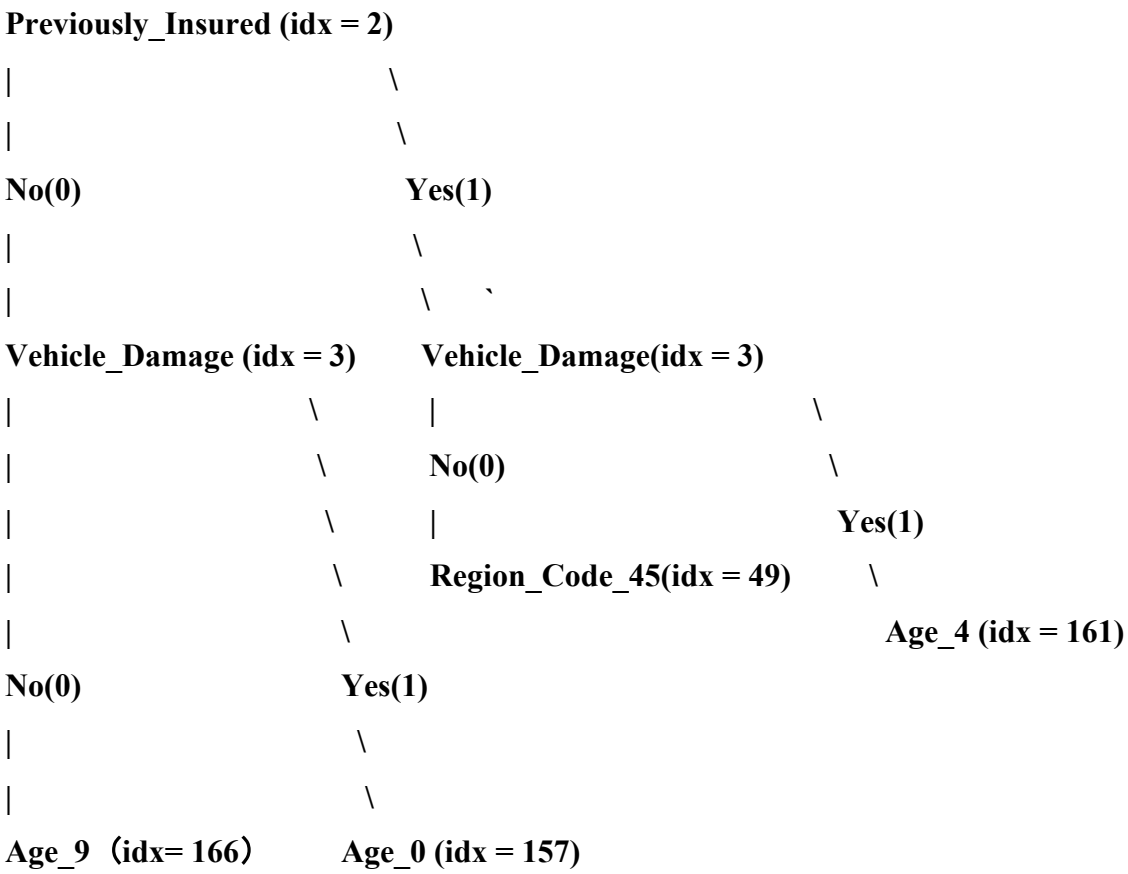
Chenyun Yu
Prof. Fern
CS-534 MACHINE LEARNING
Implementation 4 Reporter

Part 1: Decision Tree.

(a) What are the first three splits selected by your algorithm? This is for the root and the two splits immediately beneath the root. What are their respective information gains?
In my experience, the first three layers of the Decision Tree is as follows:



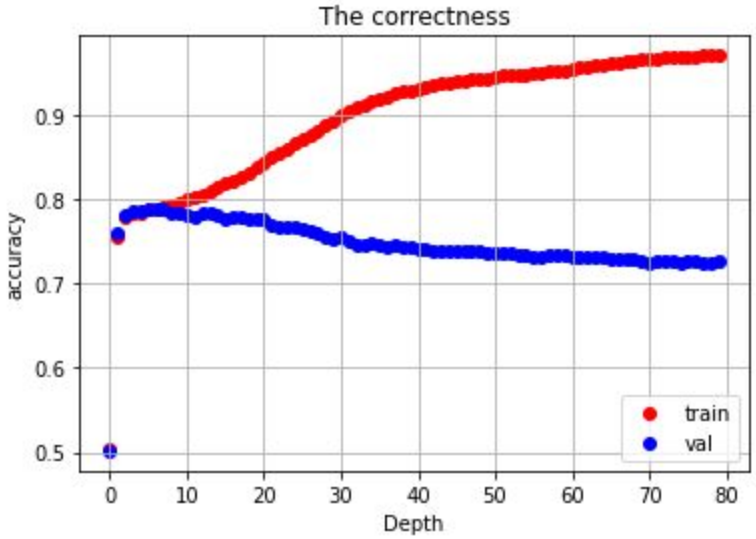
Previously_Insured (idx = 2) - No(0)->Vehicle_Damage (idx = 3):
The information gained is 0.089
Previously_Insured (idx = 2) - Yes(0)->Vehicle_Damage (idx = 3):

