Université Alioune Diop de Bambey
Master 1 SIR

Projet d'Intelligence Artificielle

Sujet : LightGBM

Professeur : Dr. Seydou Nourou SYLLA

Présenté par :

Papa Sidy Mactar TRAORE (Master 1 SI)

Serigne Saliou Faye (Master 1 SI)

Sangoulé Ndao (Master 1 SI)

Ndeye Sokhna Diagne (Master 1 SR)

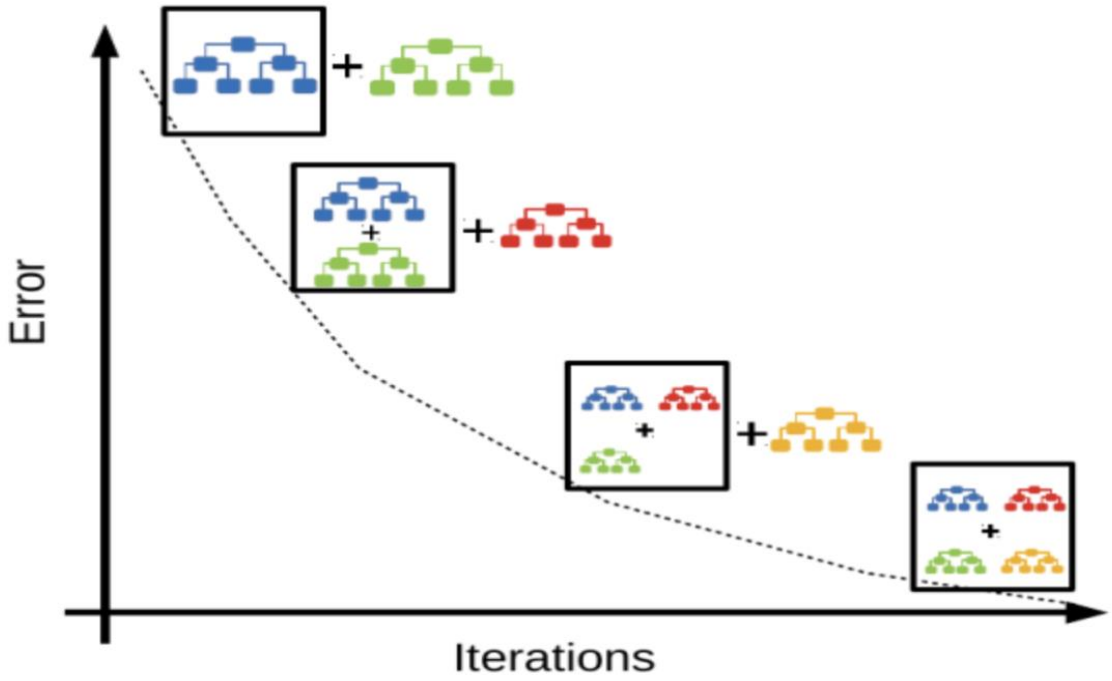
Introduction

Les modèles basés sur le renforcement de gradient et leurs variantes offertes par plusieurs communautés ont gagné beaucoup de traction ces dernières années. LightGBM (Light Gradient Boosting Machine), développé par Microsoft et publié en 2016, représente un exemple très populaire. Il est rapide, distribué et de haute performance. Basé sur des algorithmes d'arbre de décision, il est utilisé pour le classement, la classification et de nombreuses autres tâches de Machine Learning. Il représente aussi une excellente alternative à XGBoost (eXtreme Gradient Boosting).

Qu'est-ce que le renforcement de gradient ?

Avant de détailler l'algorithme, comprenons rapidement le concept fondamental de renforcement de gradient (Gradient Boosting) qui fait partie de LightGBM. Cette notion fait référence à une méthodologie de Machine Learning dans laquelle l'idée est de former plusieurs modèles utilisant le même algorithme d'apprentissage. Une combinaison de modèles individuels créant un modèle plus fort et plus puissant. Il s'agit de centaines ou de milliers d'apprenants très dépendants les uns des autres, avec un objectif commun fusionnés pour résoudre un problème. Ces modèles sont généralement des arbres de décision.

1. La première étape consiste à créer un premier modèle de base. Il est entraîné sur les données.
2. Ensuite, un second modèle est construit pour tenter de corriger les erreurs présentes dans le premier modèle. Les erreurs sont minimisées par l'algorithme de descente de gradient, chaque arbre ajouté va compenser les erreurs commises précédemment sans détériorer les prédictions qui ont été justes en changeant en cours de route la base d'apprentissage.
3. Cette procédure se poursuit et des modèles sont ajoutés jusqu'à ce que l'ensemble complet des données de formation soit prédit correctement ou que le nombre maximal de modèles soit ajouté.
4. Les prédictions du dernier modèle ajouté seront les prédictions globales fournies par les anciens modèles d'arbres.



Le renforcement de gradient est l'une des techniques les plus puissantes pour construire des modèles prédictifs. Il existe plusieurs modèles qui se basent sur cette technique dont LightGBM.

Fonctionnement de LightGBM

Light Gradient Boosted Machine, ou LightGBM pour faire court, est un modèle d'apprentissage d'ensemble séquentiel, créé par des chercheurs de Microsoft, se basant sur le renforcement de gradient et des arbres de décision (GBDT ou Gradient Boosting decision Trees).

Ces derniers sont combinés de manière à ce que chaque nouvel apprenant ajuste les résidus de l'arbre précédent afin que le modèle s'améliore. Le dernier arbre ajouté regroupe les résultats de chaque étape et un apprenant puissant est atteint.

Les arbres de décision sont construits en fractionnant les observations (c'est-à-dire les instances de données) en fonction des valeurs des variables. L'algorithme CART recherche la meilleure répartition qui se traduit par le gain d'information le plus élevé qui se calcule en utilisant des outils statistiques comme l'entropie et l'indice de Gini. Pour trouver la meilleure structure de l'arbre, celle qui apporte le plus d'information. On essaye toutes les combinaisons de points de partage possibles, on les évalue et puis on les trie, la structure avec le gain en information le plus élevé est choisie. Ce n'est certainement pas une manière optimale. Cette méthode est de plus en plus lente que la taille du jeu de données augmente est utilisée dans la plupart des GBDT.

Avec LightGBM, on vise à résoudre ce problème de temps et de puissance de calcul. Donc on introduit une nouvelle méthode de recherche de la bonne structure d'arbre pour chaque apprenant ajouté. Cette technique se base deux notions clés : GOSS (Gradient-based One-Side Sampling ou échantillonnage d'un côté en dégradé) et EFB (Exclusive Feature Bundling ou Offre groupée de fonctionnalités exclusives).

GOSS (échantillonnage d'un cote en dégradé)

GOSS est une nouvelle méthode d'échantillonnage qui échantillonne les observations en se basant sur les gradients. Les gradients donnent un aperçu précieux du gain d'information.

- **Petit gradient**: l'algorithme a été entraîné sur cette observation et l'erreur qui lui est associée est faible.
- **Grand gradient**: l'erreur associée à cette observation est importante, elle fournira donc plus d'informations.

L'approche consiste à écarter les cas ayant de petits gradients en se concentrant uniquement sur les cas ayant de grands gradients, cela modifierait la distribution des données. En un mot, GOSS conserve les instances avec de grands gradients et effectue un échantillonnage aléatoire sur les instances avec des gradients plus petits.

EFB (offre groupée de fonctionnalités exclusives)

EBF fusionne des variables pour réduire la complexité de l'entrainement. En effet c'est une méthode de réduction de dimension.

Les ensembles de données avec un nombre élevé de caractéristiques sont susceptibles d'avoir des caractéristiques rares (c'est-à-dire beaucoup de valeurs nulles). Ces caractéristiques clairsemées sont généralement mutuellement exclusives, ce qui signifie qu'elles n'ont pas de valeurs non nulles simultanément. Dans ce cas on peut utiliser la EFB pour fusionner deux variables par exemple en une seule.

Cette technique se fait généralement en deux étapes :

1. **Déterminer les caractéristiques qui pourraient être regroupées**

Cette étape consiste à construire un graphique avec les mesures du conflit entre les variables du jeu de données (La mesure de conflit est la fraction des valeurs non nulles qui se chevauchent et le nombre total des valeurs).

On va ensuite les trier et si la mesure de conflit est inférieure au seuil, on va créer une nouvelle variable .

2. **Fusionnement des variables**

Après avoir choisi les variables à fusionner, on en construit une nouvelle.

Ensemble, ces deux notions peuvent accélérer le temps d’entraînement de l’algorithme jusqu'à 20 fois. Donc, on peut dire tout simplement que LightGBM peut être considéré comme GBDT avec l’ajout de GOSS et EFB.

Comparaison de LightGBM avec XGBoost et CatBoost

	XGBoost	LightGBM	CatBoost
Construction des arbres	Par niveau ou par feuille	Par feuille	Par niveau (Arbres symétriques)
Algorithme de pondération des échantillons pour la segmentation sur les grands datasets	Aucun	Gradient-based One-Side Sampling (GOSS)	Minimum Variance Sampling (MVS)
Variables catégorielles	Non-supportées (à traiter séparément)	À encoder numériquement et indiquer lors de l'entraînement (segmentation intelligente)	À indiquer lors de la création du modèle (dummification ou target encoding automatique en fonction de la taille du dataset)
Calcul de l'importance des variables	Gain moyen sur la fonction de coût suite au noeuds où la feature est utilisée	Nombre de noeuds où la feature est utilisée	Valeur moyenne du changement de prédiction lorsqu'un changement de la valeur de la feature change la prédiction finale

Comparaison des caractéristiques des modèles

Quand utiliser LightGBM ?

Il est devenu difficile pour les algorithmes traditionnels de donner des résultats rapides, car la taille des données augmente rapidement de jour en jour. LightGBM est appelé « Light » ou léger en raison de sa puissance de calcul et de ses résultats plus rapides. Il lui faut moins de mémoire pour fonctionner et est capable de traiter de grandes quantités de données.

Contrairement aux autres modèles basés sur les arbres de décision, qui représentent des risques de sur-apprentissage avec des grands volumes de données, LightGBM n'est pas conçu pour des jeux de données de tailles peu importantes. Il peut facilement être débordé en raison de sa sensibilité au sur-ajustement .

LightGBM a une meilleure précision que n'importe quel autre algorithme GBDT : Il produit des arbres beaucoup plus complexes en suivant une approche de division foliaire plutôt qu'une approche de niveau qui est le principal facteur pour atteindre une plus grande précision. Cependant, il peut parfois conduire à un débordement qui peut être évité en réglant le paramètre max_depth.

Il est l'algorithme le plus utilisé dans les hackathons et les compétitions de Machine Learning, ainsi que les concours de Kaggle grâce à sa capacité d'obtenir une bonne précision des résultats et aussi renforcer le GPU penchant.

Explication des données utilisées pour l'implémentation

Lien kaggle vers le dataset :

https://www.kaggle.com/datasets/ealaxi/paysim1

Nom du dataset : Synthetic Financial Datasets For Fraud Detection

Contexte

Il existe un manque d'ensembles de données publiques disponibles sur les services financiers, en particulier dans le domaine émergent des transactions d'argent mobile. Les ensembles de données financières sont importants pour de nombreux chercheurs et en particulier pour nous qui effectuons des recherches dans le domaine de la détection de la fraude. Une partie du problème réside dans la nature intrinsèquement privée des transactions financières, qui conduit à l'absence d'ensembles de données accessibles au public.

Nous présentons un ensemble de données synthétiques généré à l'aide du simulateur appelé PaySim comme une approche à un tel problème. PaySim utilise les données agrégées de l'ensemble de données privées pour générer un ensemble de données synthétiques qui ressemble au fonctionnement normal des transactions et injecte des comportements malveillants pour évaluer ultérieurement les performances des méthodes de détection de la fraude.

Contenu

PaySim simule des transactions d'argent mobile à partir d'un échantillon de transactions réelles extraites des journaux financiers d'un mois d'un service d'argent mobile mis en place dans un pays africain. Les journaux originaux ont été fournis par une société multinationale, qui est le fournisseur du service financier mobile actuellement en place dans plus de 14 pays dans le monde.

Cet ensemble de données synthétiques est réduit d'un quart par rapport à l'ensemble de données original et a été créé uniquement pour Kaggle.

En-têtes

Ceci est un échantillon de 1 ligne avec l'explication des en-têtes :

1,PAYMENT,1060.31,C429214117,1089.0,28.69,M1591654462,0.0,0.0,0,0

step - correspond à une unité de temps dans le monde réel. Dans ce cas, 1 étape correspond à 1 heure de temps. Total des étapes 744 (simulation de 30 jours).

type - CASH-IN, CASH-OUT, DEBIT, PAIEMENT et TRANSFERT.

amount-
montant de la transaction en monnaie locale.

nameOrig - client qui a commencé la transaction

oldbalanceOrg - solde initial avant la transaction

newbalanceOrig - nouveau solde après la transaction.

nameDest - client qui est le destinataire de la transaction

oldbalanceDest - solde initial du destinataire avant la transaction. Notez qu'il n'y a pas d'informations pour les clients dont le nom commence par M (Merchants).

newbalanceDest - destinataire du nouveau solde après la transaction. Notez qu'il n'y a pas d'informations pour les clients dont le nom commence par M (Merchants).

isFraud - Il s'agit des transactions effectuées par les agents frauduleux dans la simulation. Dans cet ensemble de données spécifique, le comportement frauduleux des agents vise à faire du profit en prenant le contrôle des comptes des clients et en essayant de vider les fonds en les transférant vers un autre compte, puis en les encaissant hors du système.

isFlaggedFraud - Le modèle d'entreprise vise à contrôler les transferts massifs d'un compte à un autre et signale les tentatives illégales. Une tentative illégale dans cet ensemble de données est une tentative de transfert de plus de 200.000 en une seule transaction.