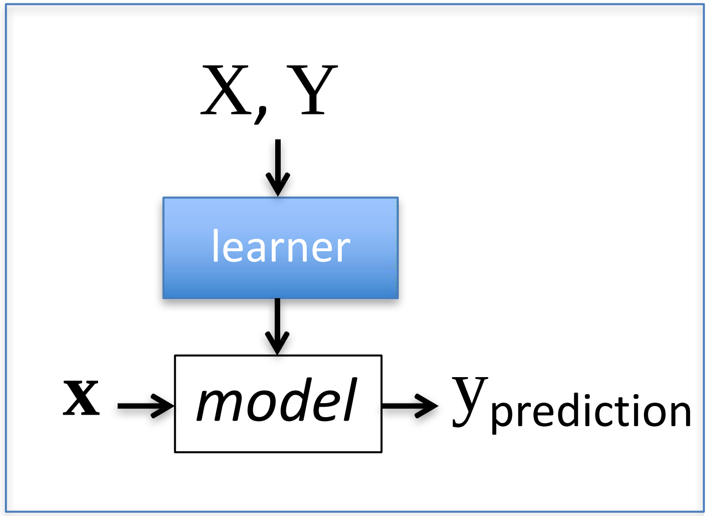
# Supervised Learning

# Decision Tree
# ID3

# Stages of Machine Learning

**Given:** labeled training data $X, Y$

- Assumes each $\boldsymbol{x}_i \leftarrow D(X)$ with $y_i = f_{target}(\boldsymbol{x}_i)$




**Train the model:**

*model* ← *classifier*.train(X, Y)

**Apply the model to new data:**

- Given: new unlabeled instance $\boldsymbol{x} \leftarrow D(X)$

$y_{prediction}$ ← *model*.predict($\boldsymbol{x}$)

# Sample Dataset (was Tennis Played?)

- Columns denote features $X_i$
- Rows denote labeled instances $\boldsymbol{x}_i, y_i$
- Class label denotes whether a tennis game was played

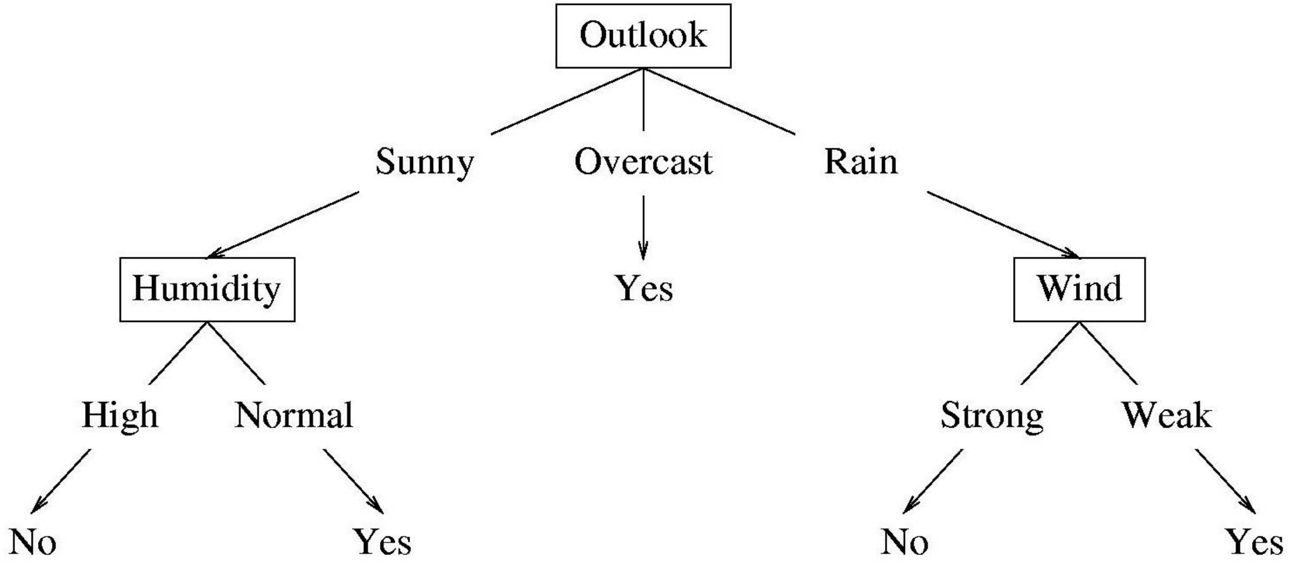
| Predictors | | | | Response |
|---|---|---|---|---|
| **Outlook** | **Temperature** | **Humidity** | **Wind** | **Class** |
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Overcast | Hot | High | Weak | Yes |
| Rain | Mild | High | Weak | Yes |
| Rain | Cool | Normal | Weak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

