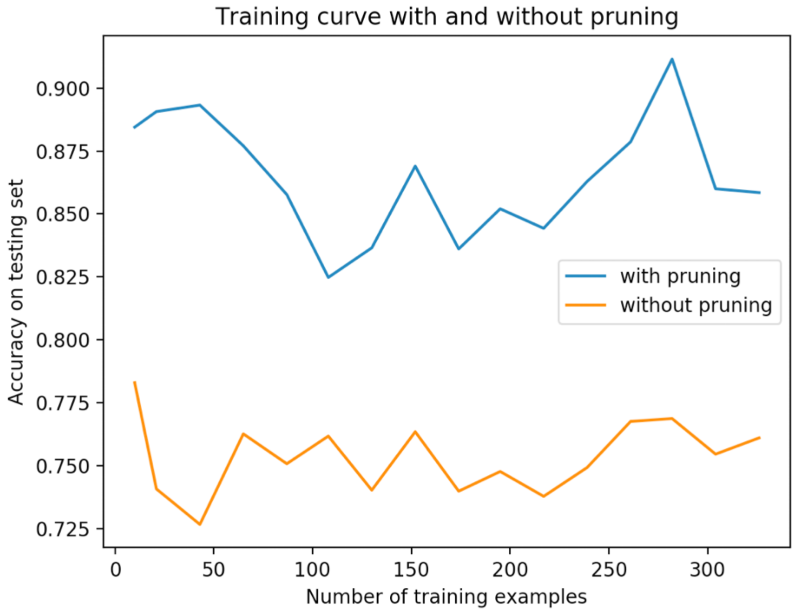
Which other student, if any, is in your group? (either names or netIDs is fine)
Kelly Roemer

(0.5 points) Did you alter the Node data structure? If so, how and why? (2 sentences)
Yes we altered the Node data structure. We added the following attributes:
listOfChildren: which contains the list of all of the nodes which fall under one node
attribute: contains the name of the attribute which the node split on
listOfPeople: contains a list of all of the dictionaries which are still being sorted at the current node
orderedListOfAttributeStates: contains a list of the attribute values for the attribute which the node split on

(1 point) How did you handle missing attributes, and why did you choose this strategy? (2 sentences)
To handle our missing attributes we would take all the attributes with a ? for the splitting attribute at a node and pass them into ALL of the categorical attribute children which were no ?'s. In addition, when calculating information gain we ignored the presence of the missing attributes and treated the overall length of the list of examples as the length without the missing attributes. We chose this method because it allowed us to continue using the remaining attributes which had not yet been split on for the example and it did not affect the choice of best attribute to split on in information gain.

(1 point) How did you perform pruning, and why did you choose this strategy? (4 sentences)
The purpose of pruning is eliminating unnecessary splitting. Because of this, we wanted to continuously test on each node if it is more accurate to have the remaining splits occur or to just assume that all of the results from that point on will be the mode of the class of all of the nodes that remain at that point.

(2 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:



In about a sentence, what is the general trend of both lines as training set size increases, and why does this make sense?

When our training size is still relatively small (<25% of our data) and relatively large (>75% of our data) pruning accuracy seems to decrease. However, in the center range it is consistently increasing. In addition, across the entire range of percentages it provides higher accuracy than without pruning. This makes sense because too small of a training set will create a shorter tree thereby reducing accuracy when further decreasing the nodes and a larger training set has a very deep, accurate, and specific decision tree so forcing that tree to prune will decrease the accuracy on the training set

In about two sentences, how does the advantage of pruning change as the data set size increases? Does this make sense, and why or why not?

See previous answer