“My Weekend” Example

Imagine you only ever do one of the following four things for any weekend:

- go shopping
- watch a movie
- play tennis
- just stay in

What you do depends on three factors:

1. weather (windy, rainy or sunny)
2. how much money you have (rich or poor)
3. whether your parents are visiting (yes or no)

You say to yourself: if my parents are visiting, we'll go to the cinema. If they're not visiting and it's sunny, then I'll play tennis, and so on.

Suppose we have the following instances in our (training) dataset:

<table>
<thead>
<tr>
<th>Weekend (Example)</th>
<th>Weather</th>
<th>Parents</th>
<th>Money</th>
<th>Decision (Category)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>Sunny</td>
<td>Yes</td>
<td>Rich</td>
<td>Cinema</td>
</tr>
<tr>
<td>W2</td>
<td>Sunny</td>
<td>No</td>
<td>Rich</td>
<td>Tennis</td>
</tr>
<tr>
<td>W3</td>
<td>Windy</td>
<td>Yes</td>
<td>Rich</td>
<td>Cinema</td>
</tr>
<tr>
<td>W4</td>
<td>Rainy</td>
<td>Yes</td>
<td>Poor</td>
<td>Cinema</td>
</tr>
<tr>
<td>W5</td>
<td>Rainy</td>
<td>No</td>
<td>Rich</td>
<td>Stay in</td>
</tr>
<tr>
<td>W6</td>
<td>Rainy</td>
<td>Yes</td>
<td>Poor</td>
<td>Cinema</td>
</tr>
<tr>
<td>W7</td>
<td>Windy</td>
<td>No</td>
<td>Poor</td>
<td>Cinema</td>
</tr>
<tr>
<td>W8</td>
<td>Windy</td>
<td>No</td>
<td>Rich</td>
<td>Shopping</td>
</tr>
<tr>
<td>W9</td>
<td>Windy</td>
<td>Yes</td>
<td>Rich</td>
<td>Cinema</td>
</tr>
<tr>
<td>W10</td>
<td>Sunny</td>
<td>No</td>
<td>Rich</td>
<td>Tennis</td>
</tr>
</tbody>
</table>

Apply the ID3 algorithm on this dataset to induce a decision tree. Show all your work.
Solution

The first thing we need to do is work out which attribute will be put into the node at the top of our tree: either weather, parents or money. To do this, we need to calculate:

\[
\text{Entropy}(S) = -p_{\text{cinema}} \log_2(p_{\text{cinema}}) -p_{\text{tennis}} \log_2(p_{\text{tennis}}) - p_{\text{shopping}} \log_2(p_{\text{shopping}}) - p_{\text{stay in}} \log_2(p_{\text{stay in}})
\]

\[
= -(6/10) * \log_2(6/10) - (2/10) * \log_2(2/10) - (1/10) * \log_2(1/10) - (1/10) * \log_2(1/10)
\]

\[
= -(6/10) * -0.737 - (2/10) * -2.322 - (1/10) * -3.322 - (1/10) * -3.322
\]

\[
= 0.4422 + 0.4644 + 0.3322 + 0.3322 = 1.571
\]

and we need to determine the best of:

\[
\text{Gain}(S, \text{weather}) = 1.571 - (|S_{\text{sunny}}|/10)*\text{Entropy}(S_{\text{sunny}}) - (|S_{\text{windy}}|/10)*\text{Entropy}(S_{\text{windy}}) - (|S_{\text{rainy}}|/10)*\text{Entropy}(S_{\text{rainy}})
\]

\[
= 1.571 - (0.3)*\text{Entropy}(S_{\text{sunny}}) - (0.4)*\text{Entropy}(S_{\text{windy}}) - (0.3)*\text{Entropy}(S_{\text{rainy}})
\]

\[
= 1.571 - (0.3)*(0.918) - (0.4)*(0.81125) - (0.3)*(0.918) = 0.70
\]

\[
\text{Gain}(S, \text{parents}) = 1.571 - (|S_{\text{yes}}|/10)*\text{Entropy}(S_{\text{yes}}) - (|S_{\text{no}}|/10)*\text{Entropy}(S_{\text{no}})
\]

\[
= 1.571 - (0.5) * 0 - (0.5) * 1.922 = 1.571 - 0.961 = 0.61
\]

\[
\text{Gain}(S, \text{money}) = 1.571 - (|S_{\text{rich}}|/10)*\text{Entropy}(S_{\text{rich}}) - (|S_{\text{poor}}|/10)*\text{Entropy}(S_{\text{poor}})
\]

\[
= 1.571 - (0.7) * (1.842) - (0.3) * 0 = 1.571 - 1.2894 = 0.2816
\]

This means that the first node in the decision tree will be the **weather** attribute. As an exercise, convince yourself why this scored (slightly) higher than the parents attribute - remember what entropy means and look at the way information gain is calculated.

