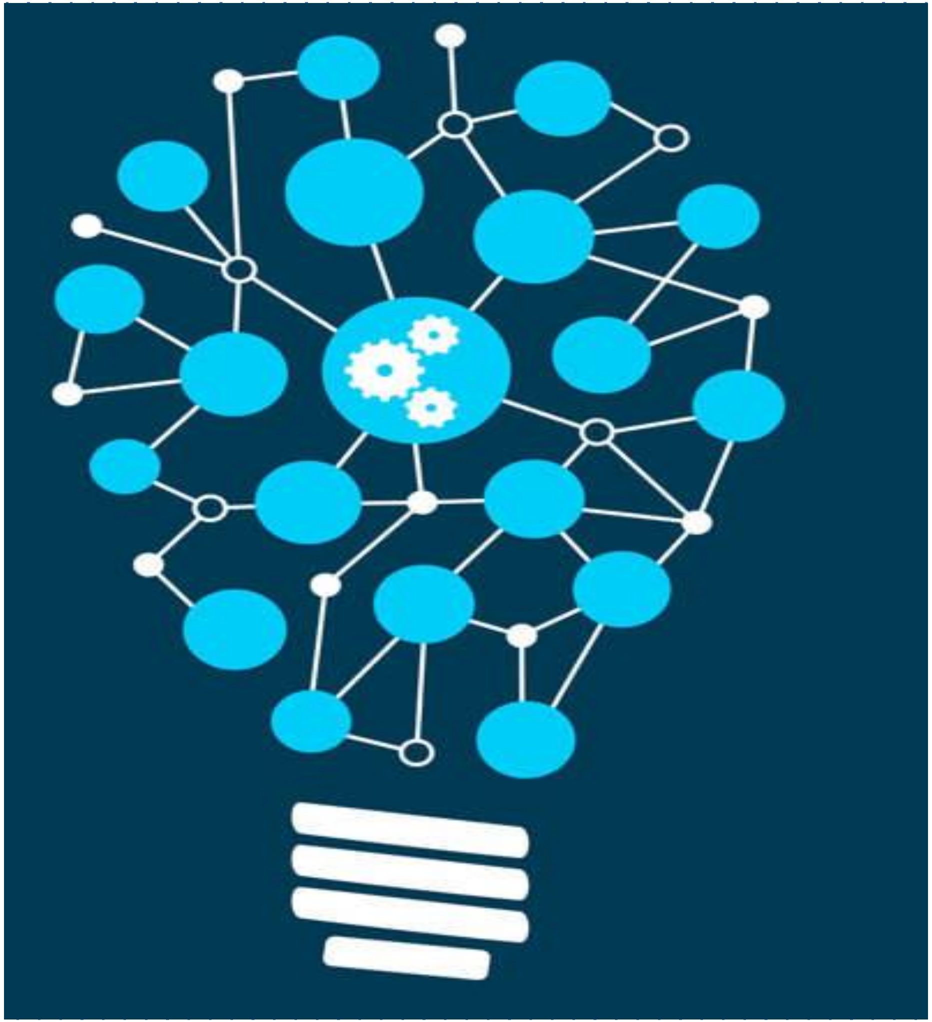
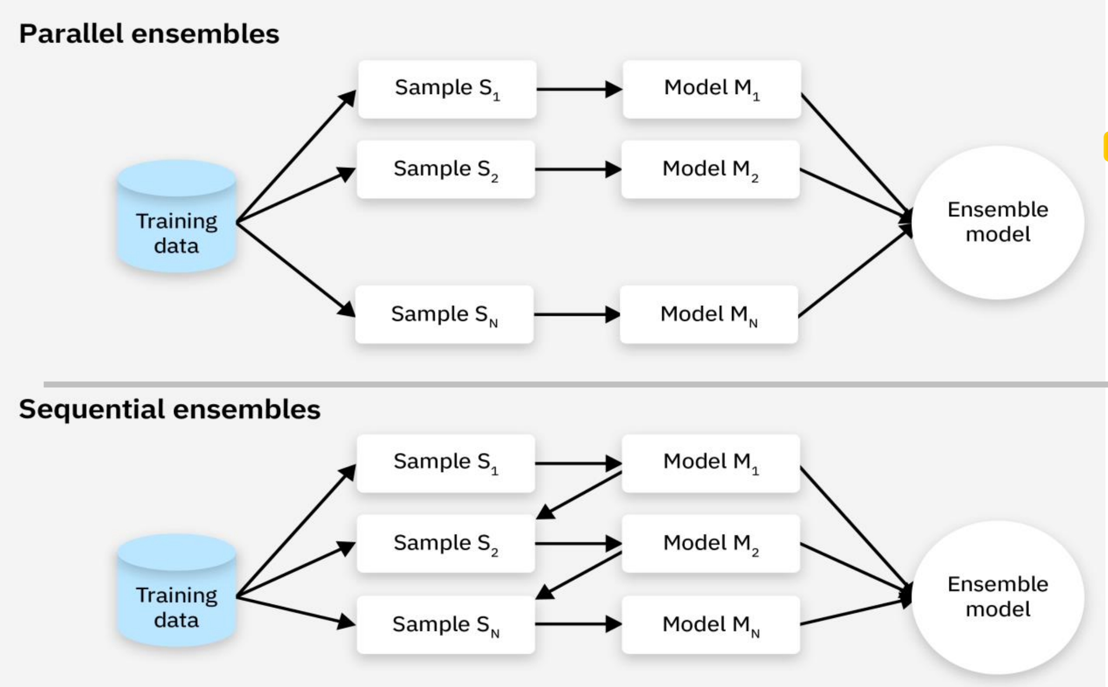
Boosting Algorithms