The information gained is 0.95

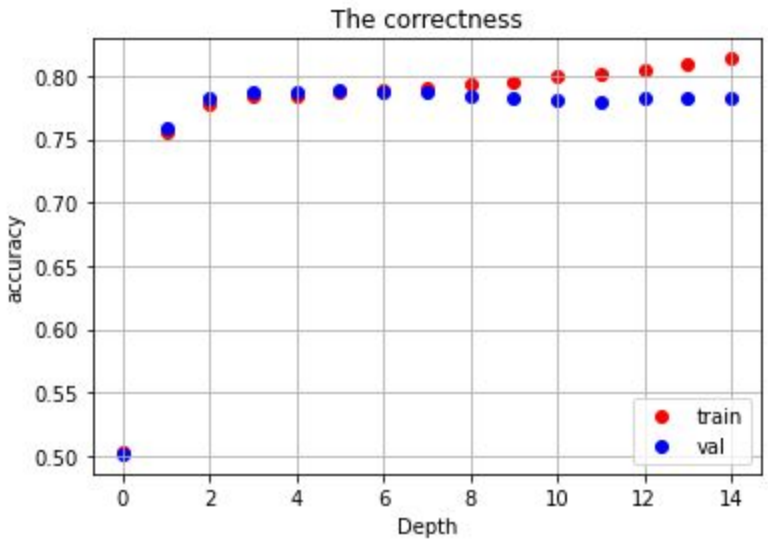
Vehicle_Damage (idx = 3) - No(0)->Age_9（idx= 166):
The information gained is 0.1008
Vehicle_Damage (idx = 3) - No(0)-> Age_0 (idx = 157):
The information gained is 0.0307

Vehicle_Damage (idx = 3) - Region_Code_45(idx = 49):
The information gained is 0.016
Vehicle_Damage (idx = 3) - Yes(0)->Age_4 (idx = 161):
The information gained is -0.194

(b) Evaluate and plot the training and validation accuracies of your trees as a function of max. When do you see your tree start overfitting?



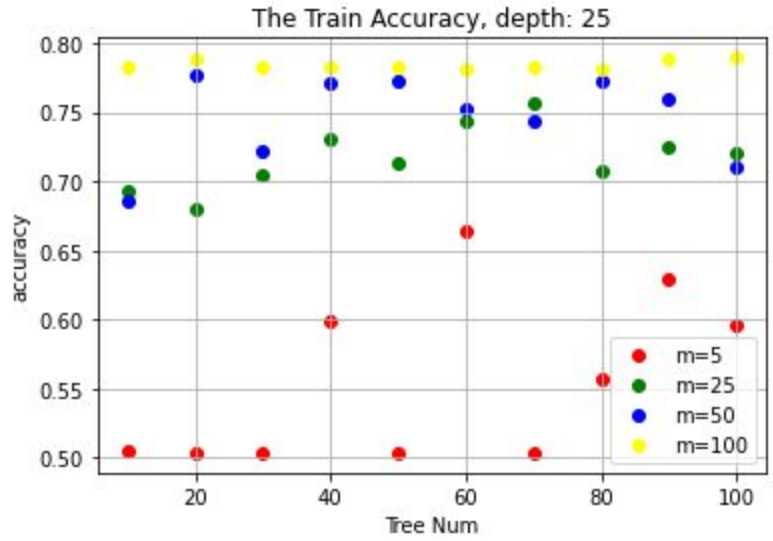
I test the max depth from 0 to 80, it is clear that the train correctness is still growing with the depth, a larger depth also will harm the validation correctness. This is a typical overfitting phenomenon.
The starting point of overfitting is early than I imagined, with this small range test of depth, I believe the overfitting starts at 7 or 8 depth.

