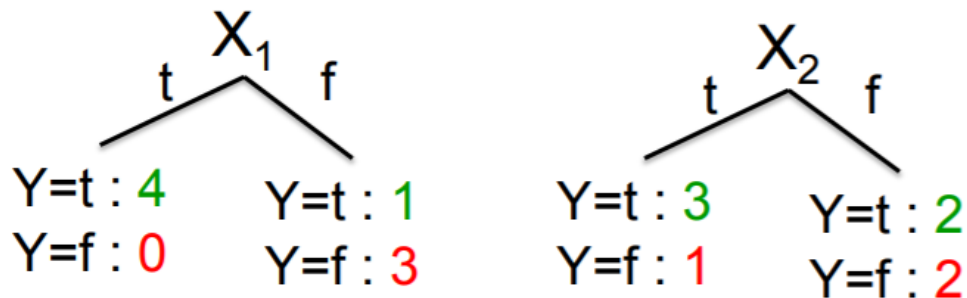


# Full Tree



# Splitting

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



**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

Use entropy-based measure (Information Gain)

# Entropy

Suppose  $X$  can have one of  $m$  values...  $V_1, V_2, \dots, V_m$

$P(X=V_1) = p_1$	$P(X=V_2) = p_2$	....	$P(X=V_m) = p_m$
------------------	------------------	------	------------------

What's the smallest possible number of bits, on average, per symbol, needed to transmit a stream of symbols drawn from  $X$ 's distribution? It's

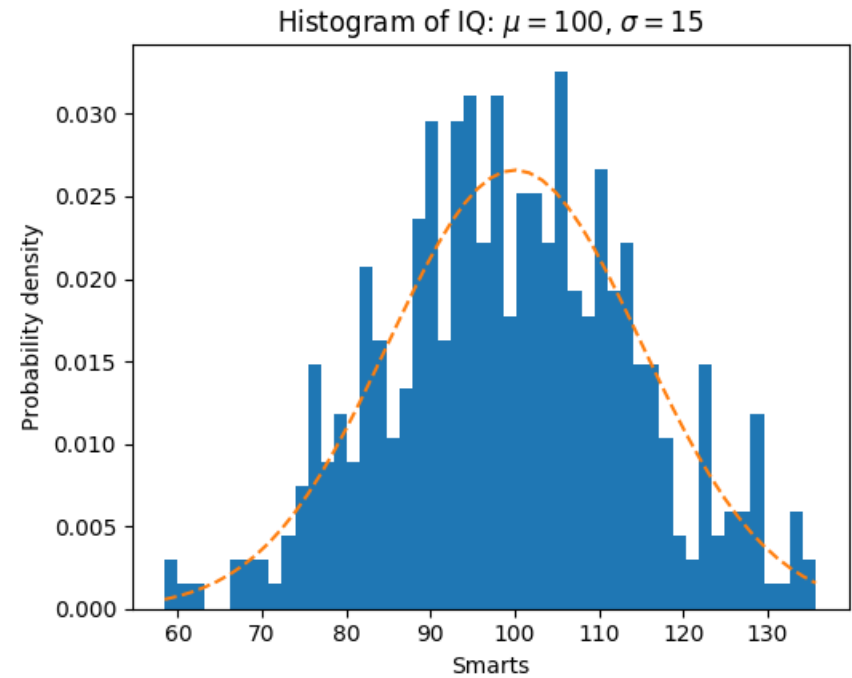
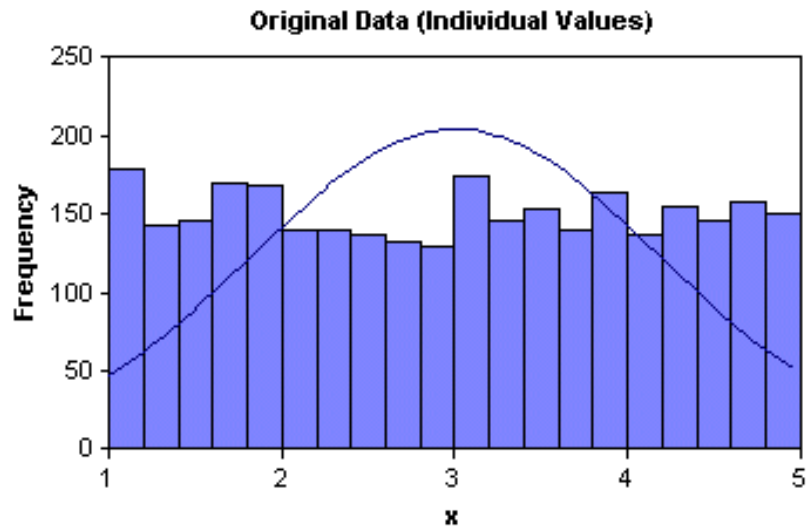
$$\begin{aligned} H(X) &= -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \dots - p_m \log_2 p_m \\ &= -\sum_{j=1}^m p_j \log_2 p_j \end{aligned}$$

