Machine learning

Prepared by : Dr. Hanaa Bayomi
Updated By: Prof Abeer ElKorany

Lecture 7: Decision tree

Content

1. What is Decision Tree
2. Entropy calculation for discrete features
3. Information Gain
4. Entropy Calculation for continuous features
5. Overfitting

DECISION TREE

There are Two kinds of **nonparametric** learning: k-Nearest Neighbor and Decision Trees.

Weather	Raining	Sunny	Windy	Decision
Cloudy	No	No	No	Stay indoors
Rainy	Yes	No	No	Stay indoors
Sunny	No	Yes	No	Go for a walk
Windy	No	No	Yes	Go for a walk
Snowy	No	No	No	Stay indoors

How ML could be interpreted

Question: Is the weather suitable for a walk?

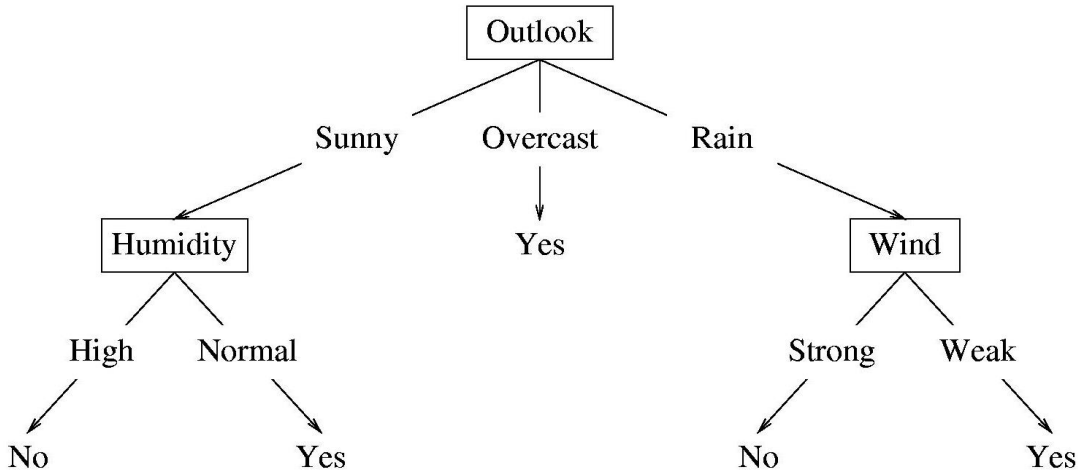
Question: Is it raining?

- ├─ Yes: Stay indoors.
- └─ No:
  - Question: Is it sunny?
    - ├─ Yes: Go for a walk.
    - └─ No:
      - Question: Is it windy?
        - ├─ Yes: Go for a walk.
        - └─ No: Stay indoors.

Weather	Raining	Sunny	Windy	Decision
Cloudy	No	No	No	Stay indoors
Rainy	Yes	No	No	Stay indoors
Sunny	No	Yes	No	Go for a walk
Windy	No	No	Yes	Go for a walk
Snowy	No	No	No	Stay indoors

DECISION TREE

- Decision tree algorithm (C4.5) is a Greedy recursive algorithm that successively splits the training data into "hyperrectangles"



Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

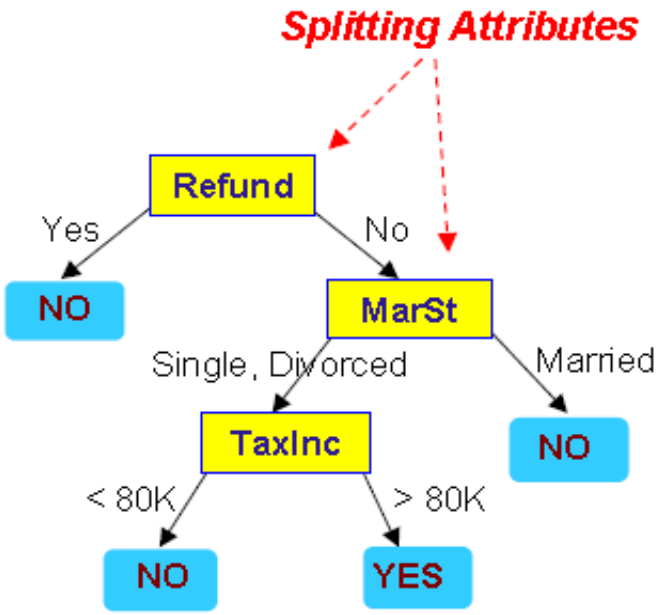
Decision Tree constructs the tree by recursively selecting the best features to split the data and make decisions based on the information gain

DECISION TREE: EXAMPLE

Key Idea: good features partition the data into subsets that are either “all positive” or “all negative” (ideally)

