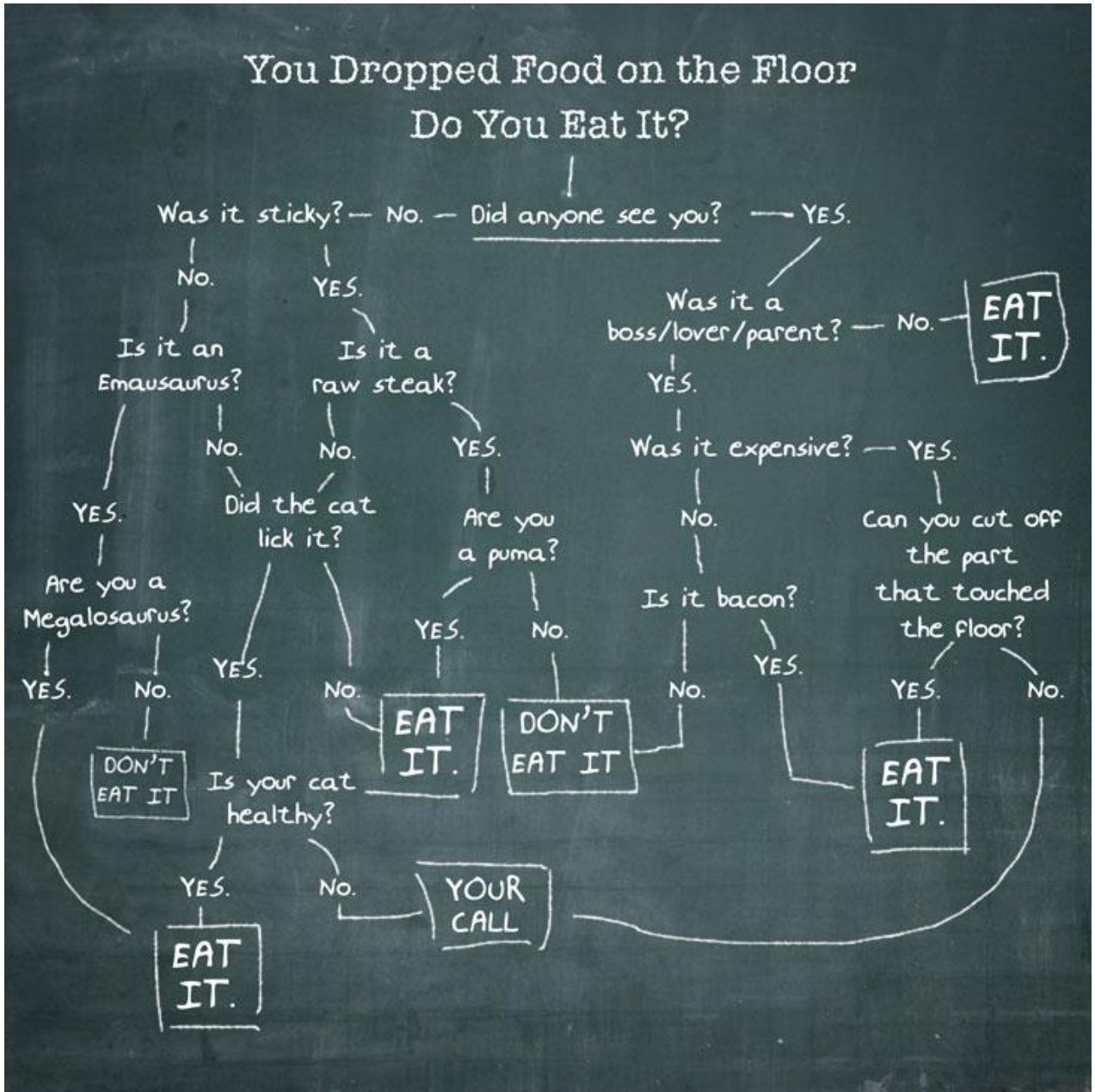
VBM683
Machine Learning

Pinar Duygulu

Slides are adapted from
Dhruv Batra

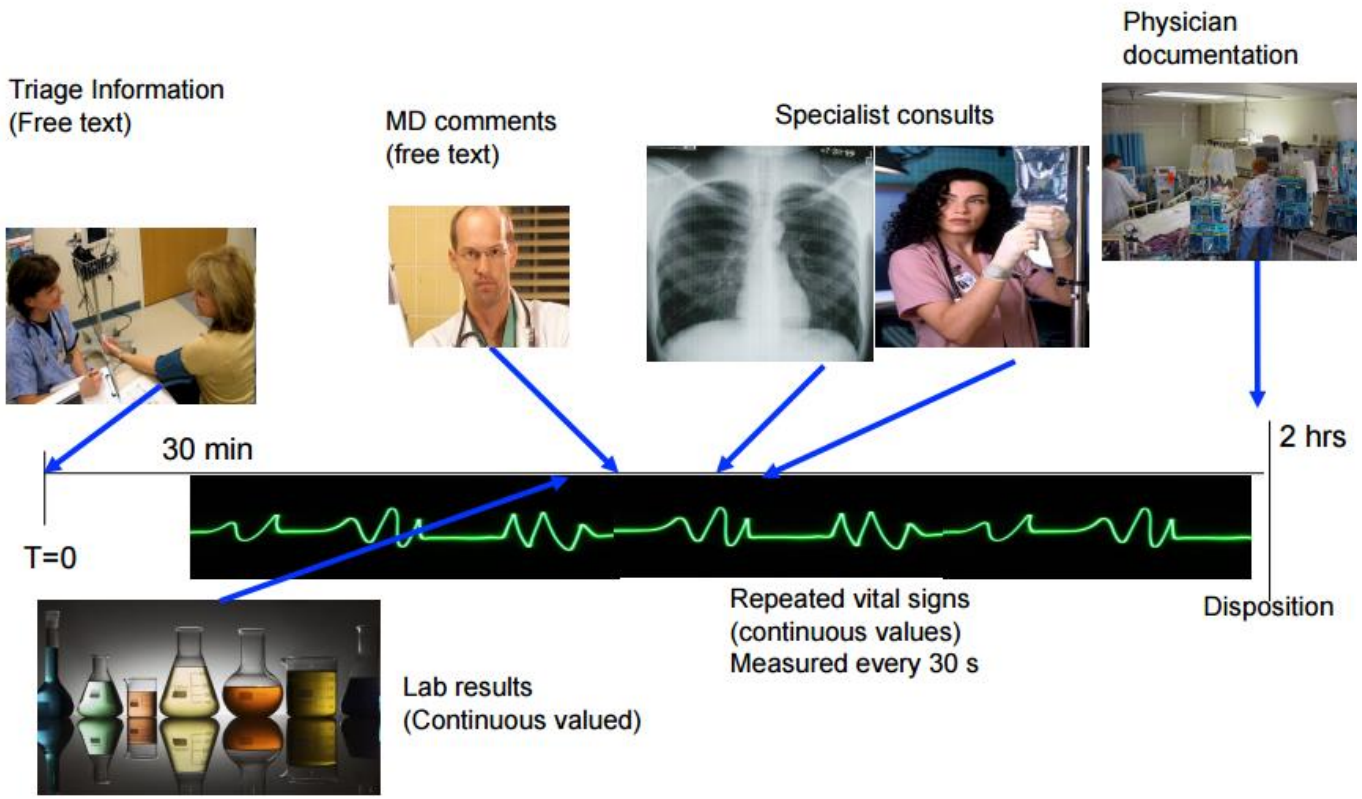
Decision Trees



Synonyms

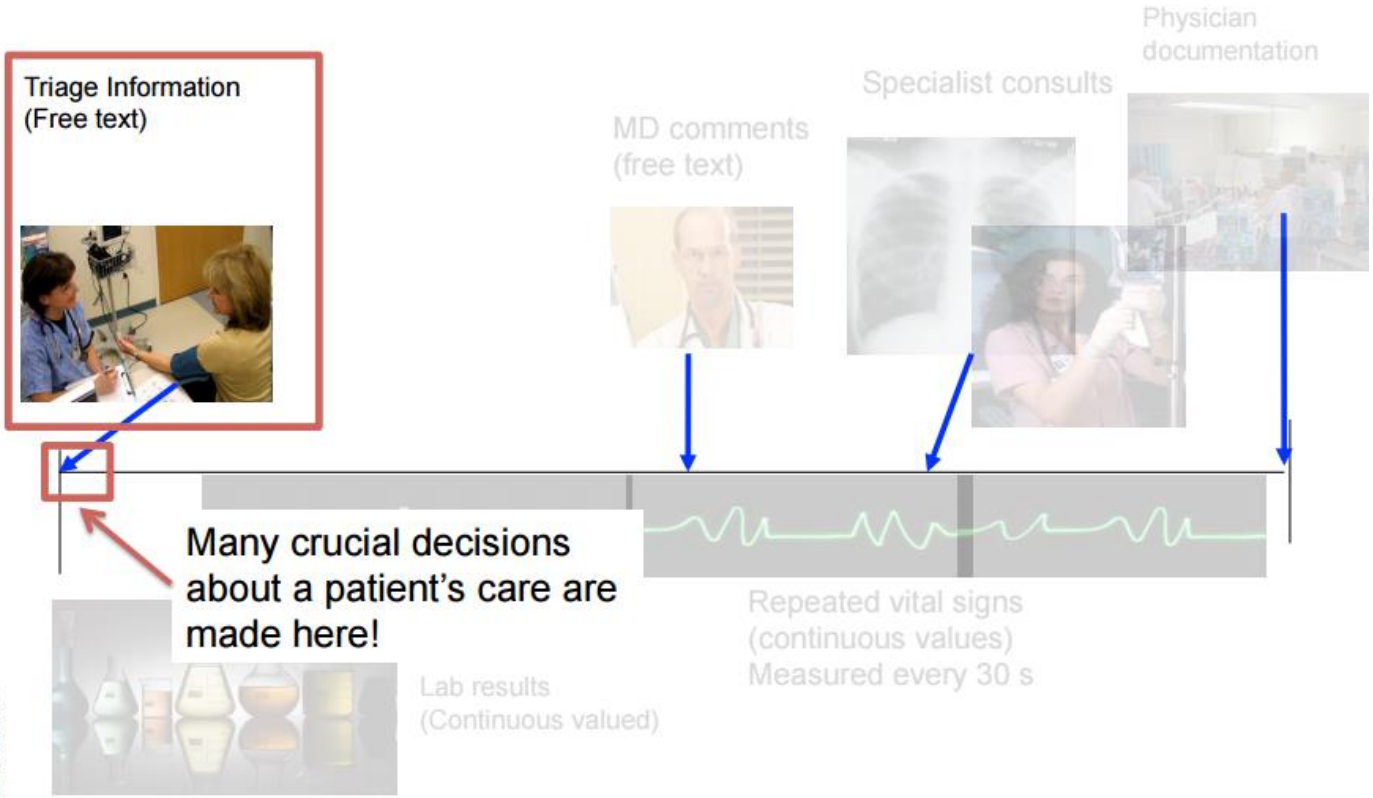
- Decision Trees
- Classification and Regression Trees (CART)
- Algorithms for learning decision trees:
  - ID3
  - C4.5
- Random Forests
  - Multiple decision trees

Machine Learning in the ER



slide by David Sontag

Can we predict infection?

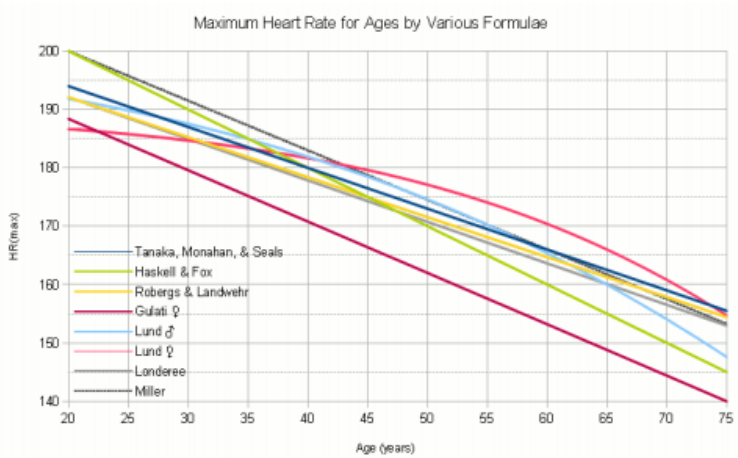


slide by David Sontag

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!



