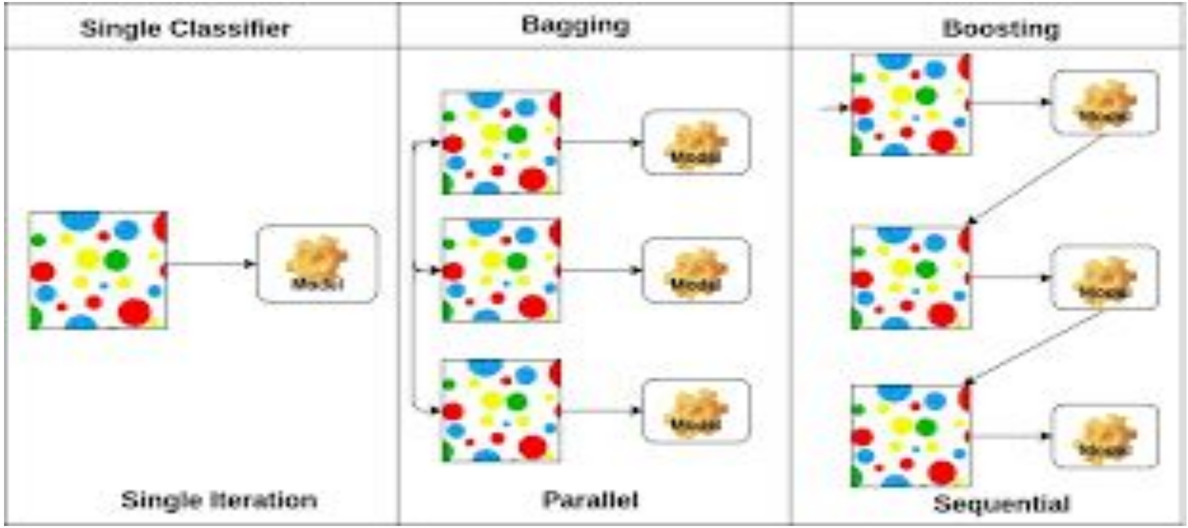
Boosting Algorithm

Ada Boost
Is the meta-ensembler

Comparison bw Bagging and Boosting

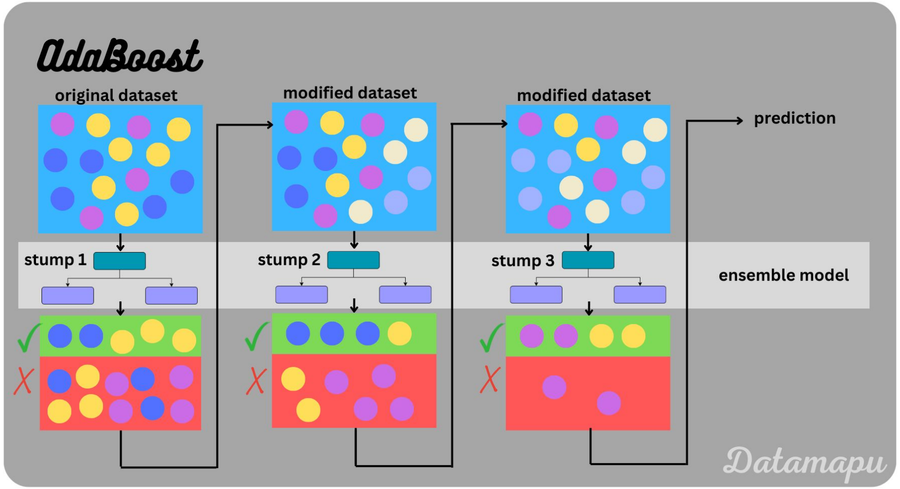


Bagging	**Boosting**
Independent Model	Dependent Model
parallel learner	sequential learner
Most occurred result will be the best model	Uses an Interactive approach to learn from mistake

