SD201

DECISION TREE

October 19, 2018

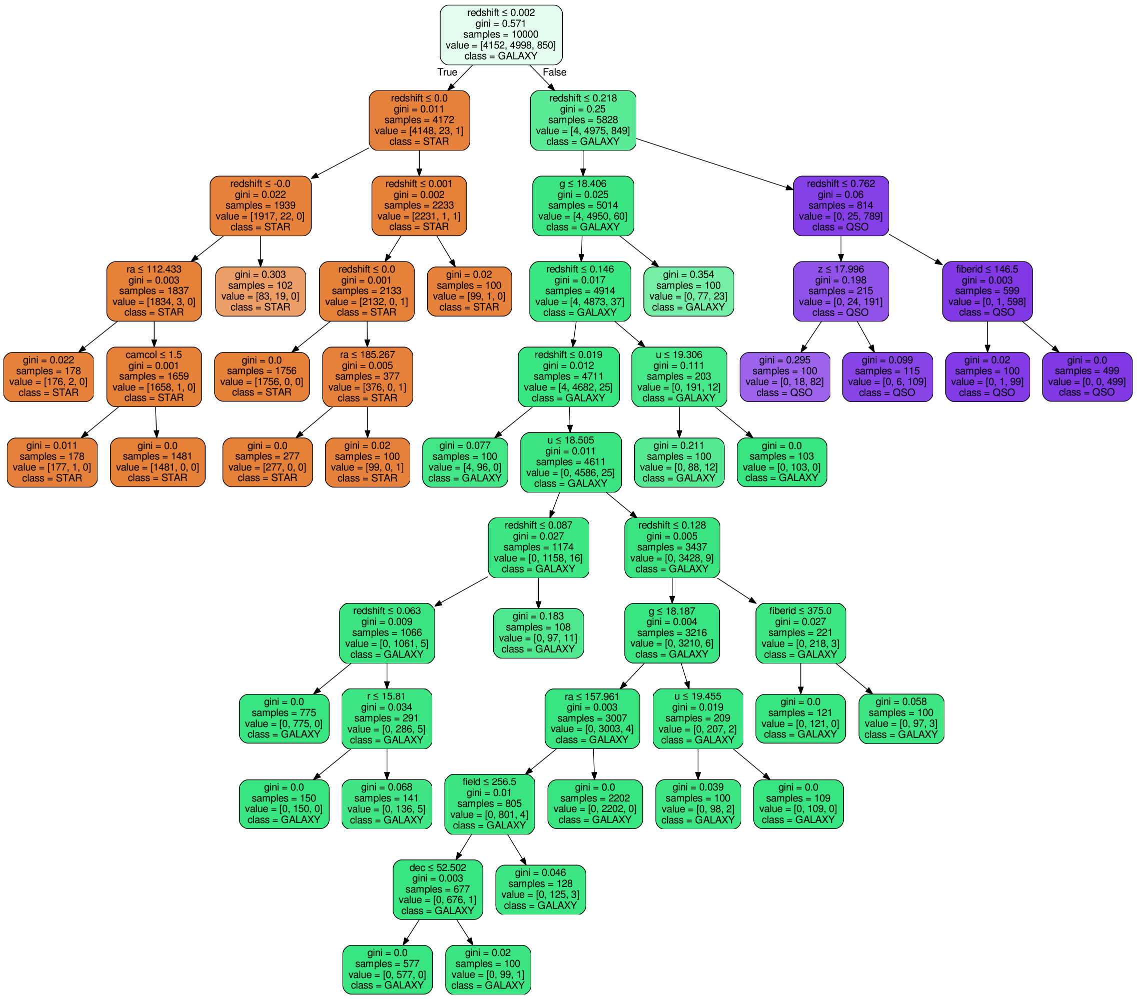
Student : Raphael REME

TELECOM-PARISTECH

Department of data sciences

1 Given the dataset we provided to you, build a decision tree using the default input parameters.

With the data provided I made a similar tree as the one given in example.



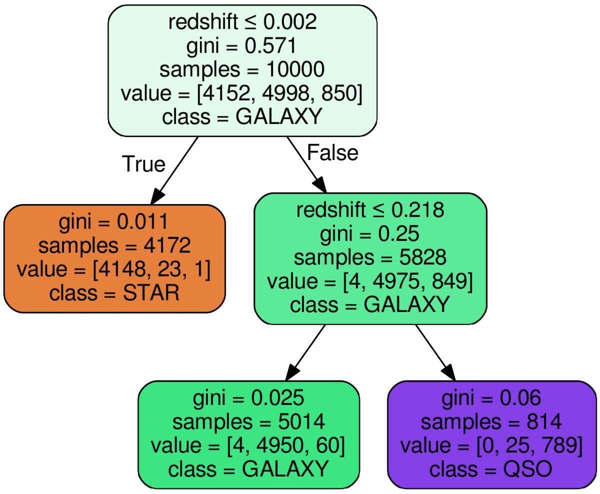
2 Compute the generalization error of the decision tree you built.

To compute it I use : $GenError = TrainError + \frac{1}{2} * NbLeaves$
Thanks to the library it is easy to compute $TrainError = 1.13\%$
And after computation I find $NbLeaves = 28 \Rightarrow GenError = TrainError + \frac{1}{2} * \frac{28}{10000} = 1.27\%$

3 Build a decision tree with minimum generalization error.

The generalization error represents the expected error on a testing set. It takes in account the error on the testing set and an error due to new data with the possibility that the tree has over-fit the training set (It is represented by the number of leaves in the formula).
In order to minimize the generalization error we can therefore try to minimize the testing error or the added error (NbLeaves). To prevent overfitting (and to limit NbLeaves) two attributes can be used : max_depth attribute to limit depth of the tree (it prevents it to divided the space of data in to much sub-spaces and then to over-fit training data) and min_samples_leaf attribute which set a minimal size to nodes (their size is then large enough to make a statistically significant decision). In both case NbLeaves decreases but the training error will probably increase. But min_samples_leaf attribute is set to 0.01 and should keep its value, so I will only try to change the max_depth attribute. Then we can pre-pruned the tree with min_impurity_decrease attributes in order to be sure that every split does reduce the impurity and it also possible to try to change giny with entropy and see this give better results (It should be almost the same, but as it can be better let's try !)
I compute all the possibility and had $bestGenError = 1.145\%$ for the parameters :

1. Criterion = **gini or entropy**
2. Max_depth = **2 or more**
3. Min_impurity_decrease = **0.01**



4 Compare the decision trees you built in point 1 and the best one you obtained in point 2.

The second I got is I think better than the first one : it gives precisely the same testing error and have a lower generalization error.
It is basically the same as the first one but pruned, with no useless division of the data space.
So it should have less over-fitted my training set.

5 Predict the class value of an object of your choice.

I chose the objid : $1,23764870457714E+018$ (the first one). Its redshift is lower than 0.002, so it will be classed as a STAR !
What is interesting here is the fact that it only depends on redshift ! Other data aren't taken into account !

6 Could the second tree be pruned ?

No we already minimized its generalization error. So it can't be pruned in order to improve the generalization error as it is already minimal.
A quick way to be sure it will not be better is to reduce max_depth to 1 : It's equivalent to prune the right side of the tree (which is the only one that can be pruned as the left side is already a leaf). But we already know from the question 3 algorithm that max_depth = 1 is not an optimal solution.

7 Implement a post-pruning strategy and run it on the best decision tree so far. Does this improve the generalization error?

No it doesn't. But it does give this minimal tree if we run it on some other trees (For instance on the first tree) !