Modèles ExtraTrees

5. Modèles – Bagging - ExtraTrees

L'algorithme **Extremely Randomized Trees**, ou **Extra Trees** en abrégé, est un algorithme d'apprentissage automatique d'ensemble d'arbres de décision, lié à d'autres ensembles d'algorithmes d'arbres de décision tels que l'agrégation bootstrap (bagging) et la forêt aléatoire.
L'algorithme Extra Trees fonctionne en créant un grand nombre d'arbres de décision non élagués à partir de l'ensemble de données d'apprentissage (pas bootstrap, échantillonne sans remplacement, réduit biais).
Les prédictions sont faites en faisant la moyenne des prédictions des arbres de décision dans le cas de la régression.

L'algorithme Extra-Trees construit un ensemble d'arbres de décision ou de régression non élagués selon la procédure descendante classique. Ses deux principales différences avec les autres méthodes d'ensemble basées sur les arbres sont qu'il divise les nœuds en choisissant les caractéristiques des points de coupe de manière totalement aléatoire (pas un algorithme gourmand pour sélectionner un point de division optimal, réduit la variance) et qu'il utilise l'échantillon d'apprentissage entier (plutôt qu'une réplique bootstrap) pour faire croître les arbres.

Avantages :

- sélection aléatoire des points de fractionnement rend les arbres de décision de l'ensemble moins corrélés (mais augmente la variance donc augmenter le nombre d'arbres).
- Coût de calcul plus faible que RF ➔ plus rapide.

Inconvénients :

- sélection aléatoire ➔ jamais le même résultat lorsque l'algorithme est exécuté.
- Évaluation : faire une validation croisée, ajustement : augmenter les arbres jusqu'à variance stabilisée

5. Modèles – Bagging - ExtraTrees

Hyperparamètres :

n_estimators : le nombre d'arbres de décision dans l'ensemble.

max_features : le nombre de caractéristiques d'entrée à sélectionner aléatoirement et à prendre en compte pour chaque point de séparation. Les bonnes valeurs empiriques par défaut sont max_features=None (toujours considérer toutes les caractéristiques au lieu d'un sous-ensemble aléatoire) pour les problèmes de régression, et max_features="sqrt" (utiliser un sous-ensemble aléatoire de taille sqrt(n_features)) pour les tâches de classification (où n_features est le nombre de caractéristiques dans les données).

min_samples_split :le nombre minimum d'échantillons requis dans un nœud pour créer un nouveau point de séparation.

min_samples_leaf : Le nombre minimum d'échantillons requis pour être à un noeud feuille. Un point de séparation à n'importe quelle profondeur ne sera considéré que s'il laisse au moins min_samples_leaf échantillons d'entraînement dans chacune des branches gauche et droite. Cela peut avoir pour effet de lisser le modèle, en particulier dans la régression.

max_depth : profondeur , de bons résultats sont souvent obtenus en définissant max_depth=None (tout déployé)

bootstrap : un paramètre optionnel qui permet aux utilisateurs d'utiliser des répliques bootstrap, mais par défaut, il utilise l'échantillon d'entrée entier. Cela peut augmenter la variance car le bootstrap la rend plus diversifiée.

n_jobs : construction parallèle des arbres et calcul parallèle des prédictions

`min_impurity_decrease : Un nœud sera divisé si cette division induit une diminution de l'impureté supérieure ou égale à cette valeur.`

`random_state` : valeur aléatoire

`warm_start` : Lorsqu'elle est définie sur True, la solution de l'appel précédent à l'ajustement est réutilisée et d'autres estimateurs sont ajoutés à l'ensemble, sinon, une nouvelle forêt est ajustée.

Verbose : Contrôle la verbosité lors de l'ajustement et de la prédiction.

Criterion : fonction permettant de mesurer la qualité d'un fractionnement. Les critères pris en charge sont "mse" pour l'erreur quadratique moyenne, qui est égale à la réduction de la variance comme critère de sélection des caractéristiques, et "mae" pour l'erreur absolue moyenne.