	Parameter	Acceptable Values	Default Values	Description	Impact
Boosting Parameters	learning_rate	Float	0.1	the effect of each tree on the outcome is shrunk by this factor.	> lower always preferred > inversely proportional to n_estimators > use high value for tuning and lower for final submissions
	n_estimators	Integer	100	the number of trees to fit sequentially	> tune using CV for a given learning rate > higher value for low learning rate but computationally expensive
	subsample	Float (0,1]	1	the fraction of observations to be used in individual tree	> typical value 0.8 > reduces variance in model > tune using CV
Tree-Specific Parameters	min_samples_split	Int (0,inf)	2	the minimum number of observations required to split an internal node	> higher values prevent overfitting > risk of underfitting with too high values > tune using CV > ~0.5-2% of total observations
	min_samples_leaf	Int (0,inf)	1	the minimum number of observations required in a terminal node for a split to be valid	> higher values prevent overfitting > risk of underfitting with too high values > tune using CV
	min_weight_fraction_leaf	Float	0	same as min_samples_leaf but defined as a ratio of total number of observations	> higher values prevent overfitting > risk of underfitting with too high values > tune using CV
	max_depth	Integer or None	None	the maximum depth of each tree. None specified no limit on depth.	> lower values prevent overfitting > risk of underfitting with too low values > tune using CV > typical 5-20
	max_leaf_nodes	Integer or None	None	the maximum number of terminal nodes in each tree. If None then no limit	> lower values prevent overfitting > risk of underfitting with too low values > tune using CV
	max_features	Integer, Float, String or None	None	the number of features to consider for each split	> typical sqrt to 30-40% of total features > possible values: - Int: actual number of features - Float: the fraction of total number of features - 'auto' or 'sqrt': square root of total features - 'log2': log to base 2 of total features - None: equal to total number of features
Miscellaneous Parameters	loss	String for different loss functions	deviance (class) ls (regression)	the cost function to be minimized by optimization	> use default value if not sure > options: - class: deviance / exponential - regr: ls / lad / huber / quantile
	init	BaseEstimator or None	None	estimator for getting the initial predictions	> BaseEstimator can be any class object with fit and predict functions
	random_state	Integer, RandomState instance or None	None	set the seed of random number generator	> use fixed value for parameter tuning > try different values only if possible computationally
	verbose	Integer	0	select the type of output to be generated by model fit	>options: - 0: No output - 1: output for few trees (selected automatically) - >1: output all trees
	warm_start	Boolean	False	if True, it will start from the previous call to fit and add more trees to it	> should be used judicially for advanced applications
	presort	Boolean or 'auto'	'auto'	whether to presort data to speed up the finding of best splits	> mostly 'auto' works fine