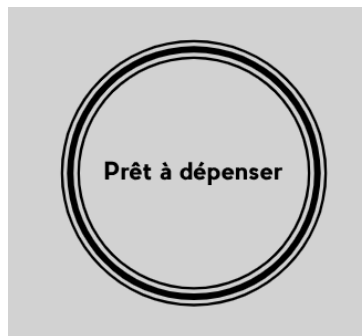


# Note méthodologique



## Sommaire

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Classification</b>	<b>1</b>
2.1	Jeu de données . . . . .	1
2.2	”Feature engeneering” . . . . .	1
2.3	Déséquilibre des valeurs cible . . . . .	2
2.4	Modèles . . . . .	2
<b>3</b>	<b>Évaluation, coût métier et optimisation</b>	<b>3</b>
3.1	Courbe ROC et aire sous la courbe (AUC) . . . . .	3
3.2	Coût métier . . . . .	3
3.3	Optimisation . . . . .	3
<b>4</b>	<b>Interprétabilité</b>	<b>4</b>
4.1	Globale . . . . .	4
4.2	Locale . . . . .	5
<b>5</b>	<b>Limites et améliorations</b>	<b>6</b>

# 1 Introduction

L'entreprise Prêt à dépenser souhaite utiliser un outil de "scoring" afin de calculer la probabilité qu'un client fasse ou non défaut lors du remboursement de son crédit. Pour cela nous devons entraîner un modèle de classification sur des données variées (comportementales, autres institutions financières, etc).

## 2 Classification

### 2.1 Jeu de données

Afin de mieux comprendre les données nous avons procédé à une analyse exploratoire des données sur le jeu application\_train.csv. Pour l'entraînement du modèle nous utiliserons uniquement ces données car l'utilisation des fichiers supplémentaires demande beaucoup de ressources tant en temps d'analyse et d'exploration des données qu'en capacités de calculs. Nous utiliserons le fichier application\_test.csv pour le dashboard.

### 2.2 "Feature engineering"

Calcul de variables polynomiales à partir des meilleurs variables (EXT\_SOURCES). Il n'y a pas d'amélioration significative des résultats (Fig. 1)

Scores pour les différents modèles

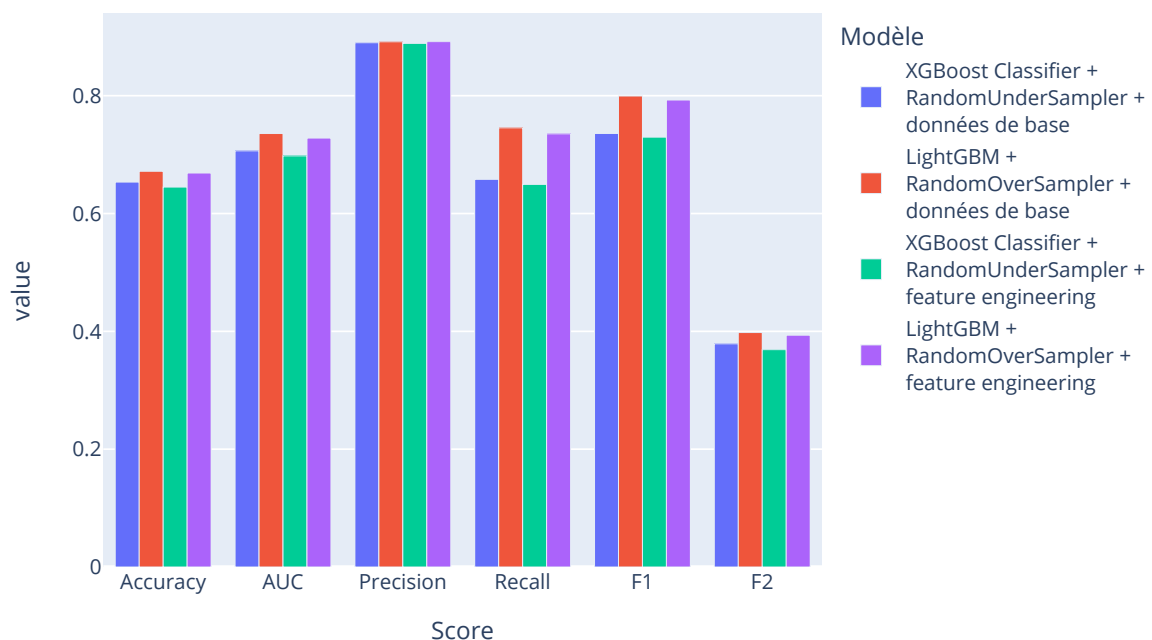


FIG. 1 : Comparaisons des différents scores avec (\_featEng) et sans (\_base) variables polynomiales

## 2.3 Déséquilibre des valeurs cible

Du fait d'un déséquilibre dans les valeurs cible il est difficile pour le modèle de classer efficacement les clients (Fig. 2).

