CSE 446: Week 2
Decision Trees

Administrative

- Homework goes out today, please contact Isaac Tian (iytian@cs.washington.edu) if you have not been added to Gradescope

Recap: Algorithm

until Base Case 1 or Base Case 2 is reached:

  step over each leaf

    step over each attribute X

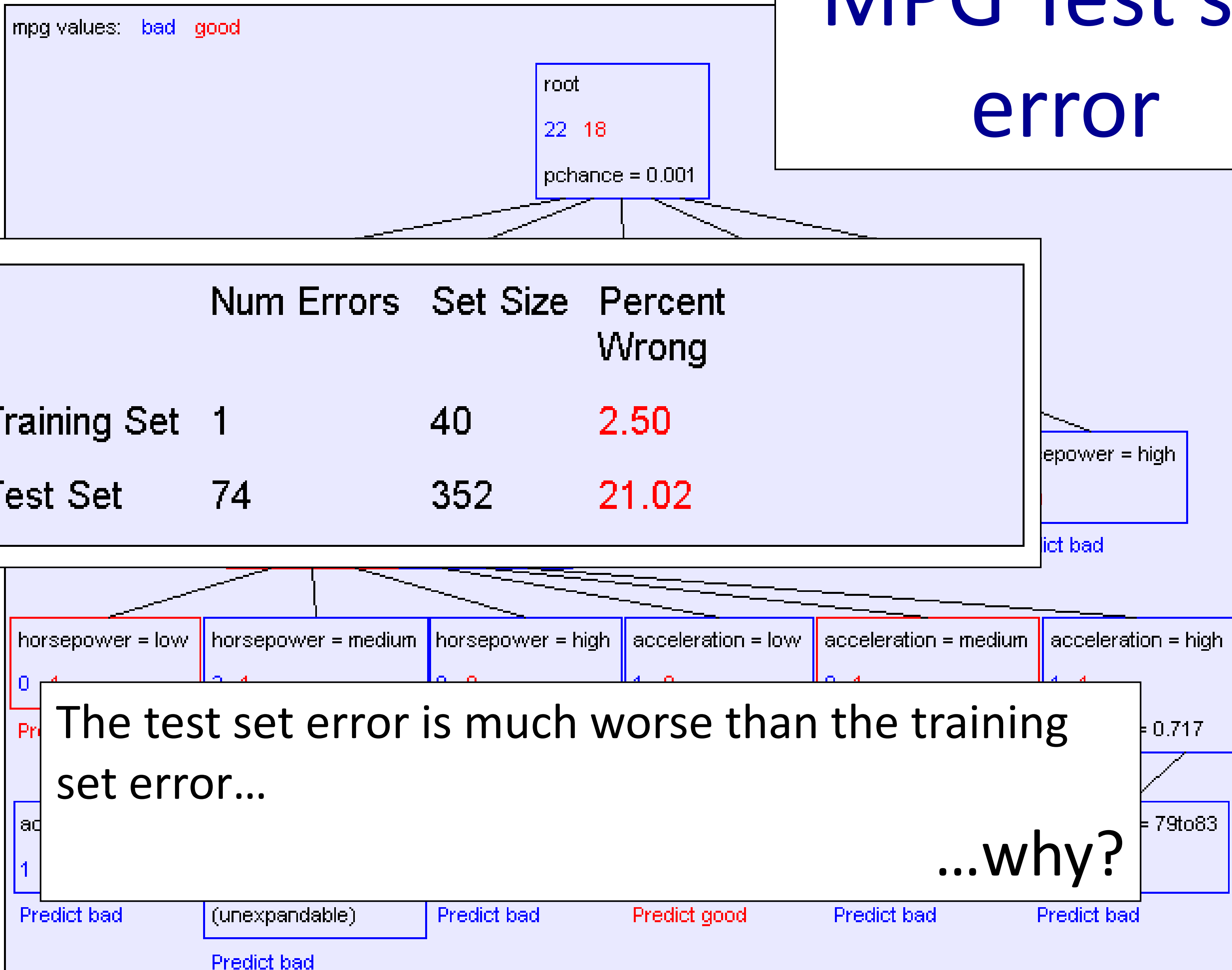
      compute IG(X)

  choose leaf & attribute with highest IG

  split that leaf on that attribute

  repeat

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…

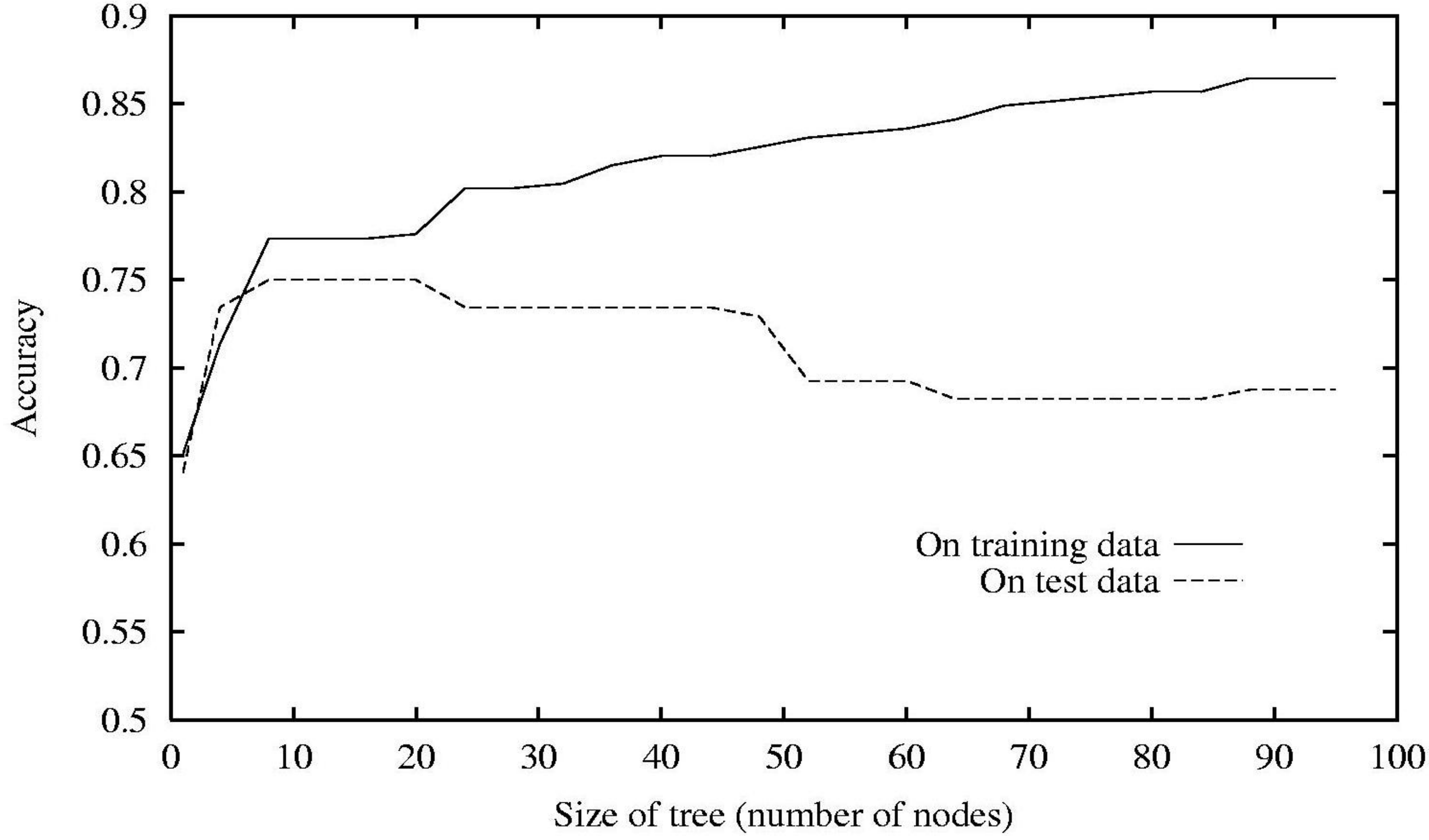
…why?

