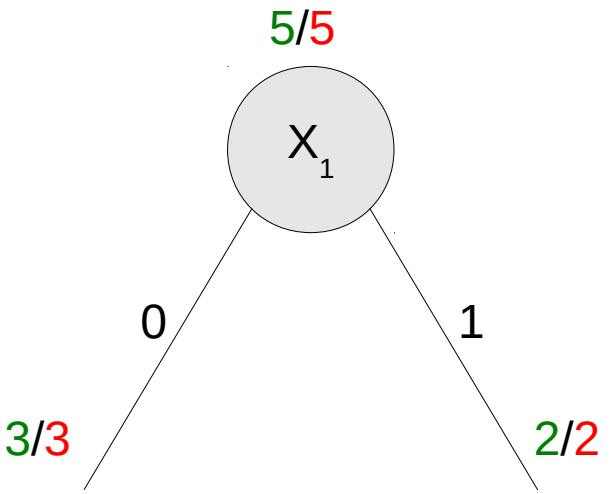
The ID3 Decision Tree Learning Algorithm

	Y	X*	X_1				
M_1	L	1	1				
M_2	L	1	1				
M_3	L	1	0				
M_4	L	1	0				
M_5	L	1	0				
M_6	S	0	0				
M_7	S	0	0				
M_8	S	0	0				
M_9	S	0	1				
M_{10}	S	0	1				

A possible training set for a spam/legitimate email classifier:

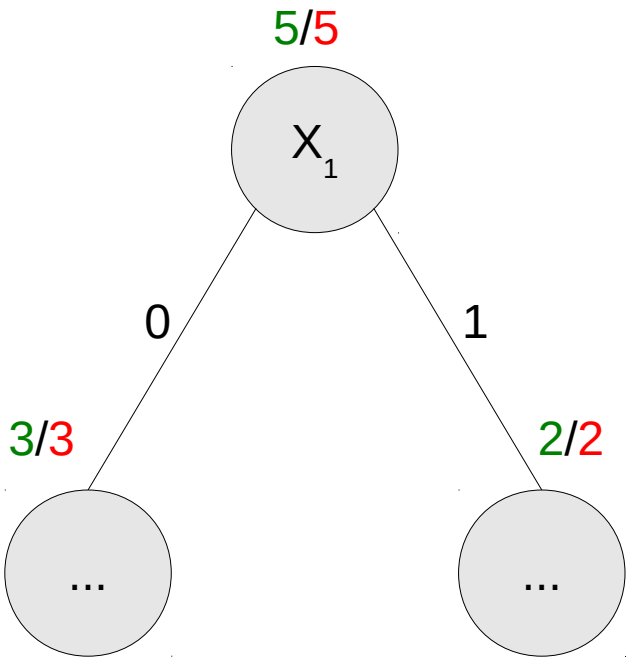
- ten examples (M_1-M_{10}): five legitimate (L) and five spam (S) emails
- two boolean attributes X* and X_1, denoting the occurrence of two given words in an email

	Y	X*	X_1				
M_1	L	1	1				
M_2	L	1	1				
M_3	L	1	0				
M_4	L	1	0				
M_5	L	1	0				
M_6	S	0	0				
M_7	S	0	0				
M_8	S	0	0				
M_9	S	0	1				
M_{10}	S	0	1				



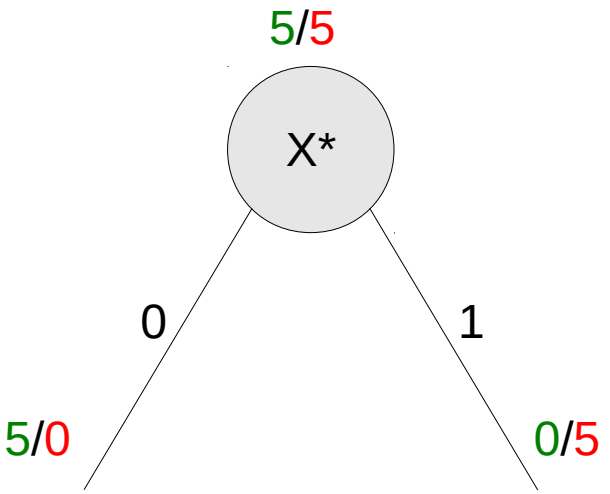
If X_1 is chosen for the root node of the decision tree, the 5/5 examples in the training set are split according to its values into 3/3 and 2/2. Intuitively, this means that **X_1 has no discriminant capability**: the corresponding word has the same probability of occurring both in legitimate and in spam emails...

