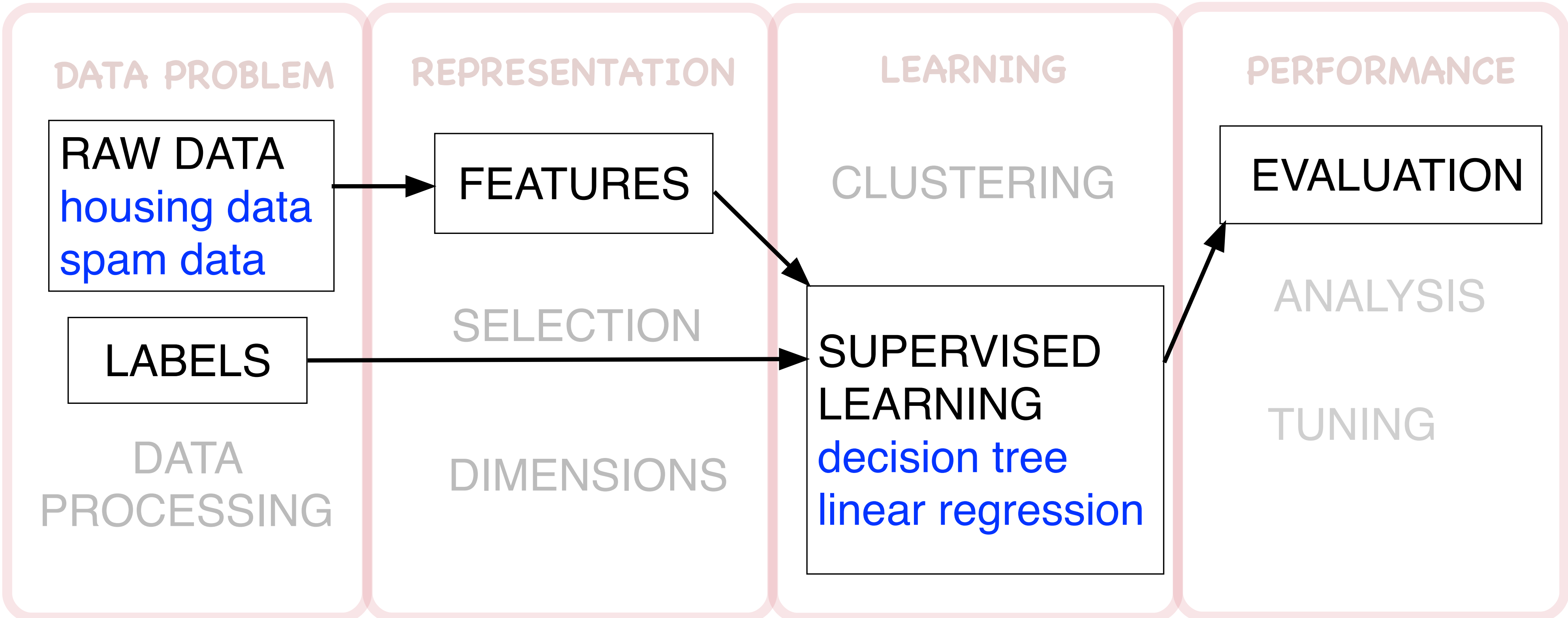
Decision Trees

some slides/drawings thanks to Carlos Guestrin@CMU

Course Map / module1

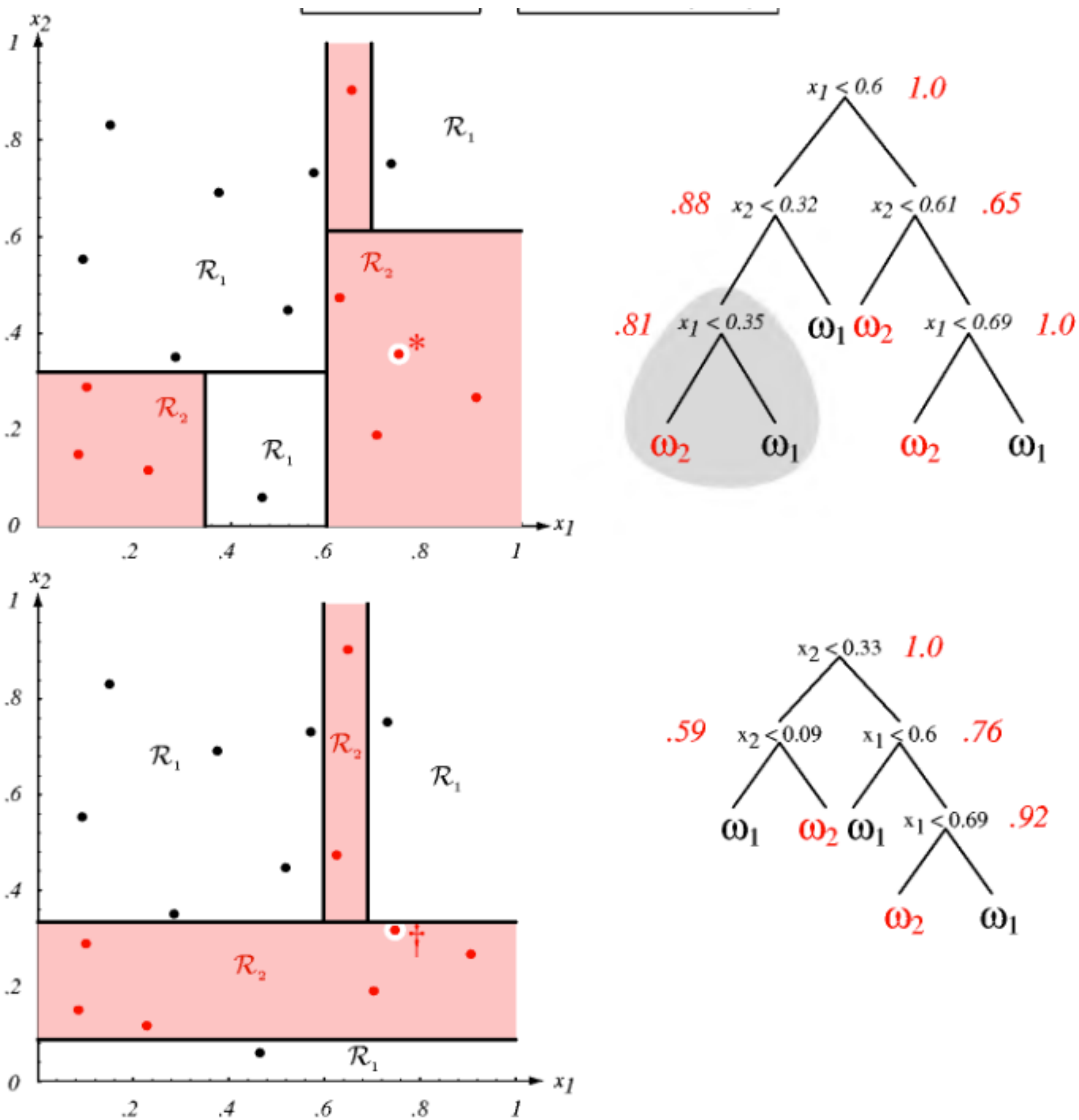


- two basic supervised learning algorithms
  - decision trees
  - linear regression
- two simple datasets
