

Classification 2

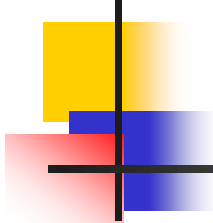


Induction of Decision Trees

ID3

Bassam Kurdy Ph.D
<bassam.kurdy@apinum.fr>

Training Examples



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	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	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

Training Examples

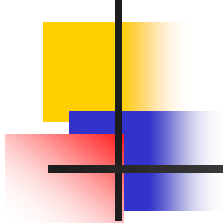
Exemple [Quinlan,86]

Attributs	Pif	Temp	Humid	Vent
Valeurs possibles	soleil,couvert,pluie	chaud,bon,frais	normale,haute	vrai,faux

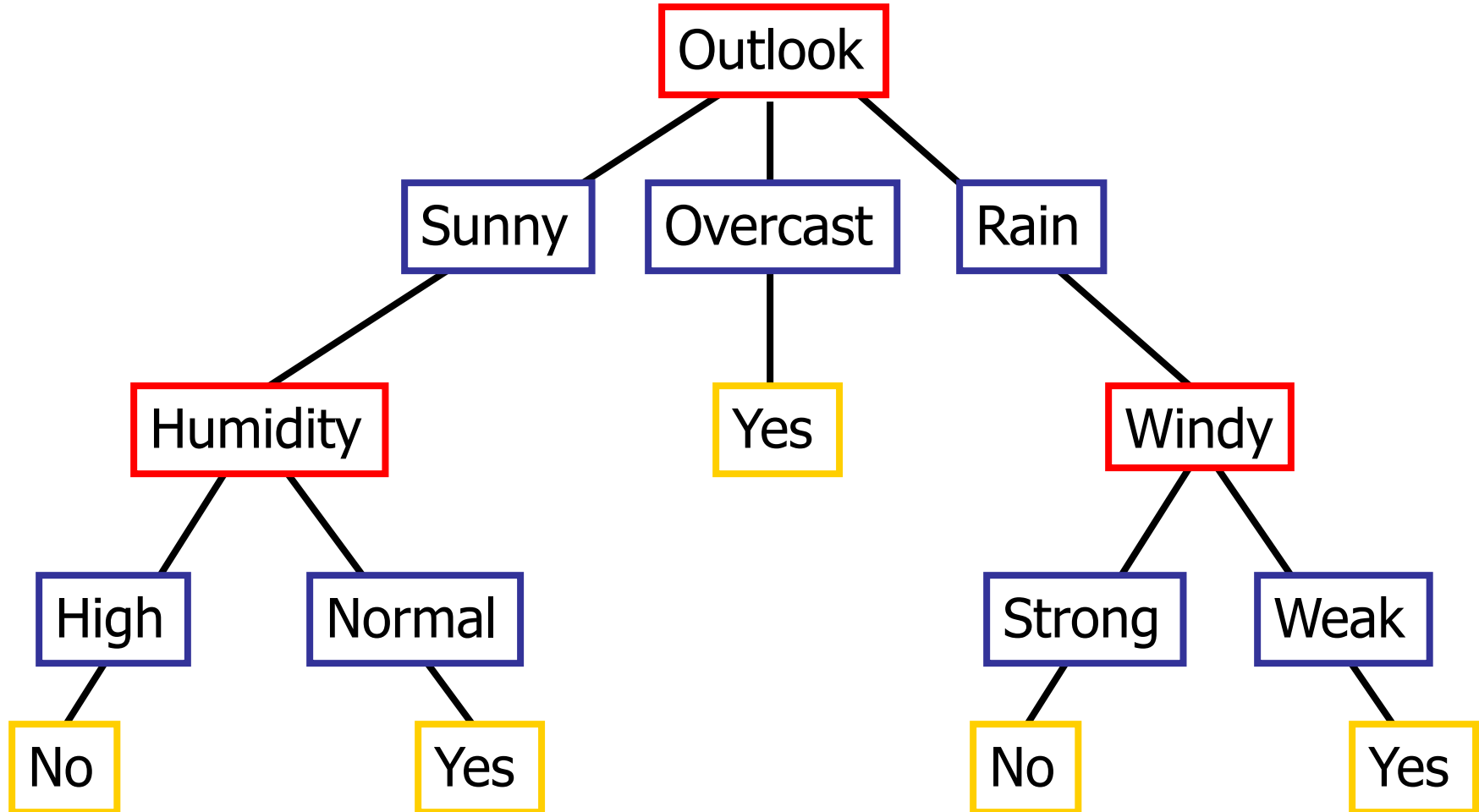
N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJouer
2	soleil	chaud	haute	vrai	NePasJouer
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJouer
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJouer
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJouer

la classe

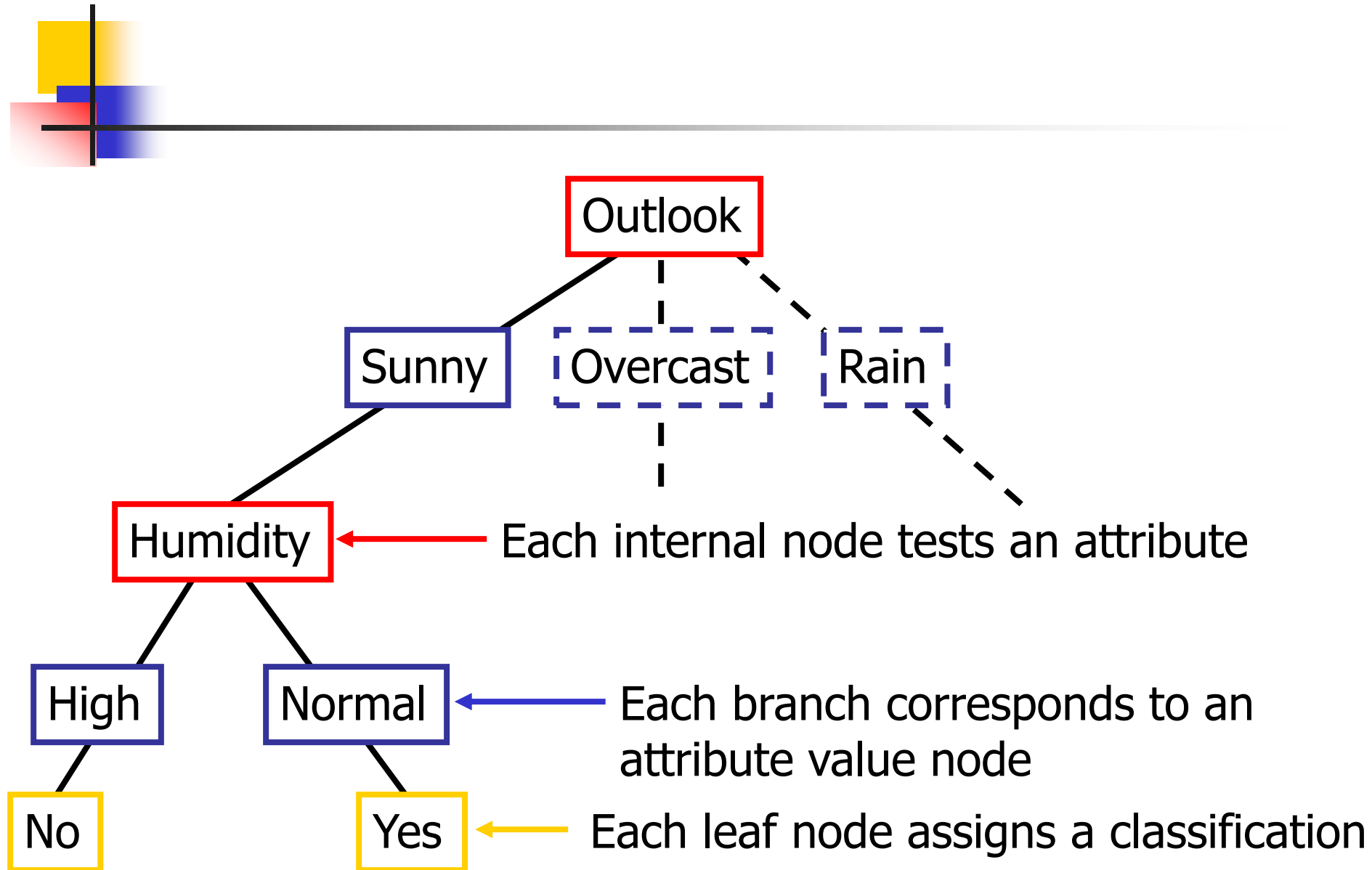
Decision Tree for PlayTennis



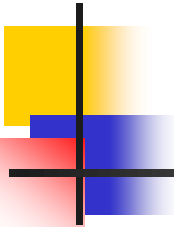
Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?



