# Implémentez un modèle de scoring

## Note méthodologique

### Méthodologie d'entraînement du modèle

Comme proposé par notre manager, le travail se base sur le notebook Kaggle d’Olivier G disponible à l’adresse suivante (<https://www.kaggle.com/code/ogrellier/lighgbm-with-selected-features/log>). Après exécution de celui-ci (hormis la partie modélisation en LGBM), on obtient donc un dataset complet (tous les sous-jeux de données ont été merge), et prêt à être nettoyé. On fait un traitement des valeurs infinies (pas des NaN) avant de séparer le dataset en 3 sous-set : train, validation, et test (ce dernier ayant déjà été distingué à la source). On va donc entrainer et valider nos modèles sur le duo train/validation.

Etant donné le nombre de valeurs manquantes très important dans le daataset (%), on implémente une méthode d’imputation SimpleImputer qui va remplacer chaque NaN par la médiane de la variable à imputer. En effet certaines variables sont très peu renseignées et/ou présentent un maximum et un écart-type très importants. Nous préférons donc utiliser la médiane que la moyenne. En outre la présence de ces outliers nous invite à choisir un RobustScaler pour standardiser les séries car celui-ci y est peu sensible, contrairement au StandardScaler qui va utiliser la moyenne et l’écart-type, ou le MinMax scaler qui (comme son nom l’indique) tiendra compte de l’intervalle complet. A noter que ces opérations sont toutes deux intégrées dans un pipeline avec le modèle de classification.

Nous avons retenu 3 modèles de classification. Tout d’abord un RandomForest comme modèle baseline. Nous aurions pu partir d’une régression logistique mais il était évident que le Random Forest performerait déjà mieux que la régression. A noter que c’est le modèle le plus long avec 2329 secondes d’entrainement (près de 40mins). En second et troisième modèles nous avons retenu un XGBoost pour un LGBM. Ces deux derniers sont assez ressemblants (modèles de gradient-boosting) mais sont parmi les modèles de classification (et de régression) les plus performants. Le seul inconvénient des modèles basés sur les arbres de décision (nos trois modèles) est la probabilité d’overfitting. Cependant, nos résultats initiaux (pré-optimisation des hyperparamètres) semblent exclure cette éventualité. A noter que XGB et LGBM sont tous deux bien plus rapides que RandomForest avec des temps d’entrainements respectifs de 814 et 183 secondes. Le dernier modèle : Light Gradient Boosting Machine est le plus performant en terme d’AUC et de temps d’entraînement, et nous retenons donc celui-ci comme modèle principal.