

# Information Extraction

Lecture 5 – Decision Trees (Basic Machine Learning)

CIS, LMU München

Winter Semester 2023-2024

Prof. Dr. Alexander Fraser, CIS

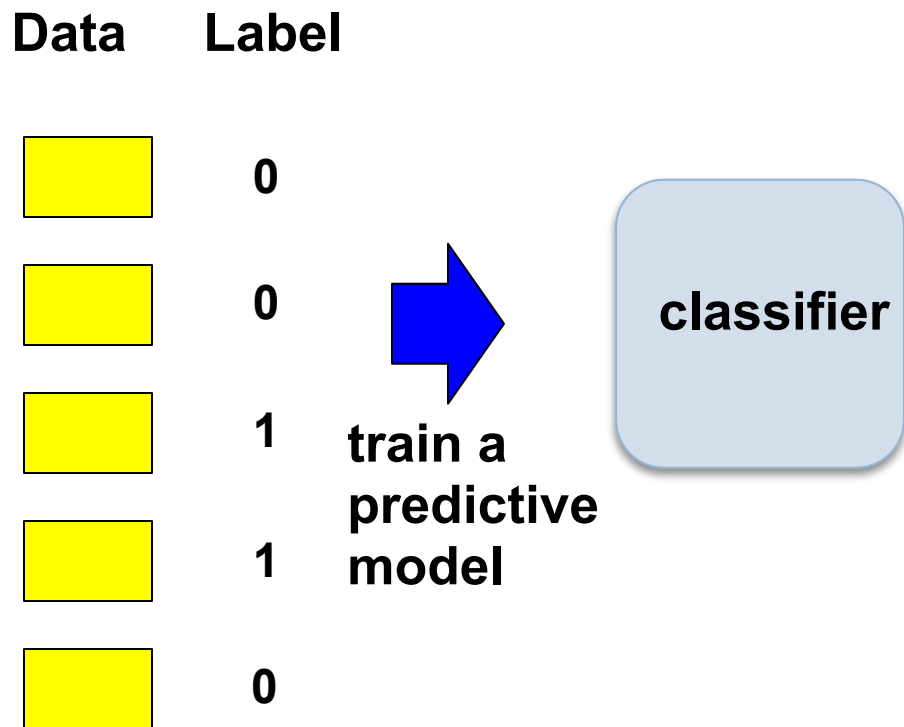
# Lecture Today

- Last week, we covered a basic idea of how NER classification works
  - We talked about sliding windows and features
  - Today I will complete this introduction by comparing sliding windows with boundary detection
- After that, we'll spend the rest of the lecture on an important topic in machine learning

# How can we pose this as a classification (or learning) problem?

... 00 : pm Place : Wean Hall Rm 5409 Speaker : Sebastian Thrun ...

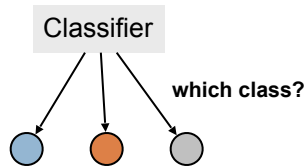
prefix                      contents                      suffix



# Lots of possible techniques

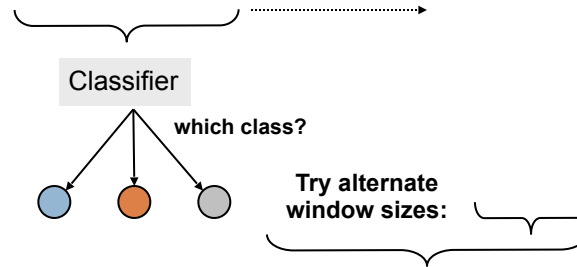
## Classify Candidates

Abraham Lincoln was born in Kentucky.

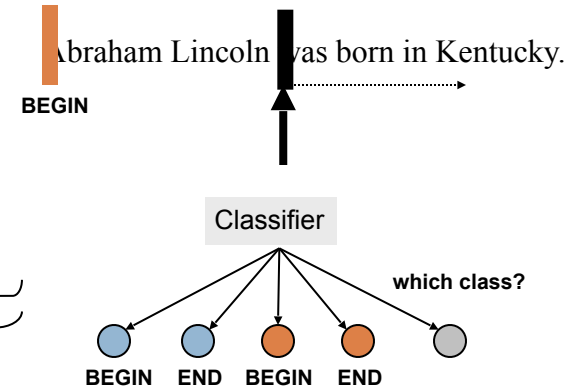


## Sliding Window

Abraham Lincoln was born in Kentucky.

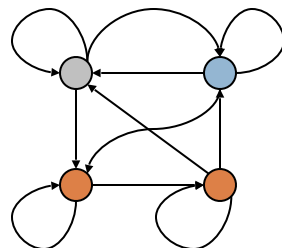


## Boundary Models



## Finite State Machines

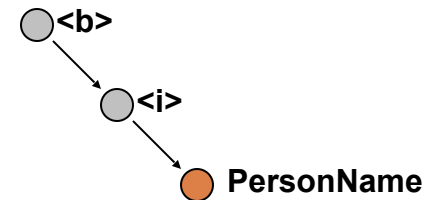
Abraham Lincoln was born in Kentucky.



## Wrapper Induction

**<b><i>Abraham Lincoln</i></b>** was born in Kentucky.

Learn and apply pattern for a website



Any of these models can be used to capture words, formatting or both.



# Information Extraction by Sliding Window

**E.g.  
Looking for  
seminar  
location**

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

CMU UseNet Seminar Announcement

# Information Extraction by Sliding Window

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# Information Extraction by Sliding Window

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# Information Extraction by Sliding Window

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# Information Extraction by Sliding Window

**E.g.  
Looking for  
seminar  
location**

GRAND CHALLENGES FOR MACHINE LEARNING

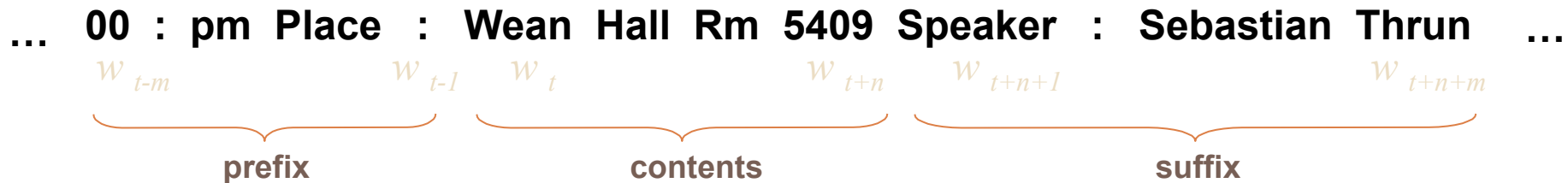
Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

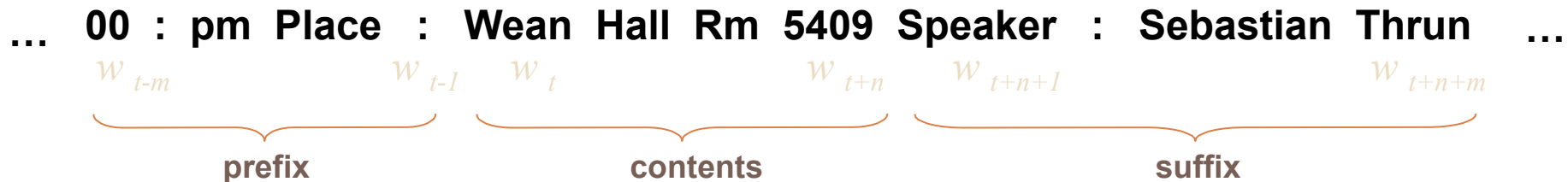
CMU UseNet Seminar Announcement

# Information Extraction by Sliding Window



- Standard supervised learning setting
  - Positive instances?
  - Negative instances?

# Information Extraction by Sliding Window



- Standard supervised learning setting
  - Positive instances: Windows with real label
  - Negative instances: All other windows
  - Features based on candidate, prefix and suffix

# IE by Boundary Detection

**E.g.  
Looking for  
seminar  
location**

GRAND CHALLENGES FOR MACHINE LEARNING



Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**CMU UseNet Seminar Announcement**



# IE by Boundary Detection

GRAND CHALLENGES FOR MACHINE LEARNING



Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# IE by Boundary Detection

GRAND CHALLENGES FOR MACHINE LEARNING

▲  
Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# IE by Boundary Detection

**E.g.  
Looking for  
seminar  
location**

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

CMU UseNet Seminar Announcement

# IE by Boundary Detection

GRAND CHALLENGES FOR MACHINE LEARNING

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University

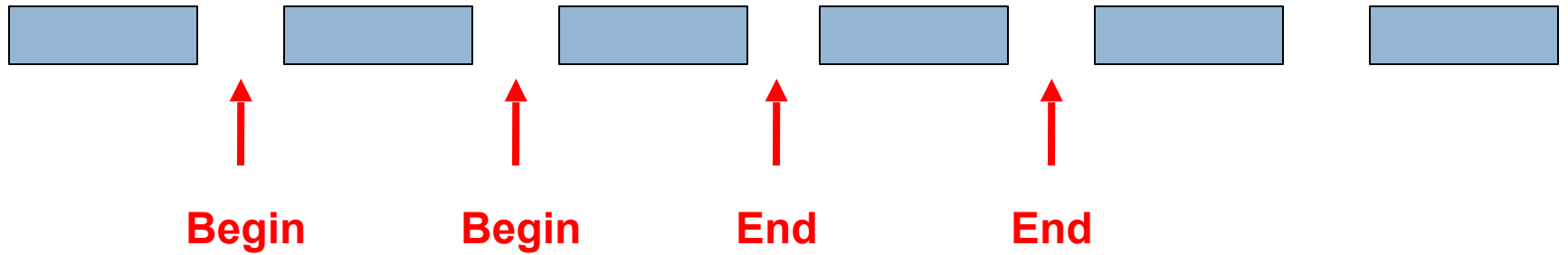
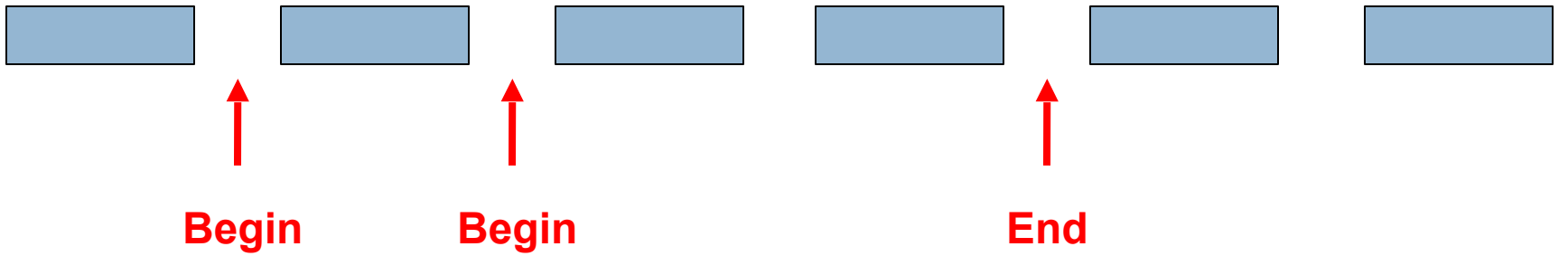
3:30 pm  
7500 Wean Hall

Machine learning has evolved from obscurity in the 1970s into a vibrant and popular discipline in artificial intelligence during the 1980s and 1990s. As a result of its success and growth, machine learning is evolving into a collection of related disciplines: inductive concept acquisition, analytic learning in problem solving (e.g. analogy, explanation-based learning), learning theory (e.g. PAC learning), genetic algorithms, connectionist learning, hybrid systems, and so on.

**E.g.  
Looking for  
seminar  
location**

CMU UseNet Seminar Announcement

# Some concerns



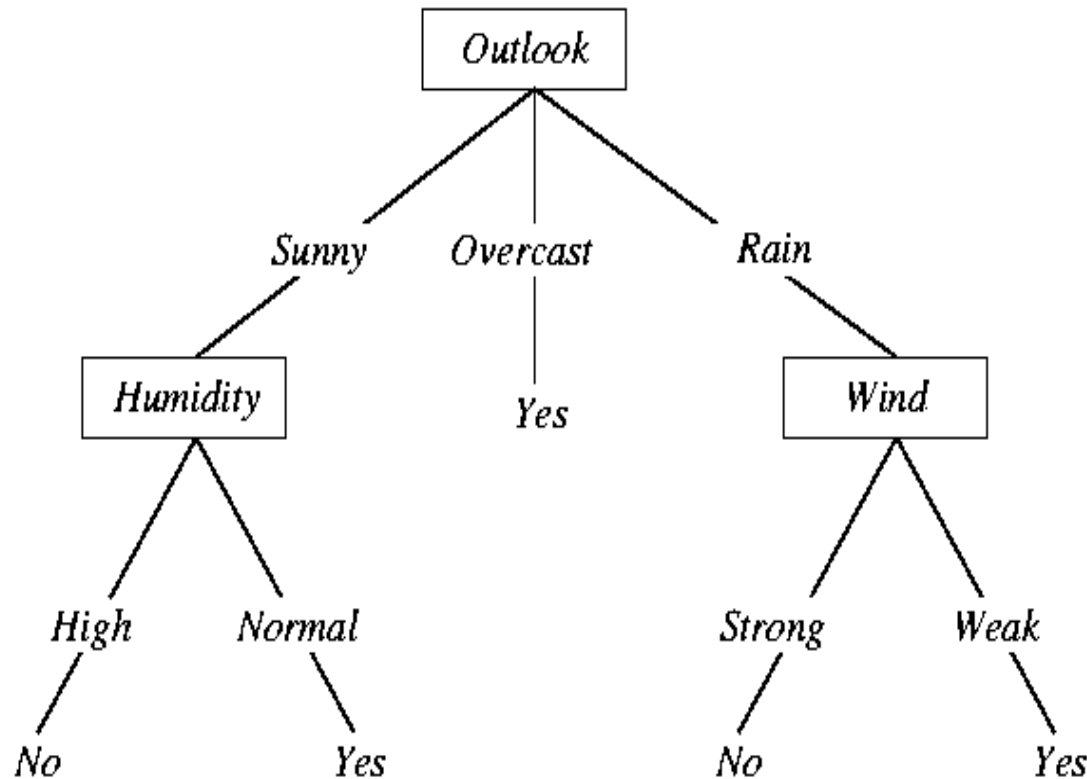
# Problems with Sliding Windows and Boundary Finders