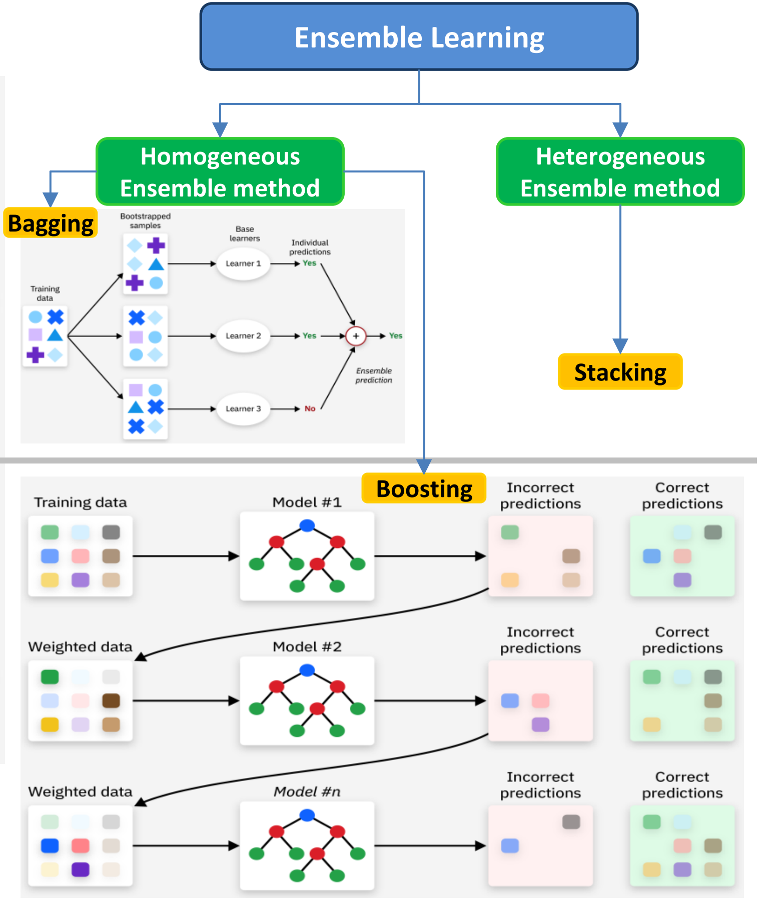


Ensemble models

Parallel ensembles train base learners in parallel and independent of one another



