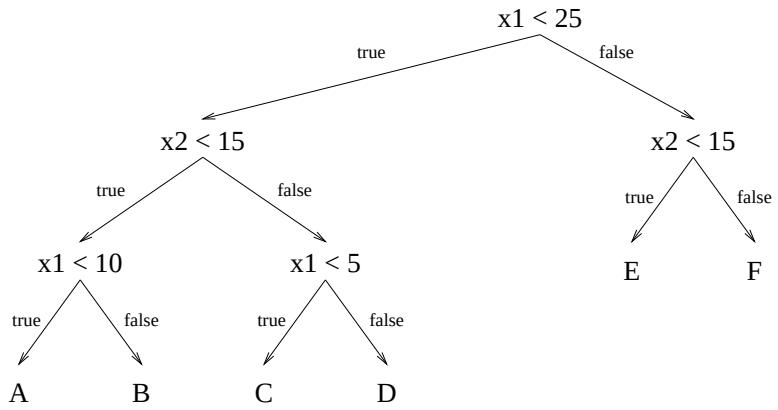
AI534 — Written Assignment 4 —

1. Consider the following decision tree:



   (a) (3 pts) Draw the decision boundaries defined by this tree. Each leaf of the tree is labeled with a letter. Write this letter in the corresponding region of input space.

   Your answer here

   (b) (3 pts) Give another decision tree that is syntactically different but defines the same decision boundaries. This demonstrates that the space of decision trees is syntactically redundant.

   Your answer here

   (c) (2 pts) How does this redundancy influence learning (does it make it easier or harder to find an accurate tree)?
   Your answer here

2. In the basic decision tree algorithm (assuming we always create binary splits), we choose the feature/value pair with the maximum information gain as the test to use at each internal node of the decision tree. Suppose we modified the algorithm to choose at random from among those feature/value combinations that had non-zero mutual information, and we kept all other parts of the algorithm unchanged.

   (a) (2 pts) What is the maximum number of leaf nodes that such a decision tree could contain if it were trained on m (distinct) training examples?
   Your answer here

   (b) (2 pts) What is the maximum number of leaf nodes that a decision tree could contain if it were trained on m training examples using the original maximum mutual information version of the algorithm? Is it bigger, smaller, or the same as your answer to (b)?
   Your answer here

   (c) (3 pts)How do you think this change (using random splits vs. maximum information mutual information splits) would affect the accuracy of the decision trees produced on average? Why?
   Your answer here

3. (10 pts) Consider the following training set:

A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

Learn a decision tree from the training set shown above using the information gain criterion. Show your steps, including the calculation of information gain (you can skip $H(y)$ and just compute $H(y|\mathbf{x})$) of different candidate tests. You can randomly break ties (or better, choose the one that give you smaller tree if you do a bit look ahead for this problem).

Your answer here

4. (6 pts) Please show that in iteration l of Adaboost, the weighted error of h_l on the updated weights D_{l+1} is exactly 50%. In other words, $\sum_{i=1}^{N} D_{l+1}(i)I(h_l(X_i) \neq y_i) = 50\%$, where $I(\cdot)$ is the indicator function that takes value 1 if the argument is true. (Hint: start with the condition that, post update, the total weight of correct examples equals the total weight of in-correct examples, i.e., 50% each.)

   Your answer here

5. (6 pts) In class we showed that Adaboost can be viewed as learning an additive model via functional gradient descent to optimize the following exponential loss function:

$$\sum_{i=1}^{N} \exp(-y_i \sum_{l=1}^{L} \alpha_l h_l(x_i))$$

   Our derivation showed that in each iteration l, to minimize this objective, we should seek an h_l that minimizes the weighted training error, where the weight of each example $w_l^i = \exp(-y_i \sum_{t=1}^{l-1} \alpha_t h_t(x_i))$ prior to normalization. Show how this definition of $w_l^i \propto D_l(i)$ computed by Adaboost (Hint: use induction).

   Your answer here