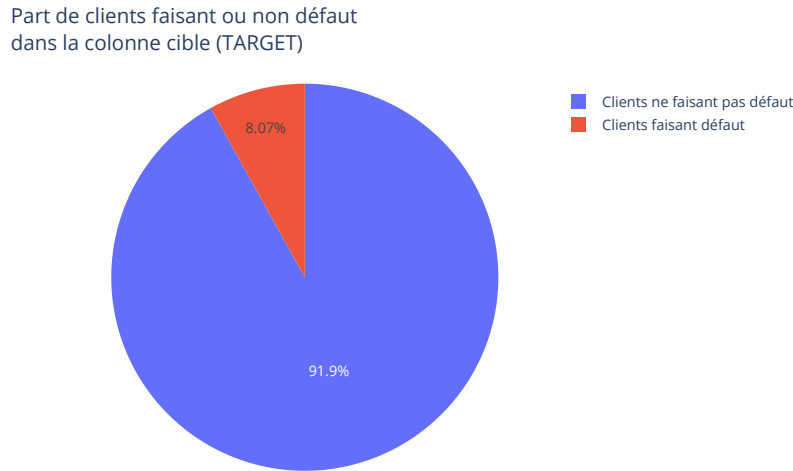


FIG. 2 : Diagramme circulaire illustrant le déséquilibre des types de clients dans la colonne cible

Lors de l'optimisation du score AUC le modèle va préférer classer tous les clients comme ne faisant pas défaut car cela améliorera son score. Nous avons donc dû rééquilibrer la part des valeurs cible grâce à la librairie imblearn. Cette librairie propose des outils de sur- et de sous-échantillonnage.

Le sur-échantillonnage (over-sampling) permet d'augmenter le nombre de cibles faisant défaut. Soit en dédoublant aléatoirement des cibles faisant défaut (random over sampling). Soit en créant des données synthétiques (SMOTE) à partir de cibles voisines (KNN)

Le sous-échantillonnage (under-sampling) permet de diminuer le nombre de cibles ne faisant pas défaut. Soit en échantillonnant aléatoirement un nombre de cible ne faisant pas défaut égale au nombre de cible faisant défaut (random under sampling) Soit en créant des données (Tomek links) à partir de groupe de cibles ne faisant pas défaut (KNN)

Un mélange de sur- et de sous-échantillonnage avec SMOTEENN et SMOTETomek qui combinent les effets de SMOTE avec une réduction à partir des plus proches voisins

## 2.4 Modèles

Nous avons utilisé deux modèles XGBoost et LightGBM qui ont l'avantage d'être résistants aux valeurs manquantes. Ces deux modèles reposent sur le boosting de gradient qui consiste en une agrégation d'arbres de décisions simples afin d'obtenir un meilleur résultat

### 3 Évaluation, coût métier et optimisation

#### 3.1 Courbe ROC et aire sous la courbe (AUC)

La courbe ROC représente les vrais positifs en fonction des faux positifs (Fig. 3). Plus la courbe est proche du coin supérieur gauche meilleur est le modèle. L'aire sous la courbe (AUC) nous donne une valeur numérique pour comparer ces modèles.

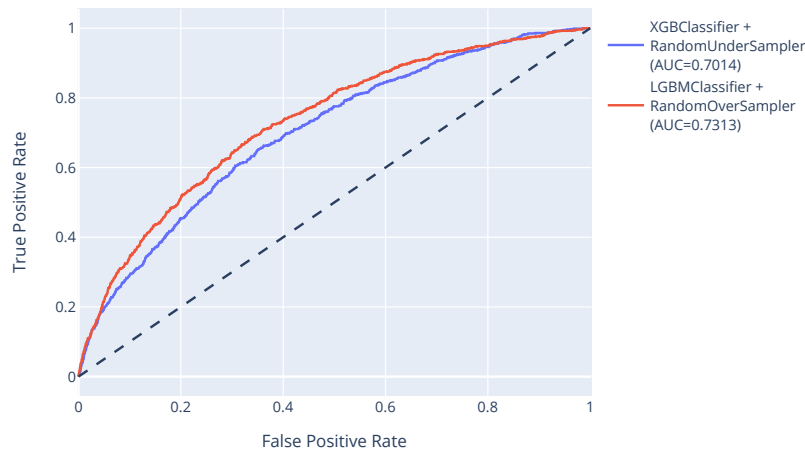


FIG. 3 : Courbes ROC pour les deux modèles dont nous avons testé les hyperparamètres

