Subrat Mahapatra

1. Worked on this assignment individually
2. I added the “mode” of the class examples to each node so that my pruning algorithm would know how to properly excise the node, used the label was the string of the attribute that I split on, and implemented the children as a dictionary that contained the branch and child node. Choosing the dictionary to store my sub-trees was an useful way of creating subtrees recursively and adding them to the root node.
3. Missing attributes were handled by taking the mode of the attribute. Adopted this because it made sense that the majority of those attributes gave a better idea about who was republican or democrat, and by adding a similar value, we were only adding to the majority.
4. I handled post-pruning the tree with a reduced error algorithm. This pruned the tree starting at the leaves and checked the accuracy with the validation data. The algorithm I designed was a recursive algorithm where I am passing a pointer to the original root to do accurate accuracy tests, and before I excised the node I would make a copy of the original just incase the prune was bad. In that case I would stop the algorithm and return the copy.
5.
    a. Pruning the decision tree does better when you expose a larger amount of training examples, while increasing the examples on an algorithm without pruning is not as effective.
    b. As your data set increases, pruning is more effective because it combats overfitting in these larger datasets where the decision tree has more sets of data to train on. Overcoming overfitting is what gives tress that are pruned an edge over without pruning.