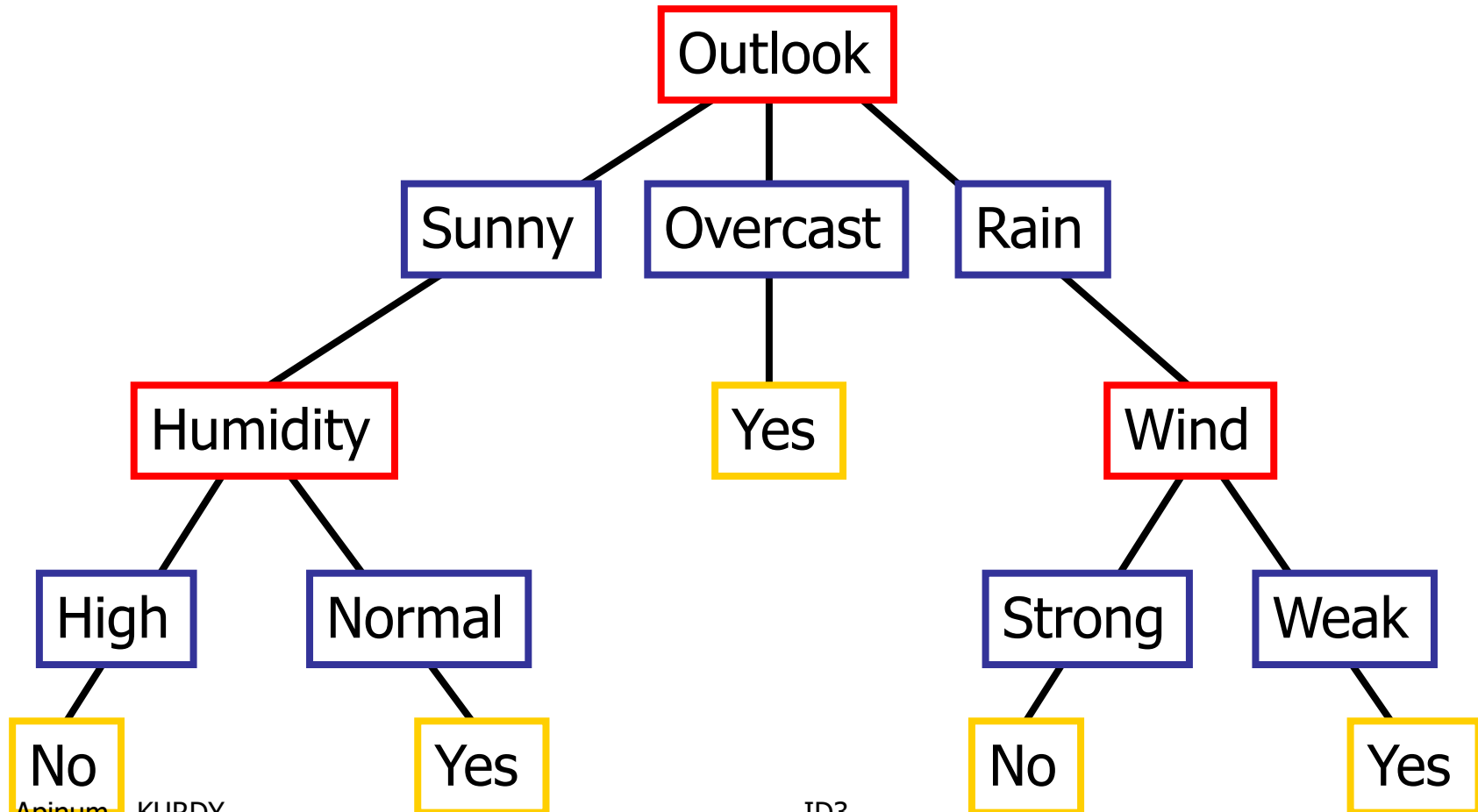
Decision Tree for PlayTennis



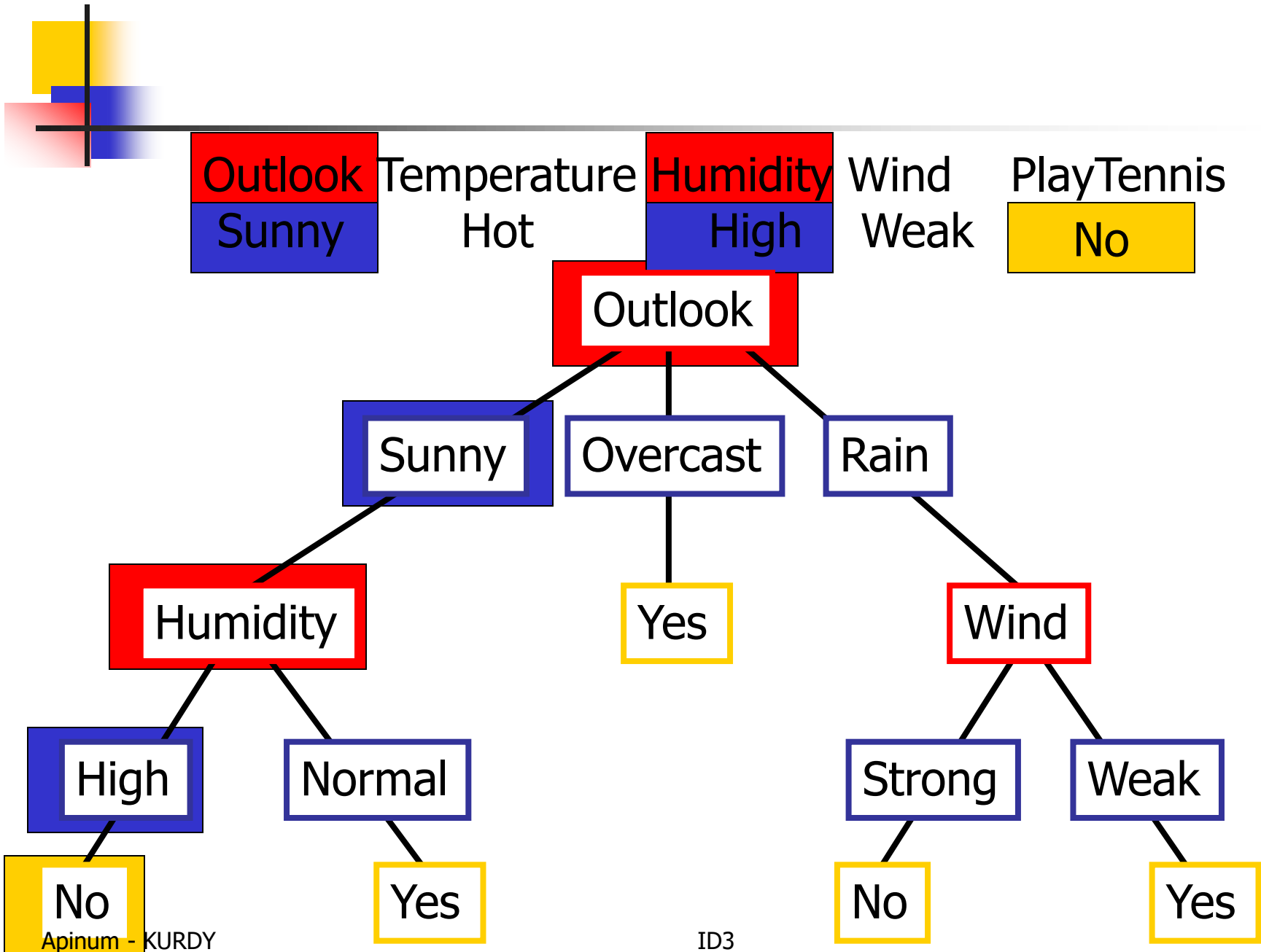
Decision Tree for PlayTennis



Outlook Temperature Humidity Wind PlayTennis
Sunny Hot High Weak ?

