1. Consider the following dataset:

	y_1	y_2	y_3	z
x_1	a	a	a	+
x_2	c	b	c	+
x_3	c	a	c	+
x_4	b	a	a	-
x_5	a	b	c	-
x_6	b	b	c	-

Plot the learned decision tree using information gain (Shannon entropy). Show your calculations.

Intuitively, the amount of information we can gather upon observing a random event is inversely proportional to the probability of it happening. Shannon's notion of **entropy** is a formalization of this idea. The entropy of a random variable is the average amount of information we can gather upon observing it, and is defined as follows:

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

Here, $\mathcal{X}$ is the set of all possible values of X and $p(x)$ is the probability of X taking the value x.

The **information gain** concept is a measure of how much information we gain by observing a random variable X - the larger the information gain value, the more we can extract from a given feature for the given dataset. It is defined as the difference between the entropy of the random variable X and the *conditional entropy* of X given Y: a measure of how much information is needed to describe X given that we know Y.

$$\text{IG}(X, Y) = H(X) - H(X|Y)$$

Having these definitions in mind, it should now be trivial to plot the decision tree for the dataset above. We start by computing the entropy of the target variable z:

$$H(z) = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1$$

Learning a decision tree will be intrinsically related to the features' information gain measure: for each level, we will select the feature that maximizes the information gain with respect to the target variable, making it that level's decision node. Let's compute the information gain for each feature:

$$H(z \mid y_1) = \sum_{y \in \mathcal{Y}_1} p(y_1 = y) H(z \mid y_1 = y) = \frac{2}{6}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right) + \frac{2}{6}\left(-\frac{2}{2}\log_2\frac{2}{2}\right) + \frac{2}{6}\left(-\frac{2}{2}\log_2\frac{2}{2}\right) = \mathbf{\frac{1}{3}}$$

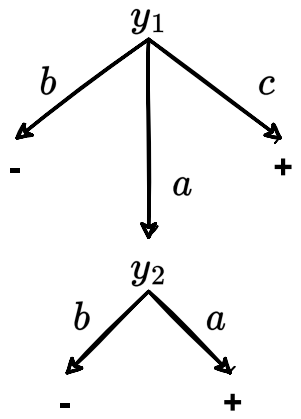
$$IG(z, y_1) = 1 - \frac{1}{3} = \mathbf{\frac{2}{3}}$$

Performing similar computations for the other features, we obtain:

$$H(z \mid y_2) = 0.9183, \qquad IG(z, y_2) = 0.082$$
$$H(z \mid y_3) = 1, \qquad IG(z, y_3) = 0$$

As mentioned above, we will select the feature that maximizes the information gain for each level. In this case, y_1 is the feature that maximizes the information gain, hence it will be the root node of the decision tree. Note how there are already a couple of leaves on our tree: these indicate that, regarding the respective decision tree node, the target variable is already fully determined by the feature values (i.e for a given feature value, the target variable is always the same).

Continuing the process, and following the path for the feature y_1 that takes the value a, we can choose either y_2 or y_3 as the next decision node, since the entropies of y_2 and y_3 are the same, zero (considering data conditioned by $y_1 = a$). Choosing y_2 as the next decision node, we're now left with only lead nodes, therefore no further decision nodes should be added to the tree.



2. Show if a decision tree can learn the AND, OR and XOR logical functions.

Note that the afore-mentioned logical functions can be represented as shown in the following "dummy data sets":

