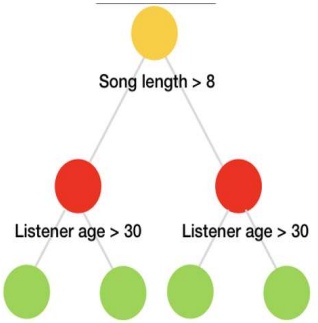
5

Modèles CatBoost

5. Modèles – Boosting - CatBoost

CatBoost est un algorithme d'apprentissage automatique récemment (2017) mis en libre accès par Yandex (russes) qui s'appuie sur la théorie des arbres de décision et du boosting de gradient.
Category : la bibliothèque fonctionne bien avec plusieurs catégories de données, telles que l'audio, le texte, l'image, y compris les données historiques.
Boosting : "Boost" vient de l'algorithme d'apprentissage automatique Gradient Boosting, car cette bibliothèque est basée sur la bibliothèque Gradient Boosting.

Dans la procédure de croissance des arbres de décision, CatBoost ne suit pas les modèles similaires de boosting de gradient. Au lieu de cela, CatBoost fait croître des arbres oblivious, ce qui signifie que les arbres sont cultivés en imposant la règle selon laquelle tous les nœuds au même niveau, testent le même prédicteur avec la même condition, et donc l'index d'une feuille peut être calculé avec des opérations bitwise. La procédure d'arbre oblivious permet un schéma d'ajustement simple et efficace sur les CPU, tandis que la structure de l'arbre fonctionne comme une régularisation pour trouver une solution optimale et éviter l'overfitting.



Avantage :

CatBoost peut traiter automatiquement les variables catégorielles, les données audio, texte.., ne nécessite pas de prétraitement approfondi des données comme d'autres algorithmes d'apprentissage automatique. Performant .Robuste (peu de réglage d'hyperparamètres). Feature importances. Utilisation GPU. Gère valeurs manquantes. Régression/Classification.

Inconvénient :

- Long si pas GPU. Récent : pas de documentation et peu d'exemples.

5. Modèles – Boosting - CatBoost

Hyperparamètres :

loss_function : alias as objective - Métrique utilisée pour l'apprentissage. Il s'agit de métriques de régression telles que l'erreur quadratique moyenne pour la régression et le logloss pour la classification.

Iterations : Le nombre maximum d'arbres qui peuvent être construits. Le nombre final d'arbres peut être inférieur ou égal à ce nombre (1000 par défaut).

learning_rate : - Le taux d'apprentissage qui détermine la rapidité ou la lenteur de l'apprentissage du modèle. La valeur par défaut est généralement de 0,03. Utilisé pour réduire le pas du gradient.

border_count : Il spécifie le nombre de splits pour les caractéristiques numériques. Il est similaire au paramètre max_bin.

depth : Définit la profondeur des arbres.

eval_metric - Métrique utilisée pour détecter le surajustement.

random_seed : La graine aléatoire utilisée pour l'apprentissage.

l2_leaf_reg alias reg_lambda - Coefficient du terme de régularisation L2 de la fonction de coût. La valeur par défaut est 3,0.

min_data_in_leaf alias min_child_samples - Il s'agit du nombre minimum d'échantillons d'entraînement dans une feuille. Ce paramètre est uniquement utilisé avec les politiques Lossguide et Depthwise growing.

max_leaves alias num_leaves - Ce paramètre est utilisé uniquement avec la politique Lossguide et détermine le nombre de feuilles dans l'arbre.

ignored_features - Indique les caractéristiques qui doivent être ignorées dans le processus de formation.

bootstrap_type - Détermine la méthode d'échantillonnage pour les poids des objets, par exemple Bayesian, Bernoulli, MVS et Poisson.

grow_policy - Détermine la façon dont l'algorithme de recherche gloutonne sera appliqué. Il peut être soit SymmetricTree, Depthwise, ou Lossguide. SymmetricTree est la valeur par défaut. Dans SymmetricTree, l'arbre est construit niveau par niveau jusqu'à ce que la profondeur soit atteinte. À chaque étape, les feuilles de l'arbre précédent sont divisées avec la même condition. Lorsque l'option Depthwise est choisie, un arbre est construit étape par étape jusqu'à ce que la profondeur spécifiée soit atteinte. À chaque étape, toutes les feuilles non terminales du dernier niveau de l'arbre sont séparées. Les feuilles sont divisées en utilisant la condition qui entraîne la meilleure amélioration des pertes. Dans Lossguide, l'arbre est construit feuille par feuille jusqu'à ce que le nombre de feuilles spécifié soit atteint. À chaque étape, la feuille non terminale présentant la meilleure amélioration des pertes est divisée.

nan_mode - Méthode de traitement des valeurs manquantes. Les options sont Interdit, Min et Max. La valeur par défaut est Min. Lorsque l'option Forbidden est utilisée, la présence de valeurs manquantes entraîne des erreurs. Avec Min, les valeurs manquantes sont considérées comme les valeurs minimales pour cette caractéristique. Avec Max, les valeurs manquantes sont traitées comme la valeur maximale pour cette caractéristique.

leaf_estimation_method - La méthode utilisée pour calculer les valeurs dans les feuilles. Dans la classification, 10 itérations de Newton sont utilisées. Les problèmes de régression utilisant la perte quantile ou MAE utilisent une itération Exact. La classification multiple utilise une itération Netwon.

leaf_estimation_backtracking - Le type de backtracking à utiliser pendant la descente de gradient. La valeur par défaut est AnyImprovement. AnyImprovement réduit le pas de descente, jusqu'à ce que la valeur de la fonction de perte soit plus petite que lors de la dernière itération. Armijo réduit le pas de descente jusqu'à ce que la condition Armijo soit remplie.

5. Modèles – Boosting - CatBoost

Hyperparamètres :

boosting_type - Le schéma de boosting. Il peut être simple pour le schéma classique de boosting par gradient, ou ordonné, qui offre une meilleure qualité sur les petits ensembles de données.

score_function - Le type de score utilisé pour sélectionner la prochaine division pendant la construction de l'arbre. Cosinus est l'option par défaut. Les autres options disponibles sont L2, NewtonL2 et NewtonCosine.

early_stopping_rounds - Lorsque True, définit le type de détecteur d'overfitting à Iter et arrête l'apprentissage lorsque la métrique optimale est atteinte.

classes_count - Le nombre de classes pour les problèmes de multi-classification.

task_type - Si vous utilisez un CPU ou un GPU. CPU est la valeur par défaut.

devices - Les ID des périphériques GPU à utiliser pour la formation.

cat_features - Le tableau avec les colonnes catégoriques.

text_features - Utilisé pour déclarer les colonnes de texte dans les problèmes de classification.

Avantage :

- CatBoost peut traiter automatiquement les variables catégorielles et ne nécessite pas de prétraitement approfondi des données comme d'autres algorithmes d'apprentissage automatique. Performant .
- Robuste (peu de réglage d’hyperparamètres).
- Feature importances.
- Utilisation GPU.
- Gère valeurs manquantes.
- Elle produit des résultats de pointe sans la formation intensive des données généralement requise par les autres méthodes d'apprentissage automatique, et
- Elle offre une prise en charge puissante et prête à l'emploi des formats de données plus descriptifs qui accompagnent de nombreux problèmes commerciaux.