	Y	X*	X_1				
M_1	L	1	1				
M_2	L	1	1				
M_3	L	1	0				
M_4	L	1	0				
M_5	L	1	0				
M_6	S	0	0				
M_7	S	0	0				
M_8	S	0	0				
M_9	S	0	1				
M_{10}	S	0	1				



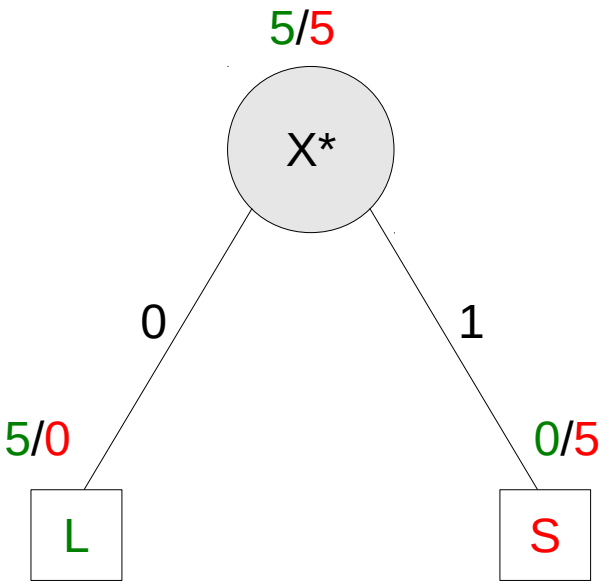
... therefore, it is not possible to stop the construction of the DT by inserting two leaf nodes, since the resulting DT would misclassify some training examples. For the DT to be consistent with the above training set, it is necessary to add two new nodes.

	Y	X*	X_1				
M_1	L	1	1				
M_2	L	1	1				
M_3	L	1	0				
M_4	L	1	0				
M_5	L	1	0				
M_6	S	0	0				
M_7	S	0	0				
M_8	S	0	0				
M_9	S	0	1				
M_{10}	S	0	1				



If X* is chosen for the root of the DT instead, training examples are split according to its values into 5/0 and 0/5...

	Y	X*	X_1				
M_1	L	1	1				
M_2	L	1	1				
M_3	L	1	0				
M_4	L	1	0				
M_5	L	1	0				
M_6	S	0	0				
M_7	S	0	0				
M_8	S	0	0				
M_9	S	0	1				
M_{10}	S	0	1				

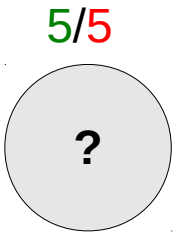


... this allows one to obtain the smallest possible DT, consistent with the training set, by inserting two leaves just below the root node.
This means that **X* is a perfectly discriminant attribute.**

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1

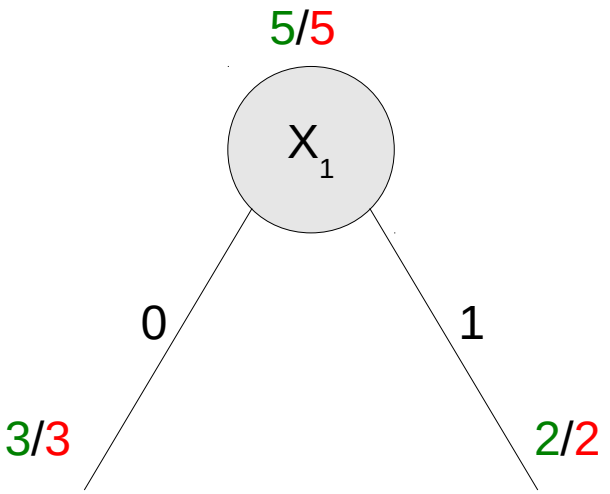
In practice, perfectly discriminant attributes are very rare.
In the figure above, the same emails are represented using five attributes (words), none of which is perfectly discriminant.
In the following, a widely used learning algorithm for DTs, named ID3, is sketched.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



The learning algorithm starts by constructing the root node of the DT. The corresponding attribute has to be chosen from all the available ones. ID3 makes this choice by looking for the **most discriminant** attribute, i.e., the one whose values split the training examples as much as possible according to their class. This favours the construction of a **small** and **consistent** DT.

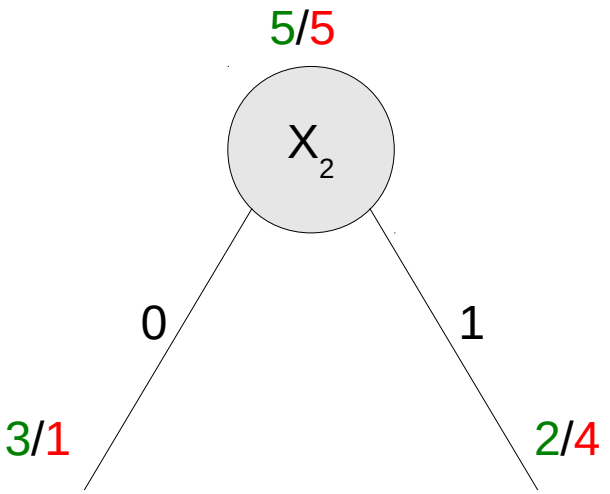
	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



Let us consider each of the possible attributes.