$\boldsymbol{x}_i, y_i$

# Decision Tree

# Decision Tree

- A possible decision tree for the data:



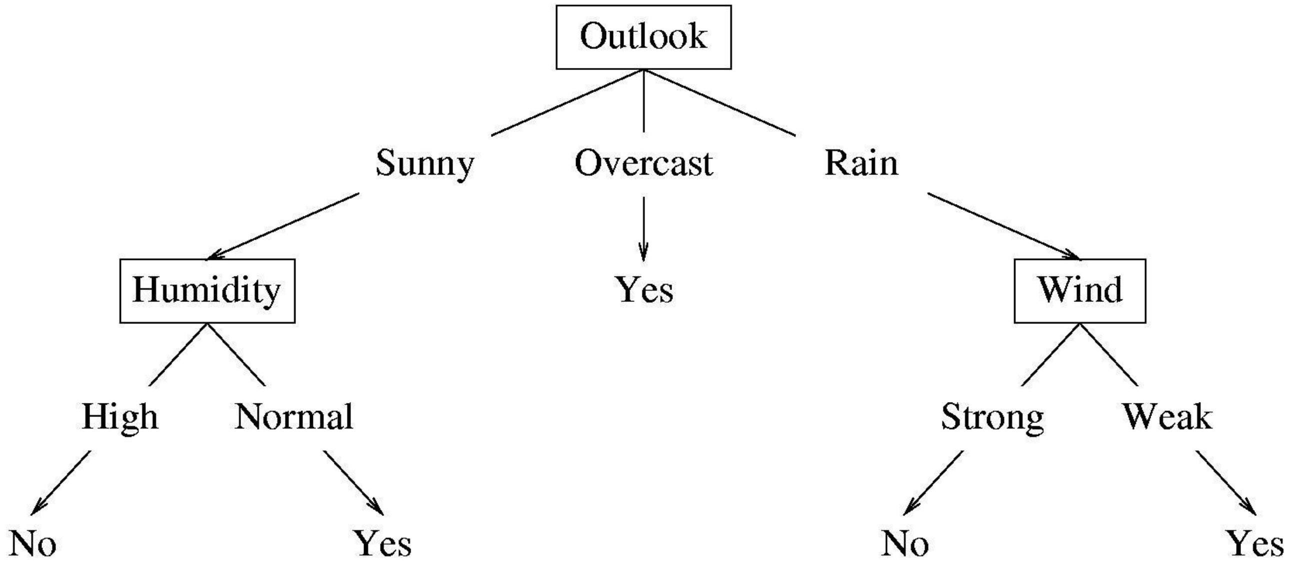

| Predictors | | | | Response |
|---|---|---|---|---|
| **Outlook** | **Temperature** | **Humidity** | **Wind** | **Class** |
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Overcast | Hot | High | Weak | Yes |
| Rain | Mild | High | Weak | Yes |
| Rain | Cool | Normal | Weak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

- Each internal node: test one attribute $X_i$
- Each branch from a node: selects one value for $X_i$
- Each leaf node: predict $Y$

Based on slide by Tom Mitchell

# Decision Tree

- A possible decision tree for the data:




- What prediction would we make for

  <outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

Based on slide by Tom Mitchell

# Basic Algorithm for Top-Down Learning of Decision Trees

[ID3, C4.5 by Quinlan]

*node* = root of decision tree

Main loop:

1. *A* ← the "best" decision attribute for the next node.
2. Assign *A* as decision attribute for *node*.
3. For each value of *A*, create a new descendant of *node*.
4. Sort training examples to leaf nodes.
5. If training examples are perfectly classified, stop. Else, recurse over new leaf nodes.

