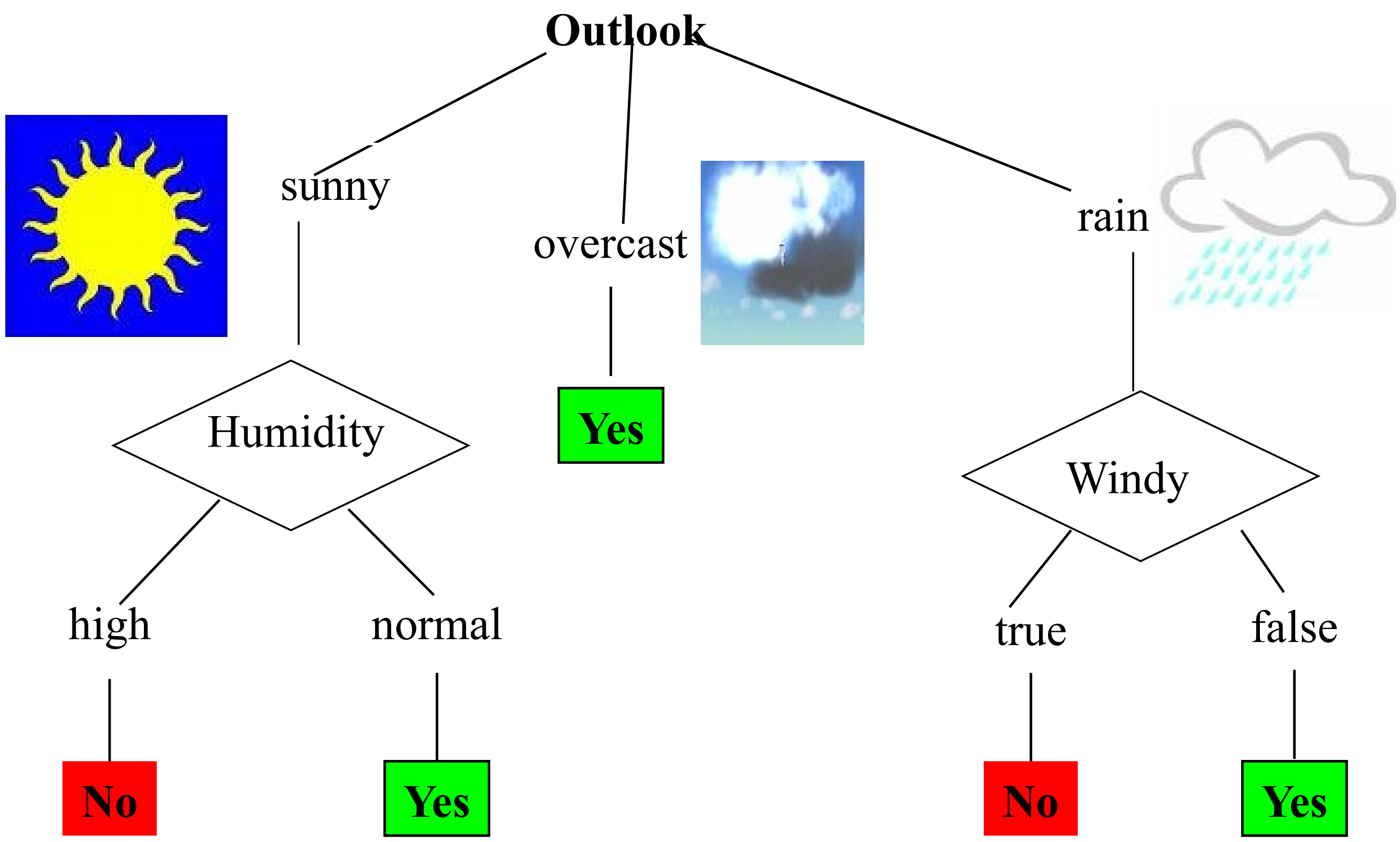
Decision Tree

Decision Tree learning

- Rules for classifying data using attributes.
- The tree consists of decision nodes and leaf nodes.
- A decision node has two or more branches, each representing values for the attribute tested.
- A leaf node attribute produces a homogeneous result (all in one class), which does not require additional classification testing.