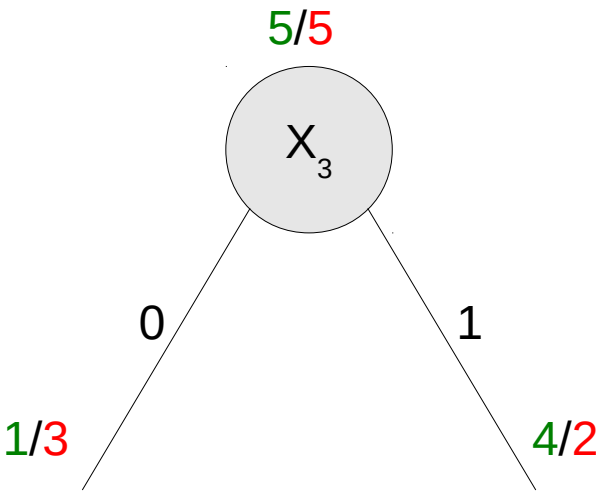
We have already seen that X_1 is not a good choice: it has no discriminant capability.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



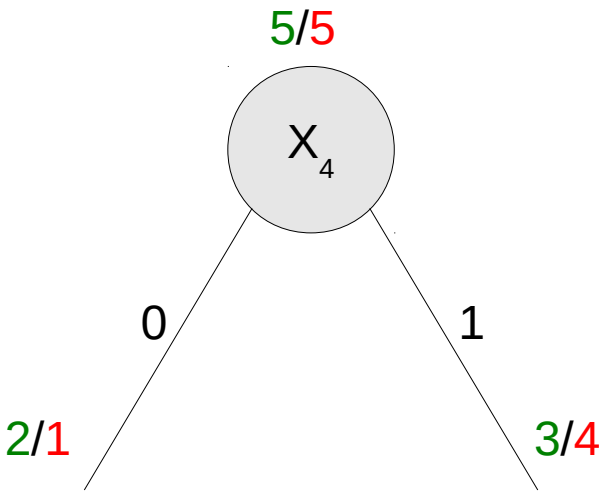
X_2 has a better discriminant capability: for each of its values, most of the corresponding training examples belong to only one of the classes (legitimate when X_2=0, spam when X_2=1).

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



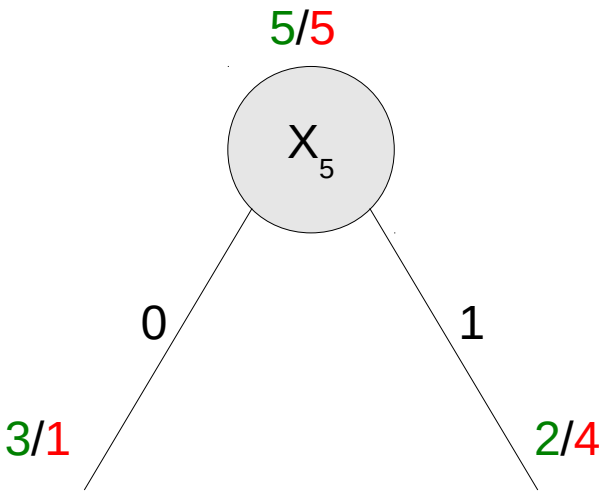
X_3 has the same discriminant capability as X_2, since it produces the same **distribution** of spam and legitimate training emails according to its outputs (the class proportions are switched with respect to X_2, but this is not relevant to the discriminant capability).

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



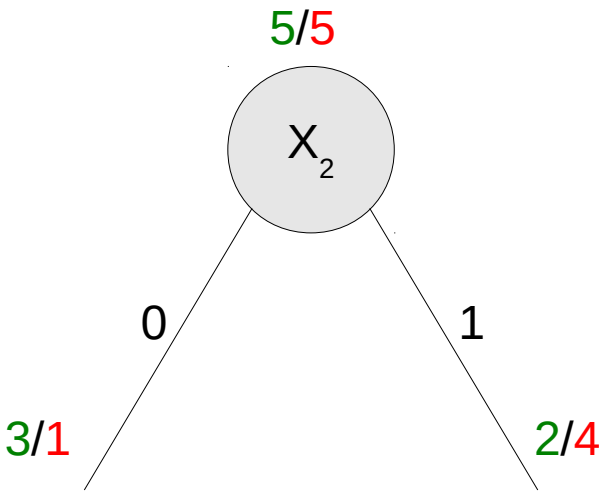
Intuitively, X_4 has a lower discriminant capability than X_2 and X_3 (but still better than X_1), since it produces a **more balanced** distribution of spam and legitimate training emails according to its outputs.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



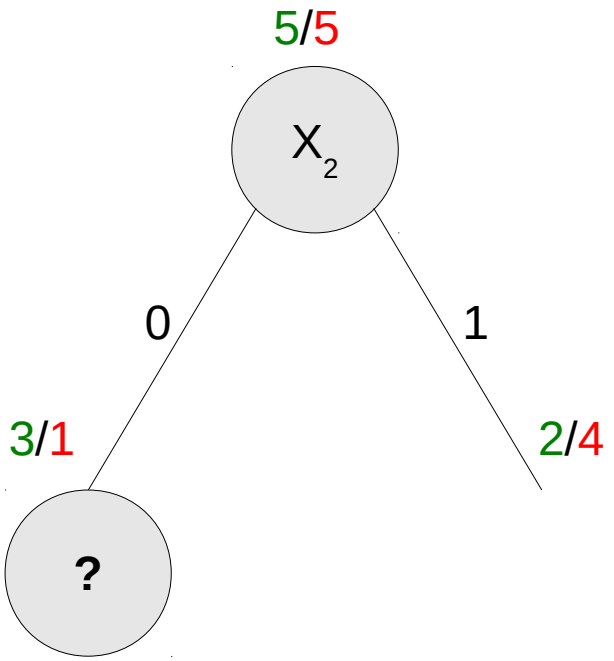
Finally, X_5 has the same discriminant capability as X_2, and X_3.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



