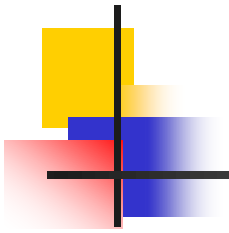


# Large Scale Decision Tree Learning

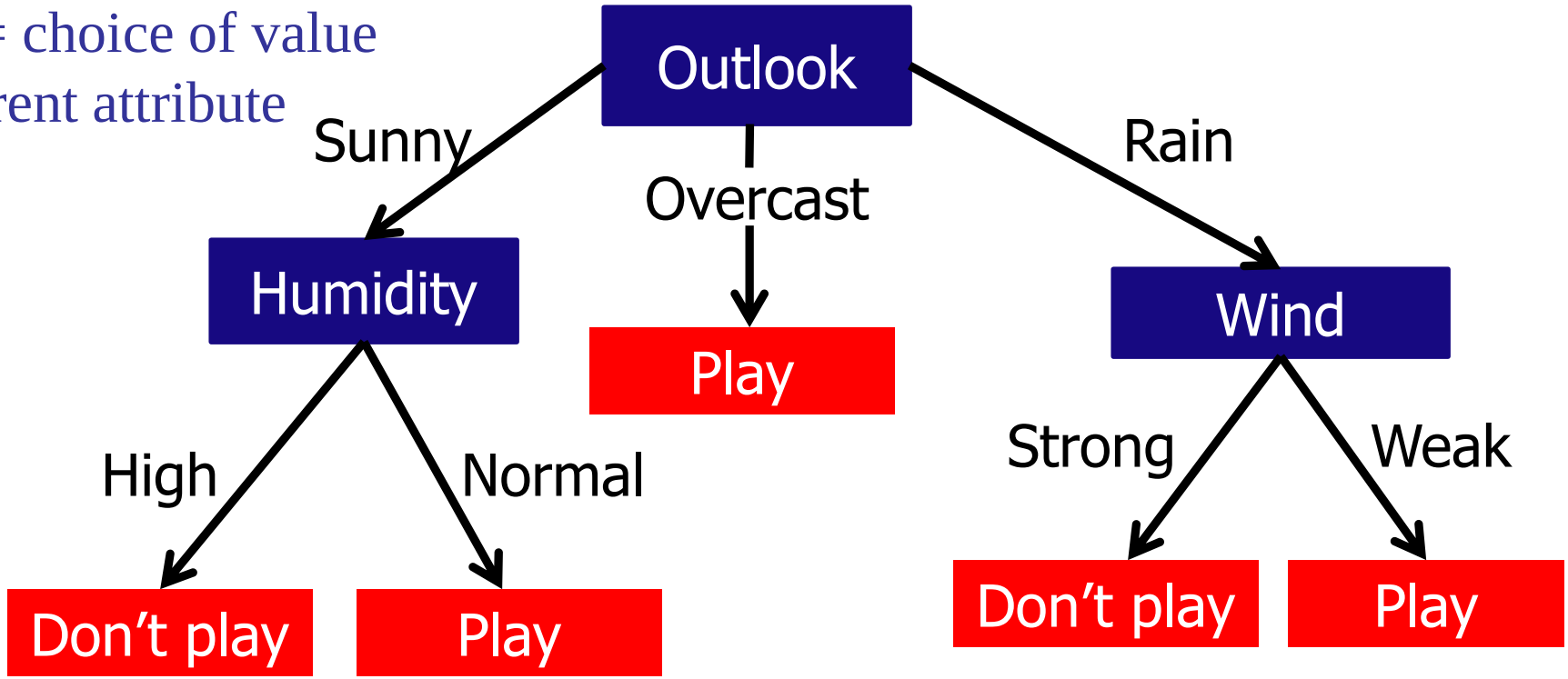