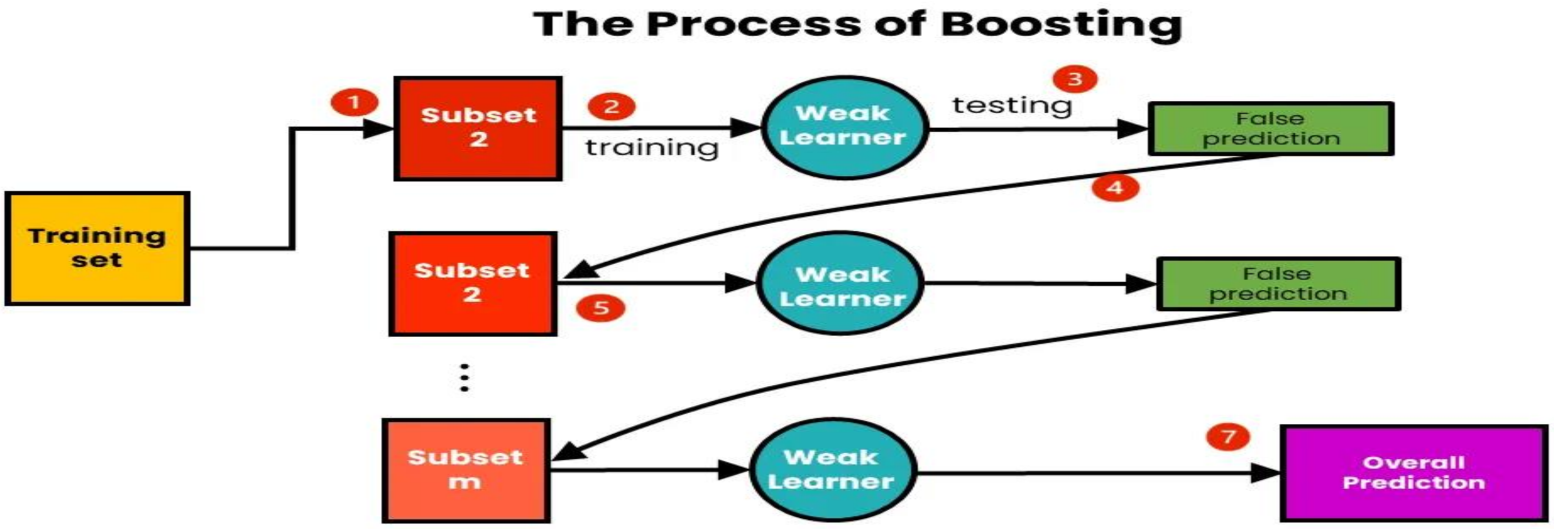
Sequential methods construct base models sequentially in stages