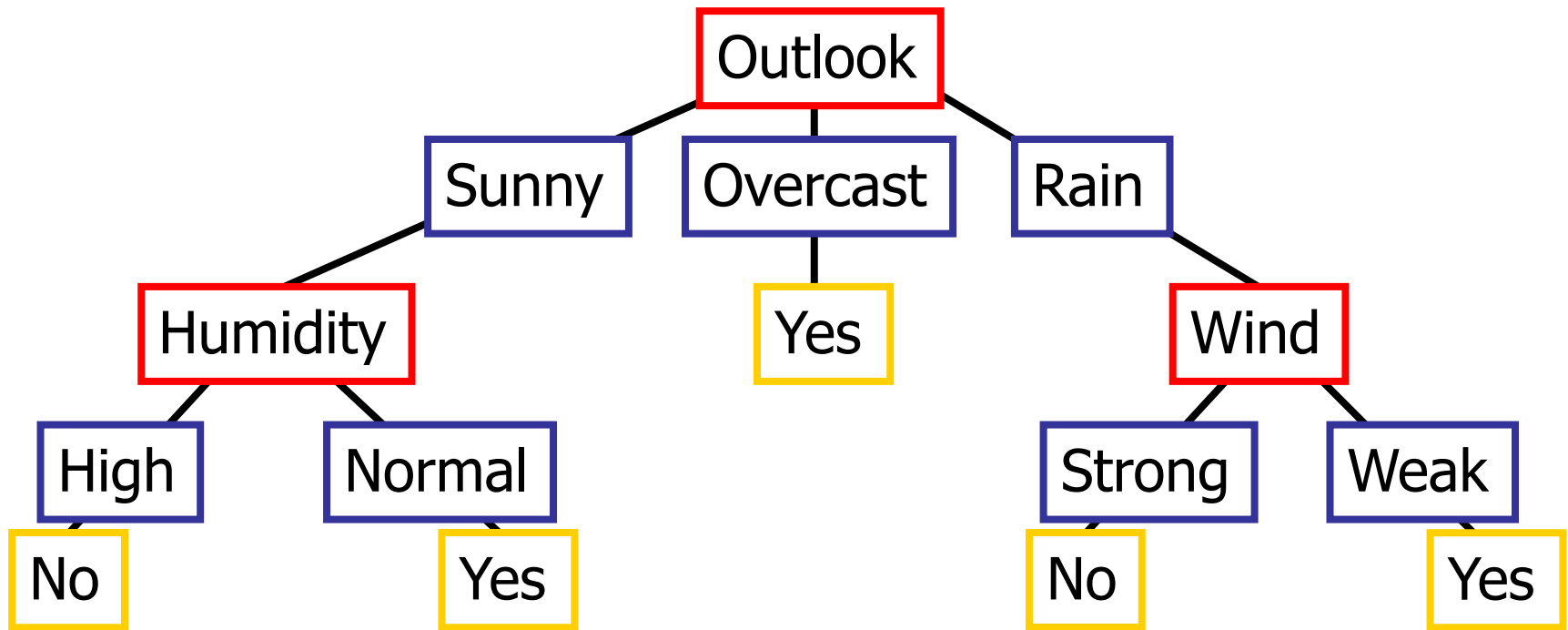


Decision Tree for PlayTennis



Decision Tree

- decision trees represent disjunctions of conjunctions

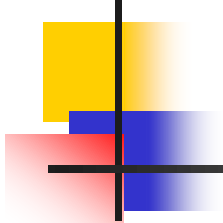


(Outlook=Sunny \wedge Humidity=Normal)

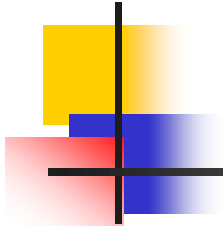
✓ (Outlook=Overcast)

✓ (Outlook=Rain \wedge Wind=Weak)

When to consider Decision Trees

- 
- Instances describable by attribute-value pairs
 - Target function is discrete valued
 - Disjunctive hypothesis may be required
 - Possibly noisy training data
 - Missing attribute values
 - Examples:
 - Medical diagnosis
 - Credit risk analysis
 - Object classification for robot manipulator

Motivation # 1: Analysis Tool



- Suppose that a company have a data base of sales data, lots of sales data
- How can that company's CEO use this data to figure out an effective sales strategy
- Safeway, Giant, etc cards: what is that for?

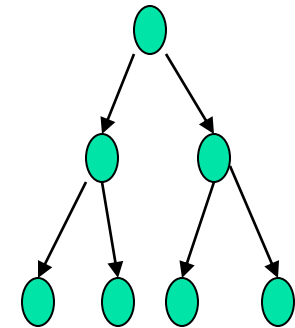
Motivation # 1: Analysis Tool (cont'd)

Sales data

Ex'ple	Bar	Fri	Hun	Pat	Type	Res	wait
x1	no	no	yes	some	french	yes	yes
x4	no	yes	yes	full	thai	no	yes
x5	no	yes	no	full	french	yes	no
x6							
x7							
x8							
x9							
x10							
x11							

induction

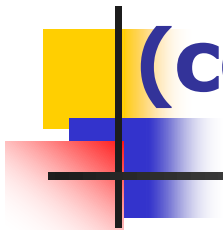
Decision Tree



"if buyer is male & and age between 24-35 & married
then he buys sport magazines"

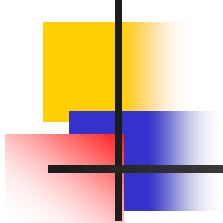
Motivation # 1: Analysis Tool

(cont'd)



- Decision trees has been frequently used in IDSS
- Some companies:
 - SGI: provides tools for decision tree visualization
 - Acknosoft (France), Tech:Inno (Germany): combine decision trees with CBR technology
- Several applications
- Decision trees are used for Data Mining

Parenthesis: Expert Systems

- 
- Have been used in :
 - medicine
 - oil and mineral exploration
 - weather forecasting
 - stock market predictions
 - financial credit, fault analysis
 - some complex control systems
 - Two components:
 - Knowledge Base
 - Inference Engine

The Knowledge Base in Expert Systems



A knowledge base consists of a collection of IF-THEN rules:

if buyer is male & age between 24-50 & married
then he buys sport magazines

if buyer is male & age between 18-30
then he buys PC games magazines

Knowledge bases of fielded expert systems contain hundreds and sometimes even thousands such rules. Frequently rules are contradictory and/or overlap



The Inference Engine in Expert Systems

The inference engine reasons on the **rules** in the knowledge base and the **facts** of the current problem

Typically the inference engine will contain policies to deal with conflicts, such as “**select the most specific rule in case of conflict**”

Some expert systems incorporate **probabilistic** reasoning, particularly those doing predictions

Expert Systems: Some Examples



MYCIN. It encodes expert knowledge to identify kinds of bacterial infections. Contains 500 rules and use some form of uncertain reasoning

DENDRAL. Identifies interpret mass spectra on organic chemical compounds

MOLGEN. Plans gene-cloning experiments in laboratories.

XCON. Used by DEC to configure, or set up, VAX computers. Contained 2500 rules and could handle computer system setups involving 100-200 modules.

Main Drawback of Expert Systems: The Knowledge Acquisition Bottle-Neck

The main problem of expert systems is acquiring knowledge from human specialist is a difficult, cumbersome and long activity.

