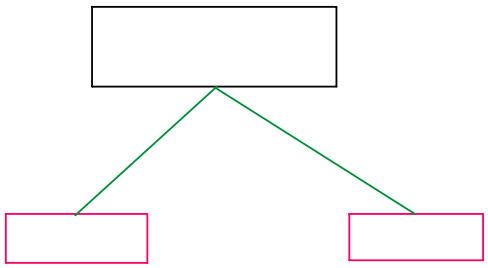
AdaBoost Intuition

Monday, 30 October 2023 6:23 PM

Adaboost makes a forest of trees, but these trees are just depth of 1 with 1 node and two leaves.



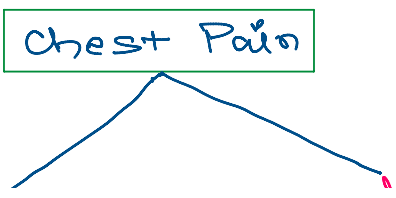
These are called stumps and these are not enough for making accurate classifications.

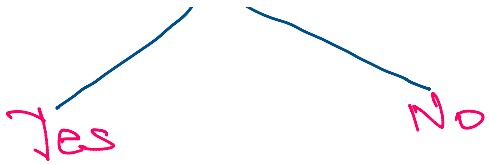
Example →

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
Yes	Yes	205	Yes
No	Yes	180	Yes
Yes	No	210	Yes
Yes	Yes	167	Yes
No	Yes	156	No
No	Yes	125	No
Yes	No	168	No
Yes	Yes	172	No

For above data, a full size decision tree would take advantage of all 3 variables that we measured and make a full multilayered tree.

But a stump can only do this →





So, stumps are technically weak learners.

Now, in RF, every tree has equal weightage in deciding the final classification.

But in Adaboost, some stumps have more weight than the others.

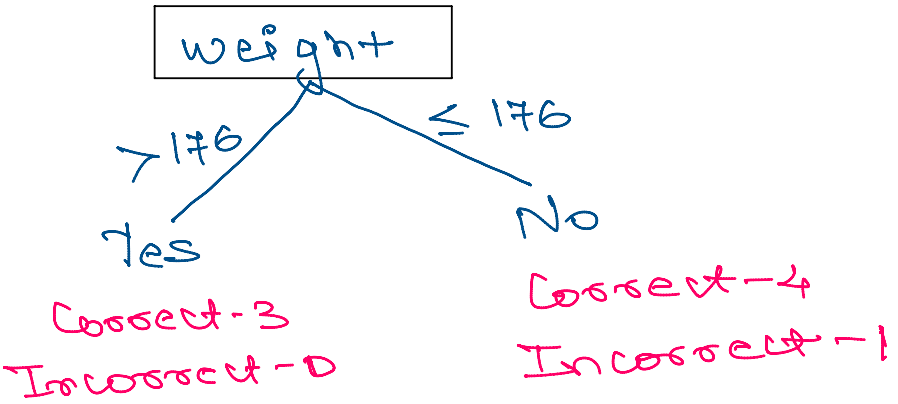
In RF, every tree is made independant of other trees (parallely). But in Adaboost there is an order being followed and the errors made by the first tree influences the second tree. Since second tree has to fix the errors made by the first tree.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

At the start, we give all samples the equal weights.

Now we need to make the first stump in the forest. It is that variable which does

Now ~ ... the forest. It is that variable which does the best job in classifying the samples correctly and that variable will have best gini score.



We found this stump to have best gini score. Now we have to determine how much say this stump will have in final classification.

Only 1 error so Total Error = 1/8.

$$\text{Amount of say (of the above stump)} \quad \alpha = \frac{1}{2}\log\left(\frac{1-\text{total error}}{\text{total error}}\right)$$

$$\alpha = \frac{1}{2}\log 7$$

$$\alpha = 0.97$$

If a stump classifies half of the samples correctly and half of the samples incorrectly then total error = 0.5 which is the same as a random guess. So the amount of say that stump has is 'Zero'.

The ... stumps makes error close to 1, means

If a stump makes error close to 1, means it is largely giving opposite prediction, then amount of say of that stump will be a large negative number, and hence we do 'opposite' of what that stump says.

Now, the next stump that we make has to correct the error made by the previous stump. So before we make the next stump we need to 'increase' the sample weight of the misclassified data point and 'decrease' the sample weight of correctly classified data points.

For correctly classified samples →

$$\text{New weight} = \text{old sample weight} \times e^{-\text{amount of say}}$$

$$= \frac{1}{8} \times e^{-0.97}$$

$$= 0.05$$

For incorrectly classified samples →

$$\text{New weight} = \text{old sample weight} \times e^{\text{amount of say}}$$

$$= \frac{1}{8} \times e^{0.97}$$

$$= 0.33$$

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weights	Normalized Weights
Yes	Yes	205	Yes	1/8	0.05	0.07
No	Yes	180	Yes	1/8	0.05	0.07
Yes	No	210	Yes	1/8	0.05	0.07
Yes	Yes	167	Yes	1/8	0.33	0.49
No	Yes	156	No	1/8	0.05	0.07
No	Yes	125	No	1/8	0.05	0.07
Yes	No	168	No	1/8	0.05	0.07
Yes	Yes	172	No	1/8	0.05	0.07

Now we will use the normalized weights as the new sample weight to make the next stump in the forest.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	New Weights
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Next we will make a new collection of samples that contain duplicate copies of sample which now has largest sample weight.

We take a new empty dataset with same size as the original. Then we start picking samples from original to insert in the new data. Since samples are picked on the basis of weights, the samples with higher weight will be selected more. Hence the new data will look like →

weight will be [illegible]
the new data will look like →

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
No	Yes	156	No
Yes	Yes	167	Yes
Yes	Yes	167	Yes
No	Yes	125	No
Yes	Yes	167	Yes
Yes	Yes	172	No
Yes	Yes	205	Yes
Yes	Yes	167	Yes

Now we will this data for further stumps and hence we will attach the sample weight to the these samples.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
No	Yes	156	No	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	125	No	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	172	No	1/8
Yes	Yes	205	Yes	1/8
Yes	Yes	167	Yes	1/8

Now since few samples are duplicates here, so they will be treated as a block of samples Hence, there will be a large penalty on the stump that misclassifies these.