Source: https://www.ibm.com/think/topics/ensemble-learning

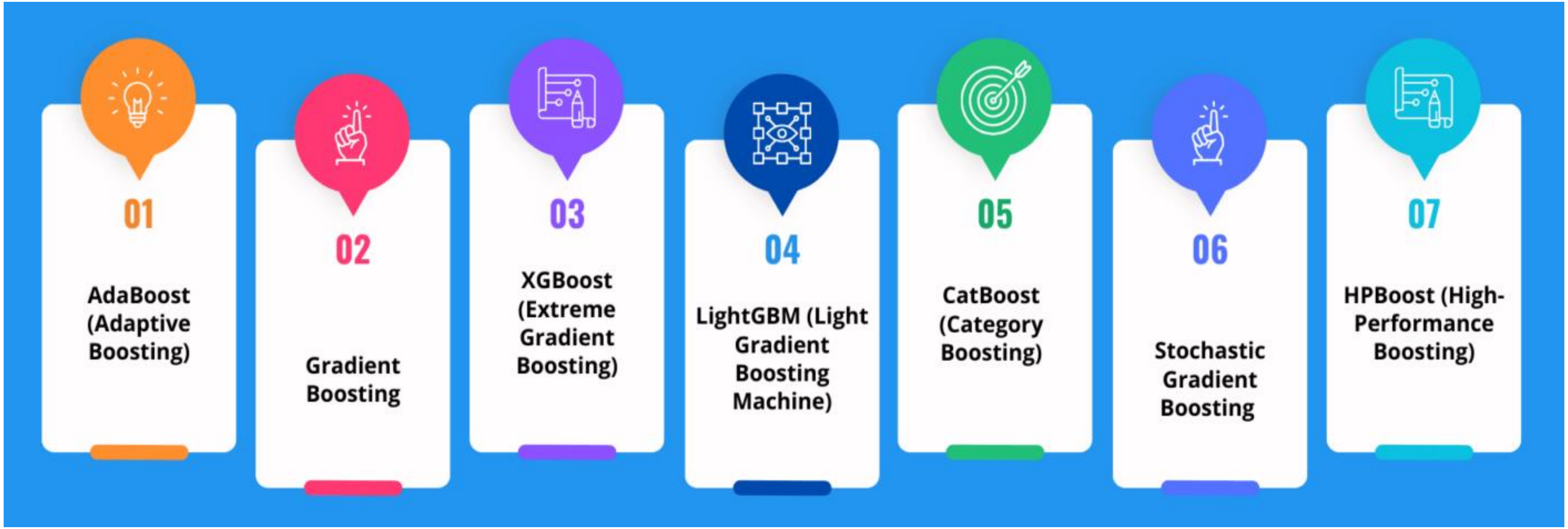
Boosting Algorithm

Supervised machine learning strategy that combines the predictions of multiple weak models (base models) to generate a powerful ensemble model



Source: https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6

Different Boosting Algorithm



01 AdaBoost (Adaptive Boosting)

02 Gradient Boosting

03 XGBoost (Extreme Gradient Boosting)

04 LightGBM (Light Gradient Boosting Machine)

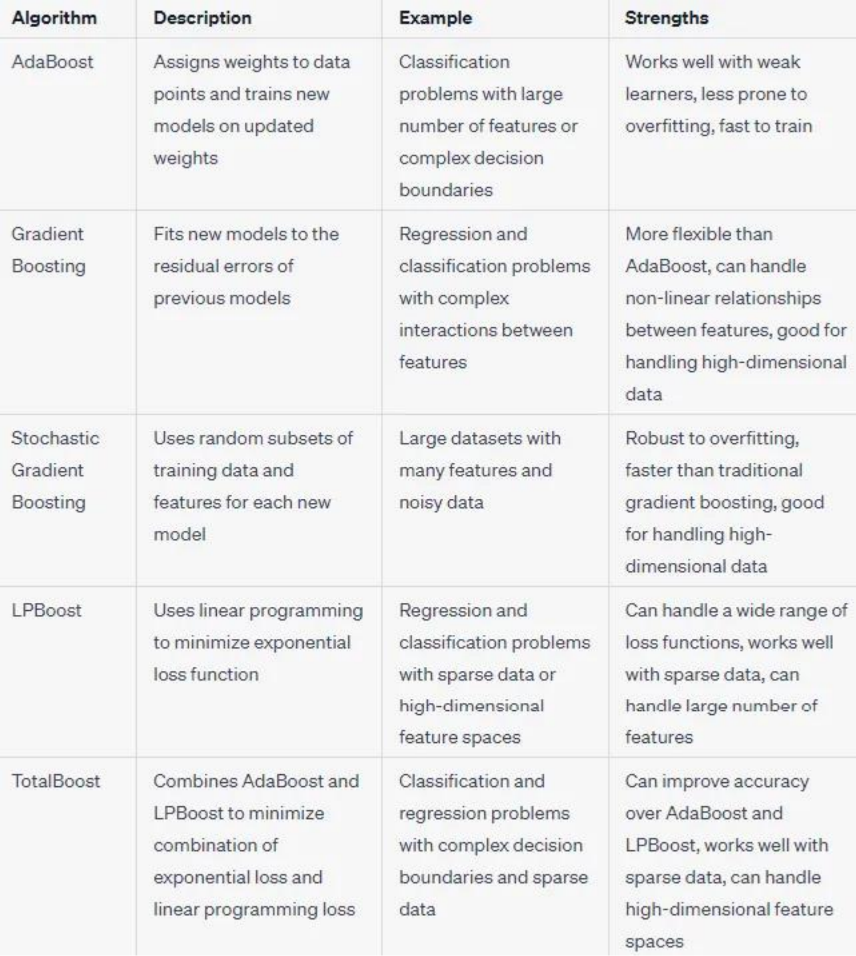
05 CatBoost (Category Boosting)