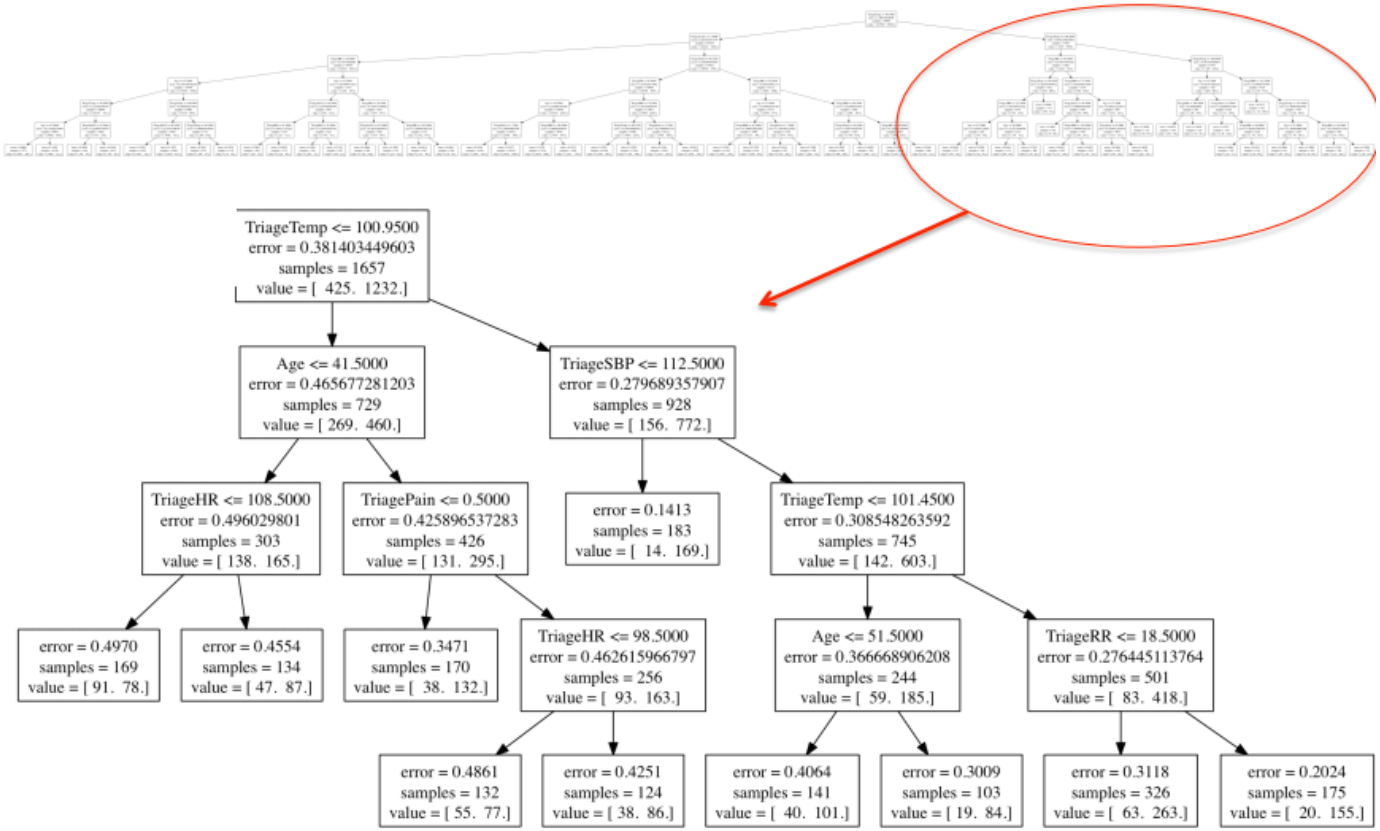
slide by David Sontag

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

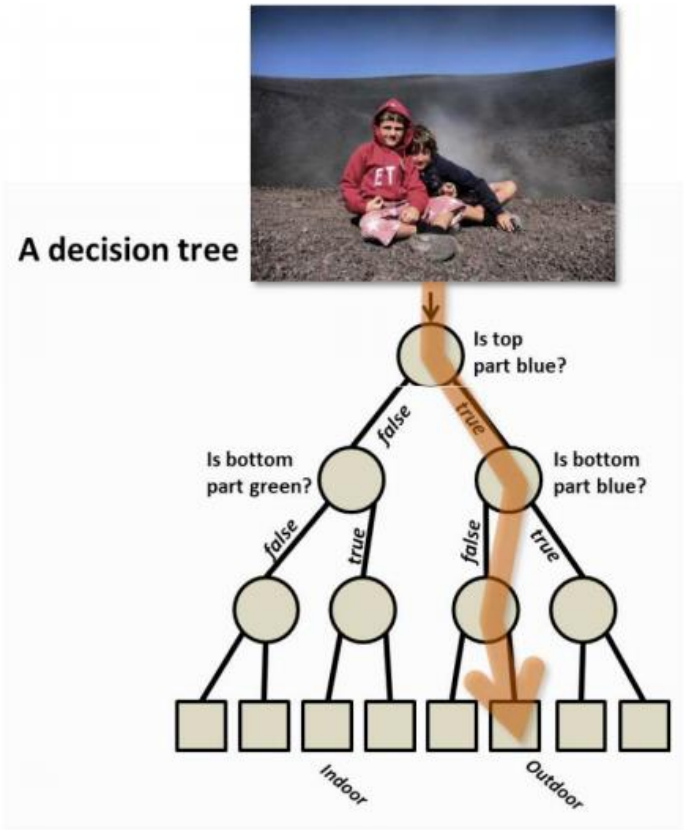
slide by David Sontag

Predicting infection using decision trees



slide by David Sontag

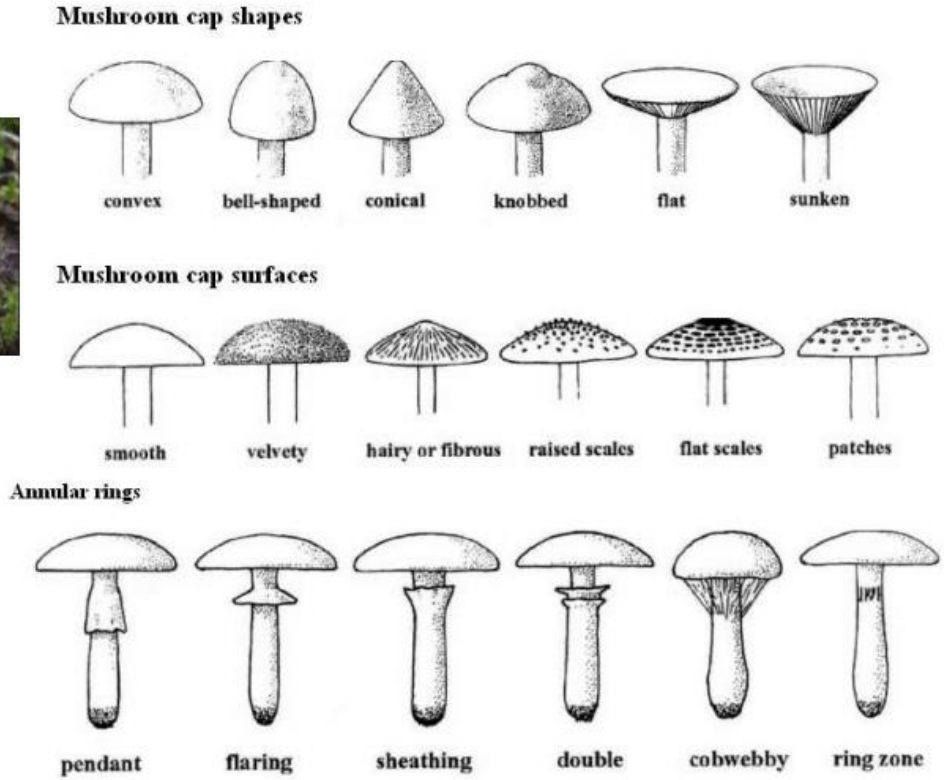
Example: Image Classification



[Criminisi et al, 2011]

slide by Nando de Freitas

Example: Mushrooms



http://www.usask.ca/biology/fungi/

slide by Jerry Zhu

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. ...

slide by Jerry Zhu

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. …

slide by Jerry Zhu

Pose Estimation

- Random Forests!
  - Multiple decision trees
  - http://youtu.be/HNkbG3KsY84

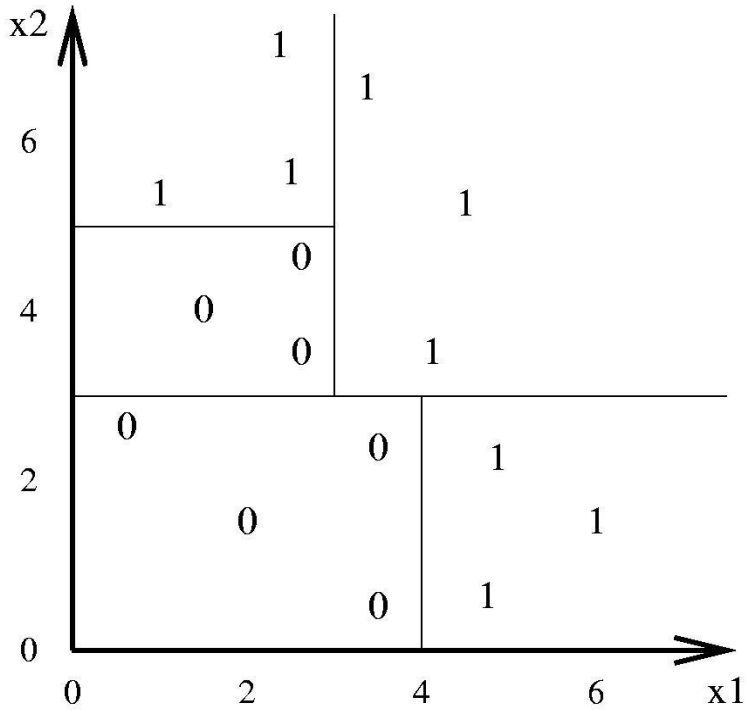
Learning Decision Trees

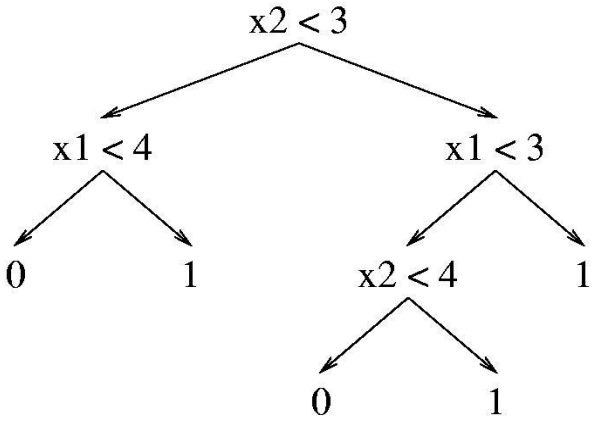
Decision trees provide a very popular and efficient hypothesis space.

- **Variable Size**. Any boolean function can be represented.
- **Deterministic**.
- **Discrete and Continuous Parameters**.

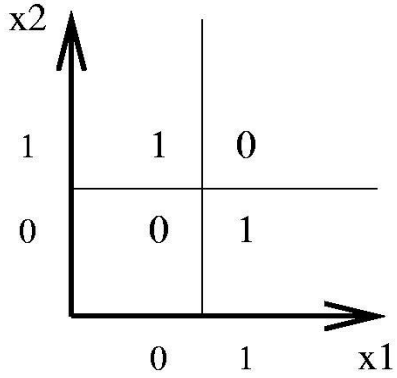
Decision Tree Decision Boundaries

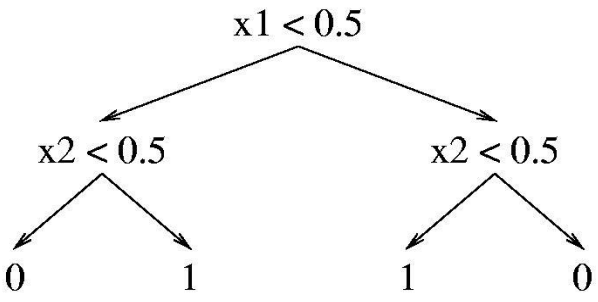
Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.





Decision Trees Can Represent Any Boolean Function





The tree will in the worst case require exponentially many nodes, however.