$H(X)$  = The entropy of  $X$

- "High Entropy" means  $X$  is from a uniform (boring) distribution
- "Low Entropy" means  $X$  is from varied (peaks and valleys) distribution

# High/Low Entropy

Which distribution has high entropy?



# Conditional Entropy

**Suppose I'm trying to predict output Y and I have input X**

**X = College Major**

**Y = Likes "Gladiator"**

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

**Let's assume this reflects the true probabilities**

**E.G. From this data we estimate**

- $P(\text{LikeG} = \text{Yes}) = 0.5$
- $P(\text{Major} = \text{Math} \ \& \ \text{LikeG} = \text{No}) = 0.25$
- $P(\text{Major} = \text{Math}) = 0.5$
- $P(\text{LikeG} = \text{Yes} \mid \text{Major} = \text{History}) = 0$

**Note:**

- $H(X) = 1.5$
- $H(Y) = 1$

# Conditional Entropy

X = College Major

Y = Likes "Gladiator"

**Definition of Specific Conditional Entropy:**

$H(Y|X=v)$  = **The entropy of  $Y$  among only those records in which  $X$  has value  $v$**

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

**Example:**

- $H(Y|X=Math) = 1$
- $H(Y|X=History) = 0$
- $H(Y|X=CS) = 0$



# Conditional Entropy

**X = College Major**

**Y = Likes "Gladiator"**

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

## Definition of Conditional Entropy:

$H(Y|X)$  = The average specific conditional entropy of  $Y$

= if you choose a record at random what will be the conditional entropy of  $Y$ , conditioned on that row's value of  $X$

= Expected number of bits to transmit  $Y$  if both sides will know the value of  $X$

$$= \sum_j \text{Prob}(X=v_j) H(Y | X = v_j)$$

# Conditional Entropy

X = College Major

Y = Likes "Gladiator"

## Definition of Conditional Entropy:

$H(Y|X)$  = The average conditional entropy of  $Y$

$$= \sum_j \text{Prob}(X=v_j) H(Y | X = v_j)$$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