Instructor: Jesse Davis

# Decision Tree Representation

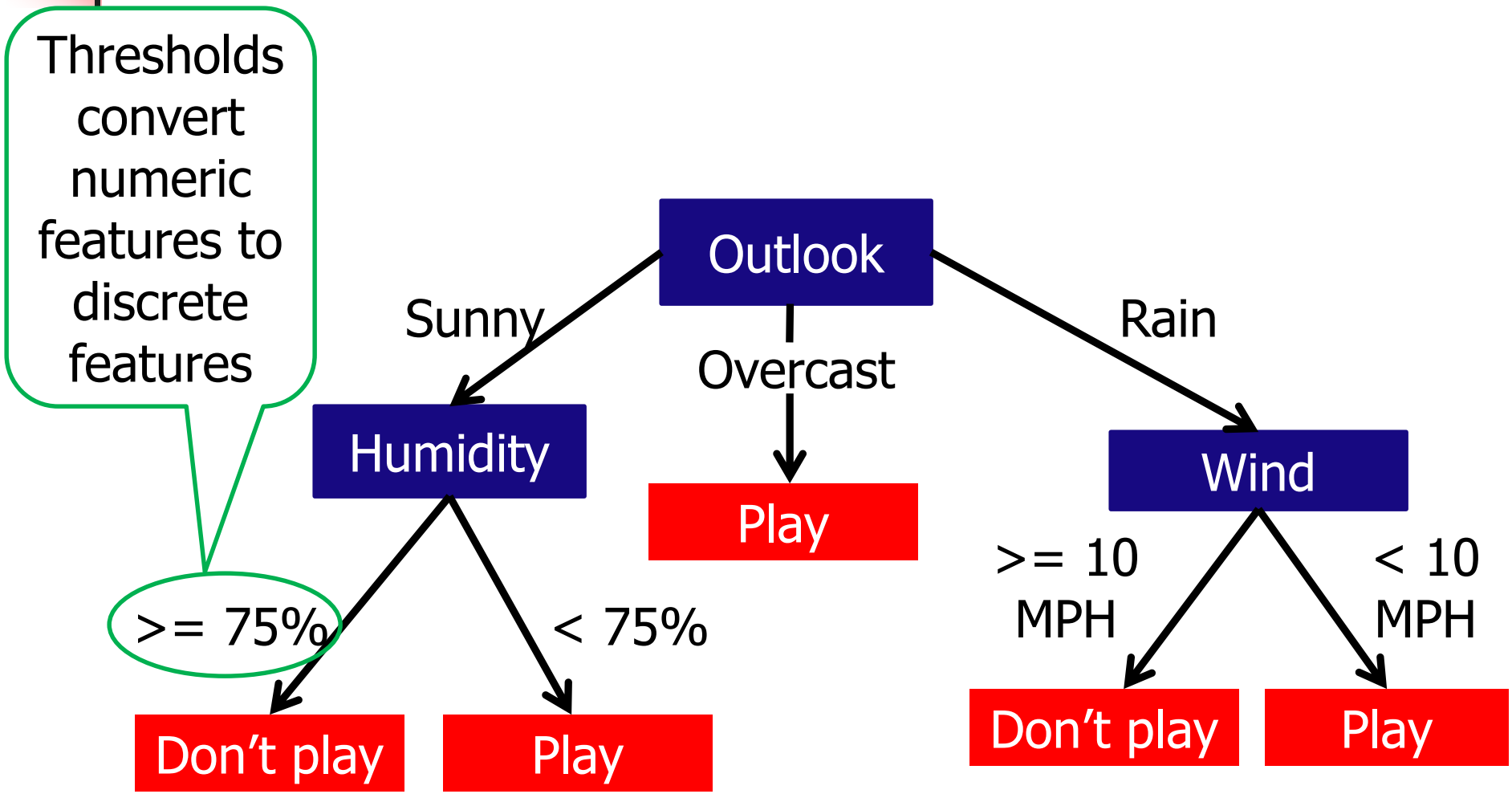
Good day for tennis?