Name	KB	#Rules	Const. time (man/years)	Maint. time (man/months)
MYCIN	KA	500	10	N/A
XCON	KA	2500	18	3

KB = Knowledge Base

KA = Knowledge Acquisition

Motivation # 2: Avoid Knowledge Acquisition Bottle-Neck



- GASOIL is an expert system for designing gas/oil separation systems stationed of-shore
- The design depends on multiple factors including:
 - proportions of gas, oil and water, flow rate, pressure, density, viscosity, temperature and others
- To build that system by hand would had taken 10 person years
- It took only 3 person-months by using inductive learning!
- GASOIL saved BP millions of dollars

Motivation # 2 : Avoid Knowledge Acquisition Bottle-Neck

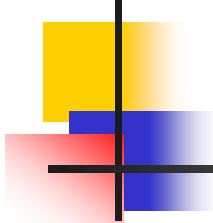
Name	KB	#Rules	Const. time (man/years)	Maint. time (man/months)
MYCIN	KA	500	10	N/A
XCON	KA	2500	18	3
GASOIL	IDT	2800	1	0.1
BMT	KA (IDT)	30000+	9 (0.3)	2 (0.1)

KB = Knowledge Base

KA = Knowledge Acquisition

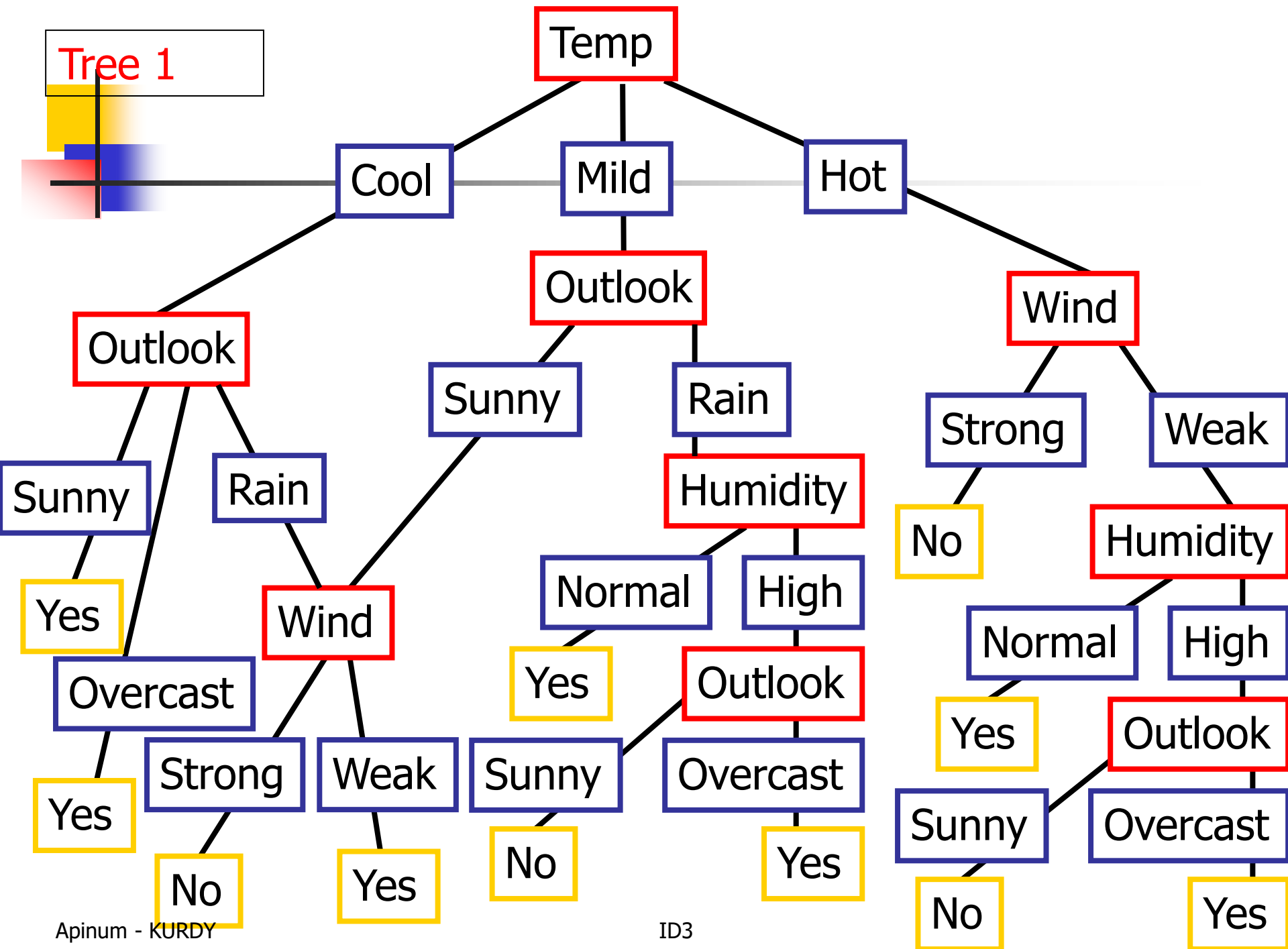
IDT = Induced Decision Trees

Training Examples

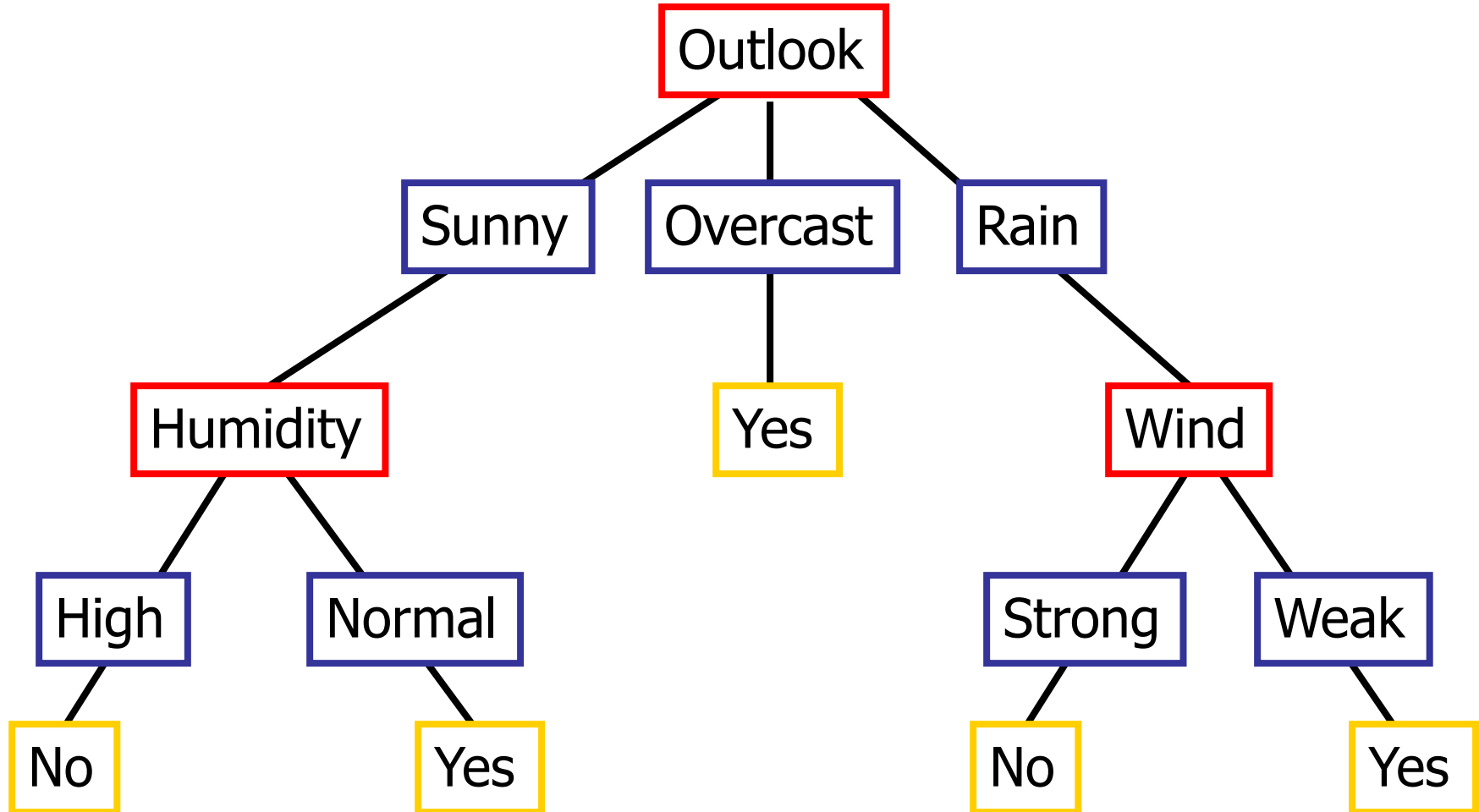


Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	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