From the weather node, we draw a branch for the values that weather can take: sunny, windy and rainy:

Now we look at the first branch. \(S_{\text{sunny}} = \{W1, W2, W10\}\). This is not empty, so we do not put a default categorisation leaf node here. The categorisations of W1, W2 and W10 are Cinema, Tennis and Tennis respectively. As these are not all the same, we cannot put a categorisation leaf node here. Hence we put an attribute node here, which we will leave blank for the time being.

Looking at the second branch, \(S_{\text{windy}} = \{W3, W7, W8, W9\}\). Again, this is not empty, and they do not all belong to the same class, so we put an attribute node here, left blank for now. The same situation happens with the third branch, hence our amended tree looks like this:
Now we have to fill in the choice of attribute A, which we know cannot be weather, because we've already removed that from the list of attributes to use. So, we need to calculate the values for Gain($S_{sunny}$, parents) and Gain($S_{sunny}$, money). Firstly, Entropy($S_{sunny}$) = 0.918. Next, we set S to be $S_{sunny} = \{W1, W2, W10\}$ (and, for this part of the branch, we will ignore all the other examples). In effect, we are interested only in this part of the table:

<table>
<thead>
<tr>
<th>Weekend (Example)</th>
<th>Weather</th>
<th>Parents</th>
<th>Money</th>
<th>Decision (Category)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>Sunny</td>
<td>Yes</td>
<td>Rich</td>
<td>Cinema</td>
</tr>
<tr>
<td>W2</td>
<td>Sunny</td>
<td>No</td>
<td>Rich</td>
<td>Tennis</td>
</tr>
<tr>
<td>W10</td>
<td>Sunny</td>
<td>No</td>
<td>Rich</td>
<td>Tennis</td>
</tr>
</tbody>
</table>

Hence we can calculate:

\[
\text{Gain}(S_{sunny}, \text{parents}) = 0.918 - (\frac{|S_{yes}|}{|S|})\text{Entropy}(S_{yes}) - (\frac{|S_{no}|}{|S|})\text{Entropy}(S_{no}) \\
= 0.918 - (1/3)*0 - (2/3)*0 = 0.918
\]

\[
\text{Gain}(S_{sunny}, \text{money}) = 0.918 - (\frac{|S_{rich}|}{|S|})\text{Entropy}(S_{rich}) - (\frac{|S_{poor}|}{|S|})\text{Entropy}(S_{poor}) \\
= 0.918 - (3/3)*0.918 - (0/3)*0 = 0.918 - 0.918 = 0
\]

Notice that Entropy($S_{yes}$) and Entropy($S_{no}$) were both zero, because $S_{yes}$ contains examples which are all in the same category (cinema), and $S_{no}$ similarly contains examples which are all in the same category (tennis). This should make it more obvious why we use information gain to choose attributes to put in nodes.

Given our calculations, attribute A should be taken as \textbf{parents}. The two values from parents are yes and no, and we will draw a branch from the node for each of these. Remembering that we replaced the set S by the set $S_{sunny}$, looking at $S_{yes}$, we see that the only example of this is W1. Hence, the branch for yes stops at a categorisation leaf, with the category being Cinema. Also, $S_{no}$ contains W2 and W10, but these are in the same category (Tennis). Hence the branch for no ends here at a categorisation leaf. Hence our upgraded tree looks like this:
And the final tree looks like:

Weather = Sunny
| Parents = Yes: Cinema
| Parents = No: Tennis
Weather = Windy
| Parents = Yes: Cinema
| Parents = No
| Money = Rich: Shopping
| Money = Poor: Cinema
Weather = Rainy
| Parents = Yes: Cinema
| Parents = No: Stay_in
Pruning

“Pruning a decision node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with that node.” (p. 69 in Tom Mitchell’s textbook)

(0) Original fully grown tree

(1) Choose the subtree under ‘Money’ to prune, because that’s the lowest subtree. Doing so results in the following tree (where the ‘most common classification’ was a tie between cinema and shopping (1 instance each in the subtree, W7 and W8; cinema was selected because it was more common in the entire dataset).

But since the classification under the ‘Parent’ node above becomes the same for both of its branches, the tree is further modified as:
(2) Next choose the subtree under ‘Parent’ under Weather=Rainy.
There are 3 instances which fall in this subtree (W4, W5, W6). Replace the subtree with a classification ‘cinema’ because that's the majority classification of the 3 instances.

[Note that the other subtree, the one under Weather=Sunny, was a tie with this subtree, in that both subtrees had the same number of instances (3), as well as the same entropy value (2 out of 3 instances had the same classification). In this example the subtree under Weather=Rainy was chosen arbitrarily.]

(3) Next choose the subtree under ‘Parent’ under Weather=Sunny.
There are 3 instances which fall in this subtree (W1, W2, W10). Replace the subtree with a classification ‘tennis’ because that's the majority classification of the 3 instances.