  - housing
  - spam emails

Module 1 Objectives / Decision Trees

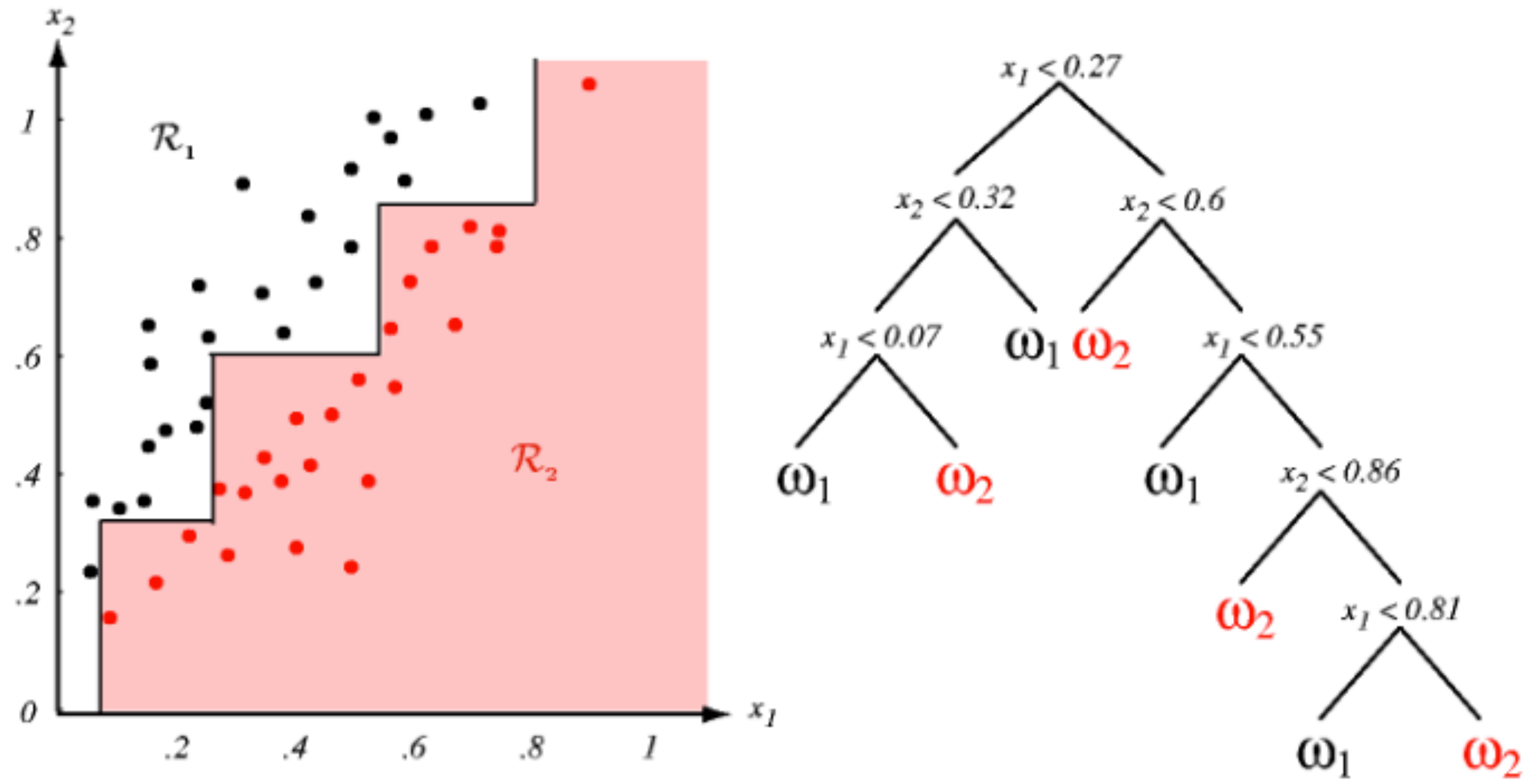
- Decision Trees
- Splitting Criteria
  - decision stumps
  - how to look for the best splits
- Regression Trees
  - regression criteria
- Run a Decision Tree in practice
- Pruning