How do we choose which attribute is best?

# Choosing the Best Attribute

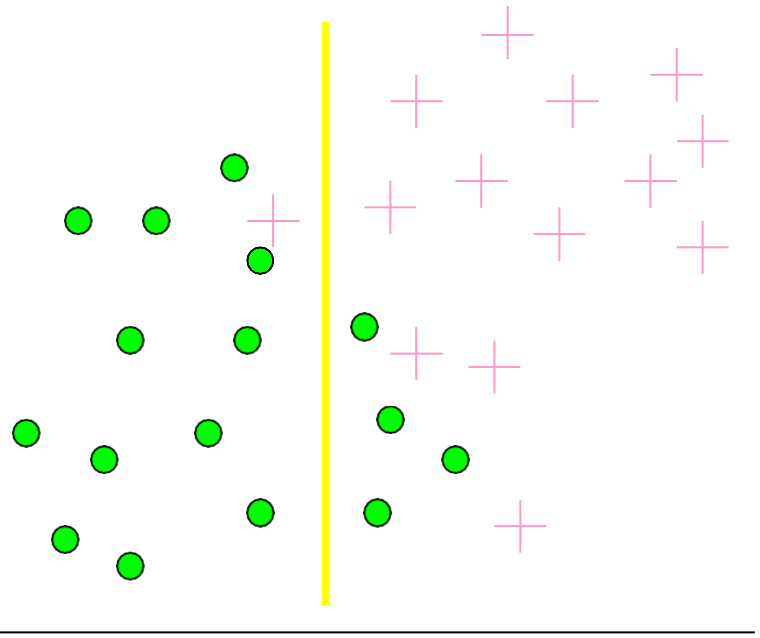
**Key problem**: choosing which attribute to split a given set of examples

- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest expected *information gain*
    - i.e., attribute that results in smallest expected size of subtrees rooted at its children
- The ID3 algorithm uses the Max-Gain method of selecting the best attribute

# Information Gain

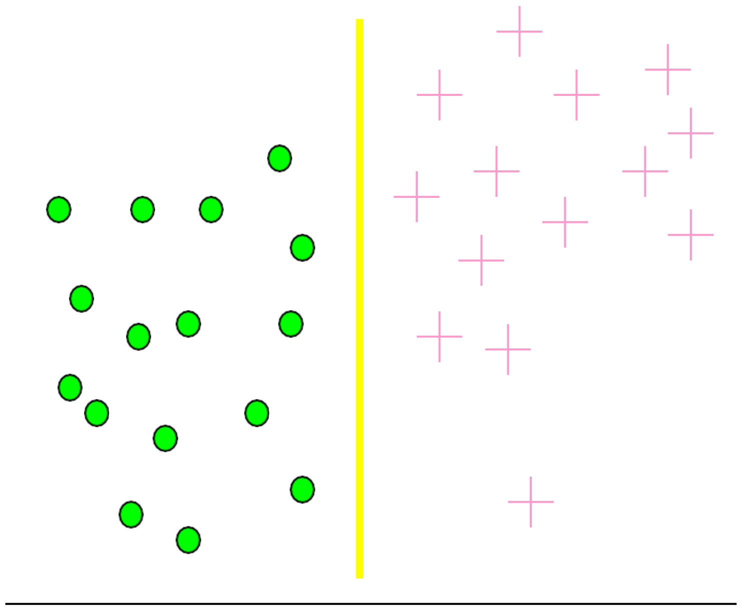
Which test is more informative?