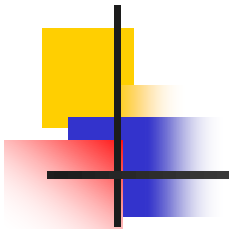
Leaves = classification

Arcs = choice of value for parent attribute



# Numeric Attributes

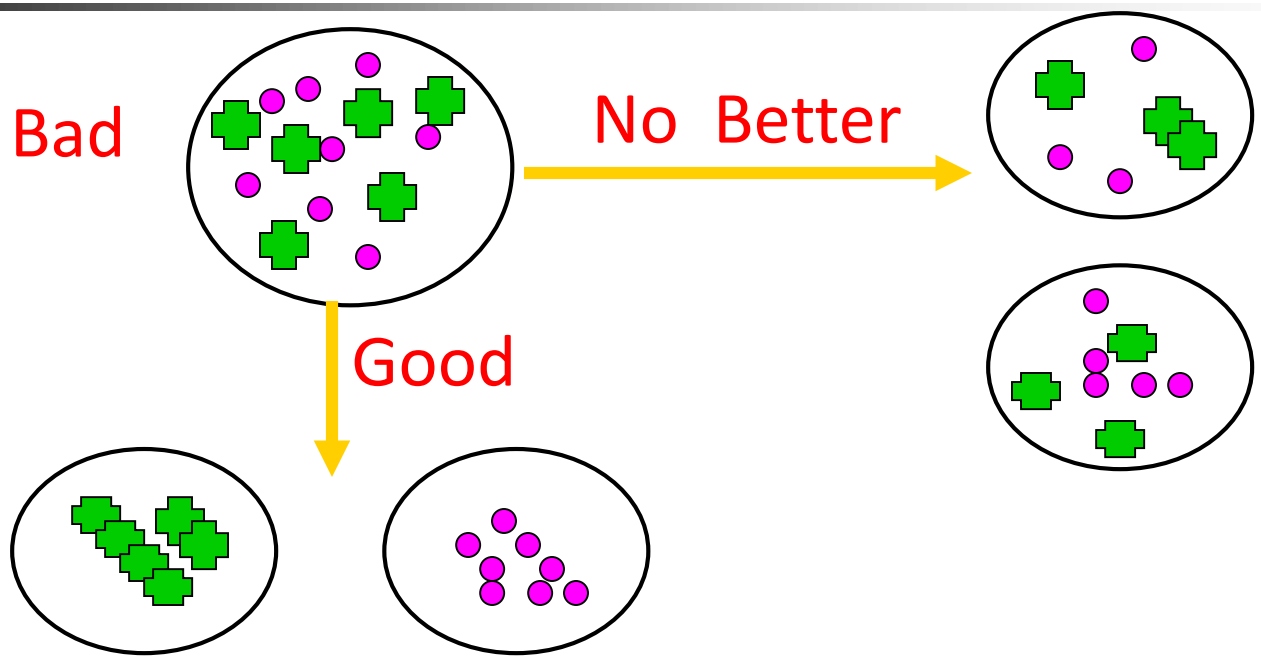




# DT Learning as Search

- Nodes: Decision Trees:
  1) Internal: Attribute-value test
  2) Leaf: Class label
- Operators: Tree Refinement: Sprouting the tree
- Initial node: Smallest tree possible: a single leaf
- Heuristic: Information Gain
- Goal: Best tree possible (???)

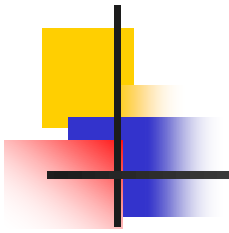
# Splitting the Data




**Intuition:** Disorder is bad and homogeneity is good

$$Gain(S,A) = Entropy(S) - \Sigma(|S_v| / |S|)\ Entropy(S_v)$$

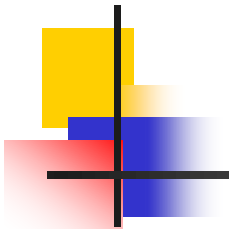
Where $Entropy(S) = -P \log_2(P) - N \log_2(N)$



# Basic Decision Tree Algorithm

```
BuildTree(TraingData)
        Split(TrainingData)

Split(D)
        If (all points in D are of the same class)
                Then Return
        For each attribute A
                Evaluate splits on attribute A
                Use best split to partition D into D1, D2
                Split(D1)
                Split(D2)
```

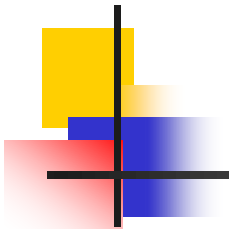


# Example: Good Day For Tennis

- Attributes of instances
  - Outlook = {rainy (r), overcast (o), sunny (s)}
  - Temperature = {cool (c), medium (m), hot (h)}
  - Humidity = {normal (n), high (h)}
  - Wind = {weak (w), strong (s)}
- Class value
  - Play Tennis? = {don't play (n), play (y)}
- Sample instance
  - outlook=***sunny***, temp=***hot***, humidity=***high***, wind=***weak***

