Homework 1 Writeup
CS 349 - Machine Learning
Spencer Rothfleisch, Louie Shapiro, Max Ward

1. (2.0 points) Did you alter the Node data structure? If so, how and why?

We modified the node data structure by inserting fields for entropy and for the most common class of the examples when the node was created. The entropy field was used to determine the information gain of that node. It was created for possible pruning methods that utilize information gained from each node. The common class field is also utilized for reduced error pruning in the case when a node is found to increase validation accuracy when it is removed. When the node is removed, it is replaced with a leaf node of the most common class.

2. (2.0 points) How did you handle missing attributes, and why did you choose this strategy?

We handled missing attributes by automatically assigning the data the most common attribute value in the dataset at that point in the tree. This method was used due to its simplicity to implement. Additionally, it provides an easy way to determine a rough estimate of what the vector with the missing attribute would have if the attribute was present.

3. (2.0 points) How did you perform pruning, and why did you choose this strategy?

We chose to use reduced error pruning. This was done because it is a good way to see if a node is contributing to test accuracy or not. When a node is found to increase validation accuracy when it is pruned, it is replaced with a leaf node. This was done using a post order traversal, so the nodes could be visited from the bottom of the tree up to the parents. Using reduced error pruning, we were able to increase overall tree accuracy post-pruning.

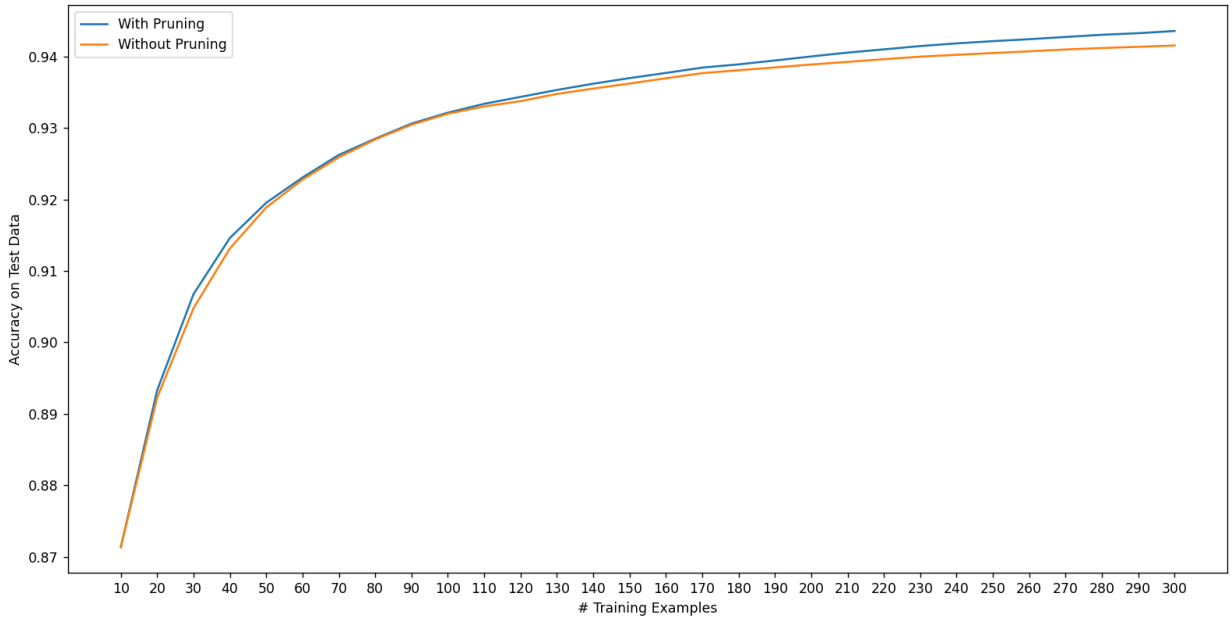
4. (4.0 points) Now you will try your learner on the house_votes_84.data, and plot learning curves. Specifically, you should experiment under two settings: with pruning, and without pruning. Use training set sizes ranging between 10 and 300 examples. For each training size you choose, perform 100 random runs, for each run testing on all examples not used for training (see testPruningOnHouseData from unit_tests.py for one example of this). Plot the average accuracy of the 100 runs as one point on a learning curve (x-axis = number of training examples, y-axis = accuracy on test data). Connect the points to show one line representing accuracy with pruning, the other without. Include your plot in your pdf, and answer two questions:

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

As training set size increases, both lines tend to increase, denoting that accuracy increases with training set size increase. This makes sense, because the model is able to predict trends better as it is able to generalize those trends across more examples. Additionally, with a larger training set, the risk of overfitting decreases, which will improve accuracy.

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

While this was not consistently the case, the advantage of pruning tended to increase as the data set size increased. This does not necessarily make sense, because pruning is meant to help with the overfitting that occurs when small training datasets are used, but overfitting tends to become less of an issue as the size of the training dataset grows.



Note: depending on your particular approach, pruning may not improve accuracy consistently or may decrease it (especially for small data set sizes). You can still receive full credit for this as long as your approach is reasonable and correctly implemented.