**Split over whether Balance exceeds 50K**



Less or equal 50K    Over 50K

**Split over whether applicant is employed**



Unemployed    Employed

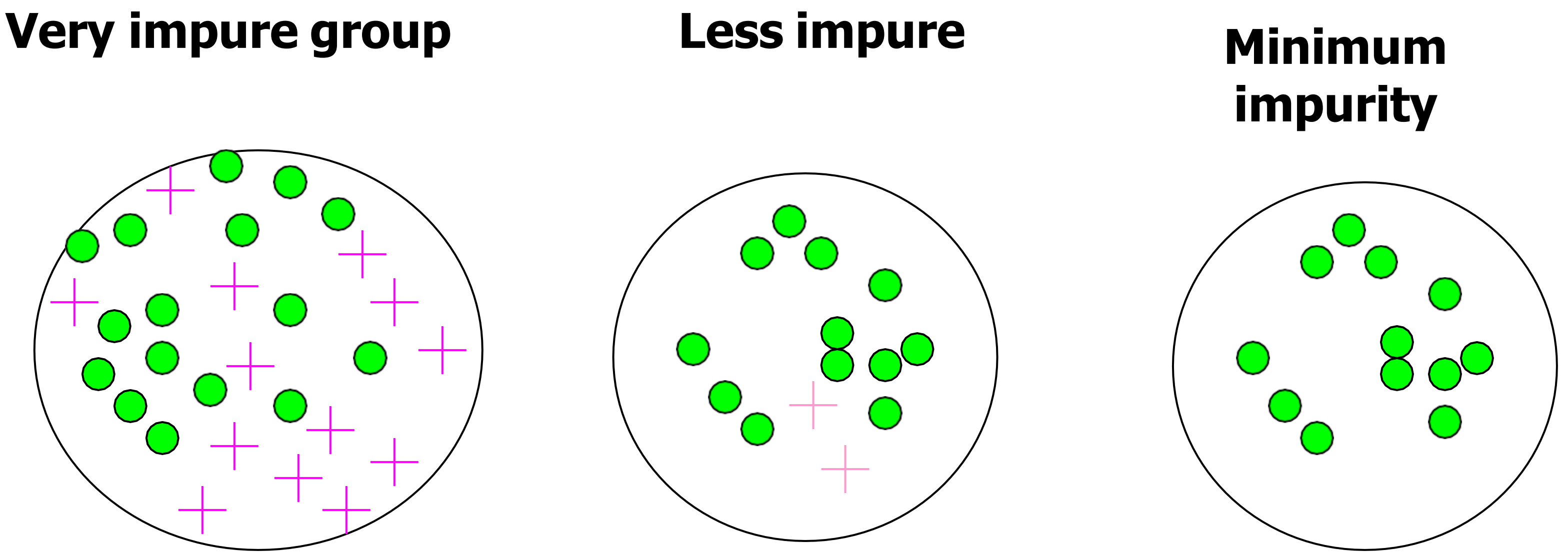


Based on slide by Pedro Domingos

# Information Gain

## **Impurity/Entropy** (informal)

– Measures the level of **impurity** in a group of examples

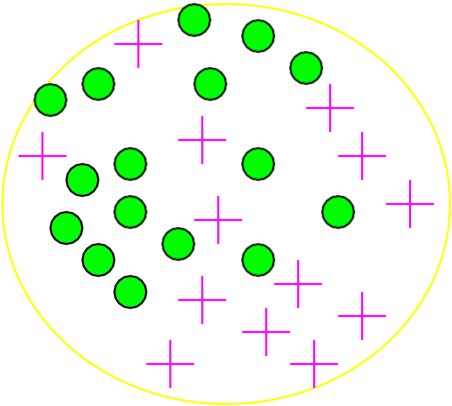





Based on slide by Pedro Domingos

# Entropy: a common way to measure impurity

- Entropy = $$\sum_i - p_i \log_2 p_i$$



  $p_i$ is the probability of class i

  Compute it as the proportion of class i in the set.

- Entropy comes from information theory. The higher the entropy the more the information content.

What does that mean for learning from examples?



Based on slide by Pedro Domingos

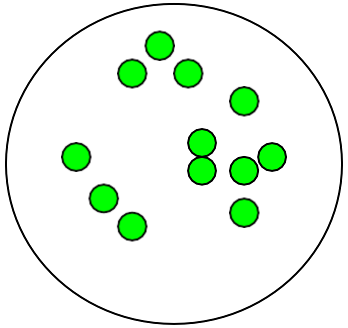
# 2-Class Cases:

$$\text{Entropy} \quad H(x) = -\sum_{i=1}^{n} P(x=i)\log_2 P(x=i)$$

- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

  not a good training set for learning

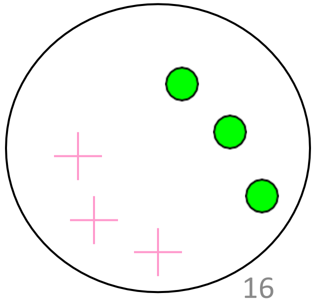
**Minimum impurity**



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

  good training set for learning

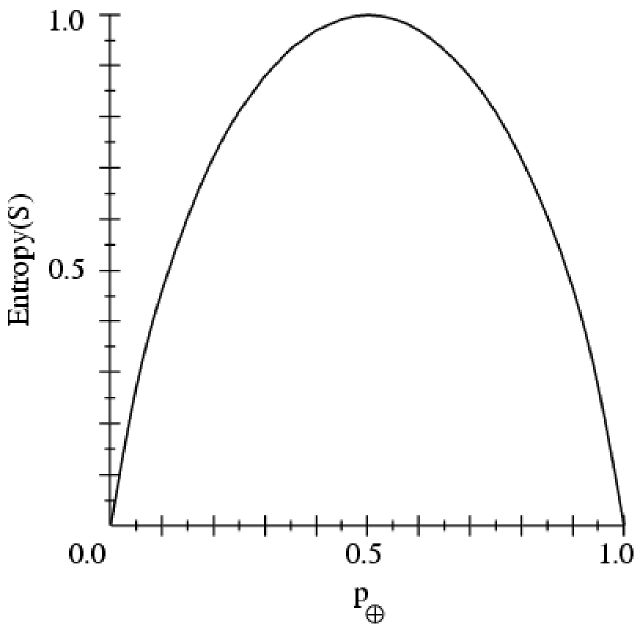
**Maximum impurity**





Based on slide by Pedro Domingos

# Sample Entropy




- $S$ is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in $S$
- $p_{\ominus}$ is the proportion of negative examples in $S$
- Entropy measures the impurity of $S$

$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

# Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
- Information gain tells us how important a given attribute of the feature vectors is.
- We will use it to decide the ordering of attributes in the nodes of a decision tree.