- Decisions in neighboring parts of the input are made independently from each other.
  - ⊠ Sliding Window may predict a “seminar end time” before the “seminar start time”.
  - ⊠ It is possible for two overlapping windows to both be above threshold.
  - ⊠ In a Boundary-Finding system, left boundaries are laid down independently from right boundaries

# Where we are going

- What we just completed: introduction to NER using classifiers
  - Features, sliding windows vs. boundaries
- Rest of today: look at decision trees as part of a general introduction to machine learning
  - I will present a different perspective from the Statistical Methods course
  - Necessary background to linear models, which I will present next week
    - Next week, I will connect linear models with NER
    - The rest of this lecture is a short break from NER

# Decision Tree Representation for 'Play Tennis?'



- Internal node  
~ test an attribute
- Branch  
~ attribute value
- Leaf  
~ classification result



# When is it useful?

- ❑ Medical diagnosis
- ❑ Equipment diagnosis
- ❑ Credit risk analysis
- ❑ etc

# Outline

- Contingency tables
  - Census data set
- Information gain
  - Beach data set
- Learning an unpruned decision tree recursively
  - Gasoline usage data set
- Training error
- Test error
- Overfitting
- Avoiding overfitting

# Here is a dataset

| age | employe   | education | edun | marital    | ... | job         | relation   | race       | gender | hour | country     | wealth |
|-----|-----------|-----------|------|------------|-----|-------------|------------|------------|--------|------|-------------|--------|
|     |           |           |      |            | ... |             |            |            |        |      |             |        |
| 39  | State_gov | Bachelors | 13   | Never_mar  | ... | Adm_cleric  | Not_in_fan | White      | Male   | 40   | United_Stat | poor   |
| 51  | Self_emp_ | Bachelors | 13   | Married    | ... | Exec_man    | Husband    | White      | Male   | 13   | United_Stat | poor   |
| 39  | Private   | HS_grad   | 9    | Divorced   | ... | Handlers_(  | Not_in_fan | White      | Male   | 40   | United_Stat | poor   |
| 54  | Private   | 11th      | 7    | Married    | ... | Handlers_(  | Husband    | Black      | Male   | 40   | United_Stat | poor   |
| 28  | Private   | Bachelors | 13   | Married    | ... | Prof_speci  | Wife       | Black      | Female | 40   | Cuba        | poor   |
| 38  | Private   | Masters   | 14   | Married    | ... | Exec_man    | Wife       | White      | Female | 40   | United_Stat | poor   |
| 50  | Private   | 9th       | 5    | Married_sp | ... | Other_ser   | Not_in_fan | Black      | Female | 16   | Jamaica     | poor   |
| 52  | Self_emp_ | HS_grad   | 9    | Married    | ... | Exec_man    | Husband    | White      | Male   | 45   | United_Stat | rich   |
| 31  | Private   | Masters   | 14   | Never_mar  | ... | Prof_speci  | Not_in_fan | White      | Female | 50   | United_Stat | rich   |
| 42  | Private   | Bachelors | 13   | Married    | ... | Exec_man    | Husband    | White      | Male   | 40   | United_Stat | rich   |
| 37  | Private   | Some_coll | 10   | Married    | ... | Exec_man    | Husband    | Black      | Male   | 80   | United_Stat | rich   |
| 30  | State_gov | Bachelors | 13   | Married    | ... | Prof_speci  | Husband    | Asian      | Male   | 40   | India       | rich   |
| 24  | Private   | Bachelors | 13   | Never_mar  | ... | Adm_cleric  | Own_child  | White      | Female | 30   | United_Stat | poor   |
| 33  | Private   | Assoc_acc | 12   | Never_mar  | ... | Sales       | Not_in_fan | Black      | Male   | 50   | United_Stat | poor   |
| 41  | Private   | Assoc_voc | 11   | Married    | ... | Craft_repai | Husband    | Asian      | Male   | 40   | *MissingV   | rich   |
| 34  | Private   | 7th_8th   | 4    | Married    | ... | Transport_  | Husband    | Amer_India | Male   | 45   | Mexico      | poor   |
| 26  | Self_emp_ | HS_grad   | 9    | Never_mar  | ... | Farming_fi  | Own_child  | White      | Male   | 35   | United_Stat | poor   |
| 33  | Private   | HS_grad   | 9    | Never_mar  | ... | Machine_c   | Unmarried  | White      | Male   | 40   | United_Stat | poor   |
| 38  | Private   | 11th      | 7    | Married    | ... | Sales       | Husband    | White      | Male   | 50   | United_Stat | poor   |
| 44  | Self_emp_ | Masters   | 14   | Divorced   | ... | Exec_man    | Unmarried  | White      | Female | 45   | United_Stat | rich   |
| 41  | Private   | Doctorate | 16   | Married    | ... | Prof_speci  | Husband    | White      | Male   | 60   | United_Stat | rich   |
| :   | :         | :         | :    | :          | :   | :           | :          | :          | :      | :    | :           | :      |

48,000 records, 16 attributes [Kohavi 1995]

# About this dataset

- It is a tiny subset of the 1990 US Census.
- It is publicly available online from the UCI Machine Learning Datasets repository

## Used Attributes

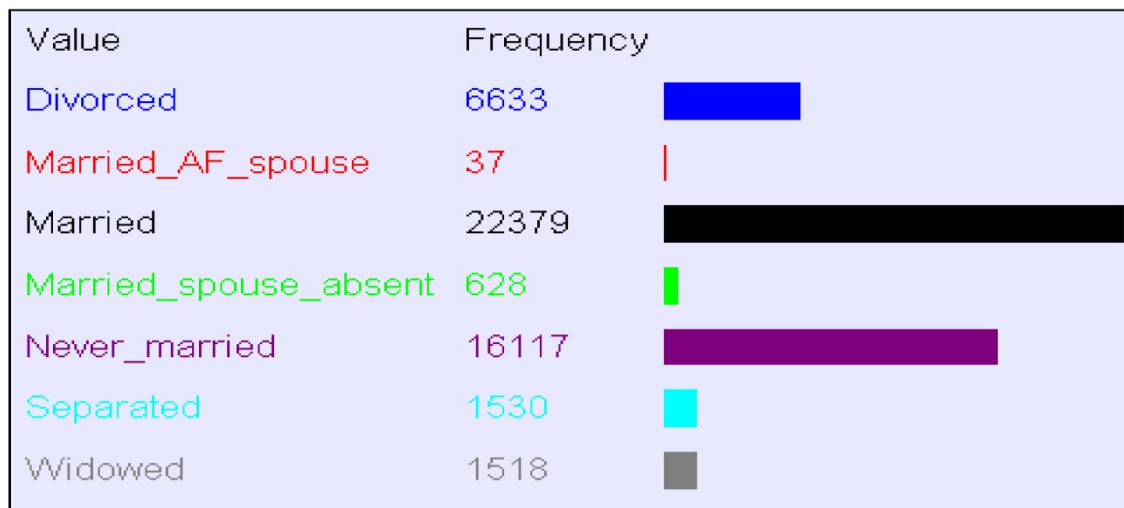
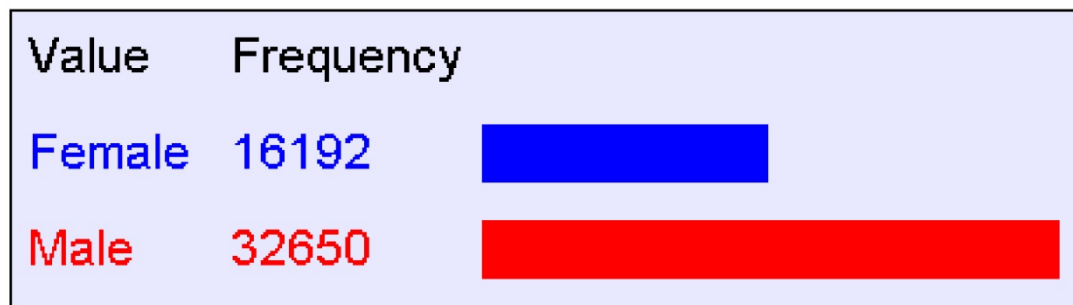
|              |          |             |              |
|--------------|----------|-------------|--------------|
| age          | edunum   | race        | hours_worked |
| employment   | marital  | gender      | country      |
| taxweighting | job      | capitalgain | wealth       |
| education    | relation | capitalloss | agegroup     |

This color = Real-valued    This color = Symbol-valued

Successfully loaded a new dataset from the file \tadult.fds. It has 16 attributes and 48842 records.

# What can you do with a dataset?

- Well, you can look at histograms...







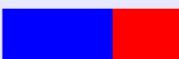




# A 2-d Contingency Table

| wealth values: |     | poor  | rich |
|----------------|-----|-------|------|
| agegroup       | 10s | 2507  | 3    |
|                | 20s | 11262 | 743  |
|                | 30s | 9468  | 3461 |
|                | 40s | 6738  | 3986 |
|                | 50s | 4110  | 2509 |
|                | 60s | 2245  | 809  |
|                | 70s | 668   | 147  |
|                | 80s | 115   | 16   |
|                | 90s | 42    | 13   |

- For each pair of values for attributes (agegroup, wealth) we can see how many records match.










# A 2-d Contingency Table

wealth values: poor rich

| agegroup | 10s   | 2507 | 3   |  |
|----------|-------|------|---|---|
| 20s      | 11262 | 743  |   |   |
| 30s      | 9468  | 3461 |   |   |
| 40s      | 6738  | 3986 |   |   |
| 50s      | 4110  | 2509 |    |   |
| 60s      | 2245  | 809  |    |   |
| 70s      | 668   | 147  |   |   |
| 80s      | 115   | 16   |  |   |
| 90s      | 42    | 13   |  |   |

- Easier to appreciate graphically

# A 2-d Contingency Table

| wealth values: |     | poor  | rich |  |
|----------------|-----|-------|------|--|
| agegroup       | 10s | 2507  | 3    |    |
|                | 20s | 11262 | 743  |    |
|                | 30s | 9468  | 3461 |    |
|                | 40s | 6738  | 3986 |    |
|                | 50s | 4110  | 2509 |    |
|                | 60s | 2245  | 809  |    |
|                | 70s | 668   | 147  |   |
|                | 80s | 115   | 16   |  |
|                | 90s | 42    | 13   |  |

- Easier to see “interesting” things if we stretch out the histogram bars



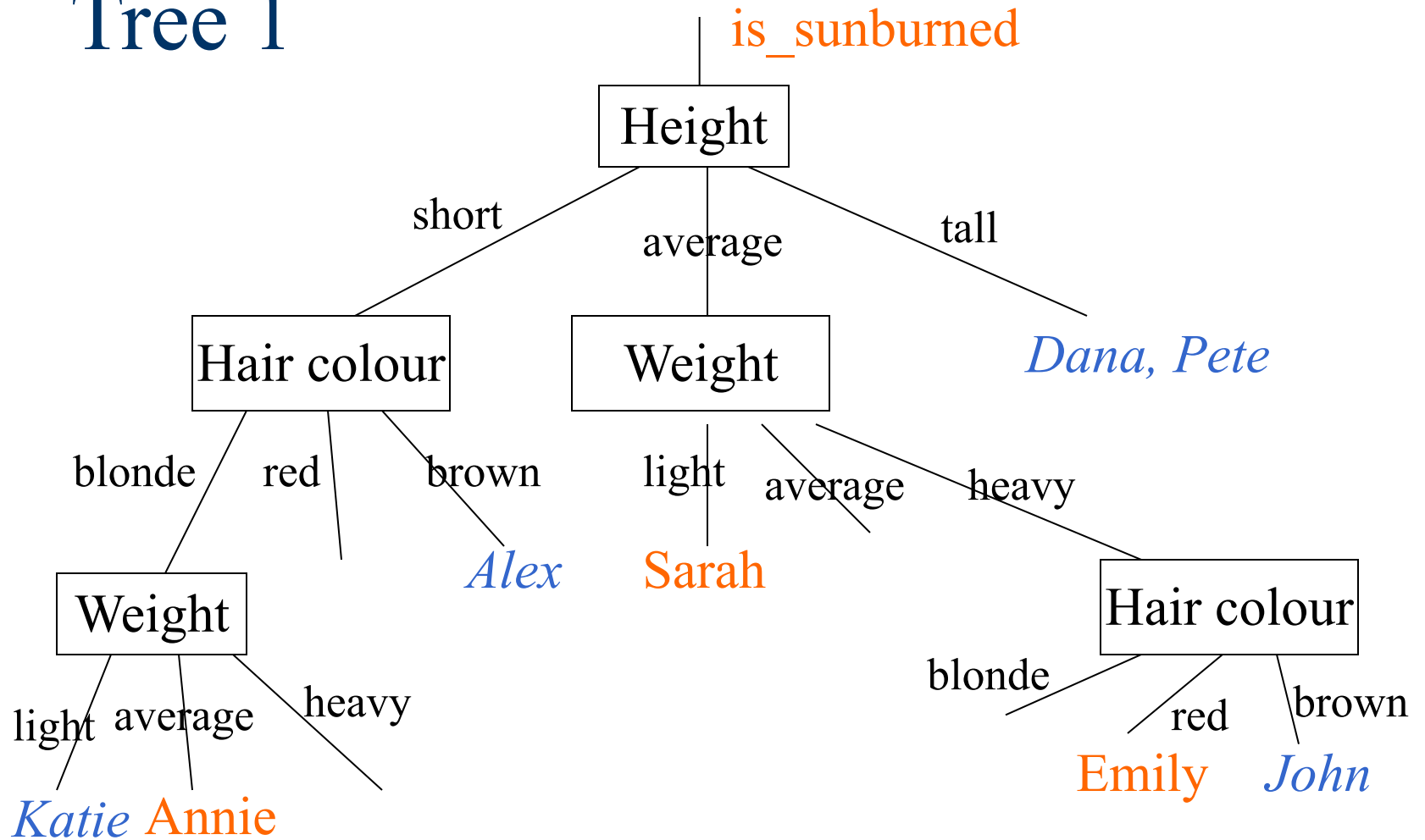
# Using this idea for classification

- We will now look at a (toy) dataset presented in Winston's Artificial Intelligence textbook
- It is often used when explaining decision trees
- The variable we are trying to pick is "got\_a\_sunburn" (or "is\_sunburned" if you like)
- We will look at different decision trees that can be used to correctly classify this data