## Example:

$v_j$	$\text{Prob}(X=v_j)$	$H(Y   X = v_j)$
Math	0.5	1
History	0.25	0
CS	0.25	0

$$H(Y|X) = 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$$

# Information Gain

**X = College Major**

**Y = Likes "Gladiator"**

**Definition of Information Gain:**

$IG(Y|X) =$  **I must transmit  $Y$ .**

**How many bits on average would it save me if both ends of the line knew  $X$ ?**

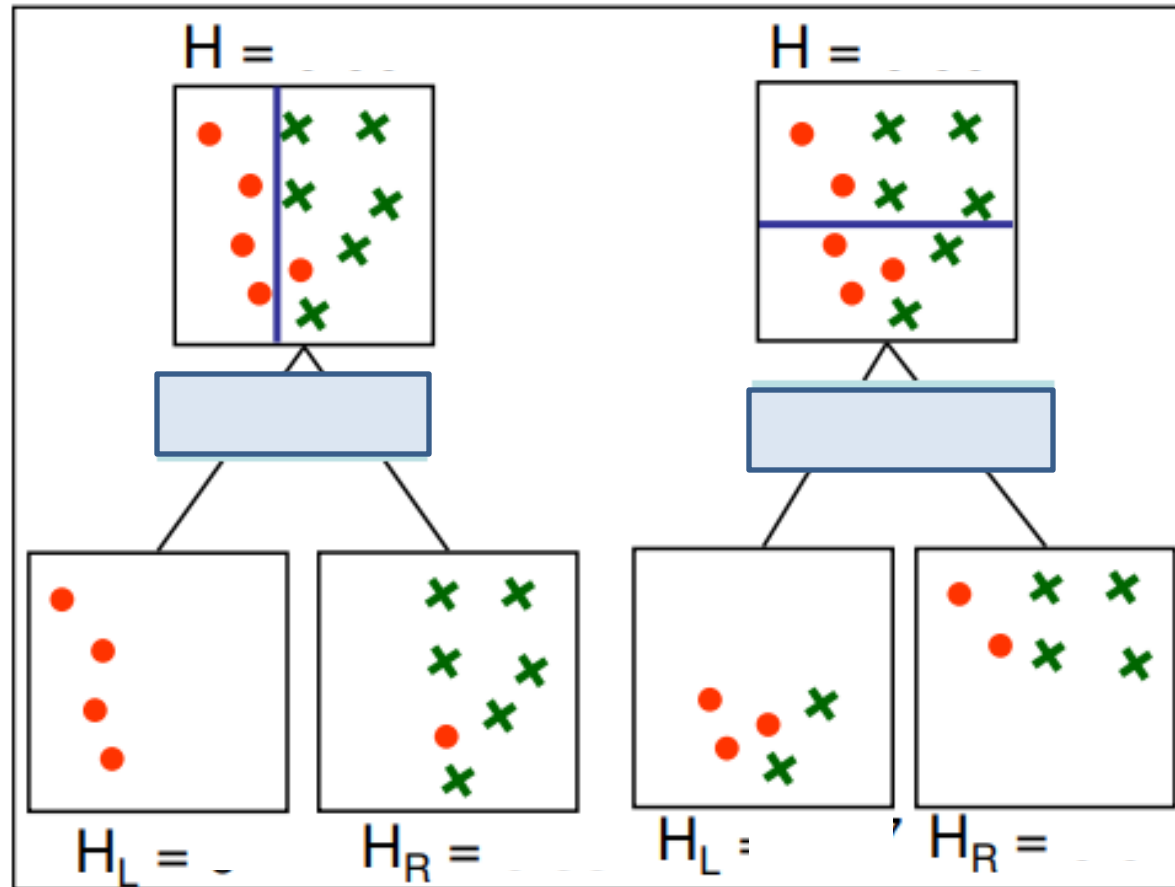
$$IG(Y|X) = H(Y) - H(Y|X)$$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