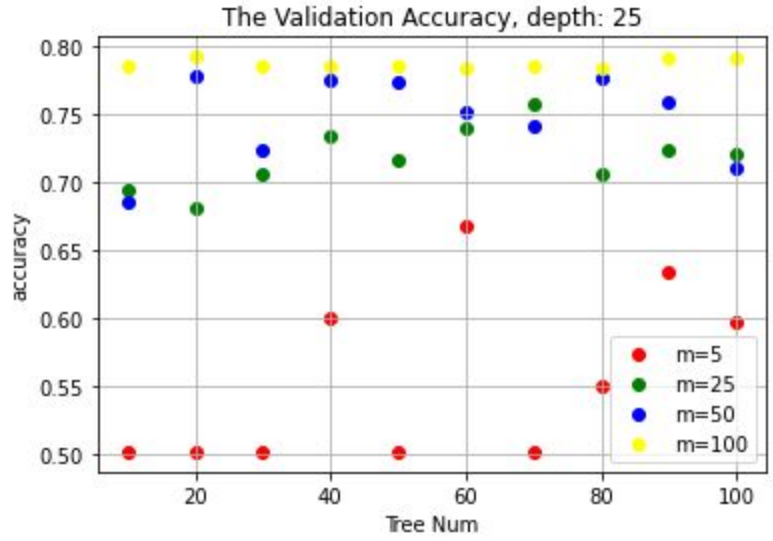


Part 2: Random forest

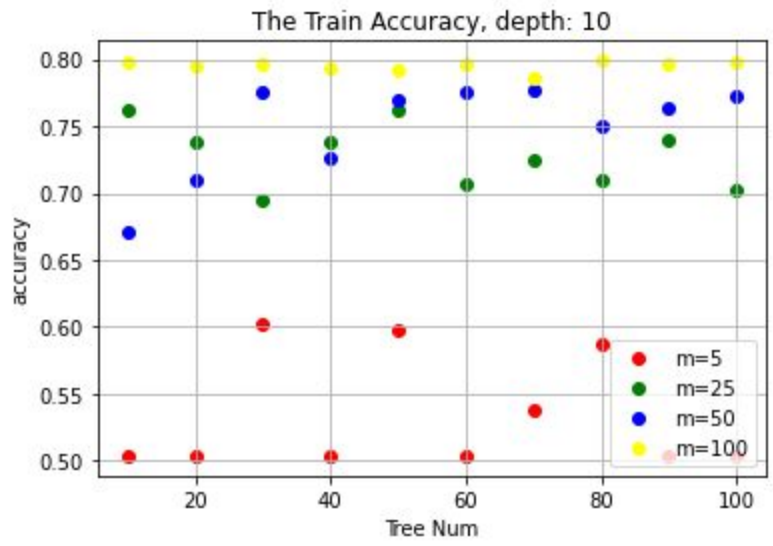
For each dmax value, create two figures, one for training accuracy and one for validation accuracy. For the training accuracy figure, it will contain four curves, each showing the training accuracy of your random forest with a particular m value as we increase the ensemble size T= 10,20, ...,100. That is, plot the training accuracy (y-axis) as a function of the ensemble size T(x-axis), for each m value. Be sure to use different colors/lines to indicate which curve corresponds to which m value, and include a clear legend for your figure to help the readability. Repeat the same process for validation accuracy. Compare your training curves with the validation curves, do you think your model is overfitting or underfitting for particular parameter combinations? And why?

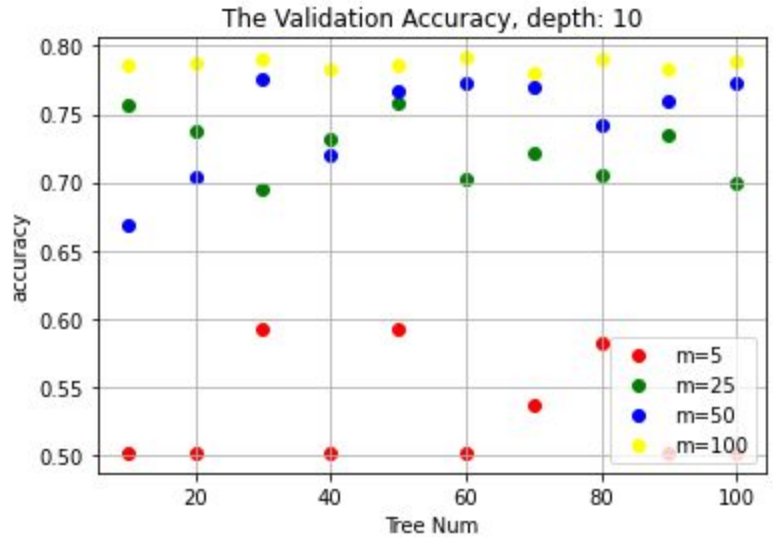
Depth = 5 (The figures notation is wrong by careless)