Decision Trees Provide Variable-Size Hypothesis Space

As the number of nodes (or depth) of tree increases, the hypothesis space grows

- **depth 1** ("decision stump") can represent any boolean function of one feature.
- **depth 2** Any boolean function of two features; some boolean functions involving three features (e.g., $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_3)$
- **etc.**

A small dataset: Miles Per Gallon

Suppose we want to predict MPG

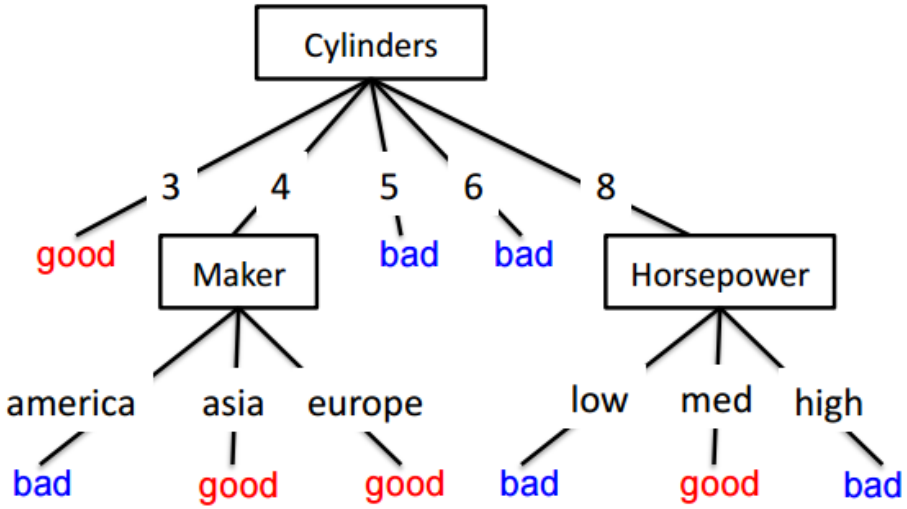
mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	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

40 Records

From the UCI repository (thanks to Ross Quinlan)

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

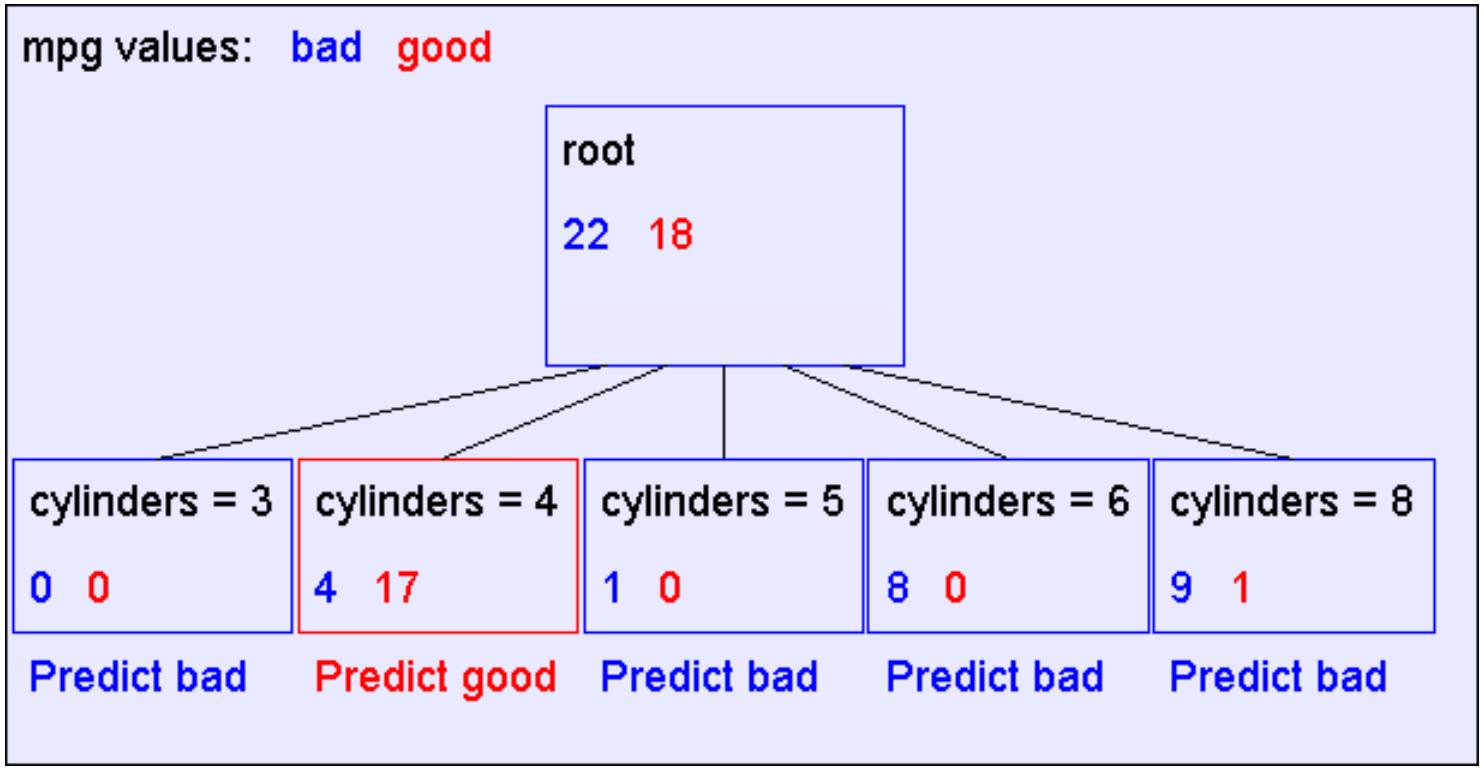
- Each internal node tests an attribute x_i
- Each branch assigns an attribute value $x_i = v$
- Each leaf assigns a class y
- To classify input x: traverse the tree from root to leaf, output the labeled y



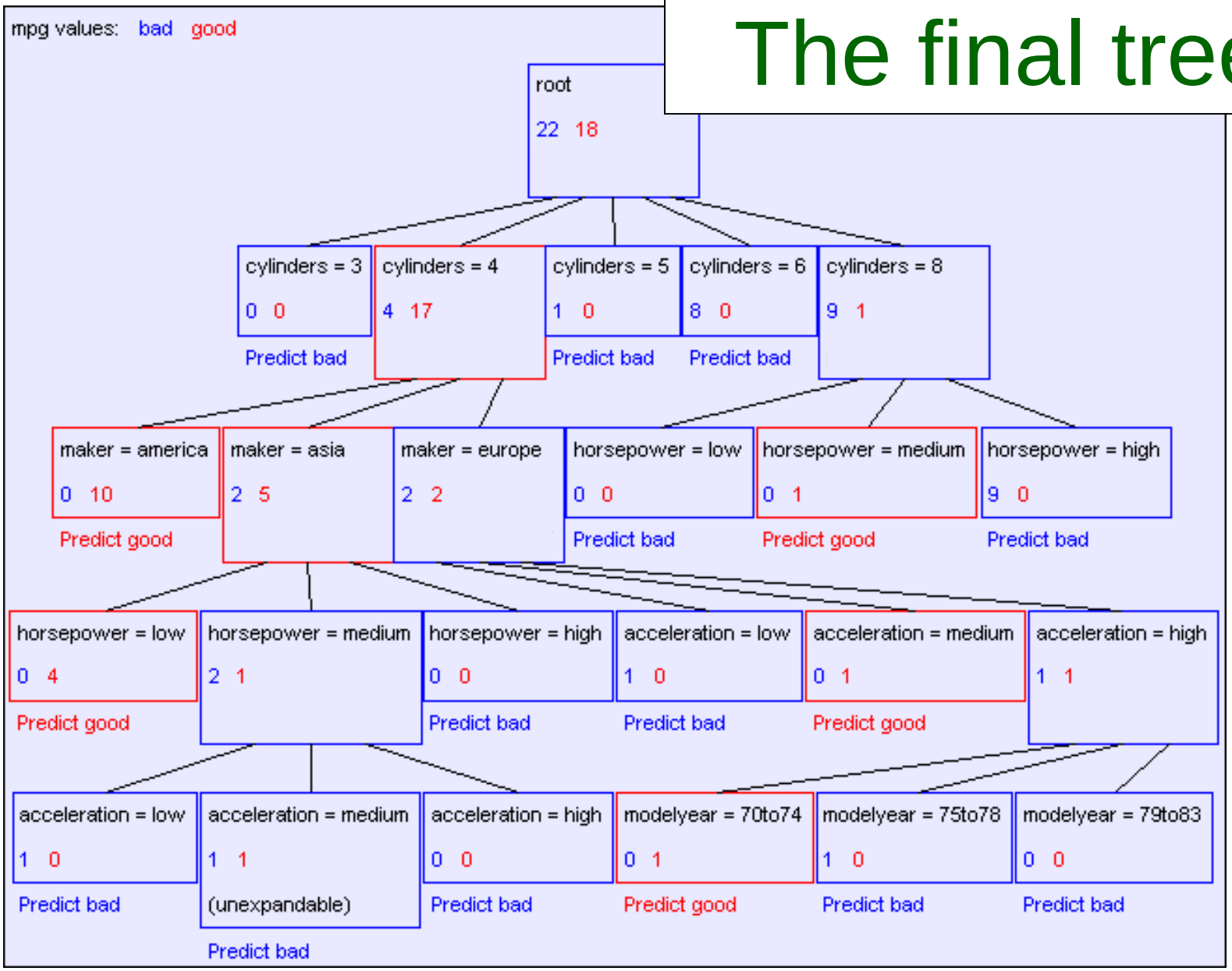
Human interpretable!

slide by David So

A Decision Stump



The final tree



Comments

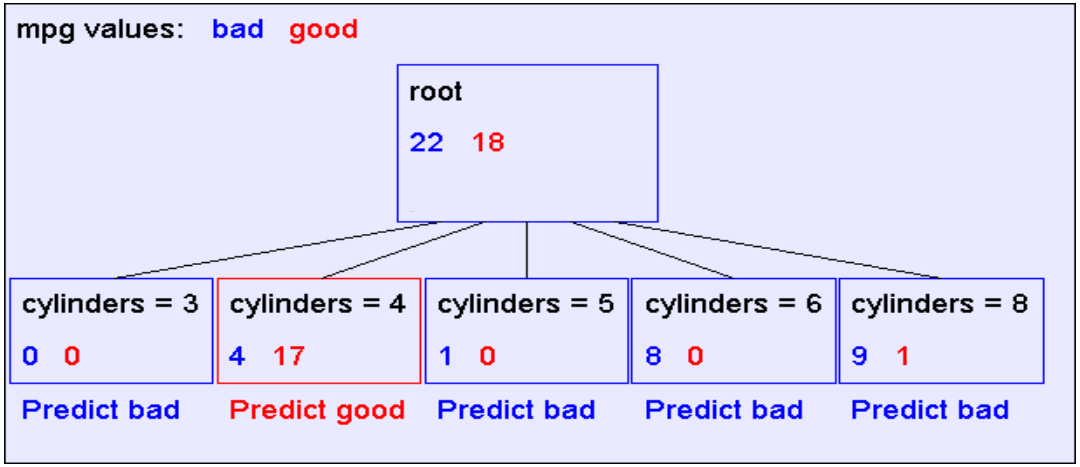
- Not all features/attributes need to appear in the tree.
- A features/attribute X_i may appear in multiple branches.
- On a path, no feature may appear more than once.
  - Not true for continuous features. We'll see later.
- Many trees can represent the same concept
- But, not all trees will have the same size!
  - e.g., $Y = (A \wedge B) \vee (\neg A \wedge C)$ (A and B) or (not A and C)

Learning decision trees is hard!!!

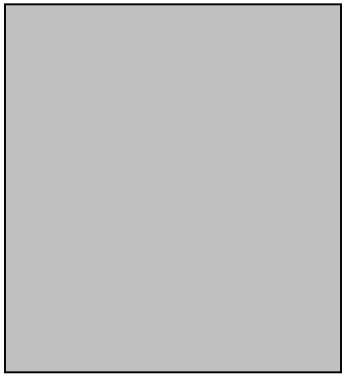
- Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest ’76]

- Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on **next best attribute (feature)**
  - Recurse
    - “Iterative Dichotomizer” (ID3)
    - C4.5 (ID3+improvements)

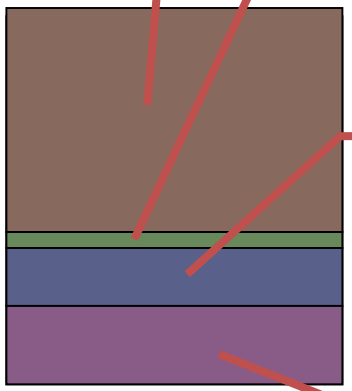
Recursion Step



Take the Original Dataset..



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



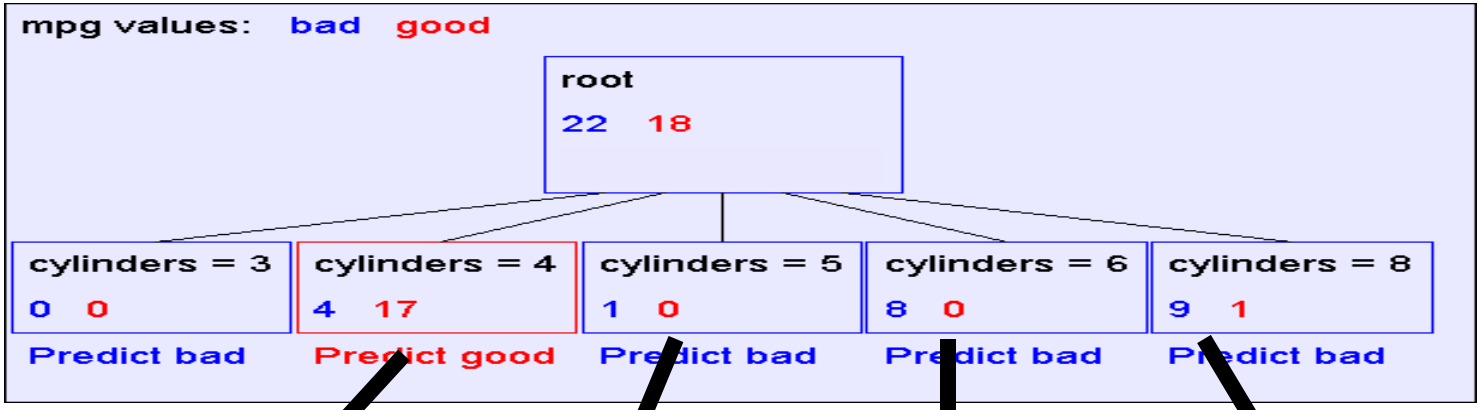
Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

Records in which cylinders = 8

Recursion Step

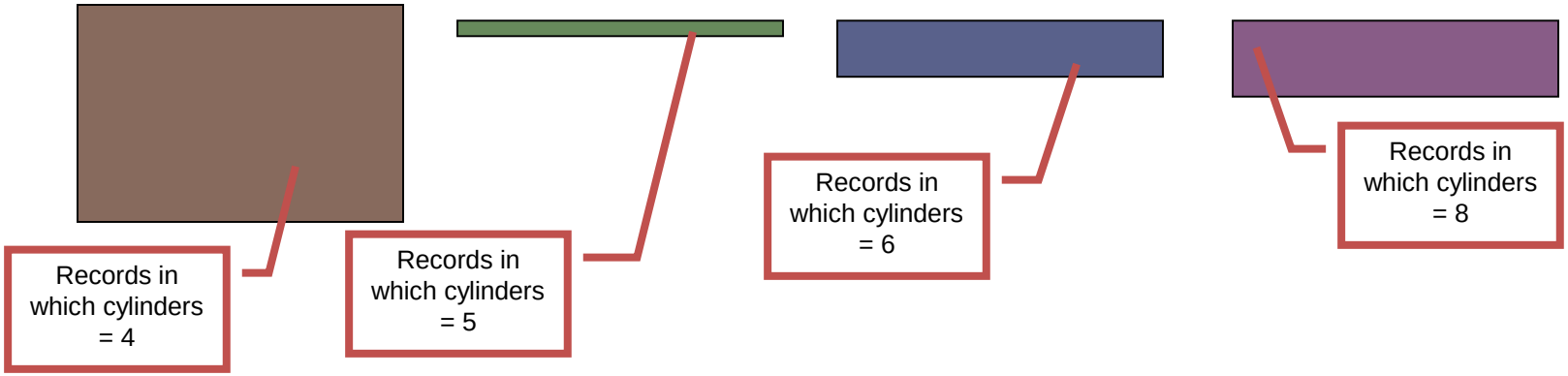


Build tree from These records..

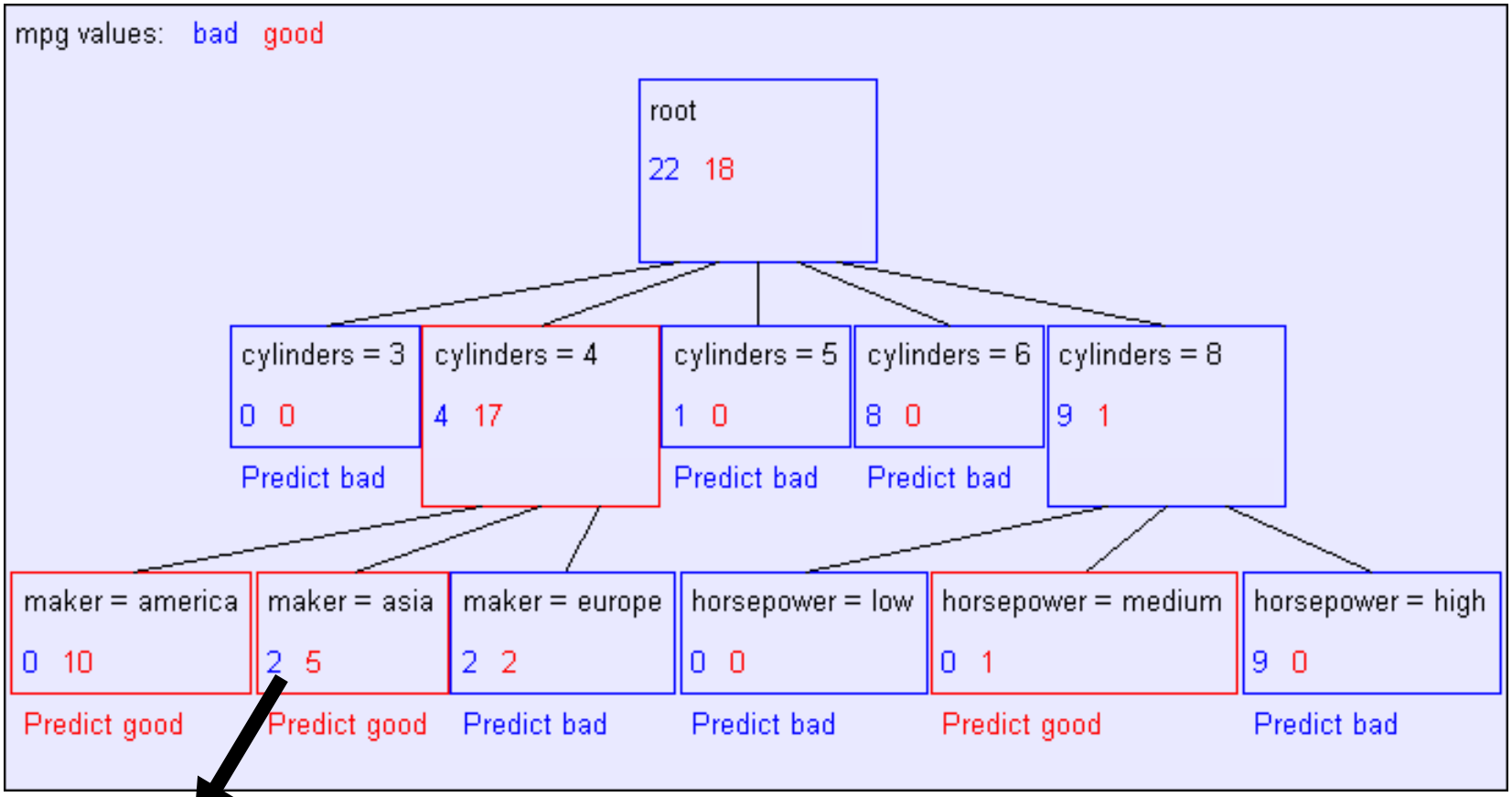
Build tree from These records..

Build tree from These records..

Build tree from These records..



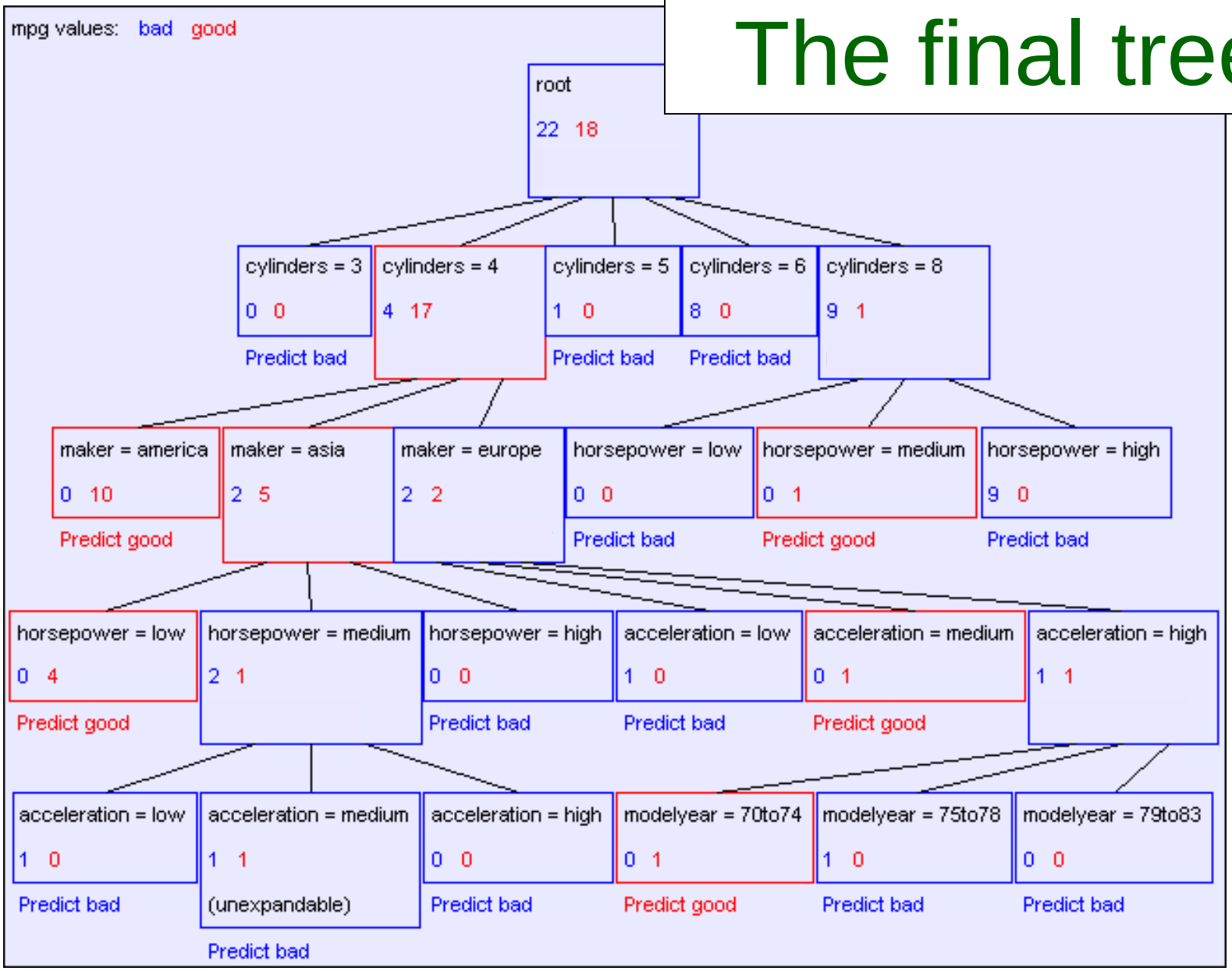
Second level of tree



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)

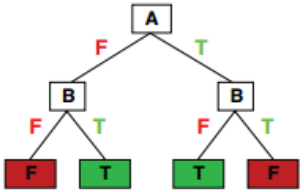
The final tree



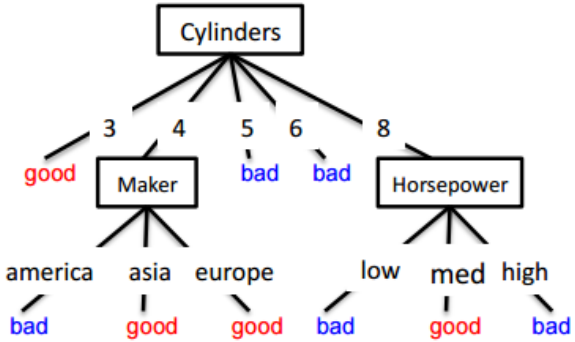
What functions can be represented?

- 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)

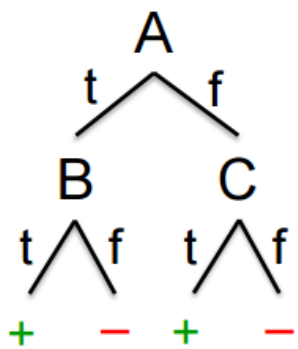


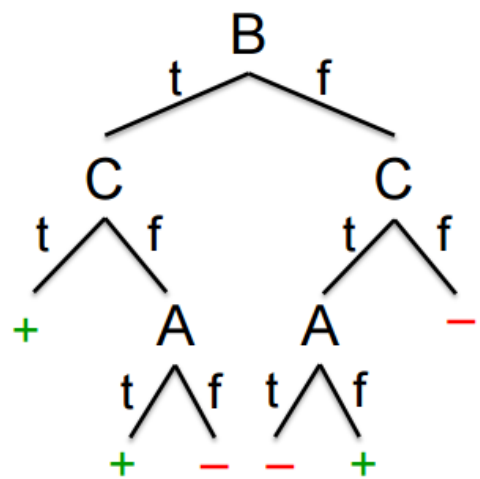
cyl=3 ∨ (cyl=4 ∧ (maker=asia ∨ maker=europe)) ∨ …

slide by David Sontag

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?

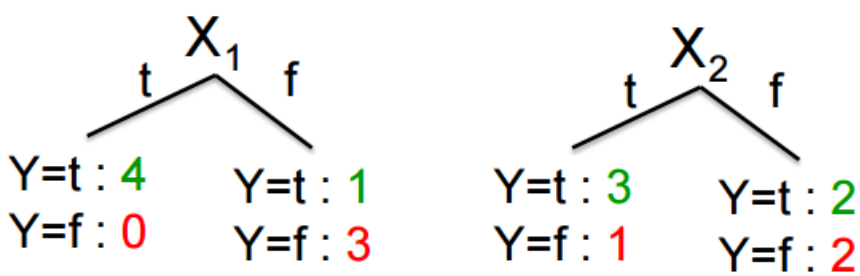
slide by David Sontag

Choosing a good attribute

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

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

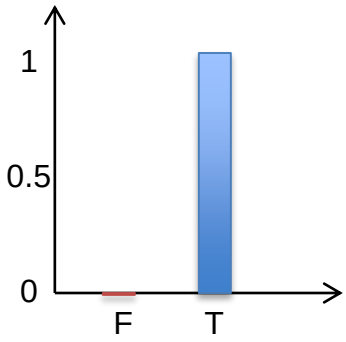
Idea: use counts at leaves to define probability distributions, so we can measure uncertainty!

slide by David Sontag

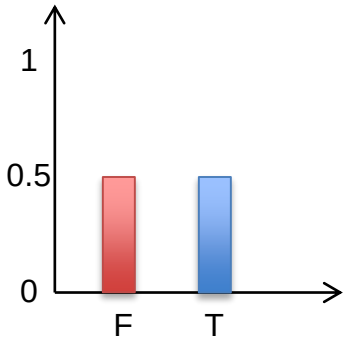
Measuring uncertainty

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

$P(Y=F \mid X_1= T) = 0$	$P(Y=T \mid X_1= T) = 1$



$P(Y=F \mid X_2=F) = 1/2$	$P(Y=T \mid X_2=F) = 1/2$



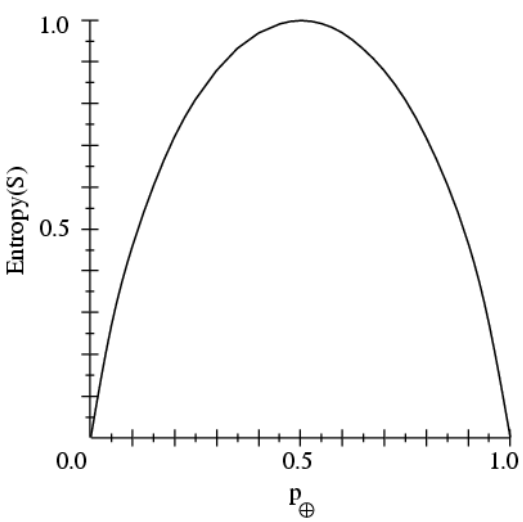
Entropy

Entropy $H(X)$ 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)

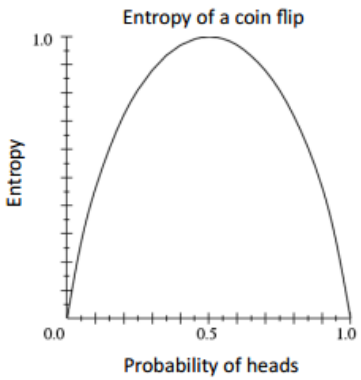


High, Low Entropy

- "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 by 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

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

slide by David Sontag

Conditional Entropy

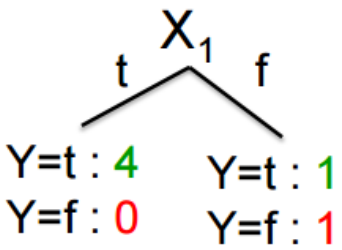
Conditional Entropy $H(Y|X)$ of a random variable Y conditioned on a random variable X

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

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

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

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

$= 2/6$

slide by David Sontag

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

slide by David Sontag

Information gain

- Advantage of attribute – decrease in uncertainty
  - Entropy of Y before you split
  - Entropy after split
    - Weight by probability of following each branch, i.e., normalized number of records

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

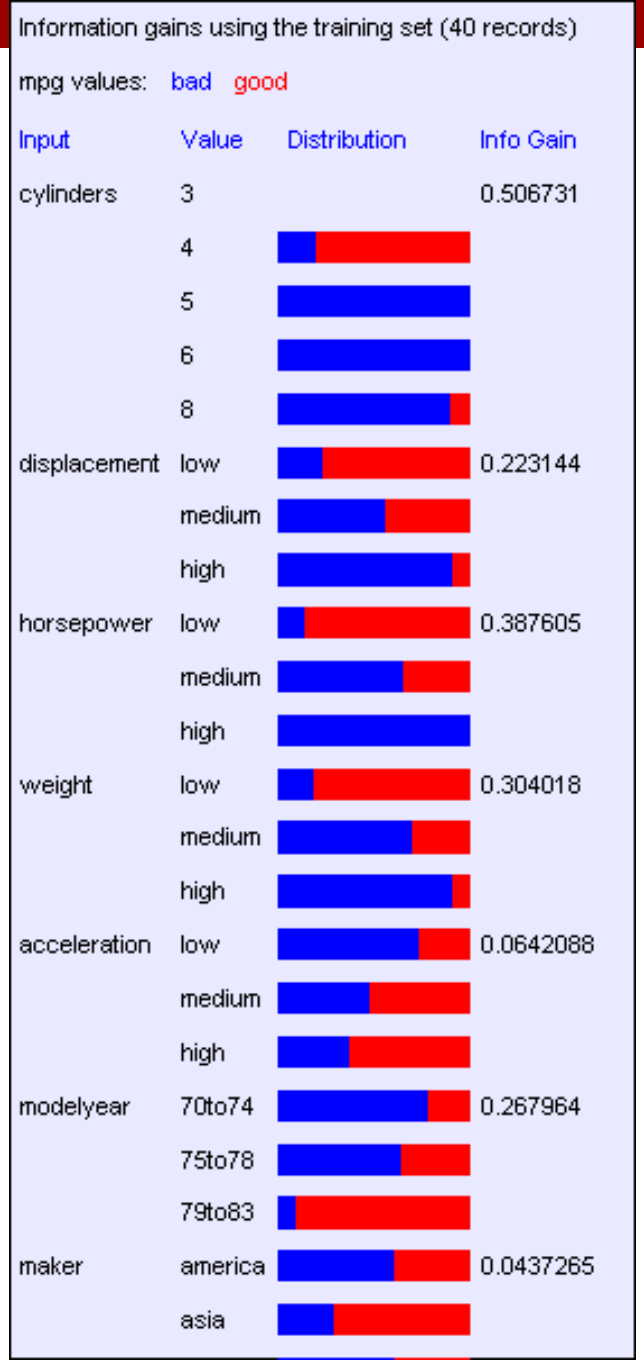
- Information gain is difference $IG(X) = H(Y) - H(Y \mid X)$
  - (Technically it's mutual information; but in this context also referred to as information gain)

Learning decision trees

- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute
  - Split on $\arg\max_i IG(X_i) = \arg\max_i H(Y) - H(Y \mid X_i)$
- Recurse

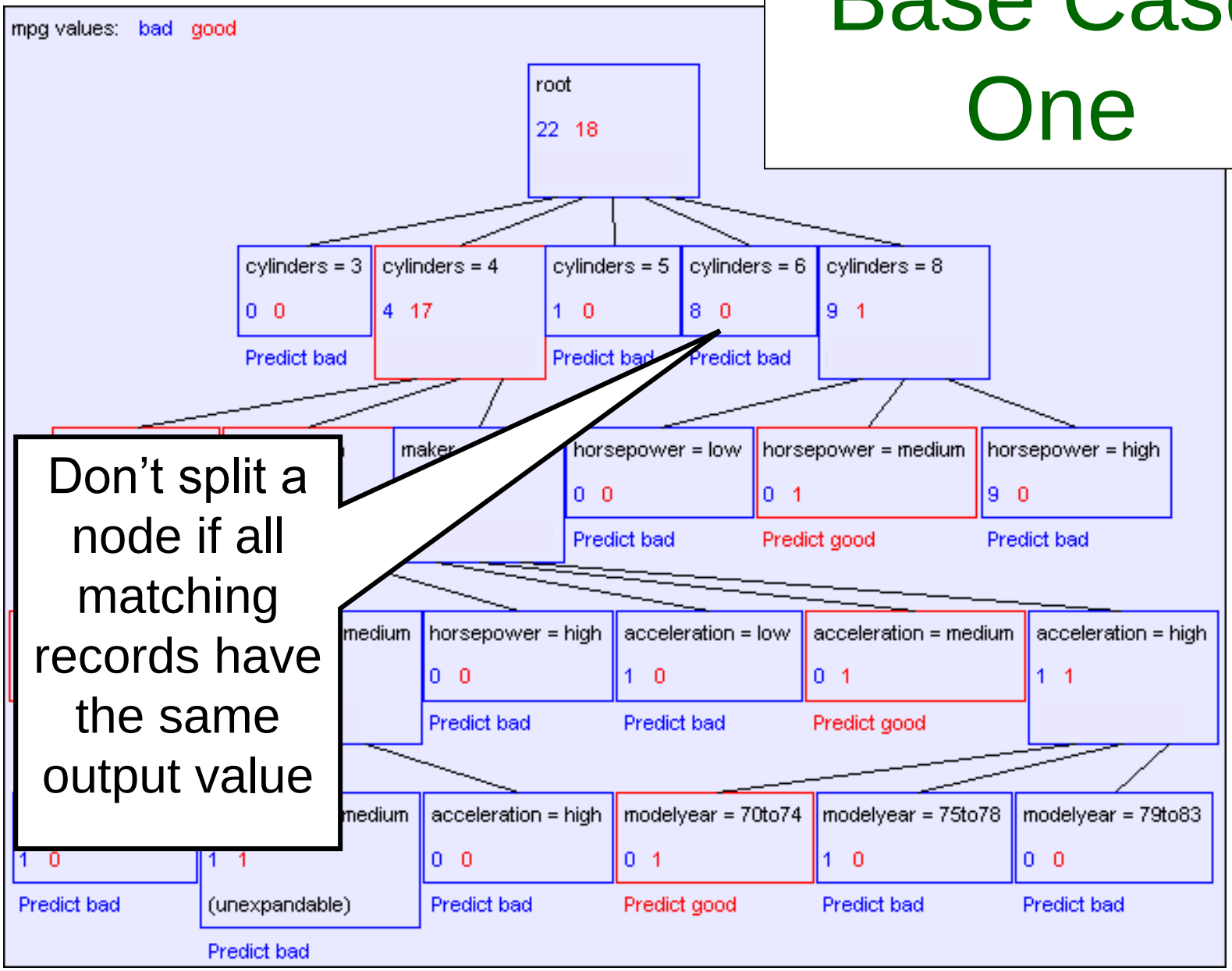
Suppose we want to predict MPG

Look at all the information gains…



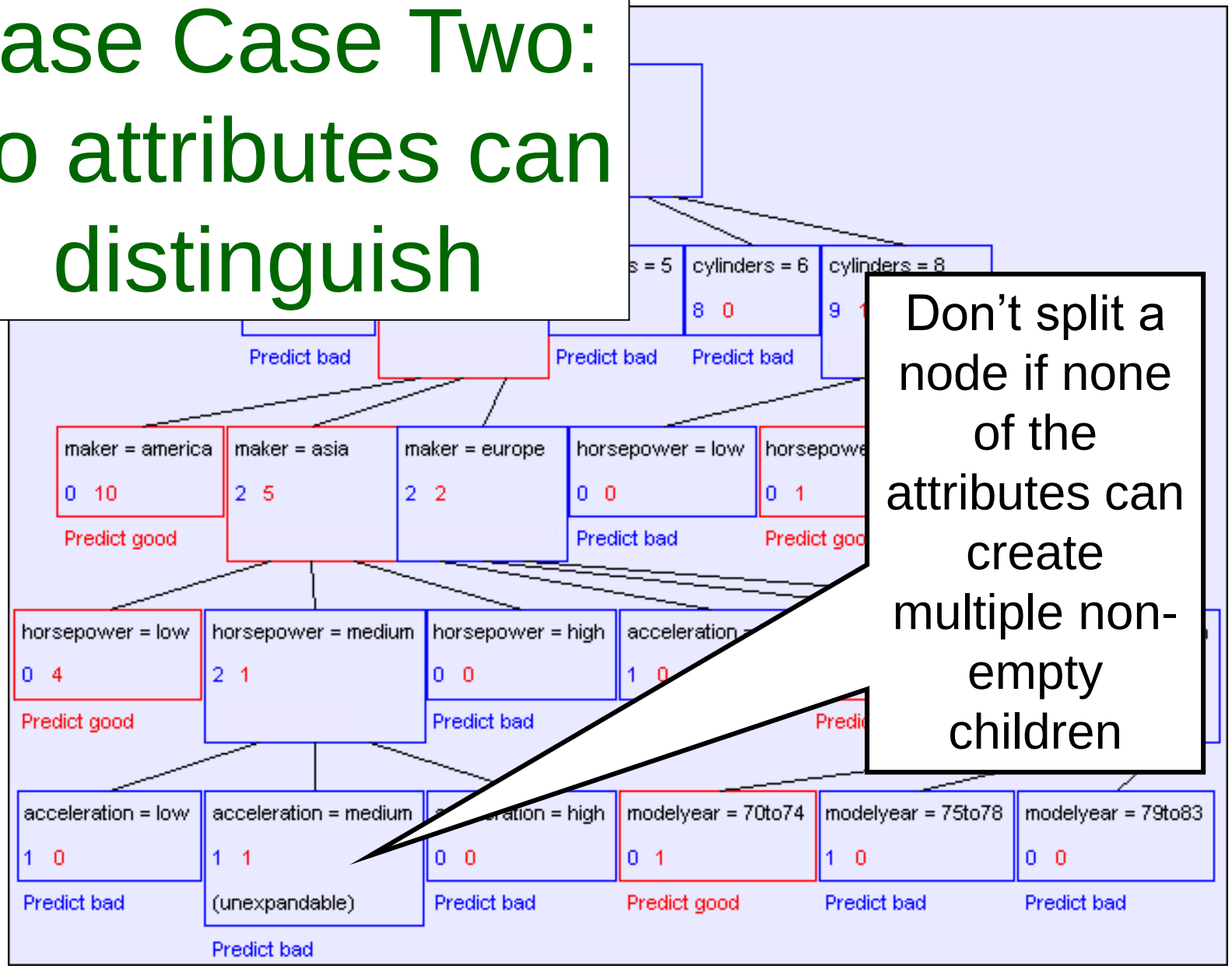
When do we stop?