	y_1	y_2	z
x_1	0	0	0
x_2	0	1	0
x_3	1	0	0
x_4	1	1	1

Table 1: AND logical function

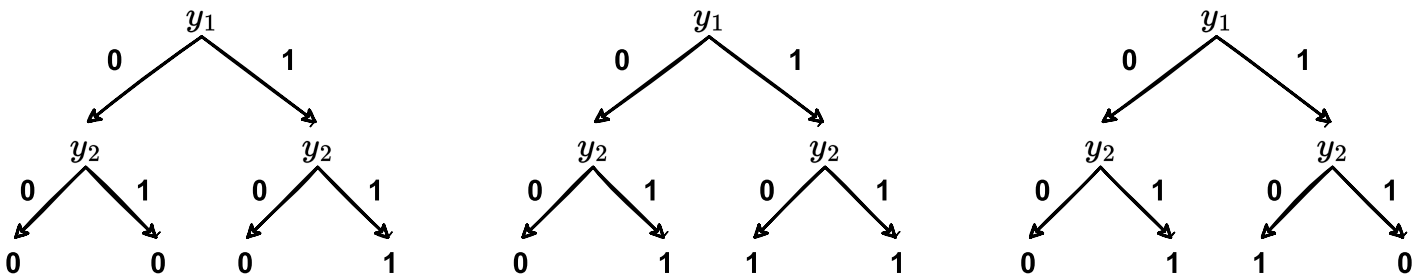
	y_1	y_2	z
x_1	0	0	0
x_2	0	1	1
x_3	1	0	1
x_4	1	1	1

Table 2: OR logical function

	y_1	y_2	z
x_1	0	0	0
x_2	0	1	1
x_3	1	0	1
x_4	1	1	0

Table 3: XOR logical function

Each one of these data sets can be learned by a decision tree, of course:



3. Consider the following testing targets, z, and the corresponding predictions, $\hat{z}$, by a decision tree:

$$z = [A, A, A, B, B, B, C, C, C, C]$$
$$\hat{z} = [B, B, A, C, B, A, C, A, B, C]$$

(a) Draw the confusion matrix.

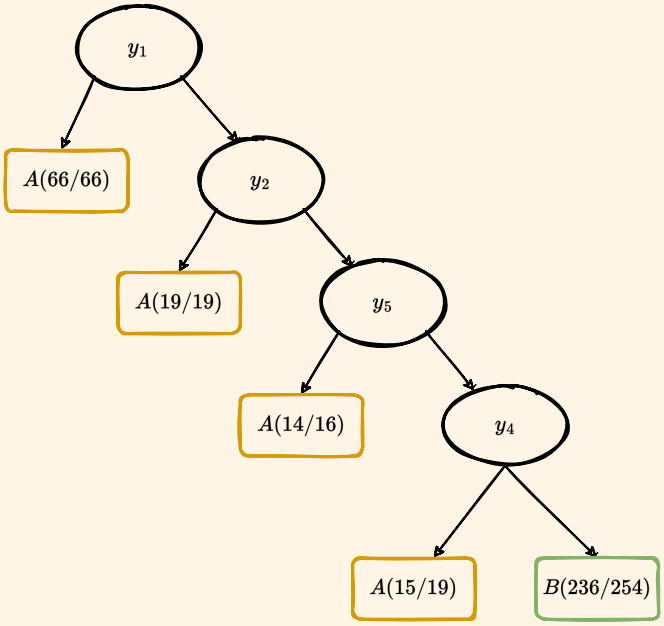
(b) Compute the accuracy and recall (sensitivity) for each class.

(c) Regarding class C, identify its precision and F-measure.

(d) identify the accuracy, sensitivity and precision of a random classifier.

As we know, a confusion matrix is a $K \times K$ matrix, where K is the number of classes in our target label. In this case, we have three classes, A, B and C, therefore our confusion matrix will be a 3×3 matrix. Considering the *real vs predicted* labels referenced above, we can fill the confusion matrix as follows:

4. Consider a dataset composed by 374 records, described by 6 variables, classified according to the following decision tree:



Each leaf in the tree shows the label, number of classified records with the label, and total number of observations in the leaf. The positive class is the minority class.

(a) Compute the confusion matrix.

(b) Compare the accuracy of the given tree versus a pruned tree, with only two nodes. Is there any evidence towards overfitting?

(c) Are decision trees learned from high-dimensional data susceptible to underfitting? Why does an ensemble of DTs minimize this problem?