Decision Tree Example



When to consider Decision Trees

- Instances of attribute-value pairs
- Target function is discrete valued
- Missing attribute values
- Examples:
  - Medical diagnosis
  - Credit risk analysis
  - Object classification

Decision Tree

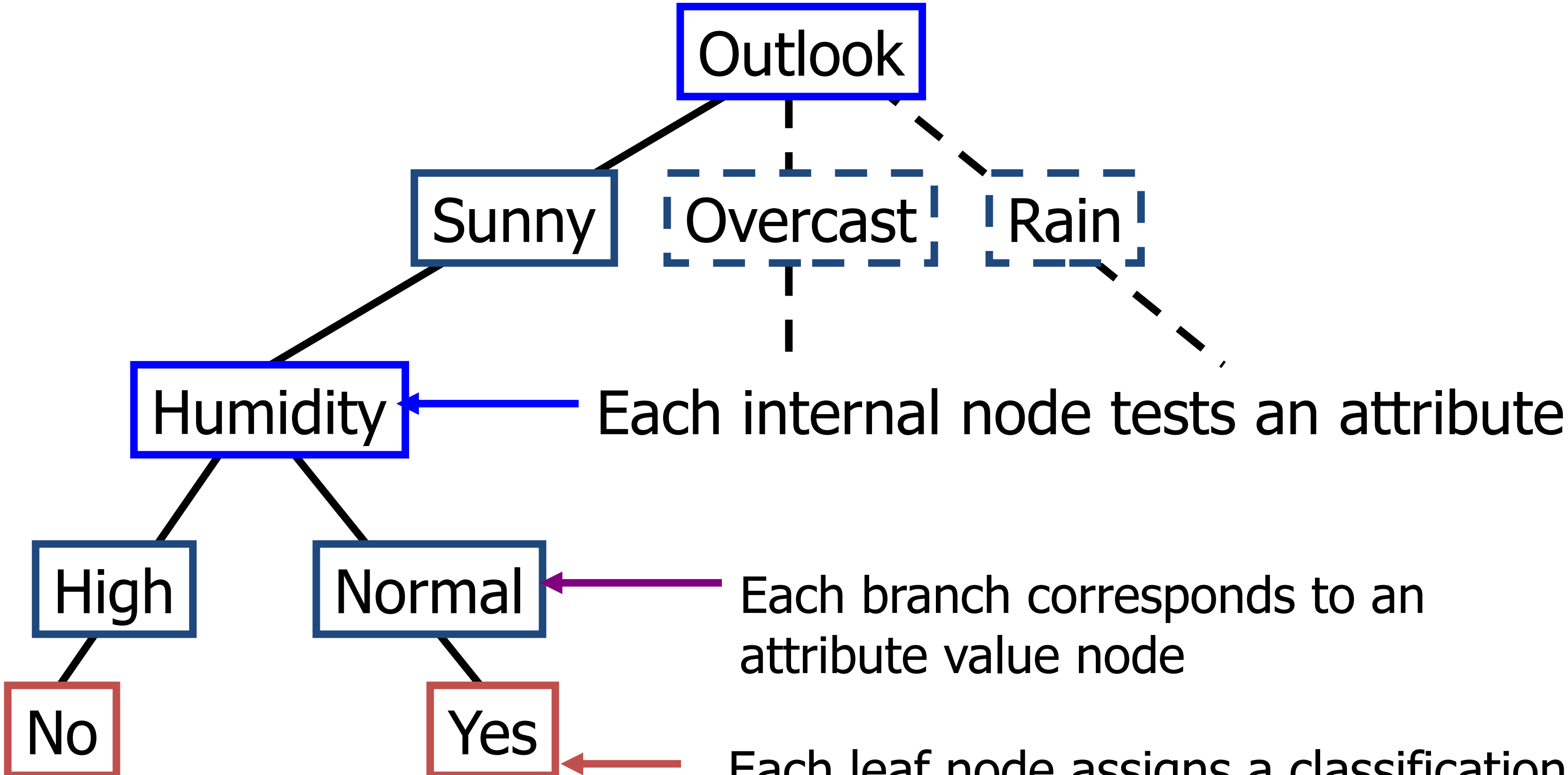
Given:

- Database schema contains {A1, A2, …, Ah}
- D = {t1, …, tn} where ti=<ti1, …, tih>
- Classes C={C1, …., Cm}

Decision or Classification Tree is a tree associated with D such that

- Each internal node is labeled with attribute, Ai
- Each arc is labeled with predicate which can be applied to attribute at parent
- Each leaf node is labeled with a class, Cj

Decision Tree



Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayCricket
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

