**Adaboost:** <https://www.youtube.com/watch?v=LsK-xG1cLYA>

1. Combines weak learners to make classifications. The weak learners are always stumps(1 node and 2 leaf).
2. Some stumps get more say in the classification than others.
3. Each stump is made by taking the previous stump’s mistake into account.

**Gradient Boosting:** <https://www.youtube.com/watch?v=3CC4N4z3GJc&t=53s>

1. Gradient boosting start by making a single leaf, instead of a tree or stump. The first leaf is the average value. Then build a tree.
2. Like Adaboost, this tree is based on the errors made by the previous tree. But the tree is larger than a stump. Gradient boost still restricts the size of tree.
3. Gradient boost builds another tree based on the errors made by previous tree.

**XGBoost:**

**Advantages:**

1. Regularization: Standard GBM implementation has no regularization like XGBoost, therefore it also helps to reduce overfitting.
2. Parallel Processing: It uses the multiple cores to perform the processing.
3. High flexibility: Custom optimization objectives and evaluation criteria.
4. Handling missing values: What does it do
5. Tree Pruning:

* A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm.
* XGBoost on the other hand make splits upto the max\_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.
* Another advantage is that sometimes a split of negative loss say -2 may be followed by a split of positive loss +10. GBM would stop as it encounters -2. But XGBoost will go deeper and it will see a combined effect of +8 of the split and keep both.

1. Built-in cross validation:

* XGBoost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run.
* This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

1. Continue on existing model:

* User can start training an XGBoost model from its last iteration of previous run. This can be of significant advantage in certain specific applications.
* GBM implementation of sklearn also has this feature so they are even on this point.