Tree 1



Tree 2

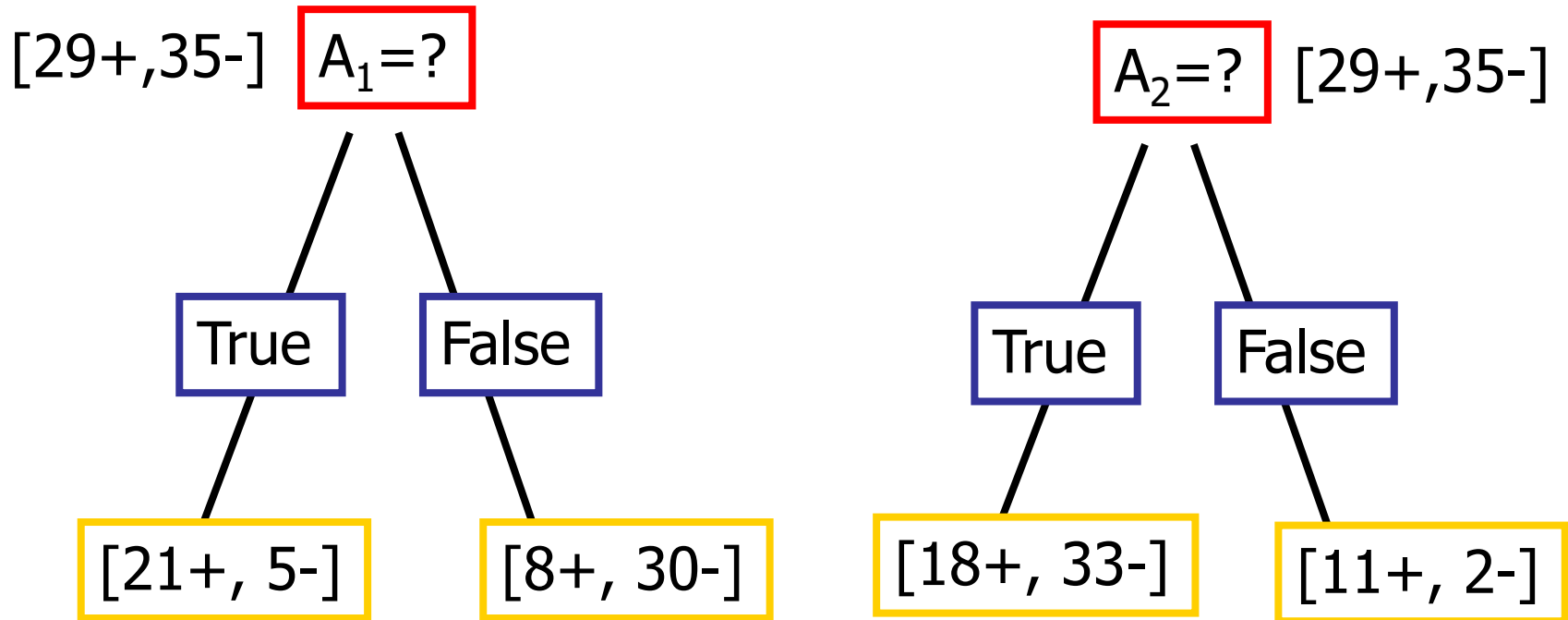




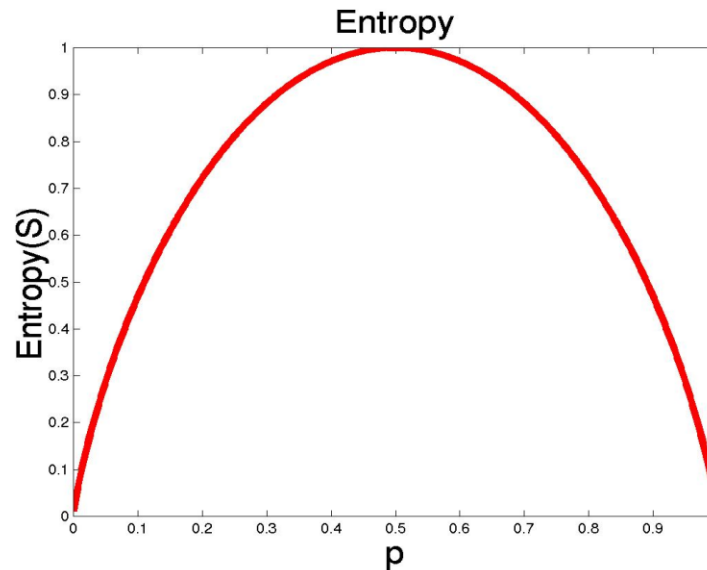
Top-Down Induction of Decision Trees ID3

1. $A \leftarrow$ the “**best**” decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A create new descendant
4. Sort training examples to leaf node according to the attribute value of the branch
5. If all training examples are **perfectly** classified (**same value of target attribute**) stop, else iterate over new leaf nodes.

Which Attribute is "best"?



Entropy



- S is a sample of training examples
- p_+ is the proportion of positive examples
- p_- is the proportion of negative examples
- Entropy measures the impurity of S

$$\text{Entropy}(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$$



Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there are two classes, P and N
 - Let the set of examples S contain p elements of class P and n elements of class N
 - The amount of information, needed to decide if an arbitrary example in S belongs to P or N is defined as

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$



Information Gain in Decision Tree Induction

- Assume that using attribute A a set S will be partitioned into sets $\{S_1, S_2, \dots, S_v\}$
 - If S_i contains p_i examples of P and n_i examples of N , the **entropy**, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on A

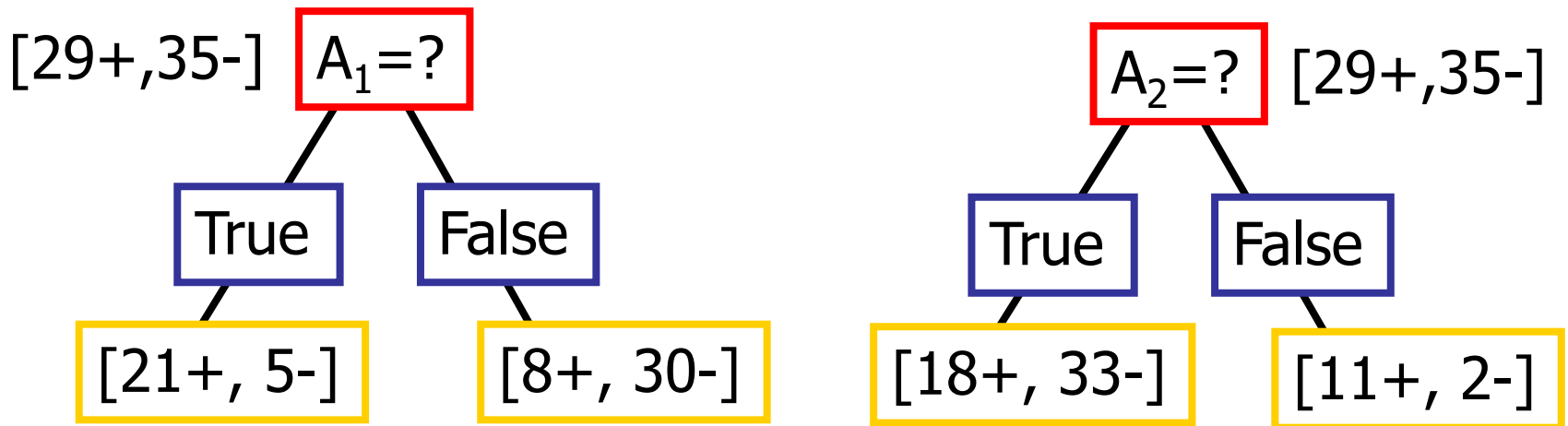
$$Gain(A) = I(p, n) - E(A)$$

Information Gain

- Gain(S,A): expected reduction in entropy due to sorting S on attribute A

$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} |S_v|/|S| \text{Entropy}(S_v)$$

$$\begin{aligned} \text{Entropy}([29+, 35-]) &= -29/64 \log_2 29/64 - 35/64 \log_2 35/64 \\ &= 0.99 \end{aligned}$$