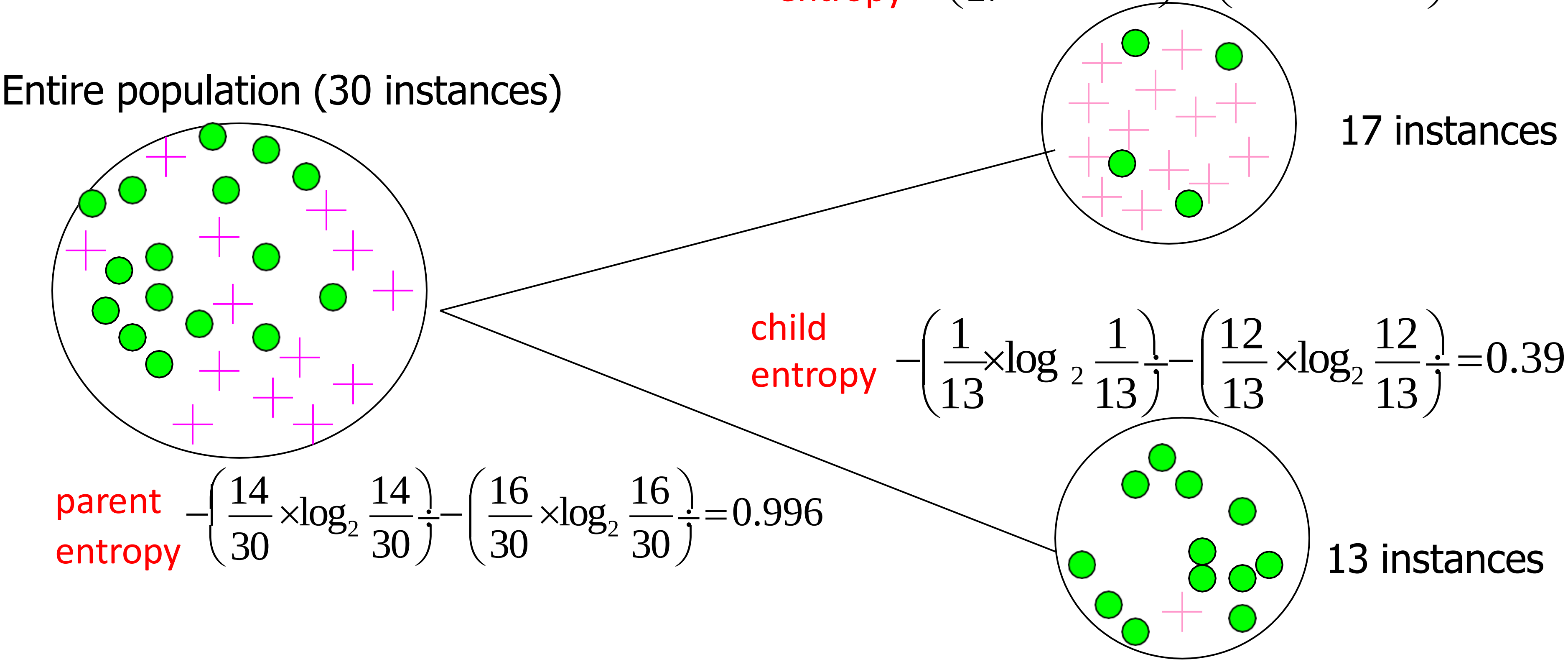


Based on slide by Pedro Domingos

# Calculating Information Gain

**Information Gain** = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\times\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\times\log_2\frac{4}{17}\right)=0.787$




child entropy $-\left(\frac{1}{13}\times\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\times\log_2\frac{12}{13}\right)=0.391$

parent entropy $-\left(\frac{14}{30}\times\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\times\log_2\frac{16}{30}\right)=0.996$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\times 0.787\right)+\left(\frac{13}{30}\times 0.391\right)=0.615$

