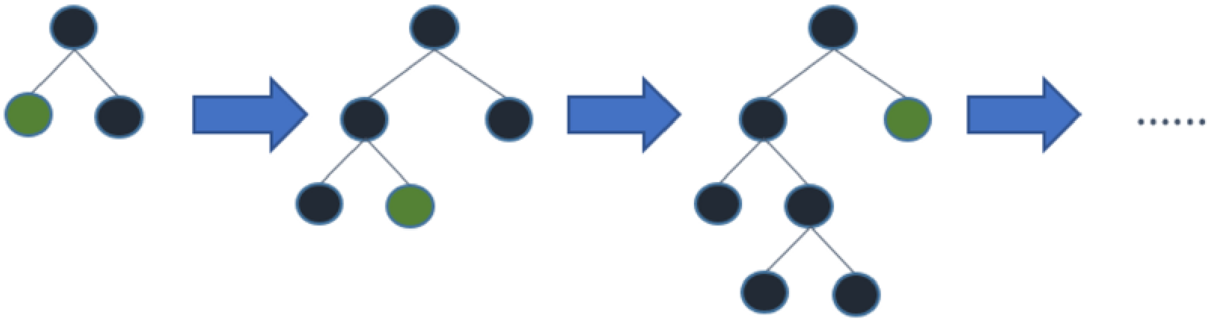
LightGBM

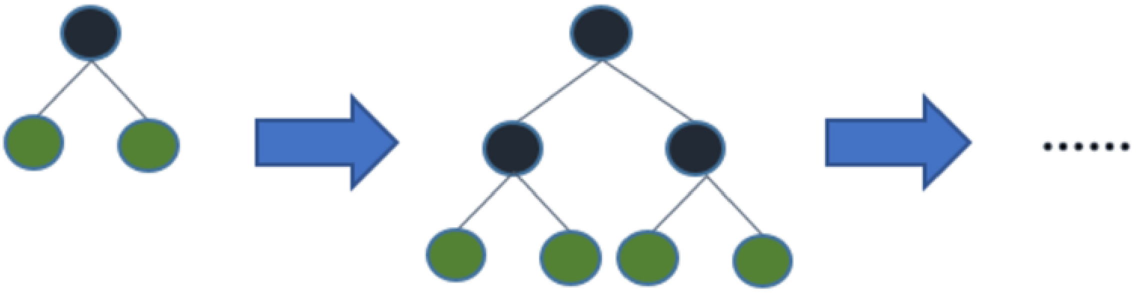
LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel, distributed, and GPU learning.
- Capable of handling large-scale data.

While other algorithms grow trees horizontally, Light GBM grows trees vertically, which translates to Light GBM growing trees leaf-wise while other algorithms grow levels-wise. The leaf with the greatest delta loss will be chosen to grow. Leaf-wise method can reduce loss more than a level-wise strategy when expanding the same leaf.



Leaf-wise tree growth



Level-wise tree growth

Light GBM is prefixed as ‘Light’ because of its high speed. Light GBM can handle the large size of data and takes lower memory to run.

Control Parameters

- **max_depth:** It describes the maximum depth of a tree. To manage overfitting of the model, utilize this parameter.
- **min_data_in_leaf:** It is the minimum number of the records a leaf has. The default value is 20.
- **feature_fraction:** % of parameters randomly selected in each iteration for building trees.
- **bagging_fraction:** It is typically used to speed up training and prevent overfitting. It sets the percentage of data to be used for each iteration.
- **early_stopping_round:** If one validation data measure does not improve in the last early stopping round rounds, the model will cease training. This will cut down on unnecessary iterations.
- **lambda:** lambda specifies the level of regularization. Typical value ranges from 0 to 1.
- **min gain to split:** This value will specify the required minimum gain to split. It can be used to control the number of useful splits in a tree.
- **max_cat_group:** Finding the split point on it is easily over-fitting when the number of categories is huge. In order to combine them, LightGBM creates groups named "max cat group" and determines the split points on the group boundaries (default: 64).

Core Parameters

- **Task:** It specifies the task you want to perform on data, train or predict.

- **application:** This is the most important parameter and specifies the application of your model,
    - regression: regression
    - binary: binary classification
    - multiclass: multiclass classification

- **boosting:** defines the type of algorithm you want to run, default=gdbt
    - gbdt: traditional Gradient Boosting Decision Tree
    - rf: random forest
    - dart: Dropouts meet Multiple Additive Regression Trees
    - goss: Gradient-based One-Side Sampling

- **learning_rate:** the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. Typical values: 0.1, 0.001, 0.003.

- **num_leaves:** number of leaves in full tree, default: 31

- **device:** cpu, gpu, default is cpu