Projet Master 2 IEF 2023/2024 Data Science

Enseignant : Sylvain Benoit (sylvain.benoit@dauphine.psl.eu) – Groupe 1
Arthur Thomas (arthur.thomas@dauphine.psl.eu) – Groupe 2

Date limite du projet : 31 mai 2024 **Nombre d'étudiants par groupe :** 2 étudiants max

Nom du notebook commenté à envoyer : Nom1(Nom2)_NomBaseDeDonnees.ipynb ; veuillez utiliser Github pour itérer sur votre projet.

Base de données : Données provenant des *challenges data* du Collège de France (de préférence), de *Kaggle* ou d'une autre source à préciser. Si les bases sont trop volumineuses pour vos ordinateurs, vous pouvez les réduire en respectant les proportions des variables d'intérêts (la variable cible surtout).

Consigne : Répondre d'une manière argumentée et détaillée à la problématique énoncée dans le challenge que vous avez sélectionné. Ce dernier est soit un problème de classification ou de régression.

1. Expliquez brièvement la problématique du challenge et comment vous allez y répondre.
2. Assurez-vous que votre base de données et bien décomposée en un échantillon de *train* (d'entrainement) et de *test*. Par ailleurs, vous pouvez ensuite vous-même décomposer la base de *train* en une base pour l'estimation et une autre pour la validation si cela vous semble pertinent.
3. Construisez les *features* (les variables explicatives) que vous allez utiliser pour expliquer la variable cible (variable dépendante). N'hésitez pas à les décrire par quelques statistiques et des mots. *Conseil : Ne passez pas vos nuits à construire des features potentiels.*
4. Construisez un modèle simple comme référence en sélectionnant la métrique d'évaluation pertinente (sur les données du Collège de France, cette métrique vous est imposée). Dans le **cadre d'une régression**, vous pouvez faire une régression par moindres carrés ordinaires, une régression pénalisée... Dans le **cadre d'une classification**, une régression logistique, un arbre de décision...
5. Construisez **au moins un modèle non supervisé** (clustering) pour répondre à la problématique puis optimisez ce réseau (recherche des paramètres par *grid search*, réglage fin des hyperparamètres par validation croisée, vérification de l'absence de sur-apprentissage...). Comparez ses performances avec votre modèle de référence.
6. Construisez **au moins un modèle supervisé** (SVM, méthode ensembliste) pour répondre à la problématique puis optimisez ce réseau (recherche des paramètres par *grid search*, réglage fin des hyperparamètres par validation croisée, vérification de l'absence de sur-apprentissage...). Comparez ses performances avec les précédents modèles.
7. Interprétez votre modèle avec les outils adéquats (variables importance, LIME, SHAP...)
8. **[Optionnel]** Construisez **un modèle de *deep learning*** (réseau de neurones dense, convolutif ou LSTM ; un transformers ; un GAN) pour répondre à la problématique puis optimisez ce réseau (réglage fin des hyper-paramètres, vérification de l'absence de sur-apprentissage...)[1].
9. Comparez les performances entre vos modèles et interprétez vos résultats.

Critères de notation :

1. Qualité de la rédaction du notebook ; Maîtrise de Python ; Esprit d'analyse.
2. Appropriation des connaissances vues en cours et leurs mises en pratique.
3. Interprétation des résultats.

[1] **Pour ceux qui ne veulent pas faire de réseau de neurones, faire deux méthodes ensemblistes**.

Version du 03/02/2024.