# Sunburn Data Collected

| Name         | Hair   | Height  | Weight  | Lotion | Result           |
|--------------|--------|---------|---------|--------|------------------|
| <b>Sarah</b> | Blonde | Average | Light   | No     | <b>Sunburned</b> |
| <i>Dana</i>  | Blonde | Tall    | Average | Yes    | <i>None</i>      |
| <i>Alex</i>  | Brown  | Short   | Average | Yes    | <i>None</i>      |
| <b>Annie</b> | Blonde | Short   | Average | No     | <b>Sunburned</b> |
| <b>Emily</b> | Red    | Average | Heavy   | No     | <b>Sunburned</b> |
| <i>Pete</i>  | Brown  | Tall    | Heavy   | No     | <i>None</i>      |
| <i>John</i>  | Brown  | Average | Heavy   | No     | <i>None</i>      |
| <i>Kate</i>  | Blonde | Short   | Light   | Yes    | <i>None</i>      |

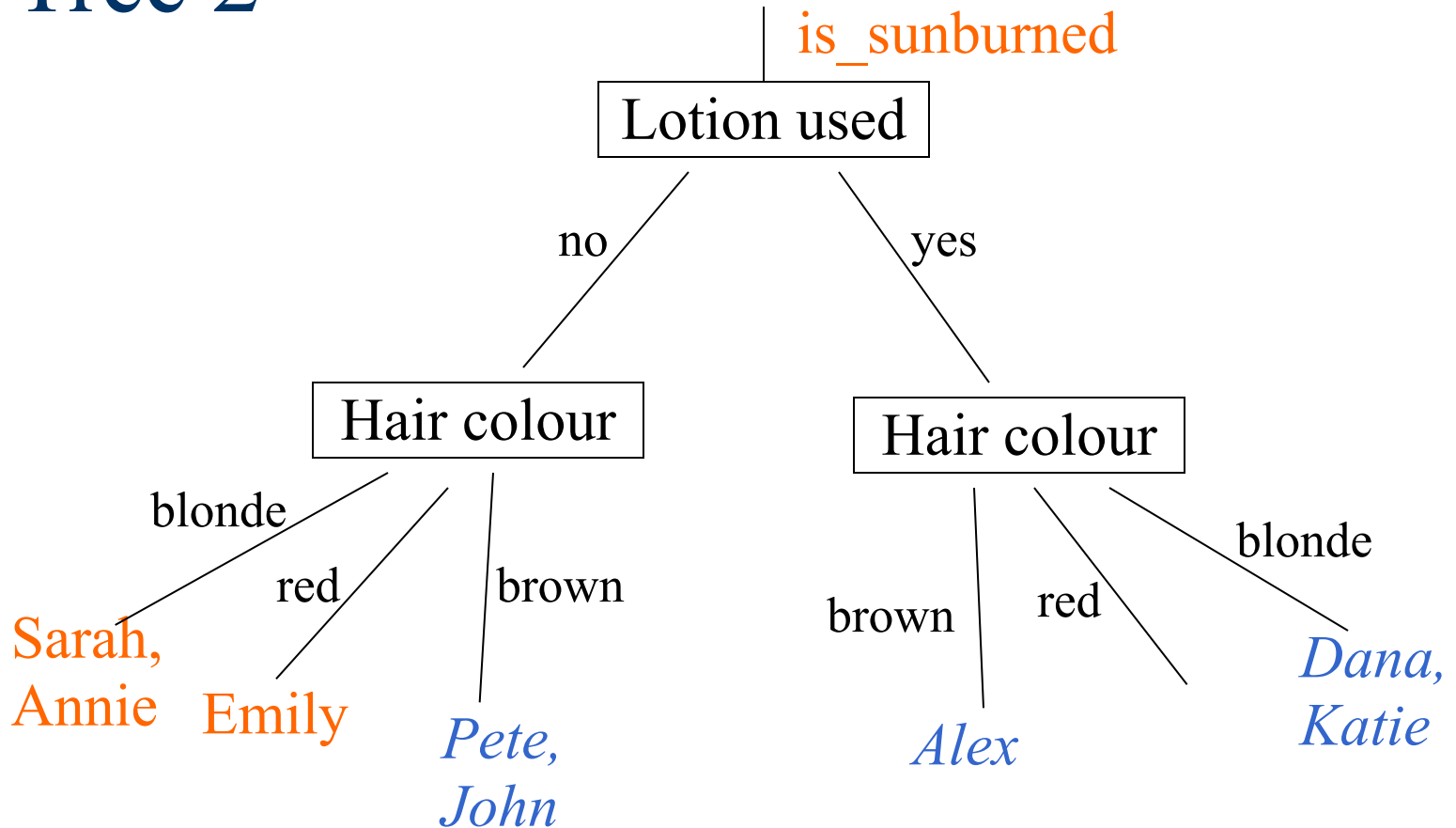
# Decision Tree 1



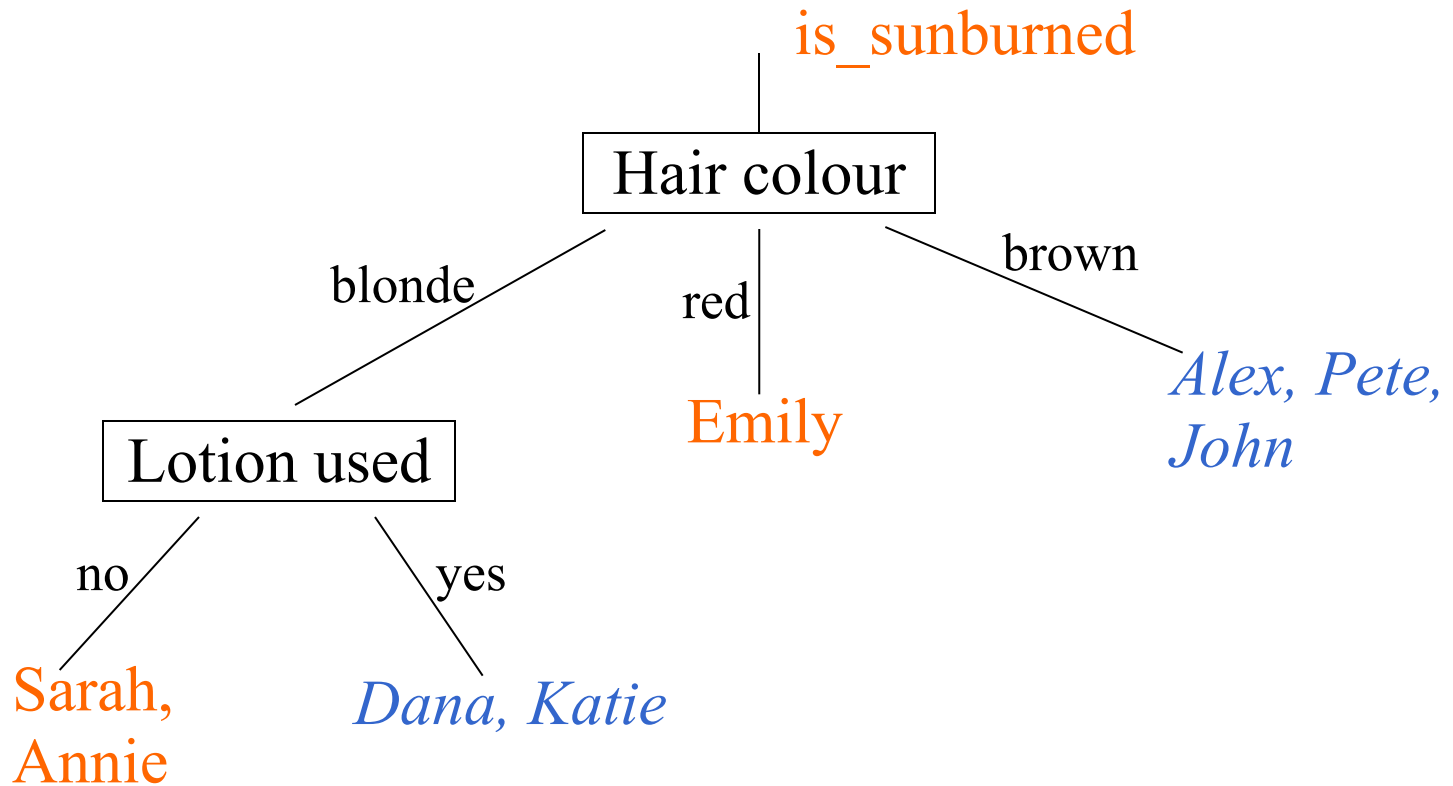
# Sunburn sufferers are ...

- n If height="average" then
  - if weight="light" then
    - `return(true) ;;; Sarah`
  - elseif weight="heavy" then
    - if hair\_colour="red" then
      - `return(true) ;;; Emily`
- n elseif height="short" then
  - if hair\_colour="blonde" then
    - if weight="average" then
      - `return(true) ;;; Annie`
- n else *`return(false) ;;;everyone else`*

# Decision Tree 2



# Decision Tree 3



# Summing up

- n Irrelevant attributes do not classify the data well
- n Using irrelevant attributes thus causes larger decision trees
- n a computer could look for simpler decision trees
- n Q: How?

# A: How WE did it?

- n Q: Which is the best attribute for splitting up the data?
- n A: The one which is *most informative* for the classification we want to get.
- n Q: What does it mean 'more informative'?
- n A: The attribute which best *reduces the uncertainty* or the disorder



- n We need a quantity to measure the *disorder* in a set of examples

$$S = \{s_1, s_2, s_3, \dots, s_n\}$$

where  $s_1 = \text{"Sarah"}$ ,  $s_2 = \text{"Dana"}$ , ...

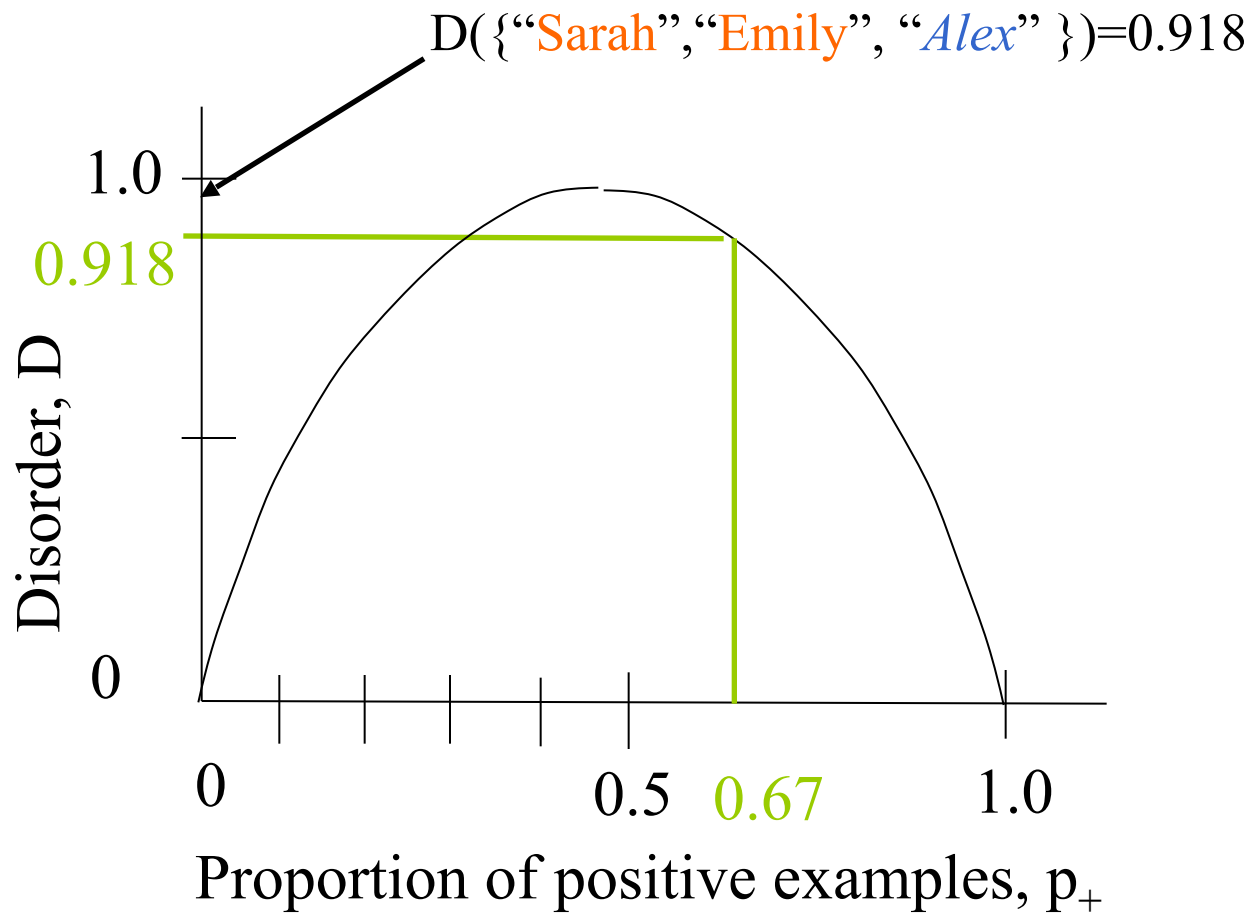
- n Then we need a quantity to measure the amount of *reduction of the disorder* level in the instance of knowing the value of a particular attribute