	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

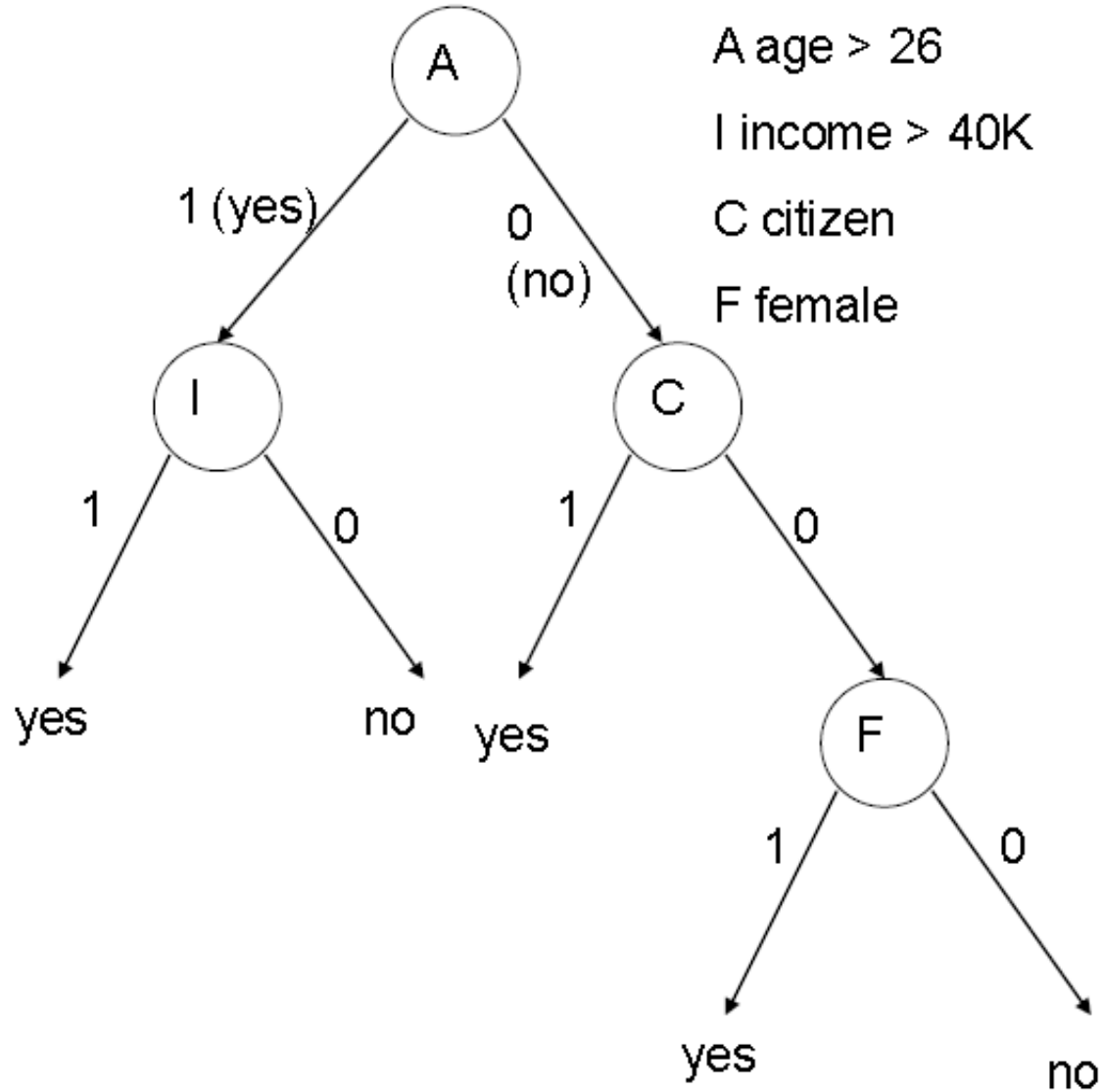


Model: Decision Tree

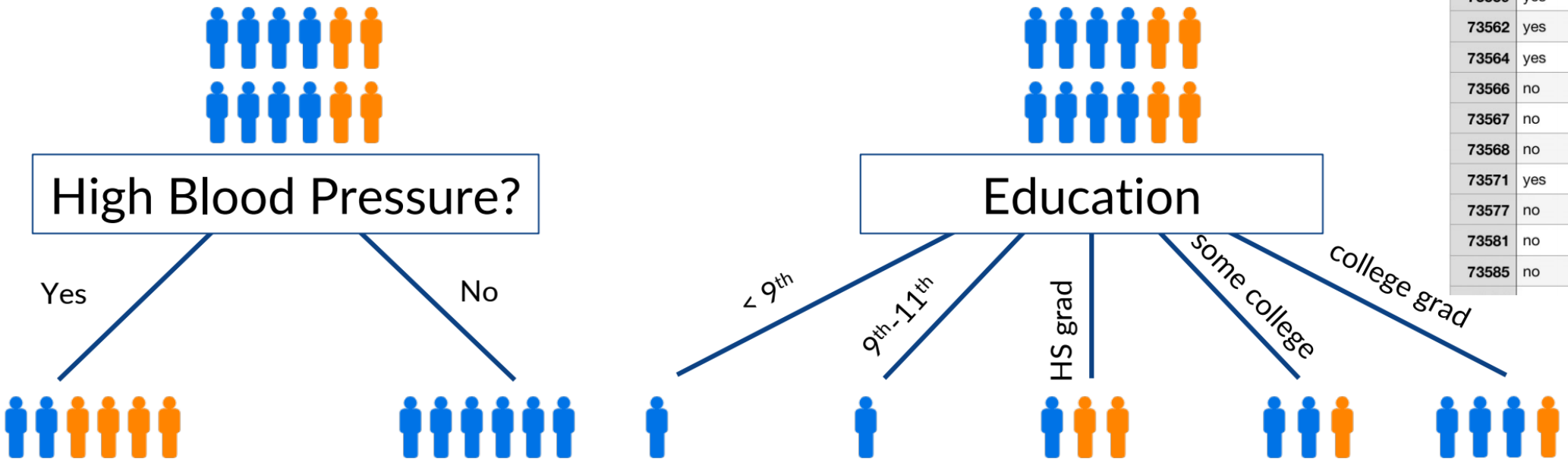
- There can be many different trees that all work equally well!

Structure of a decision tree

- Internal nodes correspond to attributes (features)
- Leafs correspond to classification outcome
- edges denote assignment



Key Idea: good features partition the data into subsets that are either “all positive” or “all negative” (ideally)



Subset of Data

ID (SEQN)	HIGH_BP (BPQ020)	EDUCATION (DMDEDUC2)	DIABETIC
73557	yes	high school graduate / GED	yes
73558	yes	high school graduate / GED	yes
73559	yes	some college or AA degree	yes
73562	yes	some college or AA degree	no
73564	yes	college graduate or above	no
73566	no	high school graduate / GED	no
73567	no	9th-11th grade	no
73568	no	college graduate or above	no
73571	yes	college graduate or above	yes
73577	no	Less than 9th grade	no
73581	no	college graduate or above	no
73585	no	some college or AA degree	no

Predict if John will play tennis

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Predict if John will play tennis

- Hard to guess
- Divide & conquer:
  - split into subsets
  - are they pure? (all yes or all no)
  - if yes: stop
  - if not: repeat
- See which subset new data falls into

Key Idea: good features partition the data into subsets that are either "all positive" or "all negative" (ideally)

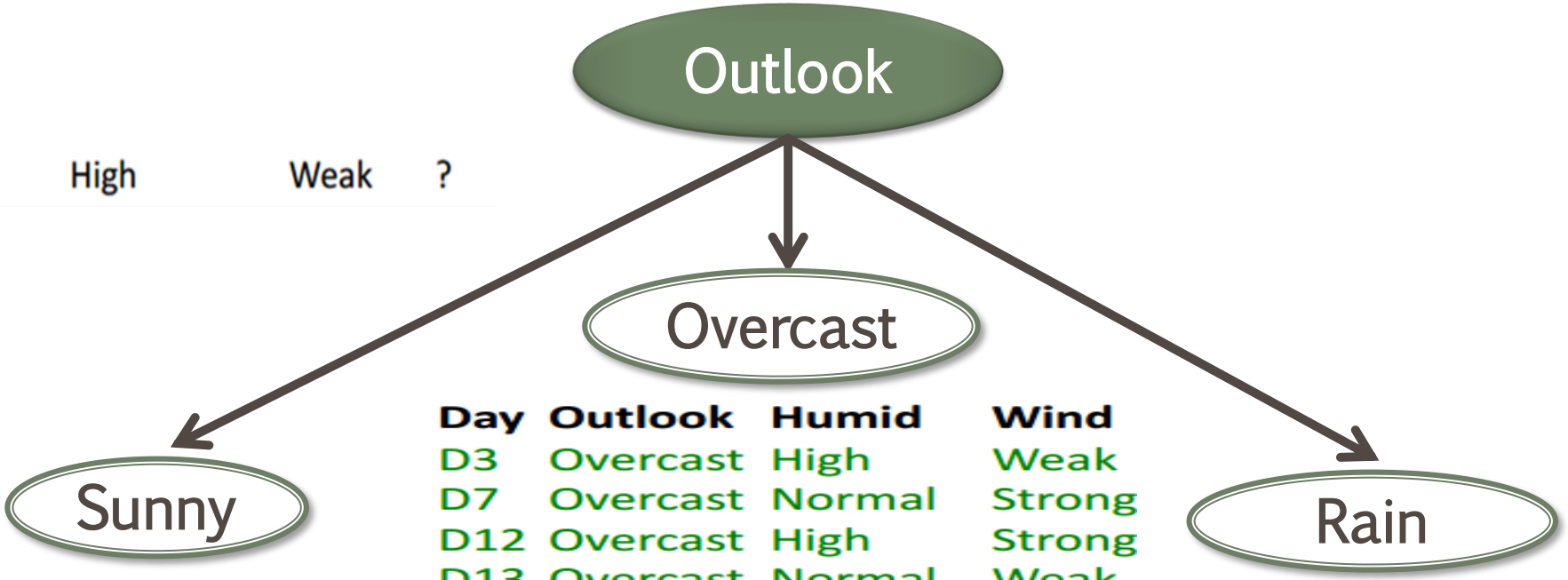
Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

New data:

D15	Rain	High	Weak	?

9 yes / 5 no



New data:

D15	Sunny	High	Weak	?

Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

4 yes / 0 no
pure subset

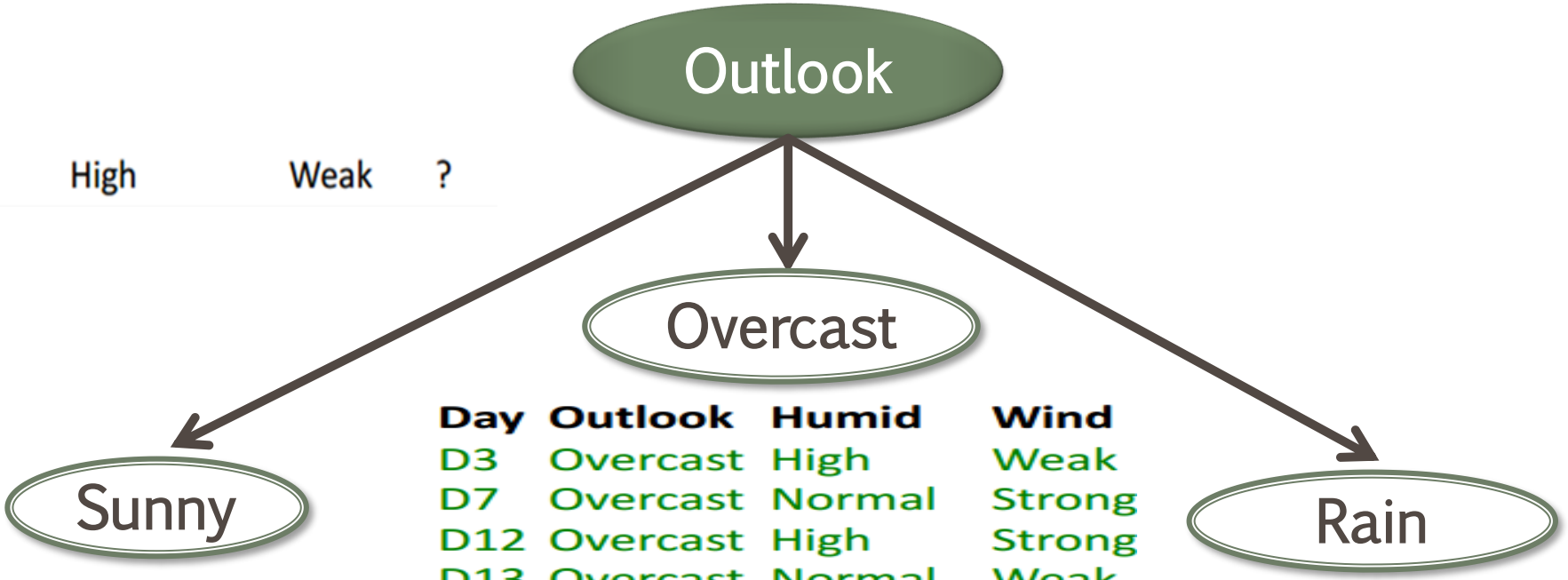
Day	Outlook	Humid	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

2 yes / 3 no
split further

Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no
split further

9 yes / 5 no



New data:

D15	Sunny	High	Weak	?

Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

4 yes / 0 no
pure subset

Day	Outlook	Humid	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

2 yes / 3 no
split further

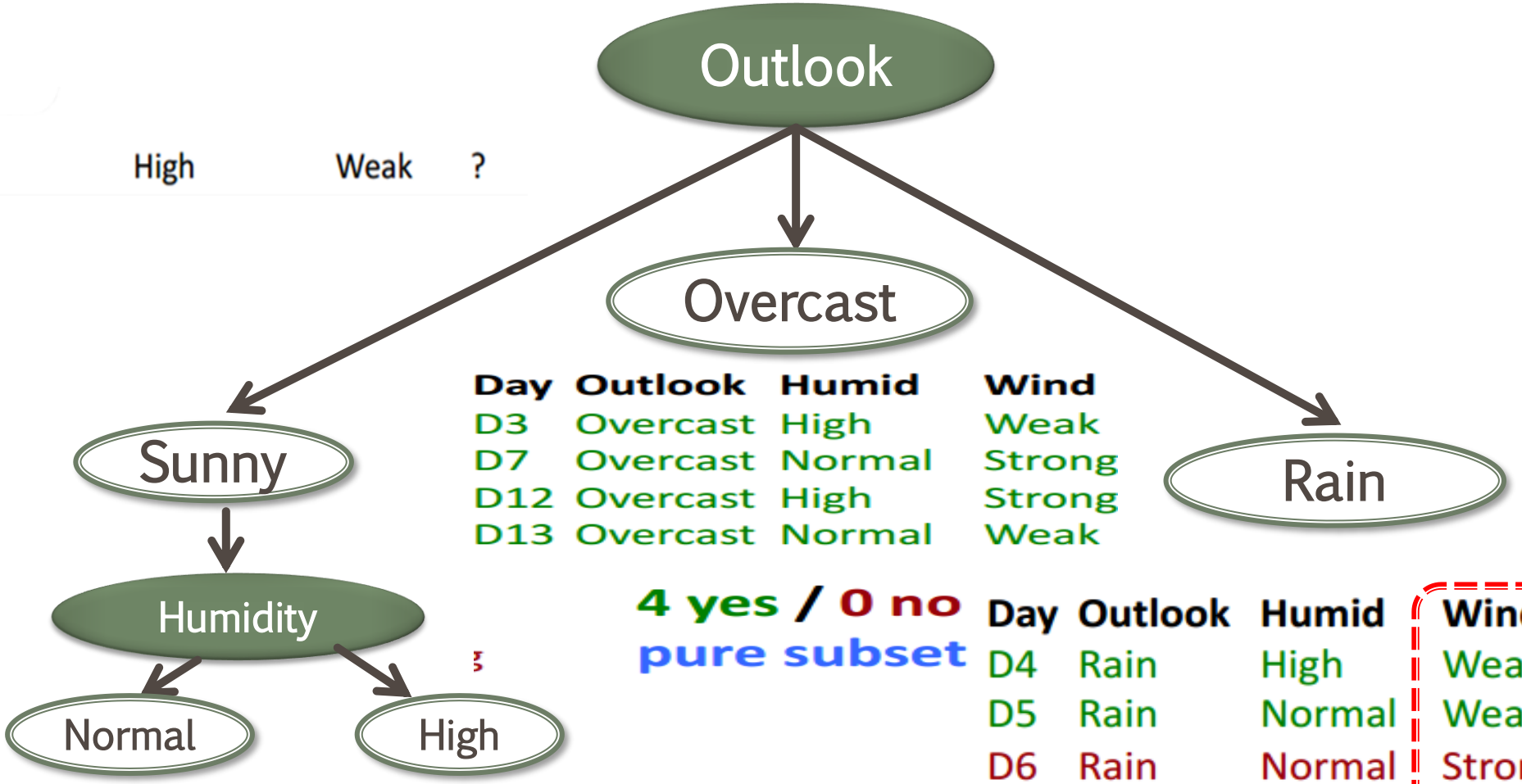
Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no
split further

9 yes / 5 no

New data:

D15	Rain	High	Weak	?



Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

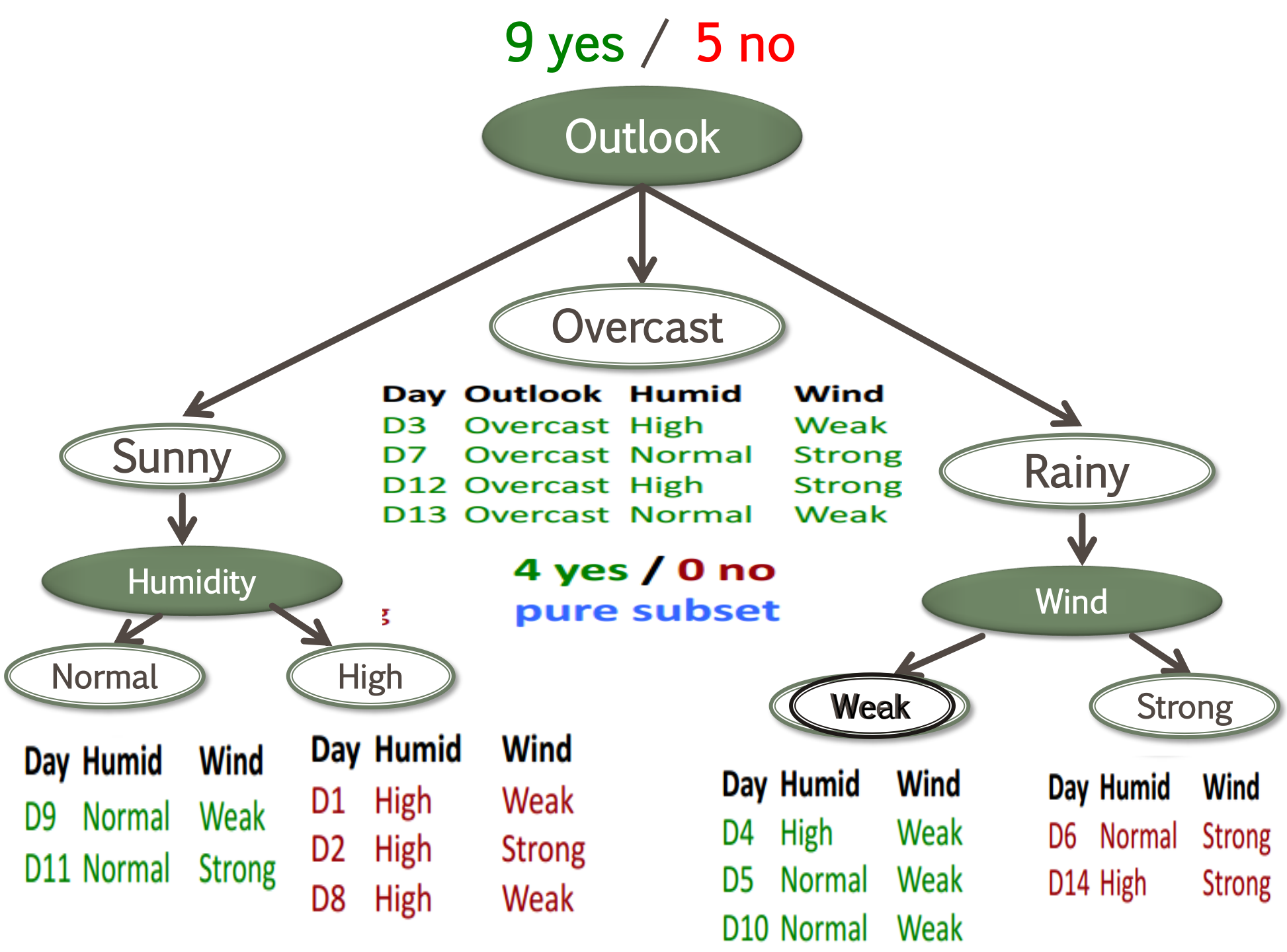
4 yes / 0 no
pure subset

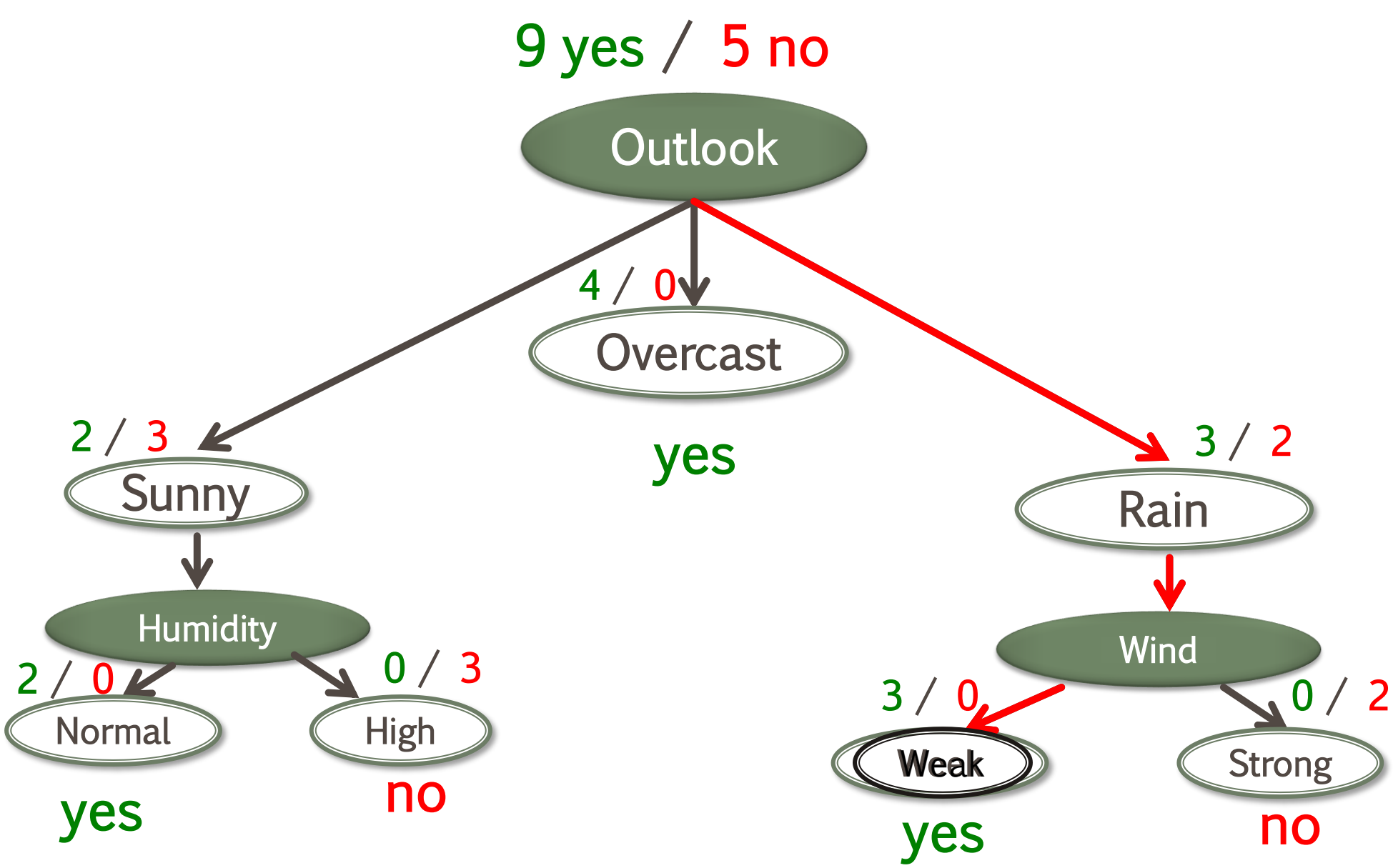
Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no
split further

Day	Humid	Wind
D9	Normal	Weak
D11	Normal	Strong

Day	Humid	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak


9 yes / 5 no
Outlook
Overcast
Sunny
Rainy
Humidity
Wind
Normal
High
Weak
Strong
Day Outlook Humid Wind
D3 Overcast High Weak
D7 Overcast Normal Strong
D12 Overcast High Strong
D13 Overcast Normal Weak
4 yes / 0 no
pure subset
Day Humid Wind
D9 Normal Weak
D11 Normal Strong
Day Humid Wind
D1 High Weak
D2 High Strong
D8 High Weak
Day Humid Wind
D4 High Weak
D5 Normal Weak
D10 Normal Weak
Day Humid Wind
D6 Normal Strong
D14 High Strong



New data:

Day	Outlook	Humid	Wind	
D15	Rain	High	Weak	yes

ID3 Algorithm

- Split (node, {examples}):
  1. A ← the best attribute for splitting the {examples}
  2. Decision attribute for this node ← A
  3. For each value of A, create new child node
  4. Split training {examples} to child nodes
  5. If examples perfectly classified: STOP
     else: iterate over new child nodes
     Split (child_node, {subset of examples})
- Ross Quinlan (ID3: 1986), (C4.5: 1993)
- Breimanetal (CaRT: 1984) from statistics

Identifying ‘bestAttribute’

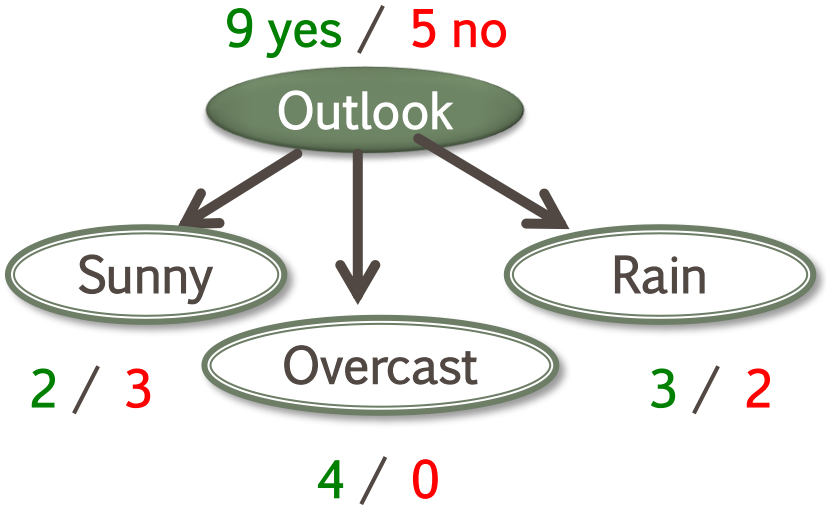
- There are many possible ways to select the best attribute for a given set.
- We will discuss one possible way which is based on information theory and generalizes well to non binary variables

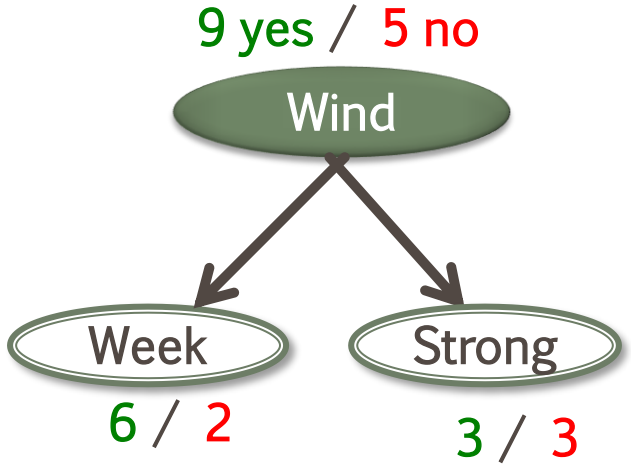
Different methods to select best feature

Random	Least-Values	Most-Values	Max-Gain
Choose any feature at random	Choose the feature with the fewest possible values	Choose the feature with the most possible values	Choose the feature with the largest expected *information gain*

i.e., the feature that is expected to result in the shortest subtree

Which attribute to split on?





