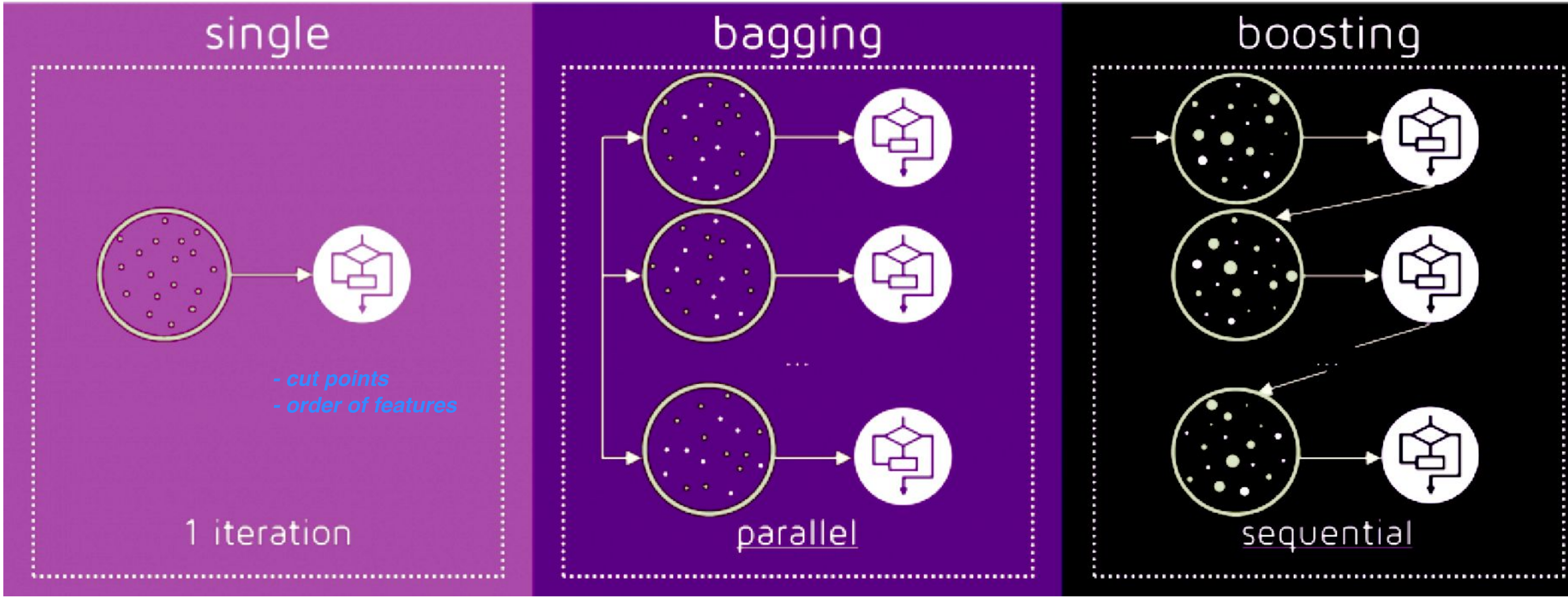
Extreme Gradient Boosting

XGBoost

XGboost is simpler than Deep Learning so before trying Deep Learning, it’s worth trying XGboot.