# Experience: "Good day for tennis"

| Day | Outlook | Temp | Humid | Wind | PlayTennis? |
|---|---|---|---|---|---|
| d1 | s | h | h | w | n |
| d2 | s | h | h | s | n |
| d3 | o | h | h | w | y |
| d4 | r | m | h | w | y |
| d5 | r | c | n | w | y |
| d6 | r | c | n | s | n |
| d7 | o | c | n | s | y |
| d8 | s | m | h | w | n |
| d9 | s | c | n | w | y |
| d10 | r | m | n | w | y |
| d11 | s | m | n | s | y |
| d12 | o | m | h | s | y |
| d13 | o | h | n | w | y |
| d14 | r | m | h | s | n |



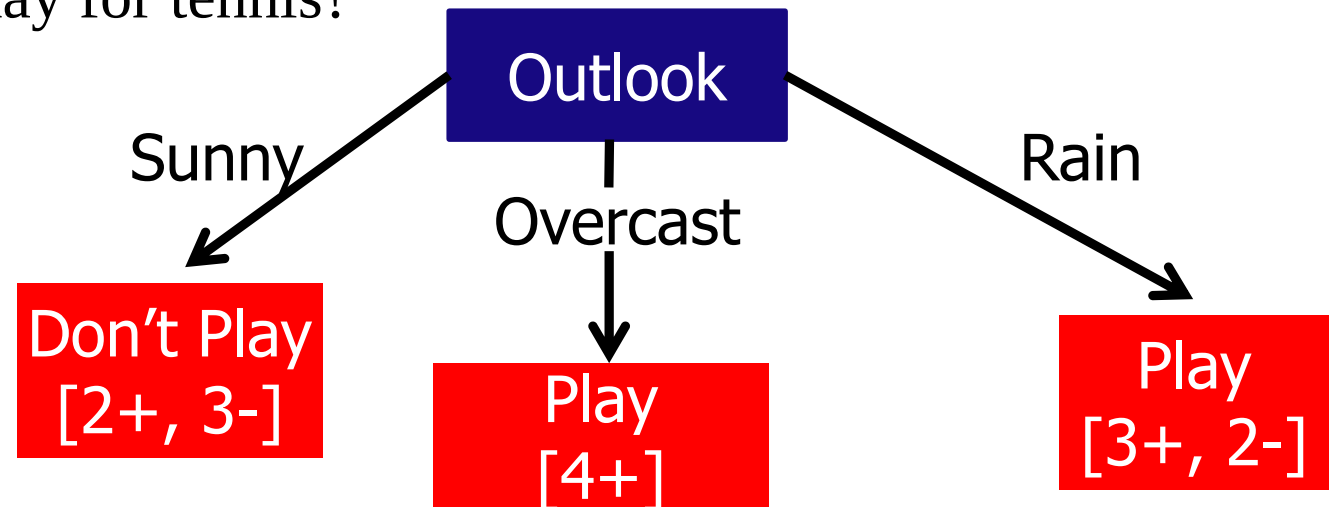
# Initial Tree

Good day for tennis?

Play
[9+, 5-]

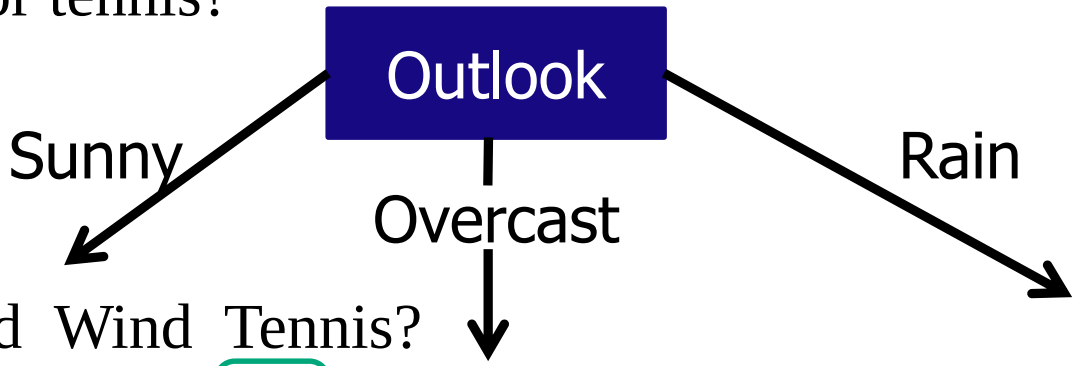
# Resulting Tree

Good day for tennis?