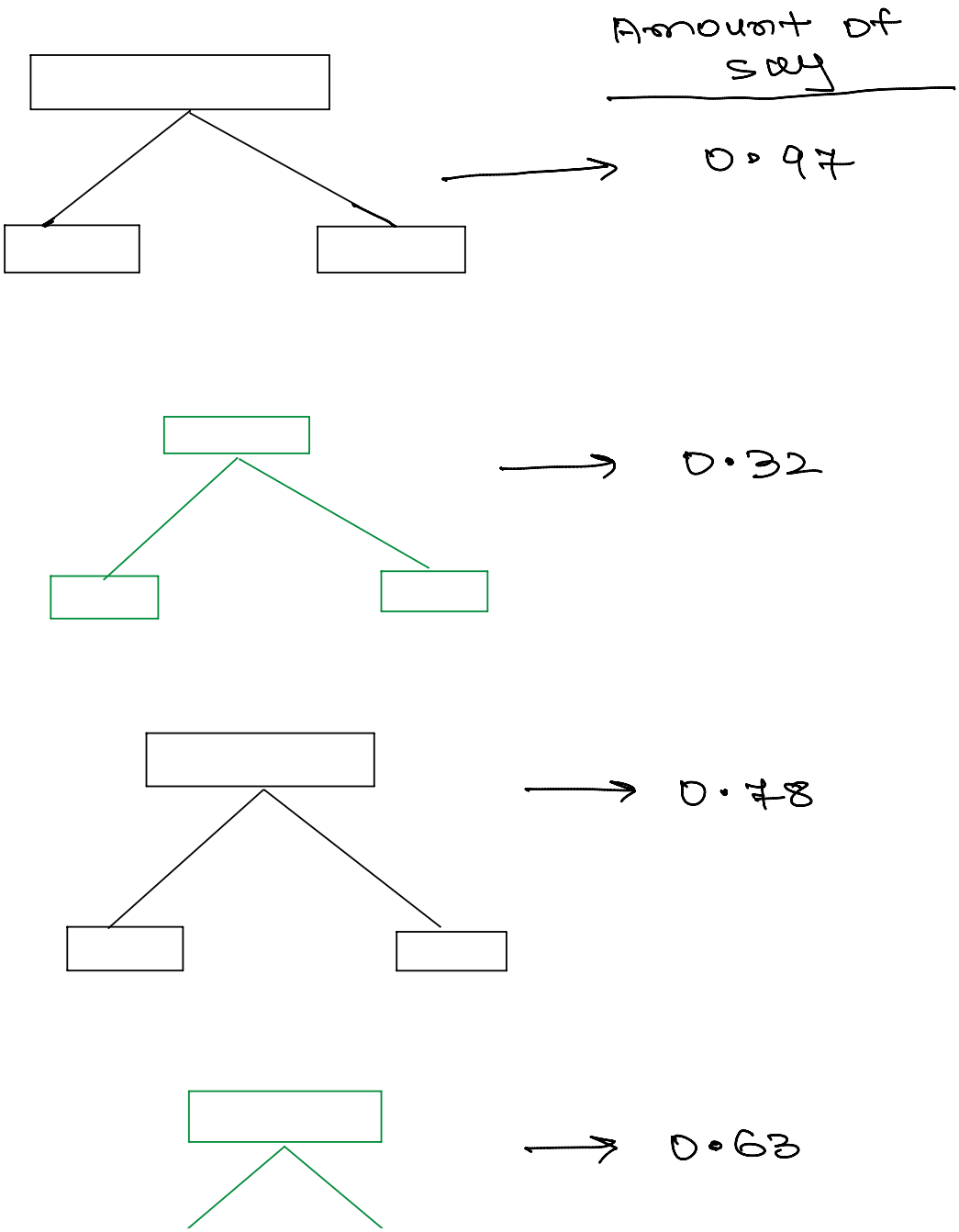
From here we go back to the first process of finding the stump that best classifies these samples.

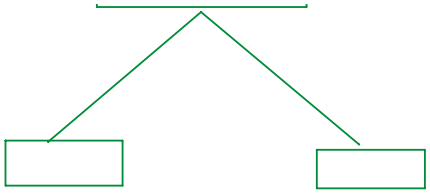
This way we understand how the classifi-

This way we understand how the classifi-cations of first tree influence the second tree and so on....

Now we need to talk about how a forest of stump created by 'Adaboost' makes classifications.

Imagine that these stumps classified the patient as 'Has Heart Disease' →



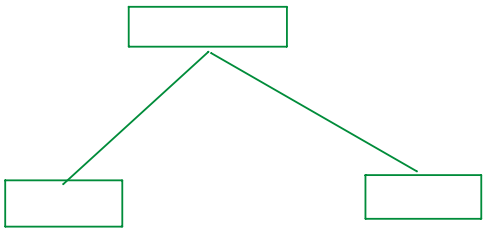


→ 0.65

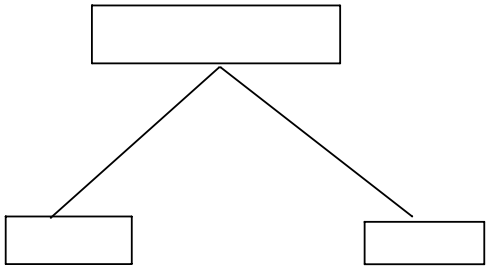
Total = 2.07

Following stumps classified the patient as 'No Heart Disease' →

Amount of say



→ 0.41



→ 0.82

Total = 1.23

Finally the patient will be classified as 'Has Heart Disease'.