06 Stochastic Gradient Boosting

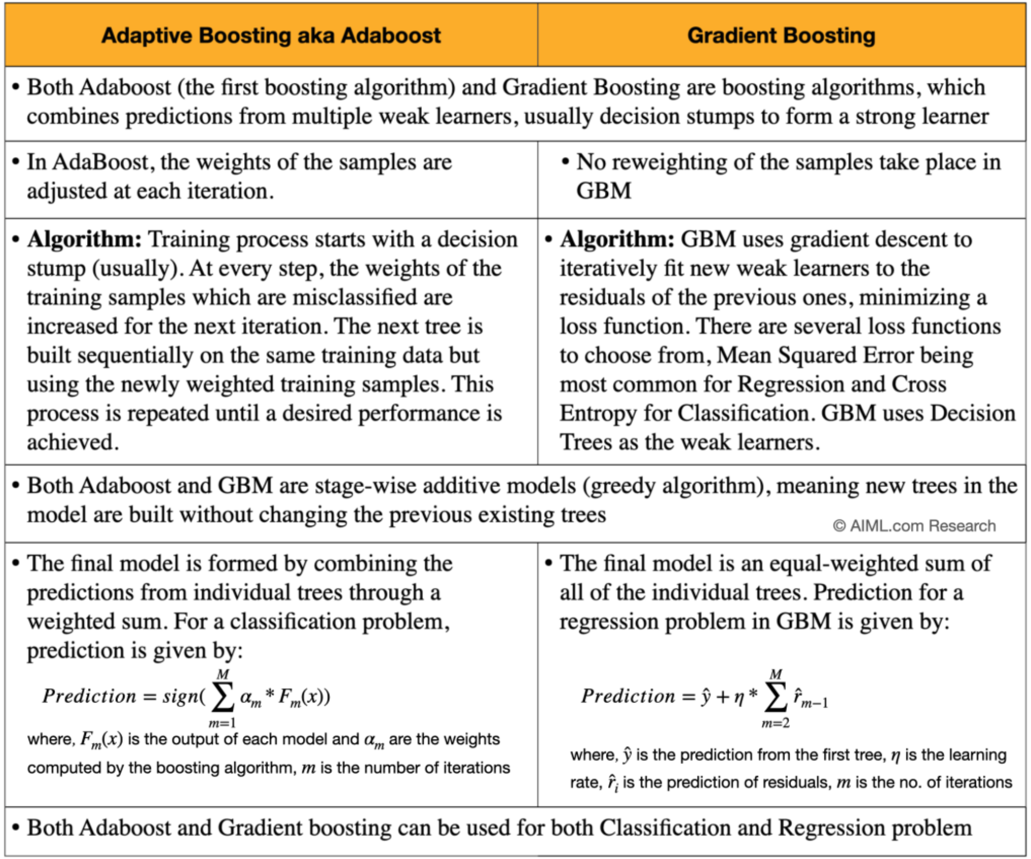
07 HPBoost (High-Performance Boosting)

