eXtreme Gradient Boosting-XG BOOST

What is XGBoost?

XGBoost stands for eXtreme Gradient Boosting. It's a parallelized and carefully optimized version of the gradient boosting algorithm. Parallelizing the whole boosting process improves the training time significantly.

Instead of training the best possible model on the data (like in traditional methods), we train thousands of models on various subsets of the training dataset and then vote for the best-performing model.

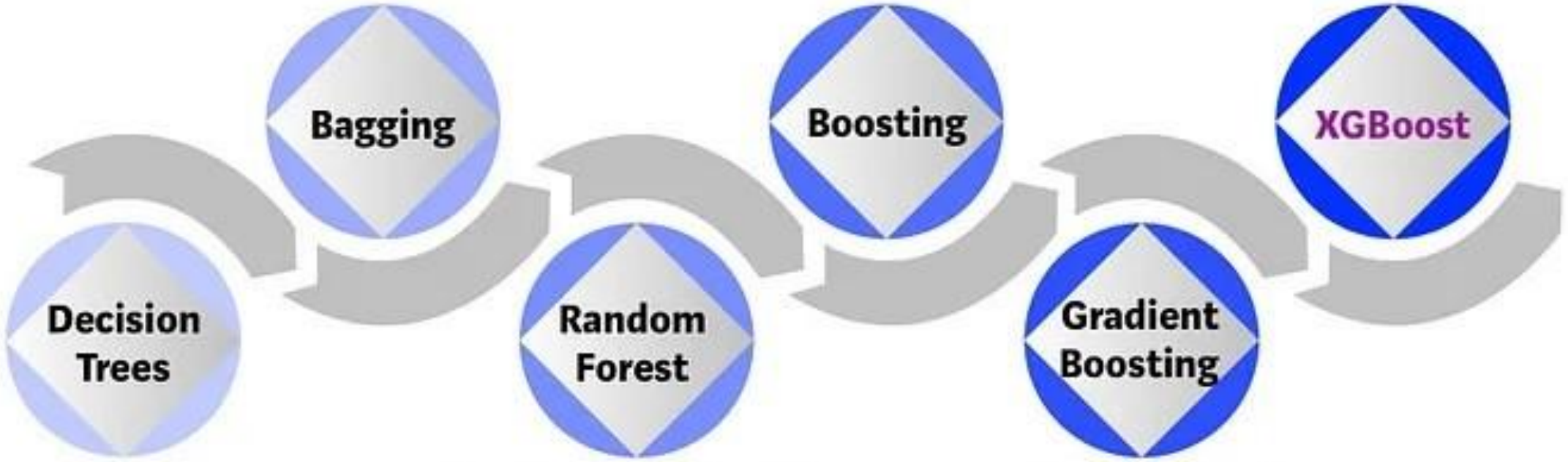
In many cases, XGBoost is better than usual gradient-boosting algorithms. The Python implementation gives access to a vast number of inner parameters to tweak for better precision and accuracy.


Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism

Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models

Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias

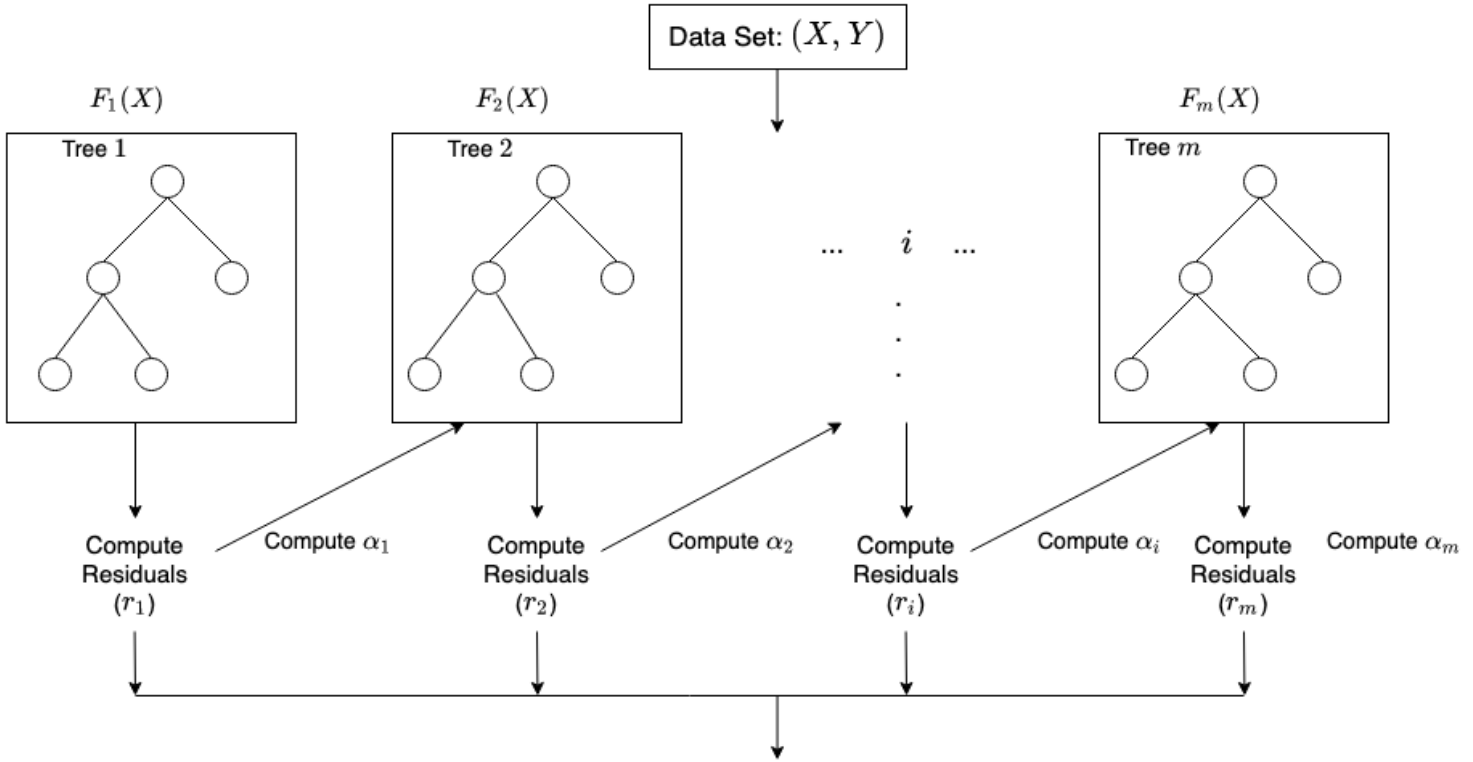


A graphical representation of possible solutions to a decision based on certain conditions

Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees

Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

How does XGBoost work?



$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

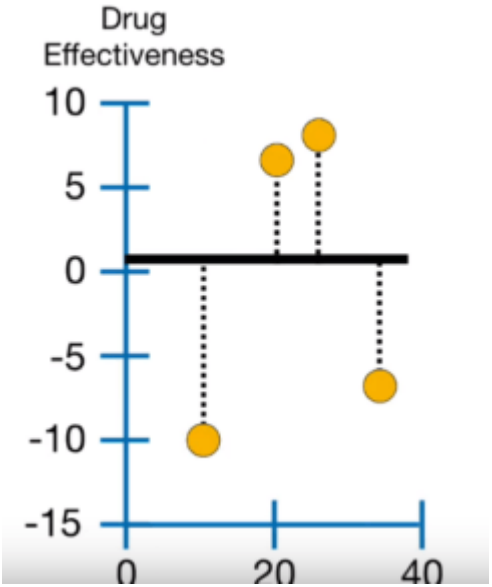
where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

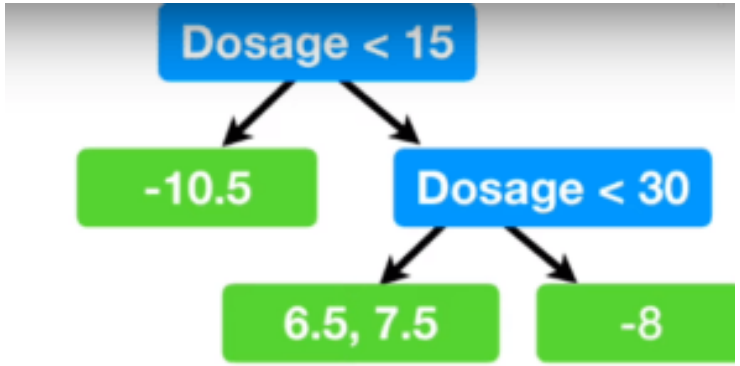
Example:

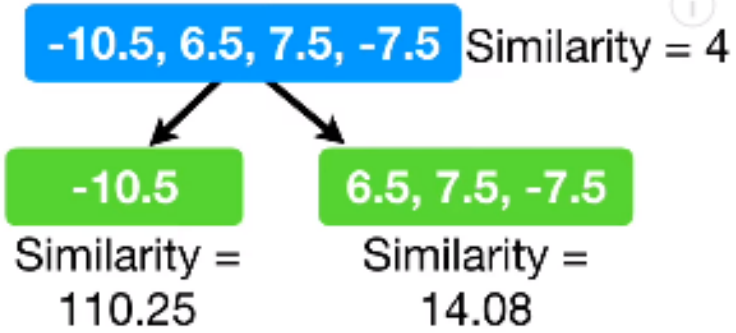
$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

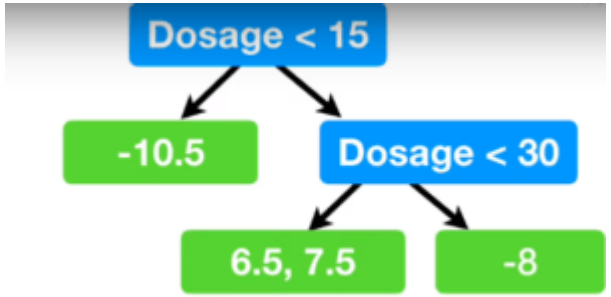
Here **λ is a regularisation parameter**.

$$\text{Similarity Score} = \frac{(-10.5 + 6.5 + 7.5 + -7.5)^2}{4 + 0}$$









Key Features and Advantages of XGBoost

01 Gradient Boosting Framework

02 Regularization

03 Handling of Missing Values

04 Tree Pruning

05 Built-in Cross-Validation

06 High Flexibility

07 Effective Handling of Various Data Types

08 Scalability and Portability

09 Weighted Quantile Sketch

10 Cache Awareness