**Nathalie MAVEL** Parcours Data Scientist

# Note méthodologique

## Projet 7

## Implémentez un modèle de scoring

Mai 2022

# 1. Introduction

Objectif de la mission :

- **Mettre en oeuvre un outil de "scoring credit"** afin de calculer la probabilité qu'un client ne rembourse pas son crédit, puis classifier et notifier la demande par un accord ou un refus.

- **Mettre en œuvre un dashboard interactif** à destination des chargés de clientèle.

Pour mettre en place un modèle efficace, il convient d'effectuer en premier lieu une Analyse Exploratoire des Données. Cela permet la bonne compréhension de l'objectif de la mission, l'identification de variables pertinentes (mesurer les corrélations entre les variables), le nettoyage de valeurs qui présentent des valeurs aberrantes ou encore la mise en place de features engineering.

Cette analyse a été adapté d'une analyse effectuée sur Kaggle disponible à l'adresse ci-dessous :

https://www.kaggle.com/code/willkoehrsen/start-here-a-gentle-introduction/notebook
[date de consultation 30/04/2022]

Plusieurs éléments de cette analyse sont ressortis dont voici quelques exemples :

- **la cible** : [qui indique si une personne a remboursé (0) ou non (1) son crédit] présente un fort déséquilibre qui conviendra de prendre en compte dans la mise en place des modélisations.

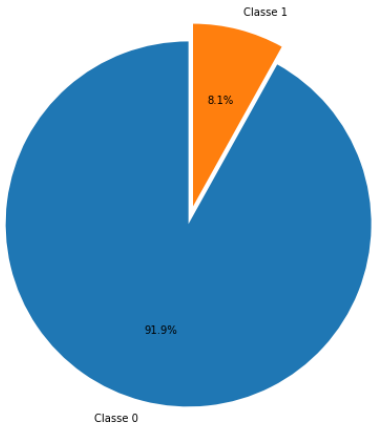



**Fig 1** : *Répartition des clients suivant la variable cible dans le jeu d'entrainement : client ayant remboursé [0] vs clients n'ayant pas remboursé [1]*

Si le modèle est laissé sans ajustement de la cible, il y aura un surajustement qui favorisera surement l'accord de prêt.

- **les transformations des variables catégorielles** : soit avec le label encoder (pour les variables à 2 catégories) et le one hot encoder pour les pour les variables à + de 2 catégories)

- La **conversion de la variable Days_birth.** Les valeurs sont négatives, car elles sont enregistrées en nombre de jour par rapport à la demande de prêt. Une conversion a été effectuée et les âges vont de 20 à 69 ans, il n'apparait pas de données aberrantes
- **Détection d'anomalies pour la variable Days_employment.** Certains clients auraient travaillé plus de 1000 ans
- **Variables d'intérê**t : ext_1, ext_2, ext_3
- **Features engennering** :

Construction de 35 variables polynomiales pour les variables : ext_1, ext_2, ext_3 et DAYS_BIRTH

CREDIT_INCOME_PERCENT : le pourcentage du montant du crédit par rapport au revenu du client.

ANNUITY_INCOME_PERCENT : le pourcentage de l'annuité du prêt par rapport au revenu du client.

CREDIT_TERM : la durée du paiement en mois (l'annuité étant le montant mensuel dû).

DAYS_EMPLOYED_PERCENT : le pourcentage des jours d'emploi par rapport à l'âge du client.

Cette étude a été réalisée sur Python, toutes les librairies utilisées sont indiquées dans les notebooks. Le notebook d'analyse qui comprend tous les détails est joint dans le dossier global.

## 2. Méthode d'entrainement des modèles

Dans cette étude, l'objectif est d'apporter un modèle qui calcule la probabilité de défaut de paiement d'un client en vue d'accepter ou de refuser sa demande de prêt selon un seuil définit.

La catégorie des algorithmes de classification supervisées a été utilisée pour arriver à une classification binaire, 0 accepter : peu de risque de défaut de paiement, 1 : refuser, risque trop important de défaut de paiements.

La problématique métier attire notre attention au fait que l'on souhaite : limiter le nombre de faux positifs. C'est-à-dire limiter le nombre de faux bons clients qui engendreraient des couts à la banque.

### 2.1 Preprocessing :

Avant la mise en place des algorithmes, une étape de pre-processing est indispensable. Cette étape se définit par :

- Imputer les valeurs manquantes
- Transformer les boolens en intégrales,
- Centrer-réduire le jeu de données
- Diviser le jeu de données en 2 : train et test
- Équilibrer les données

L'équilibre du jeu de données est nécessaire pour éviter le sur ou sous entrainement des modèles.

