Consider this binary classification data set:

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</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>
We can describe this data set with the following decision tree:
Decision Trees

All observations in the data set are perfectly described by the tree.

**Question:** How do we build such trees?
Entrophy

The key to decision tree induction is the notion of entropy,

\[
\text{Entropy} \equiv \text{measure of randomness}
\]

**Observation**: Entropy is at its maximum if we have a 50%-50% split among the positive and negative examples.

**Observation**: Entropy is zero if we have all positive or all negative examples.
We can apply entropy to measure the “randomness” of our data set.

Let

\[ D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_l, y_l)\} \subseteq A^n \times \{+1, -1\} \]

and

\[ l_+ = |\{(\vec{x}, y) | (\vec{x}, y) \land y = +1\}| \]
\[ l_- = |\{(\vec{x}, y) | (\vec{x}, y) \land y = -1\}| \]

then

\[ Entropy(D) = -\frac{l_+}{l} \log_2\left(\frac{l_+}{l}\right) - \frac{l_-}{l} \log_2\left(\frac{l_-}{l}\right) \]

Now let \( p_+ = \frac{l_+}{l} \) and \( p_- = \frac{l_-}{l} \) then

\[
Entropy(D) = -p_+ \log_2(p_+) - p_- \log_2(p_-)
\]
Def: We say that an attribute is *informative* if, when the training set is split according to its attribute values, the overall entropy in the training data is reduced.

Example: Consider the attribute $A_k = \{v_1, v_2, v_3\}$ then the split $D_{v_i}$ of $D$ only contains instances that have value $v_i$ of attribute $A_k$,

$$D_{v_i} = \{(x, y) \mid x_k = v_i\}$$

We can now split the data set $D$ according to the values of attribute $A_k$,

If $E_{A_k} < E_D$ then attribute $A_k$ is informative.
Information Gain

Rather than using the arithmetic mean we use the weighted mean,

\[ \text{Entropy}(A_k) = \sum_{v_i \in A_k} \frac{|D_{v_i}|}{|D|} \text{Entropy}(D_{v_i}) \]

Formally we define information gain as,

\[ \text{Gain}(D, A_k) = \text{Entropy}(D) - \text{Entropy}(A_k) \]

or

\[ \text{Gain}(D, A_k) = \text{Entropy}(D) - \sum_{v_i \in A_k} \frac{|D_{v_i}|}{|D|} \text{Entropy}(D_{v_i}) \]

⇒ The larger the difference the more informative an attribute!
Information Gain

We can now use the gain to build a decision tree top-down (greedy heuristic).

Example: Consider our tennis data set with

Wind = \{Weak, Strong\}

Then

\[ D = [9+, 5-] \]

\[ D_{\text{Weak}} = [6+, 2-] \]

\[ D_{\text{Strong}} = [3+, 3-] \]

Finally,

\[
\text{Gain}(D, \text{Wind}) = \text{Entropy}(D) - \sum_{v_i \in A_k} \frac{|D_{v_i}|}{|D|} \text{Entropy}(D_{v_i})
\]

\[
= .94 - \frac{8}{14} \cdot .811 - \frac{6}{14} \cdot 1
\]

\[
= .048
\]
Similarly, for Outlook, Humidity, and Temp,

\[
\begin{align*}
\text{Gain}(D, \text{Outlook}) &= .246 \\
\text{Gain}(D, \text{Humidity}) &= .151 \\
\text{Gain}(D, \text{Temp}) &= .029
\end{align*}
\]

⇒ This means the *Outlook* will become our root more.
Information Gain

Which attribute should be tested here?

\[ S_{\text{sunny}} = \{D1, D2, D8, D9, D11\} \]

\[
\text{Gain} (S_{\text{sunny}}, \text{Humidity}) = 0.970 - \left( \frac{3}{5} \right) 0.0 - \left( \frac{2}{5} \right) 0.0 = 0.970
\]

\[
\text{Gain} (S_{\text{sunny}}, \text{Temperature}) = 0.970 - \left( \frac{2}{5} \right) 0.0 - \left( \frac{2}{5} \right) 1.0 - \left( \frac{1}{5} \right) 0.0 = 0.570
\]

\[
\text{Gain} (S_{\text{sunny}}, \text{Wind}) = 0.970 - \left( \frac{2}{5} \right) 1.0 - \left( \frac{3}{5} \right) 0.918 = 0.019
\]
Information Gain

Decision Tree Induction

Basic Algorithm:
1. \( A \leftarrow \) the "best" decision attribute for a node \( N \).
2. Assign \( A \) as decision attribute for the node \( N \).
3. For each value of \( A \), create new descendant of the node \( N \).
4. Sort training examples to leaf nodes.
5. IF training examples perfectly classified, THEN STOP.
   ELSE iterate over new leaf nodes.