Base Case One



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

Base Case Two: No attributes can distinguish



Don't split a node if none of the attributes can create multiple non-empty children

Base Cases

- 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

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 zero information gain then don't recurse

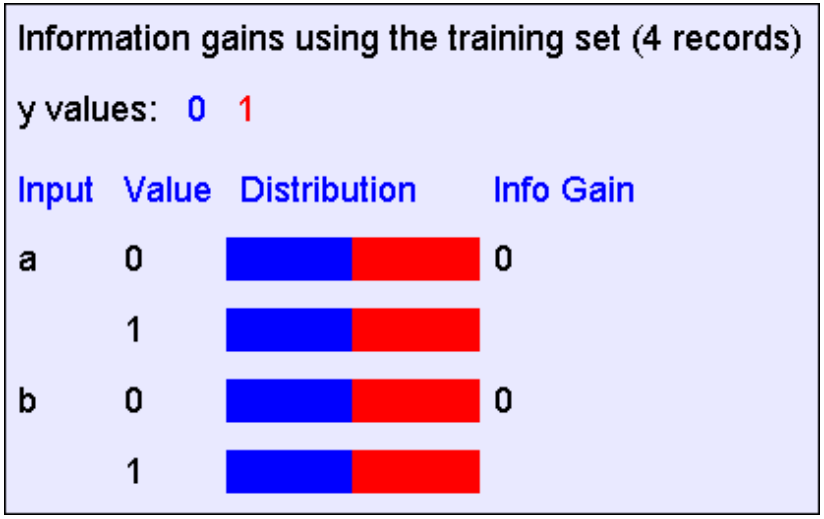
- *Is this a good idea?*

The problem with Base Case 3

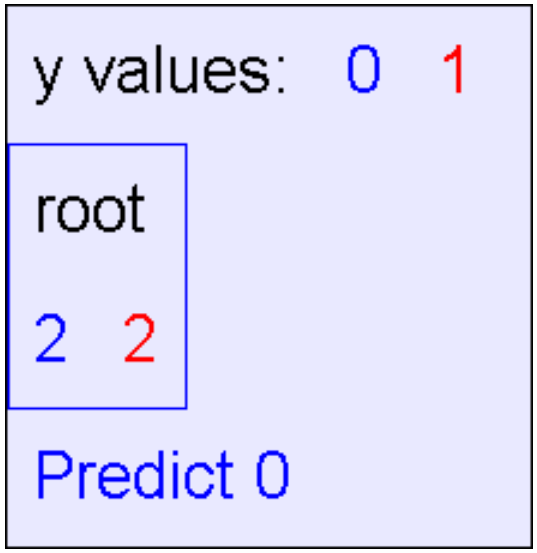
a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

y = a XOR b

The information gains:



The resulting decision tree:

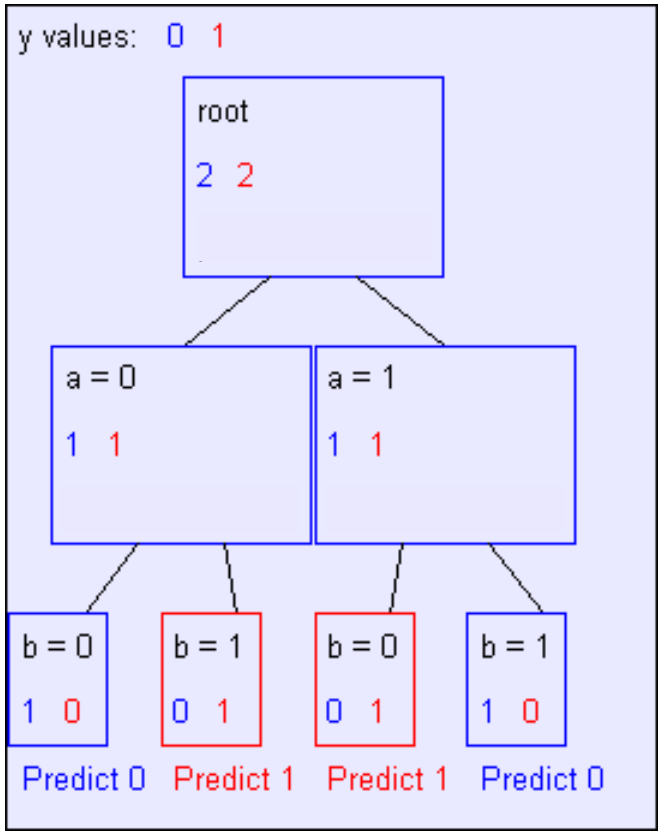


If we omit Base Case 3:

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

y = a XOR b

The resulting decision tree:



Basic Decision Tree Building Summarized

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 and return a non-leaf node with n_X children.
  - The *i*'th child should be built by calling

    BuildTree(DS_i,*Output*)

    Where DS_i built consists of all those records in DataSet for which X = *i*th distinct value of X.

output = DecisionTree(data)

-If(data.out is all one label) then return that label.

-If(data.in are identical) then return majority label.

-Split on next best feature (call it x*)

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

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

-For each value a of x* create a node and recur:

DecisionTree(data.in(data.in.x* == a))

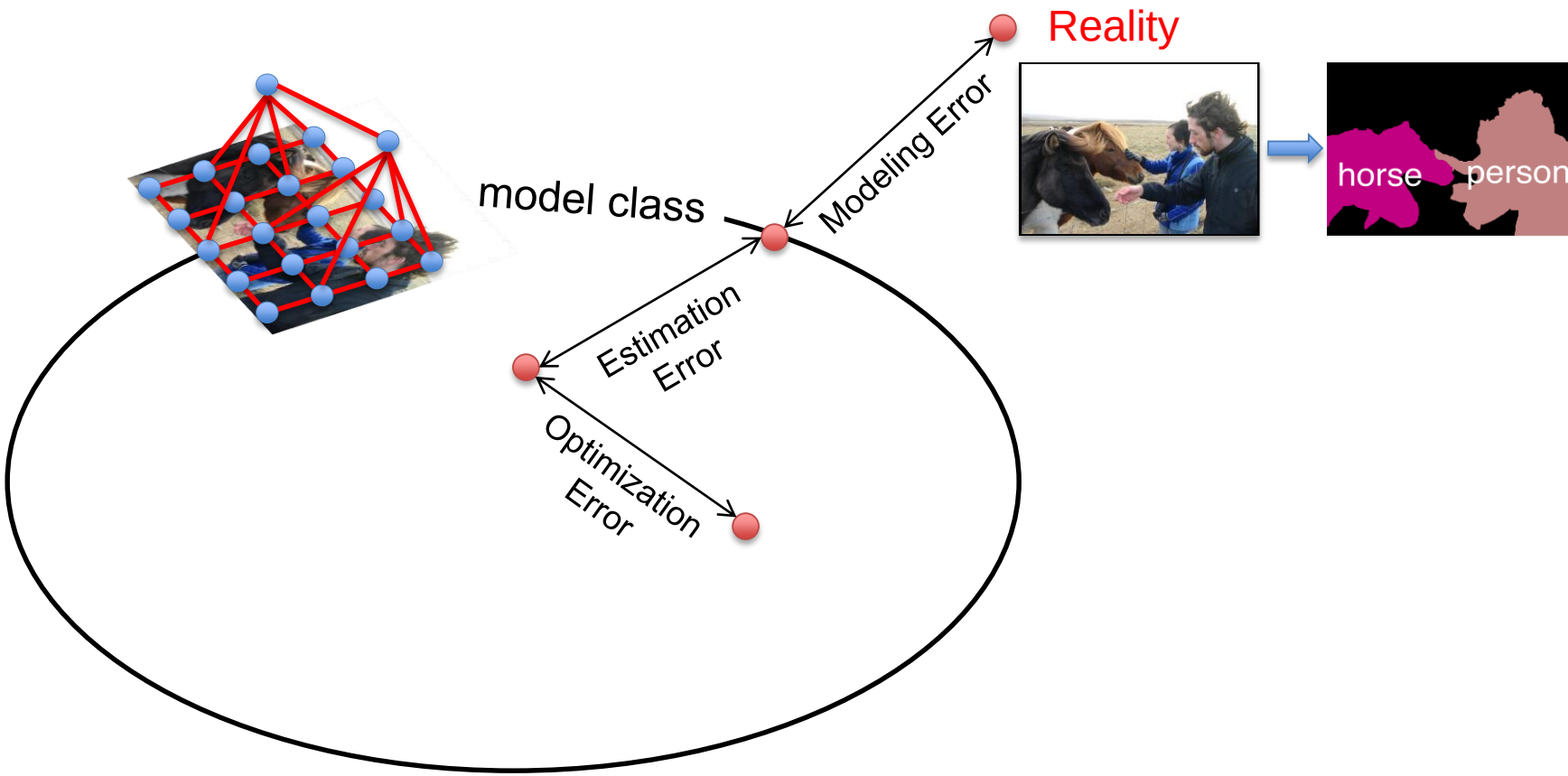
Will this mushroom kill me?					
Cap Shape	**Odor**	**Habitat**	**Cap Color**	**Stalk Shape**	**Poison**
convex	pungent	urban	brown	enlarging	Yes
convex	almond	grass	yellow	enlarging	No
bell	anise	meadows	white	enlarging	No
convex	none	urban	white	enlarging	Yes
convex	none	grass	gray	tapering	No
convex	almond	grass	yellow	enlarging	No
bell	almond	meadows	white	enlarging	Yes
bell	anise	meadows	white	enlarging	No
convex	pungent	grass	white	tapering	Yes

Decision trees will overfit

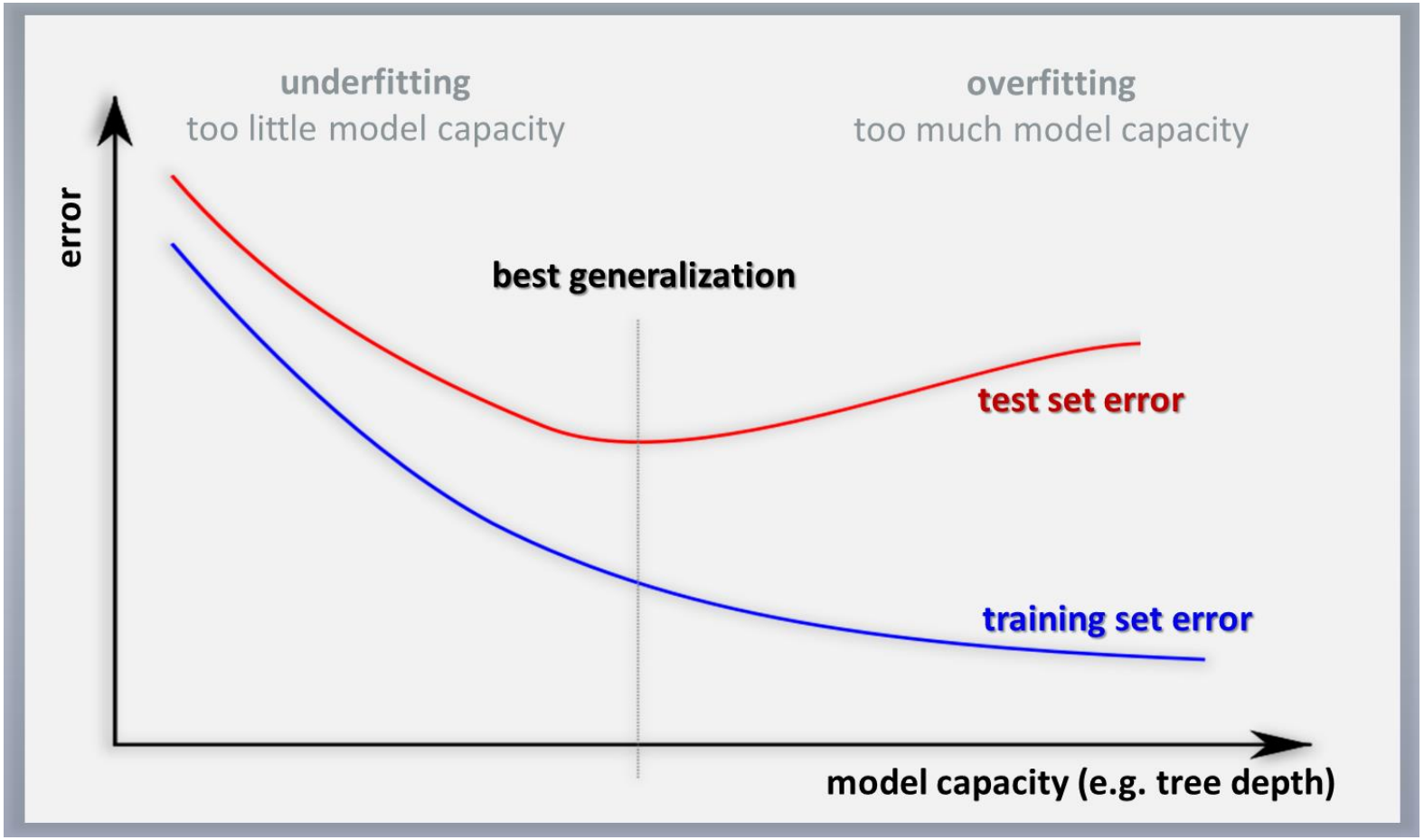
- Standard decision trees have no prior
  - Training set error is always zero!
    - (If there is no label noise)
  - Lots of variance
  - Will definitely overfit!!!
  - Must bias towards simpler trees