# What properties should the *Disorder* ( $D$ ) have?

- n Suppose that  $D(S)=0$  means that all the examples in  $S$  have the same class
- n Suppose that  $D(S)=1$  means that half the examples in  $S$  are of one class and half are the opposite class

# Examples

- n  $D(\{\textit{Dana}, \textit{Pete}\}) = 0$
- n  $D(\{\textit{Sarah}, \textit{Annie}, \textit{Emily}\}) = 0$
- n  $D(\{\textit{Sarah}, \textit{Emily}, \textit{Alex}, \textit{John}\}) = 1$
- n  $D(\{\textit{Sarah}, \textit{Emily}, \textit{Alex}\}) = ?$



# Definition of Disorder

The **Entropy** measures the disorder of a set  $S$  containing a total of  $n$  examples of which  $n_+$  are positive and  $n_-$  are negative and it is given by

$$D(n_+, n_-) = -\frac{n_+}{n} \log_2 \frac{n_+}{n} - \frac{n_-}{n} \log_2 \frac{n_-}{n} = \text{Entropy}(S)$$

where

$$\log_2 x \text{ means } 2^? = x$$

Check it!

$$D(0,1) = ? \quad D(1,0) = ? \quad D(0.5,0.5) = ?$$

# Back to the beach (or the disorder of sunbathers)!

$D(\{ \text{“Sarah”}, \text{“Dana”}, \text{“Alex”}, \text{“Annie”}, \text{“Emily”}, \text{“Pete”}, \text{“John”}, \text{“Katie”} \})$

$$= D(3,5) = -\frac{3}{8} \log_2 \frac{3}{8} - \frac{5}{8} \log_2 \frac{5}{8}$$

$$= 0.954$$

# Some more useful properties of the Entropy

$$D(n, m) = D(m, n)$$

$$D(0, m) = 0$$

$$D(m, m) = 1$$

- n So: We can measure the disorder 😊
- n What's left:
  - We want to measure how much by knowing the value of a particular attribute the disorder of a set would reduce.



- The **Information Gain** measures the expected reduction in entropy due to splitting on an attribute A

$$Gain(S, A) = Entropy(S) - \underbrace{\sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)}_{\text{average disorder}}$$

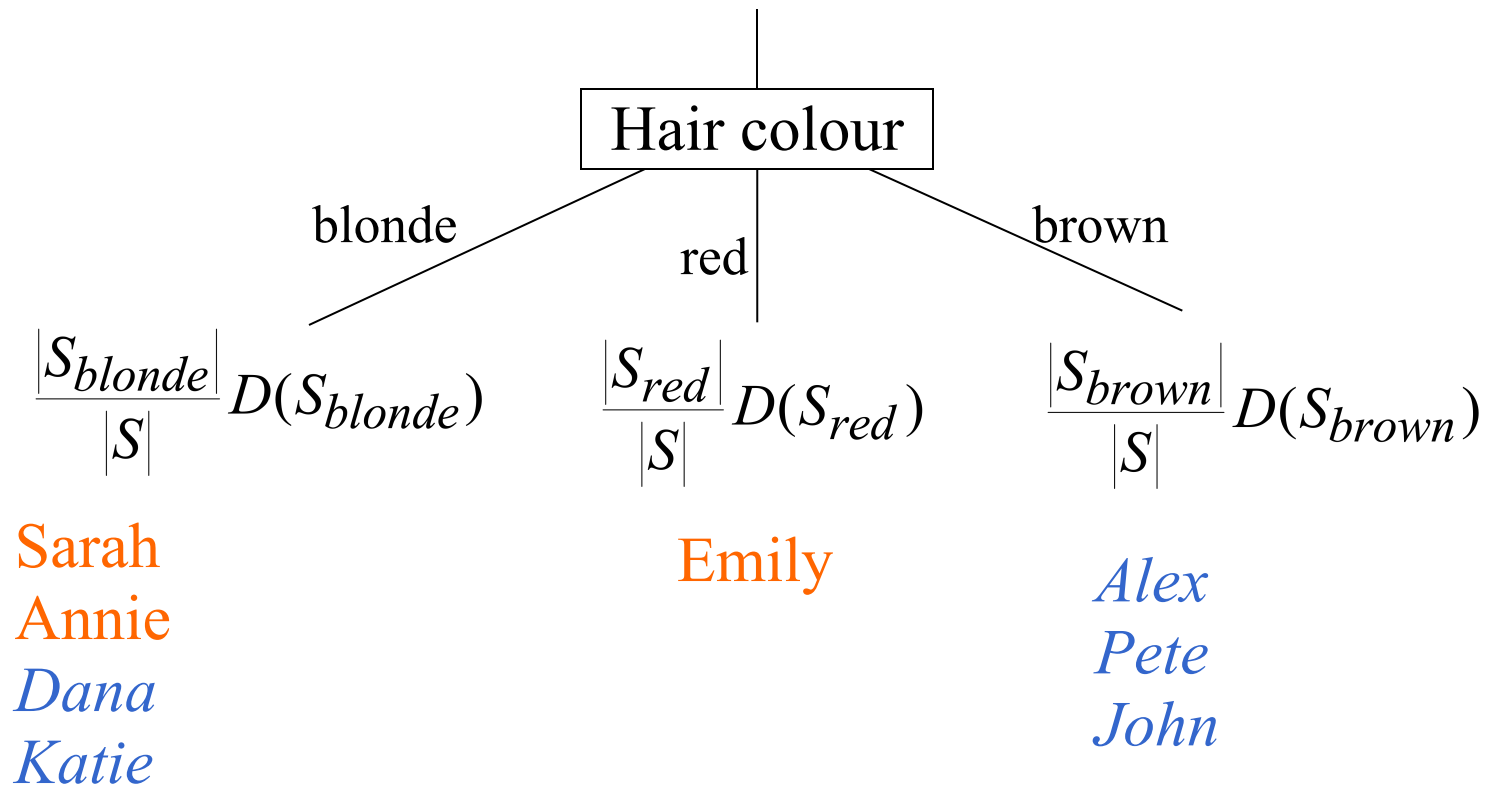
We want:

-large Gain

-same as: small avg  
disorder created

the **average disorder** is just the weighted sum of the disorders in the branches (subsets) created by the values of A.

# Back to the beach: calculate the Average Disorder associated with Hair Colour



## ...Calculating the Disorder of the “blondes”

The first term of the sum:

$$n \quad D(S_{blonde}) =$$

$$D(\{ \text{“Sarah”, “Annie”, “Dana”, “Katie”} \}) = D(2,2) \\ = 1$$

$$\frac{|S_{blonde}|}{|S|} D(S_{blonde}) = \frac{|S_{blonde}|}{|S|} = \frac{4}{8} = 0.5$$

## ...Calculating the disorder of the others

The second and third terms of the sum:

$$n S_{\text{red}} = \{ \text{“Emily”} \}$$

$$n S_{\text{brown}} = \{ \text{“Alex”}, \text{“Pete”}, \text{“John”} \}.$$

These are both 0 because within each set  
all the examples have the same class

So the avg disorder created when splitting  
on ‘hair colour’ is  $0.5+0+0=0.5$

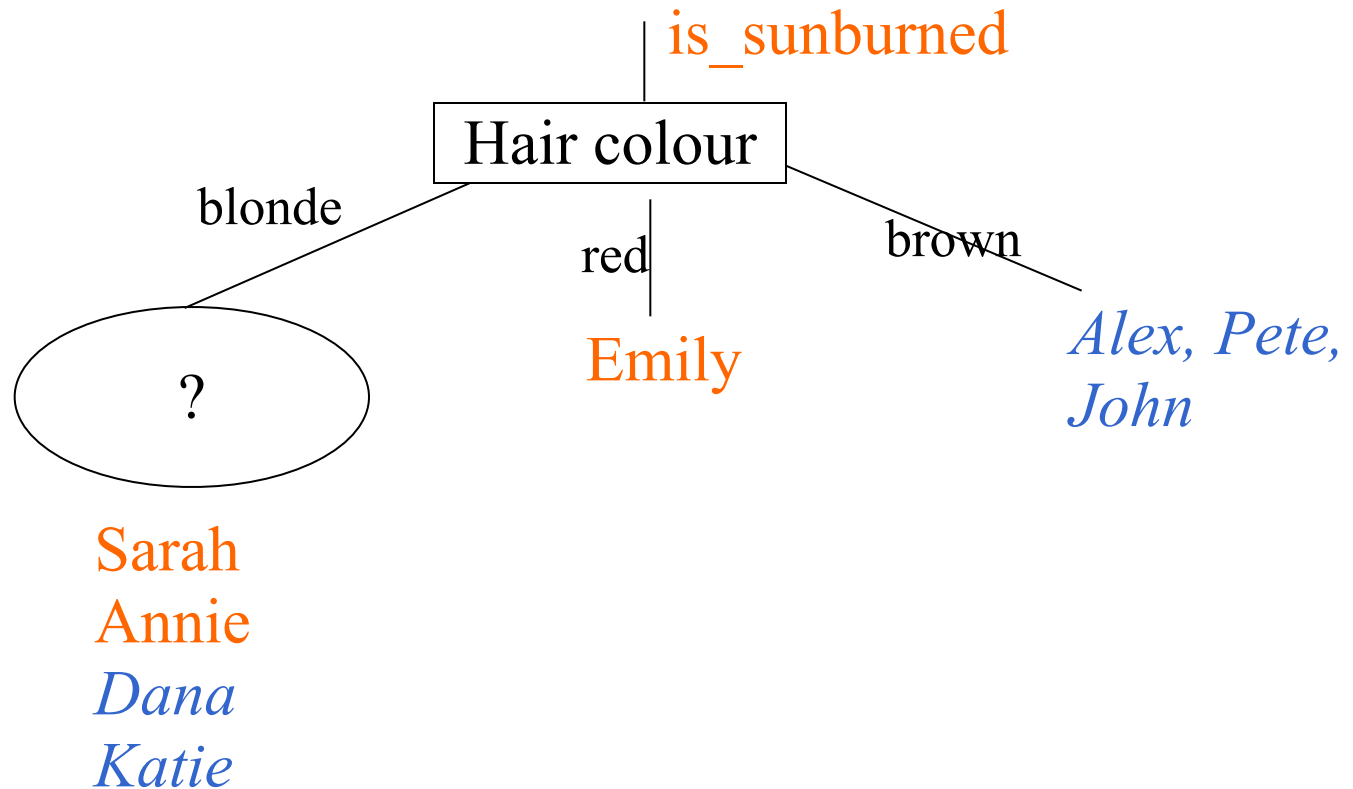
# Which decision variable minimises the disorder?

| Test   | Disorder |   |
|--------|----------|---|
| Hair   | 0.5      | – this what we just computed  |
| height | 0.69     | } these are the avg disorders of the other attributes, computed in the same way |
| weight | 0.94     |   |
| lotion | 0.61     |   |

Which decision variable maximises the Info Gain then?

Remember it's the one which minimises the avg disorder (see slide 46 for memory refreshing).

# So what is the best decision tree?



# Outline

- Contingency tables
  - Census data set
- Information gain
  - Beach data set
- Learning an unpruned decision tree recursively
  - Good/bad gasoline usage = "miles per gallon" data set
- Training error
- Test error
- Overfitting
- Avoiding overfitting

# Learning Decision Trees

- A Decision Tree is a tree-structured plan of a set of attributes to test in order to predict the output.
- To decide which attribute should be tested first, simply find the one with the highest information gain.
- Then recurse...