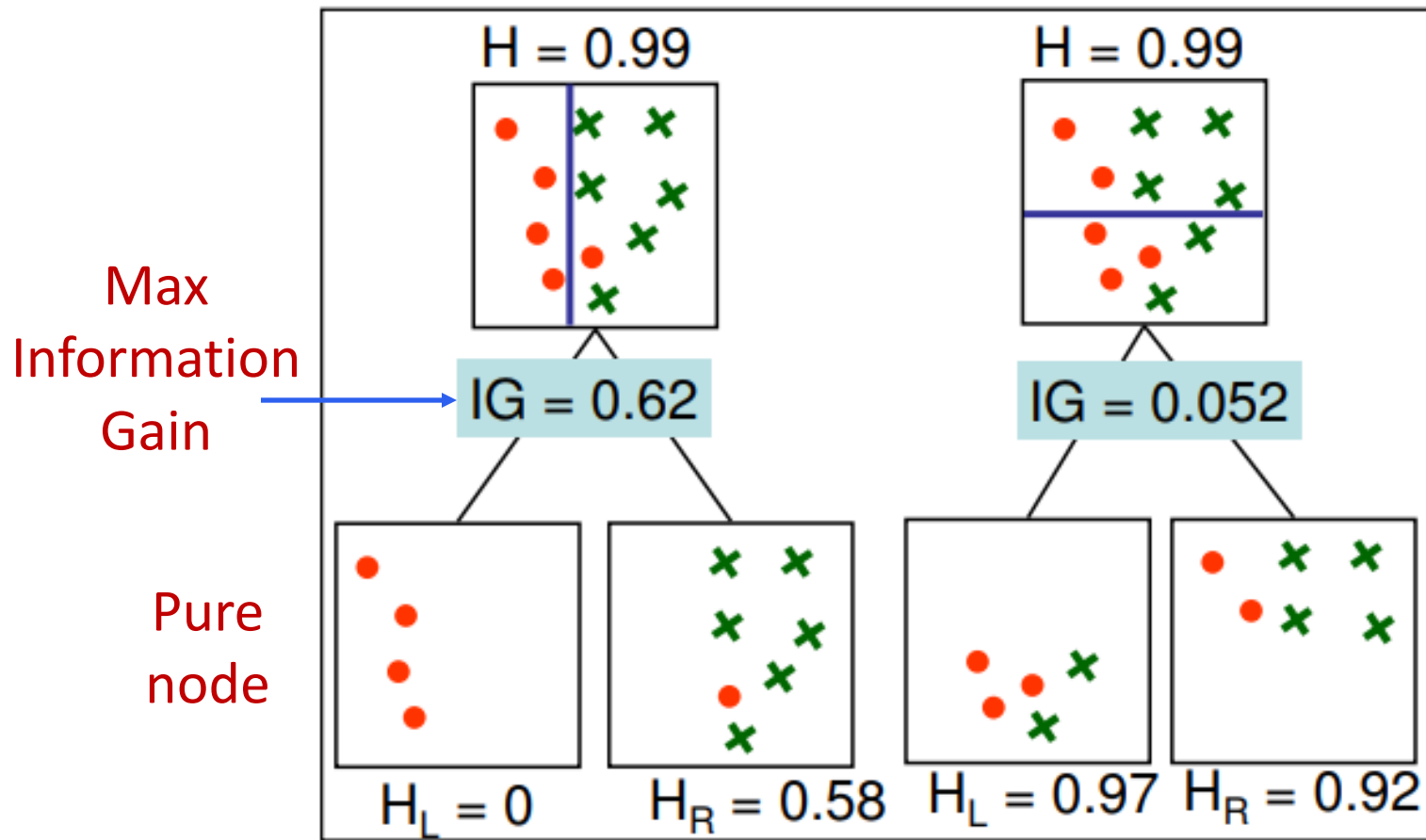
**Example:**

- $H(Y) = 1$
- $H(Y|X) = 0.5$
- Thus  $IG(Y|X) = 1 - 0.5 = 0.5$

# Example Information Gain



# Example Information Gain



# Learning Decision Trees

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

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

- Recurse

ID3 algorithm uses Information Gain  
Information Gain reduces uncertainty on Y

# Impurity Metrics

Split a node according to max reduction of impurity

## 1. Entropy

## 2. Gini Index

- For binary case with prob  $p_0, p_1$ :

$$I(p_0, p_1) = 2p_0p_1 = 2p_0(1 - p_0)$$

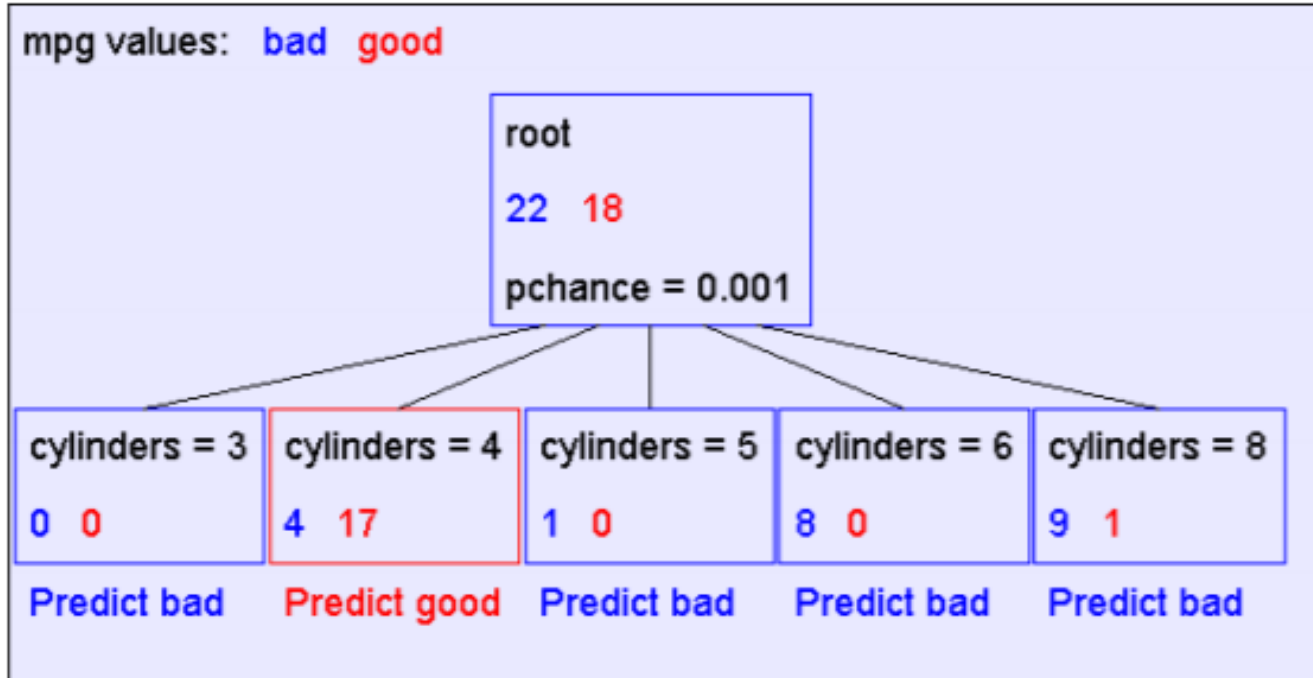
- For multi-class with prob  $p_1, \dots, p_K$ :

$$I(p_1, \dots, p_K) = 1 - \sum_{k=1}^K p_k^2$$

- Properties

- Impurity metrics have value 0 for pure nodes
- Impurity metrics are maximized for uniform distribution (nodes with most uncertainty)

