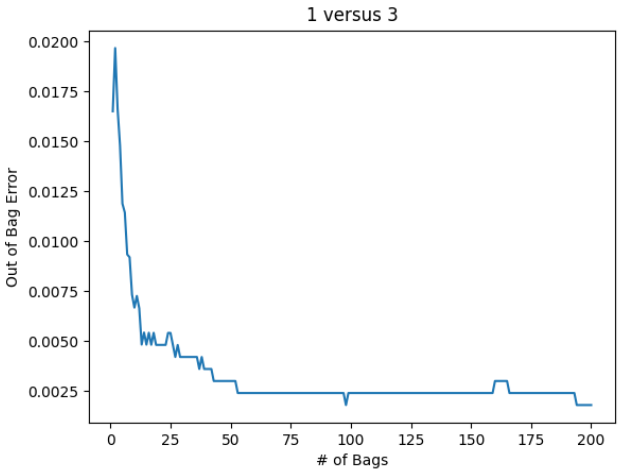
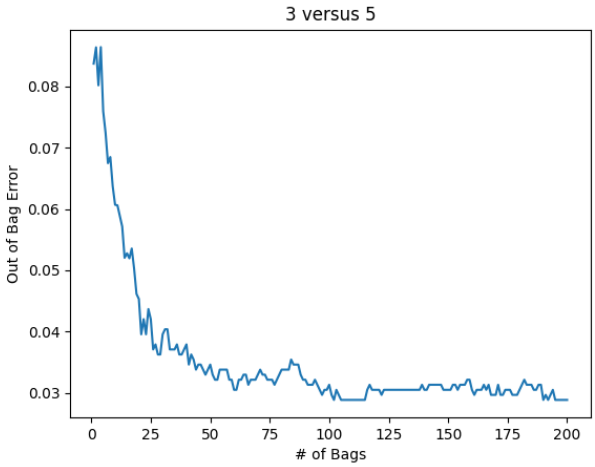
1.




| | $E_{OOB}$ 200 TREES | $E_{TEST}$ SINGLE TREE | $E_{TEST}$ 200 TREES |
|---|---|---|---|
| **1 VERSUS 3** | 0.002 | 0.016 | 0.012 |
| **3 VERSUS 5** | 0.029 | 0.110 | 0.074 |

Before even looking at the error, we can predict that the 1 versus 3 error will be less than the 3 versus 5 error. Most 3's have dark pixels where no 1's will have any. This creates an easier problem for even a single decision tree as they can learn to just look at specific pixel locations for intensity and classify based on that. The 3 versus 5 problem is a bit more complex, because 3's and 5's have pixels in a lot of the same areas and have some of the same shaping (curved). Thus we expect the error for 1 versus 3 to be less in all cases compared to the 3 versus 5 error.

In both problems, increasing the number of bags decreased the out-of-bag error and the testing error. This is because the aggregation of many high-variance, low-bias trees arrives at a lower-variance, low-bias result. The main improvement was seen between 0 and 25 trees, with improvement stopping around 50 bags for the 1 versus 3 problem and 100 for the 3 versus 5 problem.

The out-of-bag error and test error scale together for both problems, with the out-of-bag error consistently being lower than the test error. The test error and out-of-bag error should follow the same patterns and be in the same ballpark since they both make unbiased predictions—the out-of-bag error just uses less trees on average.

2. $H(D) = \sum_{i=1}^{k} P_i \log_2\left(\frac{1}{P_i}\right) =$

$$\frac{3}{5}\log_2\left(\frac{5}{2}\right) + \frac{3}{5}\log_2\left(\frac{5}{3}\right) = 0.97$$

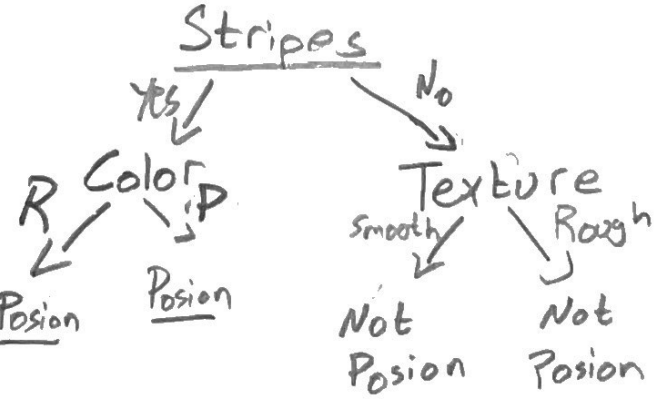
$$Gain(D, Color) = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)$$
$$= H(D) - \frac{4}{5}H(Purple) - \frac{1}{5}H(Red)$$
$$H(Purple) = \frac{2}{4}\log_2\left(\frac{4}{2}\right) + \frac{2}{4}\log_2\left(\frac{4}{2}\right) = 1$$
$$H(Red) = 0\log_2(0) + 1\log(1) = 0$$
$$= H(D) - \frac{4}{5}(1) = \boxed{0.1709}$$

