Student Name: Junhan Liu
NetID: jlt587

1. How did you handle missing attributes in examples

We substitute the mode of the attribute in the same classification with the missing attribute, i.e. assign most common value of A among other examples with same target value.

2. Apply your algorithm to the training set, without pruning. Print out a Boolean formula in disjunctive normal form that corresponds to the *unpruned* tree learned from the training set. For the DNF assume that group label "1" refers to the positive examples. NOTE: if you find your tree is cumbersome to print in full, you may restrict your print-out to only 16 leaf nodes.

(numinjured<1.0 ^ oppnuminjured<1.0 ^ opprundifferential<32.0) v (numinjured<1.0 ^ oppnuminjured<1.0 ^ opprundifferential>=32.0 ^ oppnuminjured<0.0) v (numinjured<1.0 ^ oppnuminjured<1.0 ^ opprundifferential>=32.0 ^ oppnuminjured>=0.0 ^ dayssincegame>=2.0) v (numinjured<1.0 ^ oppnuminjured>=1.0 ^ opprundifferential<43.0 ^ rundifferential>=71.0) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent<0.261062525 ^ winpercent<0.661899761) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525 ^ startingpitcher=1) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525 ^ startingpitcher=1 ^ startingpitcher=2) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525 ^ startingpitcher=1 ^ startingpitcher=2 ^ startingpitcher=3) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525 ^ startingpitcher=1 ^ startingpitcher=2 ^ startingpitcher=3 ^ startingpitcher=4) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525 ^ startingpitcher=1 ^ startingpitcher=2 ^ startingpitcher=3 ^ startingpitcher=4 ^ startingpitcher=5) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent>=0.354916694 ^ winpercent<0.777399225 ^ winpercent<0.219309677) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent>=0.354916694 ^ winpercent<0.777399225 ^ winpercent>=0.219309677) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent>=0.354916694 ^ winpercent>=0.777399225 ^ oppwinningpercent>=0.375600184) v (numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured<2.0 ^ opprundifferential<13.0) v (numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured<2.0 ^ opprundifferential>=13.0 ^ opprundifferential<32.0) v (numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured>=2.0 ^ oppwinningpercent>=0.15431423 ^ oppwinningpercent<0.482567275) v

(numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured>=2.0 ^ oppwinningpercent>=0.15431423 ^ oppwinningpercent>=0.482567275)

3. Explain in English one of the rules in this (unpruned) tree?

The lower the number of injured is, the more likely 0 will win.

4. How did you implement pruning?

Post-pruning, i.e. prune the tree after the tree was constructed. In post-pruning method, we replace each sub-tree by a leaf node, and check if it will influence the accuracy. If it will not influence the accuracy, then the sub-tree is redundant, therefore we need to substitute the sub-tree with a leaf.

5. Apply your algorithm to the training set, with pruning. Print out a Boolean formula in disjunctive normal form that corresponds to the *pruned* tree learned from the training set.

(numinjured<1.0 ^ oppnuminjured<1.0) v (numinjured<1.0 ^ oppnuminjured>=1.0 ^ opprundifferential<43.0 ^ rundifferential>=71.0) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent<0.261062525 ^ winpercent<0.661899761) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent<0.354916694 ^ oppwinningpercent>=0.261062525) v (numinjured>=1.0 ^ oppnuminjured<2.0 ^ oppwinningpercent>=0.354916694) v (numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured<2.0 ^ opprundifferential<13.0) v (numinjured>=1.0 ^ oppnuminjured>=2.0 ^ numinjured>=2.0 ^ oppwinningpercent>=0.15431423)

