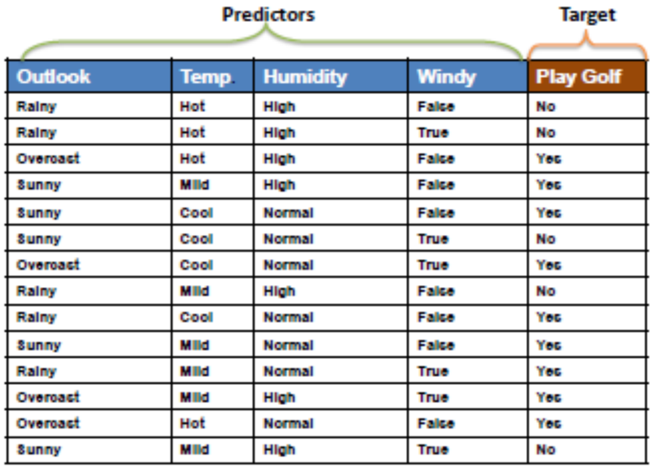
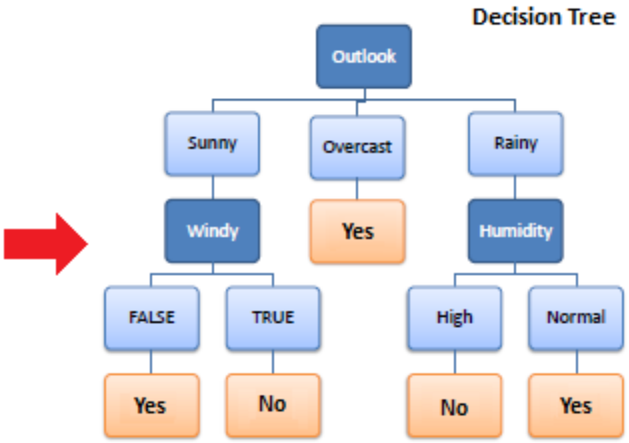
Decision Tree - Classification

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.



Predictors				Target
Outlook	**Temp.**	**Humidity**	**Windy**	**Play Golf**
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

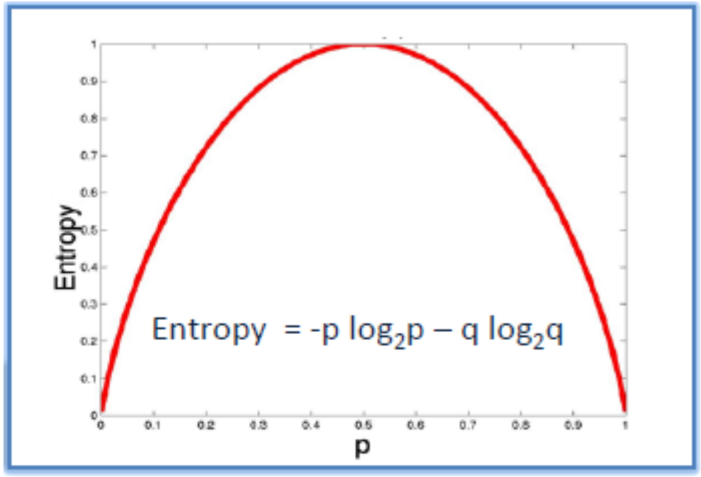


Algorithm

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

$$\log_2 p = \log_{10} p / \log_{10} 2$$

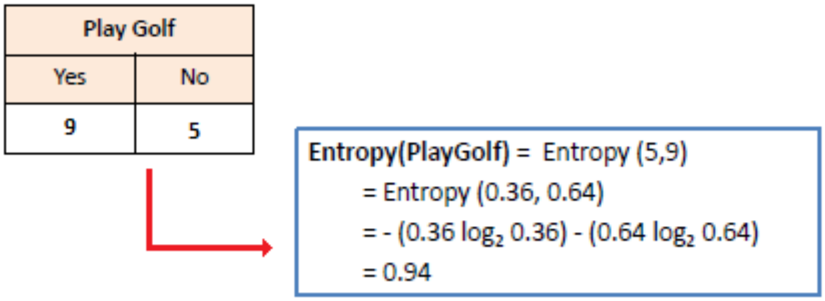
$$= -0.301/0.301 = -1$$

$$= -0.5*-1 - 0.5*-1 = 0.5 + 0.5 = 1$$

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

a) Entropy using the frequency table of one attribute:

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

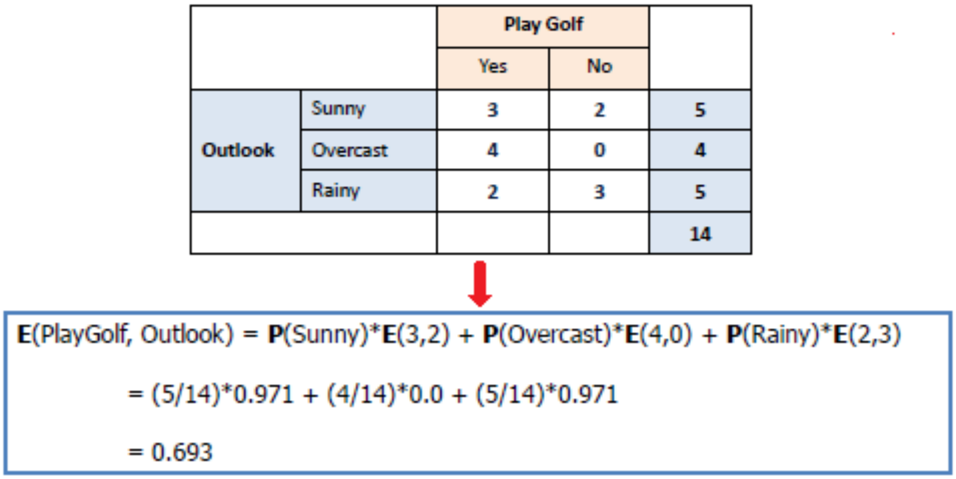


Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

b) Entropy using the frequency table of two attributes:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$



		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information Gain

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

Step 1: Calculate entropy of the target.

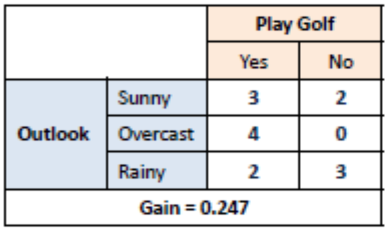
Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

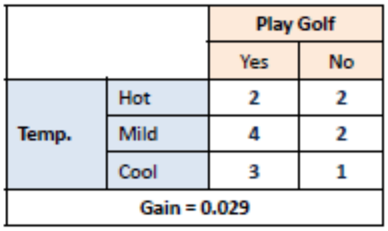
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

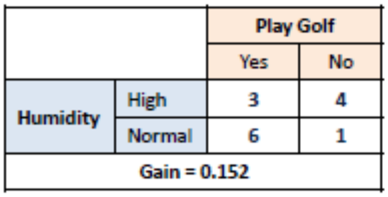
Step 2: The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.



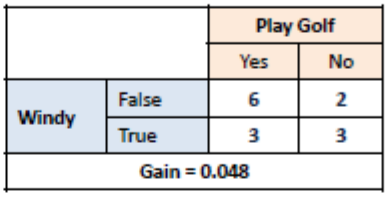
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

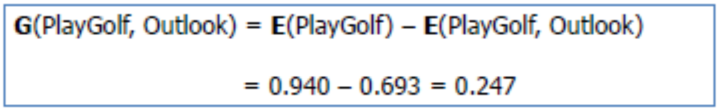


		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			



		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

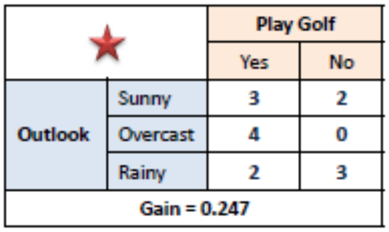
$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$



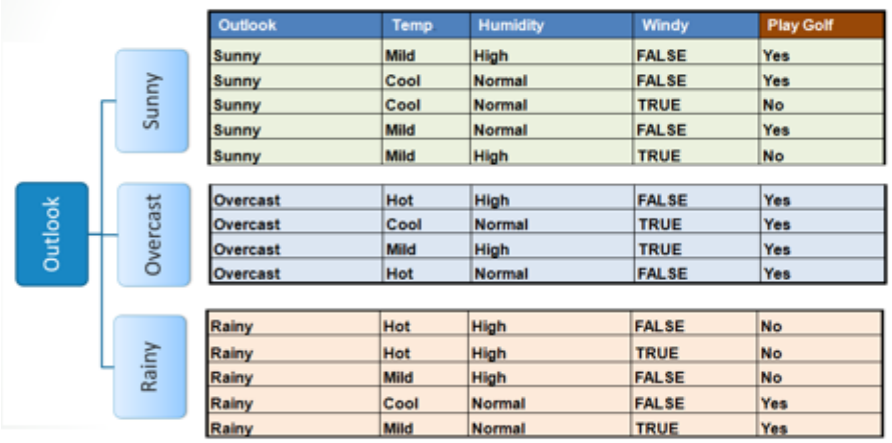
G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