- Many strategies for picking simpler trees:
  - Fixed depth
  - Fixed number of leaves
  - Or something smarter… (chi2 tests)

Remember: Error Decomposition



Decision trees will overfit



Avoiding Overfitting

How can we avoid overfitting?

- Stop growing when data split not statistically significant
- Grow full tree, then post-prune

How to select "best" tree:

- Measure performance over training data
- Measure performance over separate validation data set
- Add complexity penalty to performance measure

Reduced-Error Pruning

Split data into *training* and *validation* set

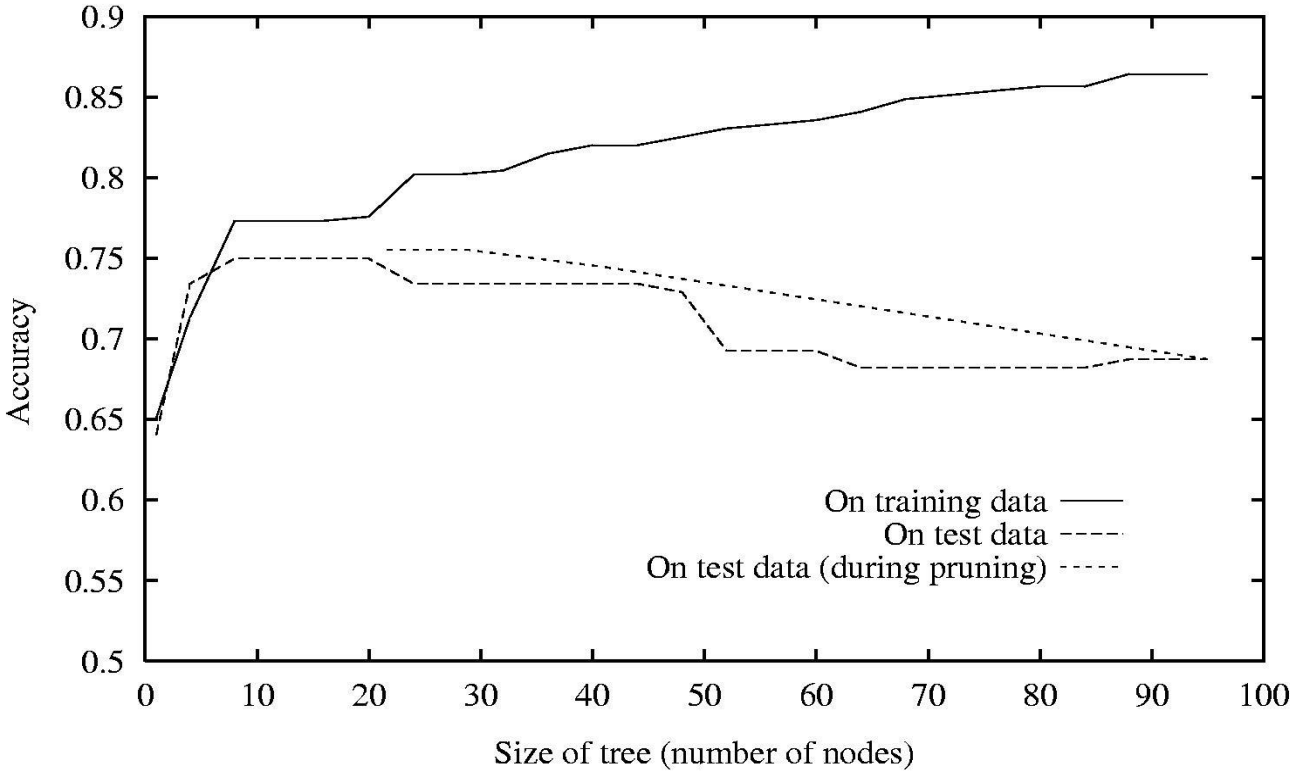
Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

Pruning Decision Trees

- Demo
  - http://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/Applet/DecisionTreeApplet.html

Effect of Reduced-Error Pruning



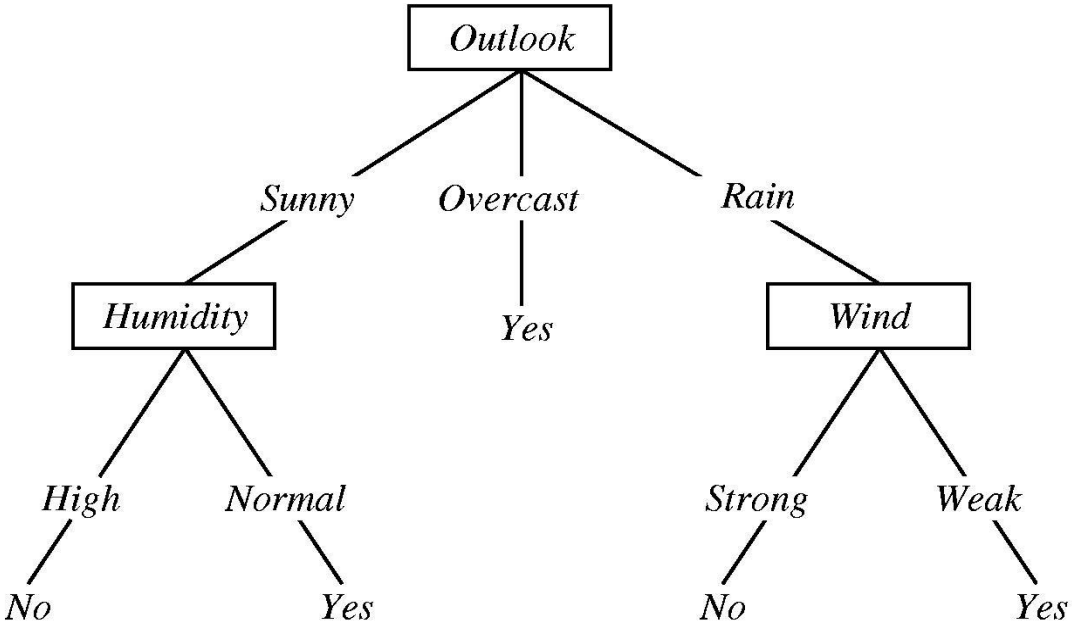
Slide Credit: Pedro Domingos, Tom Mitchel, Tom Dietterich

Rule Post-Pruning

1. Convert tree to equivalent set of rules
2. Prune each rule independently of others
3. Sort final rules into desired sequence for use

Perhaps most frequently used method (e.g., C4.5)

Converting A Tree to Rules



Slide Credit: Pedro Domingos, Tom Mitchel, Tom Dietterich

$$\begin{array}{ll} \text{IF} & (Outlook = Sunny)\ AND\ (Humidity = High) \\ \text{THEN} & PlayTennis = No \end{array}$$

$$\begin{array}{ll} \text{IF} & (Outlook = Sunny)\ AND\ (Humidity = Normal) \\ \text{THEN} & PlayTennis = Yes \end{array}$$

$\ldots$

Slide Credit: Pedro Domingos, Tom Mitchel, Tom Dietterich

Real-Valued inputs

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

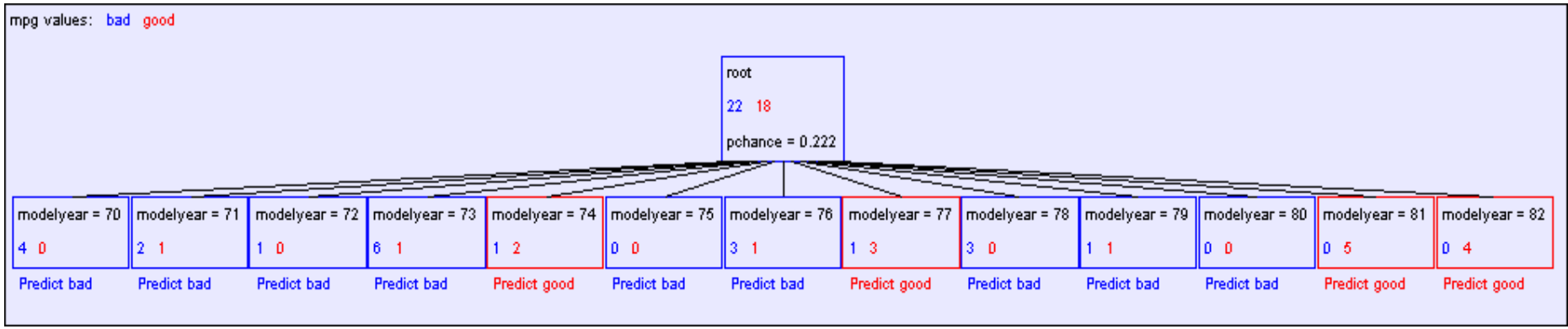
mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	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

Infinite number of possible split values!!!

Finite dataset, only finite number of relevant splits!

Idea One: Branch on each possible real value

“One branch for each numeric value” idea:



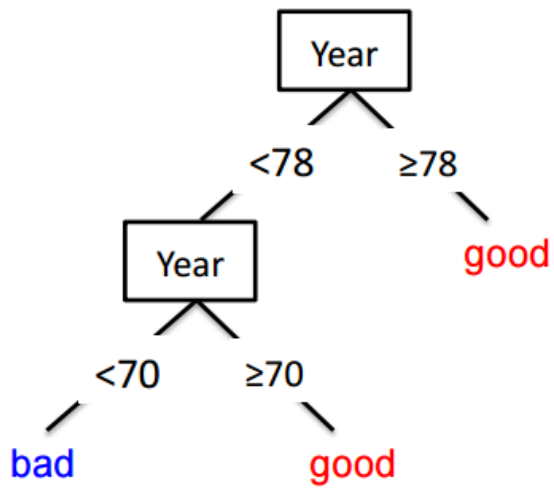
Hopeless: with such high branching factor will shatter the dataset and overfit

Threshold splits

- Binary tree, split on attribute X
  - One branch: X < t
  - Other branch: X >= t

Threshold splits

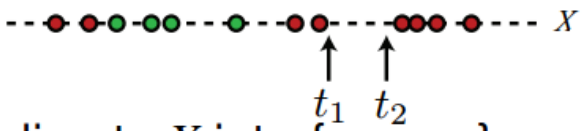
- Binary tree: split on attribute X at value t
  - One branch: $X < t$
  - Other branch: $X \geq t$
- Requires small change
  - Allow repeated splits on same variable **along a path**



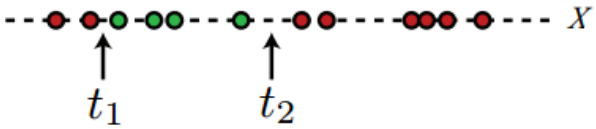
slide by David Sontag

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:



- Sort data according to X into $\{x_1,\ldots,x_m\}$
- Consider split points of the form $x_i + (x_{i+1} - x_i)/2$
- Moreover, only splits between examples from different classes matter!



slide by David Sontag

Choosing threshold split

- Binary tree, split on attribute X
  - One branch: X < t
  - Other branch: X >= t

- Search through possible values of *t*
  - Seems hard!!!