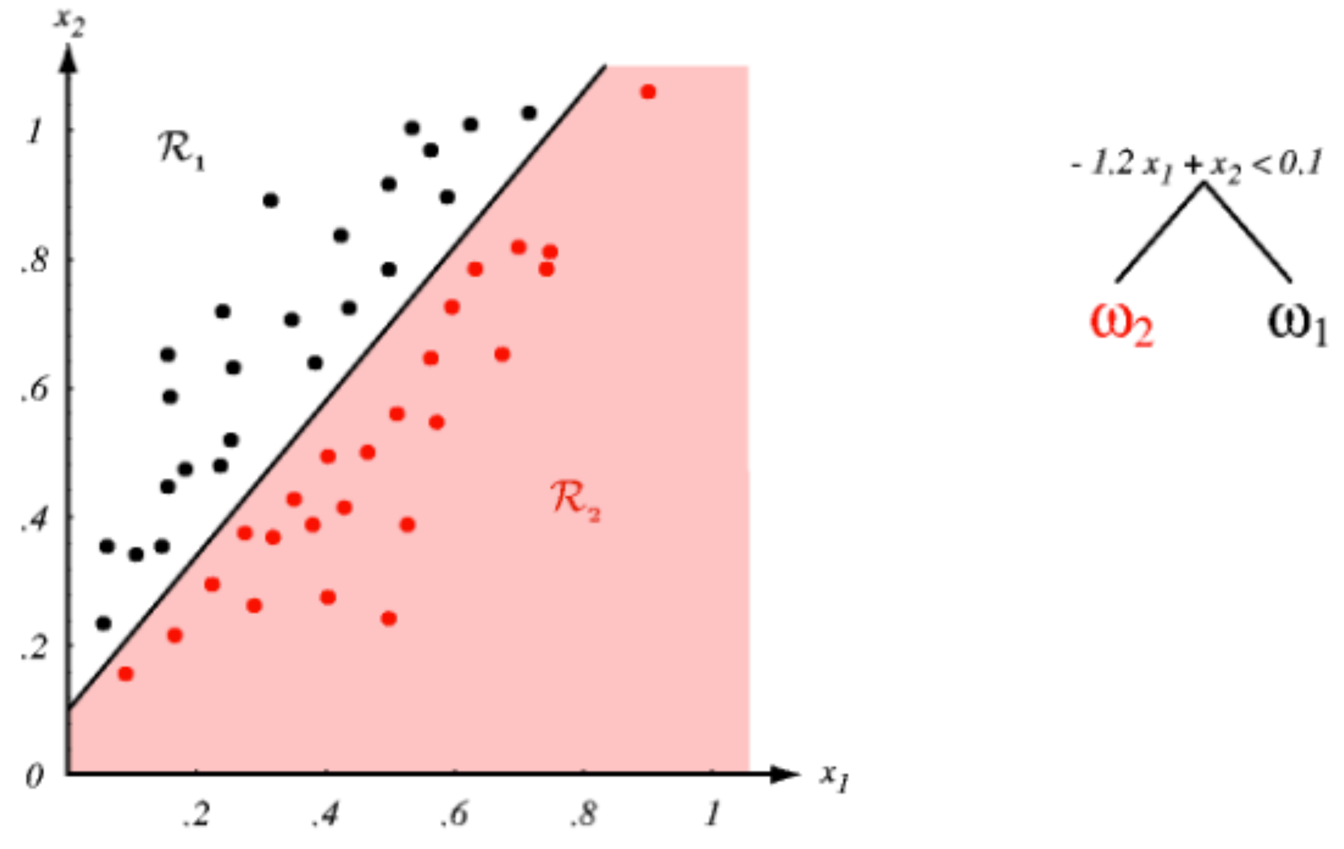
Data Partition Rules



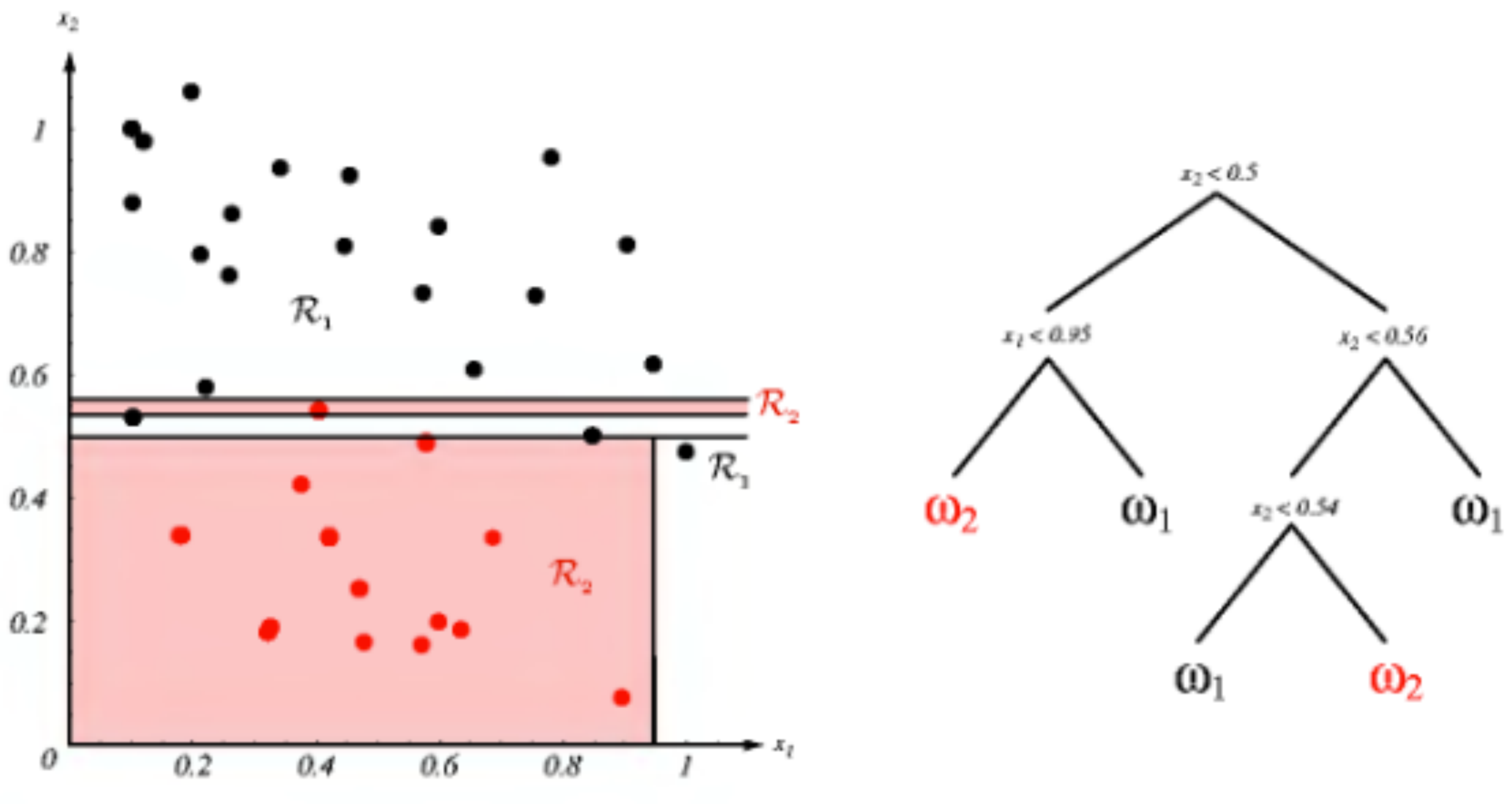
- x_1,x_2 = data features
- Each path in the tree corresponds to a region
- Deeper paths correspond to smaller regions

