# Machine Learning, Spring 2019

## Decision Trees

Reading Assignment: Chapter 6 & 7

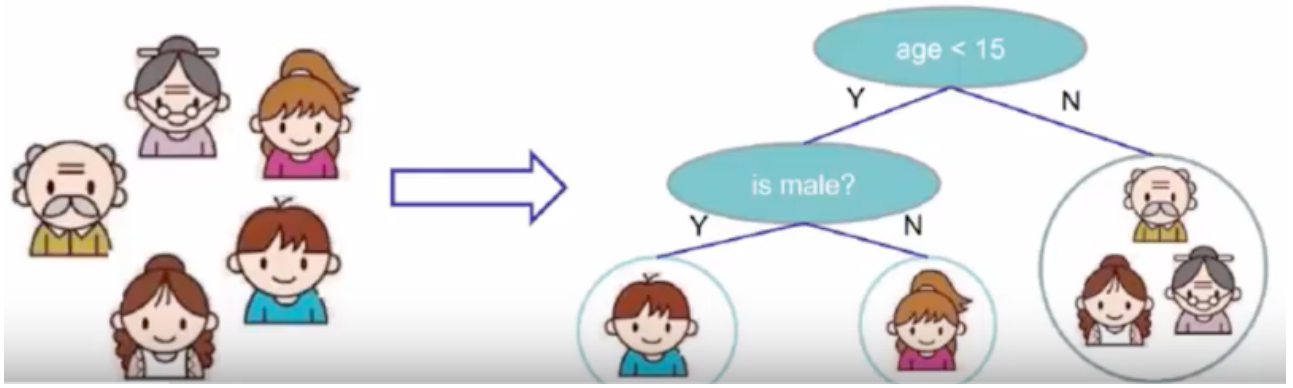
Python tutorial: http://learnpython.org/

TensorFlow tutorial: https://www.tensorflow.org/tutorials/

PyTorch tutorial: https://pytorch.org/tutorials/

# Decision Trees Model

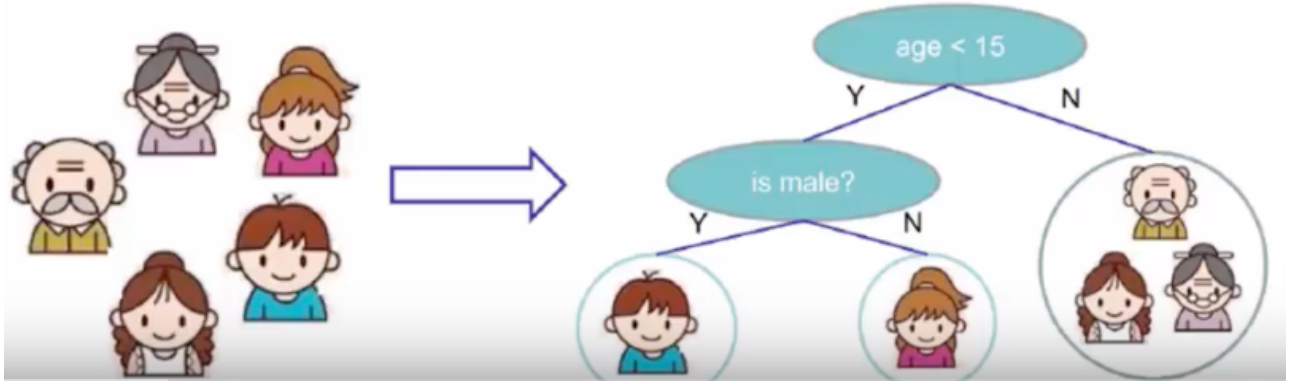
Decision tree: A decision tree decision is an inductive learning task that uses particular facts to make more generalized conclusions. Decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.




Who will play the game?