- But only finite number of *t*'s are important
  - Sort data according to X into $\{x_1,\ldots,x_n\}$
  - Consider split points of the form $x_i + (x_{i+1} - x_i)/2$

A better idea: thresholded splits

- Suppose X is real valued
- Define $IG(Y|X:t)$ as $H(Y) - H(Y|X:t)$
- Define $H(Y|X:t) =$

$$H(Y|X < t)\, P(X < t) + H(Y|X >= t)\, P(X >= t)$$

  - $IG(Y|X:t)$ is the information gain for predicting Y if all you know is whether X is greater than or less than t

- Then define $IG^*(Y|X) = max_t\, IG(Y|X:t)$

- For each real-valued attribute, use $IG^*(Y|X)$ for assessing its suitability as a split

- Note, may split on an attribute multiple times, with different thresholds

Example with MPG

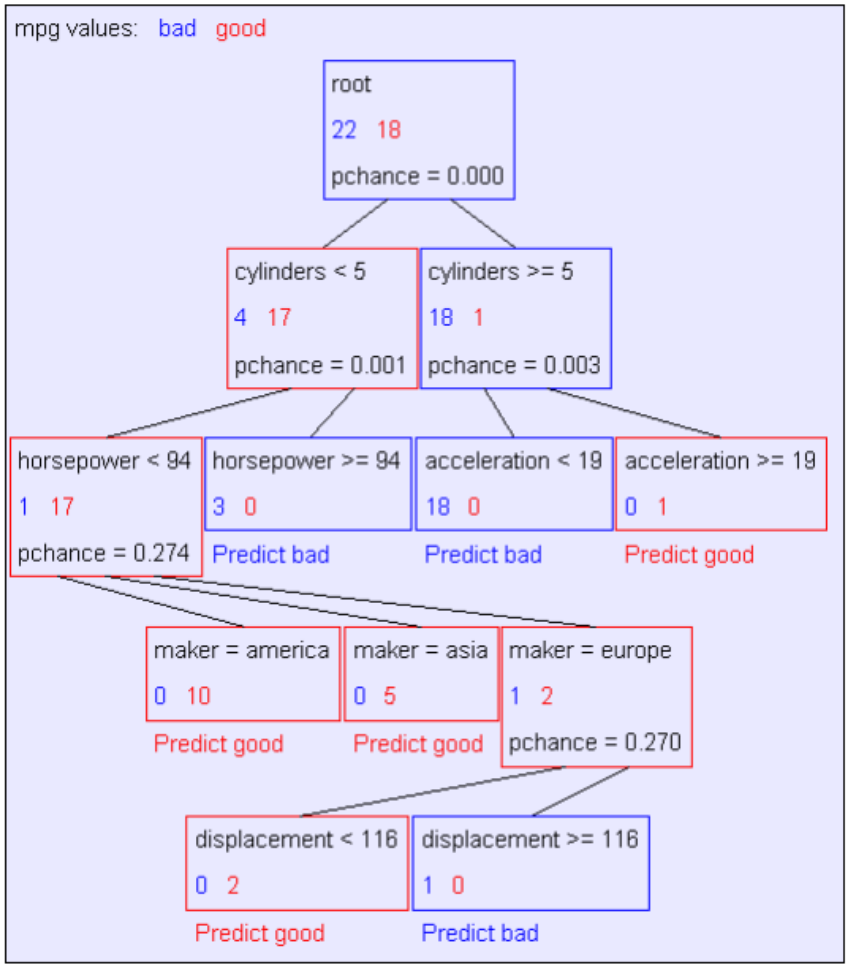
Information gains using the training set (40 records)

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		

slide by David Sontag

Example tree for our continuous dataset



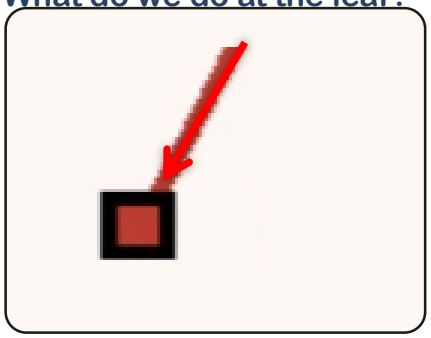
slide by David Sontag

Decision Trees

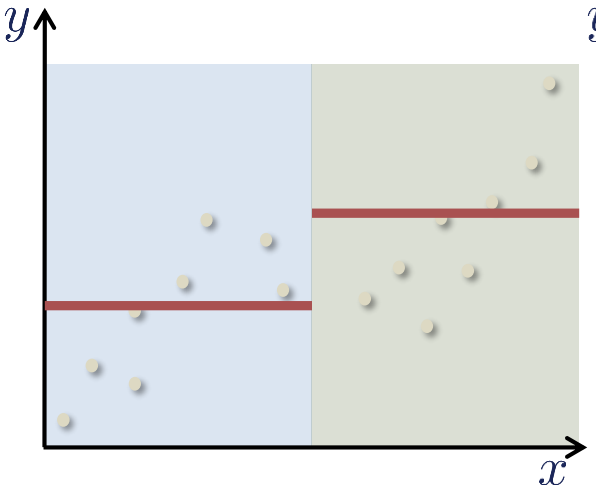
- Demo
  - http://www.cs.technion.ac.il/~rani/LocBoost/

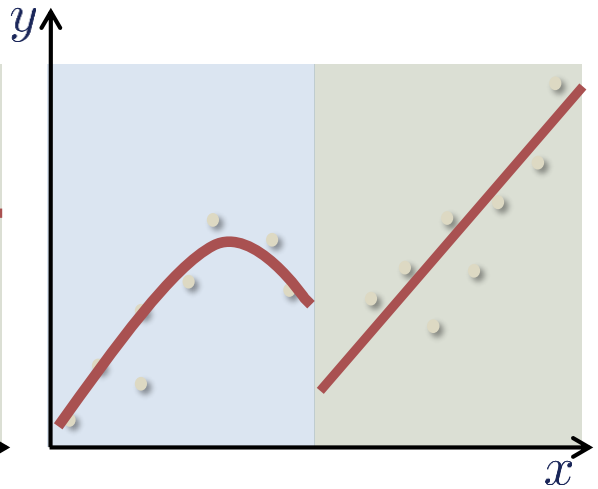
Regression Trees

What do we do at the leaf?



Examples of leaf (predictor) models





Predictor model: constant

$$y = const$$

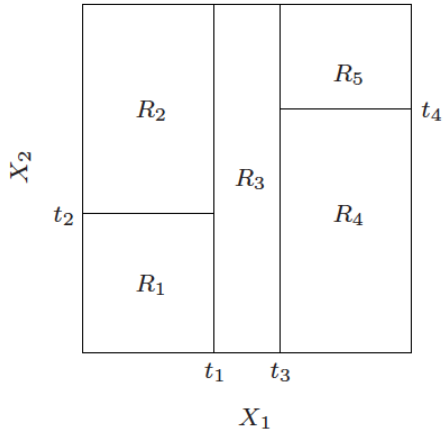
Predictor model: polynomial

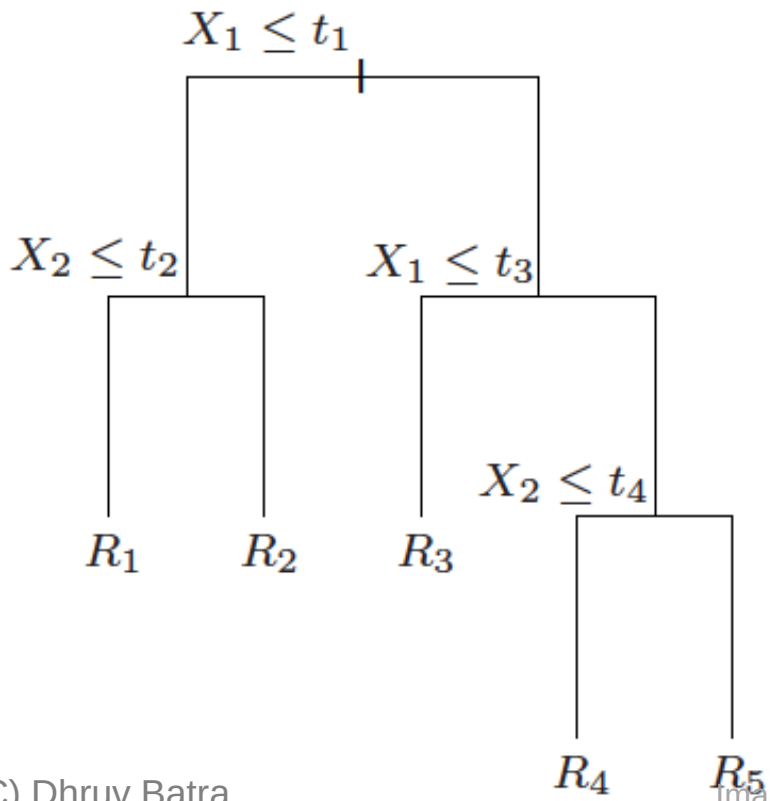
$$y = \sum_{i=0}^{n} w_i x^i$$

(note: linear for n=1, constant for n=0)

Image Credit: Jamie Shotton

Regression Trees





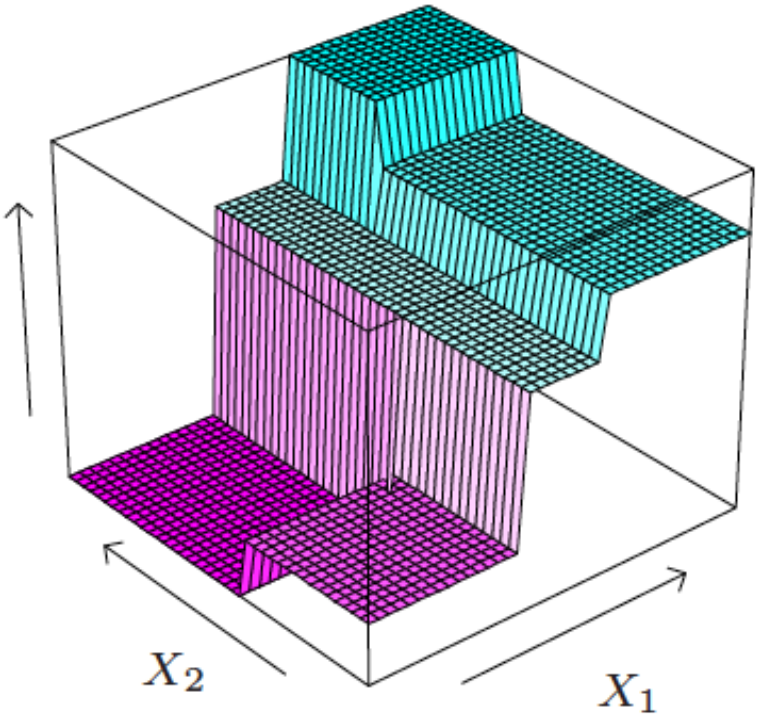
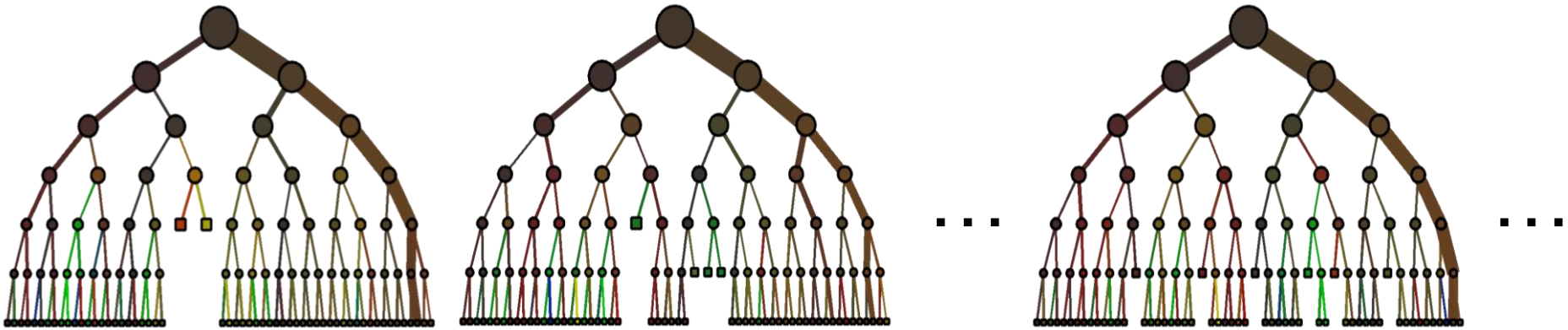


Image Credit: ESL Book

Decision Forests



Learn many trees & Average Outputs

Will formally visit this in Bagging lecture

Image Credit: Jamie Shotton

What you need to know about decision trees

- Decision trees are one of the most popular data mining tools
  - Easy to understand
  - Easy to implement
  - Easy to 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!!!
  - Zero bias classifier → Lots of variance
  - Must use tricks to find “simple trees”, e.g.,
    - Fixed depth/Early stopping
    - Pruning
    - Hypothesis testing