**Boosting Algorithm**

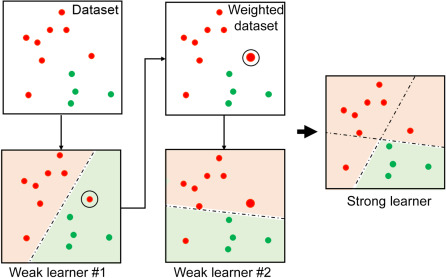
**Boosting** is an ensemble technique that combines multiple weak models (usually decision trees) to create a strong model. It builds models **sequentially**, where each new model tries to fix the errors of the previous one.

1. Ada Boosting
2. XG Boost
3. LightGBM (LGBoost)

**AdaBoost (Adaptive Boosting)**

* Starts with a simple decision tree.
* After each round, **focuses more on the wrongly predicted points** by assigning them higher weights.
* Builds the next model to fix the errors.

***Data → Model 1 → Errors → Model 2 (focuses on errors) → Model 3 → Final Prediction***



Advantages:

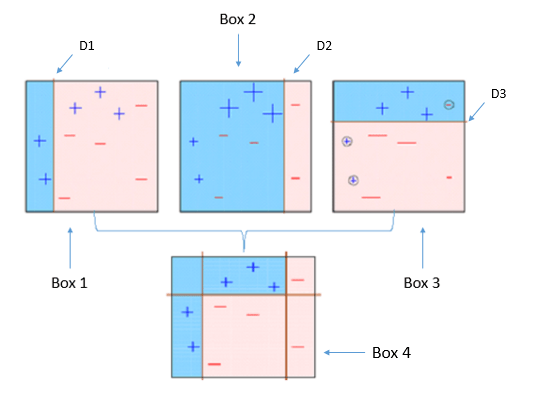
* Easy to implement.
* Works well on clean datasets.
* Reduces bias

**XGBoost (Extreme Gradient Boosting):**

* Uses **gradient descent** to minimize error.
* Tries to fix errors of the previous models, like AdaBoost.
* It’s very **fast and accurate** because it uses:

1. Regularization (to avoid overfitting)
2. Parallel processing

***Data → Tree 1 → Gradient Error → Tree 2 → Tree 3 → Final Prediction***



**Advantages:**

* High accuracy
* Handles missing data well
* Less overfitting
* Fast training

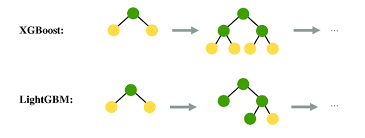
**LightGBM (Light Gradient Boosting Machine):**

Like XGBoost but **faster and more efficient**, especially on **large datasets**.

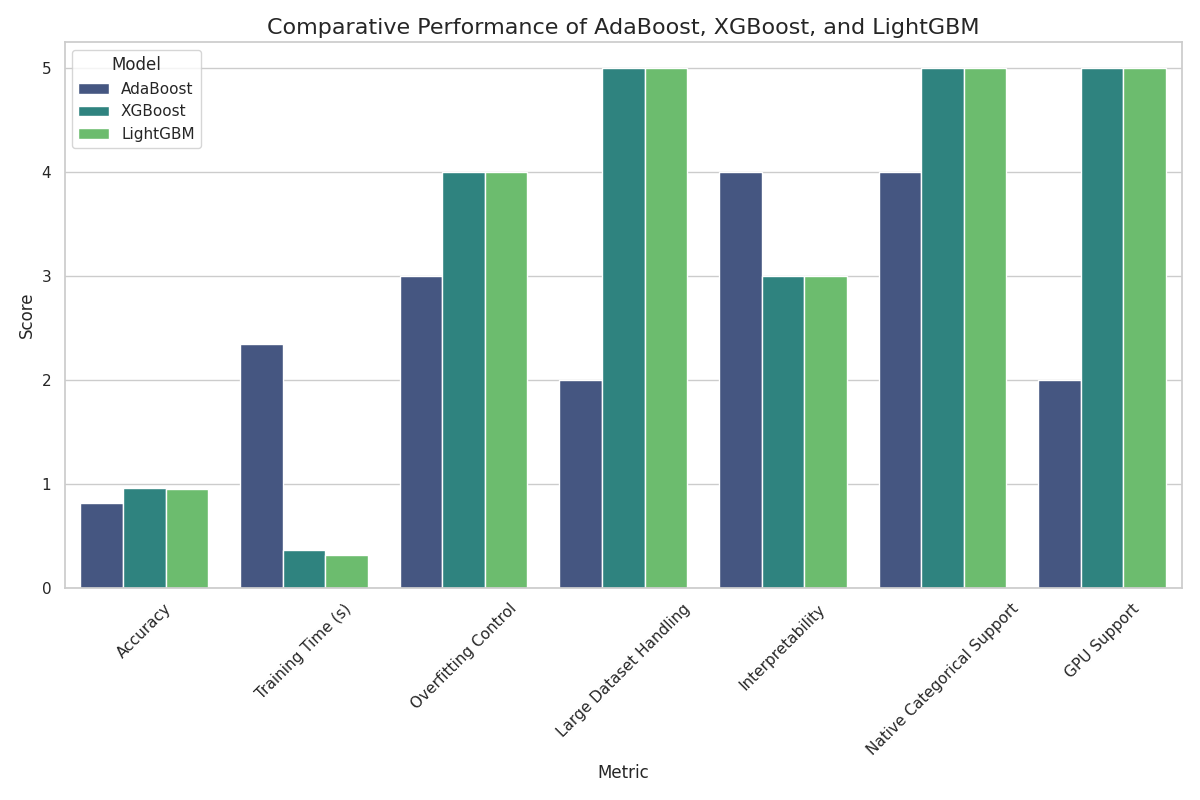
Uses:

1. **Histogram-based approach** (makes training faster)
2. **Leaf-wise tree growth** (more accurate but can overfit if not tuned)

***Data → Leaf-wise Tree Growth → Fast Training → Final Prediction***



**Comparative image if Ada,xg and LG Boosting**



**Justification:**

Boosting methods like AdaBoost, XGBoost, and LightGBM help improve model performance by focusing on previous errors. They are especially useful when we want high accuracy and need to deal with complex or large datasets. XGBoost and LightGBM are very popular in Kaggle competitions and industry due to their speed and efficiency.