A decision tree is a structure that represents a procedure for classifying objects based on their attributes. Each object is represented as a set of attribute/value pairs and a classification. For example, a set of medical symptoms might be represented as follows:

<table>
<thead>
<tr>
<th>Cough</th>
<th>Fever</th>
<th>Weight</th>
<th>Pain</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>no</td>
<td>yes</td>
<td>normal</td>
<td>flu</td>
</tr>
<tr>
<td>Fred</td>
<td>no</td>
<td>yes</td>
<td>normal</td>
<td>appendicitis</td>
</tr>
<tr>
<td>Julie</td>
<td>yes</td>
<td>yes</td>
<td>skinny</td>
<td>flu</td>
</tr>
<tr>
<td>Elvis</td>
<td>yes</td>
<td>no</td>
<td>obese</td>
<td>heart disease</td>
</tr>
</tbody>
</table>

The system is given a set of training instances along with their correct classifications and develops a decision tree based on these examples.

A decision tree is a structure that represents a procedure for classifying objects based on their attributes.
Learning of Decision Trees

The ID3 learning algorithm (J. Ross Quinlan) builds a decision tree from a fixed set of examples. The resulting tree is used to classify future samples.

Examples are described by a set of attributes (common for all examples). Each attribute has a fixed set of values. One of the attributes is the target attribute, whose value is to be determined by the decision tree. Thus, given that we know the values of the remaining attributes for a particular sample, the question to be answered is: what would be the value of the target attribute?

The advantage of learning a decision tree is that a program, rather than a knowledge engineer, elicits knowledge from an expert.

Data Description

The sample data used by ID3 has certain requirements, which are:

- Attribute-value description - the same attributes must describe each example and have a fixed number of values.
- Predefined classes - an example's attributes must already be defined, that is, they are not learned by ID3.
- Discrete classes - classes must be sharply delineated.
- Sufficient examples – there must be enough test cases to distinguish valid patterns from random occurrences

Note that

- If a crucial attribute is not represented, then no decision tree will be able to learn the concept.
- If two training instances have the same representation but belong to different classes, then the attribute set is said to be inadequate. It is impossible for the decision tree to distinguish the instances.

Decision tree nodes

Decision trees have two types of nodes:

- Leaf nodes, also called classification nodes, correspond to the values of the target attribute.
- Decision nodes – at these nodes we test the value of a specified attribute and branch according to that value. Thus a decision node will have as many branches as the number of possible values of the attribute tested at that node.
Basic ID3 algorithm:

Given: A training set $S$, a set of attributes $A$, target attribute $A^T$  
Let $A$ have values $A^1, A^2, \ldots, A^n$  

Initialize $N = \text{root}$, $E = \text{set of all training examples}$  
Let $E$ be the set of examples considered at a given node $N$  
If all examples from $E$ have the same value of the target attribute, i.e. they belong to the same class  
then label the leaf with that class  
else  
select the “best” decision attribute $A$ for the node $N$ with values $v_1, v_2, \ldots, v_k$  
divide the training set $E$ into $E_1, \ldots, E_k$ according to values $v_1, \ldots, v_k$  
recursively build subtrees $T_1, \ldots, T_k$ for $E_1, \ldots, E_n$  

Which attribute is best?

A statistical property, called information gain, is used.  
Gain measures how well a given attribute separates training examples into targeted classes. The one with the highest information (information being the most useful for classification) is selected.  
In order to define gain, we first borrow an idea from information theory called entropy.  
Entropy measures the amount of information in an attribute.  
Given a collection $S$ of $c$ outcomes  

$$\text{Entropy}(S) = \sum -p(I) \log_2 p(I)$$  
where $p(I)$ is the proportion of $S$ belonging to class $I$.  
$\Sigma$ is over $c$.  
$\log_2$ is log base 2.
Gain(S, A) is information gain of example set S on attribute A is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum (|S_v| / |S| \times \text{Entropy}(S_v))$$

Where:

$\sum$ is each value v of all possible values of attribute A

$S_v$ = subset of S for which attribute A has value v

$|S_v|$ = number of elements in $S_v$

$|S|$ = number of elements in S

Example 1

If S is a collection of 14 examples with 9 YES and 5 NO examples then

$$\text{Entropy}(S) = -(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0.940$$

Notice entropy is 0 if all members of S belong to the same class (the data is perfectly classified). The range of entropy is 0 ("perfectly classified") to 1 ("totally random").