Tree-Based Models

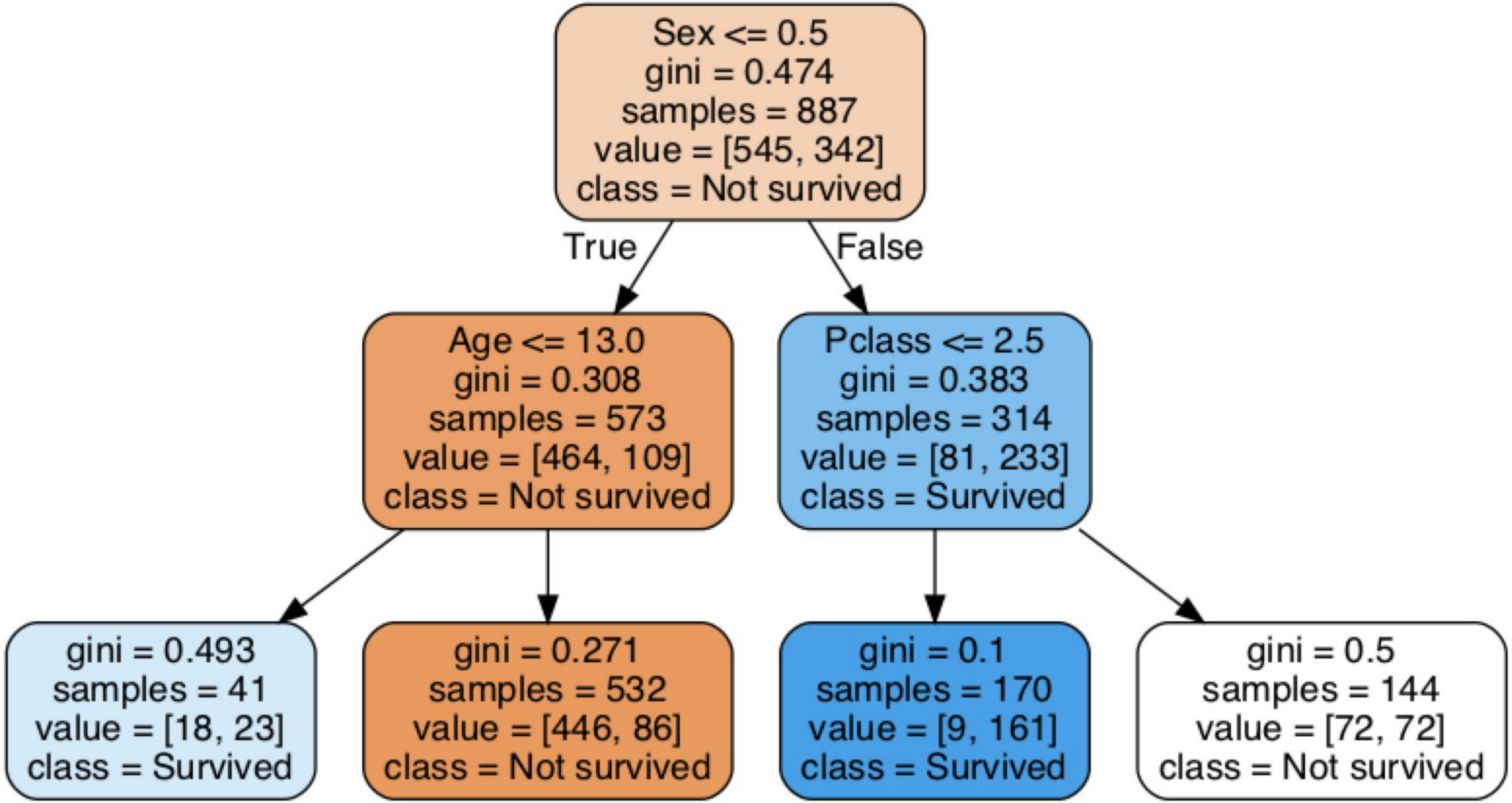
Ensemble modeling



- cut points
- order of features

Result is average of each bag.
Since it’s parallel, it’s faster than boosting.