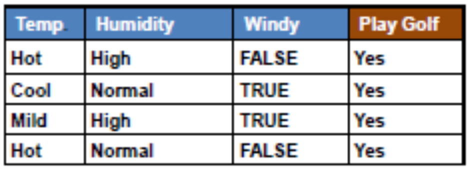


★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

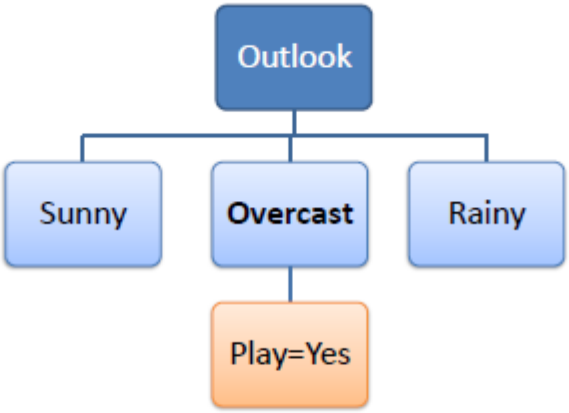


Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

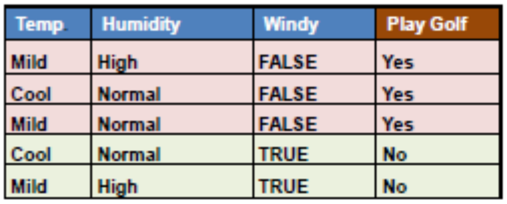
Step 4a: A branch with entropy of 0 is a leaf node.



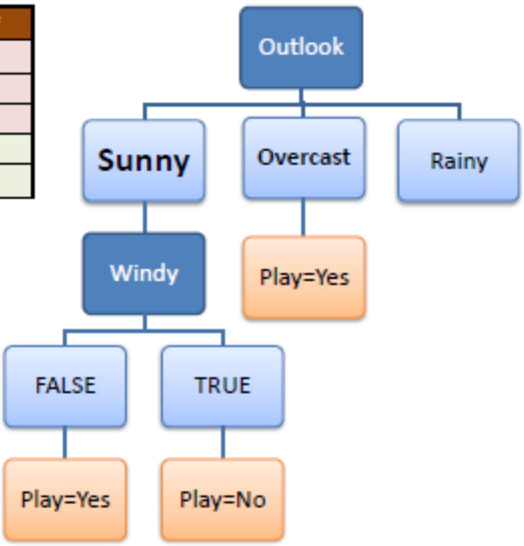
Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Step 4b: A branch with entropy more than 0 needs further splitting.



Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Decision Tree to Decision Rules

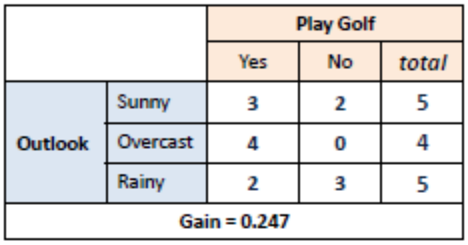
A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one.

Decision Tree - Super Attributes

The information gain equation, G(T,X) is biased toward attributes that have a large number of values over attributes that have a smaller number of values. These '**Super Attributes**' will easily be selected as the root, resulted in a broad tree that classifies perfectly but performs poorly on unseen instances. We can penalize attributes with large numbers of values by using an alternative method for attribute selection, referred to as **Gain Ratio**.

$$\mathbf{GainRatio}(T,X) = \frac{\mathbf{Gain}(T,X)}{\mathbf{SplitInformation}(T,X)}$$

$$Split(T,X) = -\sum_{c \in A} P(c)\log_2 P(c)$$



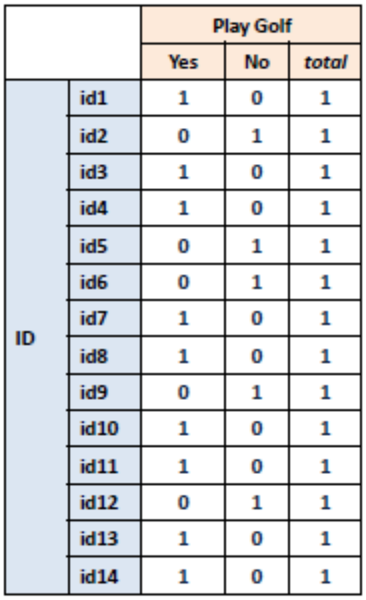
		Play Golf		
		Yes	No	*total*
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
Gain = 0.247				

