AdaBoosting

-MUKESH KUMAR

AdaBoosting

- Adaboost, short for **Adaptive Boosting**, is a popular ensemble learning technique that combines the predictions of multiple **weak learners** to form a strong classifier.

- The key idea behind Adaboost is to improve the accuracy of a model by sequentially applying weak classifiers (models that perform slightly better than random guessing) and focusing on the mistakes made by previous classifiers.

Key Concepts

- **Weak Learner**: A weak learner is a model that performs slightly better than random guessing. In Adaboost, decision stumps (single-level decision trees) are commonly used as weak learners.

- **Boosting**: Boosting is a technique that sequentially trains weak learners, giving higher importance to the instances that previous models misclassified.

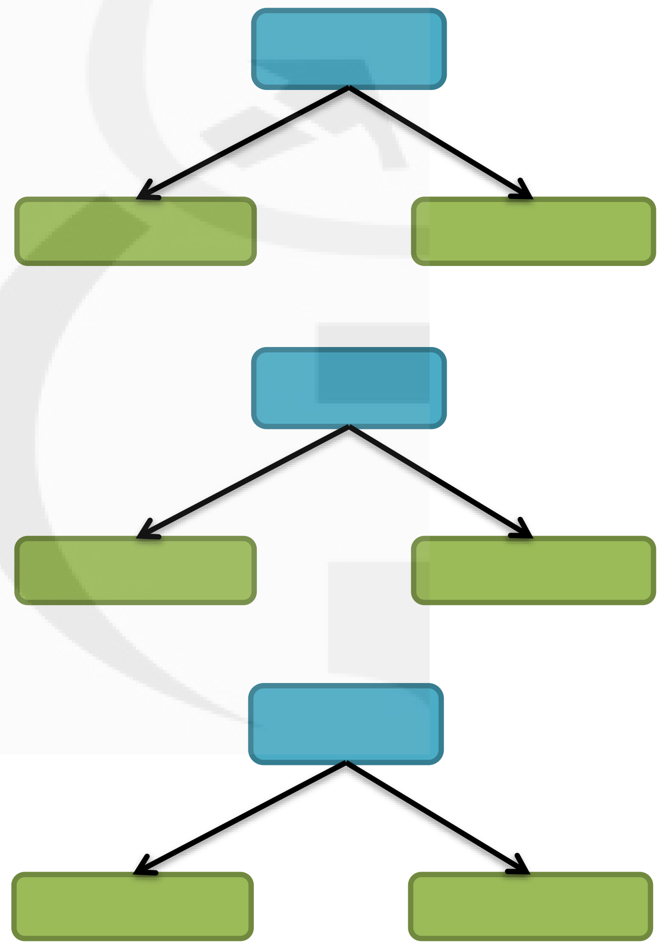
How AdaBoost Works

- **Initialize uniform weights** for all training examples.
- **For each iteration:**
  - Train a weak classifier on the **weighted** training data.
  - Calculate the weighted error rate of the weak classifier.
  - Compute the weight of the weak classifier based on its error rate.
  - Update the weights of the training examples based on the performance of the weak classifier.
- **The final strong classifier** is a weighted majority vote of all the weak classifiers, where each weak classifier's vote is weighted by its weight

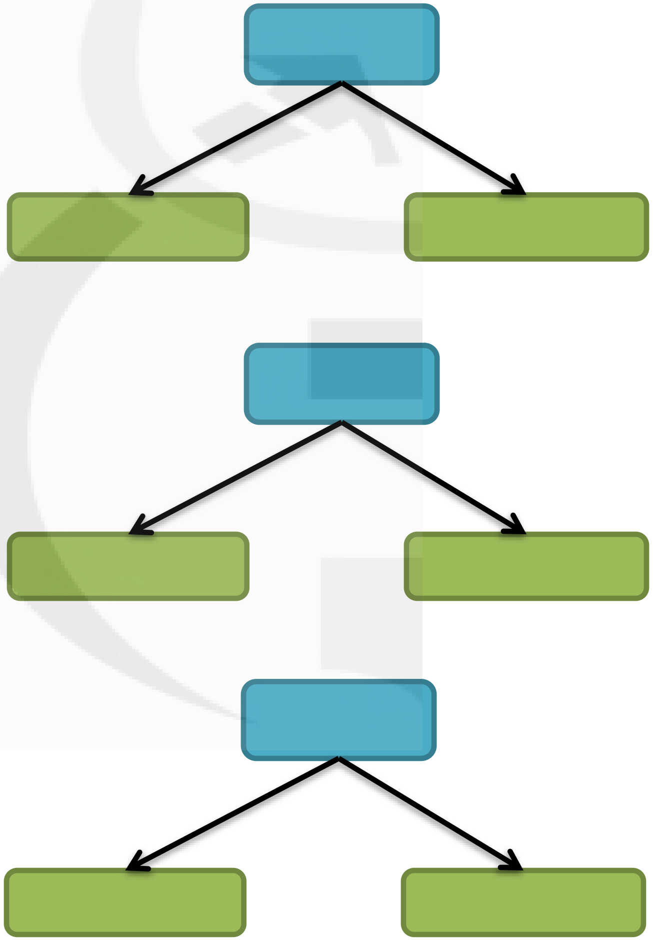
- Trees made by AdaBoost are usually just one node and two leaves.