Information Gain

$$\text{Entropy}([21+, 5-]) = 0.71$$

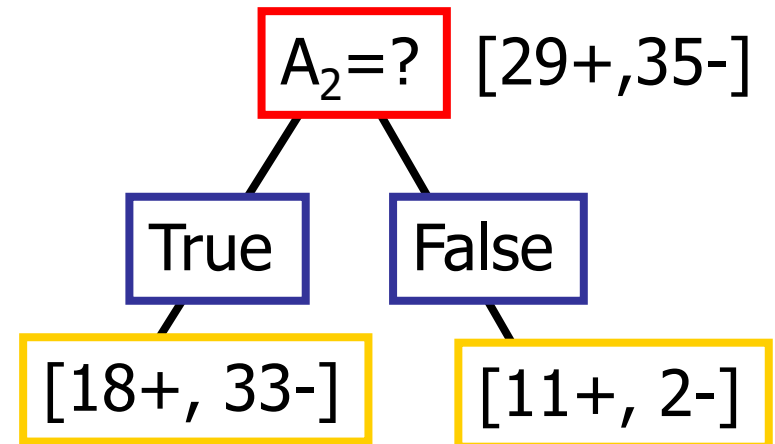
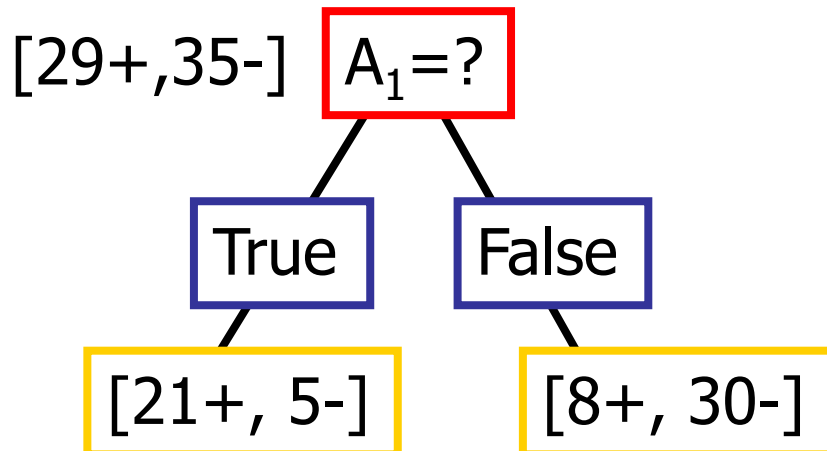
$$\text{Entropy}([8+, 30-]) = 0.74$$

$$\begin{aligned}\text{Gain}(S, A_1) &= \text{Entropy}(S) \\ &\quad - 26/64 * \text{Entropy}([21+, 5-]) \\ &\quad - 38/64 * \text{Entropy}([8+, 30-]) \\ &= 0.26\end{aligned}$$

$$\text{Entropy}([18+, 33-]) = 0.94$$

$$\text{Entropy}([11+, 2-]) = 0.62$$

$$\begin{aligned}\text{Gain}(S, A_2) &= \text{Entropy}(S) \\ &\quad - 51/64 * \text{Entropy}([18+, 33-]) \\ &\quad - 13/64 * \text{Entropy}([11+, 2-]) \\ &= 0.12\end{aligned}$$





Training Examples

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	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	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

$S=[9+,5-]$
 $E=0.940$

Humidity

High

$[3+, 4-]$

$E=0.985$

Normal

$[6+, 1-]$

$E=0.592$

$$\begin{aligned}\text{Gain}(S, \text{Humidity}) &= 0.940 - (7/14) * 0.985 \\ &\quad - (7/14) * 0.592 \\ &= 0.151\end{aligned}$$

$S=[9+,5-]$
 $E=0.940$

Wind

Weak

$[6+, 2-]$

$E=0.811$

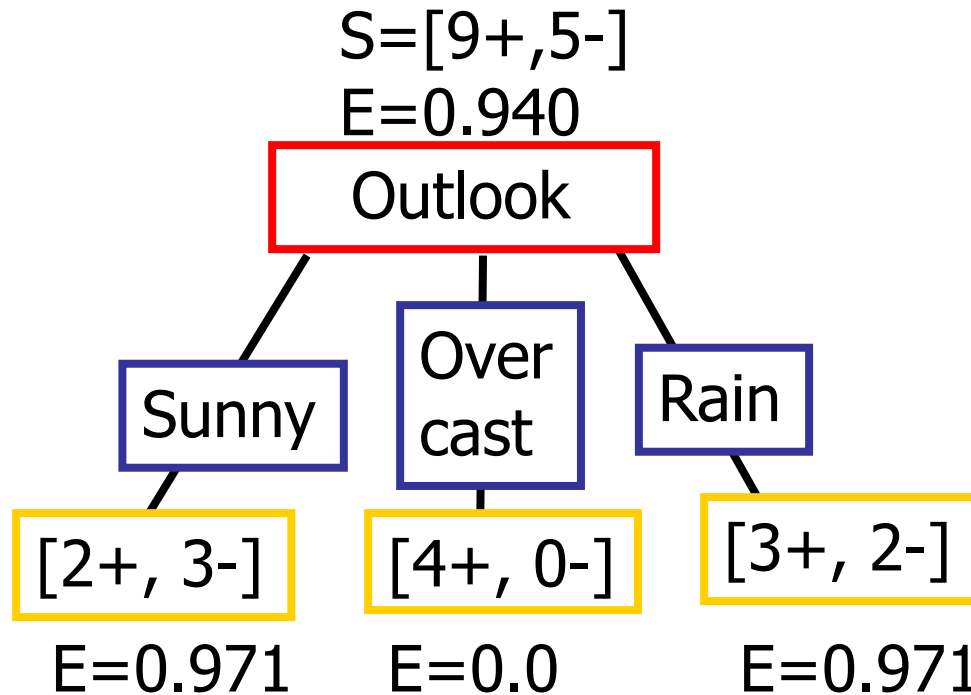
Strong

$[3+, 3-]$

$E=1.0$

$$\begin{aligned}\text{Gain}(S, \text{Wind}) &= 0.940 - (8/14) * 0.811 \\ &\quad - (6/14) * 1.0 \\ &= 0.048\end{aligned}$$

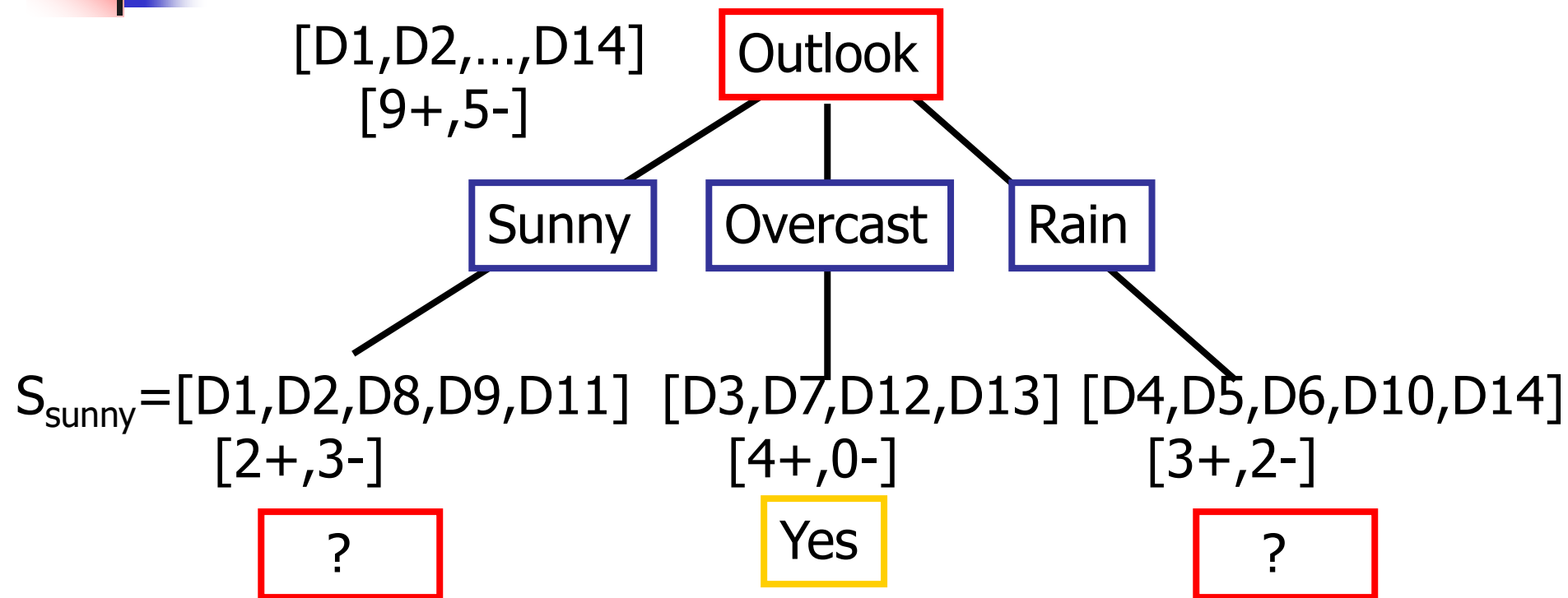
Selecting the Next Attribute



Temp ?