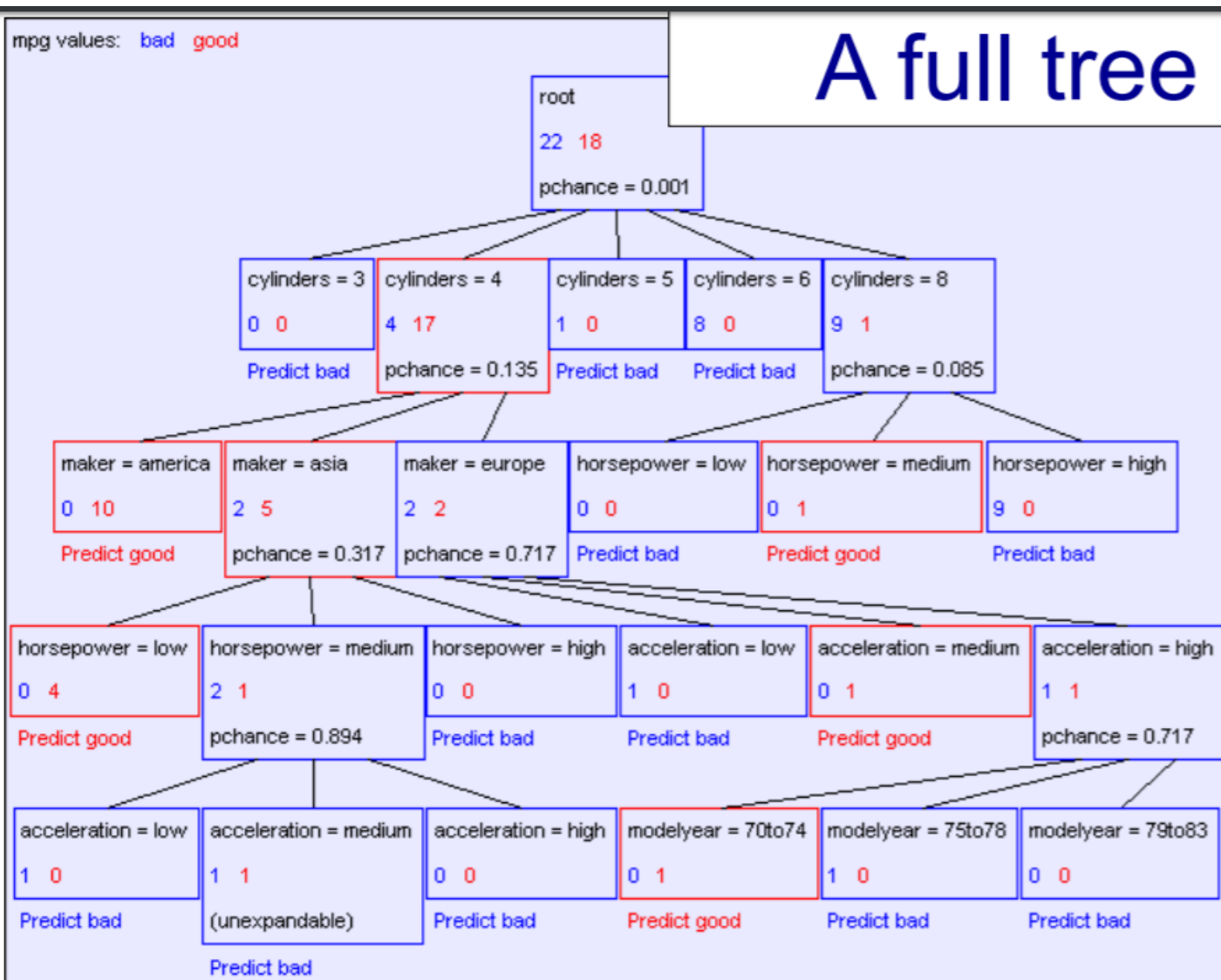
# When to stop?



