Submitted by: Oluwasegun Somefun, somefuno@oregonstate.edu

AI534 — IA4 Homework Report Due Dec 3rd 11:59pm, 2021

1 Introduction

This report is on Decision Tree and Ensembles for Mushroom Classification, with respect to the Implementation Assignment 4.
Team Name: Oluwasegun Ayokunle Somefun

2 Part 1: Decision Tree (50pts)

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?

The first 3 splits selected by the decision algorithm using maximum mutual information gain split criterion are highlighted in Table 1.

Table 1: First 3 node splits

Node	feature	information gain
Root Split	odor=n	**0.54**
True Branch Split	bruises?=f	**0.4**
False Branch Split	spore-print-color=r	**0.1**

b. Evaluate and plot the training and validation accuracies of your trees as a function of $d_{\max}$ ranging from 1 to 10. At which depth does the train accuracy reaches to 100%? Do you observe any overfitting?

At depth $d_{\max} = 6$, the train accuracy reaches 100%. This together with the validation accuracy of 100% is reported in Figure 1. Technically, there was no observation of overfitting, since the validation accuracy does not reduce even as $d_{\max} > 6$ increases. However, this would imply overtraining, which could lead to overfitted decisions, depending on the data distribution.

3 Part 2: Random Forest (35pts)

a. For each $d_{\max}$ value, create two figures, one for training accuracy and one for validation accuracy. The training accuracy figure should contain four curves, each showing the train accuracy of your random

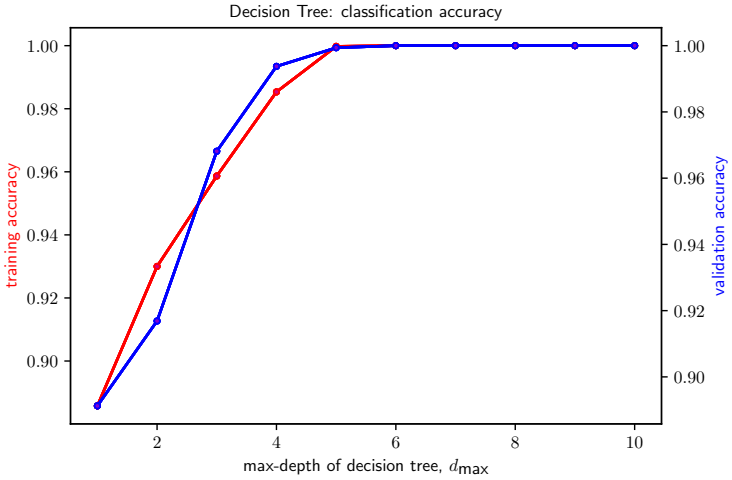

Figure 1: Decision Tree: training/validation class accuracies as a function of maximum depth

forest with a particular m value as a function of T. 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?

The training curves and validation curves are oultined in Figures 2–7. Comparing both curves, we see that for each maximum depth $d_{\max}$ the model performance of the random forest was not consistent and depended more on the number of random subsampled features m compared to the increase in the number of trees T in the forest ensemble. Also, we observe that as expected, both train and validation curves behave similarly, hence indicating random forest leads to low variance models but needs low bias models as base learners. This can be easily observed from the accuracy curves. Specifically, an additional value of m was also added to account for subsampling using the total feature population 117.