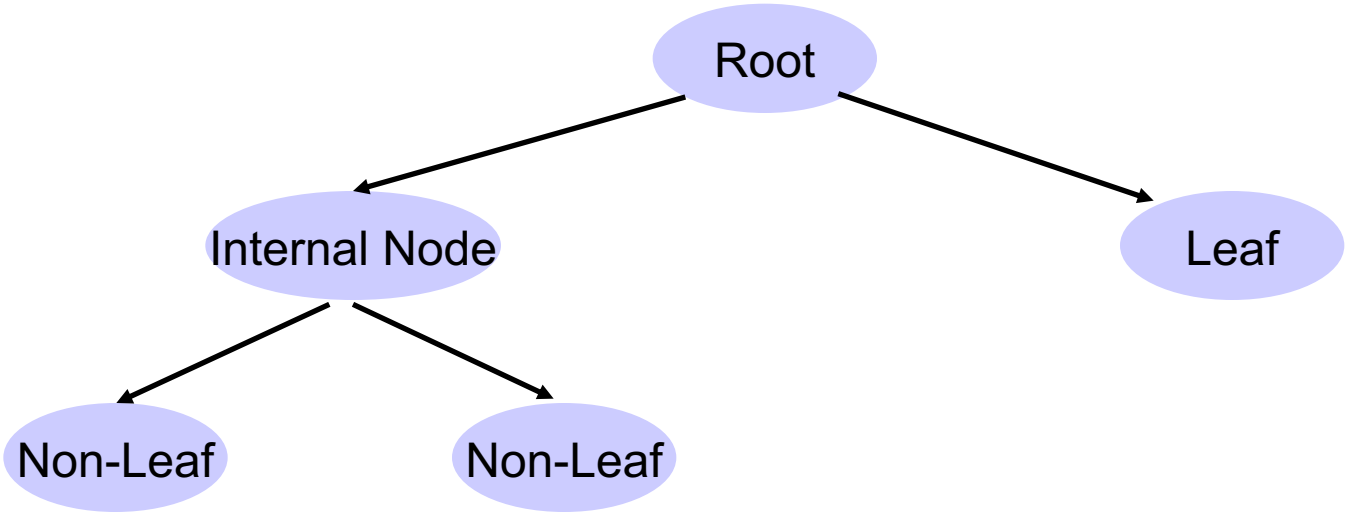
# Decision for Computer Game

- Age > 15?
- Is male?



# Decision Tree

- Root node
- Internal node
- Leaf node



# Tree Training

- Training data
- Construct the tree
- Testing with a trained tree

# Tree Training Procedure

- Choose one attribute for the root node
- Decision made based on the node
- Choose second attribute for the internal node
- Decision made based on the second node
- Iterate the steps above

# Which attribute is best?

- Which criteria should be used to evaluate the quality of a attribute?
- Answer: Entropy

# Entropy

- Entropy(S)= Expected umber of bits needed to encode class (+ or -) of randomly drawn members of S (under the optimal, shortest length-code)

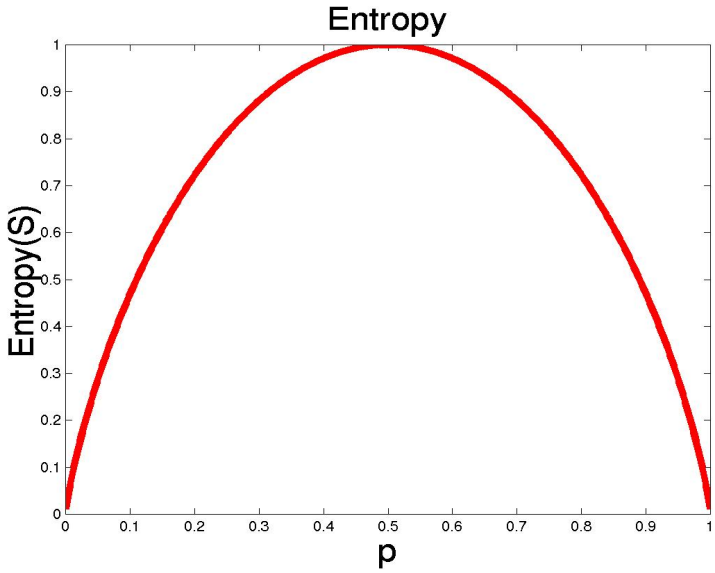
$$H(X)=-\sum pi * logpi, i=1,2, \dots , n$$

- Example: [1,1,1,1,1,1,2,2] and [1,2,3,4,5,6,7,8], which entropy is smaller?

- Binary case: encode (+ or -) of random member of S:

$$-p_+ \log_2 p_+ - p_- \log_2 p_-$$

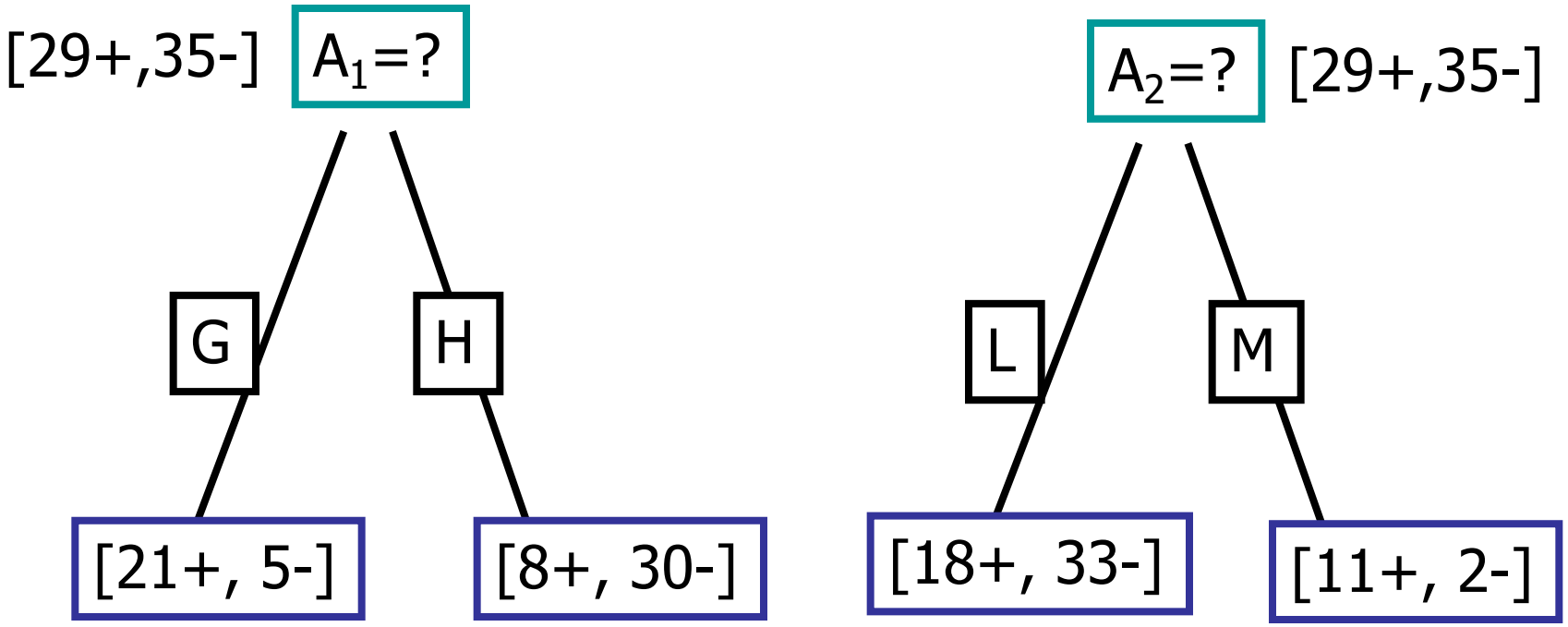
# Entropy



# Entropy

- A statistical property called information gain, measures how well a given attribute separates the training examples
- Information gain uses the notion of entropy, commonly used in information theory
- Information gain = expected reduction of entropy

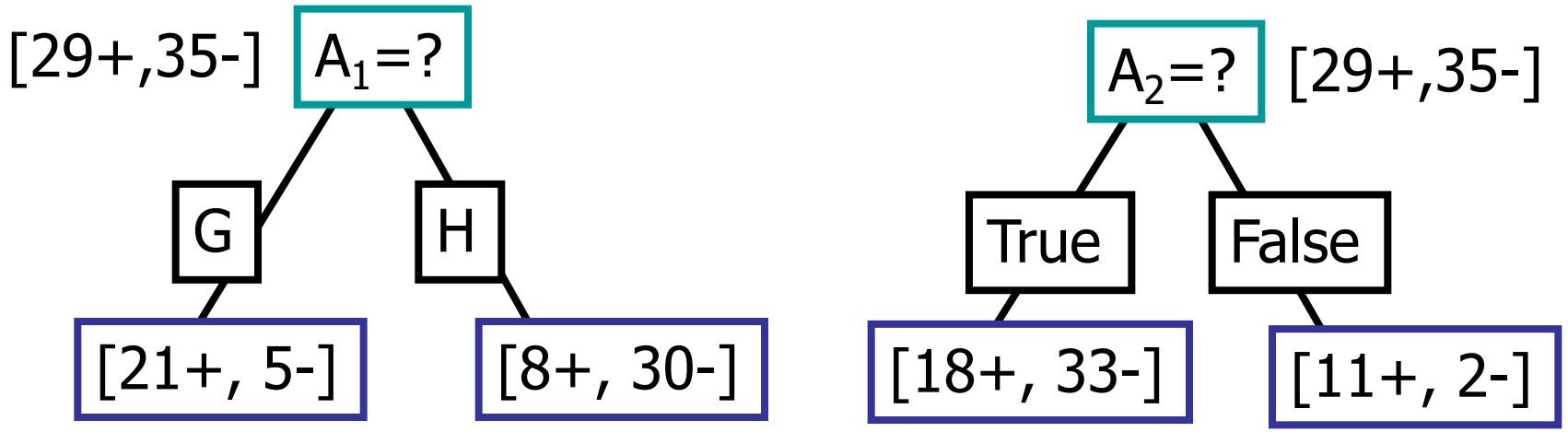
# Example of Entropy Calculation




- S is a sample of training examples
- p+ is the proportion of positive examples
- p- is the proportion of negative examples