Split (Play,Outlook) = - $(5/14*\log_2(5/14) + 4/14*\log_2(4/15) + 5/14*\log_2(5/14))$
= 1.577

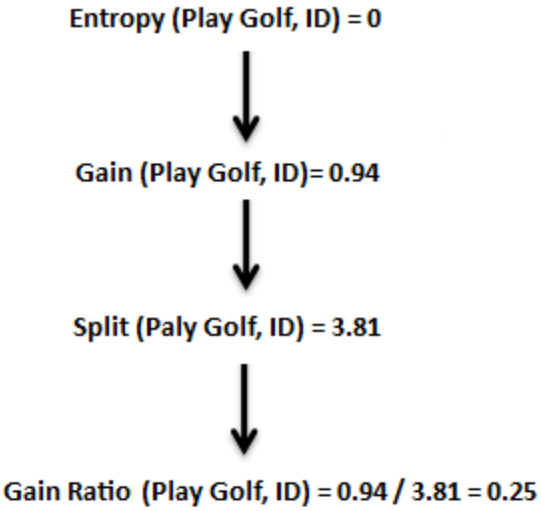
Gain Ratio (Play,Outlook) = 0.247/1.577 = **0.156**

Example:

The following example shows a frequency table between the target (Play Golf) and the ID attribute which has a unique value for each record of the dataset.



		Play Golf		
		Yes	No	*total*
ID	id1	1	0	1
	id2	0	1	1
	id3	1	0	1
	id4	1	0	1
	id5	0	1	1
	id6	0	1	1
	id7	1	0	1
	id8	1	0	1
	id9	0	1	1
	id10	1	0	1
	id11	1	0	1
	id12	0	1	1
	id13	1	0	1
	id14	1	0	1



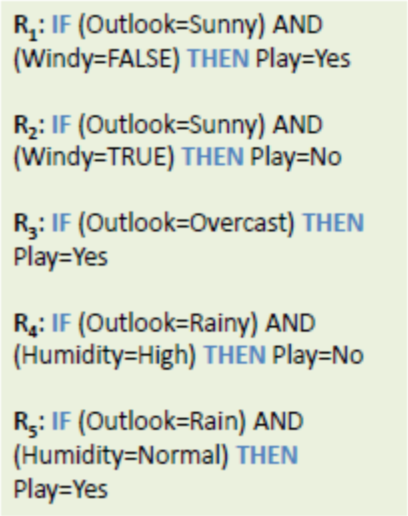
Entropy (Play Golf, ID) = 0

Gain (Play Golf, ID)= 0.94

Split (Paly Golf, ID) = 3.81

Gain Ratio (Play Golf, ID) = 0.94 / 3.81 = 0.25

The information gain for ID is maximum (0.94) without using the split information. However, with the adjustment the information gain dropped to 0.25.



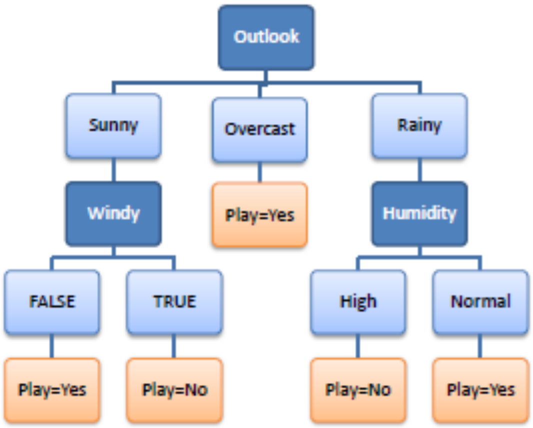
$\mathbf{R_1}$: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

$\mathbf{R_2}$: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

$\mathbf{R_3}$: IF (Outlook=Overcast) THEN Play=Yes

$\mathbf{R_4}$: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

$\mathbf{R_5}$: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



Source : https://www.saedsayad.com/