Ada Boost

Base learner (stump)1 will be selected based on entropy

Incorrect classifier send to next stump 2



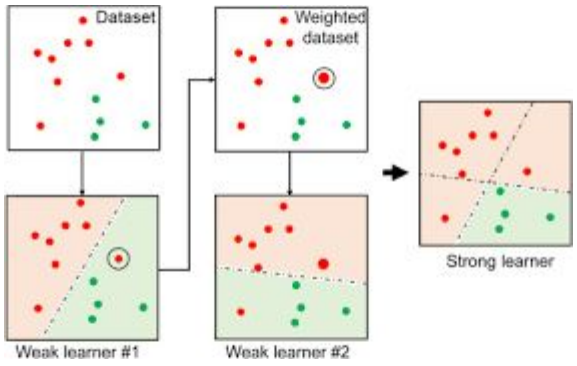
Weight changes

Initially same weight will be assigned to all dataset

Weight will be increased for incorrect response

And decreased for correct response, it will be calculated based on some formula

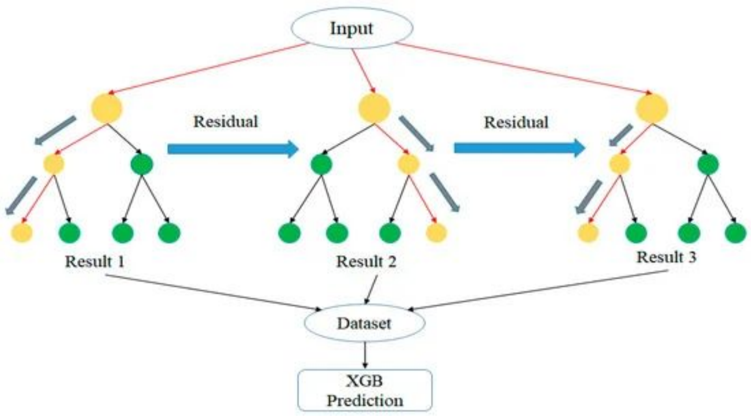
Overall efficiency will be increased



Extreme Gradient Boosting Algorithm

XG Boosting

- XG Boosting uses gradient boosting framework at its core
- Default base learners of XGBoost is tree ensembles
- It can be either set of classification / regression trees
- Trees are grown one after the other and attempts to reduce the misclassification rate in subsequent iterations.



XG Boost Algorithm Features :

- Regularization:

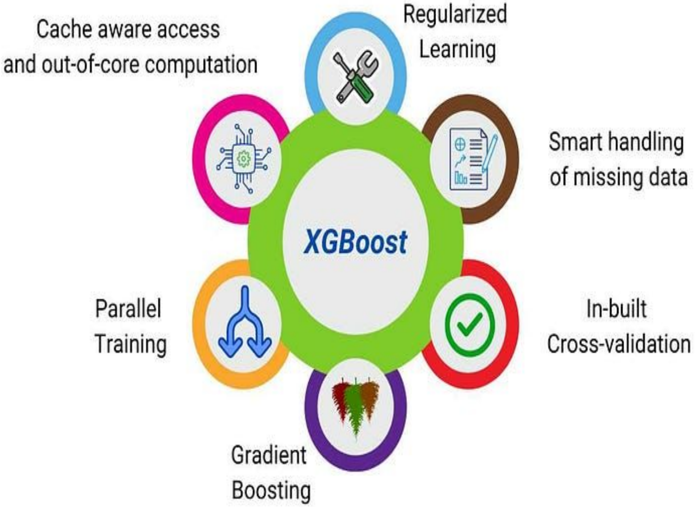
  It’s helps in preventing overfitting.

