Questions

Parameters:
limit_splits_on_numerical = 5
limit_depth = 20

1. **How did you handle missing attributes in examples?**

If the missing attribute is nominal, it will be assigned to the value that appears most in this attribute. If the missing attribute is numeric, it will not be handled. Different methods is tried to set the numerical missing value to average, median and so on. It comes out that leaving missing attributes to none could achieve a higher accuracy.

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.**

```
(oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured
< -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame < 2.0 ^
temperature >= 47.332 ^ oppstartingpitcher = 2 ^ homeaway = 0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame <
2.0 ^ temperature >= 47.332 ^ oppstartingpitcher = 2 ^ homeaway = 1 ^ weather = 1)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame <
2.0 ^ temperature >= 47.332 ^ oppstartingpitcher = 3 ^ startingpitcher = 1)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame <
2.0 ^ temperature >= 47.332 ^ oppstartingpitcher = 4 ^ temperature < 70.653 ^ oppnuminjured >= -2.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame <
2.0 ^ temperature >= 47.332 ^ oppstartingpitcher = 4 ^ temperature >= 70.653)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame <
2.0 ^ temperature >= 47.332 ^ oppstartingpitcher = 5 ^ temperature >= 54.354)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential < 13.0 ^ temperature < 73.392 ^ oppwinningpercent < 0.506)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential < 13.0 ^ temperature < 73.392 ^ oppwinningpercent >= 0.506 ^
temperature < 69.029 ^ oppwinningpercent >= 0.7)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential < 13.0 ^ temperature < 73.392 ^ oppwinningpercent >= 0.506 ^
temperature >= 69.029)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame < 2.0 ^ opprundifferential < 58.0 ^
oppdayssincegame < 1.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame < 2.0 ^ opprundifferential < 58.0 ^
oppdayssincegame >= 1.0 ^ winpercent >= 0.501 ^ rundifferential >= 48.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame >= 2.0 ^ oppwinningpercent < 0.598 ^
oppwinningpercent < 0.576 ^ winpercent < 0.156)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^
oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame
>= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame >= 2.0 ^ oppwinningpercent < 0.598 ^
oppwinningpercent < 0.576 ^ winpercent >= 0.156 ^ opprundifferential < 66.0 ^ opprundifferential < 33.0 ^ winpercent <
0.542)
```

v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame >= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame >= 2.0 ^ oppwinningpercent < 0.598 ^ oppwinningpercent < 0.576 ^ winpercent >= 0.156 ^ opprundifferential < 66.0 ^ opprundifferential < 33.0 ^ winpercent >= 0.542 ^ dayssincegame < 3.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame >= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame >= 2.0 ^ oppwinningpercent < 0.598 ^ oppwinningpercent < 0.576 ^ winpercent >= 0.156 ^ opprundifferential < 66.0 ^ opprundifferential >= 33.0 ^ winpercent < 0.524 ^ oppdayssincegame < 4.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured < -1.0 ^ opprundifferential < 76.0 ^ oppwinningpercent < 0.868 ^ rundifferential < 89.0 ^ dayssincegame >= 2.0 ^ winpercent < 0.831 ^ opprundifferential >= 13.0 ^ oppdayssincegame >= 2.0 ^ oppwinningpercent < 0.598 ^ oppwinningpercent < 0.576 ^ winpercent >= 0.156 ^ opprundifferential >= 66.0 ^ rundifferential < 53.0 ^ rundifferential >= 45.0)

3. **Explain in English one of the rules in this (unpruned) tree.**
   If the number of injured player in opponent team is less than -1.0 and number of injured player in our team is less than -0.1 and run differential for second team is less than 76.0 and winning percentage of opposing team is less than 0.567 and days since our team's last game is greater and equal to 2.0 and winning percentage of our team is less than 0.831 and days since opposing team's last game is greater and equal to 2.0 and winning percentage of out team is greater and equal to 0.156 and run differential of second team is greater and equal to 66.0 and run differential for first team is less than 53.0 and run differential for first team is greater and equal to 45.0, our team will win.

4. **How did you implement pruning?**
   Consider each node. Pruning means removing the subtree at that node, make it a leaf and assign the most common class at that node. A node is removed if the resulting tree performs no worse than the original one. Nodes are removed iteratively, choosing the node whose removal most increases the decision tree accuracy. Pruning continues until further pruning is harmful.