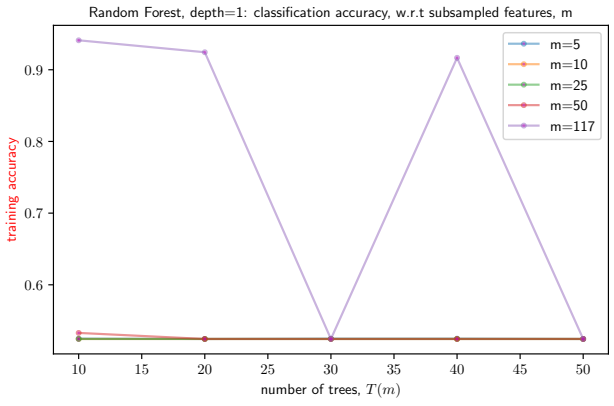

Figure 2: Random Forest $d_{\max} = 1$: Training accuracy

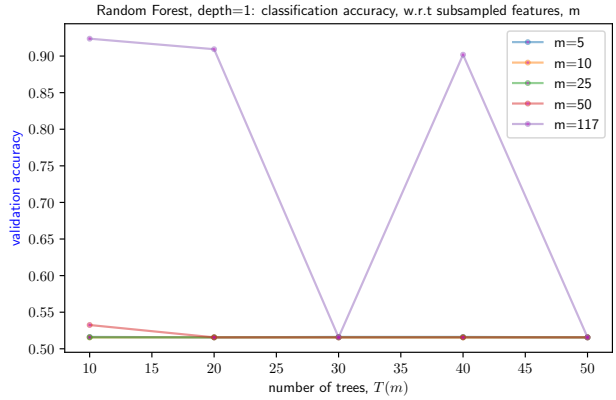

Figure 3: Random Forest $d_{\text{max}} = 1$: Validation accuracy

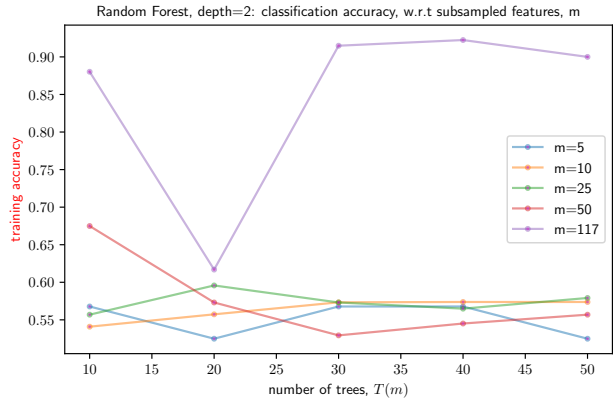

Figure 4: Random Forest $d_{\text{max}} = 2$: Training accuracy

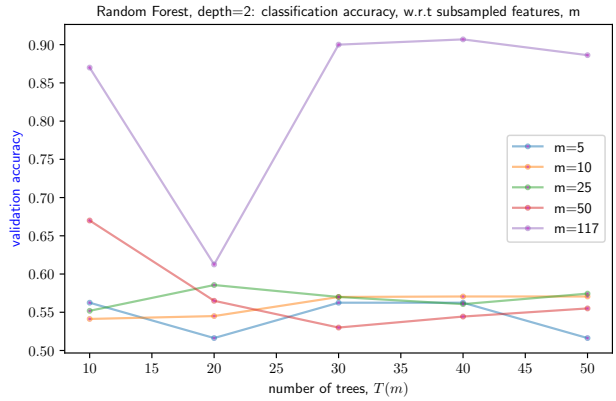

Figure 5: Random Forest $d_{\text{max}} = 2$: Validation accuracy

For $d_{\text{max}} = 1$, see Figures 2–3, observe that the model, although of low variance, generally underfits the data for all m as the number of trees $T > 10$ is increased.