- A tree with one node and two leave is called a **STUMPS**

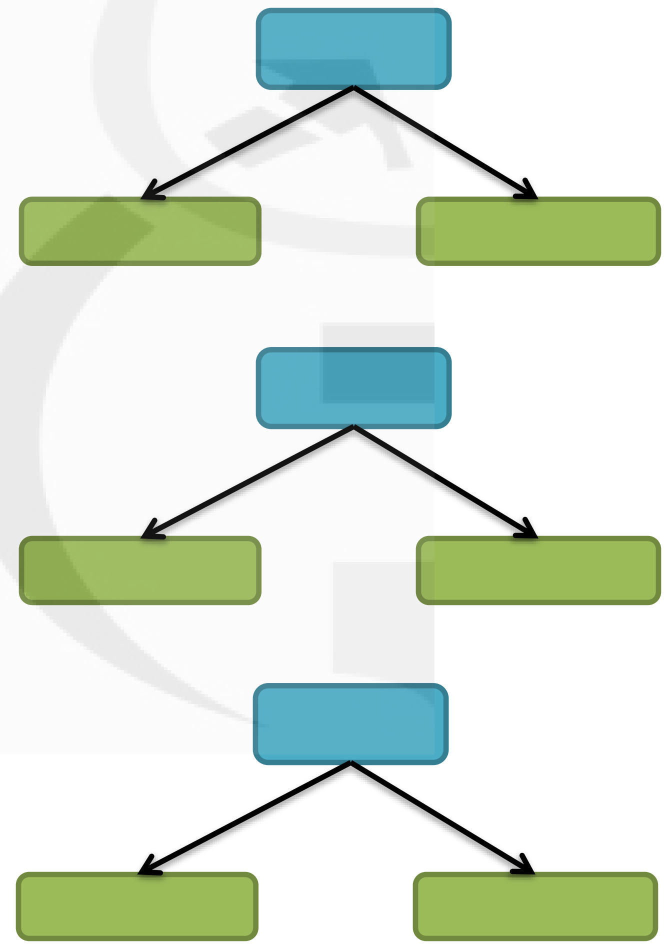
- Stumps are weak learners



- In Ada Boost some trees have more say in final classification than others



- In Ada Boost , order is important , each tree is made based on the error made by previous tree.



S.No	Alcohol Consumption	Smoking Status	Weight	Diabities
1	Yes	Yes	205	Yes
2	No	Yes	180	Yes
3	Yes	No	210	Yes
4	Yes	Yes	167	Yes
5	No	Yes	156	No
6	No	Yes	125	No
7	Yes	No	168	No
8	Yes	Yes	172	No

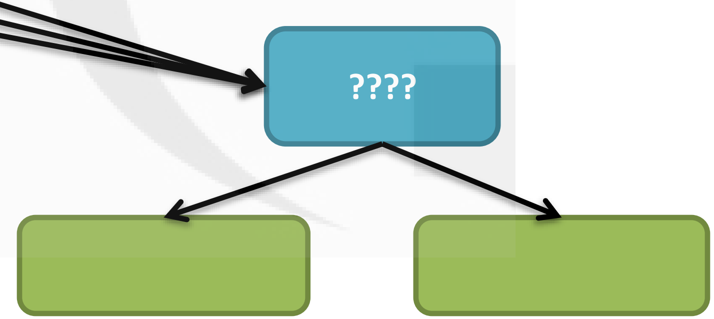
- Each sample is given a weight.
- Since we have 8 records each sample gets a weight of 1/8 such that sum of all the weights is 1.