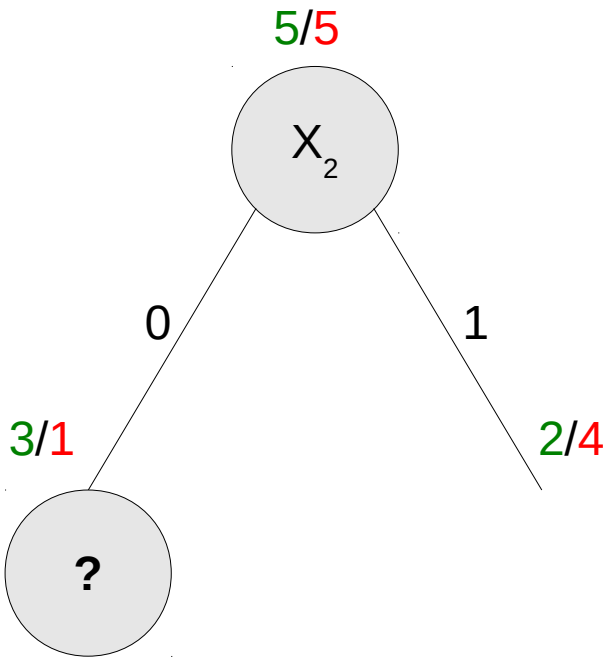
Among the three attributes exhibiting the highest discriminant capability, assume that X_2 is chosen (e.g., randomly).

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



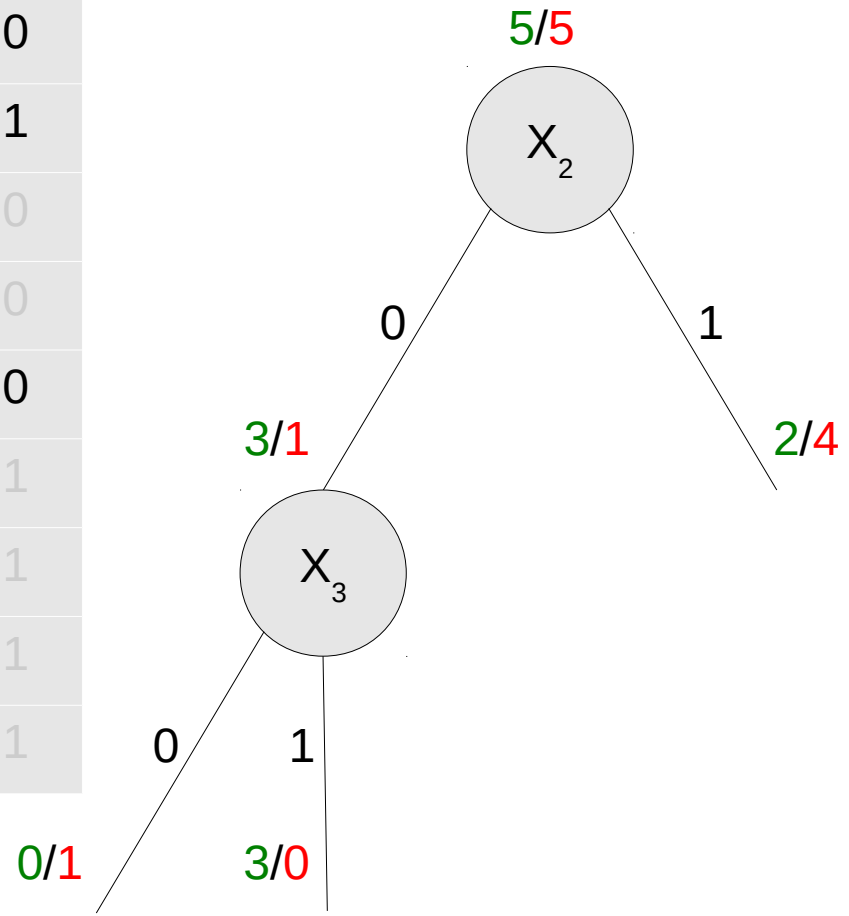
Now the ID3 algorithm proceeds **recursively** by building a sub-tree for each value of X_2. Assuming that the value X_2=0 is considered first, since there are both legitimate and spam emails for which X_2=0, the root of the sub-tree must be a node and not a leaf. The attribute must be chosen among the ones not present in the same path from the root, i.e.: X_1, X_3, X_4 and X_5.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