La stratégie d' "undersampling" a été pris en compte. Cette stratégie est motivée par une forte importance du jeu de données (246 000 individus au total dont 19860 pour la catégorie la plus faible) et un temps de calcul qui sera diminué.

Il y a ainsi 19860 individus dans chacune des 2 catégories.

## 2.2 Modélisations

3 modèles de classifications ont été testés : **la régression logistique, le Randon-Forest classifier et le LGBM.**

- **Regression logistique : "**La **régression logistique** est un **modèle** statistique permettant d'étudier les relations entre un ensemble de variables qualitatives Xi et une variable qualitative Y. Il s'agit d'un **modèle** linéaire généralisé utilisant une fonction **logistique** comme fonction de lien"

Pour plus d'information : https://medium.com/tell-ia/la-r%C3%A9gression-logistique-expliqu%C3%A9e-%C3%A0-ma-grand-m%C3%A8re-52a2ab30788

- **Random forest classifier :** cet algorithme amorce les données en choisissant au hasard des sous-échantillons pour chaque itération d'arbres

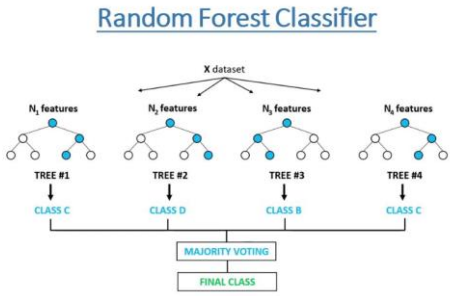



**Fig 2** : *random forest Classifier*

Pour plus d'information : **https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6**

- **LigthGBM (**Light Gradient Boosting Machine) **:** algorithme basé sur des arbres de décision et utilisé pour le classement. Il se base sur la descente de gradient (technique d'optimisation pour réduire la fonction de perte en suivant la pente à travers une taille de pas fixe).

https://towardsdatascience.com/what-makes-lightgbm-lightning-fast-a27cf0d9785e

L'entrainement du modèle a pris en compte une **grid search CV** qui permet de sélectionner les meilleurs hyper-paramètres en testant différents paramètres aléatoirement. Les paramètres ont également été sélectionnés grâce à une recherche par validation croisée et qui est basée sur le score, fbeta score=2. Cette fonctionnalité a été fait grâce à la fonction RandomizedSearchCV (librarie sklearn).

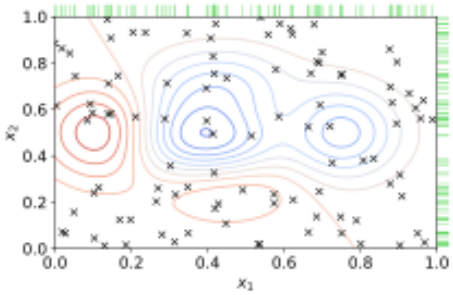



**Fig 3:** *exemple théorique de recherches des meilleurs hypeparamètres aléatoirement*

La cross-validation est une technique qui permet de limiter les biais en sous échantillonnant le jeu de données en différents blocs.

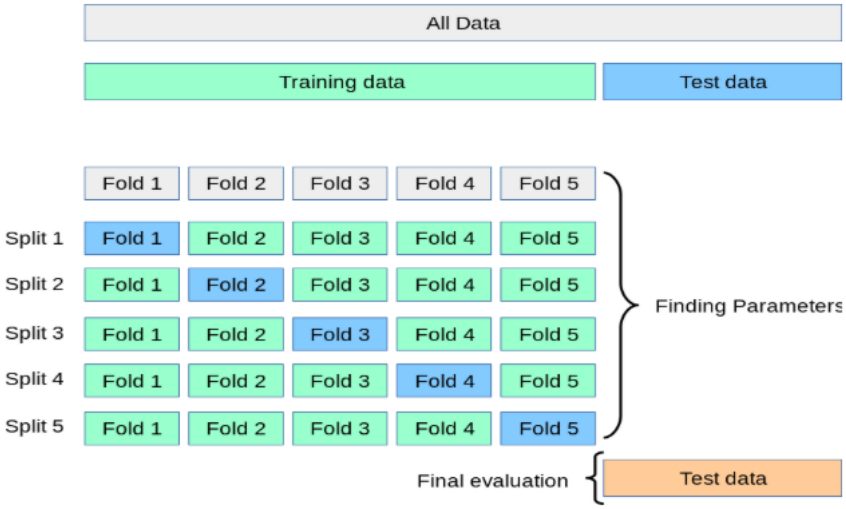



**Fig 4** : *exemple théorique de divisions du jeu de données en plusieurs sous-ensemble.*

Une fois les meilleurs paramètres sélectionnés, la **fonction .predict proba** a été utilisée pour entrainer le modèle et ainsi obtenir une valeur de probabilité de défaut de paiement.

