

XGBoost

Extreme Gradient Boosting

- XGBoost is a **supervised, tree-based** ML algorithm that utilize the concept of Gradient Boosting. GB divides the training data into small subsets. Each subset is used to build a model tree in a successive way such that it tries to overcome the performances of the last model in line.¹
- XGBoost is considered to have an improved Regularization Term (Ω) compared to Random Forest based trees.¹
- XGBoost **refit** hyper-parameter defines which metric is most important to us when evaluating the model between learning steps.¹
- Hyper-parameters are external configurations that are set prior to training. XGBoost includes many hyper-parameters including eta, gamma, max_depth and more. The GridSearchCV function

of XGBoost enables to "tune" hyper-parameters for best model results.¹

Key XGBoost Hyper-Parameters:

- **eta** learning Rate (default = 0.3, range 0-1):
Defines how large are the steps during the training process. Tiny steps could lead to over-fitting while large steps could lead to too much generalization.
³
- **gamma** split decision threshold (default = 0, range 0-infinity): Controls the threshold by which we make a new split in the tree. A larger threshold make the model more conservative as it will require larger improvement for justifying a split.
- Tree **max_depth** (default = 6, range 0-infinity): limit the maximum depth of the tree. 0 means no limitation which comes with big increase in complexity and memory.³
- **min_child_weight** (default = 1, range 0-infinity): if the child is left with very small number of samples, stop the tree in the previous branch.³

- **subsample** (default = 1): defines the amount of data is being used for building each tree (entire data set/ fractions of it). Keeping sub-sample value at 1 means that it used the whole dataset, which may result in Model Over-fitting.³
- **colsample_bytree, colsample_bylevel, colsample_bynode** [default=1]: Hyper-parameters that defines subsampling of specific features (columns) along different stages of the tree. **bynodel** sub-sample at the split level and **bylevel** sub sample before the split to decide which features to use for the split.³
- **Lambda / L2 Regularization Term (Ω)** (default = 1): Control the maximum weight each feature can have during the training, making sure no feature dominant the rest. **It does it by penalizing the squared weight of features.** Useful in high-dimensional data-sets where feature selection is more crucial.³
- **Alpha / L1 Regularization Term (Ω)** (default =0): Control the maximum weight each feature can

have during the training, making sure no feature dominant the rest. It does it by penalizing the absolute weight values. It's effective in reducing the impact of outliers and noise by smoothing the learning parameters. 3

- max_leaves (default =0): control the number of leaves in each tree. Useful to prevent **Model Over-fitting** because without limits the tree can grow and learn too much of the noise in the data. ³

1. <https://xgboost.readthedocs.io/en/stable/tutorials/model.html>
2. [ML_decision_tree_illustration](#)
3. ChatGPT