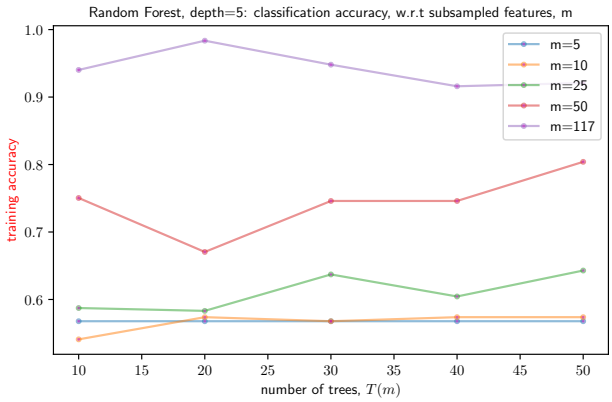

Figure 6: Random Forest $d_{\text{max}} = 5$: Training accuracy

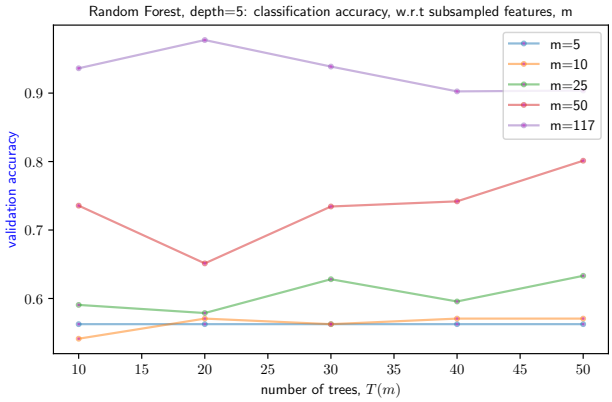

Figure 7: Random Forest $d_{\text{max}} = 5$: Validation accuracy

For $d_{\text{max}} = 2$, see Figures 4–5, observe that as m increases the model accuracy improves on the data better than the previous depth.

For $d_{\text{max}} = 5$, Figures 6–7 show that as m increases the range of model accuracy improves on the data better than the previous lower mamxium depths.

b. For each d_{max} 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?

It was observed that generally, for each d_{max} as m increases, the mean out-of-bag (OOB) estimate (see Table 2) for each ensemble configuration increases. For $d_{\text{max}} = 1$, irrespective of m and T, the model has the highest bias, and a low variance. For $d_{\text{max}} = 2$, variance is low and as m increases, the bias error generally reduces, although the behaviour is unstable as T is increased. For $d_{\text{max}} = 5$, again the variance is low, but this configuration has the lowest bias error compared to the previous depths.

Table 2: mean out-of-bag (OOB) accuracy estimates

d_{max}	m				
	5	10	25	50	117
1	0.52	0.53	0.54	0.55	0.60
2	0.53	0.55	0.57	0.57	0.67
5	0.55	0.56	0.61	0.62	0.80

It can bes seen that for this dataset, to obtain better performance with lower bias error, d_{max} should be set to a suitable value that allows the base tree learner to be strong. Also, the number of sub-sampled features with replacement m, in the **Random forest** should be increased close to the full or total number of features in this data-set, that is: $m = 117$.

4 Bonus Part 3: AdaBoost (20pts) and Kaggle competition (5pts)

a. For each d_{max} value, create a figure showing two curves, showing the accuracy (y-axis) on train and validation of your ensemble as a function of T. 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?

Comparing the training curves with the validation curves in Figures 8-10, neither underfitting nor overfitting can be observed. This is clearly seen as the training and validation accuracies are similar.

This behaviour is expected in boosting, since boosting leads to decision-tree models with a lower bias and lower variance. However, overtraining can be observed as the number of trees in the ensemble increase.

b. For each d_{max} 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 the ensemble that might lead to better performance for this data? Why do you believe so?

From the given Figures 8-10, we observe that for $d_{max} = 1$, as T increases, performance increases, and that the boosted model is of low bias, and low variance. Similarly, for $d_{max} = 2$, the model is of lower bias, and lower variance, and performance improves faster to the optimum at $T = 26$ compared to the previous $d_{max} = 1$ setting. For $d_{max} = 5$, again the the model is of lower bias, and lower variance, and for all T in the given set, performance converges faster to the optimum value of 100% compared to the previous $d_{max} = 2$ setting.

