**A Quick Guide to Boosting in ML**

[](https://medium.com/@djocz?source=post_page-----acf7c1585cb5----------------------)

[Jocelyn D'Souza](https://medium.com/@djocz?source=post_page-----acf7c1585cb5----------------------)

[Mar 21, 2018](https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5?source=post_page-----acf7c1585cb5----------------------) · 4 min read

This post will guide you through an ensemble method known as Boosting. The main idea of boosting is to modify a weak learner to become better.

**Keep Learning…**

A road to success is incomplete without any failures in life. Each failure teaches you something new and makes you stronger at each phase. Each time you make a mistake, it’s important to learn from it and try not to repeat it again.





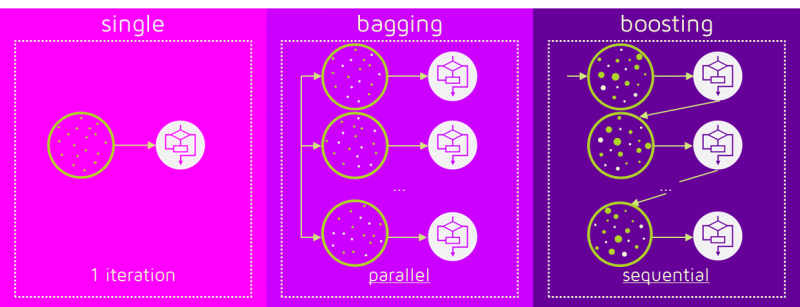
Source: [Pinterest](https://in.pinterest.com/pin/390476230170306563/)

Just as we sometimes develop life skills by learning from our mistakes, we can train our model to learn from the errors predicted and improvise the model’s prediction and overall performance. This is the most basic intuition of Boosting algorithm in Machine Learning.

**What is Boosting?**

The term ‘Boosting’ refers to a family of algorithms which converts weak learner to strong learners. Boosting is an ensemble method for improving the model predictions of any given learning algorithm. The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor.





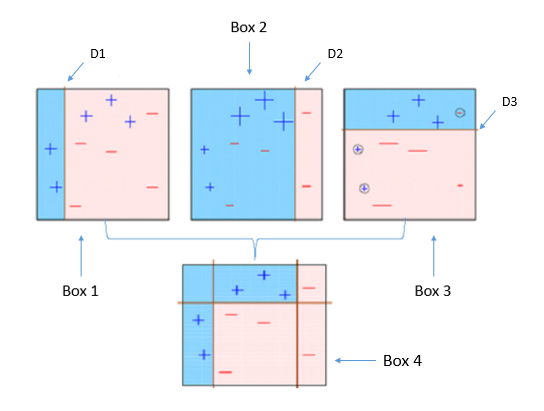
Source: [Quantdare](https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/)

**Types of Boosting**

**AdaBoost (Adaptive Boosting)**

Adaboost combines multiple weak learners into a single strong learner. The weak learners in AdaBoost are decision trees with a single split, called decision stumps. When AdaBoost creates its first decision stump, all observations are weighted equally. To correct the previous error, the observations that were incorrectly classified now carry more weight than the observations that were correctly classified. AdaBoost algorithms can be used for both classification and regression problem.



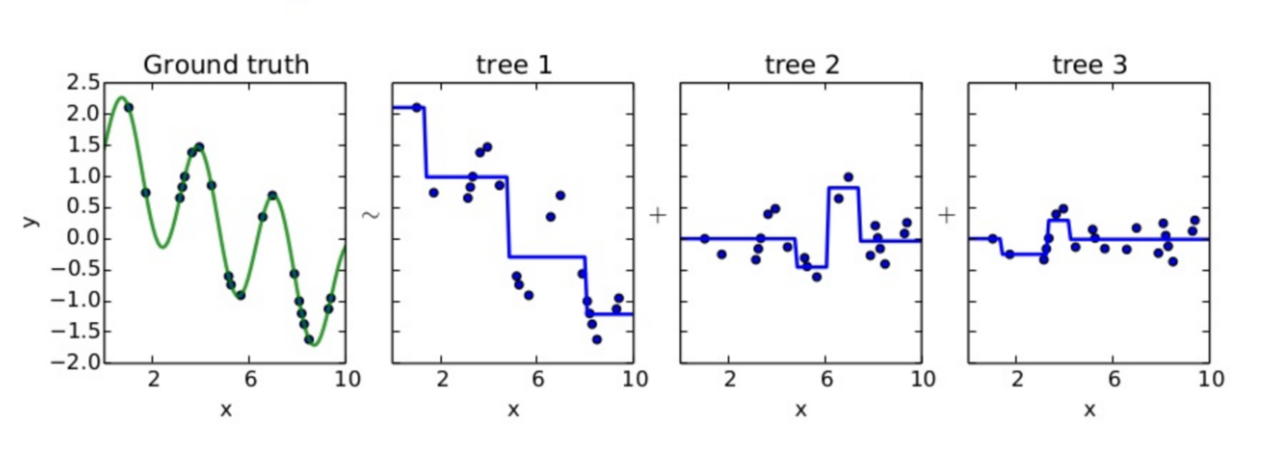


As we see above, the first decision stump(D1) is made separating the (+) blue region from the ( — ) red region. We notice that D1 has three incorrectly classified (+) in the red region. The incorrect classified (+) will now carry more weight than the other observations and fed to the second learner. The model will continue and adjust the error faced by the previous model until the most accurate predictor is built.

**Gradient Boosting**

Just like AdaBoost, Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of changing the weights for every incorrect classified observation at every iteration like AdaBoost, Gradient Boosting method tries to fit the new predictor to the residual errors made by the previous predictor.





Source: [Quora](https://www.quora.com/How-would-you-explain-gradient-boosting-machine-learning-technique-in-no-more-than-300-words-to-non-science-major-college-students)

GBM uses Gradient Descent to find the shortcomings in the previous learner’s predictions. GBM algorithm can be given by following steps.

* Fit a model to the data, F1(x) = y
* Fit a model to the residuals, h1(x) = y−F1(x)
* Create a new model, F2(x) = F1(x) + h1(x)
* By combining weak learner after weak learner, our final model is able to account for a lot of the error from the original model and reduces this error over time.

**XGBoost**



XGBoost stands for e**X**treme **G**radient **B**oosting. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. Gradient boosting machines are generally very slow in implementation because of sequential model training. Hence, they are not very scalable. Thus, XGBoost is focused on computational speed and model performance. XGBoost provides:

* **Parallelization** of tree construction using all of your CPU cores during training.
* **Distributed Computing** for training very large models using a cluster of machines.
* **Out-of-Core Computing** for very large datasets that don’t fit into memory.
* **Cache Optimization** of data structures and algorithm to make the best use of hardware.

If you wish to know more about XGBoost, follow this [post](https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/).