Single Decision Tree



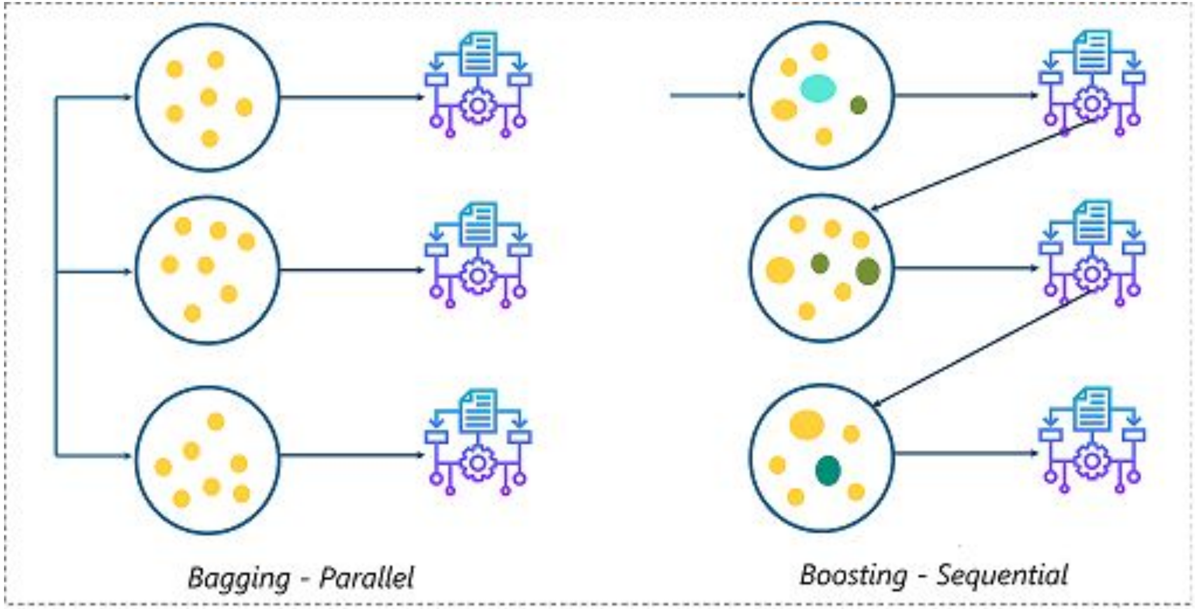
Bagging vs Boosting

Bagging algorithms:

- Bagging Ensemble
- Random Forest
- Extra Trees

Boosting algorithms:

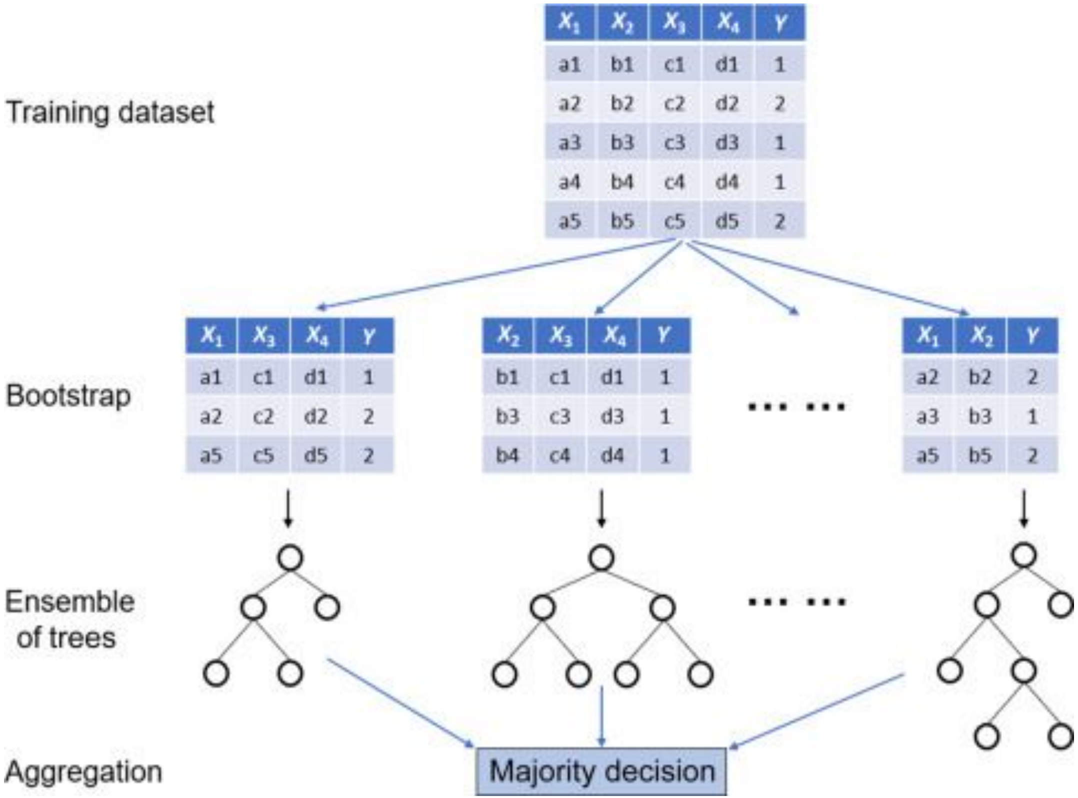
- Adaboost
- Gradient Boosting
- XGBoost



