Boosting Algorithm

.

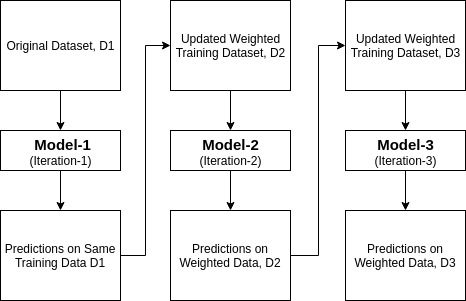
Boosting uses **many small decision trees** (called **weak learners**) that work **one after the other**.

* Each tree is **not very deep** (usually 1–5 levels).
* The first tree makes predictions.
* The second tree tries to **fix the mistakes** of the first one.
* The third tree fixes the second one’s mistakes, and so on.
* At the end, the predictions from all trees are **combined** (like averaging or weighted sum) to get a strong final result.

Boosting = Many Small Decision Trees + Error Correction + Final Ensemble Output

Instead of training **one big tree**, boosting trains **many small trees** in a sequence, each improving on the last

Ada boost algorithm:



**What is XGBoost?**

**XGBoost** stands for **Extreme Gradient Boosting**.  
It’s an advanced, super-fast, and accurate version of the **gradient boosting algorithm** — one of the most powerful tools in modern machine learning.

**XGBoost = Gradient Boosting + Extreme Optimizations**

* It still works by building **trees in sequence**, like regular Gradient Boosting.
* But it adds **tons of clever tricks** to make training faster, more efficient, and more accurate.

A diagram of a computer algorithm

AI-generated content may be incorrect.

LightGBM

**Leaf-wise Tree Growth (vs. Level-wise)**

| **Level-wise (XGBoost)** | **Leaf-wise (LightGBM)** |
| --- | --- |
| Grows all nodes at the same depth | Grows the **leaf with max loss reduction** |
| More balanced tree | More accurate, but can be deep |
| Slower | Faster & better accuracy |

Leaf-wise chooses the best leaf to split instead of expanding level by level. This leads to **deeper trees** and **faster learning**.

**LightGBM is like a high-performance version of Gradient Boosting,** optimized for speed, memory, and scalability — great for big datasets and production systems.