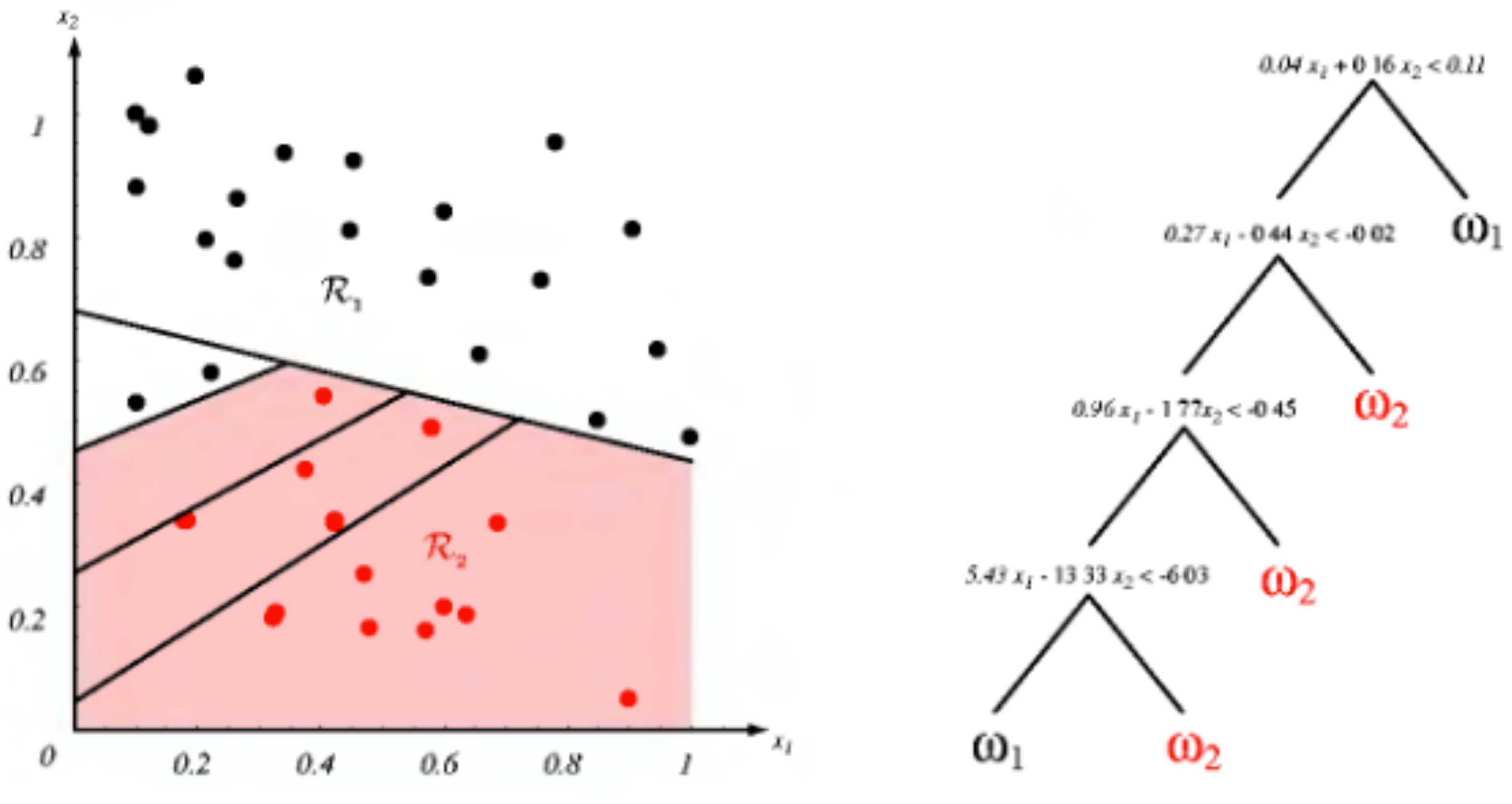
Data Partition Rules





Data Partition Rules





Decision Trees

- Goal: Learn from training set a decision tree
  - initially all training datapoints at root
- iterative splits:
  - pick a terminal node (leaf) with inconsistent labels
  - use a split criteria to branch data so that each resulting child node has [more] consistent labels
  - until no terminal nodes are inconsistent
- Use learned tree for prediction on the test set