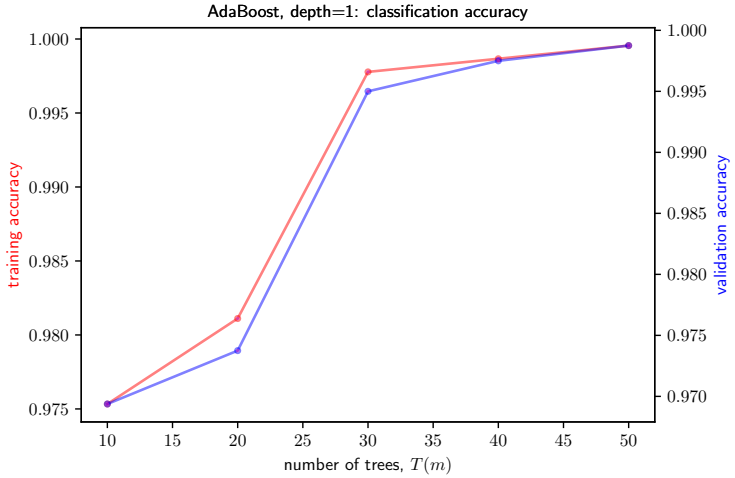

Figure 8: Adaboost: Train and validaition curves for $d_{\max} = 1$

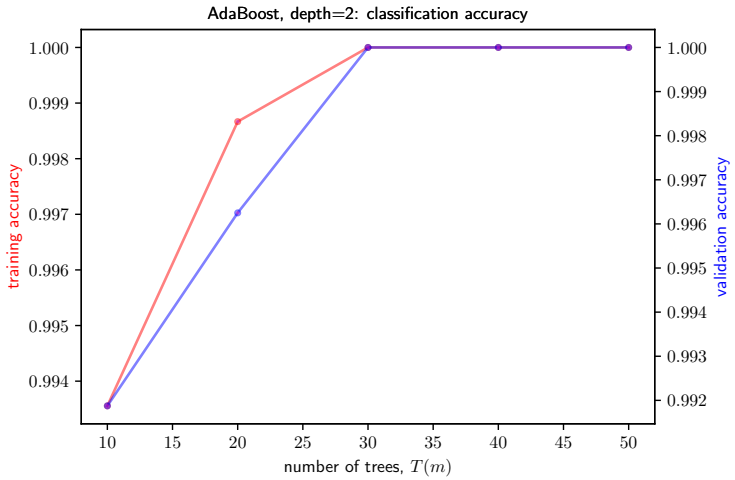

Figure 9: Adaboost: Train and validaition curves for $d_{\max} = 2$

However, we observe that on the train and validation sets, it is possible that overtraining occurs: at $T > 5$ taking $\sim 3.3\,s$ for $d_{\max} = 5$, while for $d_{\max} = 2$, similar pattern is observed at $T > 26$ taking $\sim 5.4\,s$, and for $d_{\max} = 1$, this is observed at $T > 110$ taking $\sim 15.4\,s$. That is, it was observed that, there is no benefit or loss from adding more trees to the ensemble given these thresholds. In simple words, adding more trees past these thresholds make the ensemble bloated and increase computation time.

Therefore, for this data-set, to obtain the best decision, in the fastest time possible, with the lowest performance loss, using $d_{\max} \in 1, 2, 5$, the ensemble can be minimally configured using $T = 5$ and $d_{\max} = 5$. This configuration was used for the **Kaggle Competition**.

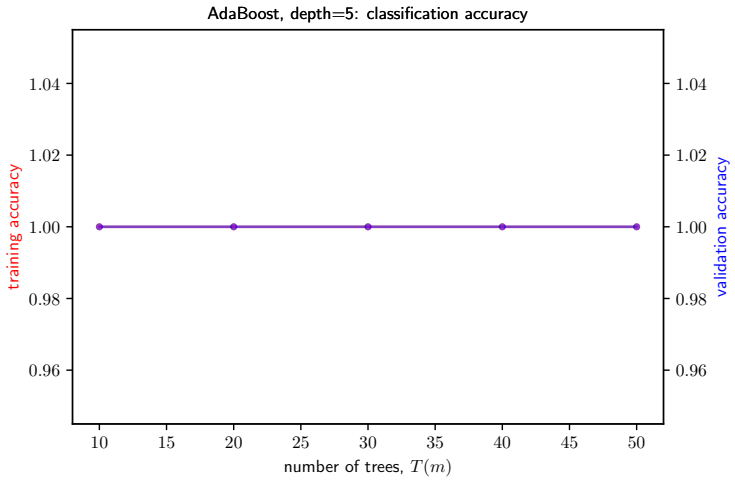

Figure 10: Adaboost: Train and validaition curves for $d_{\text{max}} = 5$

Generally, we observe that for **Adaboost**, the dominating factor in the low-bias low-variance performance is the number of trees T in the ensemble, followed by the d_{max} of each tree.