Instituto Superior de Agronomia, ULisboa

Green Data Science

Practical Machine Learning/Aprendizagem Automática Aplicada

Questionnaire #7, May 5, 2023

Name:_____

Topic: decision trees and random forests

1. Consider the following data set, with 8 examples, each one described by 4 binary variables (for instance '1100' means x_1=1, x_2=1, x_3=0, x_4=0). The labels are "A" or "B":

label A	label B
1100	1100
0000	1111
1010	1110
0011	0111

Build a decision tree with maximum depth 2 that classifies correctly all but one example. Draw the tree indicating which are the examples in each node, using the same 4 digit notation as above. Note that the root node has depth 0. (Note: during the questionnaire, it was clarified that the depth could be just one).

Response:

label A	label B
1100	1100
0000	1111
1010	1110
0011	0111

X_2=0

label A	label B
0000	
1010	
0011	

p_A=1; p_B=0

$X_2 \neq 0$

label A	label B
1100	1100
	1111
	1110
	0111

p_A=1/5=0.2: p_B=4/5=0.8

2. The Gini loss is 1- (0.5^2+0.5^2)=0.5 for the root node. Compute the Gini loss for every node on the decision tree that you built (calculating machine not needed).

Response: The Gini loss is given by $G=1-\sum p_i^2$. For the parent node, this returns 1- (0.5^2+0.5^2). For the left child node it is 1- (1^2+0^2)=0. For the right child node it is 1-

$(0.2^2+0.8^2)=8/25$. The value for the right node is larger since the left node is pure and the right isn't.

3. Consider the following pseudo-code to create a decision tree from "data" (lines starting by # are comments):

```
1.  build_tree(data)
2.      # Base case: If all samples in the data belong to the same class, return a leaf node with that class
3.          if all samples in data belong to the same class: return a leaf node with that class
4.      # Otherwise, find the best feature to split on
5.          best_split = find_best_split(data)
6.          # Create a decision node based on the best feature
7.          decision_node = DecisionNode(split=best_split)
8.          # Split the data based on the best feature
9.          left_data, right_data = split_data(data, best_split)
10.         # Recursively build the left and right subtrees
11.         decision_node.left_child = build_tree(left_data)
12.         decision_node.right_child = build_tree(right_data)
```

a) At which step is the loss function minimized? Please provide a brief explanation to justify your answer.

Response: the loss function is minimized when choosing the best split (Line 5). In a tree that corresponds to choosing the variable and the threshold for the split. Among all possibilities, the variable and the threshold that minimize the loss are the chosen ones to define the split.

b) If you want to adapt the code to regularize the tree by enforcing a minimum leaf size of 4, what changes would you make?

Response: to prevent a note to split if its size is lower than 4, one could adapt step 3 to:

3. if all samples in data belong to the same class or if the number of samples is less or equal to 4: return a leaf node with the label of the most frequent class.

4. When creating a Random Forest for classification, one concern is limiting the correlation between models. Explain briefly what are the two main strategies to achieve that goal.

Response: There are two strategies to limit the correlation between trees:

a) bootstrap, i.e. resampling with replacement so the root data set is made of different samples
b) choose randomly a subset of features to grow each tree