Modèles Random Forest

5. Modèles – Bagging - RandomForest

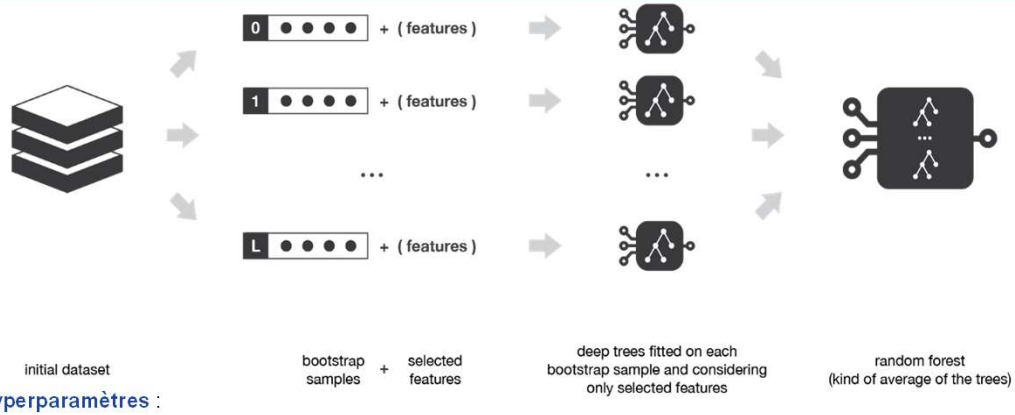
RandomForest :
Méthode de bagging où les arbres profonds (low bias, high variance), ajustés sur des échantillons bootstrap de données et des sous-ensembles aléatoires de features, sont combinés (moyenne des prédictions pour tous les arbres pour la régression) pour produire une sortie avec une variance plus faible.

1. *greedy, top-down recursive partitioning algorithm* : faire grandir l'arbre (feature à utiliser et point de séparation pour minimiser l'erreur quadratique)
2. Elaguer l'arbre : réduire l'over-fitting en prenant en compte le coût de complexité du modèle.

Avantages *:* amélioration du bagging puisqu'elles créent des arbres plus différents structurellement en échantillonnant aléatoirement seulement un sous-ensemble des features disponibles, ce qui permet de réduire la corrélation entre les arbres et donc d'obtenir un meilleur modèle final. Gros datasets. Interprétable. Estimation de l'importance des features. Rapide. Pas d'over-fitting. Plus robuste aux valeurs manquantes. Hyperparamètres de base efficaces. Parallèlisable.

Désavantages : taille mémoire pour stockage gros jeux. Un nombre élevé d'arbres peut rendre le processus de calcul beaucoup plus lent et inefficace pour les prédictions en temps réel. Datasets éparses. Pas données textuelles (grandes dimensions).

5. Modèles – Bagging - RandomForest



Algorithme :

- Des sous-ensembles aléatoires sont créés à partir de l'ensemble de données original (bootstrapping).
- À chaque nœud de l'arbre de décision, seul un ensemble aléatoire de caractéristiques est pris en compte pour décider de la meilleure répartition (algo de calcul la combinaison caractéristique/séparation localement optimale).
- Un modèle d'arbre de décision est ajusté sur chacun des sous-ensembles.
- La prédiction finale est calculée en faisant la moyenne des prédictions de tous les arbres de décision.

Hyperparamètres :

Augmenter le pouvoir prédictif :

n_estimators : Nombre d'arbres que l'algorithme fait croître avant de faire la prédiction. Un nombre plus élevé d'arbres devrait augmenter les performances et rendre la prédiction plus stable, avec l'inconvénient de ralentir le calcul (élevée mieux)

max_features : correspond au maximum de caractéristiques que Random Forest est autorisé à essayer dans un arbre individuel, lors de la recherche de la meilleure division ; (=sqrt(n_feature) classification, n_features : régression)

min_sample_leaf : nombre minimum requis de feuilles pour effectuer la division sur un noeud interne.

max_depth : La forêt aléatoire comporte plusieurs arbres de décision. Ce paramètre définit la profondeur maximale des arbres.

criterion: Il définit la fonction qui doit être utilisée pour le fractionnement. La fonction mesure la qualité d'un fractionnement pour chaque caractéristique et choisit le meilleur fractionnement.

min_samples_leaf : Ceci définit le nombre minimum d'échantillons requis pour être à un nœud feuille. Une taille de feuille plus petite rend le modèle plus enclin à capturer le bruit dans les données de train.

max_leaf_nodes : Ce paramètre spécifie le nombre maximal de nœuds de feuille pour chaque arbre. L'arbre cesse de se diviser lorsque le nombre de nœuds de feuilles devient égal au nombre maximal de nœuds de feuilles (peu aider à meilleur résultat.

Augmentation de la vitesse de traitement :

n_jobs : indique au système le nombre de processeurs qu'il est autorisé à utiliser. La valeur '-1' signifie qu'il n'y a pas de limite ;

random_state : rend la sortie du modèle reproductible. Il produira toujours les mêmes résultats si vous lui donnez une valeur fixe ainsi que les mêmes paramètres et données d'entraînement.

oob_score : méthode de validation croisée de Random Forest. On appelle cela les échantillons hors-sac. Elle est très similaire à la méthode de validation croisée leave-one-out mais sans charge de calcul.