#### 3.2 Coût métier

Afin d'utiliser une méthode d'évaluation qui soit mathématiquement fondée nous nous sommes basé sur une métrique déjà existante du F-score. Ce score calcule la moyenne harmonique de la précision et du rappel (Fig. 4). L'utilisation du  $F_\beta$ -score permet d'ajouter du poids respectivement au rappel lorsque le facteur  $\beta$  est  $>1$  ou à la précision lorsque le facteur  $\beta$  est  $<1$ . Nous avons utilisé une valeur de  $\beta=2$ . Cette métrique permet de réduire les faux négatifs qui sont très coûteux pour la société. Tandis qu'il est moins coûteux d'avoir des faux positifs car ce seront des clients à qui l'ont ne prêtera pas même si ils pourraient potentiellement rembourser leur crédit. Tant que le nombre de client est important ce n'est pas un soucis car il y aura toujours d'autres clients à qui prêter.

#### 3.3 Optimisation

L'optimisation a été effectuée par GridSearch. Pour chaque solution de rééquilibrage nous avons fait une GridSearch pour les deux modèles. Pour la classification avec XGBoost nous avons testé trois solutions avec un nombre de boosting de 100, 500 ou 1000. Pour LightGBM nous avons testé différents algorithmes : GBDT, DART, RF et GOSS et différents nombre de boosting : 100, 500 et 1000. Nous avons retenu comme meilleur modèle LightGBM avec un sur-échantillonnage aléatoire (random over sampling). Les paramètres retenus sont l'algorithme DART et un nombre de boosting de 100

		Prédit		
		0	1	
Réal	0	TN	FP	Precision
	1	FN	TP	
		Recall		

FIG. 4 : Matrice de confusion rappelant ce que sont la précision et le rappel

## 4 Interprétabilité

Dans un souci de transparence nous souhaitons pouvoir expliquer comment fonctionnent nos modèles

### 4.1 Globale

L'étude des principales variables utilisées par le modèle permet de mieux comprendre son fonctionnement global. Nous pouvons observer sur cette figure les différentes variables et leur importance dans les calculs des modèles (Fig. 5). Cela nous renseigne sur les données qui sont les plus prises en compte lors de la classification des clients.

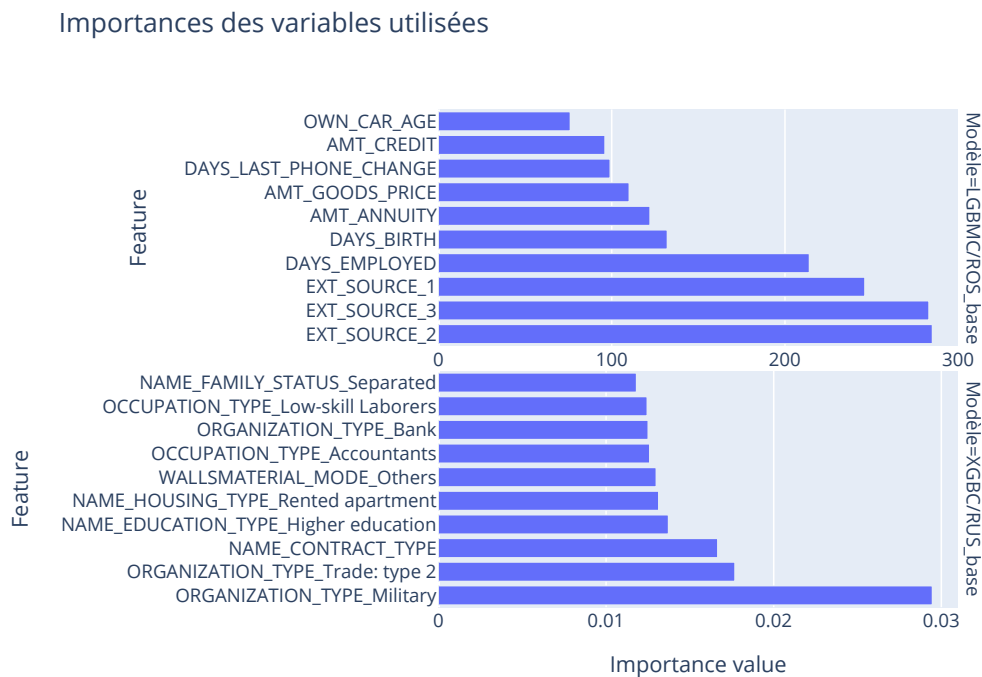


FIG. 5 : Principales variables selon leur importance dans l'entraînement des modèles

Nous avons utilisé la librairie SHAP afin d'expliquer le modèle retenu (LightGBM). Celle-ci nous permet d'observer la part des différentes variables dans la classification dans l'une ou l'autre des classes (Fig. 6).

