P04 – decision trees

In this lab you will use decision trees to classify offers of used cars. This lab is designed to give you a good understanding of how decision trees work and how to apply them. You should also get a feeling for what overfitting is and how to handle it. The accompanying template “P04_decisiontrees_task.py” is meant to be used with an interactive python console line by line (i.e. don’t try to run the whole thing at once).

At the top of the template there are some helper-function which should make it easier for you to implement your own decision trees, as well as some data preprocessing code.

1. ***Start small***
   At first you are asked to train a tree-stump (a tree of depth 1). In order to do so you need to complete the function *find_best_split* which finds the best split with respect to the gini impurity. Don’t get fancy here its best to try every possible split (this is why decision trees are slow for numerical variables).

2. ***Confusing error rates***
   Compute the confusion matrix (predicted vs actual label), as well as the overall prediction error, for the training set and the test set. Any comments?

3. ***We have to go deeper***
   Complete the *train_tree* method to recursively train deeper trees. (if you have reached a leaf return in_data otherwise return a new Tree_node)
   Train a tree of depth 5, does this tree perform better than the tree stump?

4. ***This is much easier!***
   Use the sk-learn class *DecisionTreeClassifier* to train another tree of depth 5 (also based on the gini impurity). Does this perform different from your own implementation? Hint: you have to use the encoded versions of the data because the *DecisionTreeClassifier* only takes numerical labels.

5. ***Machinelearners cookbook,***
   step 1: throw more computing power at the problem
   step 2: if necessary repeat step 1

   Create your own implementation of Adaboost, by completing the skeleton *ada_boost_trees.* For performance reasons base this the algorithm on the *DecisionTreeClassifier* and not on your own decision tree implementation. Use decision trees of depth 5 as base classifiers. Does the boosted tree perform better?

6. ***This is much easier! v2.0***
   Compare your Adaboost implemention against sk-learns *AdaBoostClassifier* are there significant performance differences?