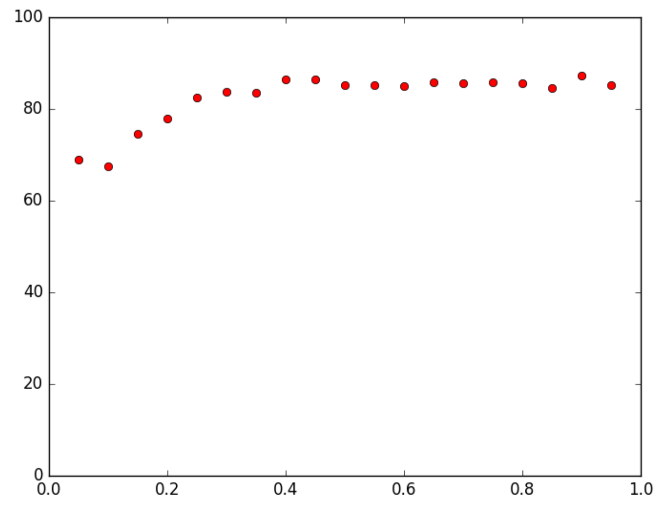
6. What is the difference in size (number of splits) between the pruned and unpruned trees?

The pruned tree has 11 splits and the unpruned tree has 21 splits.

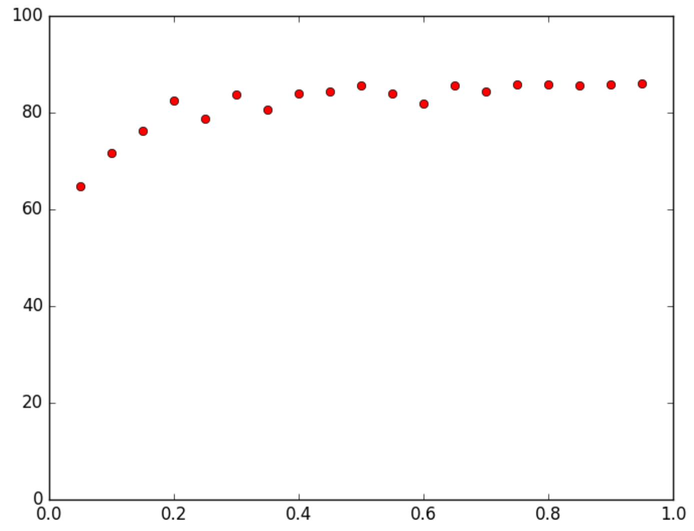
7. Test the unpruned and pruned trees on the validation set. What are the accuracies of each tree? Explain the difference, if any.

The accuracy of the unpruned tree is 85.2564102564% and the accuracy of the pruned tree is 86.1111111111%. The accuracy of the pruned tree is better the accuracy of unpruned tree because by pruning the decision tree, we reduce the risk of over-fitting to the training data.

8.Create learning curve graphs for both unpruned and pruned trees. Is there a difference between the two graphs?



Unpruned



Pruned.

The curve of pruned tree converges earlier than the graph of unpruned tree.

9. Which tree do you think will perform better on the unlabeled test set? Why? Run this tree on the test file and submit your predictions as described in the submission instructions.

Pruned tree. Because by pruning, we will avoid over-fitting which will lower the accuracy.

10. Which members of the group worked on which parts of the assignment?

Members of our group: Junhan Liu, Guixing Lin, Siyu Zhang

We worked on the ID3 file together.
Guixing Lin worked on the node file.
Junhan Liu worked on the pruning file.
Siyu Zhang worked on the predictions file and the graph file.

11. Bonus: This assignment used Information Gain Ratio instead of Information Gain (IG) to pick attributes to split on, which is expected to boost accuracy over IG. We also used a limited step side for numeric attributes instead of testing all possible attributes as split points. Were these good model selections? Try using plain IG and see if this impacts validation set accuracy. Likewise, try testing all numeric split points (doing so efficiently will probably require writing new code, rather than just setting steps = 1), and evaluate whether this improves validation set accuracy.

They are both good.

1. When using IG, the continuous attributes will have more values, and its IG will be a lot bigger than discrete attributes. Because the computation of IG will favor the attribute with more value (I read some papers about it before). This intuition is same as our experiment.
2. We also tried testing all numeric split points and found out that this decreases validation set accuracy but the running time is shorter.