- Want to measure "purity" of the split
  - more certain about Yes/No after the split
    - pure set (4 yes / 0 no) => completely certain (100%)
    - impure (3 yes / 3 no) => completely uncertain (50%)

Entropy

- Entropy: $H(S) = - p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$ bits
  - S … subset of training examples
  - $p_{(+)}$ / $p_{(-)}$ … % of positive / negative examples in S

• The Entropy is 1 when the collection contains *an equal number of positive and negative examples.*

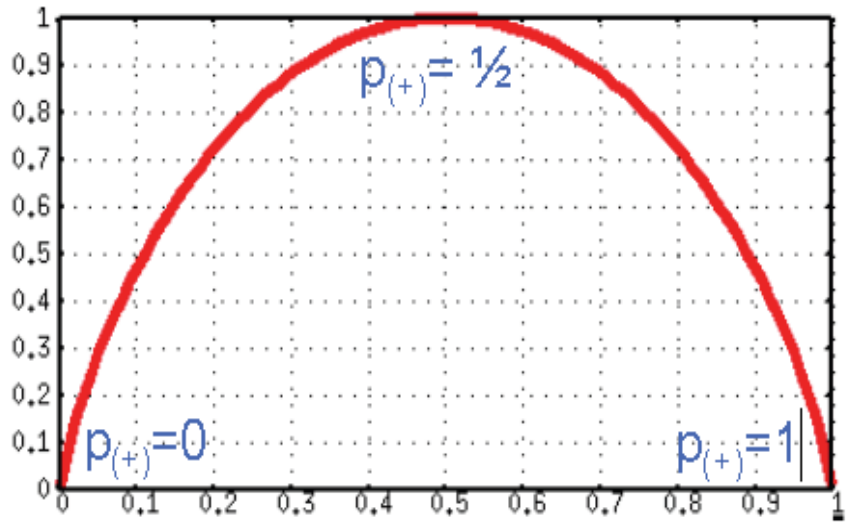
• The Entropy is 0 if all members of S belong to *the same class*

- impure (3 yes / 3 no):

$$H(S) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1 \text{ bits}$$

- pure set (4 yes / 0 no):

$$H(S) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0 \text{ bits}$$



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.

Will John play tennis today?

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

calculate current entropy

- $p_+ = \frac{9}{14} \quad p_- = \frac{5}{14}$
- $Entropy(Play) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$

$$= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}$$

H(S) ≈ 0.94

INFORMATION GAIN

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

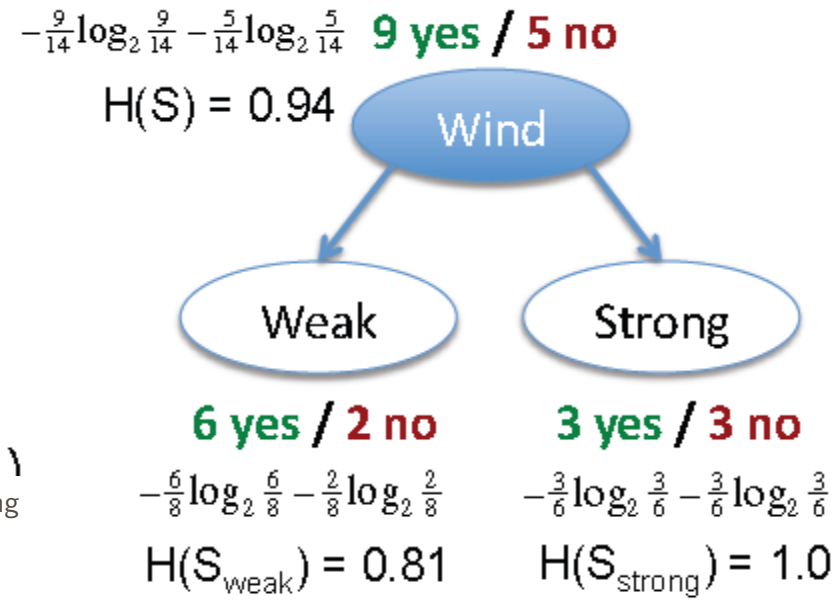
- Want many items in pure sets
- Expected drop in entropy after split:

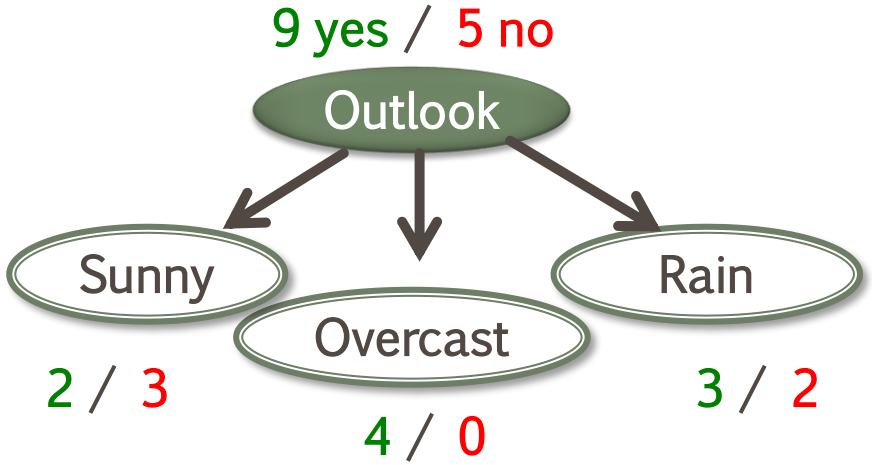
$$Gain(S,A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} H(S_V)$$

V … possible values of A
S … set of examples {X}
S_V … subset where $X_A = V$

- Mutual Information
  - between attribute A and class labels of S

Gain (S, Wind)
= H(S) – $^8/_{14}$ H(S_{weak}) – $^6/_{14}$ H(S_{strong})
= 0.94 – $^8/_{14}$ * 0.81 – $^6/_{14}$ * 1.0
= 0.049





$$H(S, outlook) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

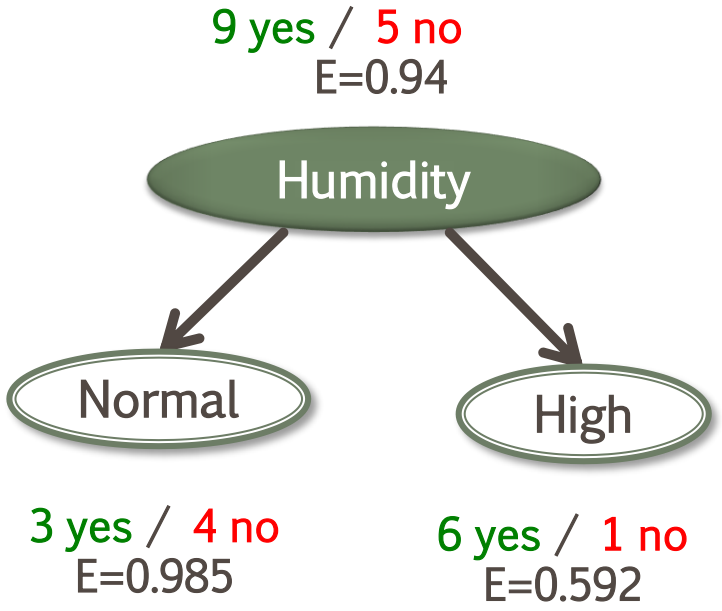
$$H(S_{sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$$

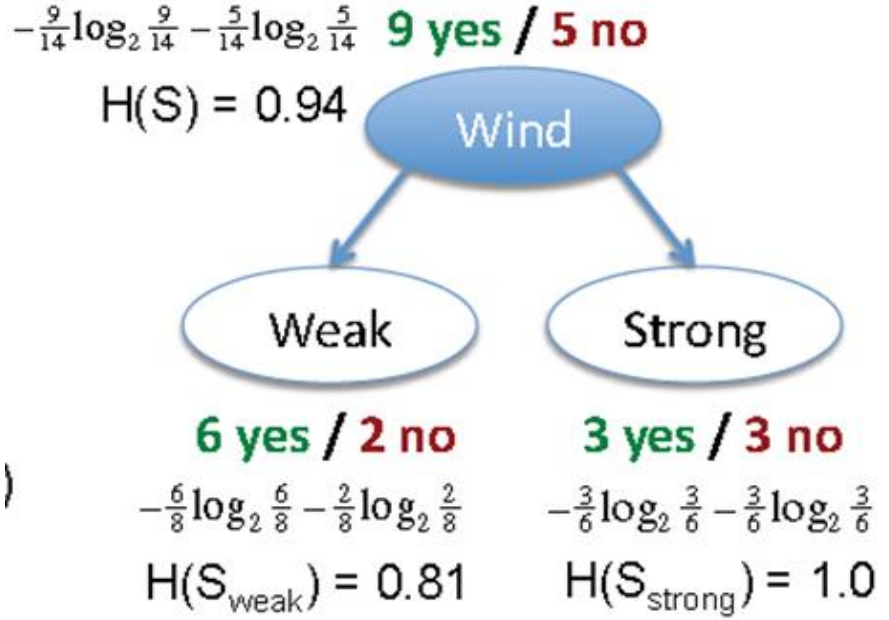
$$H(S_{overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$H(S_{rain}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14}\times 0.97 - \frac{4}{14}\times 0 - \frac{5}{14}\times 0.97 = 0.247$$

Which attribute is the best classifier? Example





Gain (S,Humidity)=0.94-(7/14)*0.985-(7/14)*0.592
=0.151

Gain (S,Wind)=0.94-(8/14)*0.81-(6/14)*1
=0.048

- ***Humidity provides greater information gain than Wind, relative to the target classification.***

E stands for entropy and

S ***the original collection of examples. Given an initial collection S of 9*** positive and 5 negative examples.