$$\begin{aligned} \text{Gain}(S, \text{Outlook}) &= 0.940 - (5/14) * 0.971 \\ &\quad - (4/14) * 0.0 - (5/14) * 0.971 \\ &= 0.246 \end{aligned}$$

ID3 Algorithm

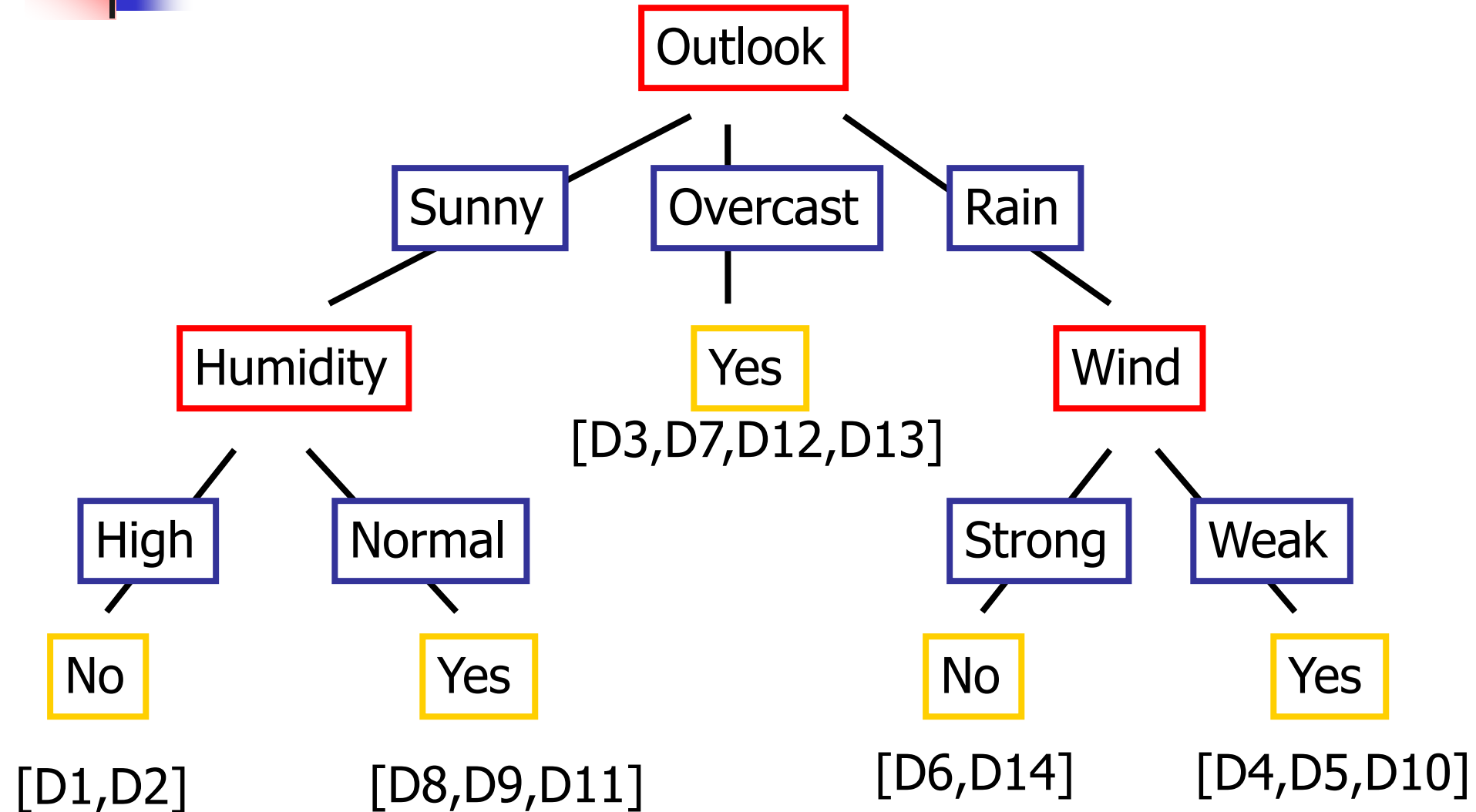


$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = 0.971 - (3/5)0.0 - 2/5(0.0) = 0.971$$

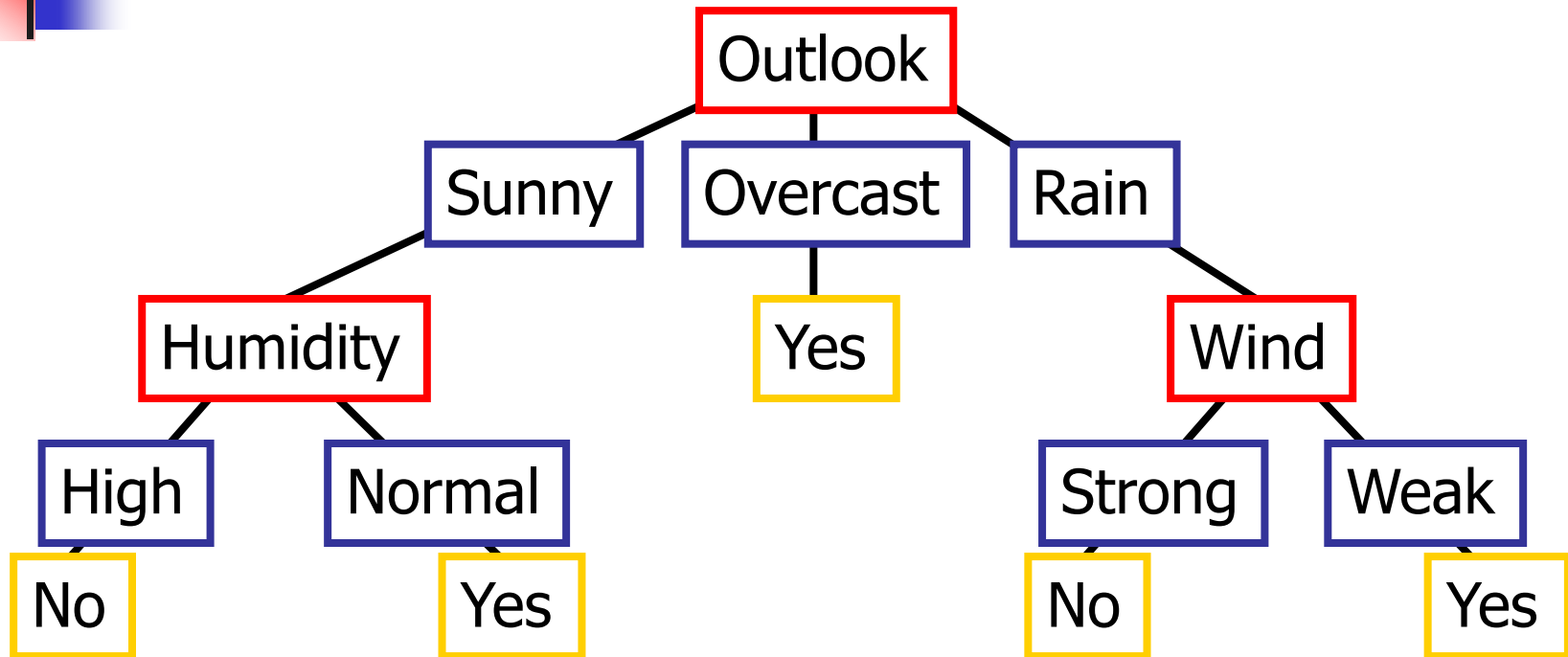
$$\text{Gain}(S_{\text{sunny}}, \text{Temp.}) = 0.971 - (2/5)0.0 - 2/5(1.0) - (1/5)0.0 = 0.571$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = 0.971 - (2/5)1.0 - 3/5(0.918) = 0.020$$