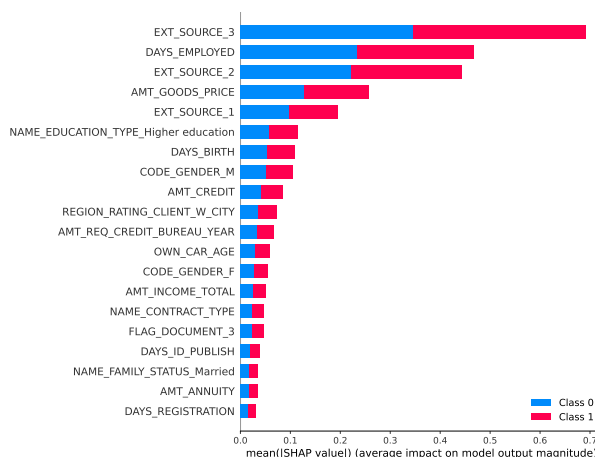


FIG. 6 : Part des principales variables dans la classification

## 4.2 Locale

La librairie SHAP nous permet d'en apprendre un peu plus sur la part des variables dans le classement d'un client en particulier. Nous pouvons observer sur ces figure la part de chaque variables dans la prédiction de la classification d'un client (Fig. 7).

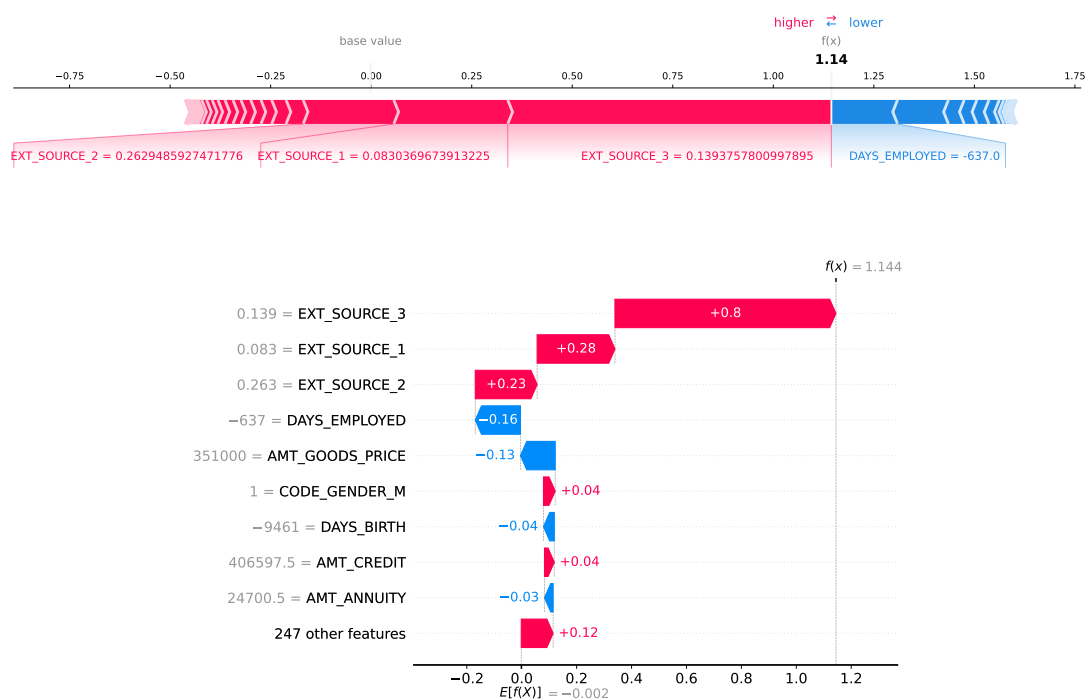


FIG. 7 : Part des variables dans la classification d'un client

## 5 Limites et améliorations

Il peut être intéressant d'intégrer quelques données concernant les autres fichiers. Cela demandera un travail plus important au niveau du nettoyage et de la sélection des données mais peut nous apporter des précisions concernant les clients. Nous pourrions essayer d'optimiser plus d'hyperparamètres dans les différents modèles afin d'améliorer encore les résultats. Il convient de rester prudent quand à l'utilisation de ces prédictions qui sont issus d'un modèle et ne reflètent pas forcément la réalité. Celles-ci doivent rester un outil d'aide quant à la prise de décision finale.

### Liens

Github : <https://github.com/l4nc3l0t/P7>

Dashboard (Heroku) : <https://bank-scoring-dash.herokuapp.com/>