**Information Gain= 0.996 - 0.615 = 0.38**



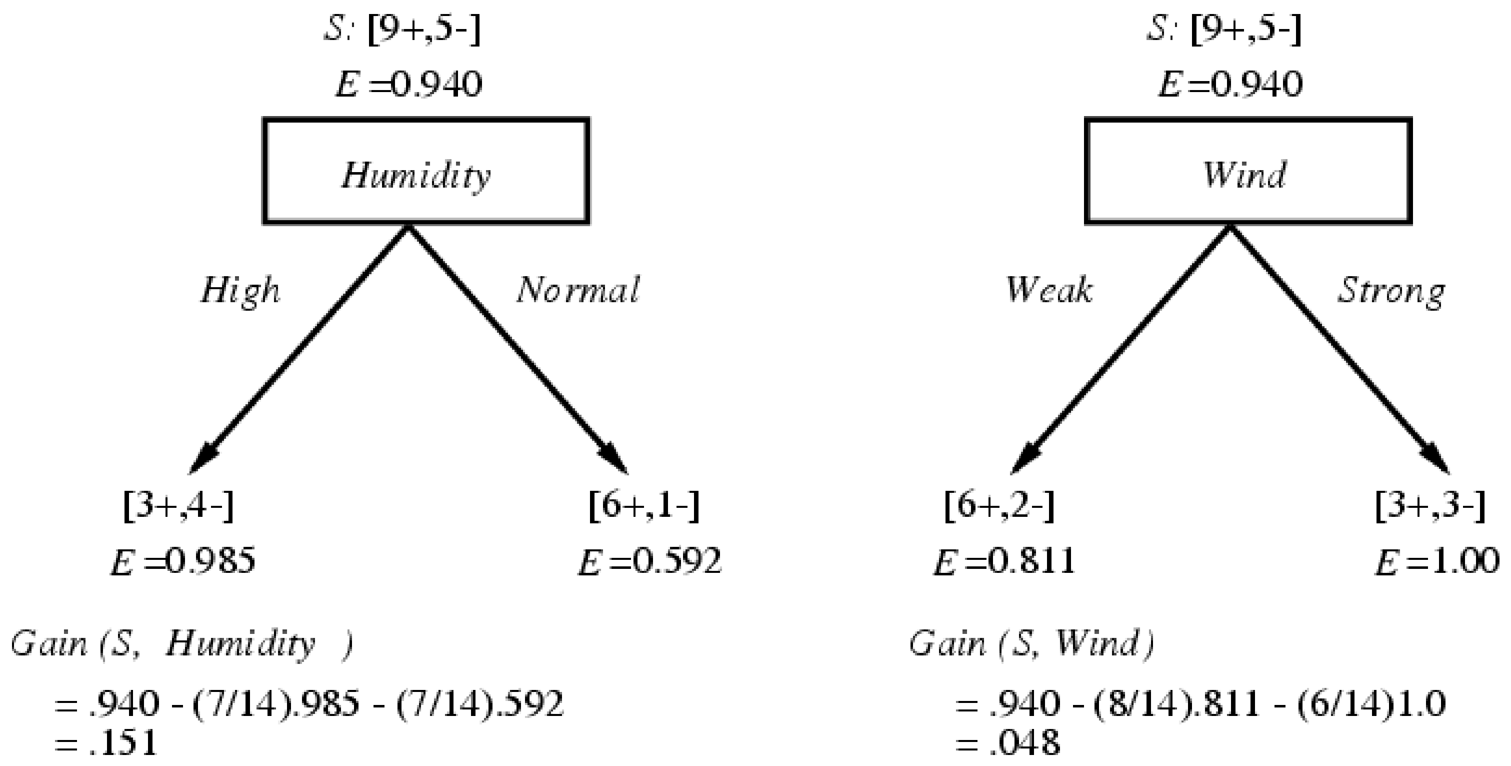
Based on slide by Pedro Domingos

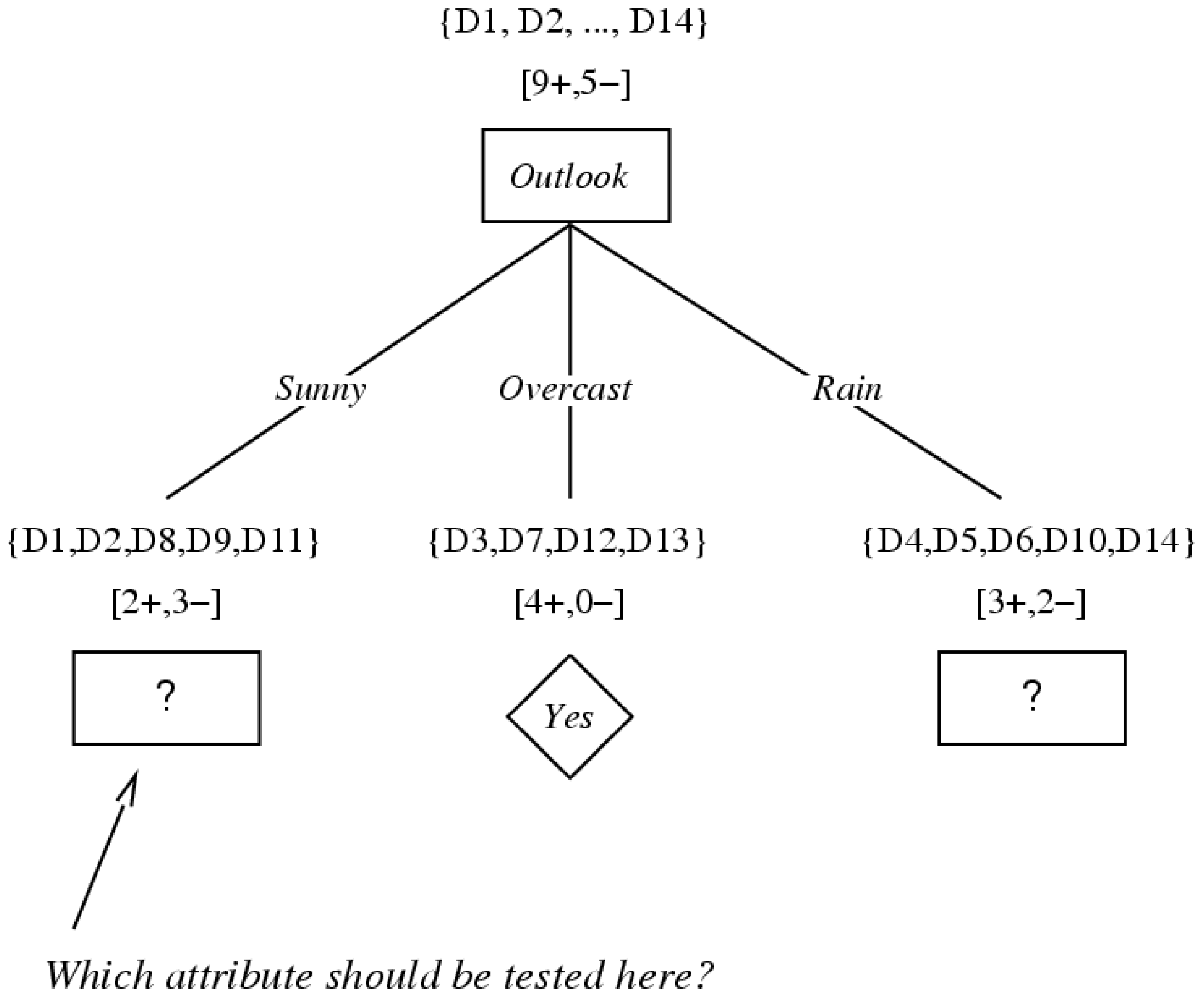
# Training Examples

| Day | Outlook | Temperature | Humidity | Wind | PlayTenr |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

# Selecting the Next Attribute

**Which attribute is the best classifier?**





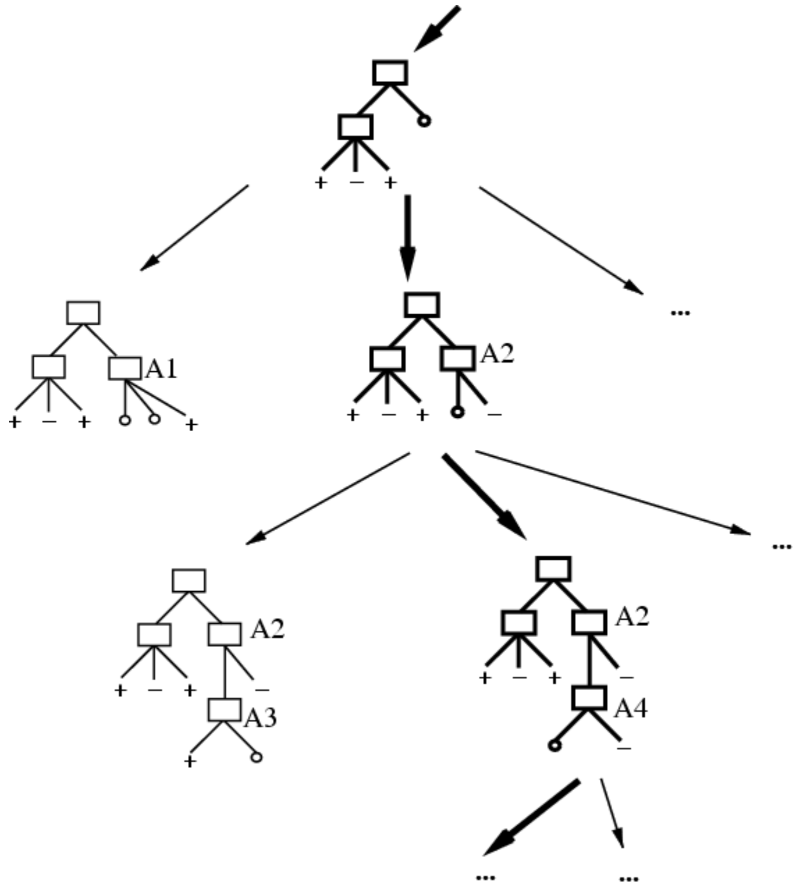

$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain\ (S_{sunny}, Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$

$Gain\ (S_{sunny}, Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$

$Gain\ (S_{sunny}, Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$

# Which Tree Should We Output?




- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

# Preference bias: Ockham's Razor

- Principle stated by William of Ockham (1285-1347)
  - "*non sunt multiplicanda entia praeter necessitatem* "
  - entities are not to be multiplied beyond necessity
  - AKA Occam's Razor, Law of Economy, or Law of Parsimony

**Idea**: The simplest consistent explanation is the best

- Therefore, the smallest decision tree that correctly classifies all of the training examples is best
  - Finding the provably smallest decision tree is NP-hard
  - ...So instead of constructing the absolute smallest tree consistent with the training examples, construct one that is pretty small