Decision trees will overfit!!!

- 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
  - Fixed number of leaves
  - Or something smarter…

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Decision trees will overfit!!!



One Definition of Overfitting

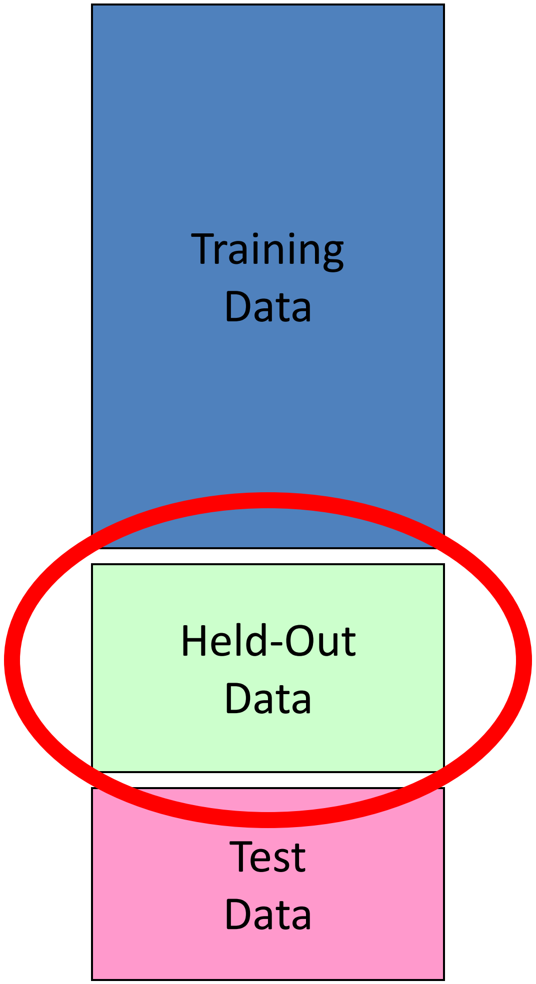
- Assume:
  - Data generated from distribution $D(X,Y)$
  - A hypothesis space H
- Define errors for hypothesis $h \in H$
  - Training error: $error_{train}(h)$
  - Data (true) error: $error_D(h)$
- We say h ***overfits*** the training data if there exists an $h' \in H$ such that:

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

and

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

Recap: Important Concepts



Pruning Decision Trees

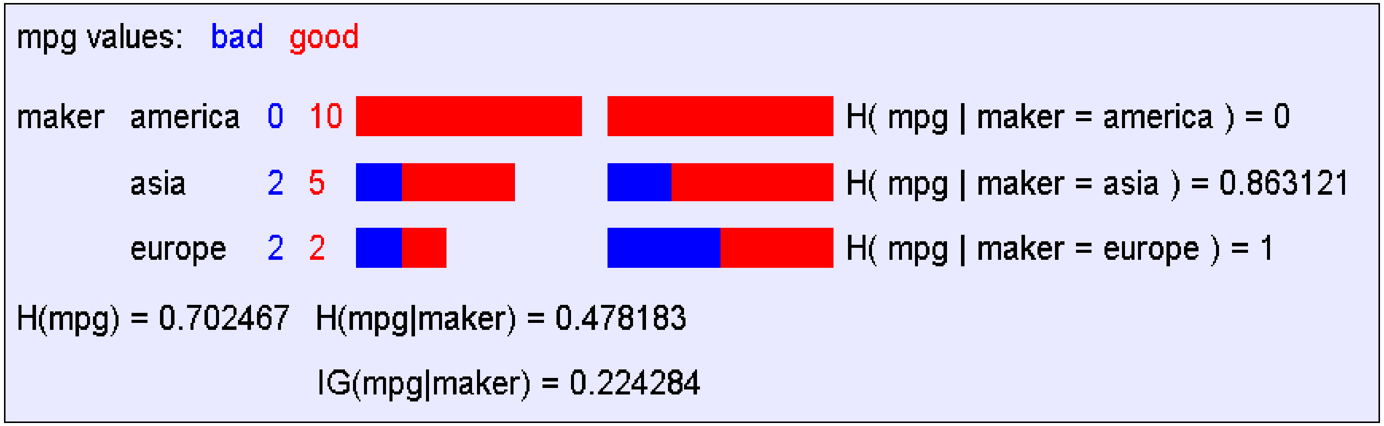
[tutorial on the board]

[see lecture notes for details]

IV. Overfitting idea #1: holdout cross-validation

V. Overfitting idea #2: Chi square test

A Chi Square Test



- Suppose that mpg was completely uncorrelated with maker.
- What is the chance we'd have seen data of at least this apparent level of association anyway?

By using a particular kind of chi-square test, the answer is
g((x1, y1) … (xn, yn)) = 13.5%

We will not cover Chi Square tests in class. See page 93 of the original ID3 paper [Quinlan, 86].

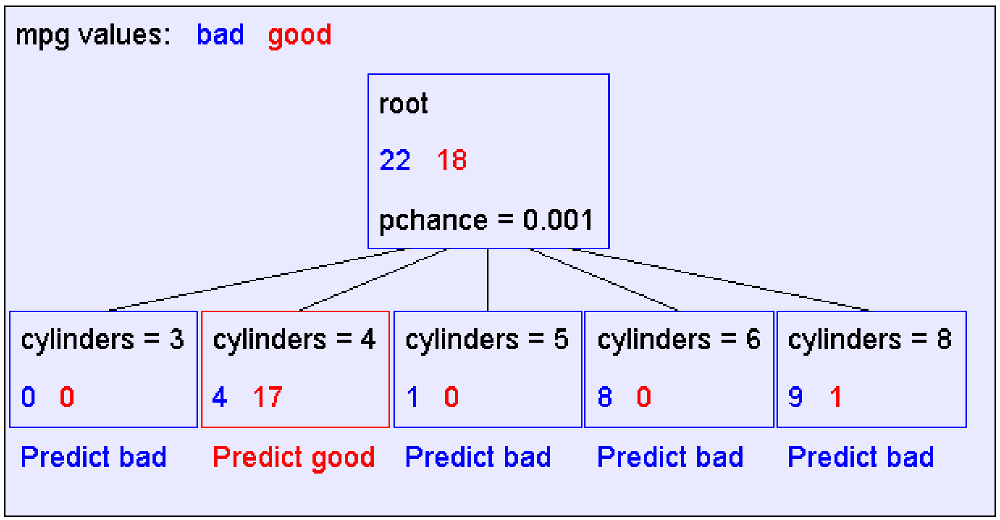
Using Chi-squared to avoid overfitting

- Build the full decision tree as before
- But when you can grow it no more, start to prune:
  - Beginning at the bottom of the tree, delete splits in which $g((x1,y1),\ldots,(xn,yn)) > MaxPchance$
  - Continue working you way up until there are no more prunable nodes

MaxPchance is a magic parameter you must specify to the decision tree, indicating your willingness to risk fitting noise

Pruning example

- With MaxPchance = 0.05, you will see the following MPG decision tree:



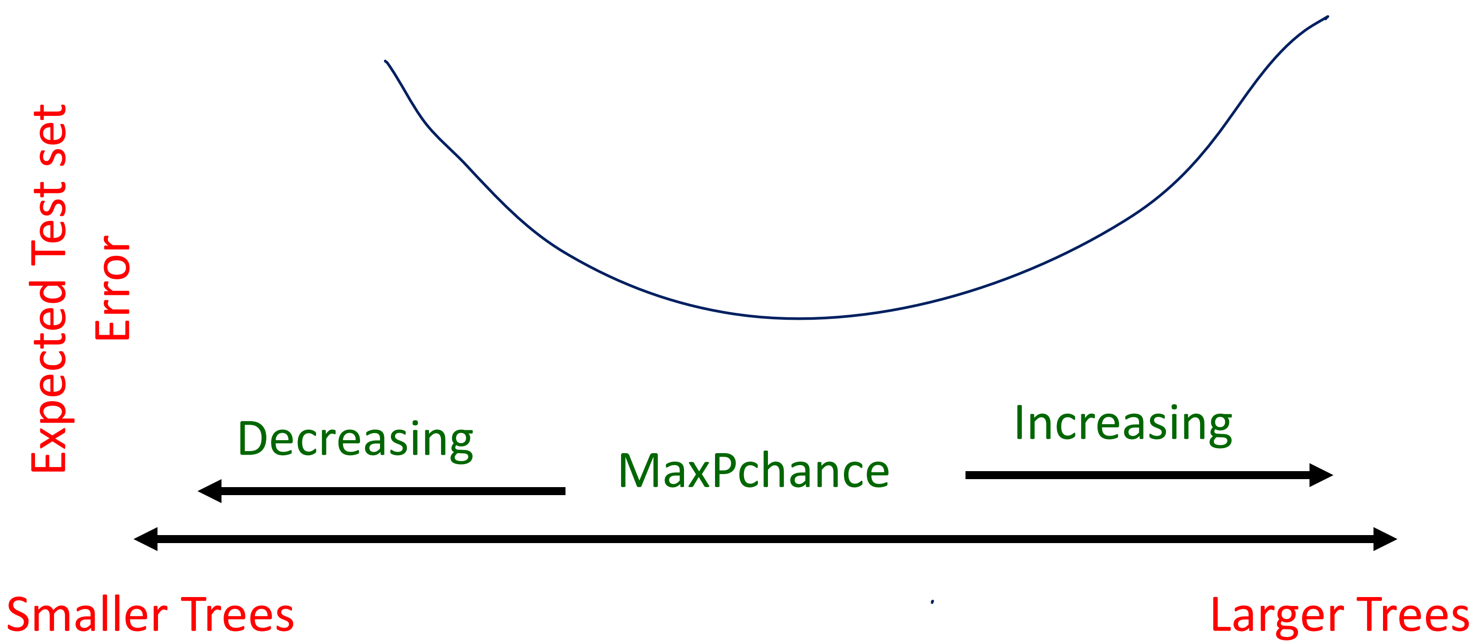
When compared to the unpruned tree

- improved test set accuracy
- worse training accuracy

	Num Errors	Set Size	Percent Wrong
Training Set	5	40	12.50
Test Set	56	352	15.91

MaxPchance

- Technical note: MaxPchance is a regularization parameter that helps us bias towards simpler models



We'll learn to choose the value of magic parameters like this one later!

