Cours / TP **Apprentissage Machine**
février 2023 **Durée: 5 séances**

Objectif: **Appentissage supervisé pour des données avec classes déséquilibrées.**

- Le projet de cette UE doit être réalisé en binôme, en R ou en Python. Merci de bien préciser les nom et prénom de chaque membre du binôme et de mettre en copie votre binôme lors de l'envoi des 4 scripts commentés (1 script par séance) et du rapport (dernière séance).
- Les résultats de vos analyses doivent être expliqués et reprendre les différents notions vues en cours de Data Science et d'Apprentissage Machine.
- Vous devez me faire parvenir vos scripts pour le 03/04/2023 à 23h59 (via un dossier drive que je vous indiquerai).

---

1. **Séance 1: Analyse exploratoire des données déséquilibrées**

   Pour cette UE, vous analyserez les données suivantes:

   - Credit fraud
   - Bank marketing
   - Employee attrition

   (a) Pour les trois jeux de données, faites un analyse exploratoire des attributs:

   - quelles sont les dimensions du jeu de données, existe-t'il des valeurs manquantes ou des attributs constants?
   - affichez à l'aide un graphe adapté la proportion d'individus qui ont *churné*
   - pour chaque variable catégorielle, affichez à l'aide un graphe adapté la proportion de *churn* vs. *non churn*
   - pour chaque variable numérique, affichez séparemment à l'aide un graphe adapté (eg. histogramme) les valeurs pour les populations *churn* & *non churn*
   - affichez la matrice de corrélation des attributs

   (a) Que pouvez-vous déjà conclure? Y-a-t'il des attributs qui semblent être fortement liés au comportement de *churn*?

   ⇒ **A la fin de cette séance, vous devez avoir une bonne connaissance de chacun des jeux de données et quelques pistes concernant les variables les plus influentes.**

2. **Séance 2: Prédiction de *churn*, Partie I**

   Pur cette séance, nous utiliserons les approches supervisées ci-dessous pour la prédiction de *churn*:

   - Arbre de décision
   - Régression Logistique
   - Support Vector Machine (sans kernel)
   - Support Vector Machine (avec kernel)

⇒ **Les approches Arbre de Décision et Régression Logistique ont été vues en cours de Data Science 1. Nous verrons en détail pendant cette séance l'approche SVM.**

(a) Pour chaque approche, avec les hyperparamètres par défaut, évaluez la prédiction du *churn* sur la base de l'AUC (Area Under the Curve). Résumez clairement vos évaluations.

(b) Pour chaque approche, définissez un modèle performant en recherchant de bons hyper-paramètres via un *grid search*. Comparez les *meilleurs* modèles pour chaque approche sur la base de l'AUC (Area Under the Curve). Résumez clairement vos évaluations.

Vous pourrez trouver ci-dessous des indications et exemples pour le *grid search* en R et Python:

- R Grid Search
  - Random Hyperparameter Search
  - Grid search in the tidyverse
- Python Grid Search
  - DecisionTree Classifier — Working on Moons Dataset using GridSearchCV to find best hyperparameters
  - Hyperparameter Optimization With Random Search and Grid Search
  - Grid Search with Logistic Regression

⇒ **A la fin de cette séance, vous aurez déterminer la meilleure approche pour chaque jeu de données, avec les meilleurs hyperparamètres.**

3. **Séance 3: Prédiction de *churn*, Partie II**

A partir du travail de la Séance 2, étudiez vos données à l'aide de deux nouveaux algorithmes d'apprentissage supervisé:

- Random Forest
- XGBoost

⇒ **Nous réserverons une partie de cette séance à l'étude des approches Random Forest et XGBoost.**

En vous appuyant sur les tutoriels ci-dessous , définissez les variables les plus importantes pour chacune des approches d'apprentissage supervisé vue au cours des Séances 2 et 3:

- *'How to Calculate Feature Importance With Python'*
- *'Scikit-learn course; Feature Importance'*

⇒ **A la fin de cette séance, vous aurez déterminer la meilleure approches pour chaque jeu de données, avec les meilleurs hyperparamètres et les features les plus influents.**

4. **Séance 4: Sur-échantillonage et sous-échantillonage**

   Afin de réduire le problème du déséquilibre des classes, nous pouvons par exemple utiliser une approche d'*oversampling*, qui consiste à augmenter le nombre d'instances de la classe minoritaire.

   Familiarisez-vous avec cette approche, et également avec d'autres approches d'*oversampling* disponibles aux liens ci-dessous. Pour remédier au déséquilibre, il est également possible de faire de l'*undersampling*, c'est-à-dire de réduire le nombre d'instances de la classe majoritaire.

   - Resampling strategies for imbalanced datasets
   - Undersampling and oversampling imbalanced data
   - Techniques to deal with imbalanced data

   (a) Pour chaque approche, avec les hyperparamètres par défaut, évaluez la prédiction du *churn* sur la base de l'AUC (Area Under the Curve). Les pré-traitements qui seront appliqués aux données sont une approche d'*oversampling* (SMOTE ou ADASYN) et une approche d'*undersampling* (Random Undersampling ou Tomek Links). Résumz clairement vos évaluations.

   (b) Pour chaque approche, définissez un modèle performant en recherchant de bons hyperparamètres via un *gird search*. Comparer les *meilleurs* modèles pour chaque approche sur la base de l'AUC (Area Under the Curve). Les pré-traitements qui seront appliqués aux données sont une approche d'*oversampling* et une approche d'*undersampling*. Résumez clairement vos évaluations.

   ⇒ **A la fin de cette séance, vous aurez évaluer l'impact du resampling sur les performances des différentes approches pour les 3 jeux de données considérés.**

5. **Séance 5: QCM, finalisation et rendu**

   Un QCM de 30 min (individuel!) aura lieu au début de ce cours. Le reste de la séance sera dédiée à la finalisation de vos scripts et de votre rapport. Attention à bien comparer et résumer les évaluations de vos modèles, selon les stratégies et les jeux de données.