Random Forest

- Bootstrap samples
- Random selection of features
- Run in parallel

- increases generability
- decreases overfit



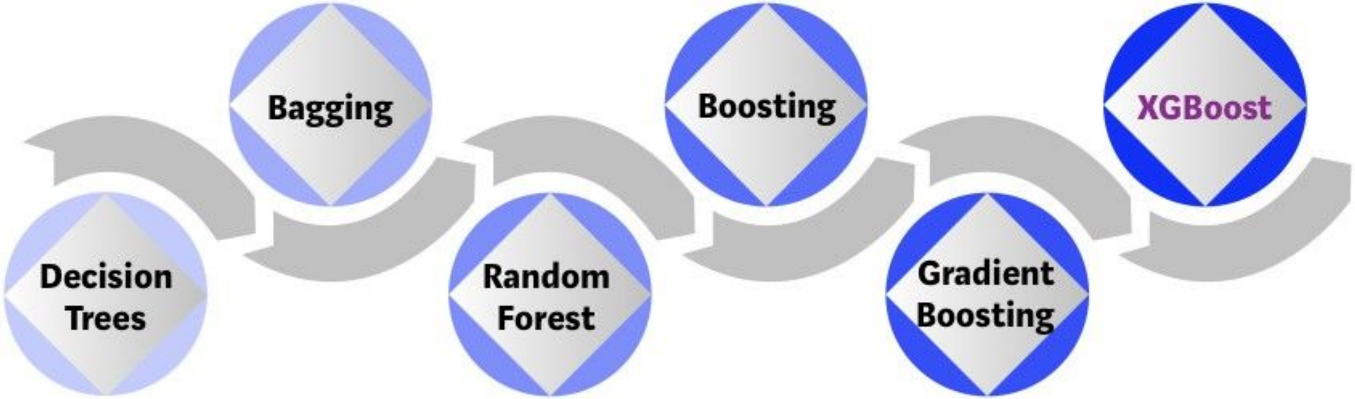
X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

