

# Note Méthodologique

Cette note méthodologique revient sur les principales étapes nécessaires à la réalisation du modèle statistique conçu pour l'entreprise Prêt à Dépenser. Le modèle sert à détecter les clients de l'entreprise susceptibles de ne pas rembourser leur crédit. Il renvoie un score qui donne la probabilité pour un client de ne pas rembourser son crédit.

## Informations Générales

**Nom** : Modélisation du Défaut de Paiement Client

**But** : Évaluer la probabilité d'un défaut de paiement pour un client de Prêt à Dépenser

**Contributeur** : Eloi Le Quilleuc

**Date** : 24-11-2021

**Description** : Ce projet Data Science propose un outil d'aide à l'identification des clients susceptibles de ne pas rembourser leur crédit pour l'entreprise Prêt à Dépenser. Ce projet apporte également des outils d'interprétation et de visualisation des résultats.

**Code source** : <https://github.com/EloiLQ/pretadepenser-score>

## Jeu de Données

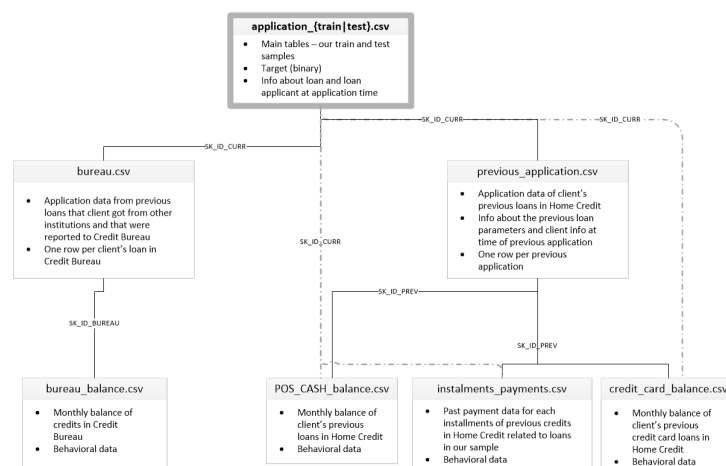
**Chemin** : <https://www.kaggle.com/c/home-credit-default-risk/data>

**Origine** : Home Credit

**Description** : informations bancaires des clients de Home Credit, réparties en sept tables.

**Variable cible** : TARGET

**Description variable cible** : variable binaire indiquant si le client a déjà eu un défaut de paiement ou non



# Préparation des Données

**Contributeur** : Ekrem Bayar

**Chemin** : <https://www.kaggle.com/ekrembayar/homecredit-default-risk-step-by-step-1st-notebook/notebook>

**Description** : cette étape de préparation des données consiste principalement à agréger les sept tables tout en conservant un maximum d'information.

**Filtre des clients** : sélection des clients présent dans le fichier principal application\_{train|test}.csv

**Filtre des Variables** : aucun

**Valeurs manquantes** : prises en compte

**Encodage** : One Hot pour les variables catégorielles

**Feature Engineering** : moyennes de scores, moyennes, min, max sur des quantités temporelles, etc

## Algorithme d'Apprentissage Machine

**Algorithme** : LightGBM

**Home-page** : <https://github.com/microsoft/LightGBM>

**Version** : 3.1.0

**Type** : Arbre de Décision Boosté

**Tâche** : Classification

**Fonction de coût** : régression logistique

**Hyperparamètres** : par défaut / pas d'optimisation

## Données pour la Modélisation

**Taille du jeu d'entraînement** : 285 979 clients

**Taille du jeu de test** : 21 526 clients

**Nombre de variables d'entrée** : 37 variables les plus importantes selon LightGBM

**Variable cible** : TARGET

# Résultats

**Sortie du modèle** : score crédit ou probabilité de défaut de paiement

**Score crédit médian** : 5 %

**Métrique d'évaluation** : aire sous la courbe ROC (ROC AUC)

**ROC AUC** : 0.783

**ROC AUC incertitude** : [0.771 - 0.794] (intervalle de confiance 95 %), i.e  $0.783 \pm 1.5 \%$

## Interprétation

**Algorithme** : Shap

**Home-page** : <http://github.com/slundberg/shap>

**Version** : 0.39.0

**Locale** : coefficients de Shapley (en pourcentage) pour chaque variable et chaque client.

**Globale** : moyenne des coefficients de Shapley sur l'ensemble des clients.

## Limites et Améliorations

**Améliorations** : optimisation des hyperparamètres de LightGBM, équilibrage des deux classes à prédire (50 % - 50 %), utilisation du bagging.

**Limites** : les scores crédit sont relativement difficiles à interpréter, car relativement faibles (< 50 %) même pour les clients qui ne remboursent pas leur crédit (non payeurs). Ceci est dû à la difficulté de séparer les clients payeurs et non payeurs avec notre modèle statistique et à la forte proportion dans la population de clients payeurs (90 %).