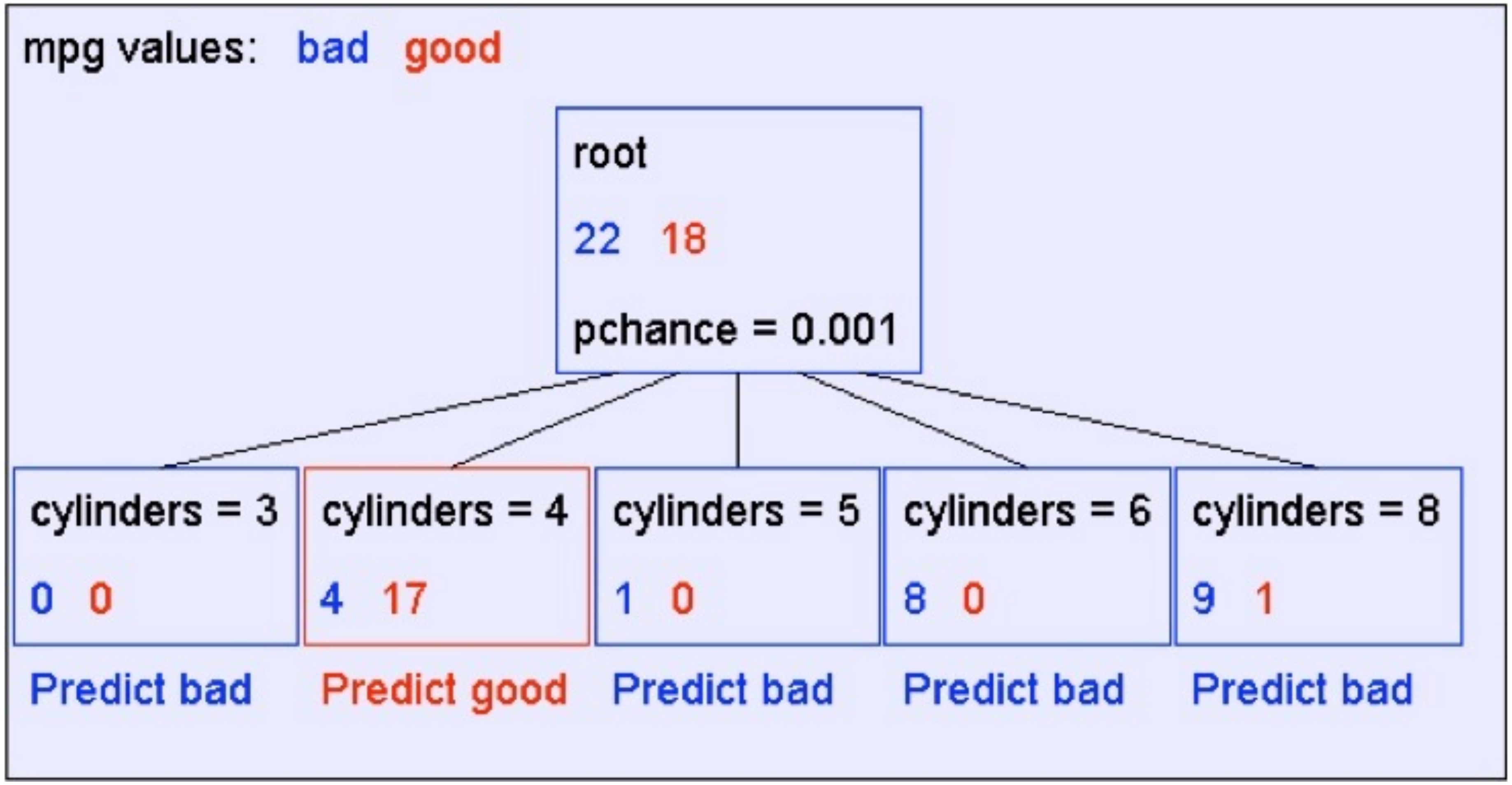
Walkthrough Decision Tree Example

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