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.**
   If the step is set to 400, we get:

(oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured < -1.0 ^ opprundifferential < 76.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured < 1.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ oppnuminjured >= -1.0 ^ oppnuminjured >= 1.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured >= 1.0)
v (oppnuminjured < 3.0 ^ numinjured >= 2.0)

   If the step if changed as the changing of size of data set, e.g. if data set has more than 400 examples, set step to (length of data set / 400):

(oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured < -2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent < 0.288 ^ rundifferential >= 19.0)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured < -2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >= 0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^ oppwinningpercent < 0.004 ^ rundifferential < 83.0 ^ oppnuminjured < 0.0 ^ temperature < 63.728 ^ weather = 1)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured < -2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >= 0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^ oppwinningpercent < 0.004 ^ rundifferential < 83.0 ^ oppnuminjured >= 0.0 ^ opprundifferential < 81.0 ^ temperature < 60.062)
v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured < -2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >= 0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^

```
oppwinningpercent < 0.004 ^ rundifferential < 83.0 ^ oppnuminjured >= 0.0 ^ opprundifferential < 81.0 ^ temperature >=
60.062)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^
oppwinningpercent >= 0.004 ^ rundifferential < 86.0 ^ oppwinningpercent >= 0.101 ^ opprundifferential < 21.0 ^
oppdayssincegame < 4.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^
oppwinningpercent >= 0.004 ^ rundifferential < 86.0 ^ oppwinningpercent >= 0.101 ^ opprundifferential < 21.0 ^
oppdayssincegame >= 4.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^
oppwinningpercent >= 0.004 ^ rundifferential < 86.0 ^ oppwinningpercent >= 0.101 ^ opprundifferential >= 21.0 ^
oppwinningpercent < 0.103)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent < 0.941 ^
oppwinningpercent >= 0.004 ^ rundifferential >= 86.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential < 87.0 ^ winpercent >= 0.941)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential < 96.0 ^ opprundifferential < 98.0 ^ opprundifferential >= 87.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent < 0.322 ^ winpercent < 0.945 ^ winpercent >=
0.288 ^ rundifferential >= 96.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent >= 0.322 ^ oppnuminjured < -1.0 ^
oppwinningpercent >= 0.331)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent >= 0.322 ^ oppnuminjured >= -1.0 ^
oppnuminjured < 0.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured < 1.0 ^ numinjured < -1.0 ^ numinjured <
-2.0 ^ opprundifferential < 102.0 ^ winpercent < 1.145 ^ oppwinningpercent >= 0.322 ^ oppnuminjured >= -1.0 ^
oppnuminjured >= 0.0)
 v (oppnuminjured < 3.0 ^ numinjured < 2.0 ^ oppnuminjured < 2.0 ^ numinjured >= 1.0)
 v (oppnuminjured < 3.0 ^ numinjured >= 2.0)
```

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

A node that has children is considered as a split.

The pruned tree has much less number of splits than the unpruned tree.

If the step is set to 400, the pruned tree has 13 splits and the unpruned tree has 49 splits.

If the step if changed as the changing of size of data set, e.g. if data set has more than 400 examples, set step to (length of data set / 400), the pruned tree has 33 splits and the unpruned tree has 49 splits.

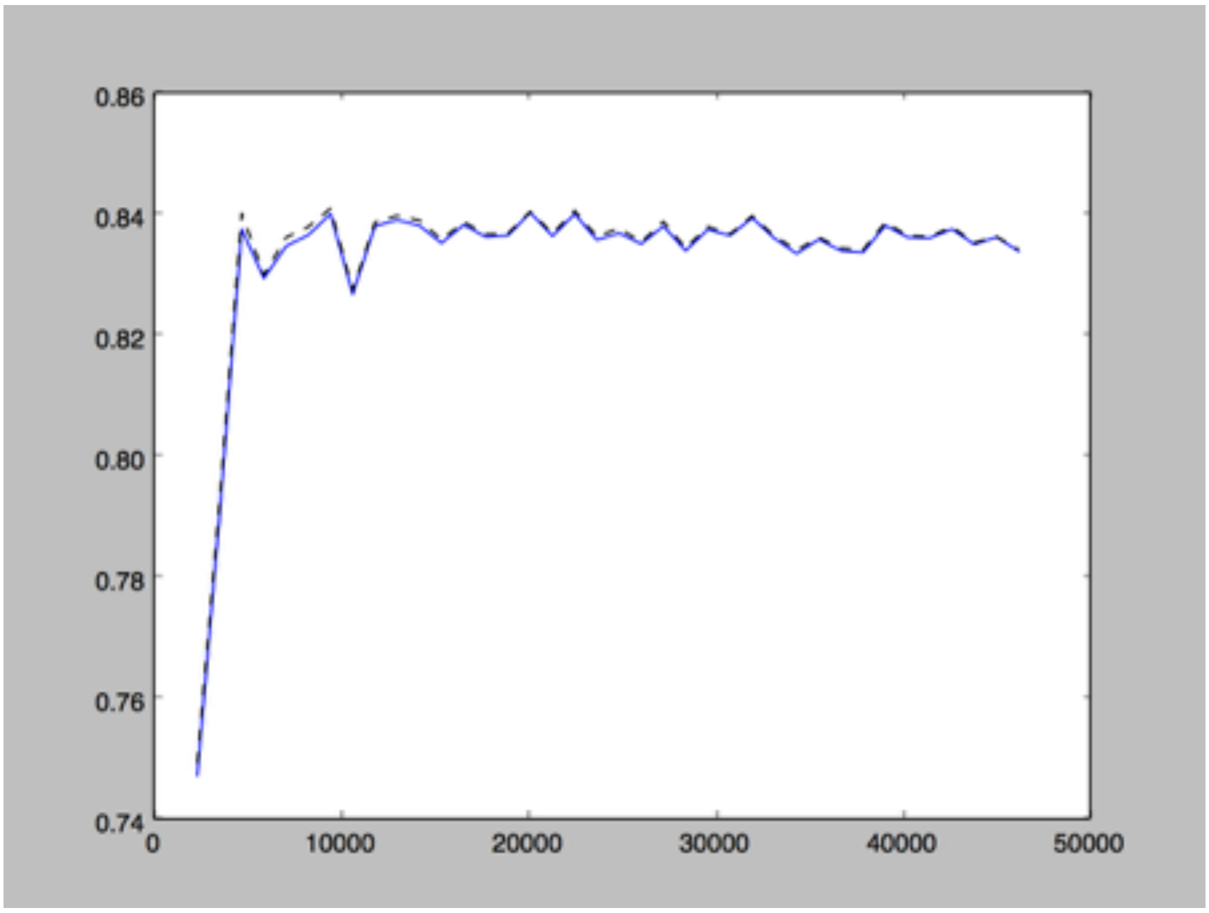
7. Test the unpruned and pruned trees on the validation set. What are the accuracies of each tree? Explain the difference, if any.