Real-Valued inputs

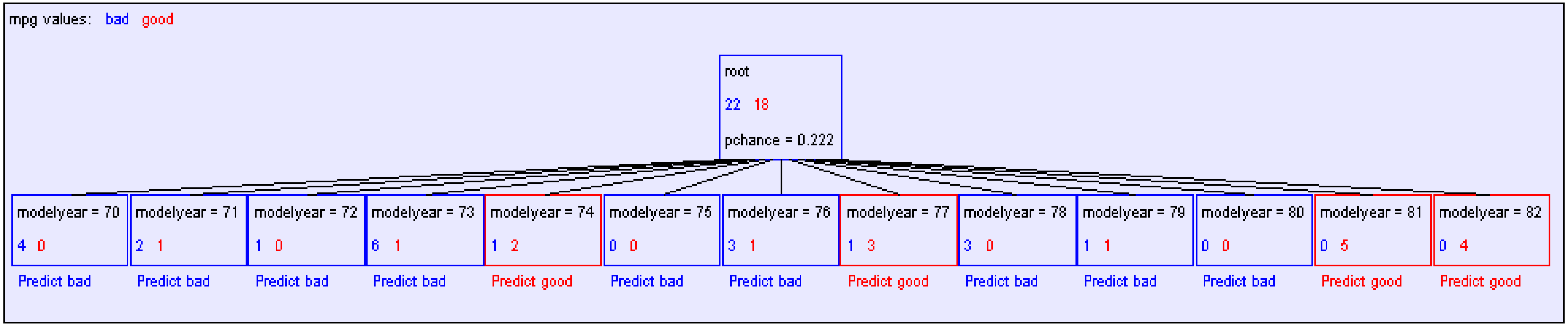
What should we do if some of the inputs are real-valued?

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

“One branch for each numeric value” idea:

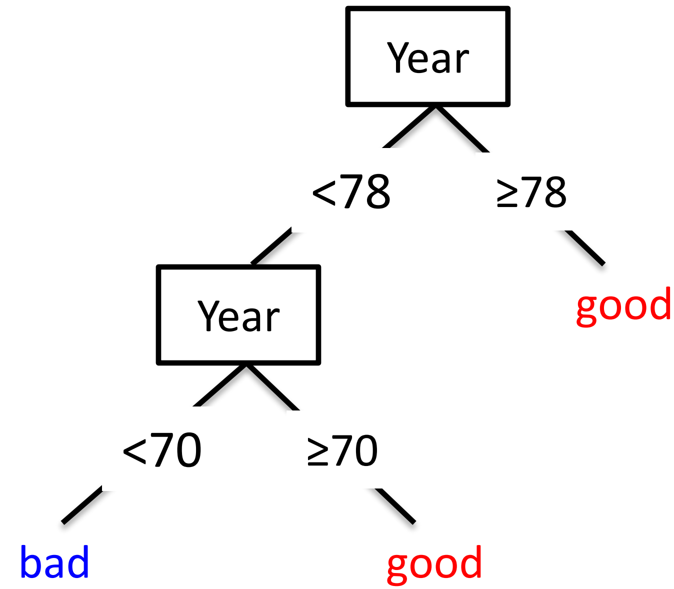


Hopeless: with such high branching factor will shatter the dataset and overfit

Threshold splits

- Binary tree: split on attribute X 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?