# A small dataset: Miles Per Gallon

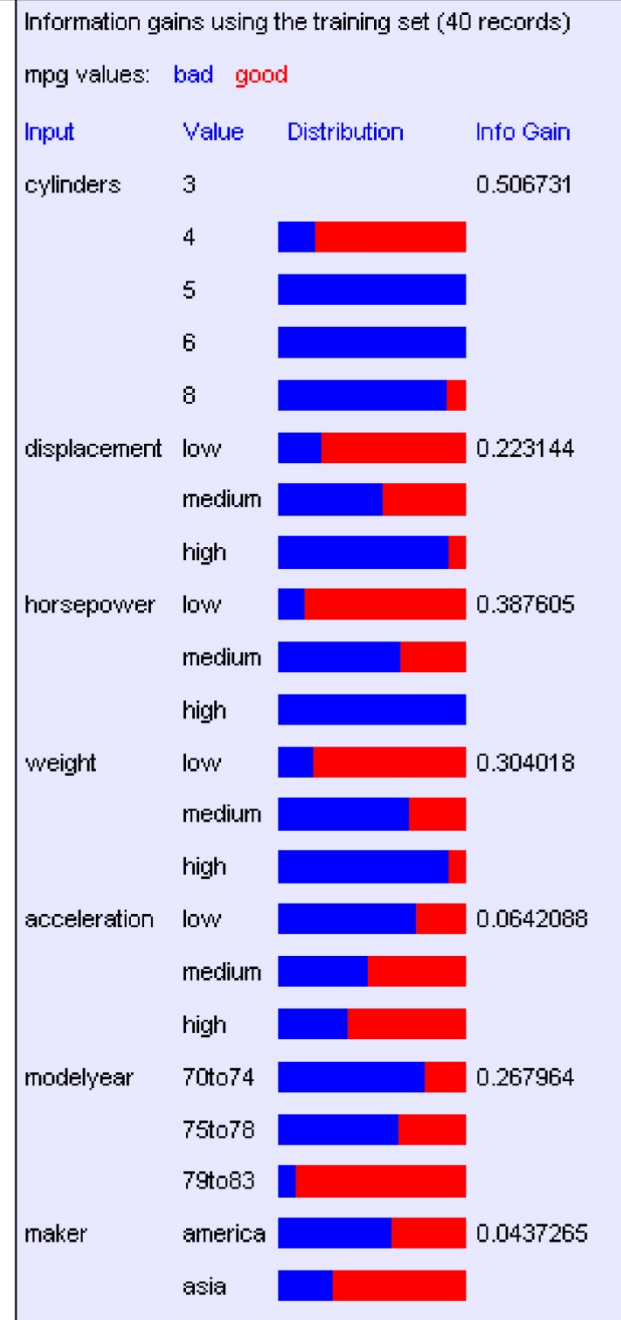
40  
Records

| 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    | europa  |
| 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    | europa  |
| bad  | 5         | medium       | medium     | medium | medium       | 75to78    | europa  |

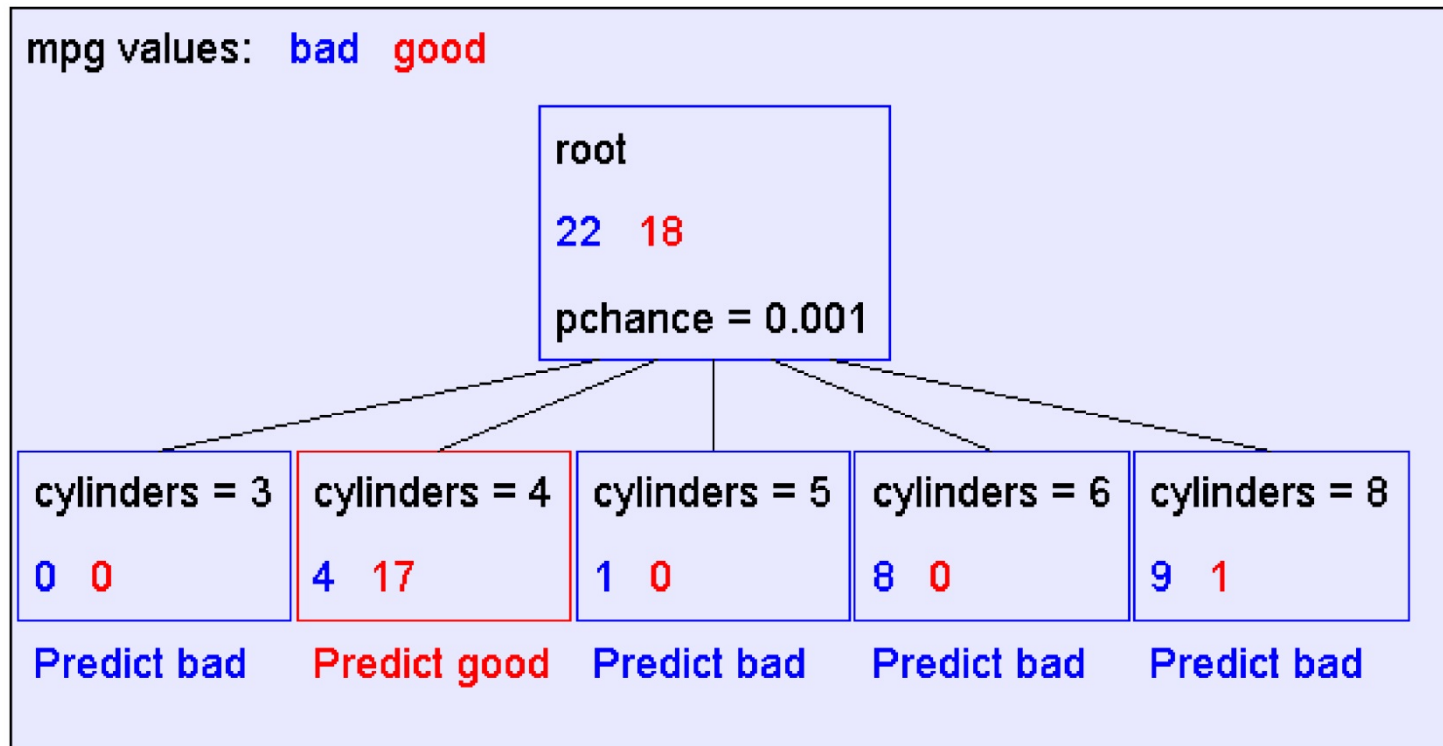
From the UCI repository (thanks to Ross Quinlan)

Suppose we want to predict MPG.

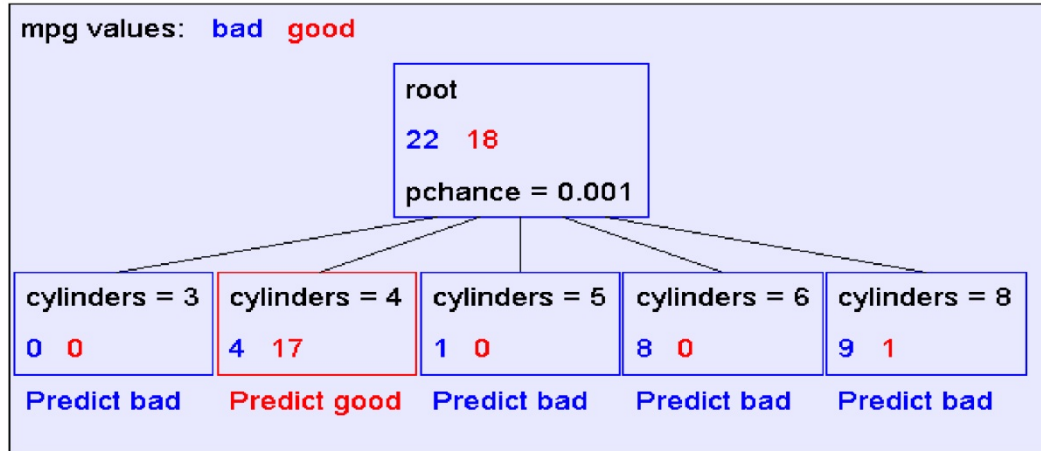
Look at all the information gains...



# A Decision Stump



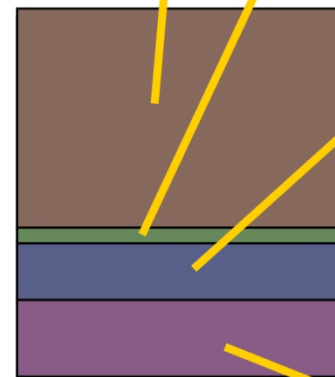
# Recursion Step



Take the Original Dataset..



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



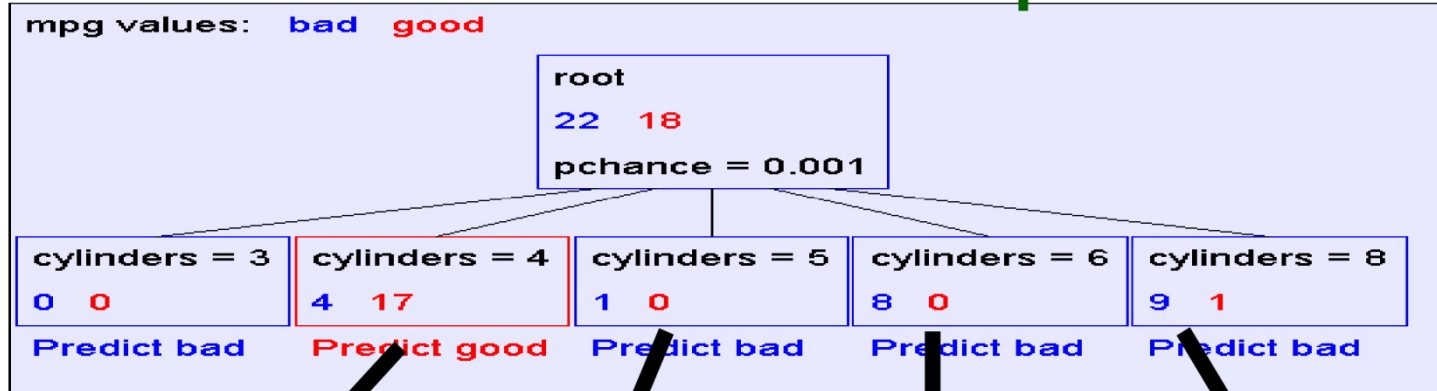
Records in which cylinders = 4

Records in which cylinders = 5

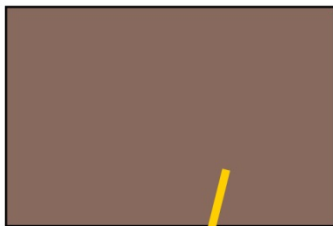
Records in which cylinders = 6

Records in which cylinders = 8

# Recursion Step



Build tree from  
These records..



Records in  
which  
cylinders = 4

Build tree from  
These records..



Records in  
which  
cylinders = 5

Build tree from  
These records..



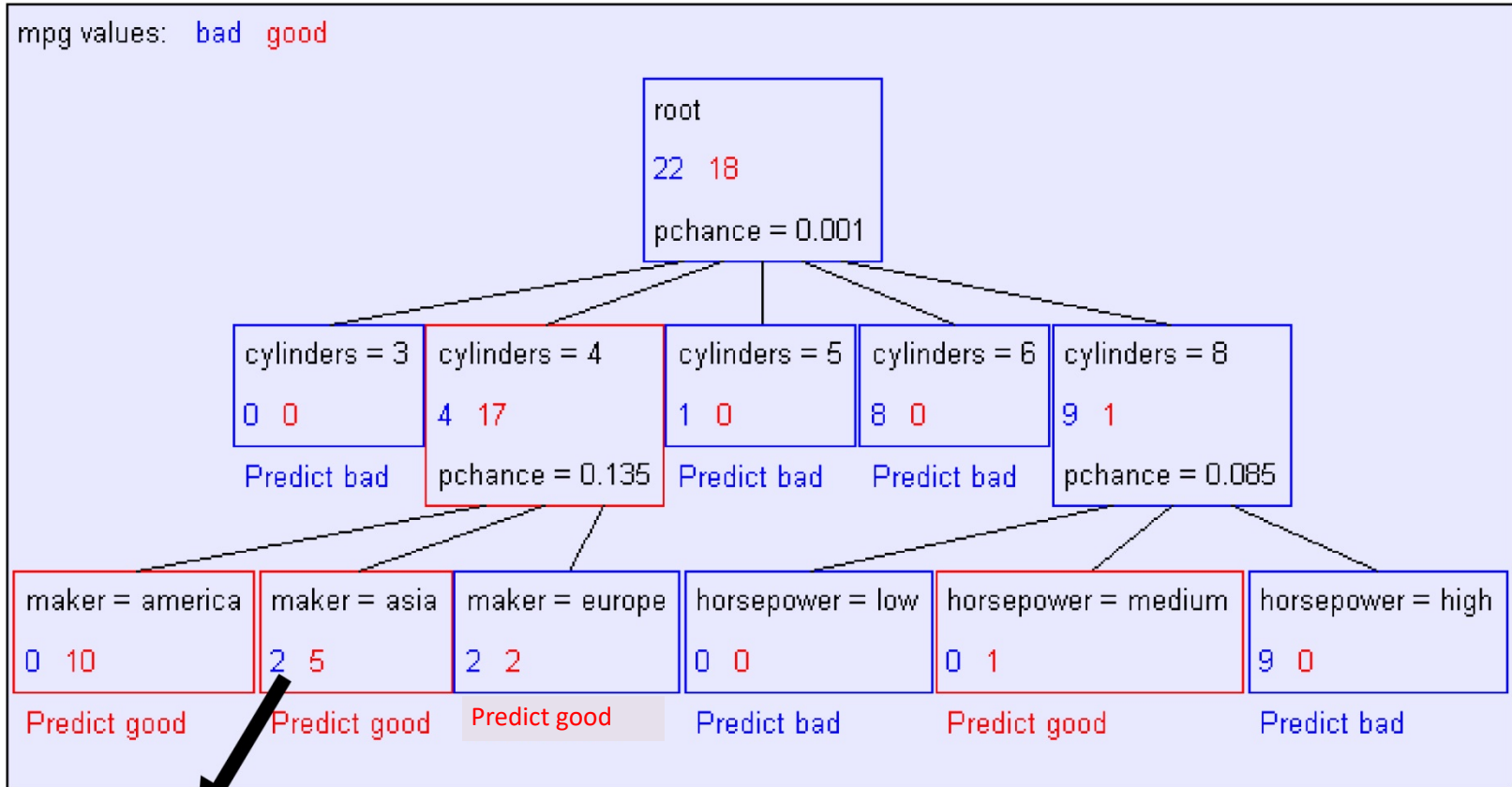
Records in  
which  
cylinders = 6

Build tree from  
These records..