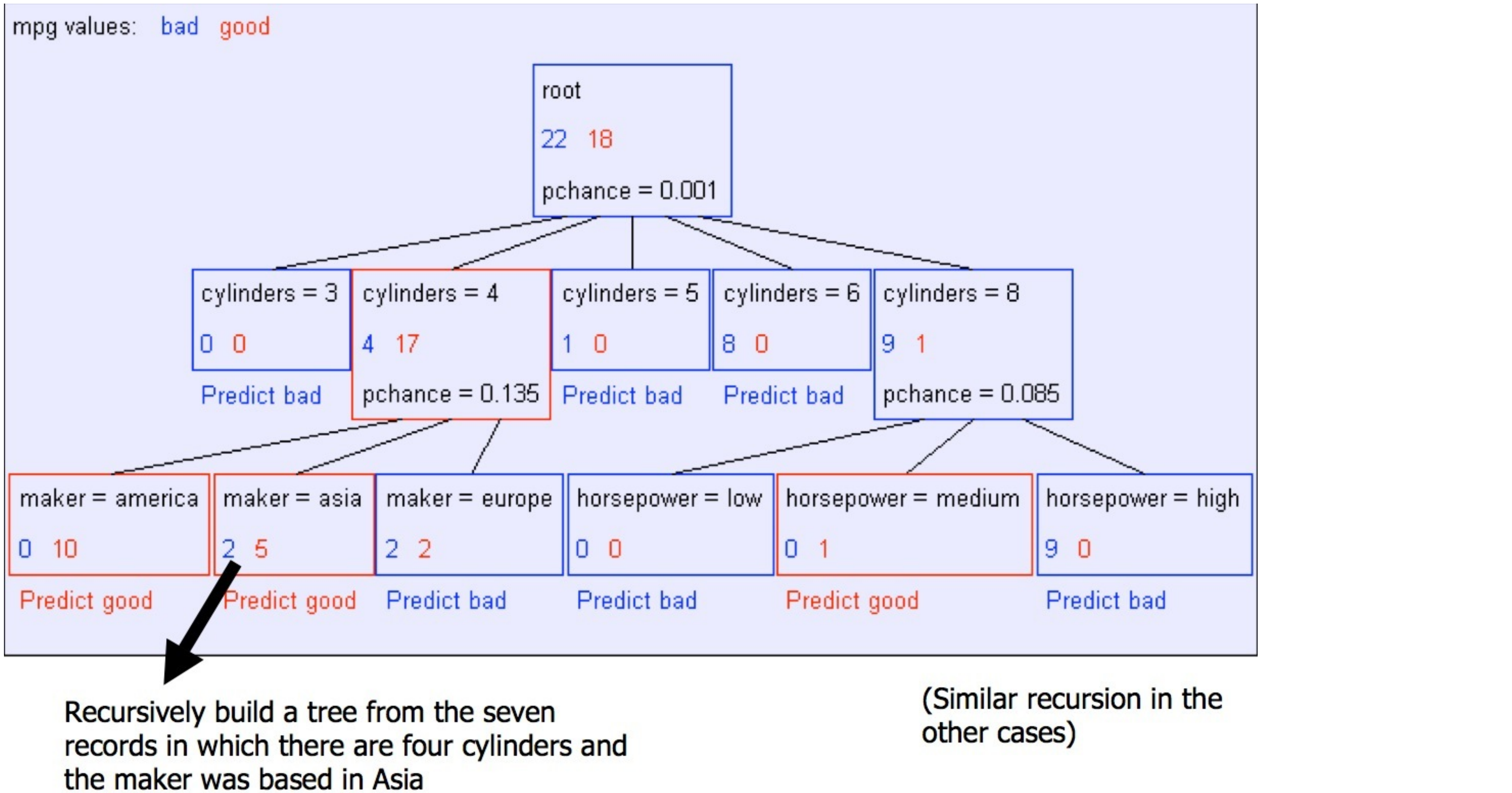
- Data (matrix) example : automobiles
- Target : mpg $\in$ {good, bad} - 2 class /binary problem