ILLUSTRATIVE EXAMPLE

Predict if John will play tennis

$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Humidity) = 0.151$$

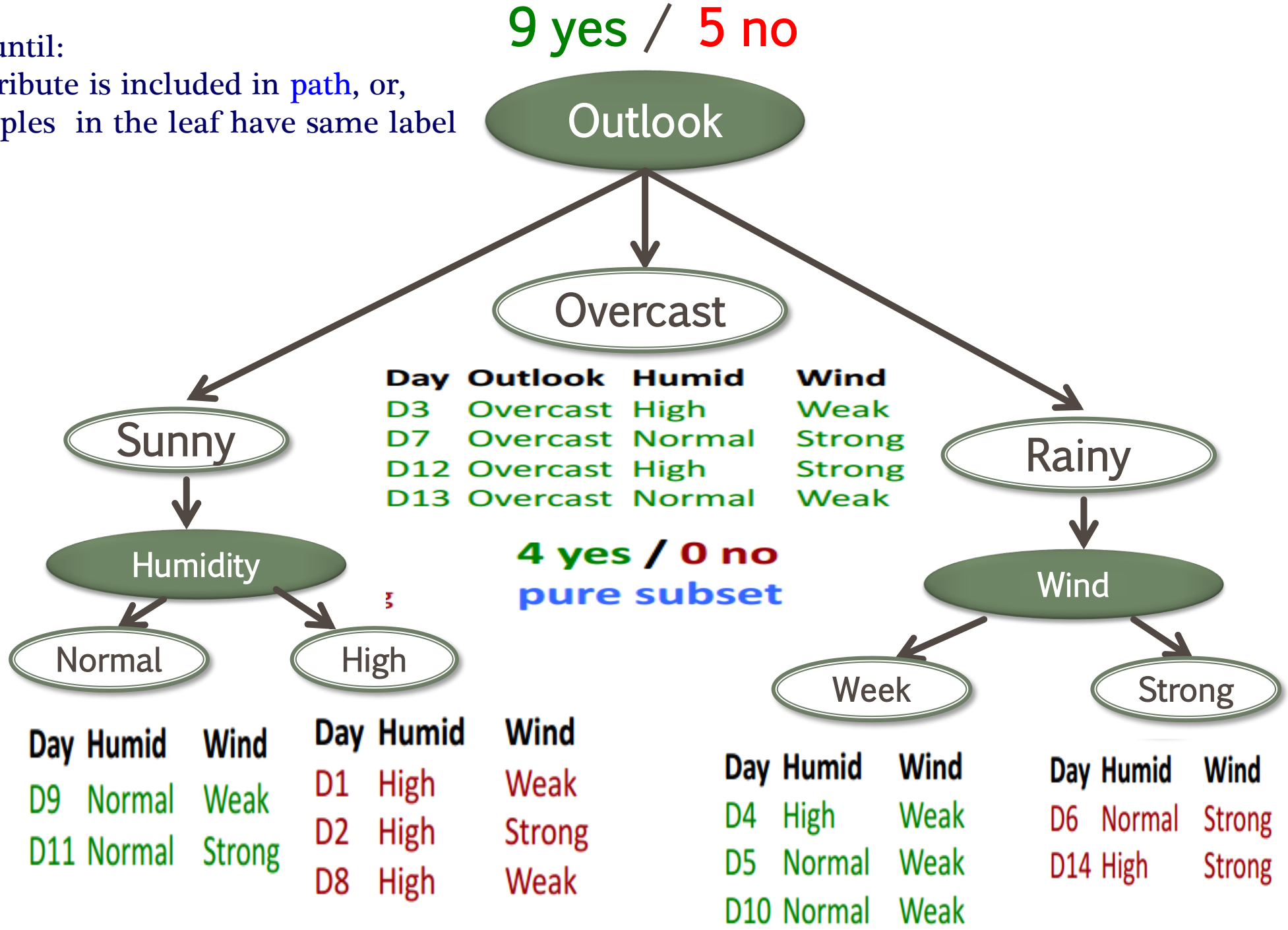
$$Gain(S, Wind) = 0.048$$

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Continue until:

- Every attribute is included in path, or,
- All examples in the leaf have same label

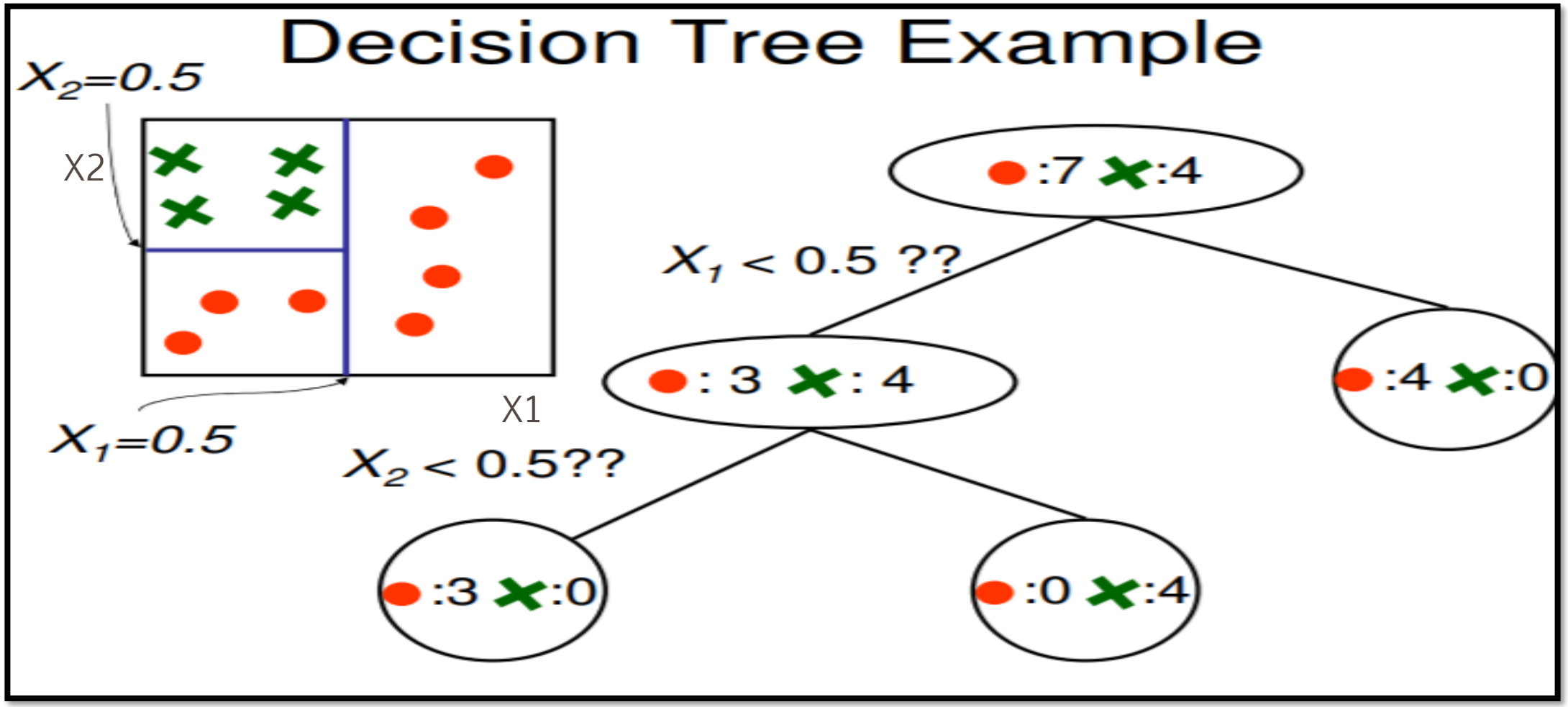


Handling Continuous Features

- Real-valued attributes can, in advance, be discretized into ranges, such as big, medium, small
- Alternatively, one can develop splitting nodes based on thresholds of the form A<c that partition the data into examples
- The information gain for these splits is calculated in the same way and compared to the information gain of discrete splits.