Records in  
which  
cylinders = 8

# Second level of tree

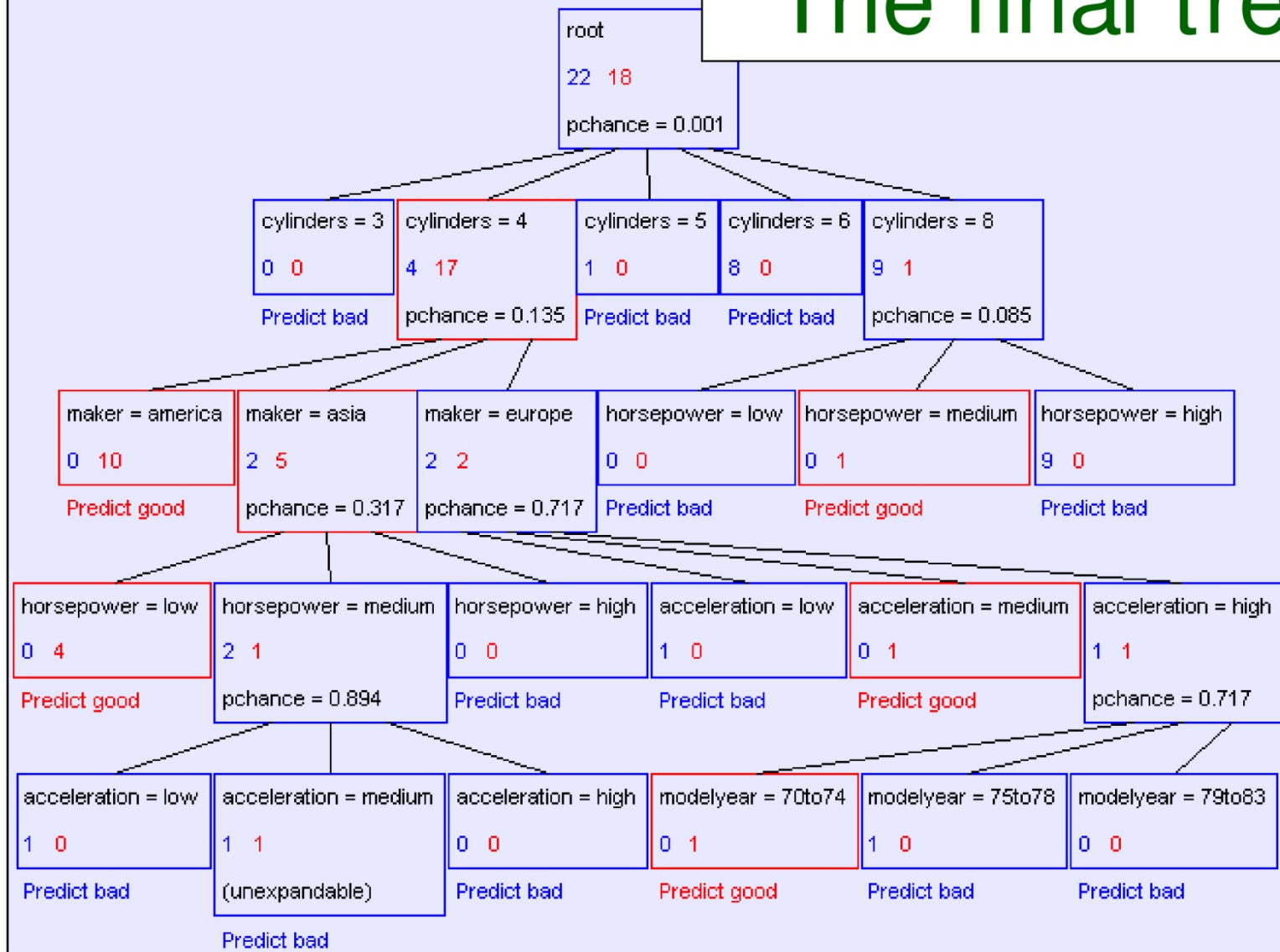


Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

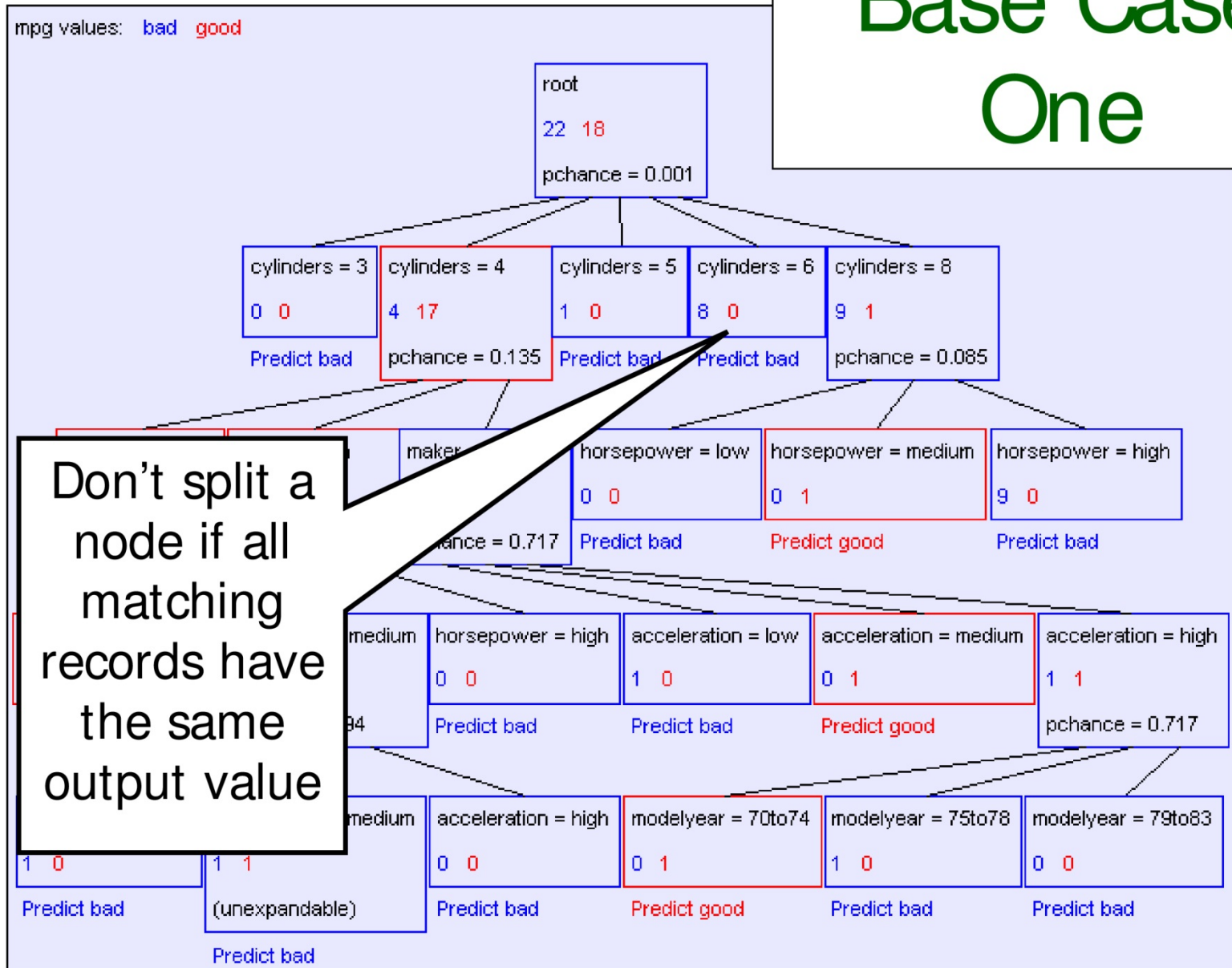
# The final tree

mpg values: bad good



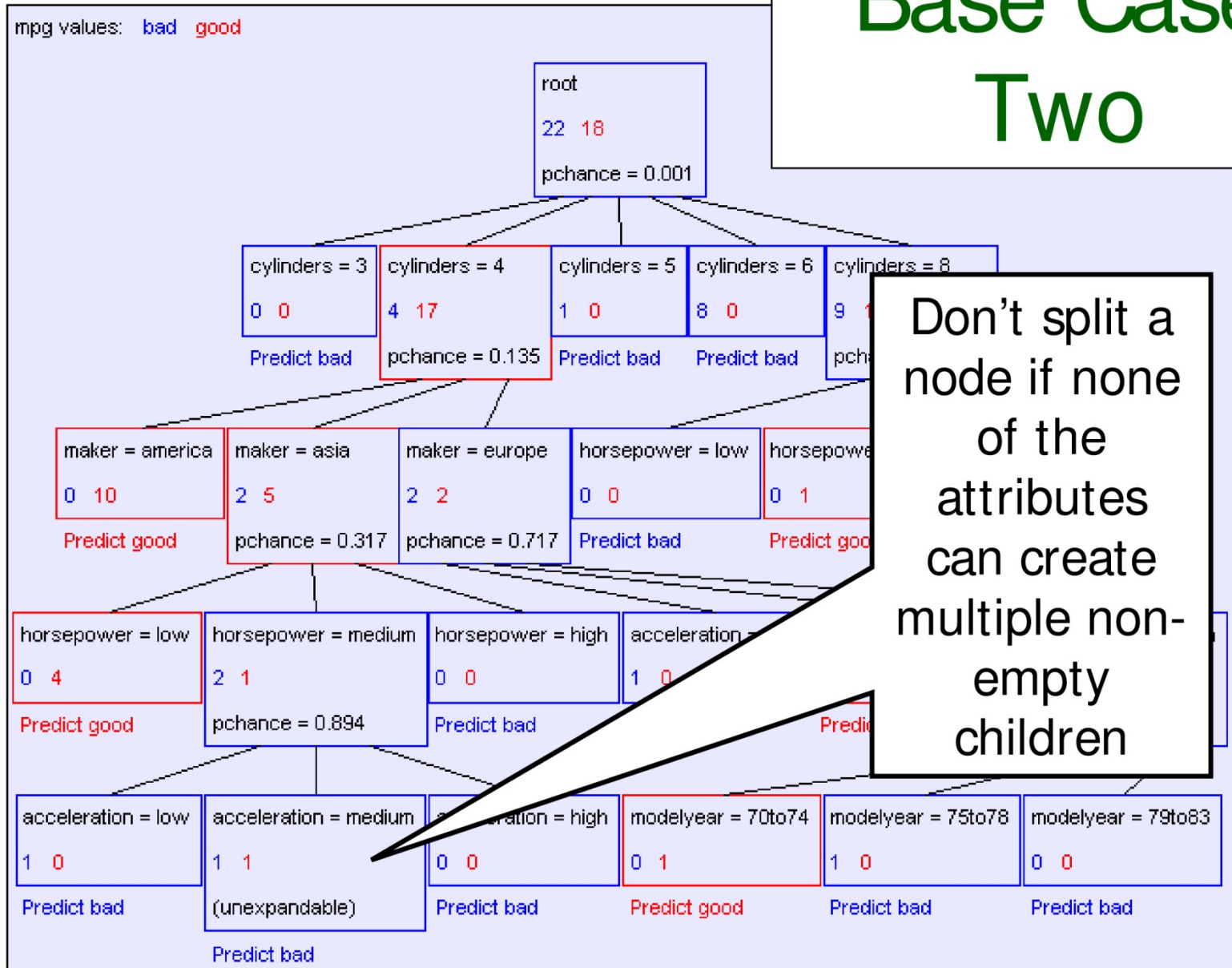


# Base Case One



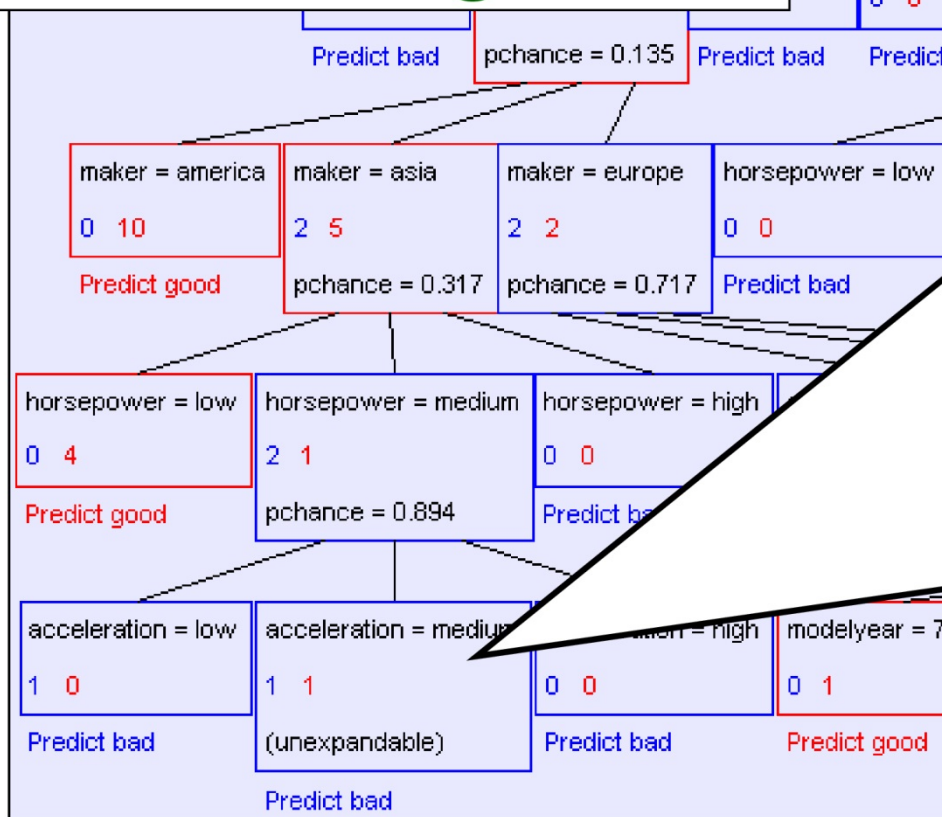


# Base Case Two



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

# Base Case Two: No attributes can distinguish



Information gains using the training set (2 records)

mpg values: bad good

| Input        | Value   | Distribution | Info Gain |
|--------------|---------|--------------|-----------|
| cylinders    | 3       |              | 0         |
|              | 4       |              |           |
|              | 5       |              |           |
|              | 6       |              |           |
| displacement | low     |              | 0         |
|              | medium  |              |           |
|              | high    |              |           |
| horsepower   | low     |              | 0         |
|              | medium  |              |           |
|              | high    |              |           |
| weight       | low     |              | 0         |
|              | medium  |              |           |
|              | high    |              |           |
| acceleration | low     |              | 0         |
|              | medium  |              |           |
|              | high    |              |           |
| modelyear    | 70to74  |              | 0         |
|              | 75to78  |              |           |
|              | 79to83  |              |           |
| maker        | america |              | 0         |
|              | asia    |              |           |
|              | europe  |              |           |