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

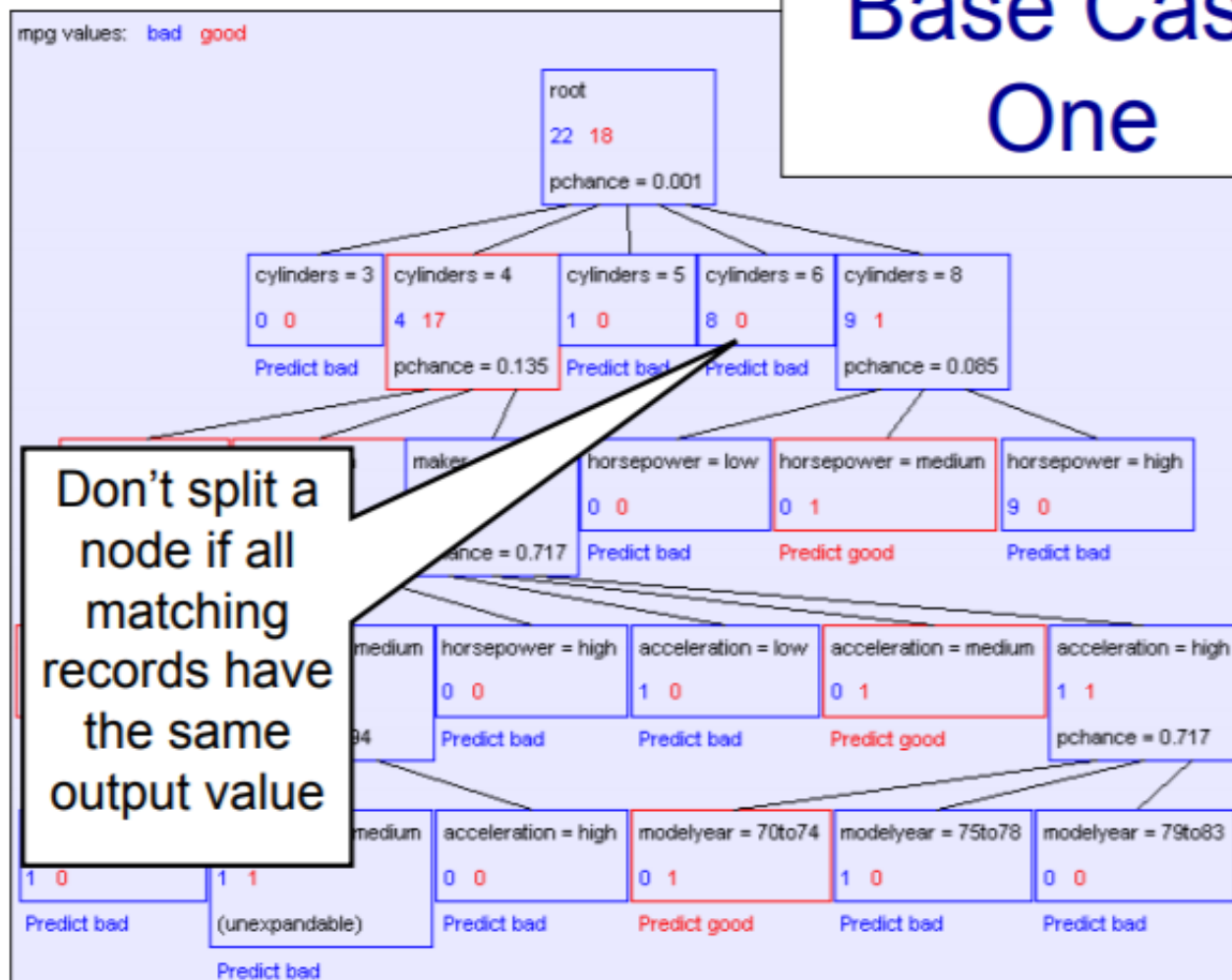


# Full Tree

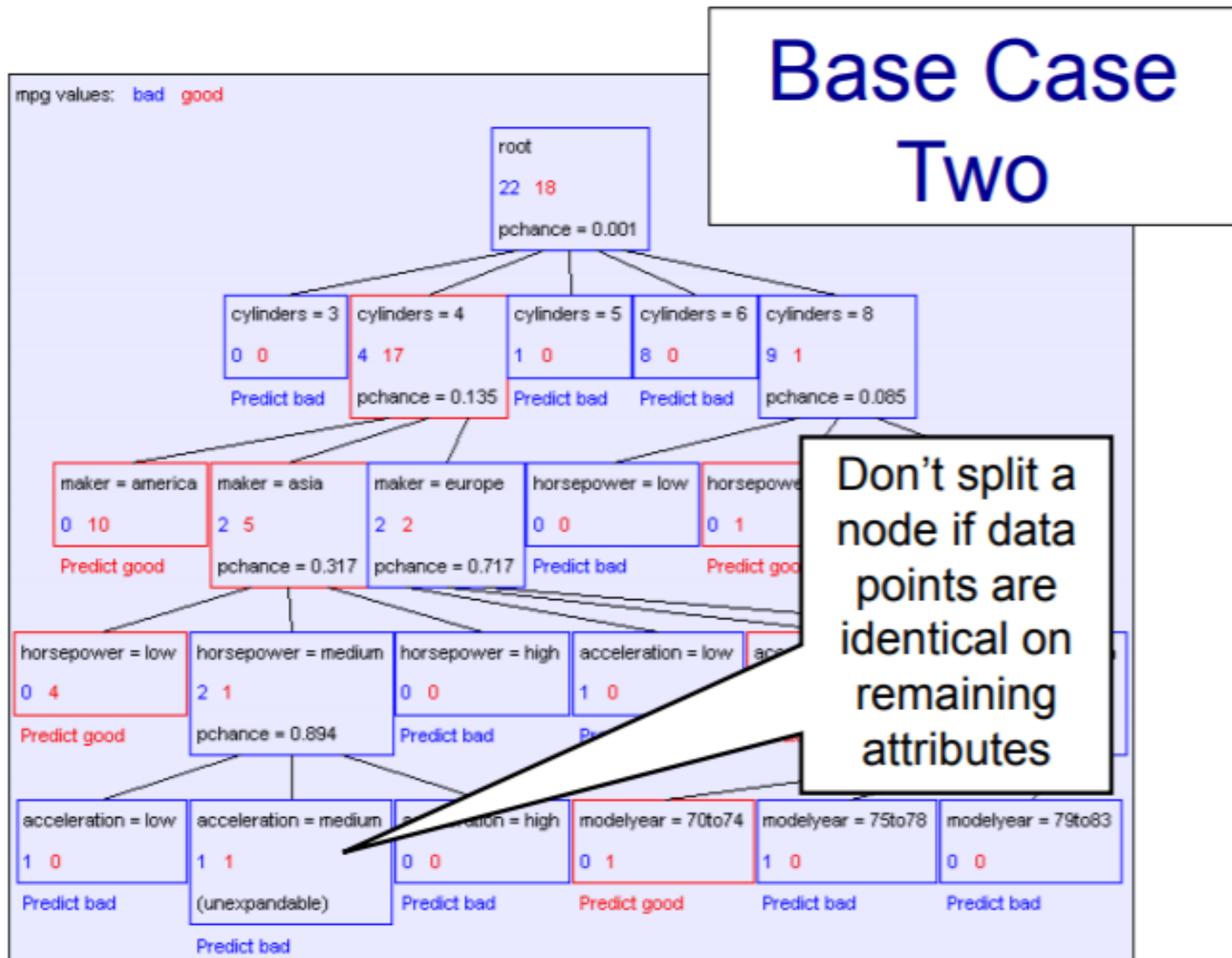


# Case 1

## Base Case One



# Case 2



# Decision Trees

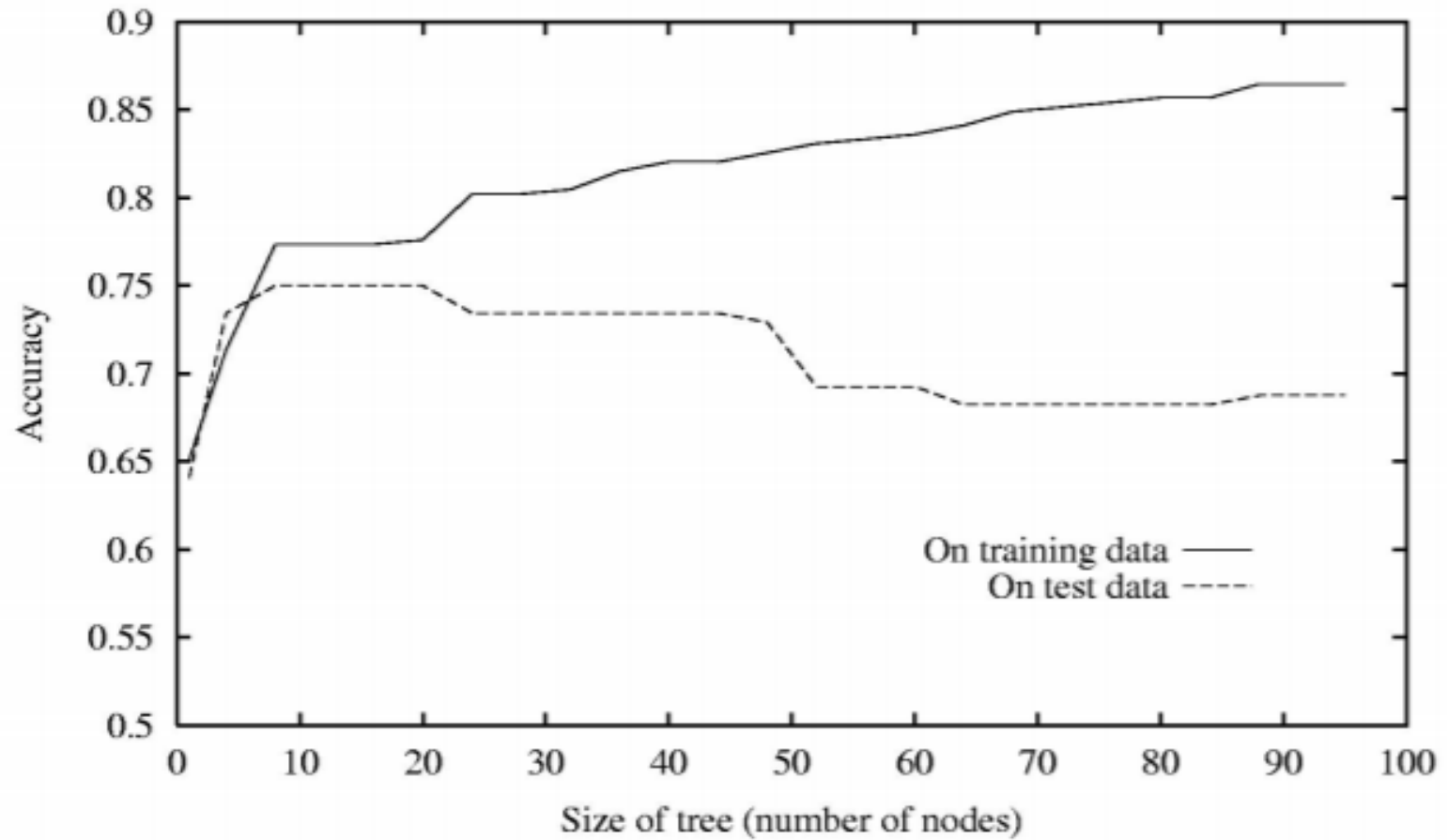
**BuildTree(*DataSet*, *Output*)**

- 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  $X$ .