Bagging vs Boosting



Decision Trees

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

Bagging

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

Random Forest

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

Boosting

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

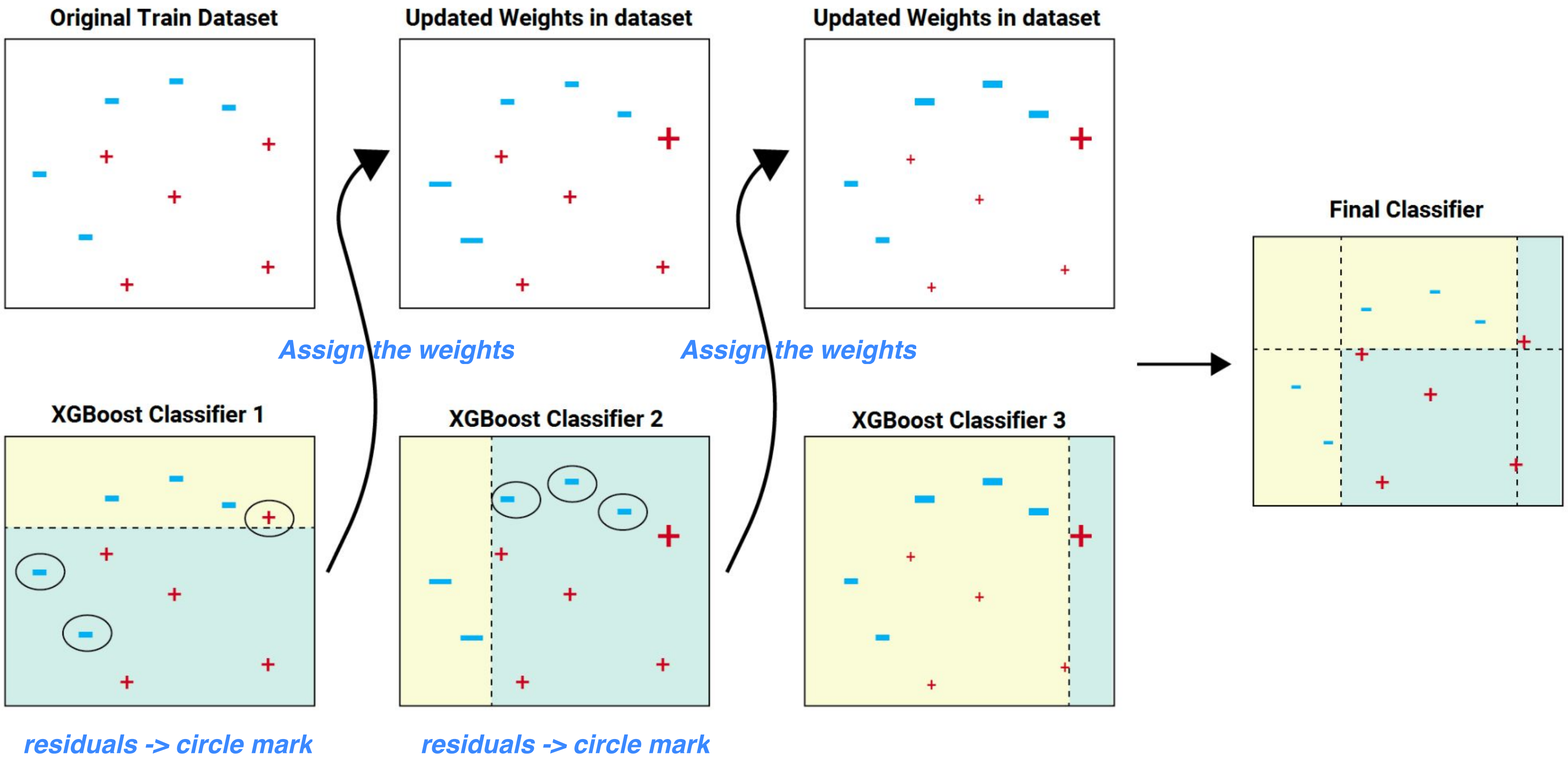
Gradient Boosting

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

XGBoost

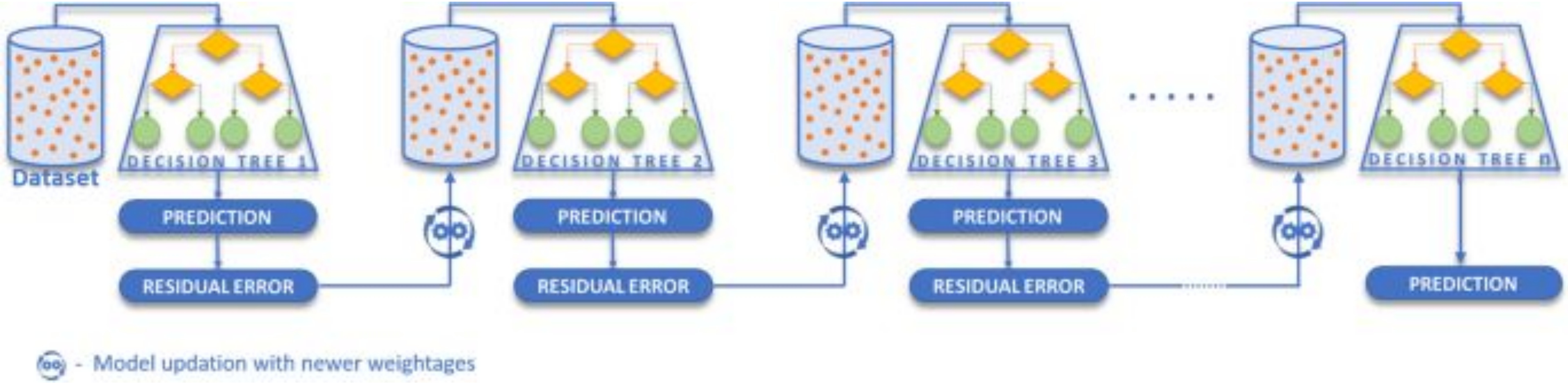
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias

XGBoost (Extreme Gradient Boosting)



