MSAI-349

Homework Assignment #1
Group 2

1. **Did you alter the Node data structure? If so, how and why?**

   We did alter the Node data structure, and added the following fields:

   a. *examples*: A list of all training examples at this node of the tree while training.
   b. *attributes:* The list of attributes on which a split has not yet been performed, derived from examples during node creation.
   c. *attr_to_split_on*: The best attribute to split on to proceed to the next level of the tree.
   d. *information*: Self-information of this node, calculated during node creation.
   e. *is_pruned*: Boolean variable that determines if a node has been pruned or not (i.e., should it be traversed during test time or not).

2. **How did you handle missing attributes, and why did you choose that strategy?**

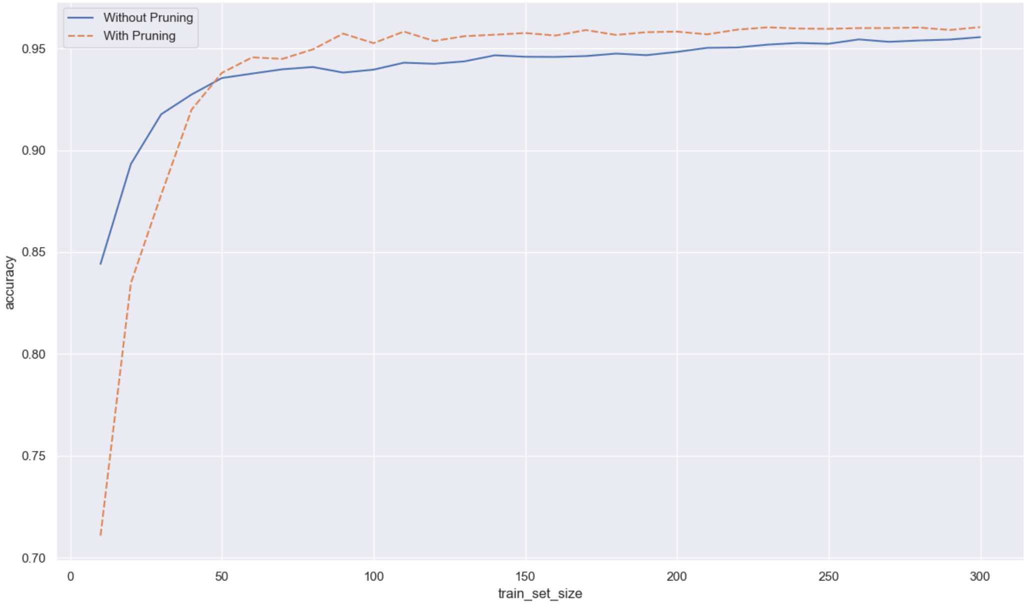
   We handled missing attributes in two ways:

   a. *Before training*, we replaced every missing value in the training set with the most common value of the concerned attribute. We decided that this was a better strategy than deleting examples with missing values altogether, since the latter would have left us with very little data to learn from.
   b. *During test time*, if a missing value was encountered, we simply went down the branch with more training examples (see 1a. to see that we store the training examples in each node). This method was chosen to increase the odds of more accurate classification further down the tree.

3. **How did you perform pruning, and why did you choose this strategy?**

   We performed Reduced Error Pruning, testing the removal of branches starting from leaf nodes and working our way upwards. This strategy was chosen since it is one of the simplest and fastest ways to prune a decision tree. It ended up increasing our test accuracy from around 93.5% to 95.5% on housing data.

4. **Plot:**



a. **What is the general trend of both lines as training set size increases, and why does this make sense?**

Both the lines go up (test accuracy increases) as the training set size increases, since there is more data to learn from. After a certain number of training examples, both the lines sort of stabilize and we get to see less steep ups-and-downs in accuracy as the training set size is further increased.

This makes sense since at the beginning, learning is limited due to less number of training examples, but as the model learns from more and more examples, the learning improves and so does the accuracy. After a certain point, the model learns enough from the examples to generalize and its performance reaches a plateau.

b. **How does the advantage of pruning change as the data set size increases? Does this make sense, and why or why not?**

Pruning, in the very beginning (with a sample size of 10), is not very effective. This is because a small training set gives rise to a small (simple) decision tree, which by definition cannot benefit from pruning (and in fact suffers from it).

However, as we increase the size of the training set, the benefits of pruning readily become apparent, and the test accuracy of a pruned tree flattens out at a noticeably higher value than that of the unpruned tree.

This makes sense, since our pruning method simplifies the decision tree model based on the reduced error criterion and hence prevents the tree from overfitting. Therefore, the pruned tree model generalizes well with more training examples and fares well while making predictions on the test data.

However, in the latter half of the plot, it is clear that the unpruned tree is catching up to the pruned tree. This is because with more data points, the tree is able to learn more complex rules and is able to predict the test distribution more accurately.

It is important to note that even though the accuracies of both the pruned and unpruned trees converge, the pruned tree will be a faster model to run, owing to its simplicity (having fewer branches).

5. **Use your ID3 code to construct a Random Forest classifier using the candy.data dataset. You can construct any number of random trees using methods of your choosing. Justify your design choices and compare results to a single decision tree constructed using the ID3 algorithm.**

We evaluated a 50-tree Random Forest vs a single decision tree, and saw the following results on the candy dataset (averaged over 500 iterations):

ID3 single tree mean test accuracy: 69.64%
Random forest mean test accuracy: 74.52%

Also, the standard deviation (SD) of the accuracies across the trees in our random forest model was 0.09. The low value of SD implies a good model with reasonably high confidence in individual estimators.

Another design choice we made was to subsample the number of features as “sqrt(n) + 1” (4 out of 9 for the candy dataset). Our belief behind this decision was that subsampling the number of features will help reduce the correlation among and between the individual decision trees and hence decrease generalization errors.

Since we built this random forest model for a classification problem, we also made an assumption that the correlation will increase slowly (as compared with the regression problems) and considering “sqrt(n) + 1” number of features instead of “log n” features will improve the performance of individual trees while not penalizing our model with out-of-sample errors.

We tried tweaking the number of trees but the performance gain plateaued at around 50 trees.