Enfin, l'étape suivante a consisté à **définir un seuil** qui permet de classer les probabilités en 2 catégories (l'individu va rembourser, ou non).

La définition du seuil a été définit grâce à la sélection du score beta F2 maximum. En effet, plus le score F2 est fort, plus les faux négatifs (un faux bon client) sont faibles.

On va regarder pour quel score compris entre 0 et 1 le fbeta score est le plus élevé. Dans le graph si dessous, le seuil de 0.5 apparait comme le meilleur score correspondant au fbeta score le plus élevé.

Pour plus d'informations : https://www.yourdatateacher.com/2021/06/14/are-you-still-using-0-5-as-a-threshold/

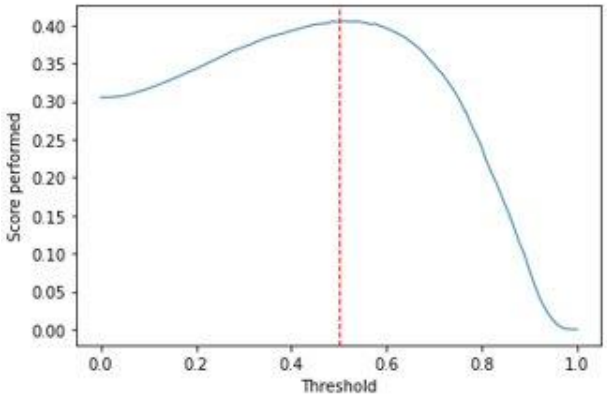


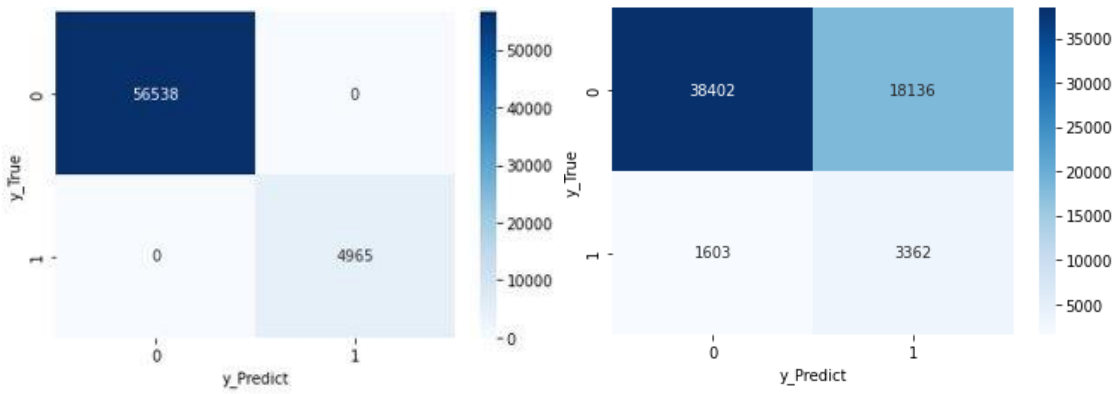

**Fig 5** : *Score en fonction de seuil allant de 0 à 1*

# 3. Evaluation des modèles

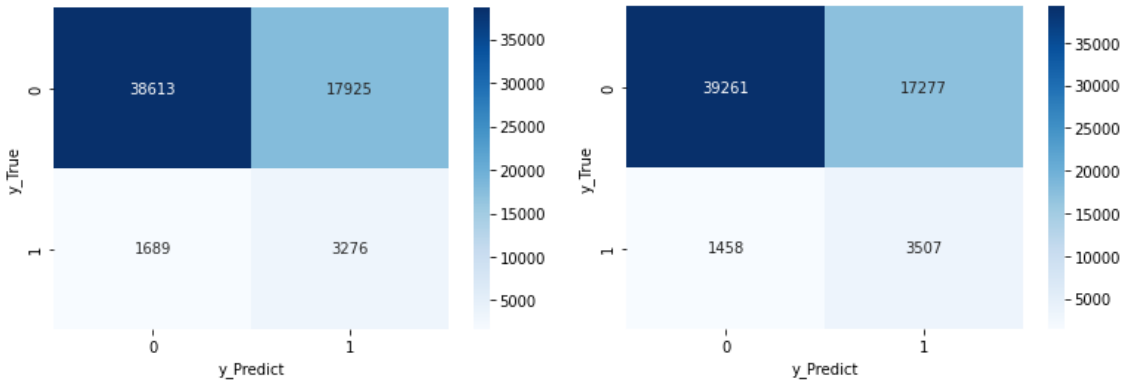
## 3.1 Matrice de confusion

La matrice de confusion permet de résumer les performances d'un modèle et voir les erreurs de prédiction commises.