# Before Classification

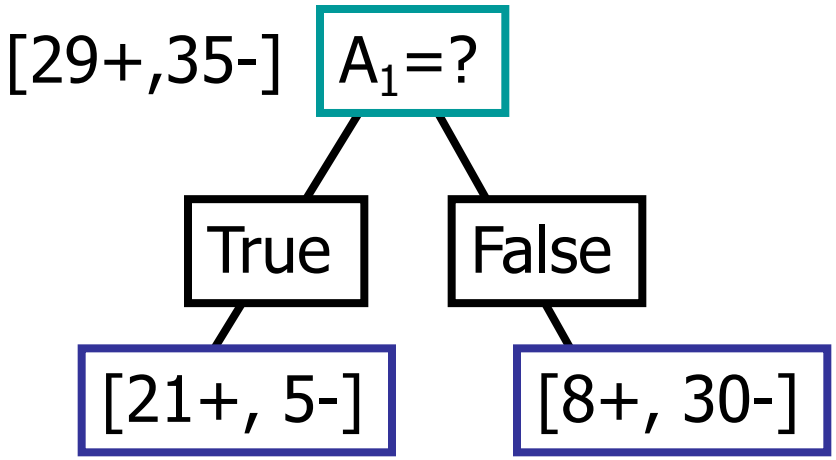
Entropy([29+,35-]) = -29/64 log2 29/64 – 35/64 log2 35/64
= 0.99

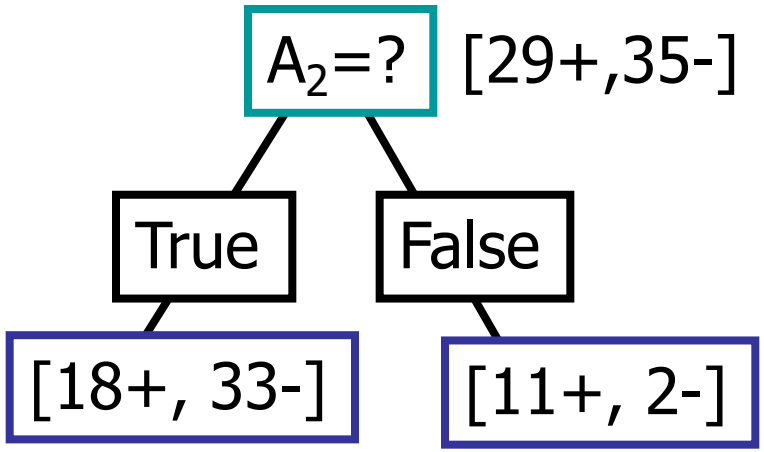


# Information Gain

Entropy([21+,5-]) = 0.71
Entropy([8+,30-]) = 0.74
Gain(S,$A_1$)=Entropy(S)
-26/64*Entropy([21+,5-])
-38/64*Entropy([8+,30-])
=0.27

Entropy([18+,33-]) = 0.94
Entropy([11+,2-]) = 0.62
Gain(S,A2)=Entropy(S)
-51/64*Entropy([18+,33-])
-13/64*Entropy([11+,2-])
=0.12