ID3 Algorithm



Converting a Tree to Rules



- R_1 : If (Outlook=Sunny) \wedge (Humidity=High) Then PlayTennis=No
 R_2 : If (Outlook=Sunny) \wedge (Humidity=Normal) Then PlayTennis=Yes
 R_3 : If (Outlook=Overcast) Then PlayTennis=Yes
 R_4 : If (Outlook=Rain) \wedge (Wind=Strong) Then PlayTennis=No
 R_5 : If (Outlook=Rain) \wedge (Wind=Weak) Then PlayTennis=Yes



Training Dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Attribute Selection by Information Gain Computation

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- $I(p, n) = I(9, 5) = 0.940$
- Compute the entropy for *age*:

age	p_i	n_i	$I(p_i, n_i)$
≤ 30	2	3	0.971
30...40	4	0	0
> 40	3	2	0.971

$$E(\text{age}) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

Hence

$$\text{Gain}(\text{age}) = I(p, n) - E(\text{age})$$

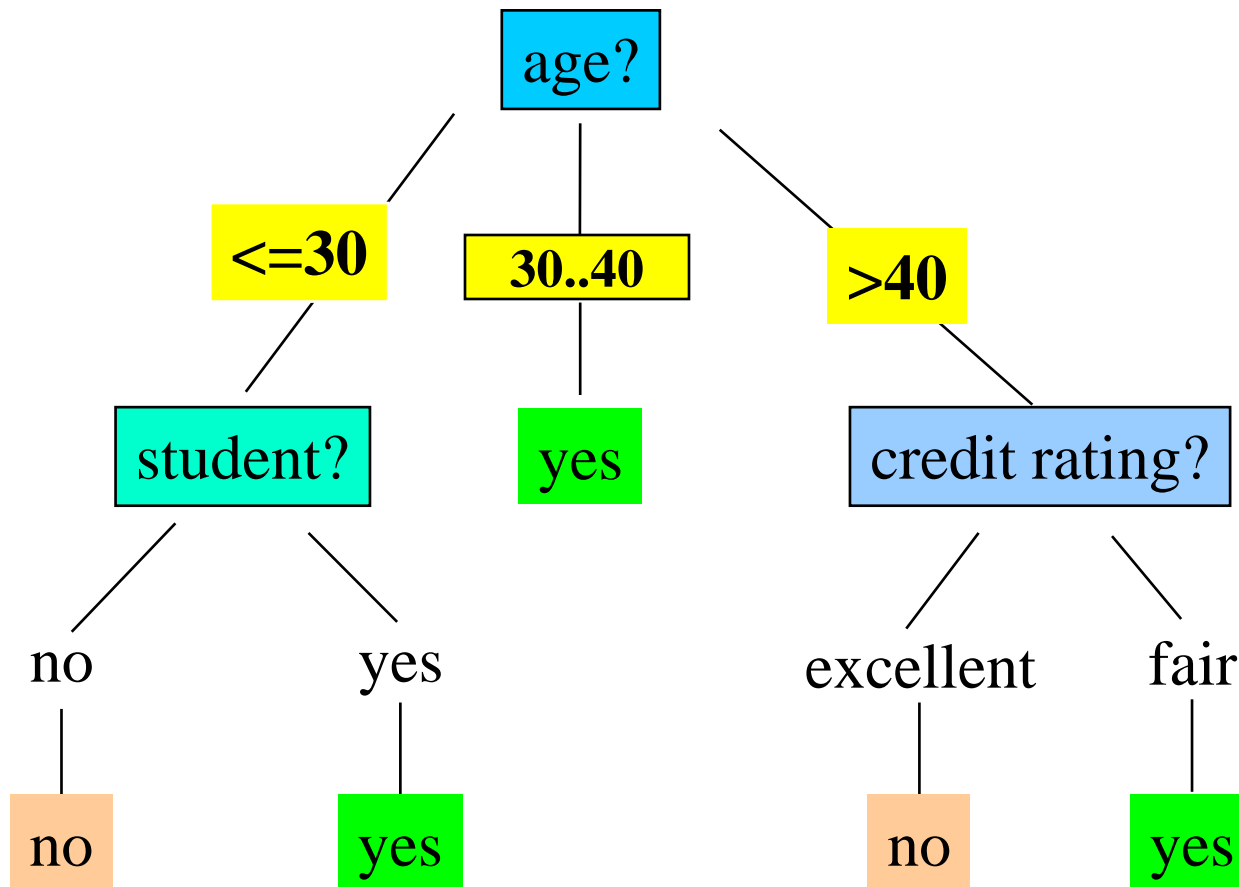
Similarly

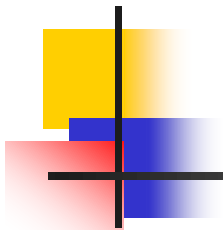
$$\text{Gain}(\text{income}) = 0.029$$

$$\text{Gain}(\text{student}) = 0.151$$

$$\text{Gain}(\text{credit_rating}) = 0.048$$

Output: A Decision Tree for “*buys_computer*”





Attribute Selection Measure: Information Gain (ID3)

- Select the attribute with the highest **information gain**
- S contains s_i tuples of class C_i for $i = \{1, \dots, m\}$
- **Information** measures info required to classify any arbitrary tuple:

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m \frac{s_i}{S} \log_2 \frac{s_i}{S}$$

- **Expected information** of attribute A with values $\{a_1, a_2, \dots, a_v\}$:

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{S} I(s_{1j}, \dots, s_{mj})$$

- **information Gained** by branching on attribute A :

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A)$$



Continuous Valued Attributes

Create a **discrete** attribute to test continuous

- Temperature = 24.5°C
- (Temperature > 20.0°C) = {true, false}

Where to set the threshold?

Temperature	15°C	18°C	19°C	22°C	24°C	27°C
PlayTennis	No	No	Yes	Yes	Yes	No



Attributes with many Values

- **Problem:** if an attribute has many values, maximizing *InformationGain* will select it.
- E.g.: Imagine using Date=12.7.1996 as attribute perfectly splits the data into subsets of size 1.
Use *GainRatio* instead of information gain as criteria:

$$\text{GainRatio}(S, A) = \text{Gain}(S, A) / \text{SplitInformation}(S, A)$$

$$\text{SplitInformation}(S, A) = -\sum_{i=1..c} |S_i|/|S| \log_2 |S_i|/|S|$$

Where S_i is the subset for which attribute A has the value v_i



Attributes with Cost

Consider:

- Medical diagnosis : blood test costs 1000 S.P.
- Robotics: width_from_one_feet has cost 23 secs.

How to learn a consistent tree with low expected
cost?

Replace *Gain* by :

$$\text{Gain}^2(S,A)/\text{Cost}(A) \quad [\text{Tan, Schimmer 1990}]$$

$$2^{\text{Gain}(S,A)-1}/(\text{Cost}(A)+1)^w \quad : w \in [0,1] \quad [\text{Nunez 1988}]$$



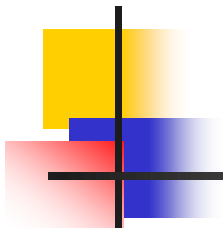
Unknown Attribute Values

What if examples are missing values of A?:

Use training example anyway sort through tree:

- If node n tests A , assign most common value of A among other examples **sorted to node n** .
- Assign most common value of A among other examples **with same target value**
- Assign probability p_i to each possible value v_i of A
 - Assign fraction p_i of example to each descendant in tree

Classify new examples in the same fashion



MS

Solution1 - Microsoft Visual Studio

File Edit View Project Build Debug Database Mining Model Tools Test Analyze Window Help



MS.dmm [Design] DSV.dsv [Design] Start Page

Mining Structure Mining Models Mining Model Viewer Mining Accuracy Chart Mining Model Prediction

Mining Model: MM

Viewer: Microsoft Tree Viewer

Decision Tree Dependency Network



Tree: Play

Default Expansion: 3 Levels

Histograms: 6

Background: All Cases

Show Level 1 Level 3