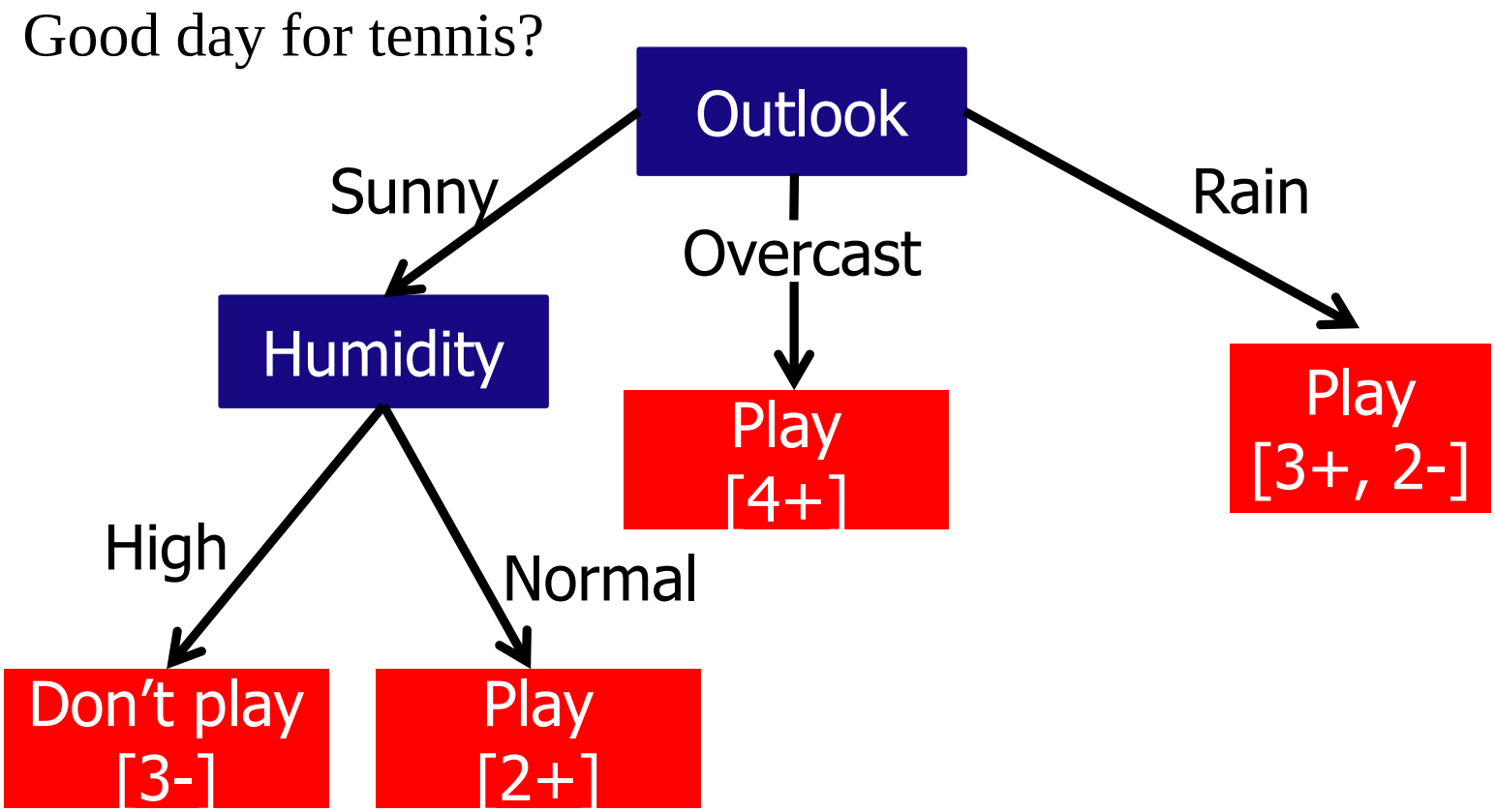
# Recurse

Good day for tennis?