Decision Tree Split



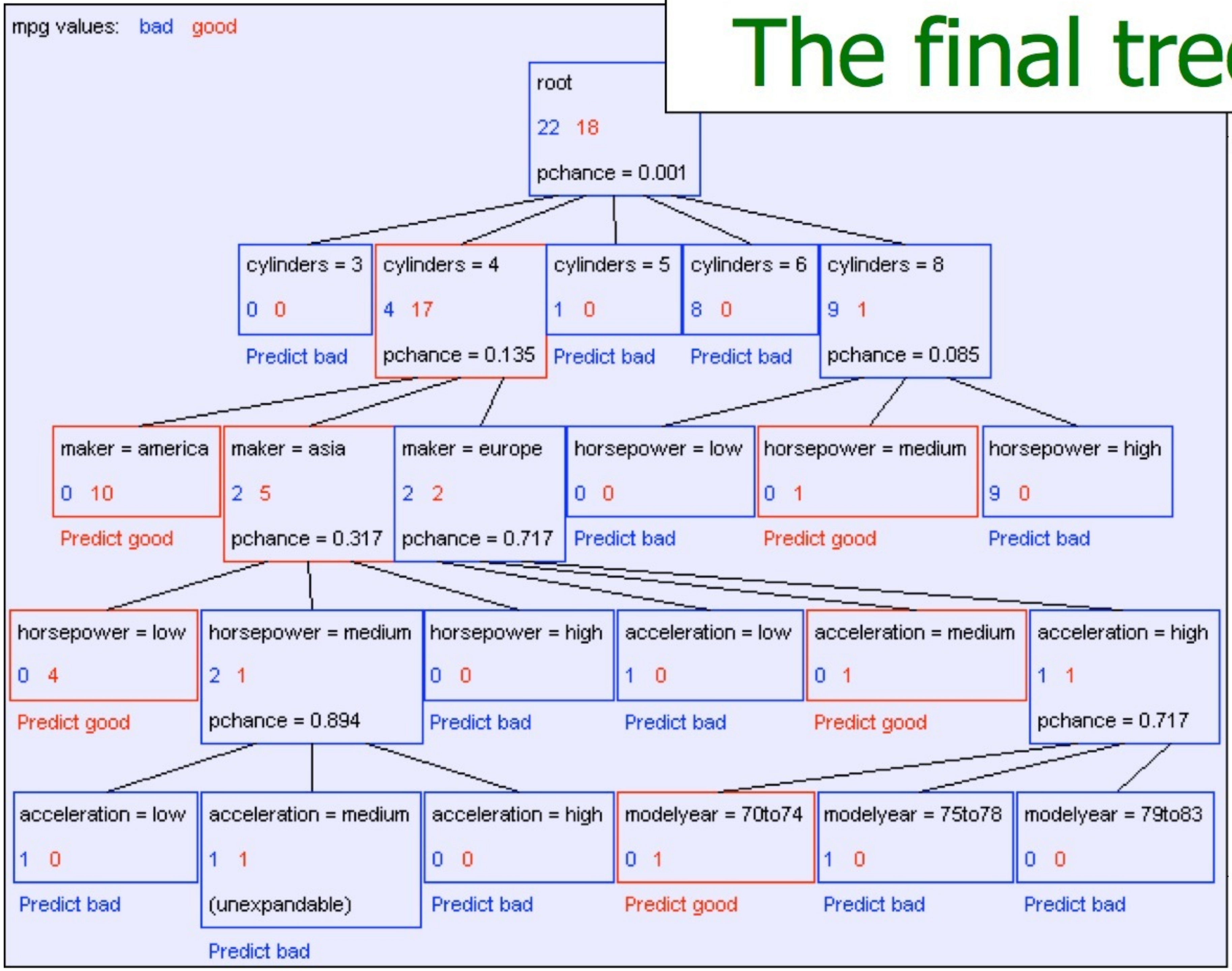
- Split by feature “cylinders”, using feature values for branches

Decision Tree Splits



- each terminal leaf is labeled by majority (at that leaf). This leaf-label is used for prediction.

Decision Tree Splits



The final tree

Splitting criteria: entropy-based gain

$$H(Y) = \sum_j P(y_j) \log_2(\frac{1}{P(y_j)})$$

Entropy after split by X feature

$$H(Y|X) = \sum_i P(x_i) \sum_j P(y_j|x_i) \log_2(\frac{1}{P(y_j|x_i)})$$

Mutual information (or Information Gain).

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

- Y = labels random variable, H(Y) its entropy
- X is a feature of the data used for splitting

Entropy gain toy example

At each split we are going to choose the feature that gives the highest information gain.

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

Figure 6: 2 possible features to split by

$$H(Y|X^1) = \frac{1}{2}H(Y|X^1=T) + \frac{1}{2}H(Y|X^1=F) = 0 + \frac{1}{2}(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}) \approx .405$$

$$IG(X^1) = H(Y) - H(Y|X^1) = .954 - .405 = .549$$

$$H(Y|X^2) = \frac{1}{2}H(Y|X^2=T) + \frac{1}{2}H(Y|X^2=F) = \frac{1}{2}(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}) + \frac{1}{2}(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}) \approx .905$$

$$IG(X^2) = H(Y) - H(Y|X^2) = .954 - .905 = .049$$

checkpoint: information gain

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

- compute the information gain for f=cylinders and for f=displacement
- once a split by f=cylinders is performed, for the branch “cylinders=4” compute the information gain for f=displacement and for f=maker

Regression Tree

- same tree structure, split criteria
- assume numerical labels
- for each terminal node compute the node label (predicted value) and the mean square error

Estimate a predicted value per tree node

$$g_m = \frac{\sum_{t \in \chi_m} y_t}{|\chi_m|}$$

Calculate mean square error

$$E_m = \frac{\sum_{t \in \chi_m} (y_t - g_m)^2}{|\chi_m|}$$

- choose a split criteria to minimize the weighted error at children nodes

Regression Tree

labels: 1, 2, 2, 3, 10, 12, 14, 15

$$g = \frac{1+2+2+3+10+12+14+15}{8} = 7.37$$
$$Error = \sum_i (label_i - g)^2 = 247.87$$

labels: 1, 2, 2, 3

$$g = \frac{1+2+2+3}{4} = 2$$
$$Error = \sum_i (label_i - g)^2 = 2$$

labels: 10, 12, 14, 15

$$g = \frac{10+12+14+15}{4} = 12.75$$
$$Error = \sum_i (label_i - g)^2 = 14.75$$

- choose a split criteria to minimize the weighted or total error at children nodes
  - in the example total error after the split is 14.75 + 2=16.75