Ci-dessous, les matrices de confusion pour les différents modèles testés.




Brut Régression logistique




Random forest LGBM

**Fig 6** : *Matrice de confusions sur la partie test du fichier applicataion_train.csv.*

La matrice de confusion pour l'algorithme LGBM indiquent le nombre de faux positifs et négatifs les moins élevés.

## 3.2 Les scores

Pour mémoire voici quelques définitions :

- Précision : Le score de précision détermine que quand le classifieur déclare une prédiction à 1, il a raison à X %
- Rappel : Le score de rappel lui détermine le pourcentage de détection des 1 du classifieur

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

- F1 score, est la moyenne de la précision et du rappel

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{TP}{TP + 0.5\,(FP + FN)}$$

- F2 score : Le F2 score permet de définir combien le rappel est important par rapport à la précision.

$$F_2 = \frac{TP}{TP + 0.2FP + 0.8FN}$$

F2 (Image de l'auteur)

- Fbeta: Enfin, la mesure Fbeta, permet de définir combien le rappel est important par rapport à la précision.

$$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{Recall} + \frac{1}{Precision}}$$

Pour plus d'informations, veuillez consulter cet article qui détaille toutes ces notions. https://towardsdatascience.com/is-f1-the-appropriate-criterion-to-use-what-about-f2-f3-f-beta-4bd8ef17e285

Le **F2 score a été la métrique d'évaluation principale**. En effet, plus le score F2 est fort, plus les faux négatifs (un faux bon client) sont faibles. Ci-dessous, voici le récapitulatif des différents scores pour la comparaison des 3 modèles.

| | accuracy | precision | recall | f1_score | f2_score | f3_score |
|---|---|---|---|---|---|---|
| LR | 0.663 | 0.153 | 0.698 | 0.250 | 0.407 | 0.514 |
| RF | 0.657 | 0.148 | 0.685 | 0.244 | 0.397 | 0.503 |
| LGBM | 0.660 | 0.158 | 0.738 | 0.260 | 0.425 | 0.539 |

**Fig 7** : *Résultats des métriques pour les 3 modèles testés*

Au vu des résultats, le LGBM a été sélectionné pour la mission. Il présente le F2-score le plus élevé. Globalement, les 3 algorithmes indiquent des résultats similaires.

Courbe ROC : La **courbe ROC** (Receiver Operating Characteristic) représente la sensibilité en fonction de 1 – spécificité

