

Decision Tree Learning

Abdus Salam Azad

**When should I
play Tennis ????**



**When should I
play Tennis ????**

**Can You Give Me
Some Examples
!!!!!!!**



Classification

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

Classification

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

Seems Like I really
enjoy it when the
outlook is
overcast!!!!

Day	Outlook	Wind	PlayTennis
D1	Sunny	Weak	No
I			No
I			Yes
I			Yes
I			Yes
I			No
I			Yes
I			No
I			Yes
D			Yes
D			Yes
D			No



Classification

**What About When
Its Sunny or
Raining ????**

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



Classification

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

Classification

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Classification

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

So, Its Humidity !!!!!!!

How did we do that ?????



Classifi

I get it !!!
Its Recursion
!!!!!!!!!!!!!!!!!!!!

Day	Outlook	Temp	Humidity	Wind Speed	Play Tennis
D1	Sunny	68	70	5	No
D2	Sunny	72	65	5	No
D8	Sunny	75	68	5	No
D9	Sunny	78	70	5	Yes
D11	Sunny	80	72	5	Yes



Download from
Dreamstime.com

This watermark comp image is for previewing purposes only.



ID 19745752

© Danil Chepko | Dreamstime.com

Classification

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

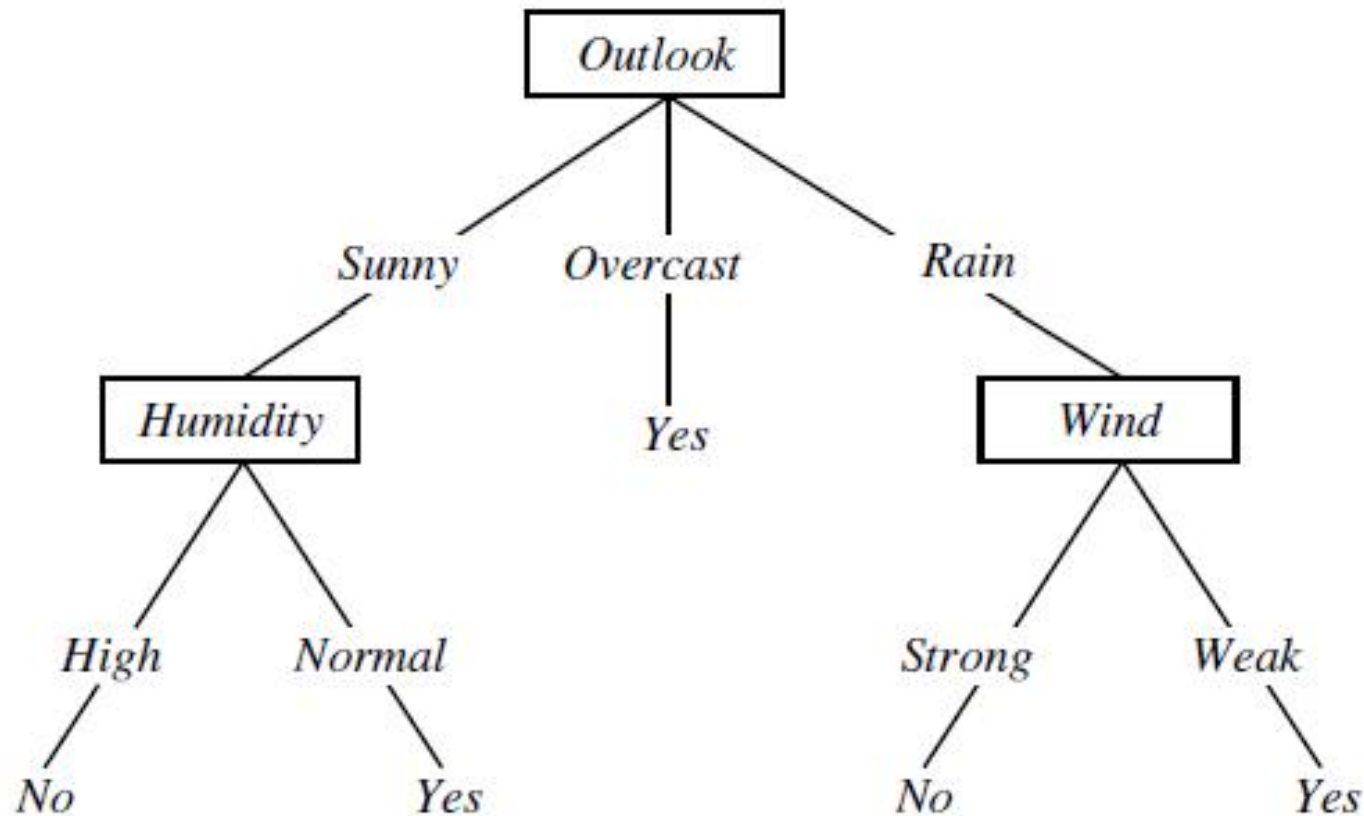
**All days here are sunny, so whats the point keeping
it !!!!!!!!!**

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D14	Rain	Mild	High	Strong	No

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D14	Rain	Mild	High	Strong	No

Its Wind !

Learned Tree for Prediction



Thank you BUET CSE!!!!!!



ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. *Target_attribute* is the attribute whose value is to be predicted by the tree. *Attributes* is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given *Examples*.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$
 - For each possible value, v_i , of A ,
 - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
 - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
 - If $Examples_{v_i}$ is empty
 - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
 - Else below this new branch add the subtree
ID3($Examples_{v_i}$, *Target_attribute*, $Attributes - \{A\}$)
- End
- Return *Root*

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. *Target_attribute* is the attribute whose value is to be predicted by the tree. *Attributes* is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given *Examples*.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$
 - For each possible value, v_i , of A ,
 - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
 - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
 - If $Examples_{v_i}$ is empty
 - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
 - Else below this new branch add the subtree
ID3($Examples_{v_i}$, *Target_attribute*, $Attributes - \{A\}$)
- End
- Return *Root*

ID3(*Examples*, *Target_attribute*)

Examples are the
predicted by the
decision tree. Return

- How to choose the BEST ???
- More importantly what is the BEST ????

- Create a *Root* node
- If all *Examples* are positive, Return the single node tree, with label = +
- If all *Examples* are negative, Return the single node tree, with label = -
- If *Attributes* is empty, Return the single node tree, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$

corresponding to the test $A = v_i$
Examples that have value v_i for A

add a leaf node with label = most common
Examples
ld the subtree
tribute, $Attributes - \{A\}$)



Entropy

- S is a sample of training examples
- p_{\oplus} is the proportion of positive examples in S
- p_{\ominus} is the proportion of negative examples in S
- Entropy measures the impurity of S

$$\textit{Entropy}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy

- S is a sample of training examples
- p_{\oplus} is the proportion of positive examples in S
- p_{\ominus} is the proportion of negative examples in S
- Entropy measures the impurity of S

$$\text{Entropy}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

- Entropy Is **Zero (Minimum)**, When all the examples belong to **same class!!!**
- Entropy Is **1(Maximum)**, When there are **equal** number of positive and negative examples!

Entropy

- Entropy Is Zero (Minimum), When all the examples belong to same class!!!
- Entropy Is 1(Maximum), When there are equal number of positive and negative examples!

To illustrate, suppose S is a collection of 14 examples of some boolean concept, including 9 positive and 5 negative examples (we adopt the notation $[9+, 5-]$ to summarize such a sample of data). Then the entropy of S relative to this boolean classification is

$$\begin{aligned} \text{Entropy}([9+, 5-]) &= -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ &= 0.940 \end{aligned} \tag{3.2}$$

$$\text{Entropy}(S) \equiv \sum_{i=1}^c -p_i \log_2 p_i$$

Information Gain

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Values(Wind) = Weak, Strong$$

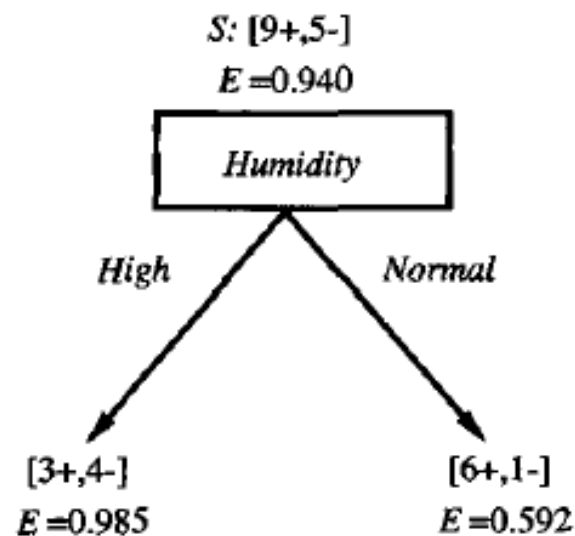
$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

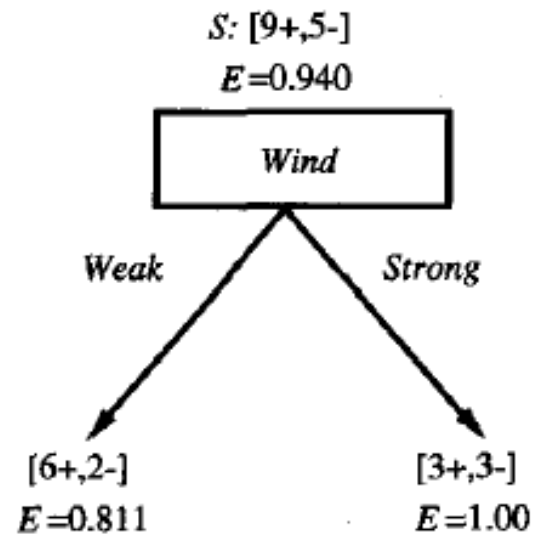
$$S_{Strong} \leftarrow [3+, 3-]$$

$$\begin{aligned} Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\ &= Entropy(S) - (8/14) Entropy(S_{Weak}) \\ &\quad - (6/14) Entropy(S_{Strong}) \\ &= 0.940 - (8/14)0.811 - (6/14)1.00 \\ &= 0.048 \end{aligned}$$

Which attribute is the best classifier?



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$



$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$

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

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

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

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

$$Gain(S, Temperature) = 0.029$$

Measuring Performance

		Predicted class		Total instances
		+	−	
Actual class	+	TP	FN	P
	−	FP	TN	N

TP: true positives

The number of positive instances that are classified as positive

FP: false positives

The number of negative instances that are classified as positive

FN: false negatives

The number of positive instances that are classified as negative

TN: true negatives

The number of negative instances that are classified as negative

$P = TP + FN$

The total number of positive instances

$N = FP + TN$

The total number of negative instances

**Why Reduce Entropy
???**



**How & Why am I
preferring shorter
trees ?**



Does it take me to the
global optimum ???



Download from
[presentation.com](https://www.presentation.com)



CC BY
You can download this

Why didn't I just
classify over the
attribute "Day" ???



Thanks to **Muhammad Ali
Nayeem** (মুহাম্মদ আলী নাইম)
for providing the basic slide