<https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>

**AdaBoost**

* Adaboost helps you **combine multiple “weak classifiers” into a single “strong classifier”**.
* The weak learners in AdaBoost are decision trees with a single split, called decision stumps.
* AdaBoost works by putting more weight on misclassified instances and less on those already handled well.

After each classifier is trained, the weight is assigned to the classifier as well based on accuracy. More accurate classifier is assigned higher weight so that it will have more impact in final outcome.

* + A classifier with 50% accuracy is given a weight of zero, and a classifier with less than 50% accuracy is given negative weight.

<https://www.youtube.com/watch?v=NLRO1-jp5F8&t=690s>

Adaboost combines multiple weak learners into a single strong learner. This method does not follow Bootstrapping. However, it will create different decision trees with a single split (one depth), called decision stumps. The number of decision stumps it will make will depend on the number of features in the dataset. Suppose there are M features then, Adaboost will create M decision stumps.

1. We will assign an equal sample weight to each observation.
2. We will create M decision stumps, for M number of features.
3. Out of all M decision stumps, I first have to select one best decision tree model. For selecting it, we will either calculate the Entropy or Gini coefficient. The model with lesser entropy will be selected (means model that is less disordered). 4.
4. Now, after the first decision stump is built, an algorithm would evaluate this decision and check how many observations the model has misclassified. 5.
5. Suppose out of N observations, The first decision stump has misclassified T number of observations. 6.
6. For this, we will calculate the total error (TE), which is equal to T/N. 7.
7. Now we will calculate the performance of the first decision stump. Performance of stump = 1/2\*loge((1-TE)/TE) 8.
8. Now we will update the weights assigned before. To do this, we will first update the weights of those observations, which we have misclassified. The weights of wrongly classified observations will be increased and the weights of correctly classified weights will be reduced. 9.
9. By using this formula: old weight \* e performance of stump 10.
10. Now respectively for each observation, we will add and subtract the updated weights to get the final weights. 11.
11. But these weights are not normalized that is their sum is not equal to one. To do this, we will sum them and divide each final weight with that sum. 12.
12. After this, we have to make our second decision stump. For this, we will make a class intervals for the normalized weights. 13.
13. After that, we want to make a second weak model. But to do that, we need a sample dataset on which the second weak model can be run. For making it, we will run N number of iterations. On each iteration, it will calculate a random number ranging between 0-1 and this random will be compared with class intervals we created and on which class interval it lies, that row will be selected for sample data set. So new sample data set would also be of N observation. 14.
14. This whole process will continue for M decision stumps. The final sequential tree would be considered as the final tree.