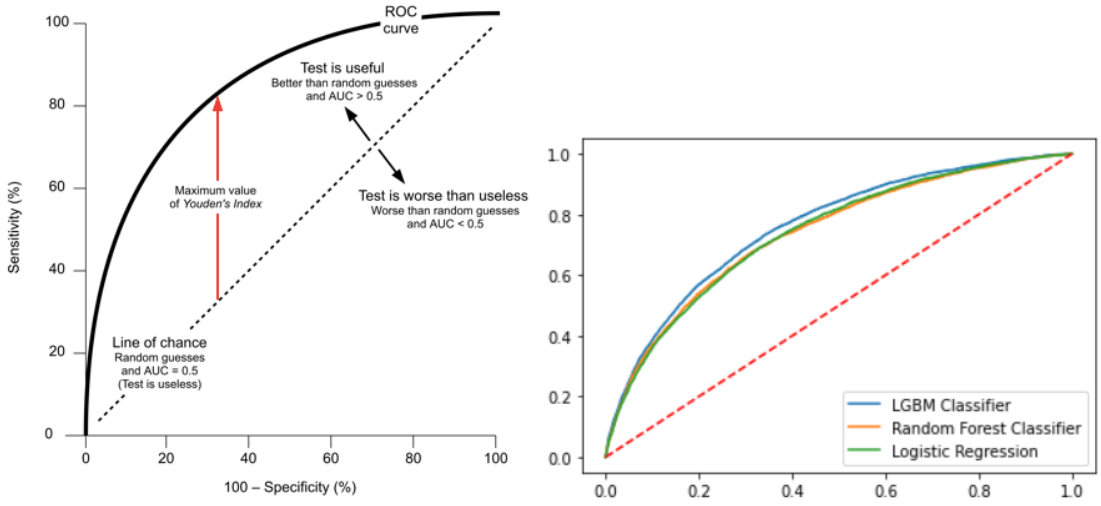



**Fig 8** : *courbe ROC théorique (à gauche) et courbe ROC appliquée aux modélisations (à droite)*

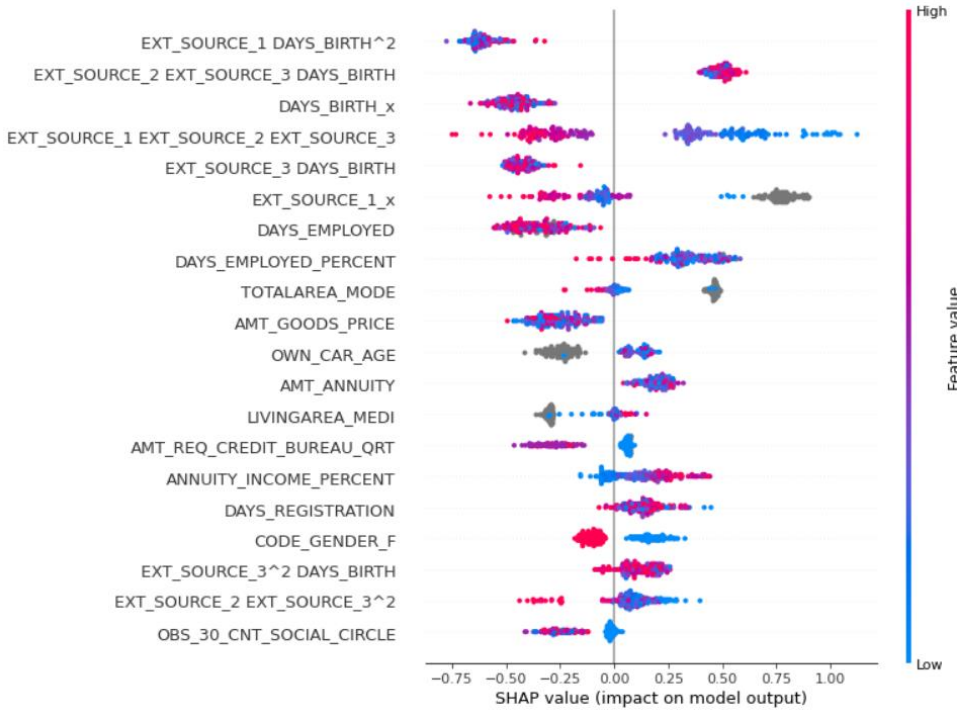
## 4. Interprétabilité du modèle

L'interprétabilité du modèle a été observé avec la librarie SHAP (**SHapley Additive exPlanations) .**

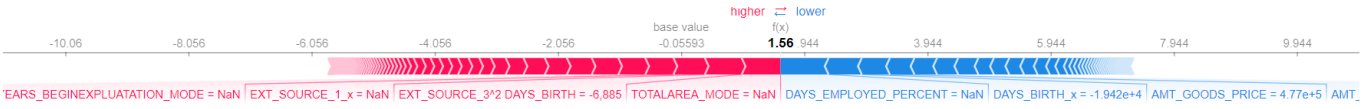
"Il relie l'allocation optimale des crédits aux explications locales en utilisant les valeurs classiques de Shapley de la théorie des jeux et leurs extensions associées (voir les articles pour les détails et les citations)". https://shap.readthedocs.io/en/latest/index.html [Date de consultation 28/04/2022].


**Fig 9** : *Exemple théorique de SHAP*

Interprétation globale du modèle :




Interprétation individuelle :



## 5. Limites et améliorations possibles

Plusieurs éléments seraient à approfondir ou à améliorer :

- **Accorder plus de temps à l'analyse exploratoire**.
- **Sélection des features à confirmer par des équipes métiers**, ce qui permettrai d'améliorer également l'interprétabilité des modèles.
- **Détailler encore plus les hyperparamètres** des modèles.
- **Analyser les faux positifs et faux négatifs.** Ont –ils des points communs ?
- **Classification multiple**. Par exemple, il est possible d'imaginer plusieurs classes et non 2 classes


- **Limites éthique et juridique** : Le data scientist a un rôle de sensibilisation auprès des équipes techniques. Par exemple, l'algorithme retenu ne doit pas créer de discrimination. Au vu du temps qui m'a été accordé pour mettre en place ce projet, je n'ai pas pu analyser d'autres biais de représentativité qui pourrait être présent dans ce jeu de données. A noter qu'en 2015, Amazon a stoppé ses algorithmes de recrutement de personnel après 1 an d'utilisation car ils se sont aperçus que les données d'entrainement n'étaient pas bien équilibrées et favoriser les recrutements masculins.