Example 2

Suppose S is a set of 14 examples in which one of the attributes is wind speed.

The values of Wind can be Weak or Strong.

The classification of these 14 examples are 9 YES and 5 NO.

For attribute Wind, suppose there are 8 occurrences of Wind = Weak and 6 occurrences of Wind = Strong.

For Wind = Weak, 6 of the examples are YES and 2 are NO. For Wind = Strong, 3 are YES and 3 are NO.

Therefore

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - (8/14) \times \text{Entropy}(S_{\text{Weak}}) - (6/14) \times \text{Entropy}(S_{\text{Strong}})$$

$$= 0.940 - (8/14) \times 0.811 - (6/14) \times 1.00$$

$$= 0.048$$

$$\text{Entropy}(S_{\text{Weak}}) = -(6/8) \times \log_2 (6/8) - (2/8) \times \log_2 (2/8) = 0.811$$

$$\text{Entropy}(S_{\text{Strong}}) = -(3/6) \times \log_2 (3/6) - (3/6) \times \log_2 (3/6) = 1.00$$

For each attribute, the gain is calculated and the highest gain is used in the decision node.
Example of ID3

Suppose we want ID3 to decide whether the weather is amenable to playing baseball. Over the course of 2 weeks, data is collected to help ID3 build a decision tree (see table 1).

The target classification is "should we play baseball?" which can be yes or no.

The weather attributes are outlook, temperature, humidity, and wind speed. They can have the following values:
- outlook = { sunny, overcast, rain }
- temperature = {hot, mild, cool }
- humidity = { high, normal }
- wind = {weak, strong }

Examples of set S are:

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play ball</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1
We need to find which attribute will be the root node in our decision tree. The gain is calculated for all four attributes:

Gain(S, Outlook) = 0.246
Gain(S, Temperature) = 0.029
Gain(S, Humidity) = 0.151
Gain(S, Wind) = 0.048 (calculated in example 2)

Outlook attribute has the highest gain, therefore it is used as the decision attribute in the root node.

Since Outlook has three possible values, the root node has three branches (sunny, overcast, rain). The next question is "what attribute should be tested at the Sunny branch node?" Since we have used Outlook at the root, we only decide on the remaining three attributes: Humidity, Temperature, or Wind.

\[ S_{\text{sunny}} = \{D1, D2, D8, D9, D11\} = 5 \text{ examples from table 1 with outlook = sunny} \]

Gain(S\textsubscript{sunny}, Humidity) = 0.970
Gain(S\textsubscript{sunny}, Temperature) = 0.570
Gain(S\textsubscript{sunny}, Wind) = 0.019

Humidity has the highest gain; therefore, it is used as the decision node. This process goes on until all data is classified perfectly or we run out of attributes.
The decision tree can also be expressed in rule format:

IF outlook = sunny AND humidity = high THEN playball = no
IF outlook = rain AND humidity = high THEN playball = no
IF outlook = rain AND wind = strong THEN playball = yes
IF outlook = overcast THEN playball = yes
IF outlook = rain AND wind = weak THEN playball = yes

ID3 has been incorporated in a number of commercial rule-induction packages. Some specific applications include medical diagnosis, credit risk assessment of loan applications, equipment malfunctions by their cause, classification of soybean diseases, and web search classification.

Source Douglas D. Dankel II, University of Florida

http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm