5

Modèles GradientBoosting

5. Modèles – Boosting – Gradient Boosting

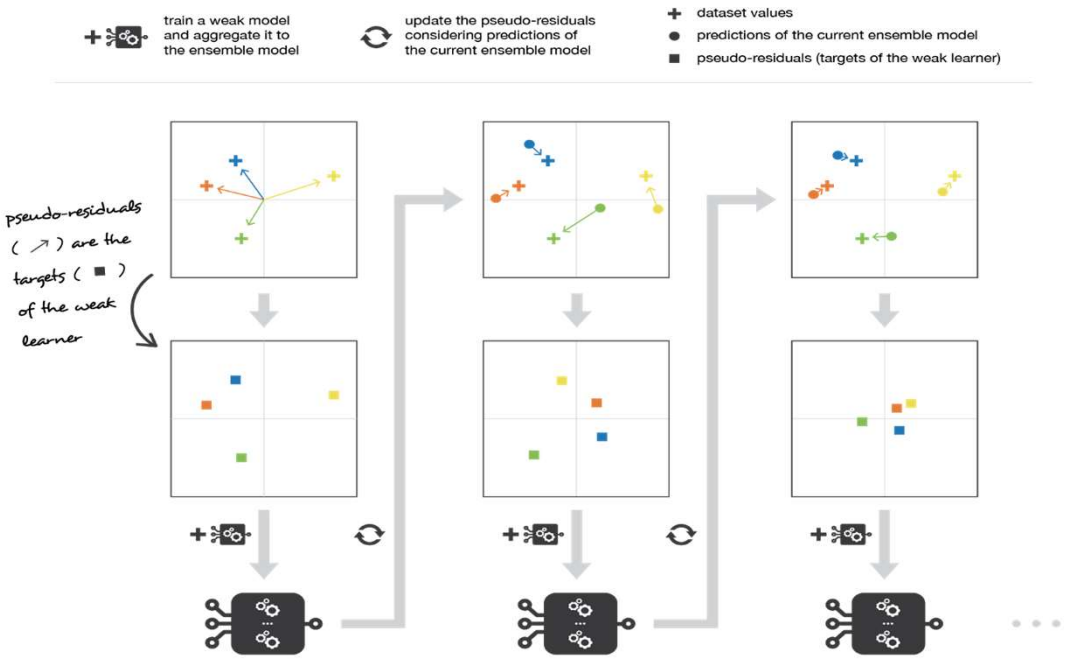
Gradient Boosting :
boosting gradient met à jour la valeur de ces observations.
Le modèle d'ensemble que nous essayons de construire est également une somme pondérée d'apprenants faibles.
L'optimisation globale est difficile ➔ optimisation séquentielle, le boosting de gradient transforme le problème en une descente de gradient : à chaque itération, nous ajustons un apprenant faible à l'opposé du gradient de l'erreur d'ajustement actuelle par rapport au modèle d'ensemble actuel.

$$s_l(.) = s_{l-1}(.) - c_l \times \nabla_{s_{l-1}} E(s_{l-1})(.)$$

où E(.) est l'erreur d'ajustement du modèle donné, c_l est un coefficient correspondant à la taille du pas et

$$-\nabla_{s_{l-1}} E(s_{l-1})(.)$$

est l'opposé du gradient de l'erreur d'ajustement par rapport au modèle d'ensemble au pas l-1.



L itérations :

- ajuster le meilleur apprenant faible possible aux pseudo-résidus (approximer l'opposé du gradient par rapport à l'apprenant fort actuel)
- calculer la valeur de la taille de pas optimale qui définit de combien nous mettons à jour le modèle d'ensemble dans la direction du nouvel apprenant faible
- mettre à jour le modèle d'ensemble en ajoutant le nouvel apprenant faible multiplié par la taille du pas (faire un pas de descente de gradient)
- calculer de nouveaux pseudo-résidus qui indiquent, pour chaque observation, dans quelle direction nous souhaitons mettre à jour les prédictions du modèle d'ensemble.

5. Modèles – Boosting – Gradient Boosting

ID	Married	Gender	Current City	Monthly Income	Age (target)	Mean Age (prediction 1)	Residual 1
1	Y	M	A	51,000	35	32	3
2	N	F	B	25,000	24	32	-8
3	Y	M	A	74,000	38	32	6
4	N	F	A	29,000	30	32	-2
5	N	F	B	37,000	33	32	1

ID	Age (target)	Mean Age (prediction 1)	Residual 1 (new target)	Prediction 2	Combine (mean+pred2)
1	35	32	3	3	35
2	24	32	-8	-5	27
3	38	32	6	3	35
4	30	32	-2	-5	27
5	33	32	1	3	35

ID	Age (target)	Mean Age (prediction 1)	Residual 1 (new target)	Prediction 2	Combine (mean+pred2)	Residual 2 (latest target)
1	35	32	3	3	35	0
2	24	32	-8	-5	27	-3
3	38	32	6	3	35	-3
4	30	32	-2	-5	27	3
5	33	32	1	3	35	-2

Algorithme :

- Un modèle est construit sur un sous-ensemble de données.
- En utilisant ce modèle, des prédictions sont faites sur l'ensemble des données.
- Les erreurs sont calculées en comparant la moyenne des prédictions et les valeurs réelles.
- Un nouveau modèle est créé en utilisant les erreurs calculées comme variable cible. Notre objectif est de trouver la meilleure répartition pour minimiser l'erreur.
- Les prédictions faites par ce nouveau modèle sont combinées avec les prédictions du modèle précédent.
- De nouvelles erreurs sont calculées en utilisant cette valeur prédite et la valeur réelle.

Ce processus est répété jusqu'à ce que la fonction d'erreur ne change pas, ou que la limite maximale du nombre d'estimateurs soit atteinte.

Hyperparamètres spécifiques à l'arbre : Ils affectent chaque arbre individuel du modèle.

min_samples_split : Définit le nombre minimum d'échantillons (ou d'observations) requis dans un nœud pour être pris en compte pour le fractionnement.
Utilisé pour contrôler le surajustement. Des valeurs plus élevées empêchent un modèle d'apprendre des relations qui pourraient être très spécifiques à l'échantillon particulier sélectionné pour un arbre.
Des valeurs trop élevées peuvent conduire à un sous-ajustement et doivent donc être réglées à l'aide de CV.

min_samples_leaf : Définit les échantillons (ou observations) minimums requis dans un nœud terminal ou une feuille.
Utilisé pour contrôler l'excès d'ajustement de manière similaire à min_samples_split.
En général, des valeurs plus faibles doivent être choisies pour les problèmes de classes déséquilibrées car les régions dans lesquelles la classe minoritaire sera majoritaire seront très petites.

min_weight_fraction_leaf : Similaire à min_samples_leaf mais défini comme une fraction du nombre total d'observations au lieu d'un nombre entier.
Une seule des options #2 et #3 doit être définie.

max_depth : La profondeur maximale d'un arbre. Utilisé pour contrôler l'excès d'ajustement car une profondeur plus élevée permettra au modèle d'apprendre des relations très spécifiques à un échantillon particulier.
Doit être réglé à l'aide de CV.

max_leaf_nodes : Le nombre maximum de nœuds terminaux ou de feuilles dans un arbre.
Peut être défini à la place de max_depth. Comme les arbres binaires sont créés, une profondeur de 'n' produirait un maximum de 2^n feuilles.
Si cette valeur est définie, GBM ignorera max_depth.

max_features : Le nombre de caractéristiques à prendre en compte lors de la recherche de la meilleure répartition. Elles seront sélectionnées aléatoirement.
En règle générale, la racine carrée du nombre total de caractéristiques fonctionne bien mais nous devrions vérifier jusqu'à 30-40% du nombre total de caractéristiques.
Des valeurs plus élevées peuvent conduire à un ajustement excessif, mais cela dépend de chaque cas.

5. Modèles – Boosting – Gradient Boosting

Paramètres de boosting : Ils affectent l'opération de boosting dans le modèle.

learning_rate : cela détermine l'impact de chaque arbre sur le résultat final (étape 2.4). Le GBM fonctionne en commençant par une estimation initiale qui est mise à jour en utilisant la sortie de chaque arbre. Le paramètre d'apprentissage contrôle l'ampleur de ce changement dans les estimations. Contrôle le dégré 'intensité avec lequel chaque arbre essaie de corriger les erreurs de l'arbre précédent. Valeurs élevées ➔ modèle plus complexe.
Des valeurs faibles sont généralement préférées car elles rendent le modèle robuste aux caractéristiques spécifiques de l'arbre et lui permettent ainsi de bien se généraliser.
Des valeurs plus faibles nécessiteraient un plus grand nombre d'arbres pour modéliser toutes les relations et seraient coûteuses en termes de calcul.

n_estimators : Le nombre d'arbres séquentiels à modéliser (étape 2). Valeur élevée ➔ modèle plus complexe.
Bien que le GBM soit assez robuste pour un nombre élevé d'arbres, il peut toujours être surajusté à un moment donné. Par conséquent, ce nombre doit être ajusté en utilisant CV pour un taux d'apprentissage particulier.

subsample : La fraction d'observations à sélectionner pour chaque arbre. La sélection se fait par échantillonnage aléatoire.
Des valeurs légèrement inférieures à 1 rendent le modèle robuste en réduisant la variance.
Les valeurs typiques ~0.8 fonctionnent généralement bien mais peuvent être affinées davantage.

Paramètres divers : Autres paramètres pour le fonctionnement global

loss : Il s'agit de la fonction de perte à minimiser dans chaque fractionnement. Elle peut avoir différentes valeurs pour la classification et la régression. En général, les valeurs par défaut fonctionnent bien. D'autres valeurs ne devraient être choisies que si vous comprenez leur impact sur le modèle.

init : Ceci affecte l'initialisation de la sortie. Ceci peut être utilisé si nous avons créé un autre modèle dont le résultat doit être utilisé comme estimations initiales pour GBM.

random_state : La graine de nombre aléatoire afin que les mêmes nombres aléatoires soient générés à chaque fois.
Ceci est important pour le réglage des paramètres. Si nous ne fixons pas le nombre aléatoire, nous aurons des résultats différents pour les exécutions suivantes sur les mêmes paramètres et il devient difficile de comparer les modèles.
Cela peut potentiellement entraîner un surajustement à un échantillon aléatoire particulier sélectionné. Nous pouvons essayer d'exécuter des modèles pour différents échantillons aléatoires, ce qui est coûteux en termes de calcul et n'est généralement pas utilisé.

verbose : Le type de sortie à imprimer lorsque le modèle s'ajuste. Les différentes valeurs peuvent être :
0 : aucune sortie générée (par défaut)
1 : sortie générée pour les arbres dans certains intervalles
>1 : sortie générée pour tous les arbres

warm_start : Ce paramètre a une application intéressante et peut être d'une grande aide s'il est utilisé judicieusement.
Il permet d'ajuster des arbres supplémentaires sur les ajustements précédents d'un modèle. Cela peut faire gagner beaucoup de temps et vous devriez explorer cette option pour les applications avancées.

presort : Sélectionnez si vous voulez présélectionner les données pour des fractionnements plus rapides.
La sélection est automatique par défaut, mais elle peut être modifiée si nécessaire.