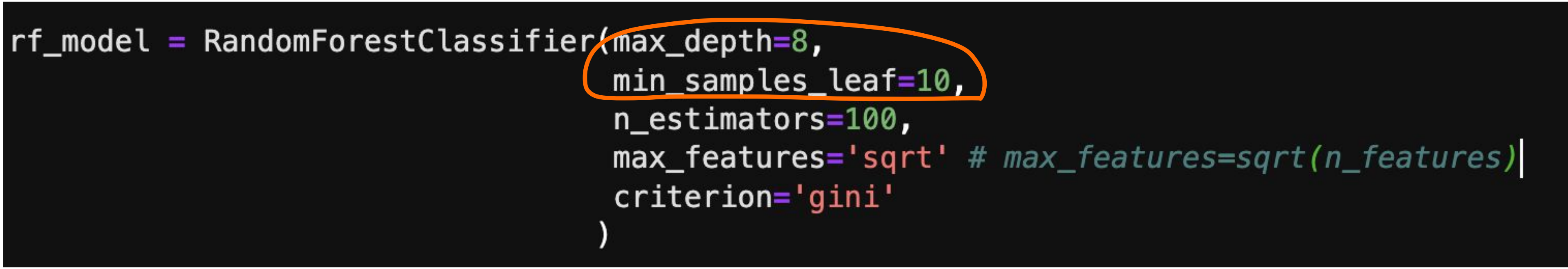
Neural network does the back-propagation.

XGBoost models the residuals



overfitting!! - regularization

Hyperparameter tuning: Random Forest

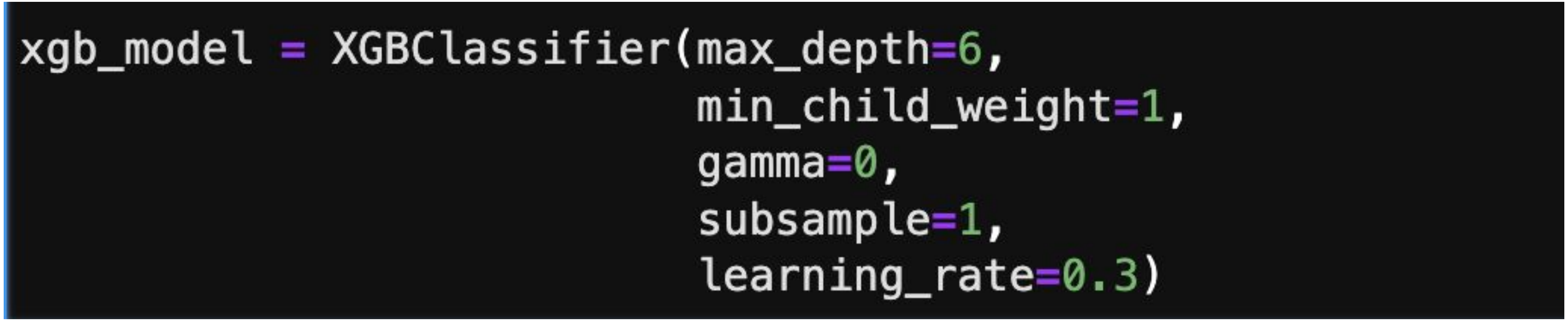


```
rf_model = RandomForestClassifier(max_depth=8,
                                  min_samples_leaf=10,
                                  n_estimators=100,
                                  max_features='sqrt' # max_features=sqrt(n_features)
                                  criterion='gini'
                                 )
```

- max_depth: (default none): The maximum depth of the tree.
- min_samples_leaf (default=1): The minimum number of samples required to be at a leaf node.
- n_estimators (default=100): The number of trees in the forest.
- max_features {“sqrt”, “log2”, None}: The number of features to consider when looking for the best split
- n_jobs (default: none): The number of jobs to run in parallel
- class_weight*{“balanced”, “balanced_subsample”}, default=None*

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Hyperparameter tuning: XGBoost



```
xgb_model = XGBClassifier(max_depth=6,
                          min_child_weight=1,
                          gamma=0,
                          subsample=1,
                          learning_rate=0.3)
```

- **min_child_weight** (default=1): Minimum sum of instance weight (hessian) needed in a child.
- **gamma** (default=0): Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be.
- **subsample** (default=1): Subsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees
- **learning_rate** (default=0.3): Step size shrinkage used in update to prevents overfitting.

https://xgboost.readthedocs.io/en/stable/parameter.html

Two ways to control overfitting in XGBoost

1. Directly control model complexity.

   - This includes max_depth, min_child_weight and gamma.

2. Add randomness to make training robust to noise.

   - This includes subsample and colsample_bytree.
   - You can also reduce stepsize eta. Remember to increase num_round when you do so.

https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html