The accuracy on unpruned tree is 83.426%, the accuracy on pruned tree is 83.447%. The accuracy increases after pruning, which means pruning benefits the performance of our tree.

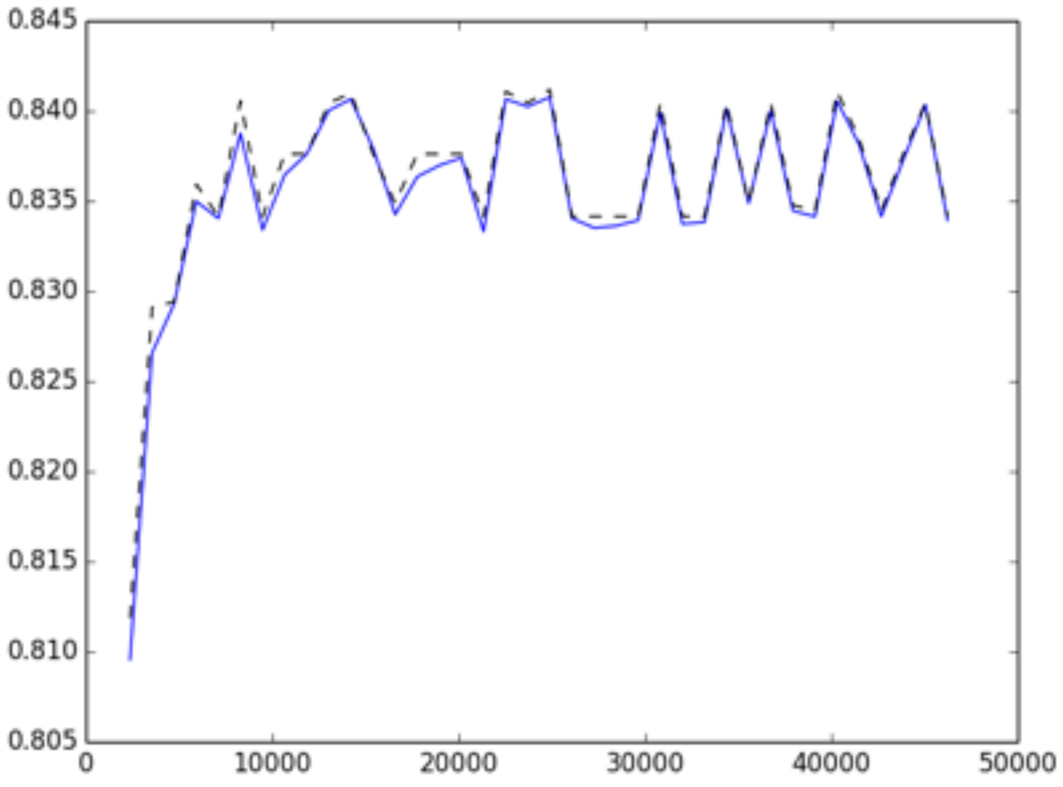
8. Create learning curve graphs for both unpruned and pruned trees (include the learning curves in your pdf document for the homework). Is there a difference between the two graphs?

X-axis means the accuracy, Y-axis means the size of the training set.

Step = 400. Iterations = 3. Lower bound = 0.05, upper bound = 1, increment = 0.025.



Step = 1, if size of dataset is less than 200; Step = (size of dataset / 200). Iterations = 1.



The dotted line shows the accuracies on the pruned tree and the solid line shows the accuracies on the unpruned tree. The pruned tree always performs slightly better than the 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.**

   I think unpruned tree will perform better on the unlabelled test set. Because pruning can avoid over-fitting.

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

    I copied pruning.py from Rui Liao and some methods in ID3.py from Yiyi Ren.

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.**

    Using IG, the result is:
    Accuracy on validation set before pruning: 81.410%
    Accuracy on validation set after pruning: 81.420%
    Using Gain Ratio could avoid splitting data into too many small categories.