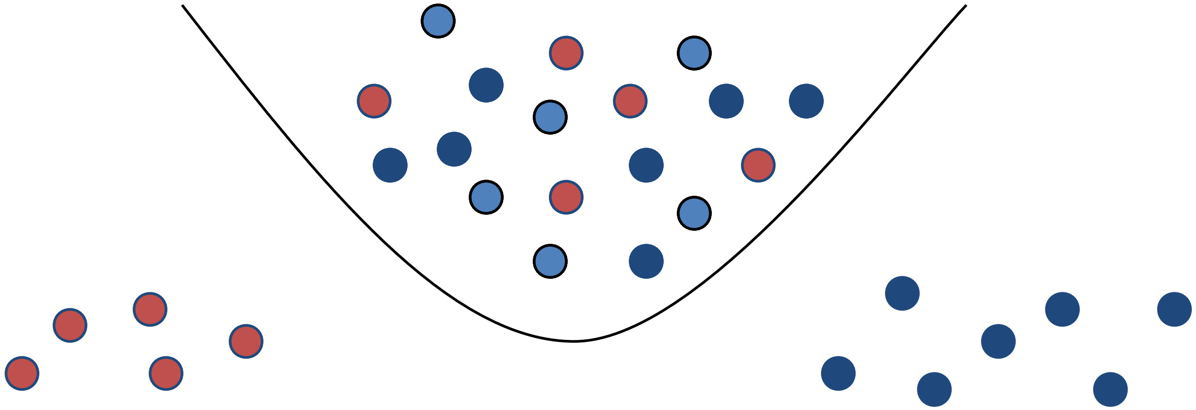
Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayCricket
D1	S	H	H	W	No
D2	S	H	H	S	No
D3	O	H	H	W	Yes
D4	R	M	H	W	Yes
D5	R	C	N	W	Yes
D6	R	C	N	S	No
D7	O	C	N	S	Yes
D8	S	M	H	W	No
D9	S	C	N	W	Yes
D10	R	M	N	W	Yes
D11	S	M	N	S	Yes
D12	O	M	H	S	Yes
D13	O	H	N	W	Yes
D14	R	M	H	S	No

DT Issues

- Choosing Splitting Attributes
- Ordering of Splitting Attributes
- Splits
- Tree Structure
- Stopping Criteria
- Training Data
- Pruning

Information



DT Induction

- When all the marbles in the bowl are mixed up, little information is given.
- When the marbles in the bowl are all from one class and those in the other two classes are on either side, more information is given.

DT Induction

Input:

D //Training data

Output:

T //Decision Tree

DTBuild Algorithm:

//Simplistic algorithm to illustrate naive approach to building DT

$T = \emptyset$;

Determine best splitting criterion;

T = Create root node node and label with splitting attribute;

T = Add arc to root node for each split predicate and label;

for each arc do

D = Database created by applying splitting predicate to D;

if stopping point reached for this path then

T' = Create leaf node and label with appropriate class;

else

$T' = DTBuild(D)$;

T = Add T' to arc;

Entropy

- **Entropy** measures the **amount of randomness or surprise or uncertainty.**
- ***Entropy*** E is defined as:
  - **E(D)** = Σci=1 pi log2 1/pi ,
    - Where D is a dataset,
    - c is the number of classes, and
    - pi is the proportion of the training dataset belongs to class i
- Goal in classification
  - no surprise (entropy = 0)
  - 0 log2 0 = 0

ID3

- It creates tree using information theory concepts.
- It chooses split attribute with the highest information gain:

- $G(D,S) = E(D) - \sum_{i=1}^{c} P(D_i)E(D_i)$
- where S is the splitting attribute

Top-Down Induction of Decision Trees ID3

1. Let A be 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 dataset to leaf node according to the attribute value of the branch
5. If all training dataset are perfectly classified (same value of target attribute) stop, else iterate over new leaf nodes.

- $\log2(X) = \log10(X) / \log10(2)$

Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	S	H	H	W	No
D2	S	H	H	S	No
D3	O	H	H	W	Yes
D4	R	M	H	W	Yes
D5	R	C	N	W	Yes
D6	R	C	N	S	No
D7	O	C	N	S	Yes
D8	S	M	H	W	No
D9	S	C	N	W	Yes
D10	R	M	N	W	Yes
D11	S	M	N	S	Yes
D12	O	M	H	S	Yes
D13	O	H	N	W	Yes
D14	R	M	H	S	No

Decision Tree Learning

- 14 training; 9(**Yes),** 5 (**No**).
- Let E([X+,Y-]) represent that there are X positive(Yes) training elements and Y negative elements.
- Therefore the Entropy for the training dataset, E(D), can be represented as E([9+,5-])

Decision Tree Learning: A Simple Example

- $E(D) = \Sigma_{ci=1}$ pi log2 1/pi

  $=\Sigma_{ci=1}$ -pi log2 pi ,

Initial Entropy of the Training Set.