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight
1	Yes	Yes	205	Yes	1/8
2	No	Yes	180	Yes	1/8
3	Yes	No	210	Yes	1/8
4	Yes	Yes	167	Yes	1/8
5	No	Yes	156	No	1/8
6	No	Yes	125	No	1/8
7	Yes	No	168	No	1/8
8	Yes	Yes	172	No	1/8

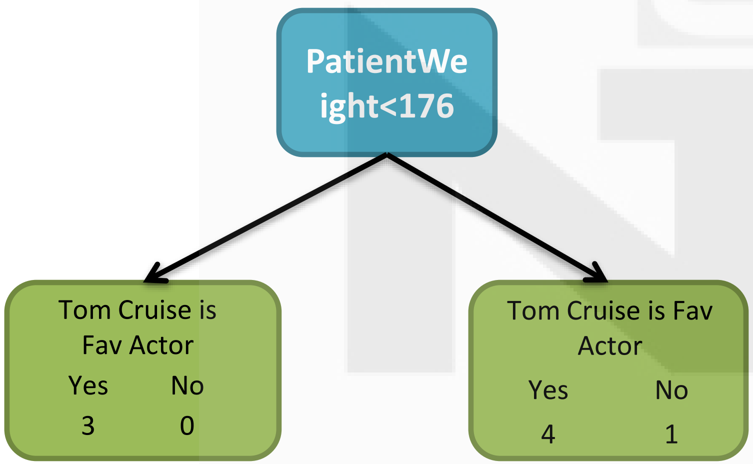
Creating the First Stump

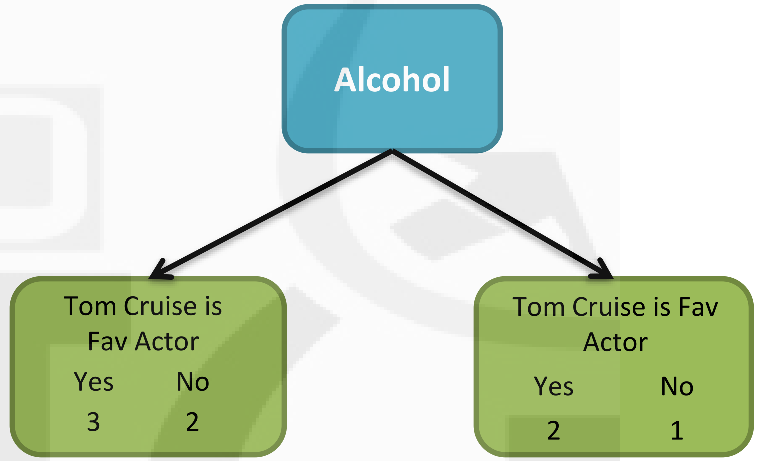
- Since we have three features , using Gini Index the best feature will be selected for the first stump
- Since all the weights are same , they can be ignored right now.

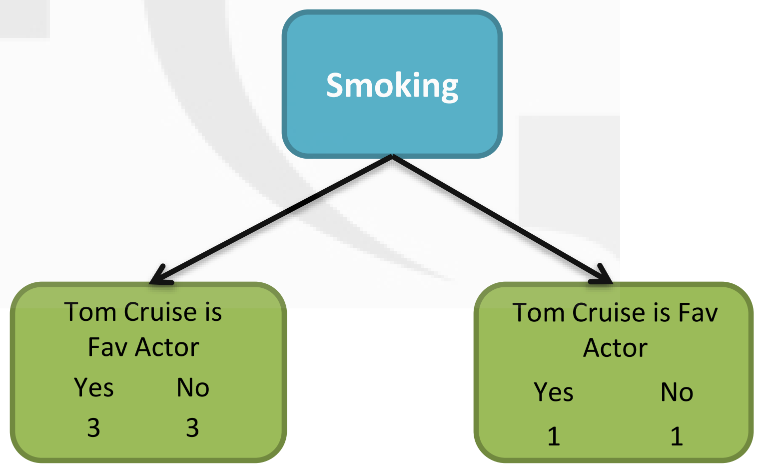
S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight
1	Yes	Yes	205	Yes	1/8
2	No	Yes	180	Yes	1/8
3	Yes	No	210	Yes	1/8
4	Yes	Yes	167	Yes	1/8
5	No	Yes	156	No	1/8
6	No	Yes	125	No	1/8
7	Yes	No	168	No	1/8
8	Yes	Yes	172	No	1/8