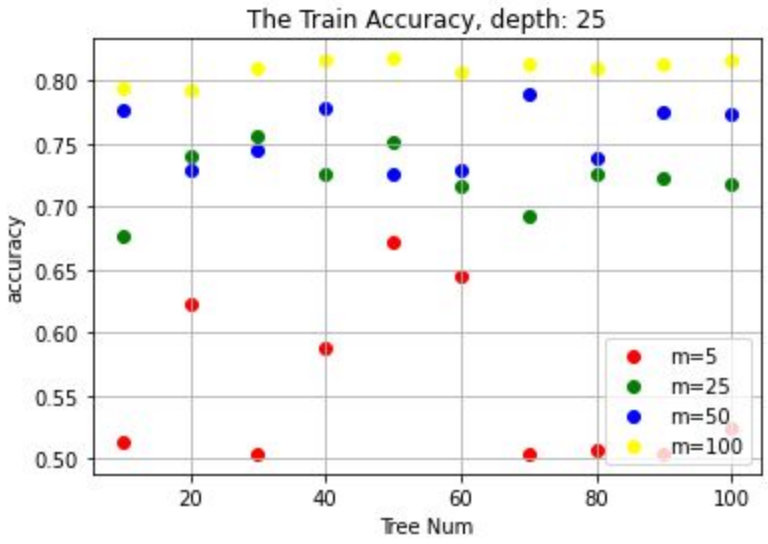


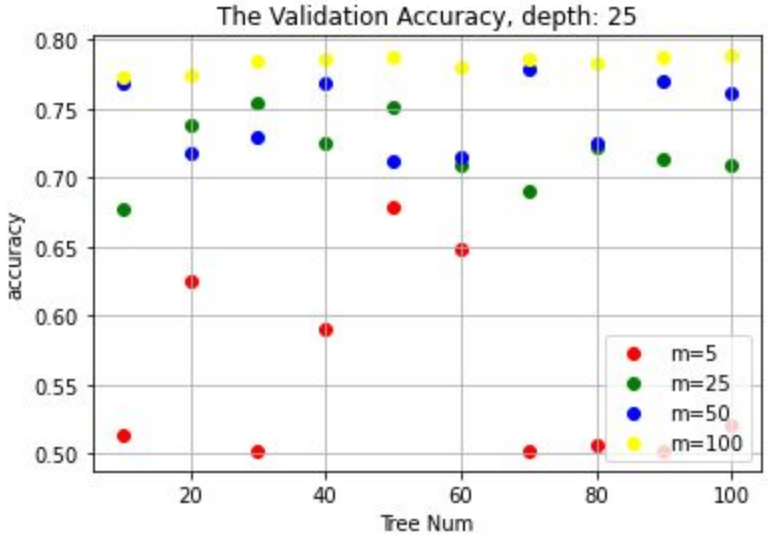
Depth = 10


The Train Accuracy, depth: 10
accuracy
Tree Num
m=5
m=25
m=50
m=100


The Validation Accuracy, depth: 10
accuracy
Tree Num
m=5
m=25
m=50
m=100

Depth = 25





I think there is some overfitting in depth = 25, but this overfitting does not influence the validation accuracy. I think the main reason caused this phenomenon is that the probability of a single tree has the overfitting absolutely will increase with the depth increasing, the majority vote may pass some specific instance in train data.

b) For each dmax value, discuss what you believe is the dominating factor in the performance loss based on the concept of bias-variance decomposition. Can you suggest some alternative configurations of random forest that might lead to better performance for this data? Why do you believe so? Are there any issues inherent with the data you can find that make the performance increase difficult

1. A small **m** is not a good choice for the random forest because some factors may be dominated, for example, in this experiment, the **Previously_Insured** is a dominating factor, When the result of **Previously_Insured** is 1, the uncertainty drops by 95% (0.99 -> 0.04). If we used a small **m,** the probability of a single tree that has the dominating factors is also very low, even if you expand the number of trees, it is still impossible to own the majority.
2. The number of trees should not be as much as possible, especially when the number of factors is large. When the number factor is large, the probability of a single tree has similar features choice will also increase, which makes the forest "proportionally increases"; for example, for a specific instance, the majority votes result is 10-15, after the "proportionally increases", the result of the votes becomes 21-29, the again increasing, it becomes 43 - 57. Proportionally increasing the similar behavior leaners will not influence the majority votes, so it will not benefit accuracy.