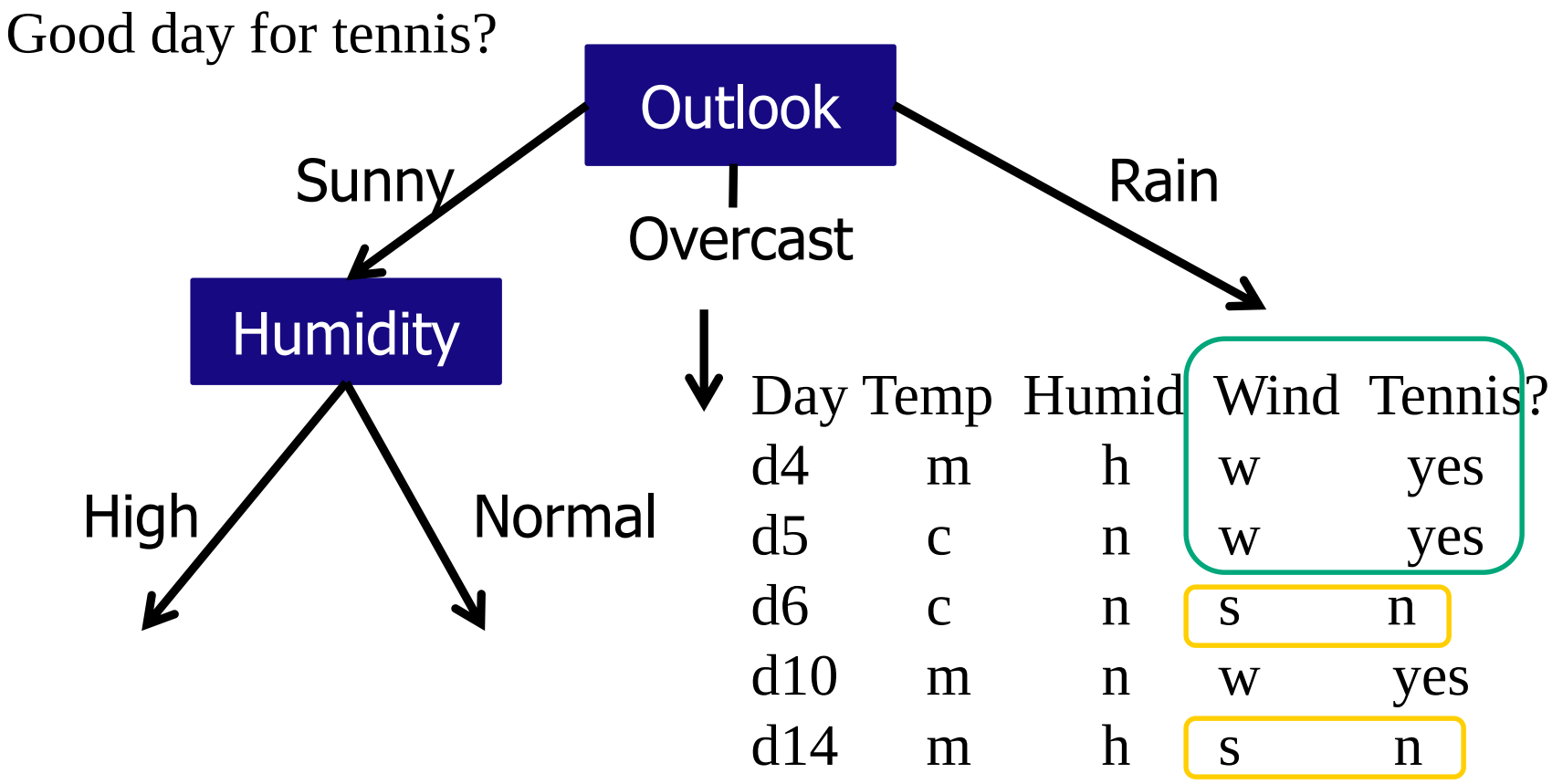



| Day | Temp | Humid | Wind | Tennis? |
|---|---|---|---|---|
| d1 | h | h | w | n |
| d2 | h | h | s | n |
| d8 | m | h | w | n |
| d9 | c | n | w | yes |
| d11 | m | n | s | yes |