Selecting the best feature





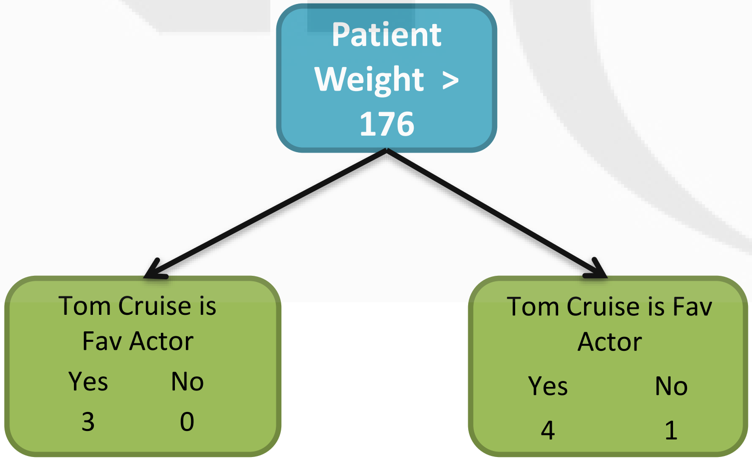


Alcohol	Smoking	Patient Weight
0.47	0.5	0.2

Selecting the best feature

- The Gini Index for Patient Weight is lowest so this will be the fist stump in the forest.

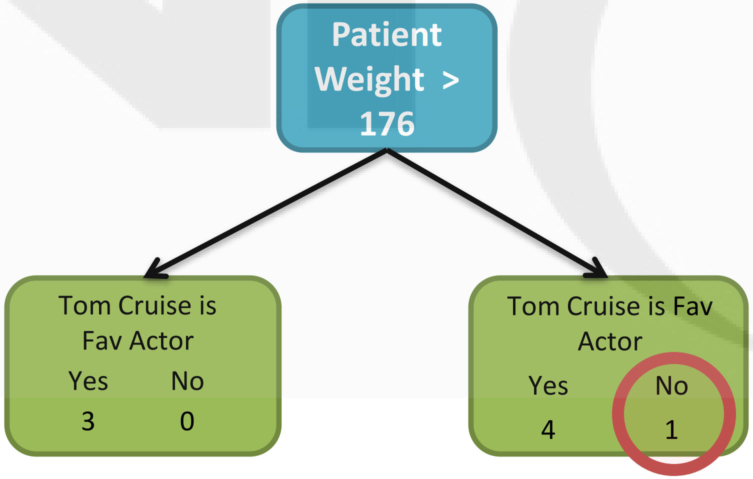
- Now we need to calculate how much say this stump will have in the final prediction



Calculate amount of say for Stump

- We determine how much say this stump will have in final classification based on how well it classifies the data points.

- This stump make one error.



Calculate amount of say for Stump

- As per the stump highlighted record has Diabities but stump say it doesn't have.

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight
1	Yes	Yes	205	Yes	1/8
2	No	Yes	180	Yes	1/8
3	Yes	No	210	Yes	1/8
4	Yes	Yes	167	Yes	1/8
5	No	Yes	156	No	1/8
6	No	Yes	125	No	1/8
7	Yes	No	168	No	1/8
8	Yes	Yes	172	No	1/8

Calculate amount of say for Stump

- The total error made by stump is the sum of weight associated with the mis classified record. = 1/8

- Total err = 1/8

Amount of say

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

This α_t measures the importance of the weak learner. If the error is small, the learner gets a higher weight (more influence), and if the error is large, it gets a lower weight.