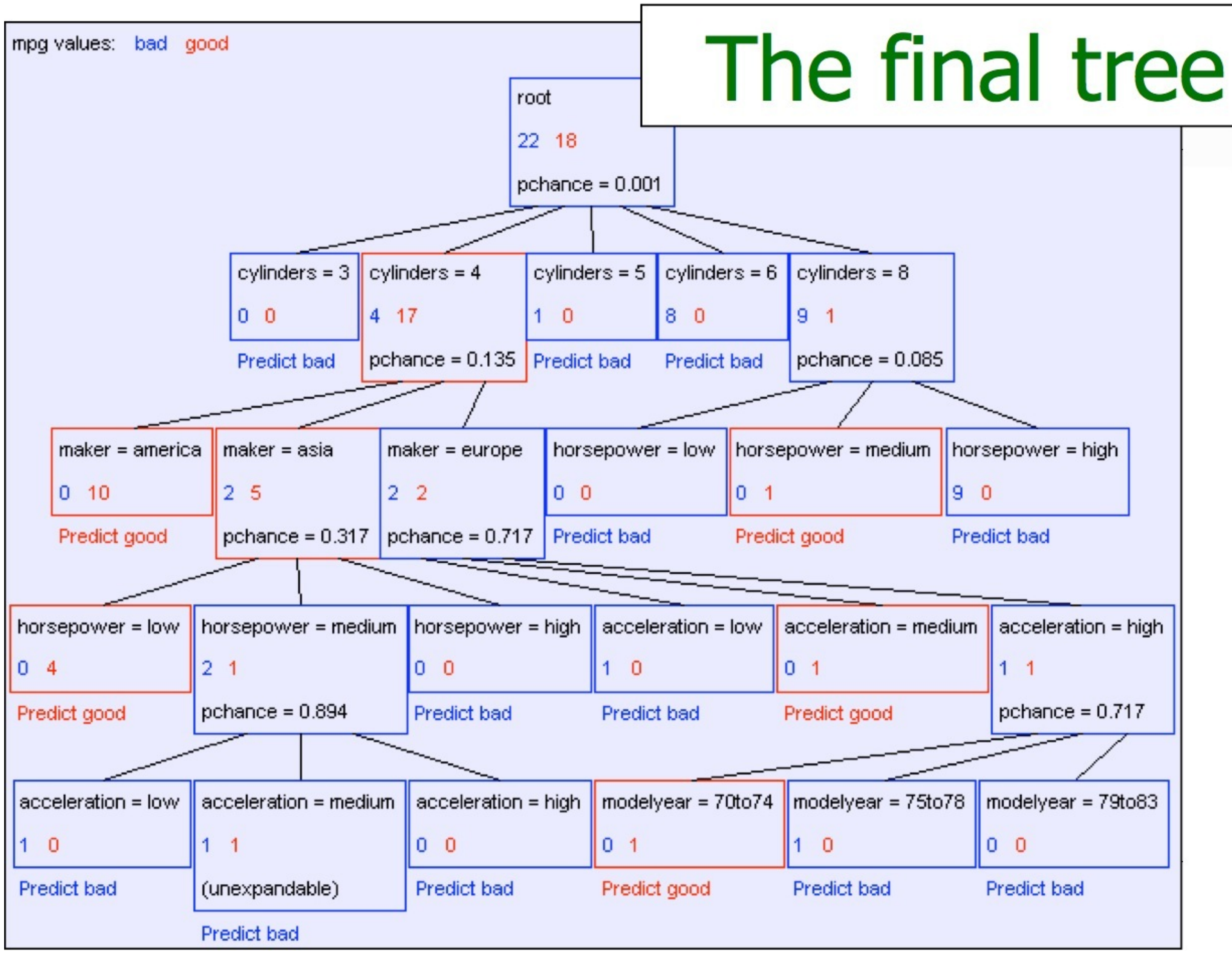
Prediction with a tree

- for each test datapoint $x=(x^1,x^2,\ldots,x^d)$ follow the corresponding path to reach a terminal node n

- predict the value/label associated with node n

Prediction with a tree

- testpoint:
  - cylinder=4
  - maker=asia
  - horsepower=low
  - weight=low
  - displacement=medium
  - modelyear=75to78



Overfitting

- decision trees can overfit quite badly
  - in fact they are designed to do so due to high complexity of the produced model
  - if a decision tree training error doesn't approach zero, it means that data is inconsistent
  -
- some ideas to prevent overfitting:
  - create more than one tree, each using a different subset of features; average/vote predictions
  - do not split nodes in the tree that have very few datapoints (for example less than 10)
  - only split if the improvement is massive

Pruning

- done also to prevent overfitting
- construct a full decision tree
- then walk back from the leaves and decide to "merge" overfitting nodes
  - when split complexity overwhelms the gain obtained by the spit

tree implementation

- perl/python : easy to use a hash
- matlab : use a vector/matrix
- C/Java: use a struct/object with pointers to children nodes.

Decision Tree Screencast

- http://www.screencast.com/t/J0jLmCdBW0M6