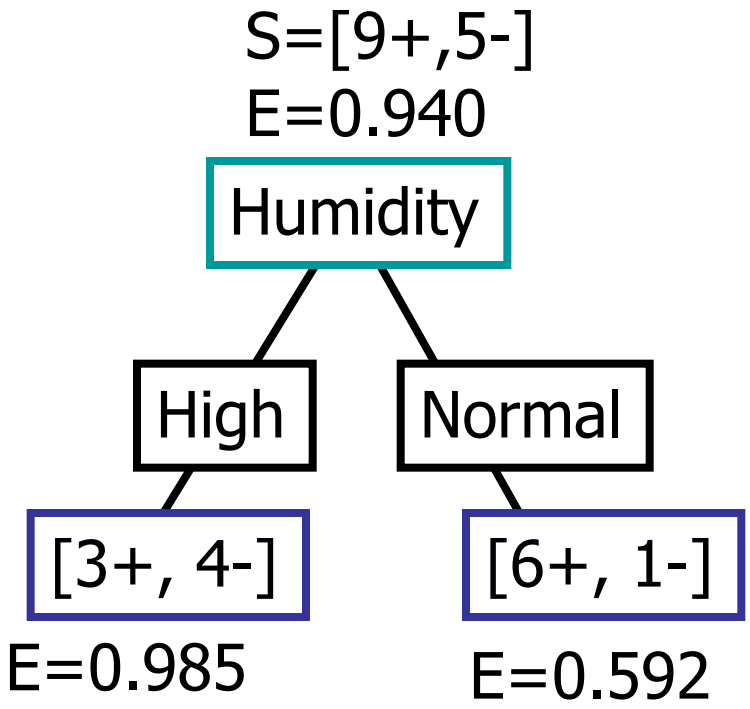


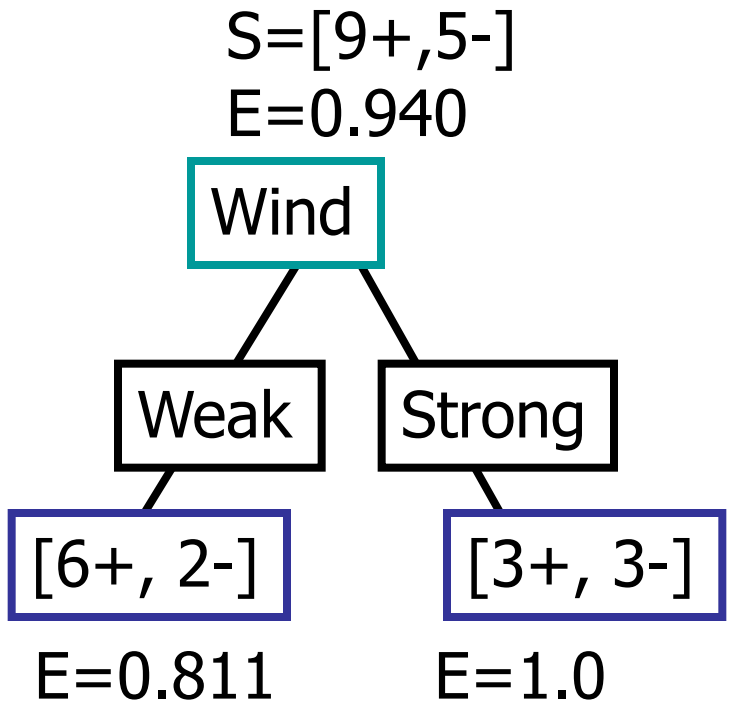



# Training A Tree

| Day | Outlook | Temp. | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| 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 | Weak | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cold | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Strong | 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



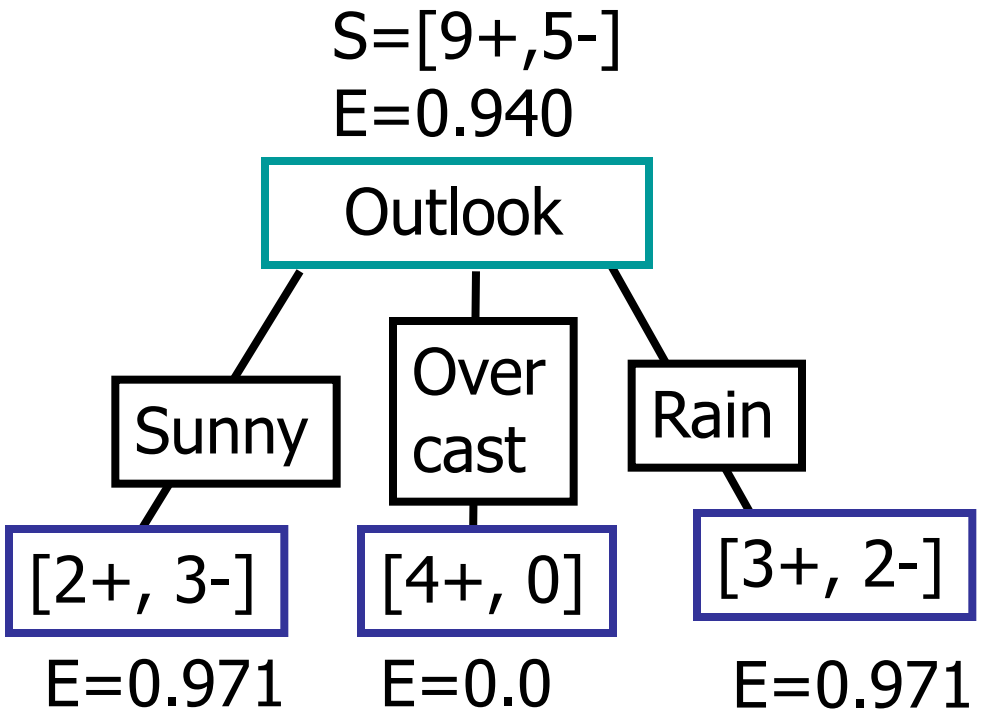




Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151

Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

Humidity provides greater info. gain than Wind, w.r.t target classification.

# Selecting the Next Attribute




Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

# Selecting the Next Attribute

The information gain values for the 4 attributes are:

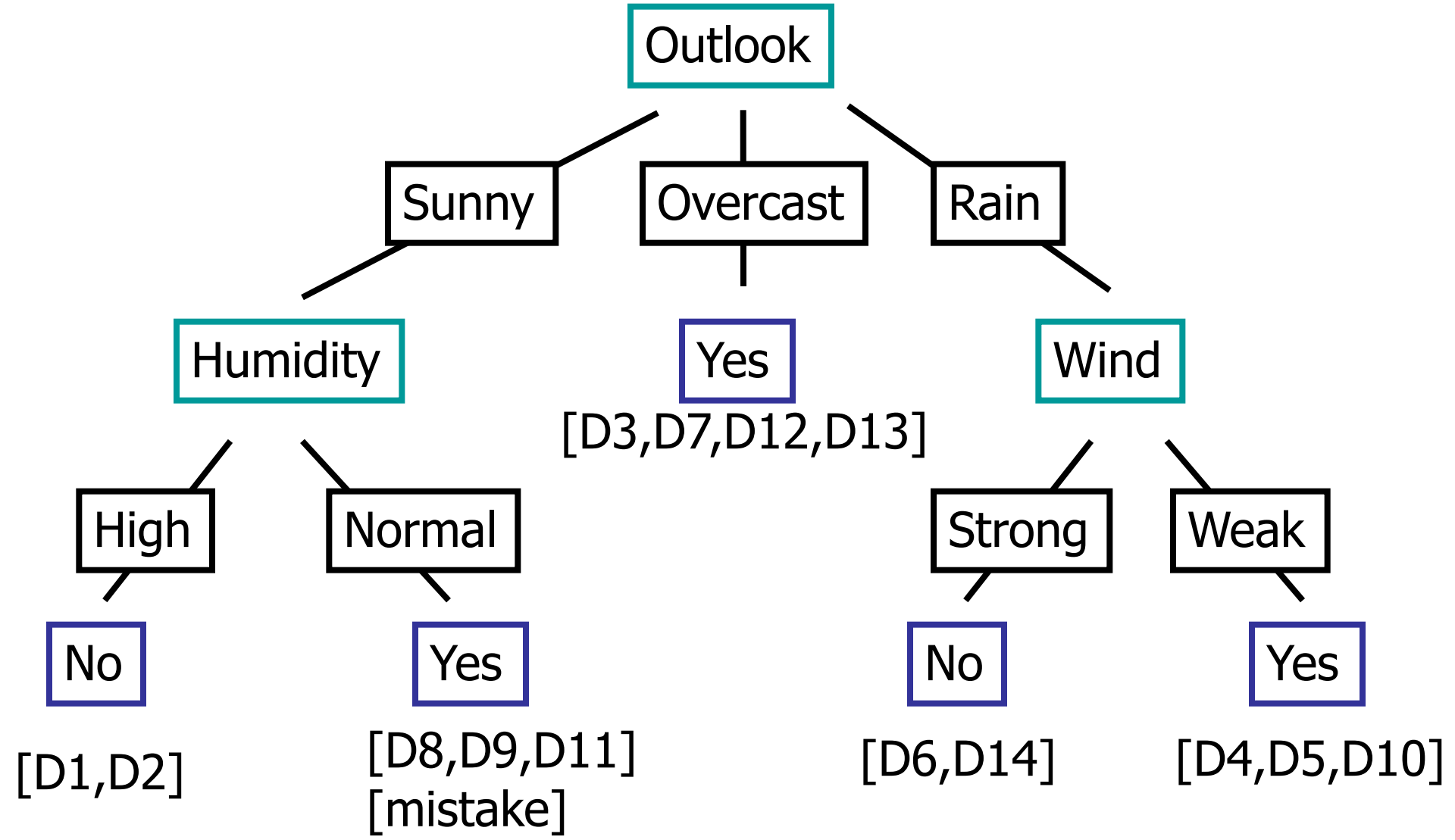
- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

