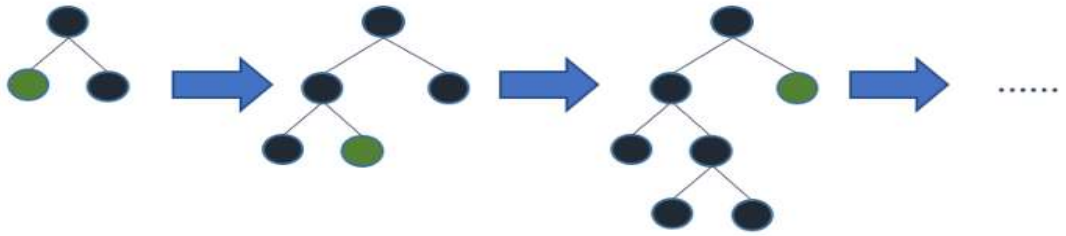
Modèles LightGBM

5. Modèles – Boosting - LightGBM

LightGBM :

LightGBM est un cadre GBDT open-source créé par Microsoft (2017) comme une alternative rapide et évolutive à XGB et GBM. Par défaut, LightGBM entraîne un arbre de décision à boost de gradient (GBDT), mais il prend également en charge les forêts aléatoires, les arbres de régression additifs multiples (DART) et l'échantillonnage unilatéral basé sur le gradient (Goss).

Light GBM est un cadre rapide, distribué et très performant de boosting de gradient basé sur l'algorithme de l'arbre de décision, basé sur la croissance de l'arbre par feuilles, contrairement aux autres algorithmes qui fonctionnent selon une approche par niveaux. Il divise l'arbre **par feuille** avec le meilleur ajustement alors que d'autres algorithmes de boosting divisent l'arbre par profondeur ou par niveau plutôt que par feuille. Ainsi, lors de la croissance sur la même feuille dans Light GBM, l'algorithme par feuille peut réduire plus de pertes que l'algorithme par niveau, ce qui donne une bien meilleure précision, rarement atteinte par les algorithmes de boosting existants. En outre, il est étonnamment très rapide, d'où le terme "léger".



Leaf-wise tree growth

5. Modèles – Boosting - LightGBM

Avantages :

- Vitesse de formation plus rapide et efficacité accrue : Light GBM utilise un algorithme basé sur l'histogramme, c'est-à-dire qu'il classe les valeurs de caractéristiques continues dans des cases discrètes, ce qui accélère la procédure de formation.
- Utilisation réduite de la mémoire : Remplace les valeurs continues par des cases discrètes, ce qui réduit l'utilisation de la mémoire.
- Meilleure précision que tout autre algorithme de boosting : Il produit des arbres beaucoup plus complexes en suivant une approche de division par feuille plutôt qu'une approche par niveau, ce qui est le principal facteur permettant d'obtenir une meilleure précision. Cependant, il peut parfois conduire à un surajustement qui peut être évité en définissant le paramètre max_depth.
- Compatibilité avec les grands ensembles de données : Il est capable de réaliser d'aussi bonnes performances avec de grands ensembles de données avec une réduction significative du temps de formation par rapport à XGBOOST.
- Prise en charge de l'apprentissage parallèle.

Inconvénients :

- Over-fitting : Les divisions en feuilles entraînent une augmentation de la complexité et peuvent conduire à un surajustement, ce qui peut être résolu en spécifiant un autre paramètre, max-depth, qui spécifie la profondeur à laquelle la division aura lieu.
- Récent : moins de documentation mais en progression

Hyperparamètres :

max_depth : Spécifie la profondeur maximale à laquelle l'arbre se développera. Ce paramètre est utilisé pour gérer l'overfitting.

min_data_in_leaf : Nombre minimum de données dans une feuille.

feature_fraction : default=1 ; spécifie la fraction de caractéristiques à prendre pour chaque itération.

bagging_fraction : default=1 ; spécifie la fraction de données à utiliser pour chaque itération et est généralement utilisé pour accélérer l'apprentissage et éviter l'overfitting.

task : valeur par défaut = train ; options = train , prediction ; Spécifie la tâche que nous souhaitons effectuer qui est soit train soit prediction.

application : valeur par défaut=regression, type=enum, options= options :

regression : exécuter la tâche de régression

binary : classification binaire

multiclass : Classification multiclasse

lambdarank : application lambdarank

data : type=string ; données d'entraînement, LightGBM s'entraînera à partir de ces données

num_iterations : nombre d'itérations de boosting à effectuer ; default=100 ; type=int

num_leaves : nombre de feuilles dans un arbre ; default = 31 ; type =int

device : default= cpu ; options = gpu,cpu. Dispositif sur lequel nous voulons entraîner notre modèle. Choisissez GPU pour un entraînement plus rapide.

min_gain_to_split : default=.1 ; gain min pour effectuer le fractionnement

max_bin : nombre maximum de bacs pour mettre les valeurs des caractéristiques dans des cases.

min_data_in_bin : nombre minimum de données dans un bin.

num_threads : default=OpenMP_default, type=int ;Nombre de threads pour Light GBM.

label : type=string ; spécifier la colonne label

categorical_feature : type=string ; spécifie les caractéristiques catégorielles que nous voulons utiliser pour l'entraînement de notre modèle

num_class : default=1 ; type=int ; utilisé uniquement pour la classification multi-classes