Source: https://dataaspirant.com/boosting-algorithms/

Compare Boosting Algorithm



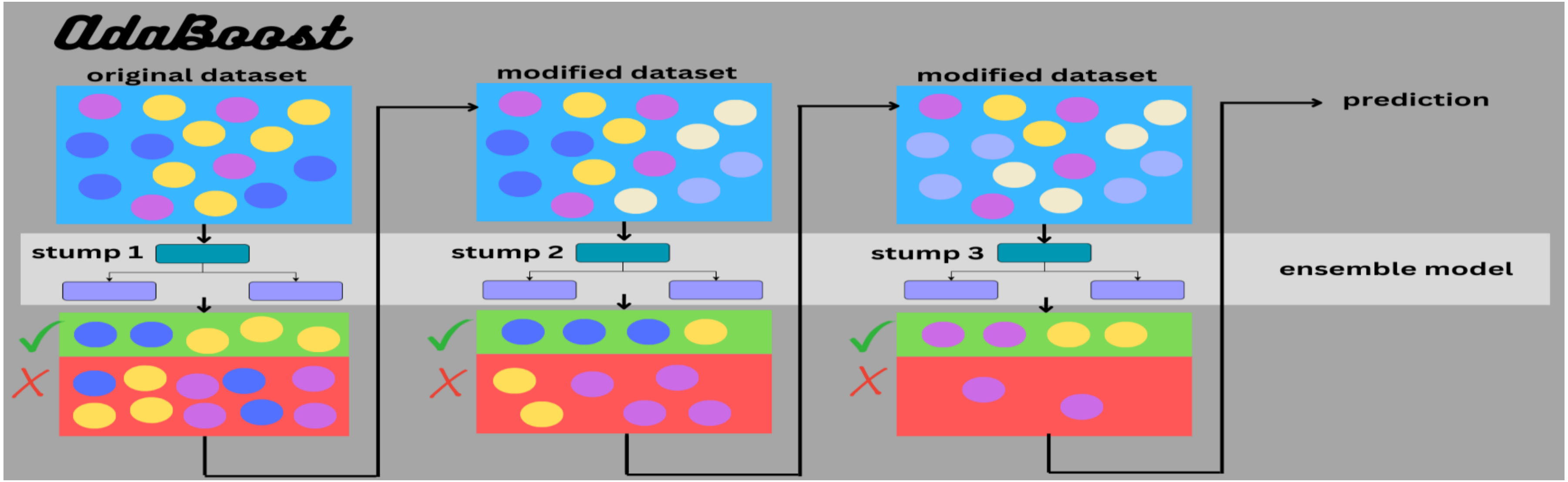
Algorithm	Description	Example	Strengths
AdaBoost	Assigns weights to data points and trains new models on updated weights	Classification problems with large number of features or complex decision boundaries	Works well with weak learners, less prone to overfitting, fast to train
Gradient Boosting	Fits new models to the residual errors of previous models	Regression and classification problems with complex interactions between features	More flexible than AdaBoost, can handle non-linear relationships between features, good for handling high-dimensional data
Stochastic Gradient Boosting	Uses random subsets of training data and features for each new model	Large datasets with many features and noisy data	Robust to overfitting, faster than traditional gradient boosting, good for handling high-dimensional data
LPBoost	Uses linear programming to minimize exponential loss function	Regression and classification problems with sparse data or high-dimensional feature spaces	Can handle a wide range of loss functions, works well with sparse data, can handle large number of features
TotalBoost	Combines AdaBoost and LPBoost to minimize combination of exponential loss and linear programming loss	Classification and regression problems with complex decision boundaries and sparse data	Can improve accuracy over AdaBoost and LPBoost, works well with sparse data, can handle high-dimensional feature spaces