where S denotes the collection of training

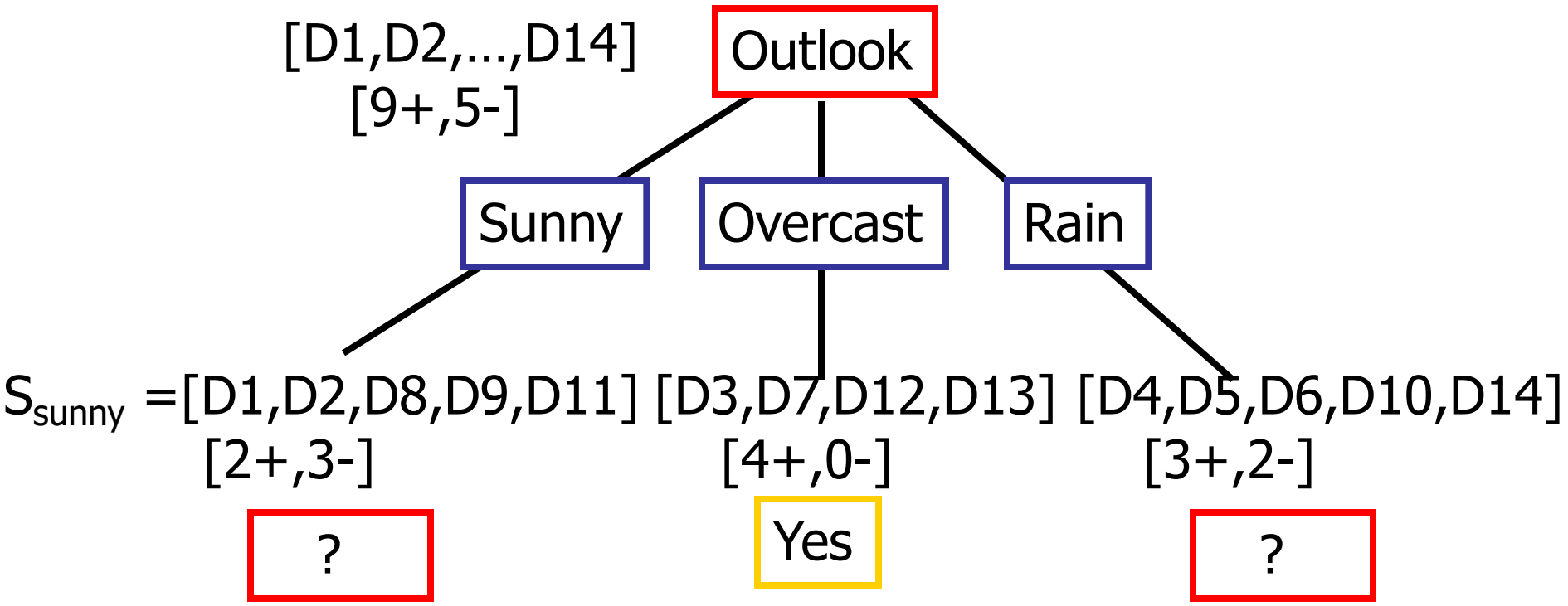
# ID3 Algorithm for Decision Tree

- ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset.

# ID3 Algorithm



# ID3 Algorithm

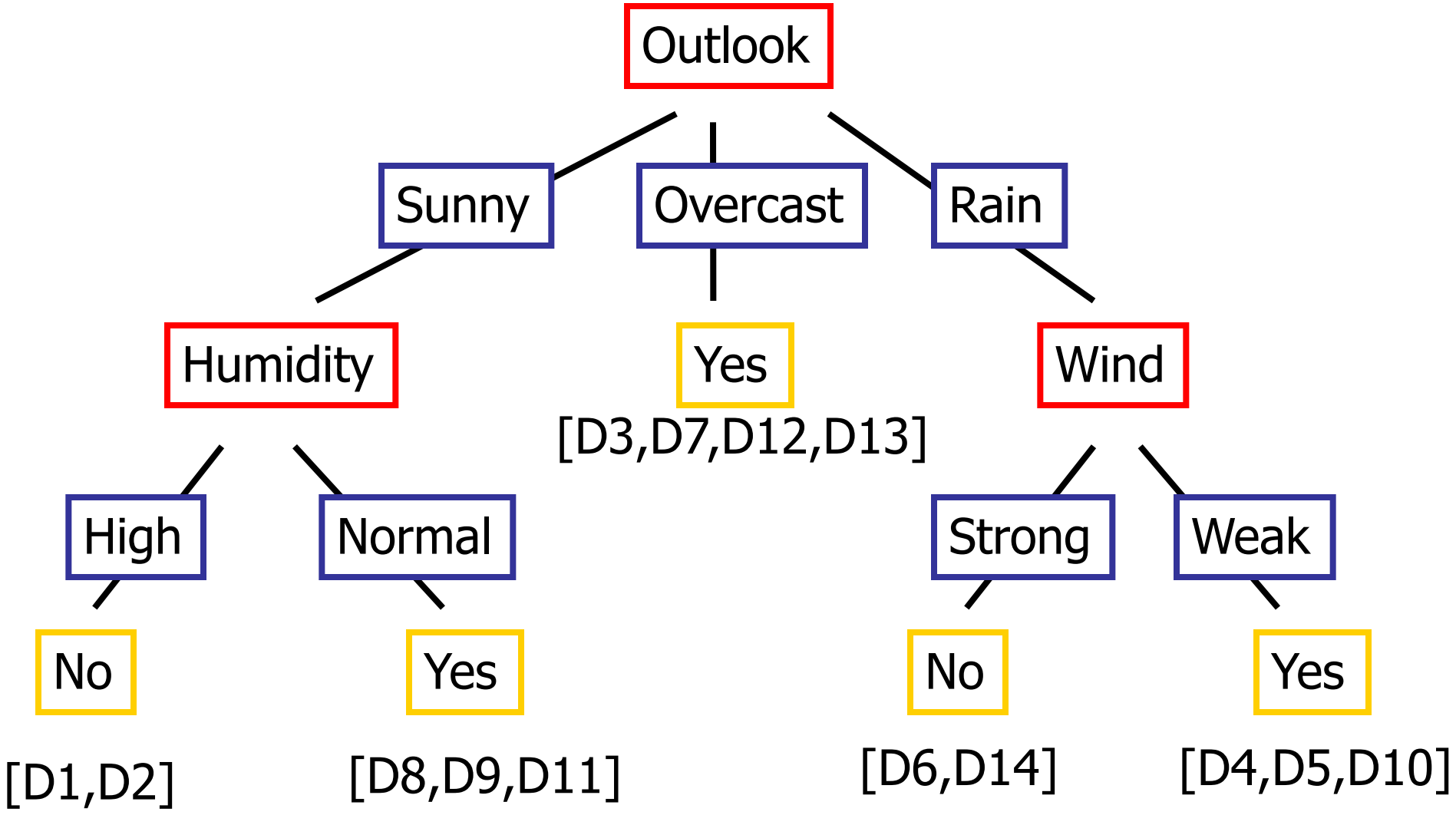



Gain($S_{sunny}$, Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970

Gain($S_{sunny}$, Temp.)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570

Gain($S_{sunny}$, Wind)=0.970= -(2/5)1.0 – 3/5(0.918) = 0.019

# ID3 Algorithm



# ID3 Algorithm

- What is the challenging issue for ID3 algorithm?
- Sparse distributed data (each class only contains few points)

# C4.5 Algorithm

- C4.5 is an extension of Quinlan's earlier ID3 algorithm.