The set of possible thresholds

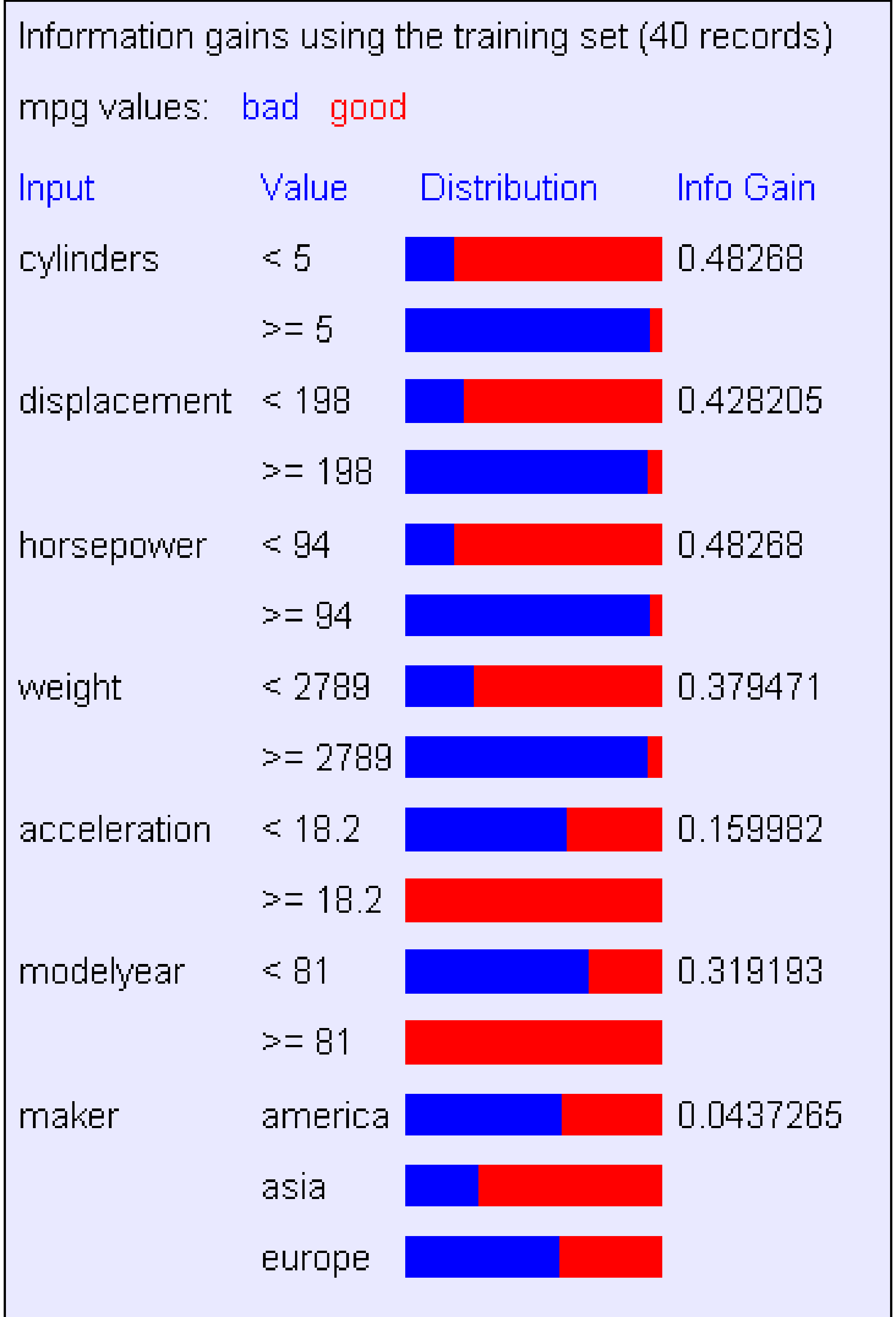
- Binary tree, split on attribute X
  - One branch: $X < t$
  - Other branch: $X \geq t$
- Search through possible values of *t*
  - Seems hard!!!
- But only finite number of *t*'s are important
  - Sort data according to X into $\{x_1, \ldots, x_m\}$
  - Consider split points of the form $x_i + (x_{i+1} - x_i)/2$