$$Gain(D, Texture) = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)$$
$$= H(D) - \frac{3}{5}H(smooth) - \frac{2}{5}H(Rough)$$
$$H(smooth) = \frac{2}{3}\log_2\left(\frac{3}{2}\right) + \frac{1}{3}\log(3) = 0.918$$
$$H(rough) = \frac{1}{2}\log_2(2) + \frac{1}{2}\log(2) = 1$$
$$= H(D) - \frac{3}{5}(0.91) - \frac{2}{5}(1) = \boxed{0.0199}$$

$$Gain(D, Stripes) = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)$$
$$= H(D) - \frac{2}{5}H(No\ Stripes) - \frac{3}{5}H(stripes)$$
$$H(No\ Stripes) = \frac{0}{2}\log_2 0 + \ldots = 0$$
$$H(stripes) = \frac{1}{3}\log_2(3/1) + \frac{2}{3}\log_2 3/2 = 0.9183$$
$$= H(D) - 0 - \frac{3}{5}(0.9182) = \boxed{0.419}$$

(a) Root is Stripes b/c most Information gain.

(b)



3. Depth 0 decision tree

Starting: $D_{t=0}(i)$ $\frac{1}{m}, \frac{1}{m}, \frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}$

$y_t$ $+1, +1, +1, +1, \ldots, -1$

$g_t(i)$ $+1, +1, +1, +1, \ldots +1$

This will return +1 for everything and have $E_{in}^{(D_t)} = 0.2$

For $D_{t+1}$, we want $E_{in}^{(D_{t+1})}(g_t) = 0.5$.

Reweight: $\alpha_0 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_0}{\varepsilon_0}\right) = \frac{1}{2}\ln\left(\frac{1-0.2}{0.2}\right)$

$= 0.6931$

$\gamma = \sqrt{\frac{1-\varepsilon_0}{\varepsilon_0}} = 2$

$Z_0 = \gamma\varepsilon_0 + \frac{1}{\gamma}(1-\varepsilon_0) = 2(0.2) + \frac{0.8}{2} = 0.8$

$$D_1(n) = \frac{1}{Z_0} D_0(n) e^{-\gamma_0 g_0(x_n) y_n}$$

- For negative (misclassified) points

$D_1 = \frac{1}{0.8}\frac{1}{m} e^{-0.6931(1)(-1)}$

$= \frac{2}{0.8m} = \frac{5}{2m}$

- For positive points

$D_1 = \frac{1}{0.8}\frac{1}{m} e^{-0.6931(1)(1)}$

$= \frac{1}{0.8m}\frac{1}{2} = \frac{5}{4m}\left(\frac{1}{2}\right) = \frac{5}{8m}$

The cumulative weight for any m is

Positive: $\frac{1}{2}$

Negative: $\frac{1}{2}$

after one update

- This will make the weighted sum of the points 0 for the next iteration. The learned hypothesis from this will be a random guess, (since the weighted sum is 0) and does not even learn anything about the distribution. Since this process will repeat, a depth 0 decision tree is not a good choice for a weak learner.

4. - Bagging works best on high variance, low bias models like the decision tree. By combining and weighting each tree, bagging is able to reduce the variance.
- Linear regression is a low variance, high bias model. Thus, there are not many gains to be made - the variance is already low. It does not really make sense to try and improve linear regression by decreasing its variance. Further, adding many high bias models together could even add different biases together in the ensamble prediction.

5.

Article Title:  Study finds gender and skin-type bias in commercial artificial-intelligence systems

(a) Summary:

This article details a study inspired when a research group was using a commercial facial recognition program. When they were trying to demonstrate their project, they found they had to, “rely on one of the lighter-skinned team members to demonstrate it”. To further investigate this, a researcher, Joy Buolamwini, began submitting photos of herself to other commercial facial-recognition programs, and they consistently classified her incorrectly, or even failed to detect her as human. This led her to systematically evaluate this problem. First she compiled a set of 1,200 images and coded them based on skin tone. She then applied three different commercial facial-analysis system. She found that error rates were higher for females than for males, and the error rate increased as someone’s skin got darker. For people with the darkest skin, some algorithms had an error of 46.5 and 46.8 percent—almost random guessing.

(b) This issue relates heavily to sampling bias, generalization and overfitting. The task we want the algorithm to learn is to classify faces of all colors as male or female—binary classification. However, Joy found that the algorithms were having trouble generalizing what they had learned to certain types of test data, namely test data of people with dark skin. This implies that the algorithms were trained on a dataset that was sampled in a biased way, one mainly with lighter skinned people, and they over-fit the data towards people with lighter skin. They are good at that type of data they over-fit, but when they need to generalize to other people, they fall short.

(c) To address this, we need to include more diverse data in our training set. This will help the models not over-fit the learning problem to just the data they see. I remember us talking about a “representative” test set or randomly drawn set. A set with mostly lighter skinned people may be convenient but it does not represent the world population nor is randomly drawn. Another issue to fix this problem could be to apply a bit of regularization. If we do this, it’s possible that models will learn more general facial structures (eyes, mouth), instead of more specific patterns associated with lighter/darker skin. This may bring the testing error up for lighter skinned people, but could overall bring the generalization down for the while dataset. One would need to test his to see if it works.