# 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**

# 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

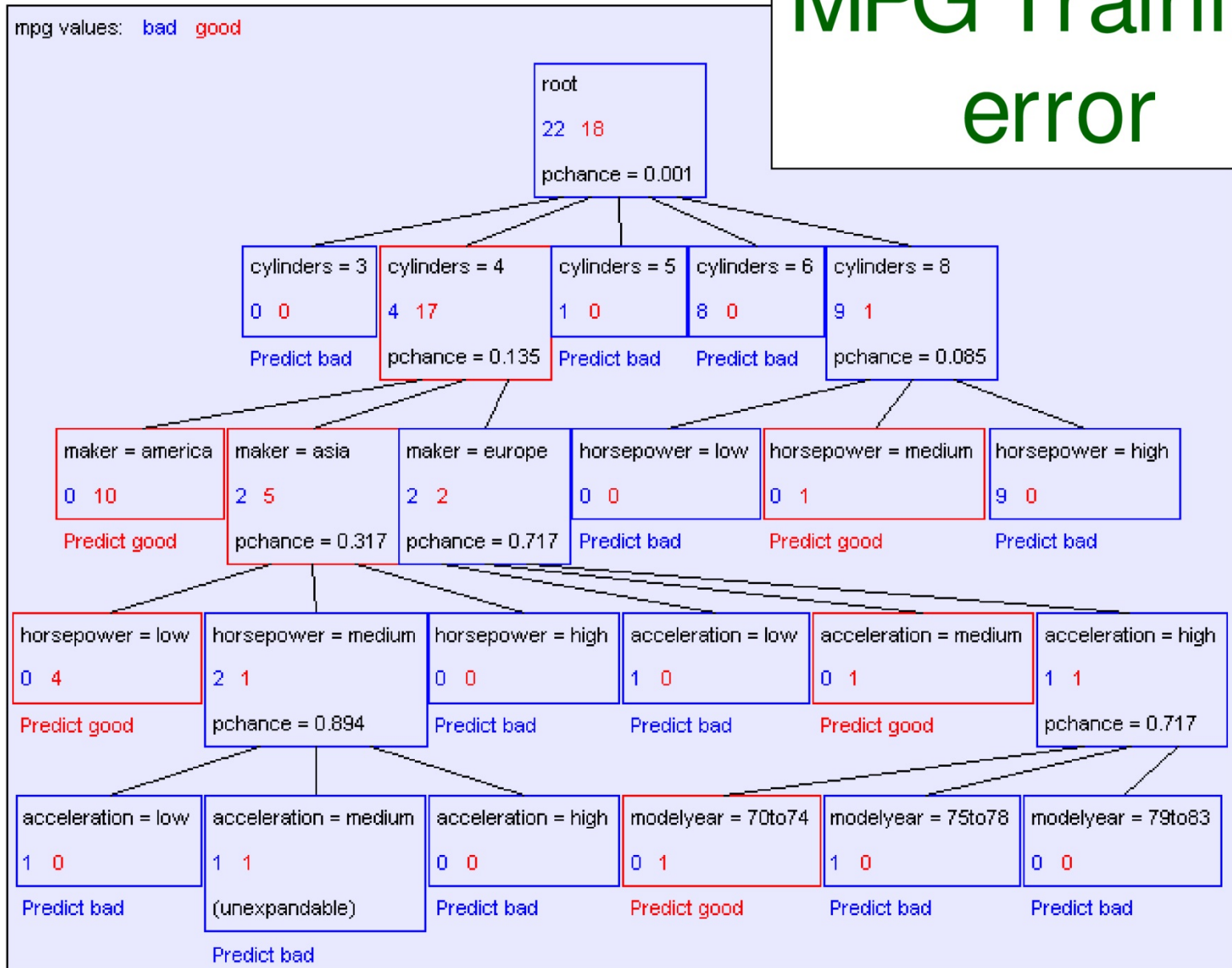
BuildTree( $DS_i$ , *Output*)

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

# Training Set Error

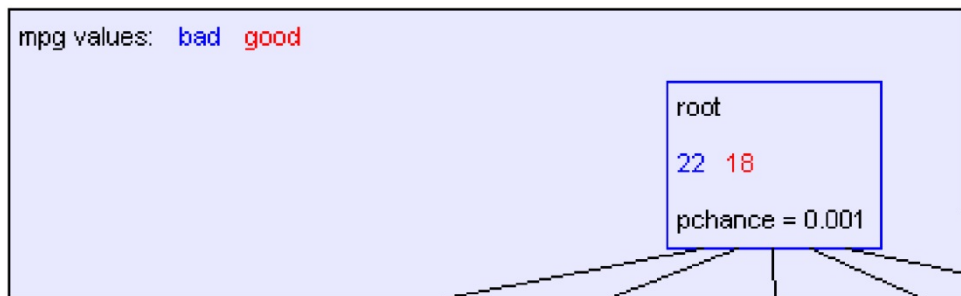
- For each record, follow the decision tree to see what it would predict  
For what number of records does the decision tree's prediction disagree with the true value in the database?
- This quantity is called the *training set error*. The smaller the better.

# MPG Training error

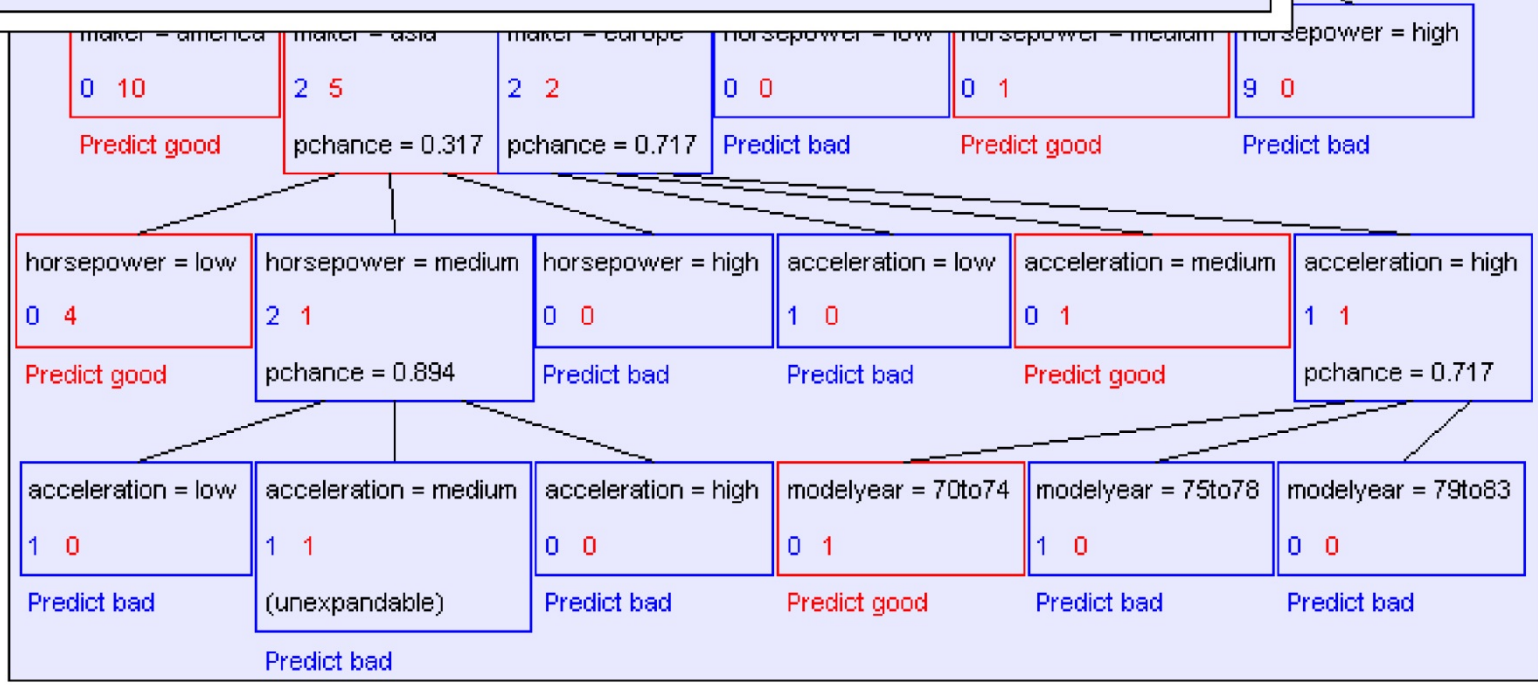




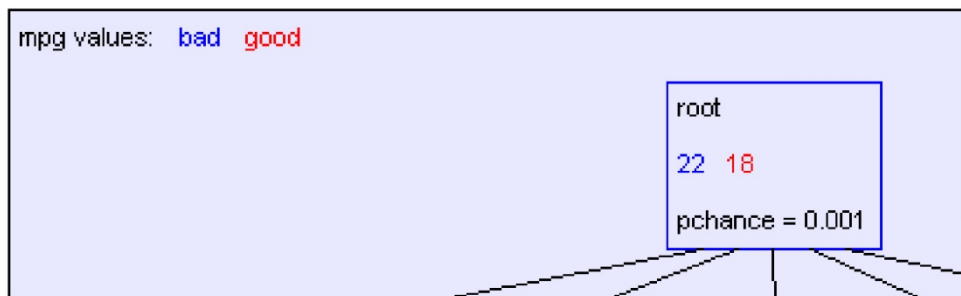
# MPG Training error



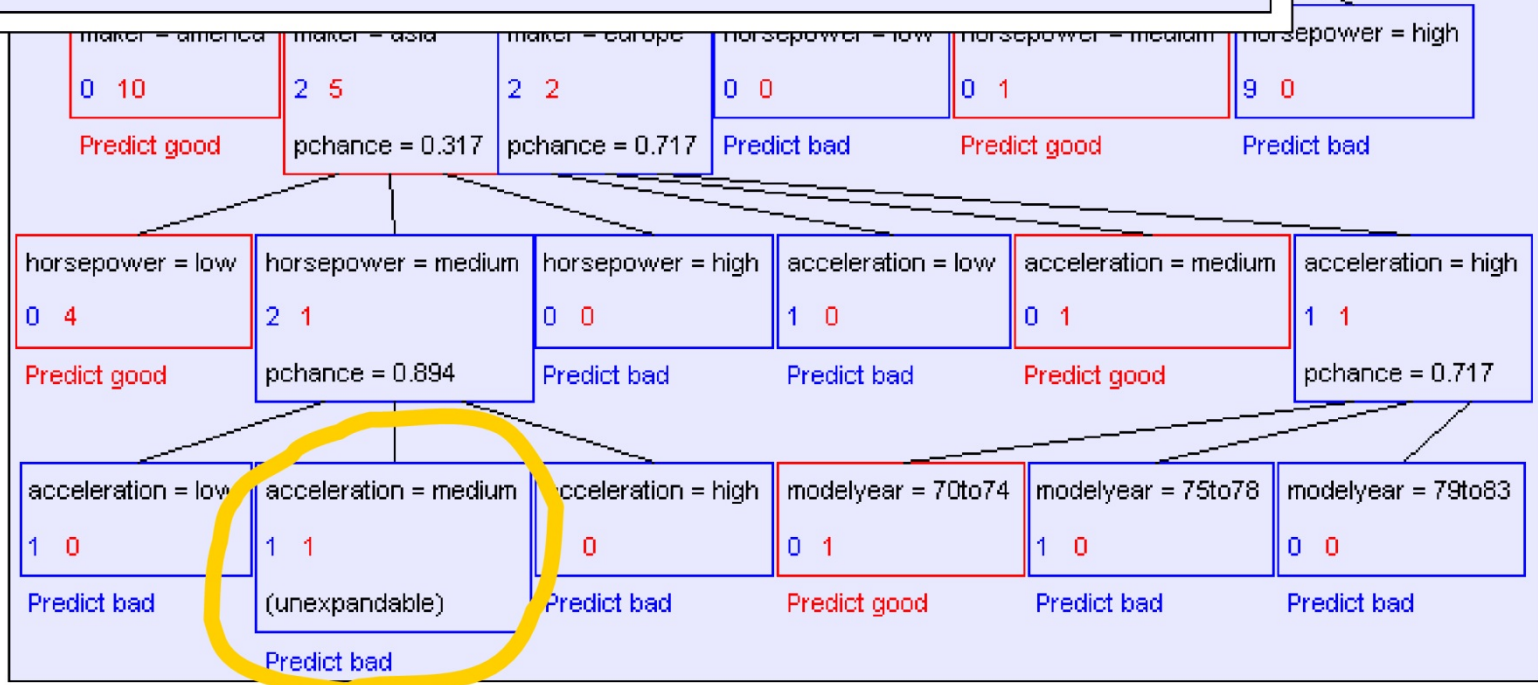
|                | Num Errors | Set Size | Percent Wrong |
|----------------|------------|----------|---------------|
| Training Set 1 | 1          | 40       | 2.50          |



# MPG Training error



|                | Num Errors | Set Size | Percent Wrong |
|----------------|------------|----------|---------------|
| Training Set 1 | 1          | 40       | 2.50          |





# Stop and reflect: Why are we doing this learning anyway?

- It is not usually in order to predict the training data's output on data we have already seen.