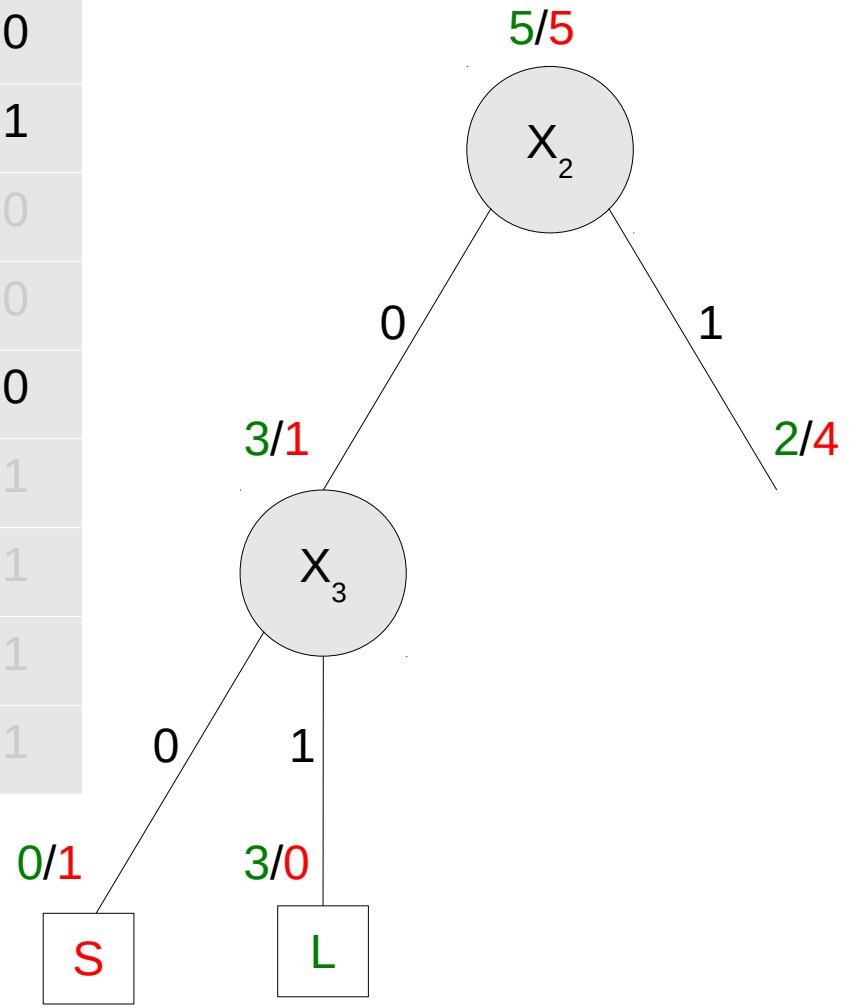
To find the most discriminant attribute among X_1, X_3, X_4 and X_5, only the training examples that reach the considered node must be considered (since the goal is to find a consistent tree), i.e., the four emails (tree legitimate and one spam email) highlighted above.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



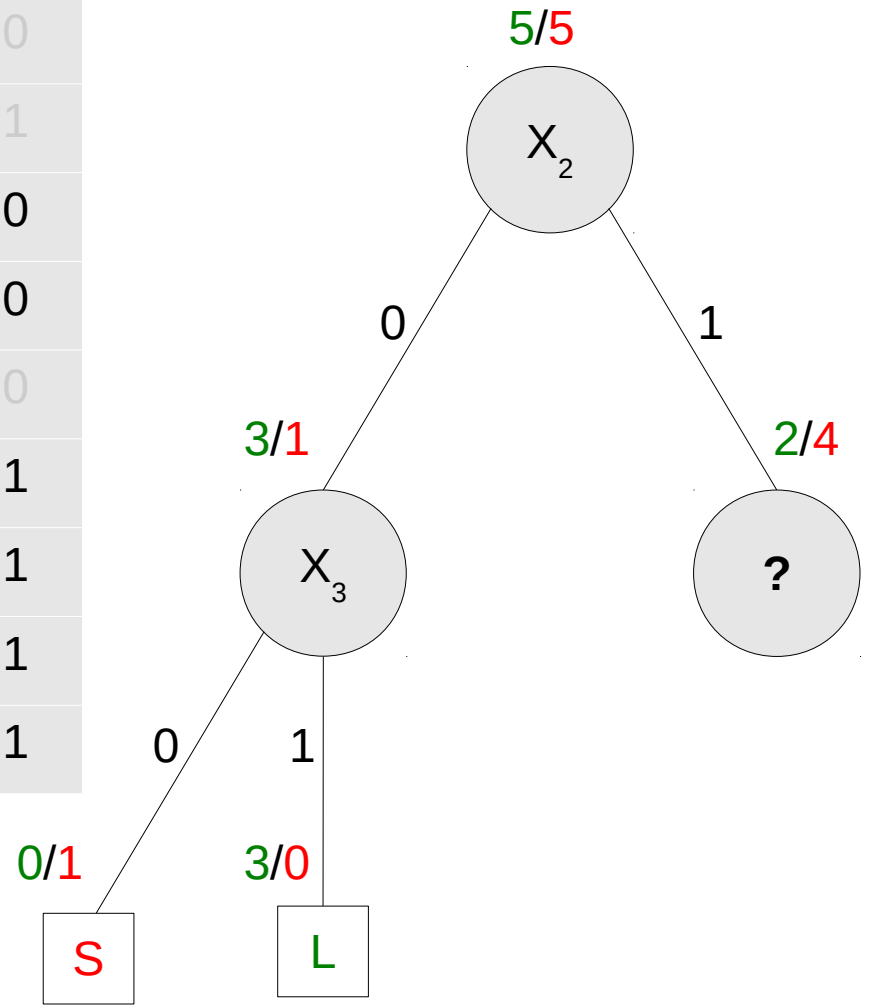
It is easy to see that X_3 is a perfectly discriminant attribute **for the four training examples at hand**, and that the other three attributes have a lower discriminant capability. Accordingly, X_3 must be chosen for this node.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