- The splitting criterion is the normalized information gain (difference in entropy).

# GINI Coefficient

- The Gini index or Gini coefficient is a statistical measure of distribution

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

# Overfitting of Decision Tree

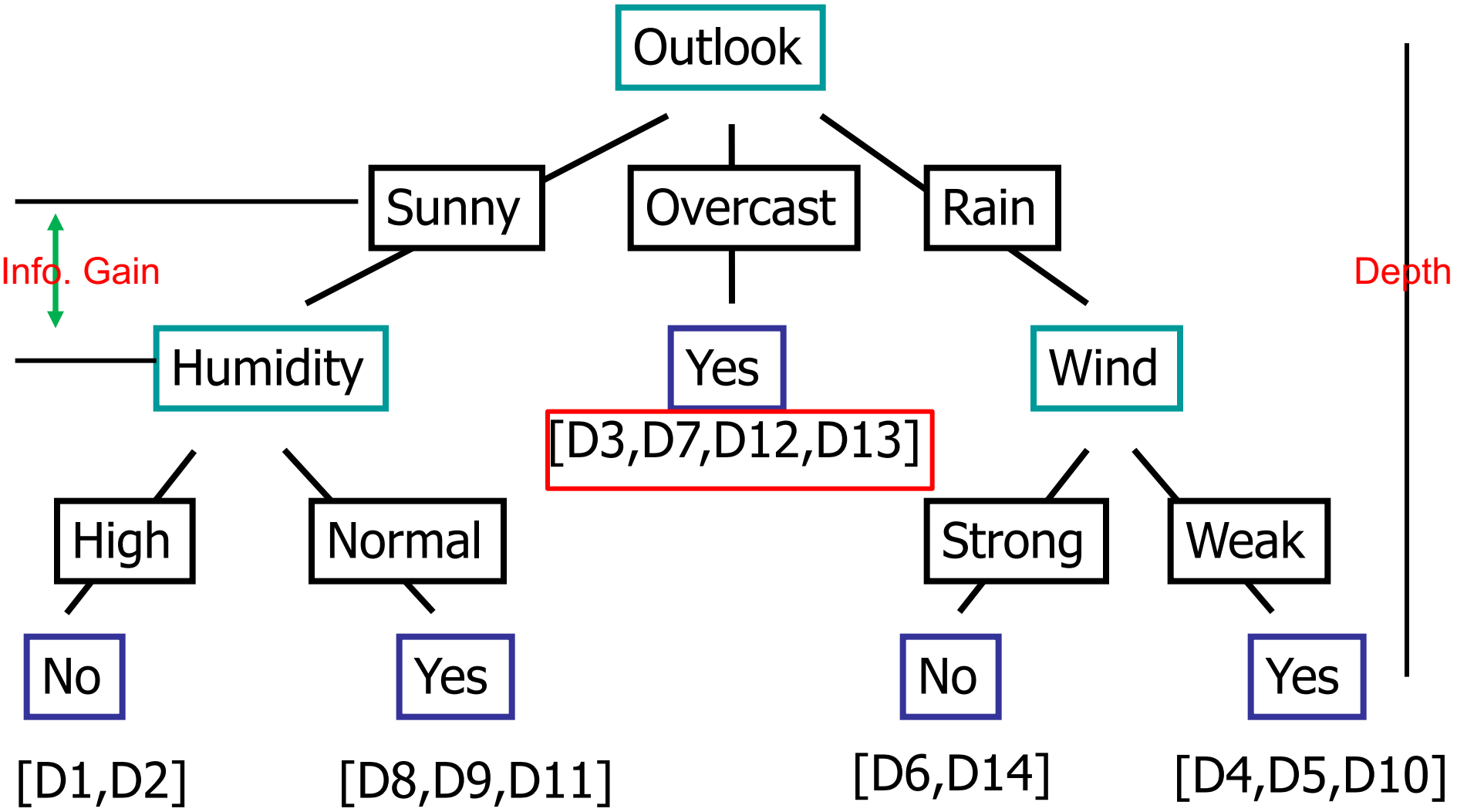
# Avoid Overfitting

- Pre-prune: stop growing when split not statistically significant
- Post-prune: stop growing when split not statistically significant

# Pre-prune

- Limits the depth of the tree
- Limits the number of samples of a internal node
- Limits the information gain

# Pre-prune


Outlook
Sunny
Overcast
Rain
Info. Gain
Depth
Humidity
Yes
[D3,D7,D12,D13]
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes
[D1,D2]
[D8,D9,D11]
[D6,D14]
[D4,D5,D10]