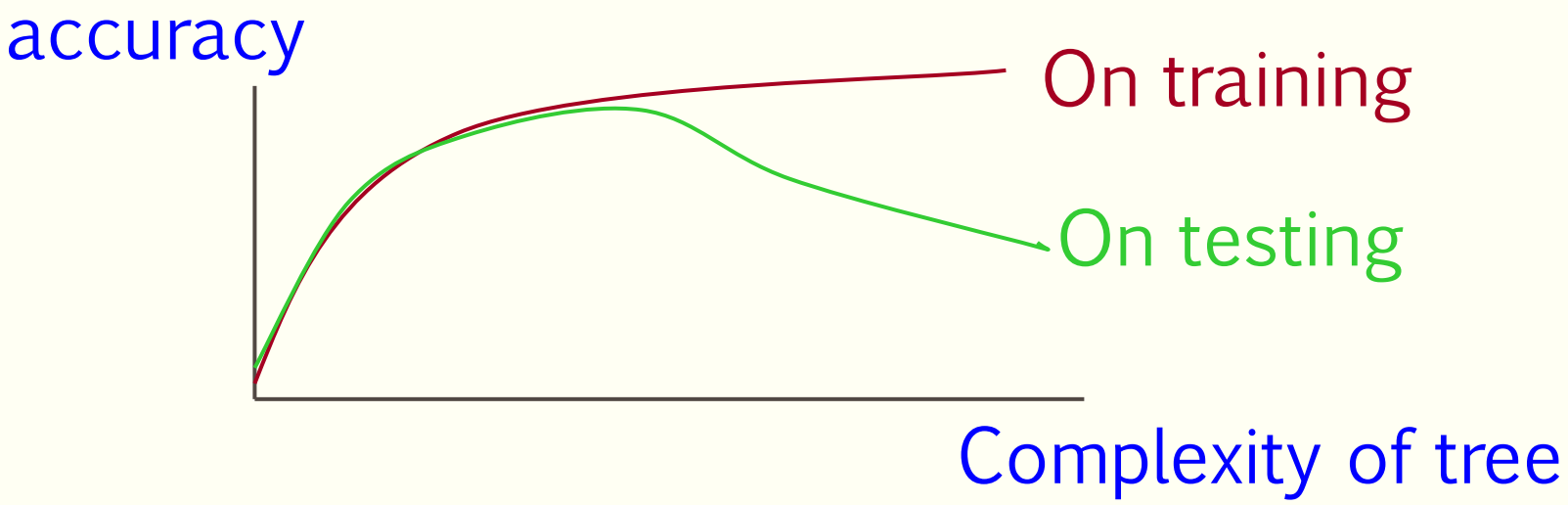
How to find the split with the highest gain?
For each continuous feature A:
    Sort examples according to the value of A
    For each ordered pair (x,y) with different labels
        Check the mid-point as a possible threshold, i.e.
        $S_{a < x}$ $S_{a >= y}$

CONTINUOUS VALUE



Overfitting the Data

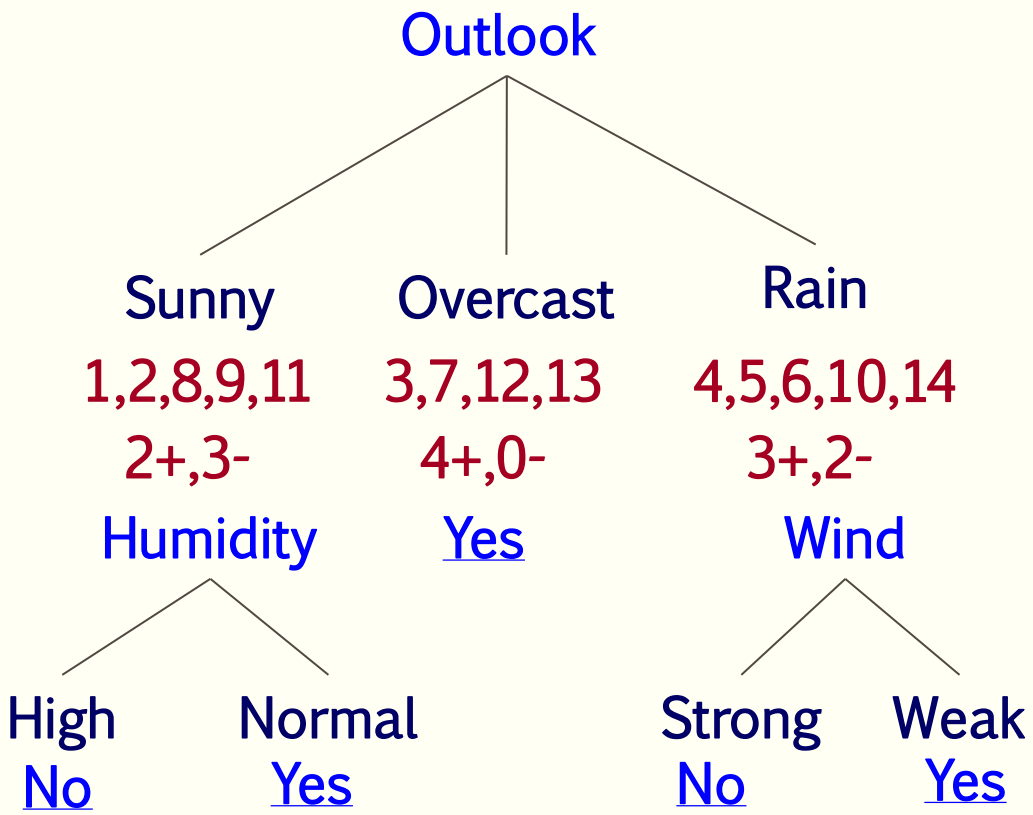
- Learning a tree that classifies the training data perfectly may not lead to the tree with the best generalization performance.
  - There may be noise in the training data the tree is fitting
  - The algorithm might be making decisions based on very little data
- A hypothesis h is said to overfit the training data if there is another hypothesis h', such that h has a smaller error than h' on the **training data** but h has larger error on the **test data** than h'.



Overfitting - Example

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

- Outlook = Sunny,
- Temp = Hot
- Humidity = Normal
- Wind = Strong
- label: NO
- this example doesn't exist in the tree



How to Deal with Overfitting?

- Stop growing the tree when the data split is not statistically significant
- Grow the full tree, then prune
  - Do we really needs all the “small” leaves with perfect coverage?

Pruning a decision tree

- Pruning refers to a technique to remove the parts of the decision tree to prevent growing to its full depth

- Prune = remove leaves and assign majority label of the parent to all items

- Prune the children of node s if:
  - all children are leaves, and
  - the accuracy on the validation set does not decrease if we assign the most frequent class label to all items at s.

Decision Tree Pre-Pruning

- Stop the algorithm before it becomes a fully-grown tree
- Typical stopping conditions for a node
  - Stop if all instances belong to the same class
  - Stop if all the feature values are the same
- More restrictive conditions
  - Stop if the number of instances is less than some use-specified threshold
  - Stop if the class distribution of instances are independent of the available features
    - Stop if expanding the current node does not improve impurity.