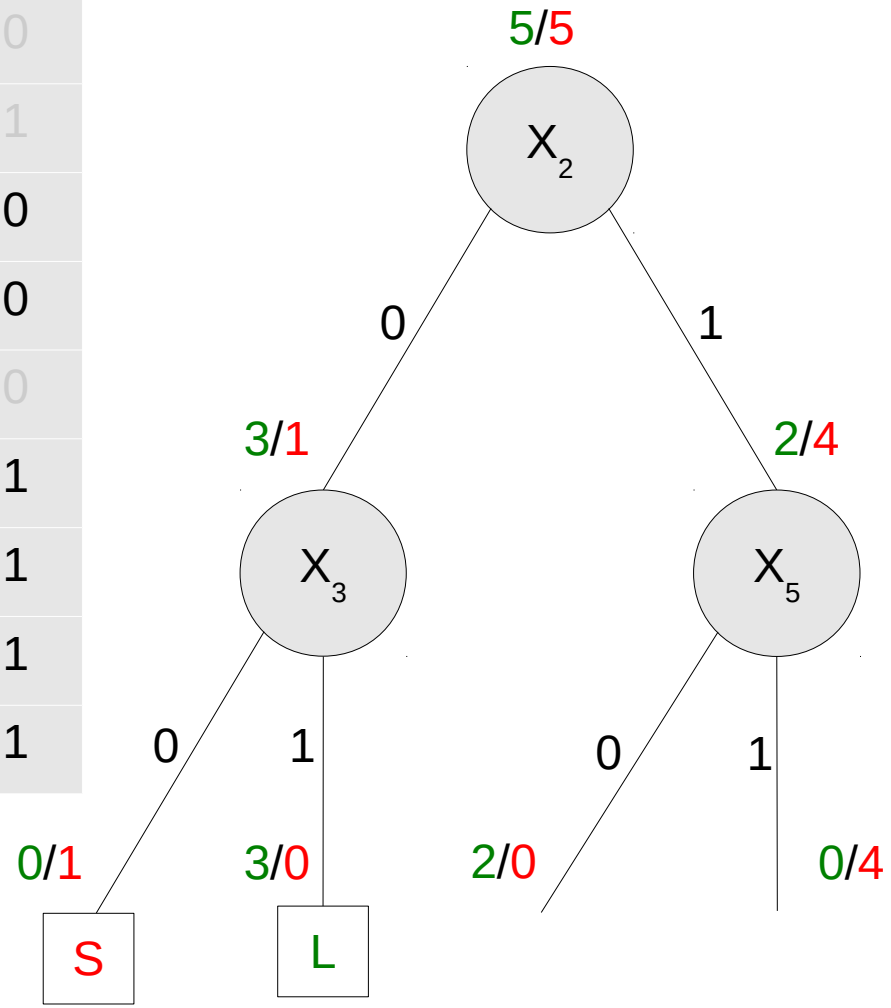
It is also easy to see that the two recursive calls to the ID3 procedure construct two leaves with the class labels shown above.
Then recursion proceeds with the right child of the root node...