- Out-of-core computing:

  It’s optimizes the available disk space

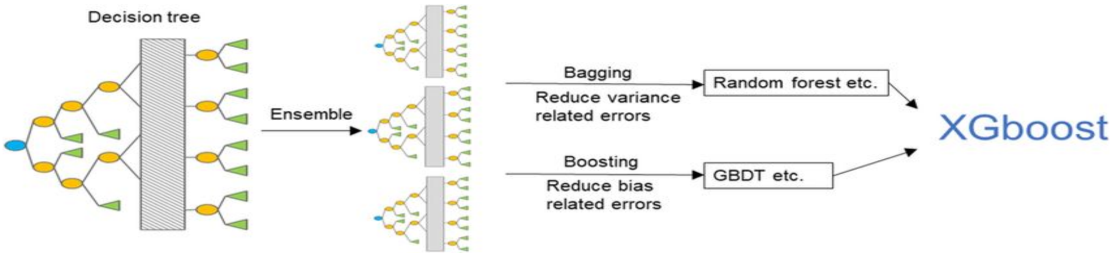
- Handling sparse data

  It’s incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data


XGBoost
Regularized Learning
Smart handling of missing data
In-built Cross-validation
Gradient Boosting
Parallel Training
Cache aware access and out-of-core computation

Why XGBoost?

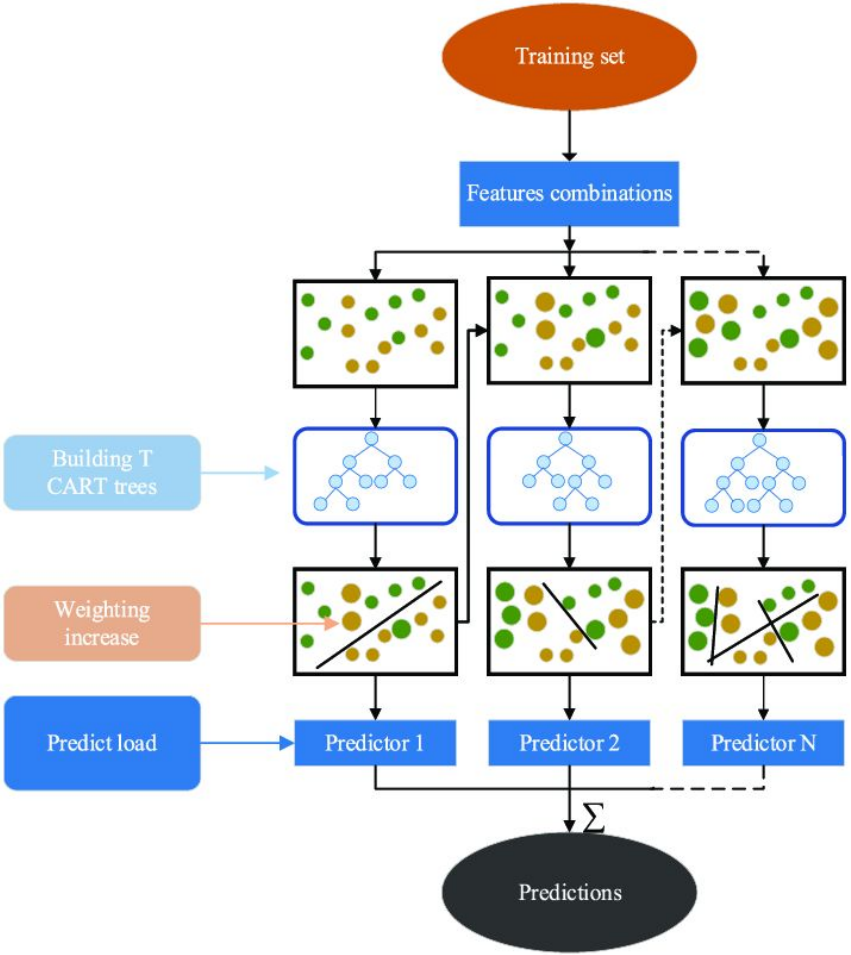
Execution Speed:



With XGBoost, trees are built in parallel, instead of sequentially like GBDT.

Model Performance.