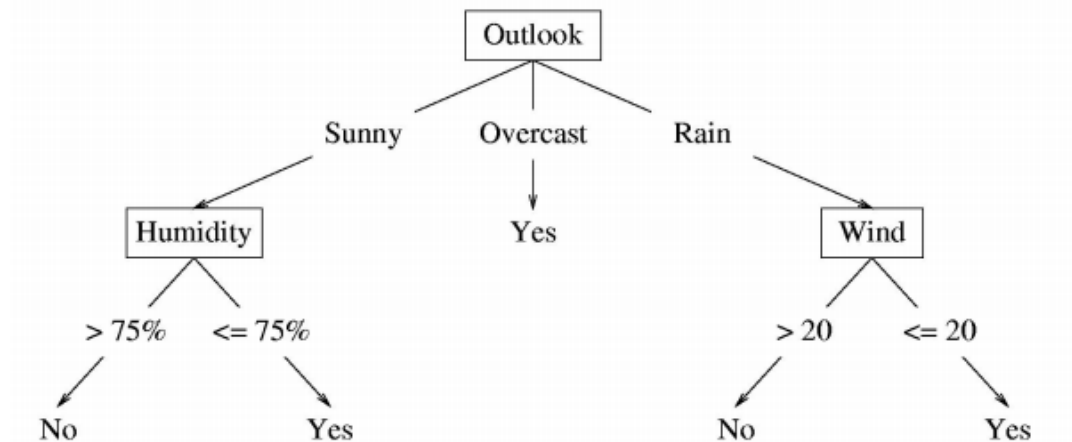
# Overfitting



# Solutions against Overfitting

- 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
  - Minimum number of samples per leaf
- Pruning
  - Remove branches of the tree that increase error using cross-validation

# Real-valued Features



- Change to binary splits by choosing a threshold
- One method:
  - Sort instances by value, identify adjacencies with different classes

Humidity	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

candidate splits

- Choose among splits by InfoGain()

# Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
  - Andrew Moore
- Thanks!