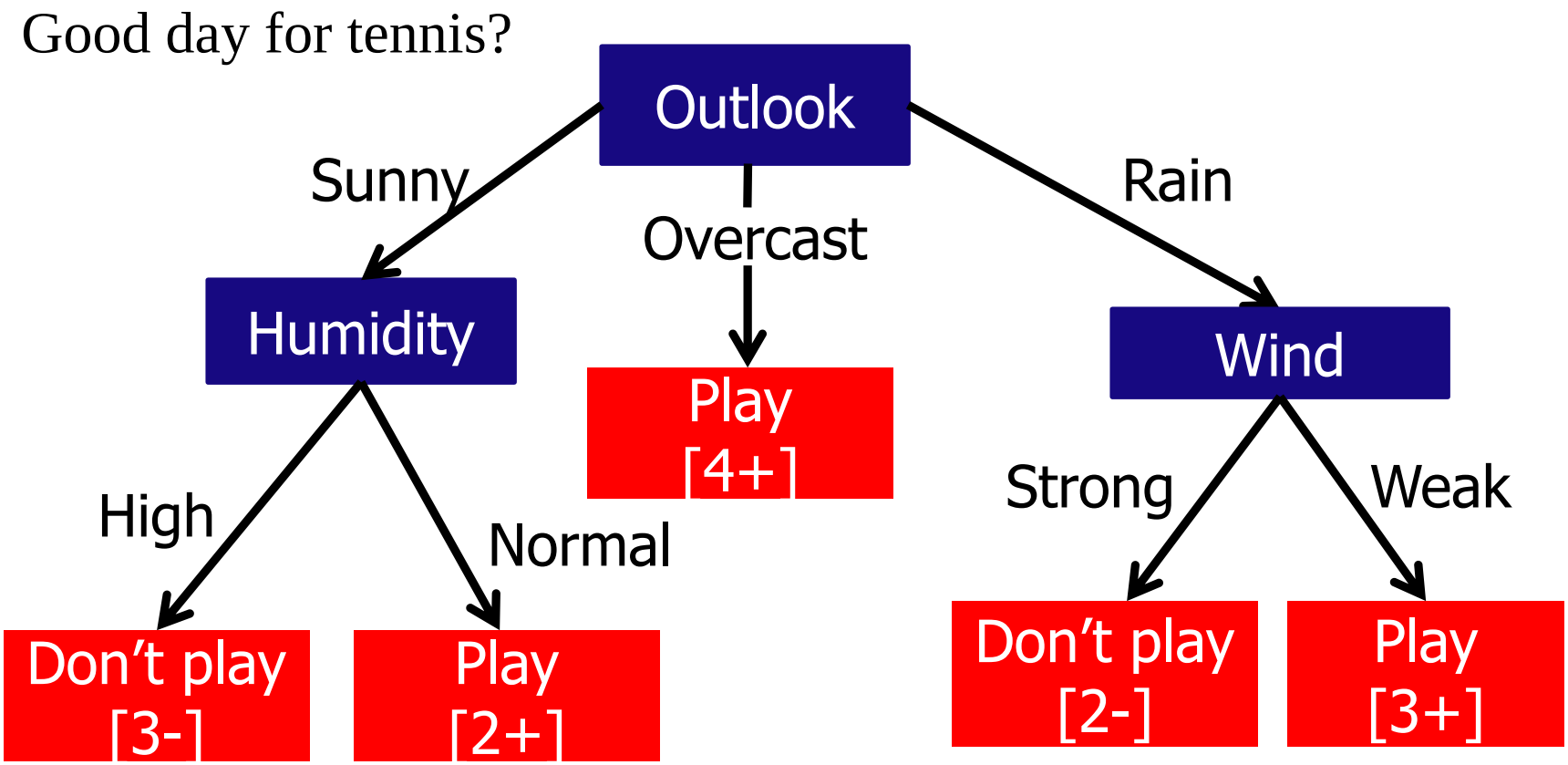
# One Step Later

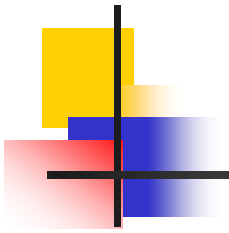


# Recurse Again



# One Step Later: Final Tree





# Group Activity

- Question: When would this basic decision tree learning algorithm be inefficient?

# Data Mining: Concepts and Techniques

## (3rd ed.)
## — Chapter 8 —

*Jiawei Han, Micheline Kamber, and Jian Pei*

*University of Illinois at Urbana-Champaign &*

*Simon Fraser University*

# Random Forest (Breiman 2001)

- *Random Forest:*
  - **Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split**
  - **During classification, each tree votes and the most popular class is returned**