Picking the best threshold

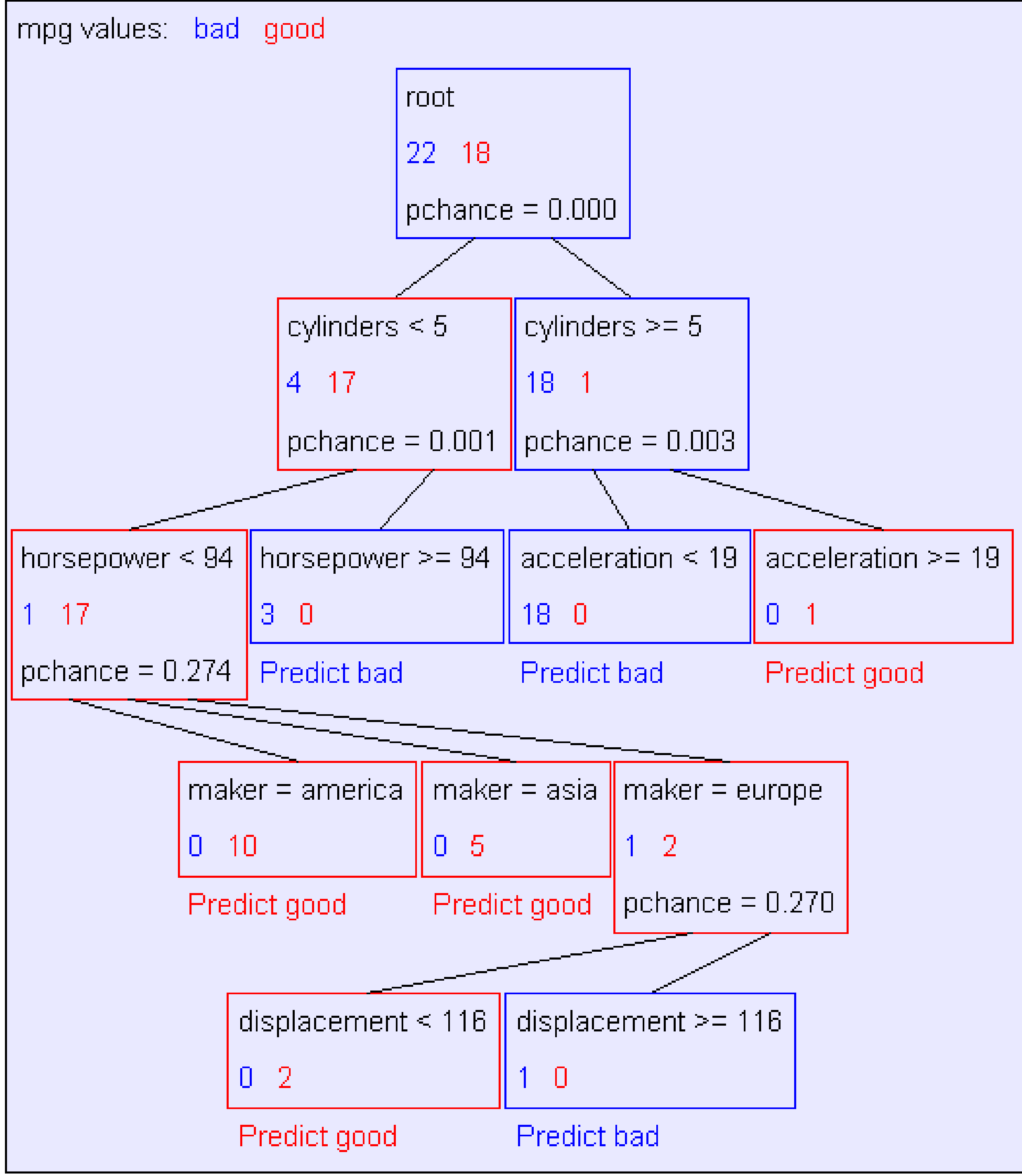
- Suppose *X* is real valued with threshold *t*
- Want *IG(Y|X:t)*: the information gain for Y when testing if *X* is greater than or less than *t*
- Define:
  - $H(Y|X:t) =$

    $$H(Y|X < t)\, P(X < t) + H(Y|X >= t)\, P(X >= t)$$
  - $IG(Y|X:t) = H(Y) - H(Y|X:t)$
  - $IG^*(Y|X) = max_t\, IG(Y|X:t)$
- Use: *IG*(Y|X)* for continuous variables

Example with MPG



Example tree for our continuous dataset



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 (ID3, C4.5,…)
- 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
    - Pruning
    - Hypothesis testing