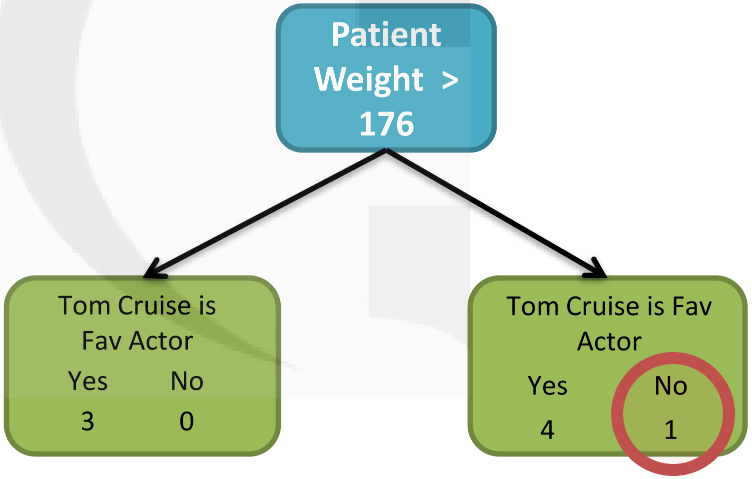
Amount of say

- 1 – log(7) = 0.97

- So the importance of first stump or amount of say of first stump in final prediction is 0.97

How to modify the weights of records

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight
1	Yes	Yes	205	Yes	1/8
2	No	Yes	180	Yes	1/8
3	Yes	No	210	Yes	1/8
4	Yes	Yes	167	Yes	1/8
5	No	Yes	156	No	1/8
6	No	Yes	125	No	1/8
7	Yes	No	168	No	1/8
8	Yes	Yes	172	No	1/8



Formula to increase the weight of incorrectly classified record

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

$$= \frac{1}{8} e^{\text{amount of say}}$$

$$= \frac{1}{8} e^{0.97} = \frac{1}{8} \times 2.64 = 0.33$$

Formula to reduce the weight of correctly classified records

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

$$= \frac{1}{8} e^{-\text{amount of say}}$$

$$= \frac{1}{8} e^{-0.97} = \frac{1}{8} \times 0.38 = 0.05$$

New weights

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight	New Weights
1	Yes	Yes	205	Yes	1/8	0.05
2	No	Yes	180	Yes	1/8	0.05
3	Yes	No	210	Yes	1/8	0.05
4	Yes	Yes	167	Yes	1/8	0.33
5	No	Yes	156	No	1/8	0.05
6	No	Yes	125	No	1/8	0.05
7	Yes	No	168	No	1/8	0.05
8	Yes	Yes	172	No	1/8	0.05

Normalize the new weight so that sum is 1

- Add all the new weight and divide each one by the sum

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weight	New Weights	Final new weights
1	Yes	Yes	205	Yes	1/8	0.05	0.07
2	No	Yes	180	Yes	1/8	0.05	0.07
3	Yes	No	210	Yes	1/8	0.05	0.07
4	Yes	Yes	167	Yes	1/8	0.33	0.48
5	No	Yes	156	No	1/8	0.05	0.07
6	No	Yes	125	No	1/8	0.05	0.07
7	Yes	No	168	No	1/8	0.05	0.07
8	Yes	Yes	172	No	1/8	0.05	0.07

- Add all the new weight and divide each one by the sum

S.No	Alcohol Consumption	Smoking Status	Patient Weight	Diabities	Sample Weights
1	Yes	Yes	205	Yes	0.07
2	No	Yes	180	Yes	0.07
3	Yes	No	210	Yes	0.07
4	Yes	Yes	167	Yes	0.49
5	No	Yes	156	No	0.07
6	No	Yes	125	No	0.07
7	Yes	No	168	No	0.07
8	Yes	Yes	172	No	0.07

