1. How did you handle missing attributes in examples?

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.

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

4. How did you implement pruning?

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.

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

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

8. Create learning curve graphs for both unpruned and pruned trees (include the learning curves in your pdf document for the homework). Is there a difference between the two graphs?

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.

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

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.

1. For numeric attributes, missing attributes were replaced by the average value of the given attribute. For nominal attributes, missing attributes were replaced by the most frequently occurring class of the given attribute.

2. The generated output is in DNF.txt.

3.

(oppnuminjured < 3.0 ^numinjured < 2.0 ^oppnuminjured < 2.0 ^numinjured >= 1.0 ^oppnuminjured < 1.0 ^oppnuminjured >= -2.0 ^weather = -1 ^oppstartingpitcher = 5 ^homeaway = 1 ^startingpitcher = 1 )

If number of injured players on the other team is less than 3, number of injured players is less than 2 and greater than or equal to 1, number of injured players on the other team is less than 1 and greater than or equal to -2, weather is -1, opposing starting pitcher is 5, homeaway is 1 and our starting pitcher is 1, then winner is 1.

4. We used Reduced Error Pruning. Starting from the root node, we used depth first search to traverse the tree. While traversing, a tree node is replaced by a leaf node with the mode as the label. If replacing it with leaf node resulted in a higher validation accuracy, the node stays as the leaf node. If replacing it resulted in a lower validation accuracy, the node is reverted back to a tree node and the loop iterates.

5. The generated output is in DNF\_prune.txt

6.

7. Our unpruned tree had a validation accuracy of 0.849474789916 and our pruned tree had a validation accuracy of 0.850735294118. The increase in accuracy is very small. However, the size of pruned tree is significantly smaller than that of unpruned tree. The unpruned tree had 182 leaf nodes that correspond to winner = 1. On the other hand, the pruned tree had only 5 leaf nodes that correspond to winner = 1.

8.

9. My hypothesis is that the pruned tree would provide a higher accuracy because it would generalize better to new data (in this case, the test set). The unpruned tree has significantly more splits and leaf nodes and may be susceptible to over-fitting.

10.