# Stop and reflect: Why are we doing this learning anyway?

- It is not usually in order to predict the training data's output on data we have already seen.
- It is more commonly in order to predict the output value for **future data** we have not yet seen.

# Test Set Error

- Suppose we are forward thinking.
- We hide some data away when we learn the decision tree.
- But once learned, we see how well the tree predicts that data.
- This is a good simulation of what happens when we try to predict future data.
- And it is called **Test Set Error**.

# MPG Test set error

mpg values: bad good

root  
22 18  
pchance = 0.001

|              | Num Errors | Set Size | Percent Wrong |
|--------------|------------|----------|---------------|
| Training Set | 1          | 40       | 2.50          |
| Test Set     | 74         | 352      | 21.02         |

horsepower = high

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

# Overfitting

- Definition: If your machine learning algorithm fits noise (i.e. pays attention to parts of the data that are irrelevant) it is **overfitting**.
- Fact (theoretical and empirical): If your machine learning algorithm is overfitting then it may perform less well on test set data.

# How do I know I am overfitting?

- The best way is to have a held-out "development" test set to measure generalization performance on
  - This should be held separate from the final test set
- An interesting empirical fact about decision trees, is that larger trees tend to overfit more, so trying to create small trees is a good idea
  - It is easy to see that very small trees can't overfit
  - for instance, always choosing majority class is a very small tree
- People often talk about the depth of the tree (distance of the longest path from root to leaf) because of this

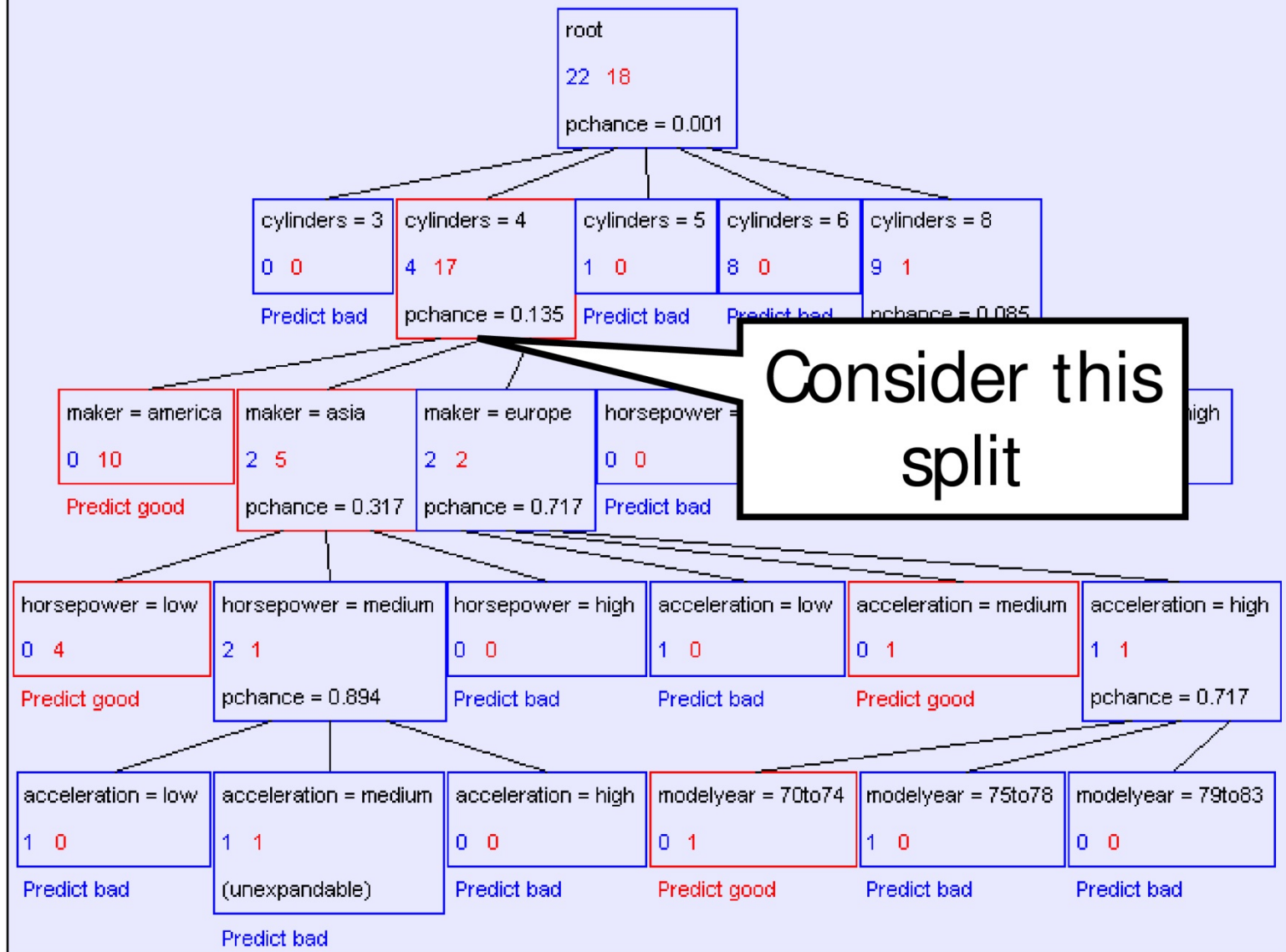
# Avoiding overfitting

- Usually we do not know in advance which are the irrelevant variables
- ...and it may depend on the context

For example, if  $y = a \text{ AND } b$  then  $b$  is an irrelevant variable only in the portion of the tree in which  $a=0$

But we can use simple statistics to warn us that we might be overfitting.

mpg values: bad good

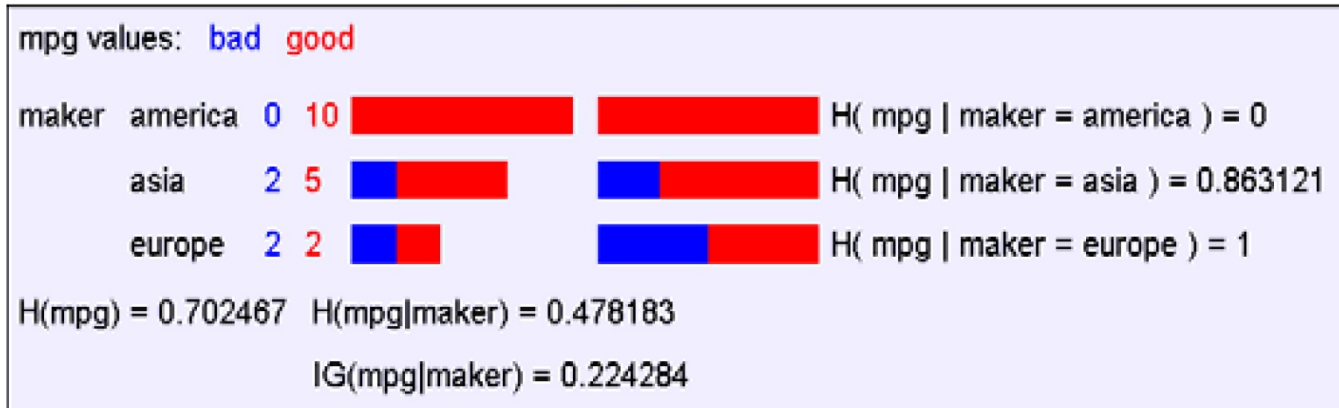


Consider this split

high

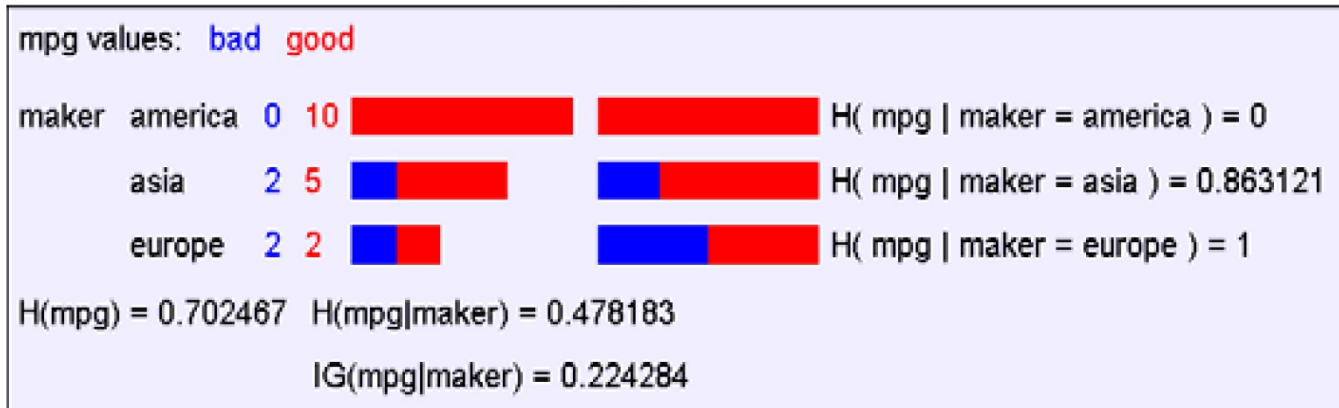


# A chi-squared test



- 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?

# A chi-squared 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-squared test, the answer is 13.5%.

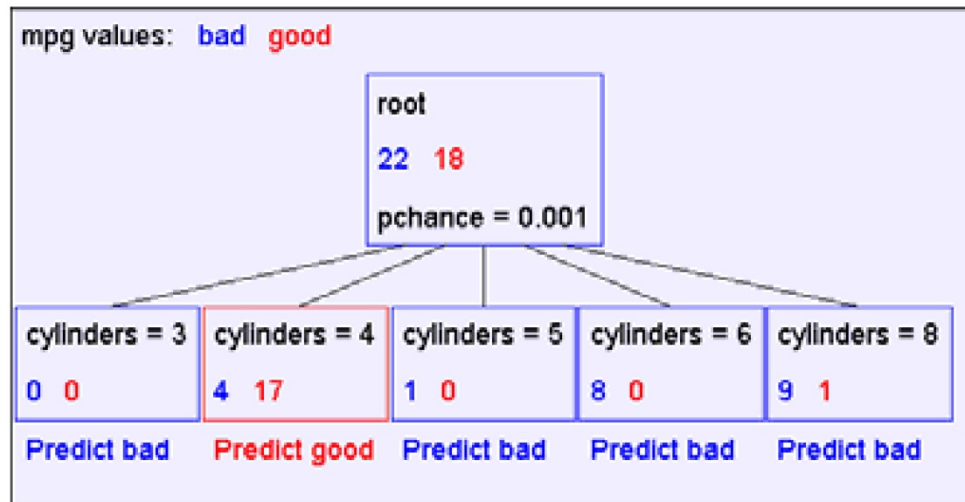
# Using Chi-squared to avoid overfitting

- 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  $p_{chance} > MaxPchance$ .
  - Continue working your 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  $\text{MaxPchance} = 0.1$ , you will see the following MPG decision tree:



Note the improved test set accuracy compared with the unpruned tree

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

# More on pruning

- Another way to prune is to work with a special pruning set, which is separate from the data used to grow the tree
- This algorithm is called **Reduced Error Pruning**, and it is due to Quinlan
  - First grow the tree completely (as we did in our example)
  - Then, starting with each split at the bottom, classify the pruning set, compare:
    - 1) the accuracy of your current tree
    - 2) the accuracy of your current tree with this split removed (i.e., with the decision being the majority class before the split)
    - If (2) wins, then remove the split
  - Then move on to examine each node in the tree in turn, applying the same test
- This approach is very easy to understand, and can also be efficiently applied
- Big disadvantage: must separate data into data for growing tree, and data used to control pruning
- See Esposito et al. 1997 for an influential study of pruning techniques for decision trees

# Conclusions

- Decision trees are the single most popular data mining tool
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- It's possible to get in trouble with overfitting
- They do classification: predict a categorical output from categorical and/or real inputs

# Things I didn't discuss - I

- How to deal with **real-valued inputs**
  - Either: discretize these into buckets before building a decision tree
    - As was done on the gasoline usage data set we just saw
  - Or: while building the decision tree, use less-than checks
    - E.g., try all of **age < 24** versus **age < 25** versus **age < 26** (etc...) and find the best split of the **age** variable (according to information gain)
    - But the details are complex and this requires many assumptions
- Information Gain can sometimes incorrectly favor splitting on **features which have many possible values**
  - There are alternative criteria that are sometimes preferred
  - There are also ways to correct Information Gain for this problem

# Things I didn't discuss - II

- There are very interesting techniques for further improving on decision trees
  - One way is to build a "random forest" of decision trees on different subsets of the data and even different subsets of the features
  - Then have the trees in the forest vote on the final classification
  - This often works really well!
- Finally, there are also **very different solutions** that work well for classification like this
  - For instance, consider Naive Bayes or **linear models** in general
  - Linear models associate one weight with each feature value as we saw in the previous lecture
  - The same basic ideas about generalization and overfitting apply!
  - We'll discuss these in detail in the next lecture
  - Following that, we'll discuss deep learning (non-linear models)



- Some credits:
  - See Ata Kaban's (Birmingham) machine learning class particularly for the intuitive discussion of the Winston sunburn problem and Information Gain
  - See Andrew W. Moore's (Carnegie Mellon) website for a longer presentation of his slides on decision trees, and slides on many other machine learning topics:  
<http://www.autonlab.org/tutorials>

- Thank you for your attention!