E(D) = E([9+,5-])

= 0.94= (-9/14 log2 9/14) + (-5/14 log2 5/14)

Decision Tree Learning: A Simple Example

- Calculate the information gain G(D,S) for each attribute S where S is taken from the set {Outlook, Temperature, Humidity, Wind}.
- $G(D,S) = E(D) - \Sigma ci=1\ P(Di)E(Di)$

Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	S	H	H	W	No
D2	S	H	H	S	No
D3	O	H	H	W	Yes
D4	R	M	H	W	Yes
D5	R	C	N	W	Yes
D6	R	C	N	S	No
D7	O	C	N	S	Yes
D8	S	M	H	W	No
D9	S	C	N	W	Yes
D10	R	M	N	W	Yes
D11	S	M	N	S	Yes
D12	O	M	H	S	Yes
D13	O	H	N	W	Yes
D14	R	M	H	S	No

Decision Tree Learning: A Simple Example

- The information gain for Outlook is:
  - **G(D,Outlook) = E(D) – [5/14 * E(Outlook=sunny) + 4/14 * E(Outlook = overcast) + 5/14 * E(Outlook=rain)]**
  - G(D,Outlook) = **E([9+,5-])** – [5/14*E(2+,3-) + 4/14*E([4+,0-]) + 5/14*E([3+,2-])]
  - G(D,Outlook) = **0.94** – [5/14*0.971 + 4/14*0.0 + 5/14*0.971]
  - **G(D,Outlook) = 0.246**

Decision Tree Learning: A Simple Example

- **G(D,Temperature) = 0.94 – [4/14*E(Temperature=hot) + 6/14*E(Temperature=mild) + 4/14*E(Temperature=cool)]**
- G(D,Temperature) = 0.94 – [4/14*E([2+,2-]) + 6/14*E([4+,2-]) + 4/14*E([3+,1-])]
- G(D,Temperature) = 0.94 – [4/14 + 6/14*0.918 + 4/14*0.811]
- **G(D,Temperature) = 0.029**

Decision Tree Learning: A Simple Example

- G(D,Humidity) = **0.94 – [7/14*E(Humidity=high) + 7/14*E(Humidity=normal)]**
- G(D,Humidity = 0.94 – [7/14*E([3+,4-]) + 7/14*E([6+,1-])]
- G(D,Humidity = 0.94 – [7/14*0.985 + 7/14*0.592]
- **G(D,Humidity) = 0.1515**