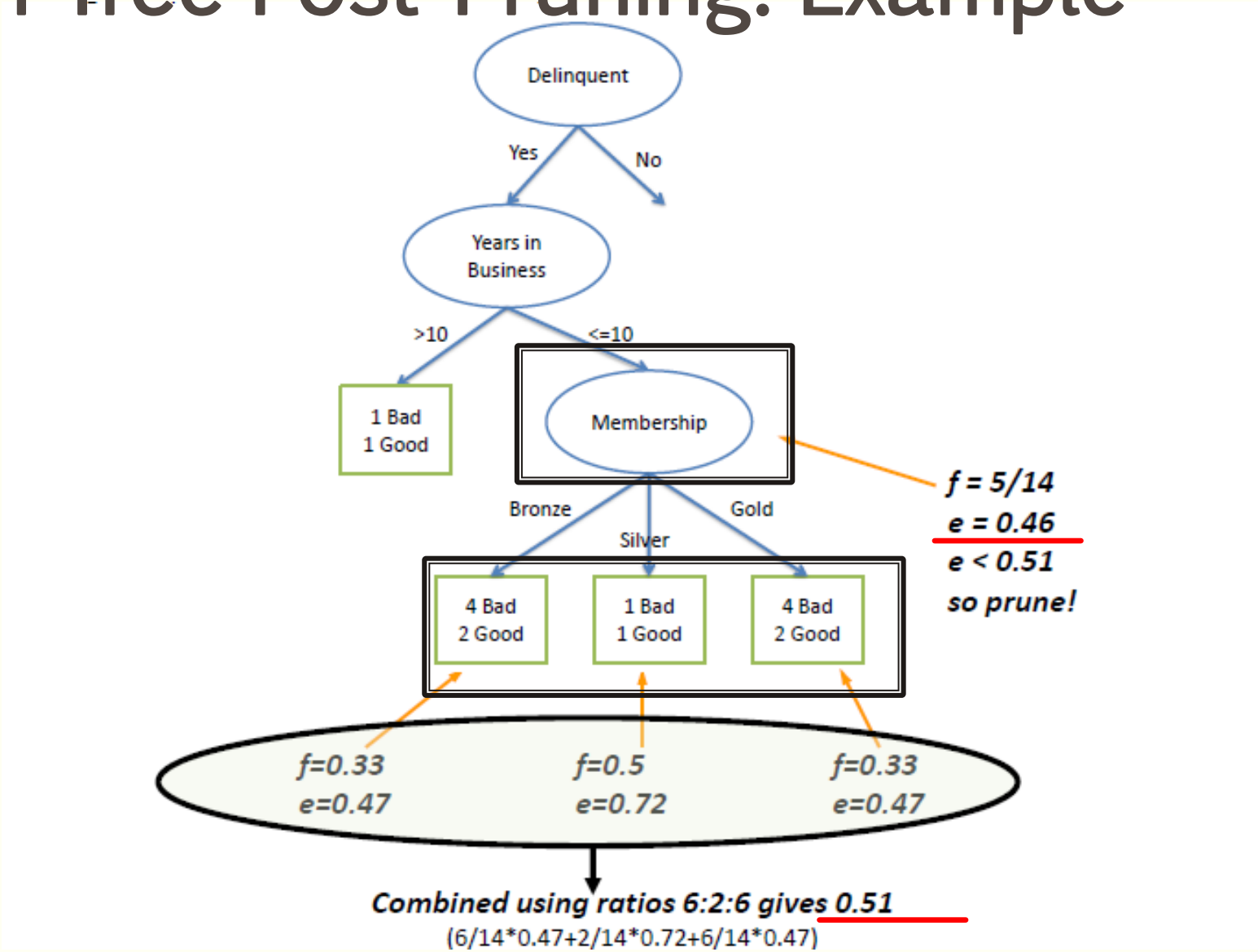
Decision Tree Post-Pruning

- Grow decision tree to its entirety
- Trim the nodes of the decision tree in a bottom-up fashion
- If generalization error improves after trimming, replace sub-tree by a leaf node
  - Class label of leaf node is determined from majority class of instances in the sub-tree

Decision Tree Post-Pruning

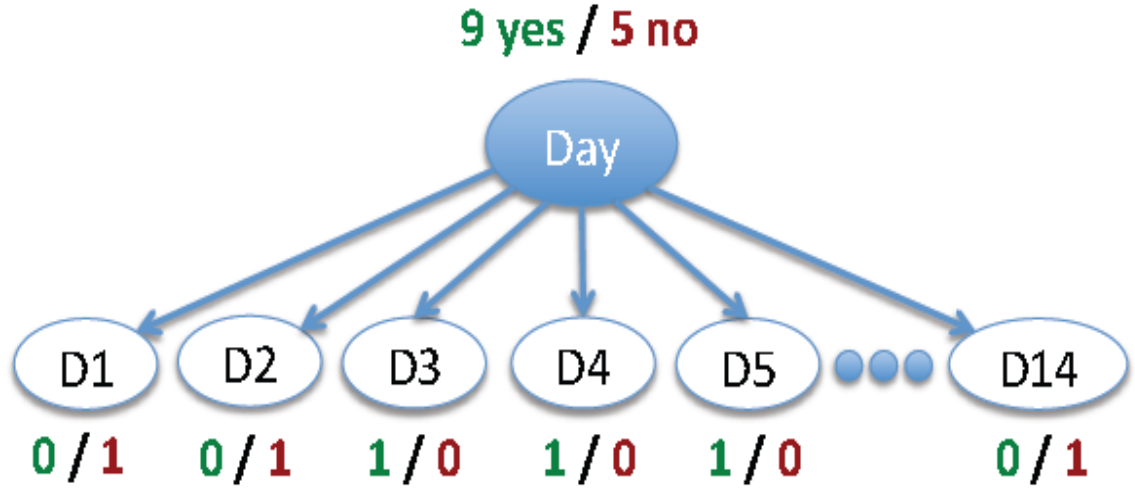
- Reduced Error Pruning
  - Split data into training and validation set
  - Remove one node at a time and evaluate the performance on the validation data
  - Remove the one that decreases the error
  - Usually produces the smallest version of a tree
  - But always requires a validation set

Decision Tree Post-Pruning: Example



Problems with Information Gain

- Biased towards attributes with many values
- Won't work for new data: D15 Rain High Weak
- Use GainRatio:



all subsets perfectly pure => optimal split

$$SplitEntropy(S,A) = -\sum_{V \in Values(A)} \frac{|S_V|}{|S|} \log \frac{|S_V|}{|S|}$$

A … candidate attribute
V … possible values of A
S … set of examples {X}
S_V … subset where $X_A = V$

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitEntropy(S,A)}$$

penalizes attributes with many values

Problems with information gain

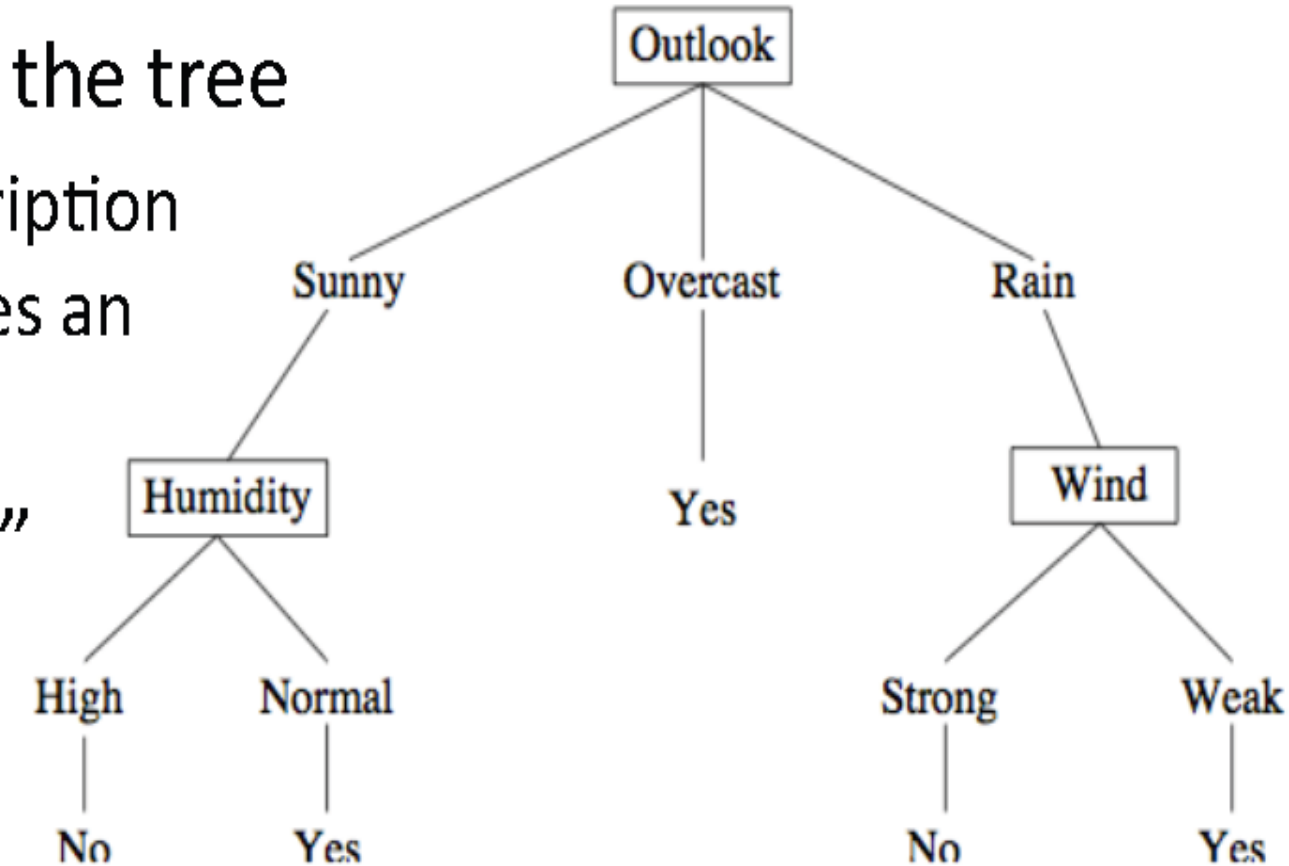
- To addresses the problem of bias in decision tree models caused by attributes with many possible values, Gain Ratio is used.

1. Calculate the Information Gain for each attribute
2. Calculate the Split Information for each attribute
3. Calculate the Gain Ratio
    1. The Gain Ratio is the ratio of Information Gain to Split Information.
    2. Gain Ratio=Information Gain/Split Entropy
    3. Choose the attribute with the highest Gain Ratio

The goal of this calculation is to penalize attributes that create subsets with many values because a high number of subsets may lead to overfitting.

Trees are interpretable

- Read rules off the tree
  - concise description of what makes an item positive
- No "black box"
  - important for users



Rule:
(Outlook = Overcast) ∨
(Outlook = Rain ∧ Wind = Weak) ∨
(Outlook = Sunny ∧ Humidity = Normal)