EECS349 Machine Learning PS2 Report

Jiawen Ou
Jom239

1. For any missing data, I assigned it the mode of that attribute values of training set if it's nominal and assigned it the average if it's numeric.

2. (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent < 0.423661897448) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.423661897448) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ rundifferential < 47.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ rundifferential >= 47.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ weather = 0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured < -1.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 12.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ temperature < 70.8322085014) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ temperature >= 70.8322085014) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ rundifferential >= 76.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^

rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent < -0.114432049387) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0 ^ opprundifferential < 7.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0 ^ opprundifferential >= 17.0 ^ winpercent < 0.308164876477) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppwinningpercent < 0.332228138397) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppwinningpercent >= 0.332228138397) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppnuminjured >= 1.50169769868 ^ opprundifferential < 15.0)

3. (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent < 0.423661897448)

   The rule above in English: For each instance, the attributes values are checked in the illustrated order. Firstly, the unpruned tree will check whether numinjured is less than 3.0. If so, it will check whether numinjured is less than 1.0 in the next step. In the similar way, if the instance satisfies all conditions (oppwinningpercent is less than 1.0, numinjured is equal to 0.0, oppwiningpercent is less than 0.616717019807 and oppwinningpercent is less than 0.423661897448) one by one, the classified output label will be set to 1.

4. To prune the decision tree, I use the reduced error pruning. To be more specific, I consider each node for pruning from leaves to root. By DFS, I move to the leaf nodes, and go back to the upper level by recursion. For each node, removing the subtree at that node, make it a leaf and assign the label to 0 or 1. A node is removed if the resulting tree performs no worse then the original, and the label is set to the higher one, which makes sure the node is removed only if the removal isn't harmful.

5. (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent < 0.423661897448) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.423661897448) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ rundifferential < 47.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ rundifferential >= 47.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ weather = 0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured < -1.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 12.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ temperature < 70.8322085014) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^

rundifferential >= 71.0 ^ temperature >= 70.8322085014) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ rundifferential >= 76.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent < -0.114432049387) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential < 17.0) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential >= 17.0 ^ winpercent < 0.308164876477) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppwinningpercent < 0.332228138397) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent < 0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppwinningpercent >= 0.332228138397) v (oppnuminjured < 3.0 ^ numinjured < 1.0 ^ oppnuminjured < 1.0 ^ numinjured = 0.0 ^ oppwinningpercent <

0.616717019807 ^ oppwinningpercent >= 0.616717019807 ^ numinjured = -2.0 ^ rundifferential >= 59.0 ^ numinjured = -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0 ^ opprundifferential < 24.0 ^ rundifferential < 71.0 ^ rundifferential >= 71.0 ^ opprundifferential >= 24.0 ^ opprundifferential < 36.0 ^ opprundifferential >= 36.0 ^ numinjured >= 1.0 ^ oppwinningpercent < 0.187070552449 ^ oppwinningpercent >= -0.114432049387 ^ opprundifferential >= 17.0 ^ oppwinningpercent >= 0.187070552449 ^ numinjured < 2.0 ^ oppnuminjured < 1.50169769868 ^ oppnuminjured >= 1.50169769868 ^ opprundifferential < 15.0)

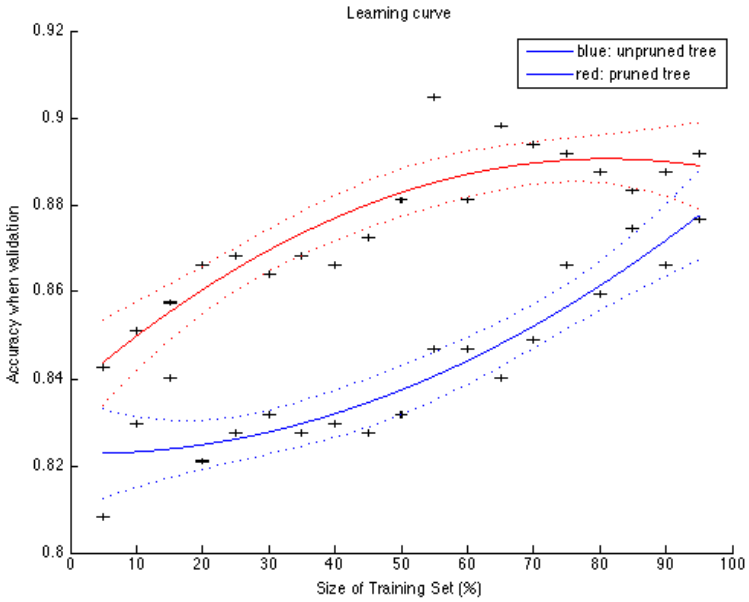
6. I wrote a function to count the non-leaf node and the number of it before the pruning is 141, the number after it is 108. Thus, the difference is 33.

7.

	Unpruned Tree	Pruned Tree
Accuracy	87.5315%	88.4768%

The main reason to the difference is that, for the unpruned tree, it is generated by the whole training set and it will overfitting to the training set. In terms of the pruned tree, it removes the useless or unnecessary subtree and relieve the overfitting problem. Thus, the pruned tree performances better than the unpruned one.

8.



Both lines increase with the increasing size of training data. The main difference is that the pruned tree always outperforms the unpruned tree. Besides, the increasing speed of the pruned tree is becoming slower while that of the unpruned tree is becoming faster.

9. I think the pruned tree will perform better. Because after pruning, the overfitting problem is kind of solved and there are less unusual cases in the pruned tree. The output of the prediction function is in the CSV file.

10. While all three teammates have made significant contributions to the project, each individual has completed some functions solely. Specifically, Siran Liu has completed the functions of check_homogenous, mode, gain_ratio_nominal, split_on_numeric, Yuling Shen has completed entropy and I have completed gain_ratio_numeric, split_on_nominal, clean_data and pick_best_attribute. The rest of the functions/modules, which are relatively more complicated, are written and debugged as a team of 3.

11.

Accuracy	Unpruned	Pruned
Gain Ratio	87.5315%	88.4768%
Information Gain	85.3517%	86.5651%

Technically, gain ratio is better than information gain because it takes number and size of branches into consideration and can reduce the bias.

In terms of the steps, when I increasing the steps, the accuracy improvement will increase and after a point, it decreases. It seems there is a threshold of steps. When less than the threshold, increasing the steps will solve the overfitting problem. When greater than it, increasing the steps will result in a loose tree and decrease the accuracy. Set Step to 1 usually not the best choice because of overfitting. In this case, we found 500 may be the approximate best one.