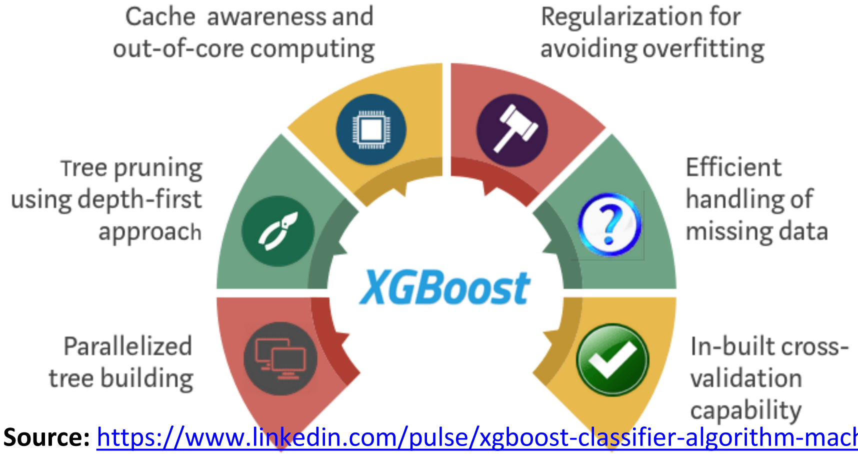


<table>
<tr><th>Adaptive Boosting aka Adaboost</th><th>Gradient Boosting</th></tr>
<tr><td colspan="2">• Both Adaboost (the first boosting algorithm) and Gradient Boosting are boosting algorithms, which combines predictions from multiple weak learners, usually decision stumps to form a strong learner</td></tr>
<tr><td>• In AdaBoost, the weights of the samples are adjusted at each iteration.</td><td>• No reweighting of the samples take place in GBM</td></tr>
<tr><td>• <b>Algorithm:</b> Training process starts with a decision stump (usually). At every step, the weights of the training samples which are misclassified are increased for the next iteration. The next tree is built sequentially on the same training data but using the newly weighted training samples. This process is repeated until a desired performance is achieved.</td><td>• <b>Algorithm:</b> GBM uses gradient descent to iteratively fit new weak learners to the residuals of the previous ones, minimizing a loss function. There are several loss functions to choose from, Mean Squared Error being most common for Regression and Cross Entropy for Classification. GBM uses Decision Trees as the weak learners.</td></tr>
<tr><td colspan="2">• Both Adaboost and GBM are stage-wise additive models (greedy algorithm), meaning new trees in the model are built without changing the previous existing trees
© AIML.com Research</td></tr>
<tr><td>• The final model is formed by combining the predictions from individual trees through a weighted sum. For a classification problem, prediction is given by:
$Prediction = sign(\sum_{m=1}^{M} \alpha_m * F_m(x))$
where, $F_m(x)$ is the output of each model and α_m are the weights computed by the boosting algorithm, m is the number of iterations</td><td>• The final model is an equal-weighted sum of all of the individual trees. Prediction for a regression problem in GBM is given by:
$Prediction = \hat{y} + \eta * \sum_{m=2}^{M} \hat{r}_{m-1}$
where, $\hat{y}$ is the prediction from the first tree, η is the learning rate, $\hat{r}_i$ is the prediction of residuals, m is the no. of iterations</td></tr>
<tr><td colspan="2">• Both Adaboost and Gradient boosting can be used for both Classification and Regression problem</td></tr>
</table>

Source: https://aiml.com/what-is-the-difference-between-adaboost-vs-gradient-boost/

Source: https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6




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

Source: https://www.linkedin.com/pulse/xgboost-classifier-algorithm-machine-learning-kavya-kumar/

THANK YOU!