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



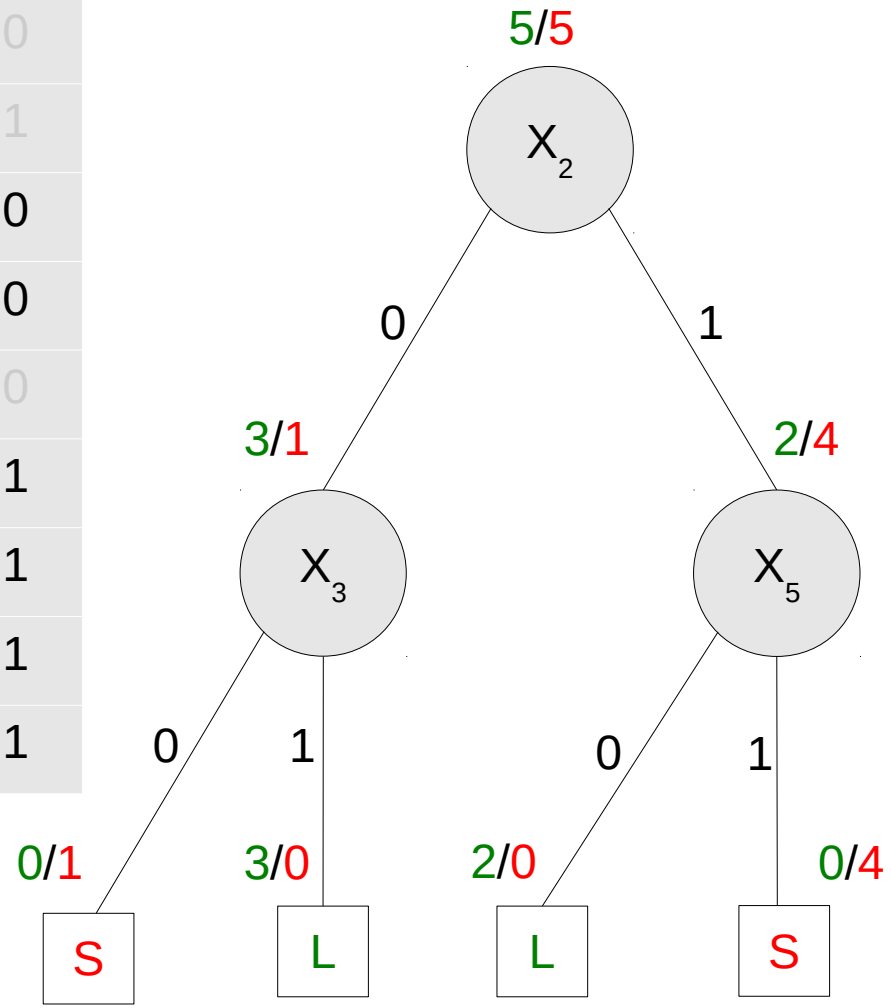
... and a sub-tree has to be built according to the six training examples highlighted above (two legitimate and four spam emails) corresponding to X_2=1.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



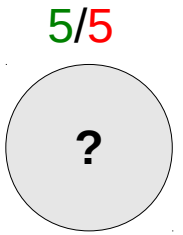
Also in this case, a perfectly discriminant attribute exists, i.e., X_5.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



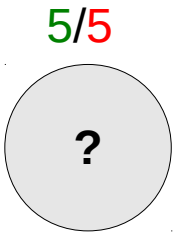
The last two recursive calls to ID3 produce the final decision tree shown above.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



So far the discriminant capability of an attribute has been evaluated only qualitatively. Among several possible quantitative measures, in the ID3 learning algorithm the **entropy** of the class distribution is chosen, estimated from the training examples that reach the considered node.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1

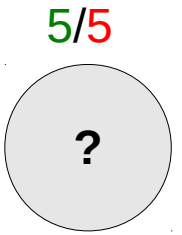


In this example, before choosing the attribute of the root node the whole training set must be considered, which is made up of 5 legitimate and 5 spam emails. Accordingly, the class distribution can be estimated as:

$$P(Y=L) = 5/10 = 0.5$$

$$P(Y=S) = 5/10 = 0.5$$

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



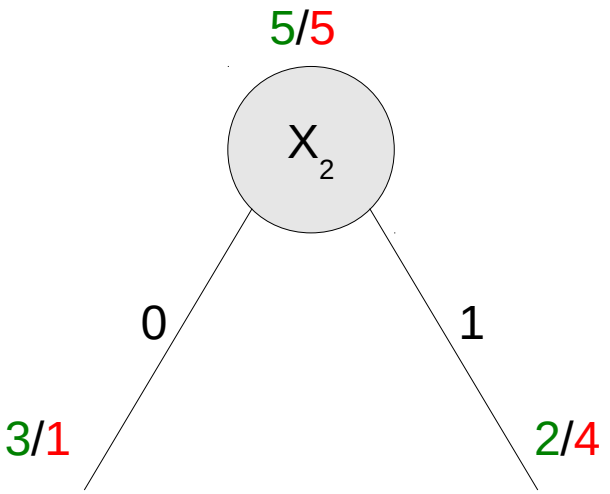
The corresponding entropy is defined as:

$$H(Y) = -P(Y=0)\log_2 P(Y=0) - P(Y=1)\log_2 P(Y=1)$$

$$= -0.5\log_2 0.5 - 0.5\log_2 0.5$$

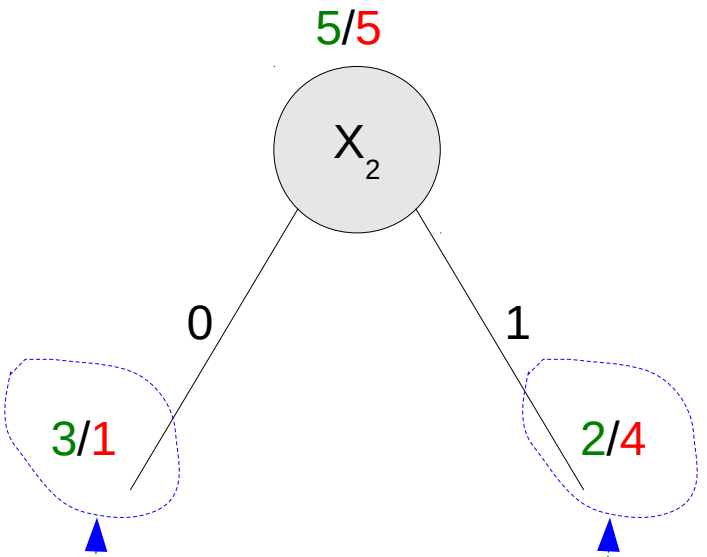
$$= 1$$

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



If X_2 is chosen as the attribute of the root node, it produces the two class distributions shown above (one for each output value).