Decision Tree Learning: A Simple Example

- **G(D,Wind) = 0.94 – [8/14*0.811 + 6/14*1.00]**
- **G(D,Wind) = 0.048**

Maximum Gain

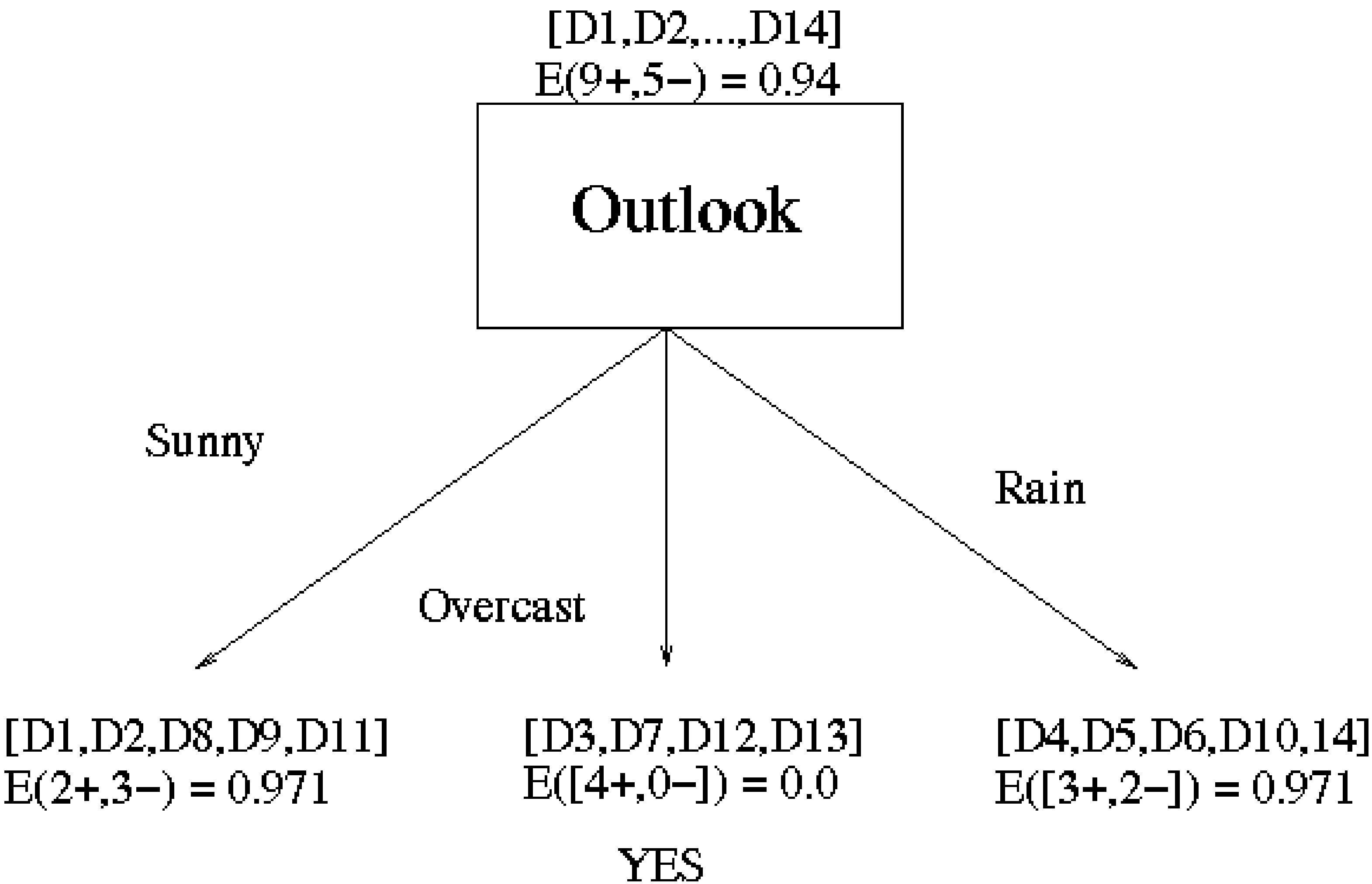
- **G(D,Outlook) = 0.246**
- **G(D,Temperature) = 0.029**
- **G(D,Humidity) = 0.1515**
- **G(D,Wind) = 0.048**
- Maximum gain is of Outlook➔ Outlook is best splitting attribute

Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	S	H	H	W	No
D2	S	H	H	S	No
D3	O	H	H	W	Yes
D4	R	M	H	W	Yes
D5	R	C	N	W	Yes
D6	R	C	N	S	No
D7	O	C	N	S	Yes
D8	S	M	H	W	No
D9	S	C	N	W	Yes
D10	R	M	N	W	Yes
D11	S	M	N	S	Yes
D12	O	M	H	S	Yes
D13	O	H	N	W	Yes
D14	R	M	H	S	No

Decision Tree Learning: A Simple Example

- Outlook is best splitting attribute



Decision Tree Learning: A Simple Example

- The root of our decision tree is Outlook (Sunny, Overcast, and Rain)
- Next, recursively find the nodes that should go below it.

Decision Tree Learning

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	S	H	H	W	No
D2	S	H	H	S	No
D3	O	H	H	W	Yes
D4	R	M	H	W	Yes
D5	R	C	N	W	Yes
D6	R	C	N	S	No
D7	O	C	N	S	Yes
D8	S	M	H	W	No
D9	S	C	N	W	Yes
D10	R	M	N	W	Yes
D11	S	M	N	S	Yes
D12	O	M	H	S	Yes
D13	O	H	N	W	Yes
D14	R	M	H	S	No

Decision Tree Learning: A Simple Example

- $G(Outlook=Rain, Humidity) = 0.971 - [2/5*E(Outlook=Rain \wedge Humidity=high) + 3/5*E(Outlook=Rain \wedge Humidity=normal]$
- **$G(Outlook=Rain, Humidity) = 0.02$**

- $G(Outlook=Rain,Wind) = 0.971- [3/5*0 + 2/5*0]$
- **$G(Outlook=Rain,Wind) = 0.971$**

Decision Tree Learning: A Simple Example

- Decision tree looks like: