Université Paris-Dauphine
Machine Learning pour la finance
Examen — 9 juillet 2024

Aucun document, calculatrice ou autres objets électroniques ne sont autorisés.

Détection de virements frauduleux

Vous intégrez en alternance une équipe dédiée à la lutte contre la fraude aux moyens de paiements, plus spécifiquement le département en charge des virements. Vous avez accès à un dataset de 10 millions de lignes sur une période de temps non renseignée où 0.01% des opérations sont frauduleuses.

Votre objectif est d'être capable de détecter les opérations frauduleuses. Nous noterons `df` le dataframe Python correspondant à ce jeu de données, X la matrice correspondante aux informations présentes dans `df` et `y` la variable cible.

1. Dans quel cadre sommes-nous : apprentissage supervisé ou non-supervisé ? Et quel sous-type ?
2. Vous n'êtes pas très encadré et vous ne connaissez pas la politique de blocage qui est appliquée. Quelles sont les métriques que vous allez considérer et laquelle allez vous maximiser ?

   Votre data engineer vous présente les informations dans votre dataset : IBAN donneur d'ordre et créditeur (anonymisé), BIC donneur d'ordre et créditeur ainsi que le pays associé, montant, heure de la transaction et motif associé au virement. `df` ne contient pas de valeurs manquantes ni de valeurs aberrantes. Vous commencez donc la partie d'exploration des données, en prenant soin de vérifier ces informations tout de même.

3. Bien qu'il existe de nombreuses banques, vous remarquez qu'il n'y a au total que 10 banques différentes représentées dans le dataset. Comment traiter ces informations ?
4. La base de données est relativement pauvre. Quels indicateurs potentiellement utiles pouvez-vous créer ?
5. Quelle est la commande Python que vous allez utiliser pour séparer votre base de données en une base de données d'entraînement et une base de données de test, et pourquoi ?
   A. `X_train, X_test, y_train, y_test = train_test_split(X, y)`
   B. `X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y)`
   C. Aucune des précédentes

   Sans directions supplémentaires, vous vous lancez dans une compétition de modèle. Vous considérez les algorithmes régression logistique, Random Forest et XGBoost.

6. Pour avoir les meilleures performances, vous décidez de trouver les meilleurs paramètres pour chaque algorithme, comment vous y prenez-vous ?
7. Après ce travail réalisé, vous obtenez les résultats suivants sur le jeu de test :

Algorithme	F1-Score	Nombre d'arbres	Profondeur des arbres	Learning rate
Régression Logistique	0.05	-	-	-
Arbre de décision	0.15	1	8	-
Random Forest	0.17	25	8	-
XGBoost	0.3	25	5	0.01

   Commenter.

8. Un précédent stagiaire avait travaillé sur la même problématique et avait atteint une RMSE de 0.25 avec l'algorithme AdaBoost avec 200 arbres de profondeurs 2 et un learning rate de 0.001. Commenter.

9. Par ailleurs, dans sa compétition de modèle la personne avait sélectionné la métrique *accuracy* comme métrique à maximiser. Commenter.

10. Si l'on mesure la complexité d'un algorithme à base d'arbre en comptant 1 pour chaque coupure réalisée, et en supposant que chaque arbre se développe complètement, quel est l'écart de complexité entre votre meilleur modèle et celui proposé ?

11. A partir du tableau suivant, après avoir expliqué ce qu'est le sur-apprentissage, semble-t-il y en avoir ? Si oui, quels sont les hyper-paramètres que vous pourriez utiliser pour limiter cet effet ?

Algorithme	F1-Score Entraînement	F1-Score Validation	F1-Score Test
Régression Logistique	0.05	0.04	0.05
Arbre de décision	0.25	0.19	0.15
Random Forest	0.27	0.18	0.17
XGBoost	0.45	0.25	0.3
AdaBoost	0.40	0.20	0.25

Suite à la présentation de votre étude, on vous propose un dataset supplémentaire qui contient une base de données clients. Avec ces informations, vous devriez pouvoir améliorer la performance de vos algorithmes. Cependant, il ne vous reste que peu de temps et la base de données contient de très nombreuses colonnes. Vous décidez de réduire la dimension de cette base.

12. Dans quel cadre sommes-nous : apprentissage supervisé ou non-supervisé ? Qu'est-ce que cela change par rapport au travail précédent ?

13. Que pensez-vous de l'algorithme DBSCAN pour cette problématique ? Décrivez rapidement le fonctionnement de l'une des méthodes adaptées.

14. Comment allez-vous mesurer la performance de votre réduction de dimensions dans notre contexte ?

Préférence humaine

Une fois qu'un modèle de langage est entraîné à produire le prochain *mot*, nous devons l'entraîner à produire des réponses qui plairont aux humains. Pour le faire, pour une demande x on génère deux réponses y_1 et y_2, puis on demande à des humains de choisir la plus adaptée. On suppose qu'il existe une fonction r^* qui prend en paramètre la demande x et la réponse y et qui renvoie un nombre. Plus le nombre est grand, plus la réponse plait. On modélise la préférence humaine par :

$$\mathbb{P}\left(y_1 \text{ est préférée à } y_2 | x\right) = \frac{e^{r^*(x,y_1)}}{e^{r^*(x,y_1)} + e^{r^*(x,y_2)}}$$

On considère par la suite un dataset de triplets demande-réponses : $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)}) | i \leqslant N\}$ avec N le nombre de triplets où y_w est la réponse préférée.

L'objectif est de construire un modèle de Machine Learning qui va s'approcher le plus possible de ce modèle de récompense idéal r^*.

1. Est-ce que la modélisation présentée décrit une vrai probabilité ?

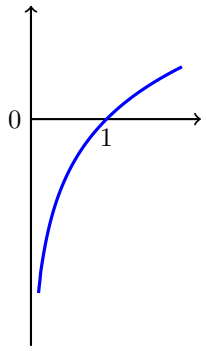
   Pour entraîner le modèle de récompense, que l'on note r, qui va s'approcher du *vrai* modèle r^*, on considère la fonction de perte :

$$\mathcal{L}(r) = -\frac{1}{N}\sum_{i=1}^{N} \ln\left[\sigma\left(r(x, y_1) - r(x, y_2)\right)\right]$$

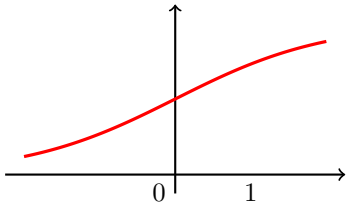
   Avec $\sigma(x) = \dfrac{1}{1+e^{-x}}$ la fonction sigmoid. Ce qui est apprit ici sont les paramètres du modèle r.

2. Que se passe-t-il si on minimise cette fonction de perte ?
3. Expliquer en quoi avoir entraîné ce modèle de récompense qui imite la préférence humaine d'une réponse permet d'améliorer le modèle de langage pour discuter avec un humain ?
4. En discutant de cette approche avec un data scientist, il vous dit qu'il vaudrait mieux réussir à éviter d'entraîner un modèle supplémentaire. Selon vous, pourquoi ?
5. En quelques mots, quels sont les biais que vous identifiez dans ce processus ?

Aides



(a) Fonction $x \mapsto \ln(x)$



(b) Fonction $x \mapsto \sigma(x)$