Creating the 2^{nd} Stump

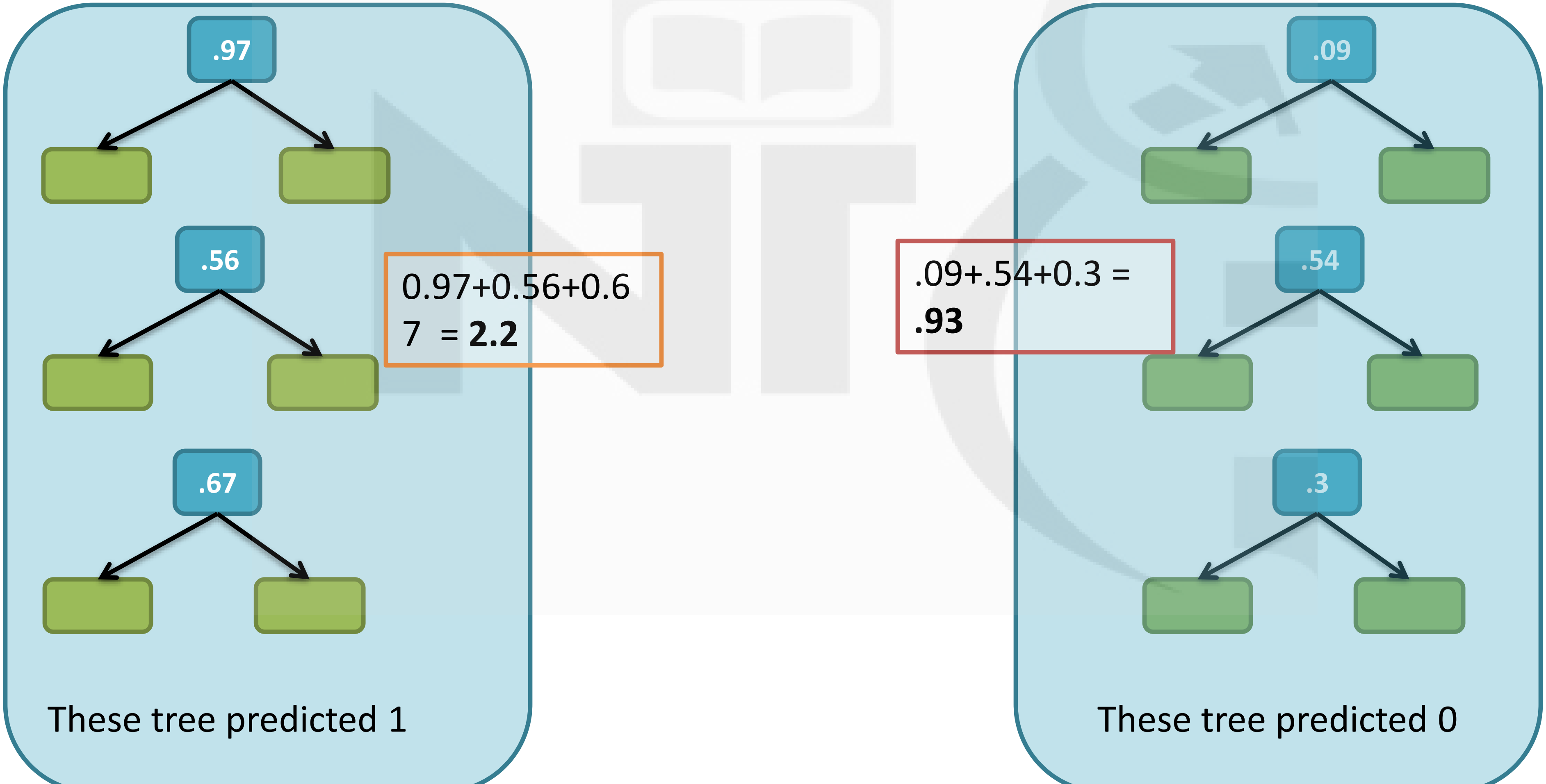
- Now we can use the new sample weights to create the second stump.

- Usually weighted Gini index is used to put more emphasis on the records with high weights

- That is how the Error that the first tree makes influences how the second tree is make and so on.

How Ada Boost makes predictions

- Patient is classified as has Diabities as 2.2 is greater



Strengths of Adaboost

- **Boosting Performance**: Even with weak classifiers like decision stumps, Adaboost often achieves excellent performance.
- **No Overfitting**: Adaboost tends to have a low risk of overfitting, even with many weak learners.
- **Versatile**: Can be used with any classifier, not just decision trees.

Limitations

- **Sensitive to Noise**: Since Adaboost increases the weight of misclassified examples, noisy data points can be given too much importance.
- **Requires Quality Weak Learners**:
- While Adaboost can boost weak learners, it still assumes that each weak learner performs slightly better than random guessing. If the weak learners consistently make poor decisions, Adaboost might struggle.
- **Not Ideal for Imbalanced Datasets**:
- Adaboost can struggle with highly imbalanced datasets. The model may focus too much on the majority class, especially if misclassified examples in the minority class have very high weights.

Assignments

- **On PIMA Indians Dataset:**

  - tune the estimator parameters for Gradient boost and Adaboost algorithm