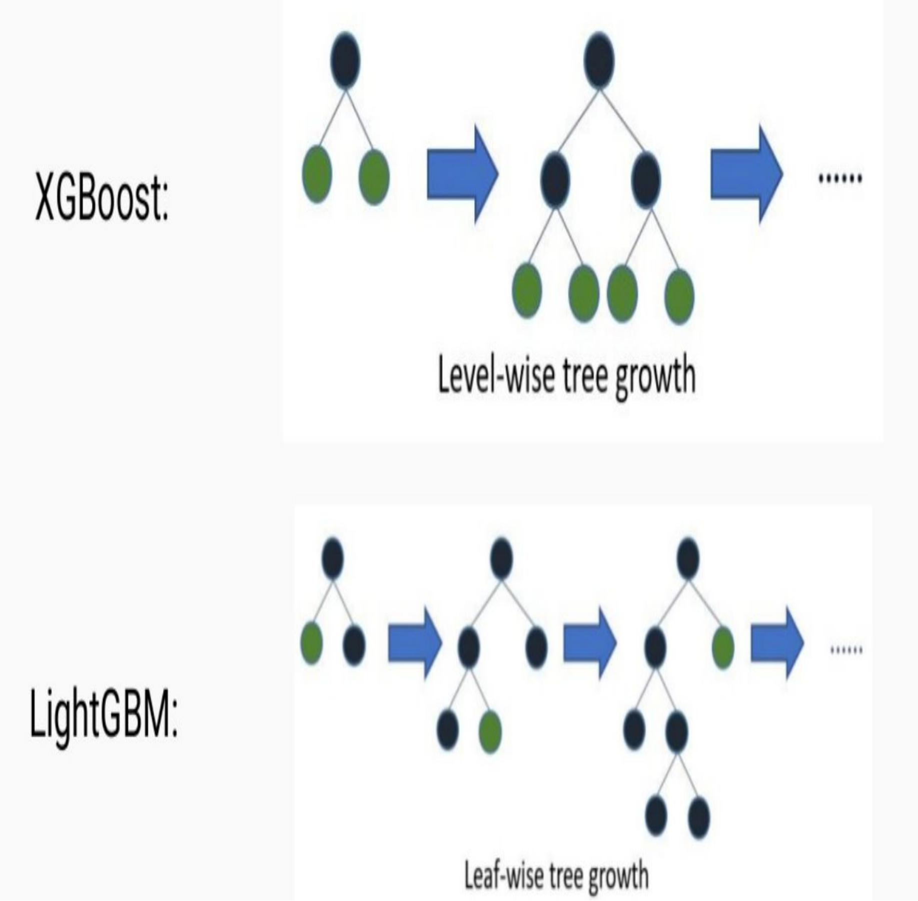
It follows a level-wise strategy, scanning across gradient values and using these partial sums to evaluate the quality of splits at every possible split in the training set.



Light Gradient Boosting Machine

LightGBM

- It is a high-performance gradient boosting framework designed for efficient and scalable machine learning tasks. It is specially tailored for speed and accuracy, making it a popular choice for both structured and unstructured data in diverse domains.
- Also it’s based on decision tree algorithm
- It splits the the tree leaf wise



Light GBM’s Unique Approach

Light GBM grows the tree by finding the best leaf to split, resulting in a more balanced and accurate tree.

Light GBM uses a histogram-based algorithm for finding the best split. It discretized continuous features into discrete bins, which dramatically speeds up the training process. This makes Light GBM highly efficient, especially on large datasets.

Advantages of LG Boosting

- Less Memory Usage
- Reduction in Communication Cost for parallel learning
- Reduction in Cost for calculating gain for each split in the decision tree.
- Handling Categorical Features
- Speed and Efficiency
- Excellent Generalization

LightGBM gets trained much faster but also it can lead to the case of overfitting sometimes.

To get the best fit following parameters must be tuned:

1. **num_leaves:** Since LightGBM grows leaf-wise this value must be less than 2^(max_depth) to avoid an overfitting scenario.
2. **min_data_in_leaf:** For large datasets, its value should be set in hundreds to thousands.
3. **max_depth:** A key parameter whose value should be set accordingly to avoid overfitting.