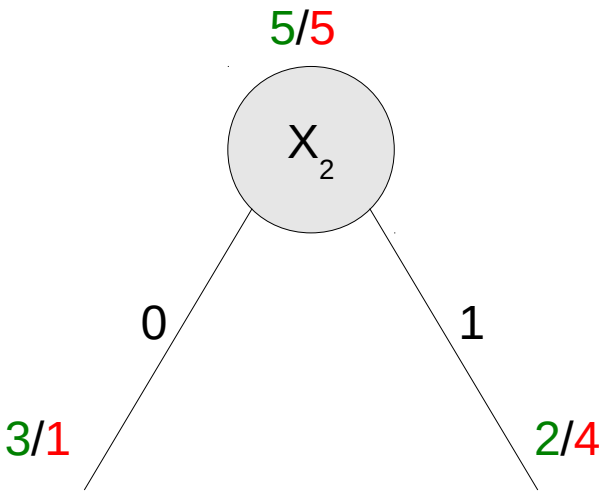
	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



The entropy of the class distribution, **after** observing the value of X_2, is defined as the **conditional** entropy:

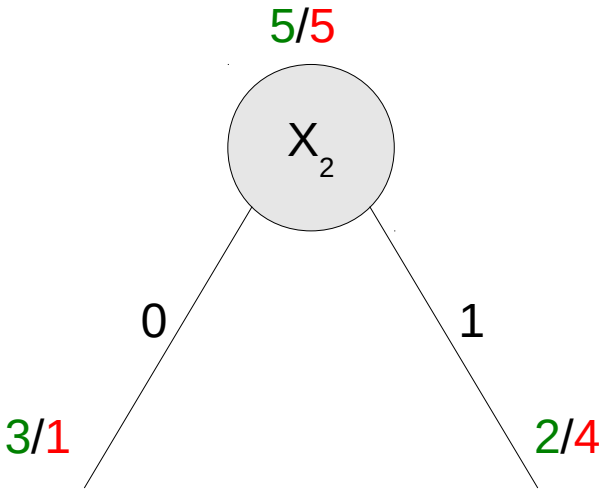
$$H(Y \mid X_2) = P(X_2=0)H(Y \mid X_2=0) + P(X_2=1)H(Y \mid X_2=1).$$

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



To compute $H(Y \mid X_2)$, the values $P(X_2=0)$ and $P(X_2=1)$ can be estimated as 4/10 and 6/10, respectively, whereas $H(Y \mid X_2=0)$ and $H(Y \mid X_2=1)$ can be computed from the distributions $P(Y \mid X_2=0)$ and $P(Y \mid X_2=1)$, respectively.

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



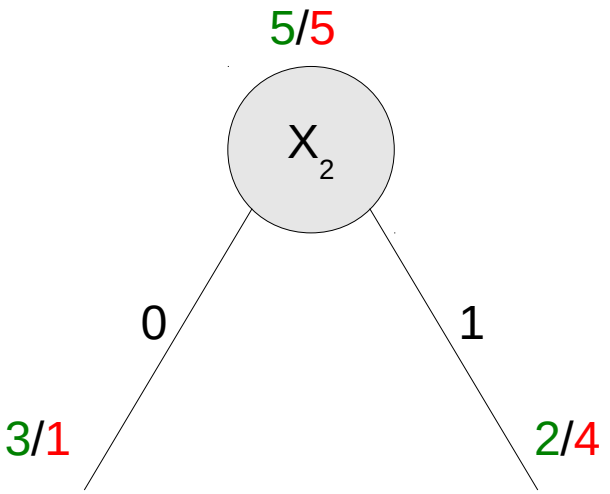
Accordingly, one obtains:

$$H(Y \mid X_2) = P(X_2=0)H(Y \mid X_2=0) + P(X_2=1)H(Y \mid X_2=1)$$

$$= 0.4(-3/4 \log_2 3/4 - 1/4 \log_2 1/4) + 0.6(-2/6 \log_2 2/6 - 4/6 \log_2 4/6)$$

$$\approx 0.875$$

	Y		X_1	X_2	X_3	X_4	X_5
M_1	L		1	0	1	0	1
M_2	L		1	0	1	1	0
M_3	L		0	0	1	1	1
M_4	L		0	1	0	0	0
M_5	L		0	1	1	1	0
M_6	S		0	0	0	0	0
M_7	S		0	1	0	1	1
M_8	S		0	1	1	1	1
M_9	S		1	1	0	1	1
M_{10}	S		1	1	1	1	1



To sum up, **before** observing the value of X_2 the entropy of the class distribution is $H(Y)=1$; **after** observing the value of X_2 the entropy **reduces** to $H(Y \mid X_2) \approx 0.875$. This means that the attribute X_2 has some discriminant capability. The discriminant capability of any attribute at **any node** of a DT can be evaluated similarly.