Modèle de prédiction du risque de crédit

Note Méthodologique

Ce document présente les étapes de modélisation du projet « *Implémentez un modèle de scoring* ». Il décrit principalement le choix du type de modèle, son entraînement & optimisation, ainsi que sa fonction coût et sa fonction d'évaluation.

1. Contexte

Le projet « *Implémentez un modèle de scoring* » s'inscrit dans le cadre de la formation diplômante Data Scientist d'OpenClassrooms. Le but est d'évaluer la réalisation d'un projet de data-science, de la préparation des données à la restitution des résultats en passant par une modélisation du problème à traiter.

L'objectif du projet est de concevoir un modèle de *machine learning* qui prédit une probabilité de défaut de paiement pour les clients d'un organisme de crédit à la consommation.

À cela s'ajoutent deux objectifs qui valident des compétences pratiques en « ML/OPs » :

1. Créer une API qui permet d'utiliser les résultats du modèle dans n'importe quel contexte.
2. Concevoir un tableau de bord accessible en ligne pour restituer les résultats de la prédiction d'une manière compréhensible pour une personne non-experte en data-science.

HomeCredit est un organisme de crédit à la consommation. L'entreprise possède une base de données qui regroupe des informations sur plus de 300.000 crédits accordés à des clients ayant peu ou pas d'historique de prêt auprès d'institutions bancaires classiques.

Le but est d'obtenir une modélisation du risque de défaut de paiement à partir de cet historique. Cela sera utile à HomeCredit pour minimiser les pertes liées à son risque de crédit, mais également aux employés des agences de prêt pour expliquer au client l'octroi ou non d'un crédit de manière transparente et objective.

2. Prétraitement & choix des variables

2.1. Les données disponibles

Les données sont disponibles à l'adresse : https://www.kaggle.com/c/home-credit-default-risk/data

Le jeu de données comporte 10 tables ; la table principale {`application`} contient les étiquettes de la variable cible (à savoir 0 lorsque la demande de prêt n'a fait l'objet d'aucun défaut ni d'aucun retard de paiement et 1 sinon).

Les autres tables contiennent des informations sur les autres prêts connus pour un même client, soit dans des établissements tiers (`bureau`, `bureau_balance`) soit auprès de HomeCredit (`previous_applications`, `POS_CASH`, `credit_card_balance`, etc.)

```
Entrée [2]: # Liste de toutes les tables présentes dans le jeu de données
            ! ls -lh ../02_data/

total 2,5G
-rw-rw-r-- 1 adrien adrien  26M juin  26  2018 application_test.csv
-rw-rw-r-- 1 adrien adrien 159M juin  26  2018 application_train.csv
-rw-rw-r-- 1 adrien adrien 359M juin  26  2018 bureau_balance.csv
-rw-rw-r-- 1 adrien adrien 163M juin  26  2018 bureau.csv
-rw-rw-r-- 1 adrien adrien 405M juin  26  2018 credit_card_balance.csv
-rw-rw-r-- 1 adrien adrien  37K juin  26  2018 HomeCredit_columns_description.csv
-rw-rw-r-- 1 adrien adrien 690M juin  26  2018 installments_payments.csv
-rw-rw-r-- 1 adrien adrien 375M juin  26  2018 POS_CASH_balance.csv
-rw-rw-r-- 1 adrien adrien 387M juin  26  2018 previous_application.csv
-rw-rw-r-- 1 adrien adrien 524K juin  26  2018 sample_submission.csv
```

Figure 1 : Capture d'écran montrant la liste des tables qui composent le jeu de données original

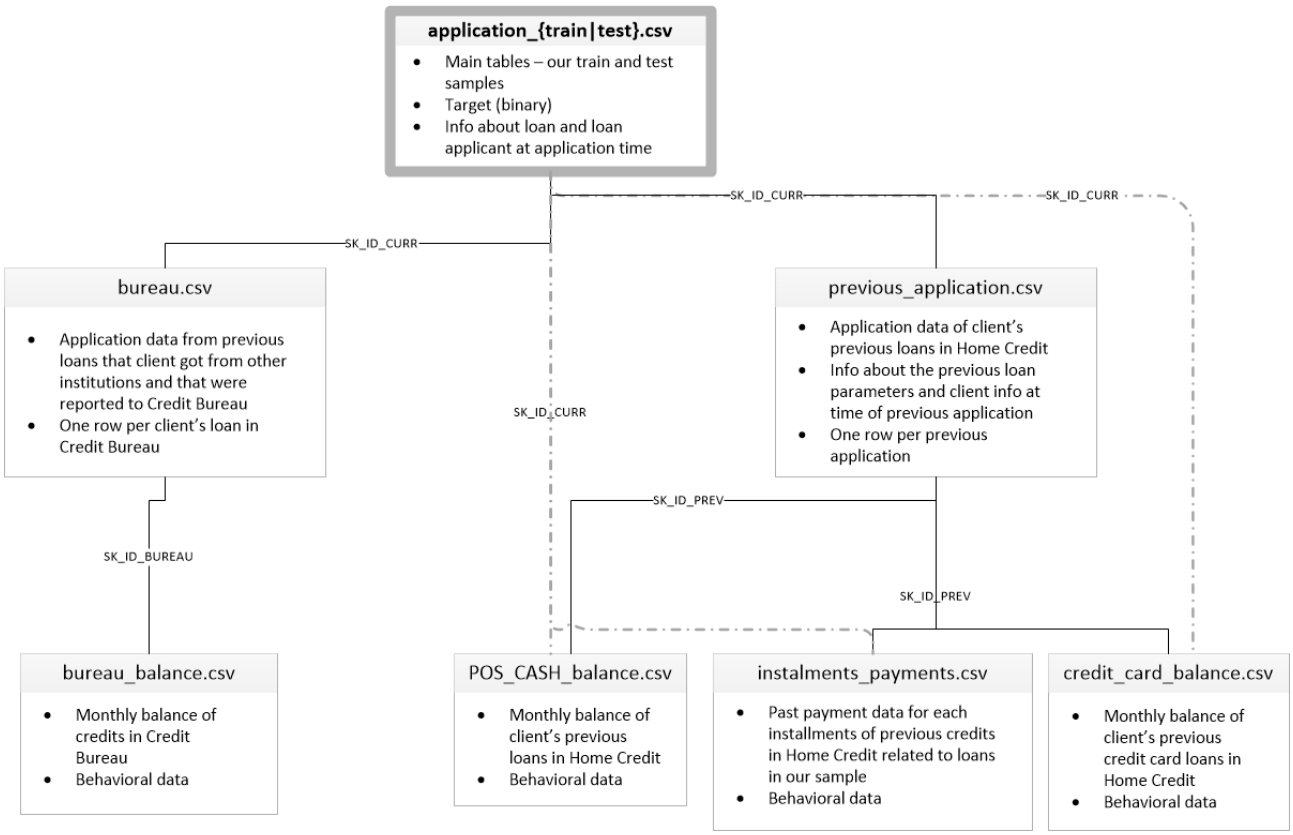


Figure 2 : Liens schématiques entre les tables du jeu de données HomeCredit

2.2. Le choix des variables

Les étapes de prétraitement et de feature engineering ne sont pas les principales compétences évaluées dans le projet. C'est pourquoi, dans un soucis de simplicité, il a été conseillé de s'inspirer de travaux déjà existants, disponibles sur la plateforme Kaggle (*kernels Kaggle*). Voici la liste exhaustive des kernels Kaggle consultés lors de la réalisation du projet :

https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction
https://www.kaggle.com/willkoehrsen/introduction-to-manual-feature-engineering
https://www.kaggle.com/willkoehrsen/intro-to-model-tuning-grid-and-random-search
https://www.kaggle.com/shanth84/home-credit-bureau-data-feature-engineering
https://www.kaggle.com/mlisovyi/lightgbm-hyperparameter-optimisation-lb-0-761
https://www.kaggle.com/ogrellier/good-fun-with-ligthgbm/script

Le choix de l'algorithme de machine learning s'est rapidement porté sur un modèle non linéaire, aussi il n'y a pas eu d'étapes de sélection ou d'élimination de variables, celles-ci n'ayant pas beaucoup d'impact sur les performance de ce genre de modèle. L'ensemble des variables de la tables principale a été sélectionné ; certaines variables de la table `bureau` ont été ajoutées pour améliorer les performances du modèle (cf kernel kaggle 2).

2.3. Prétraitement des données

Pour la table principale il a fallu réaliser les prétraitements suivants :

- Imputer les données manquantes de la manière la plus adaptée pour chaque variable : les variables qui décrivent des tendances centrales ont été imputées selon la tendance centrale utilisée dans la variable[a]. Les autres variables numériques sont imputées par leur médiane pour diminuer l'impact des valeurs extrêmes.
- Encoder les données textuelles : Pour optimiser la phase d'apprentissage, un encodage OneHot est privilégié dans la majorité des cas (si c'est une variable qualitative nominale). Un encodage ordinal est appliqué dans une minorité de cas (si la variable s'y prête) :
    - Si le nom de la variable commence par les termes `FLAG` / `REG` / `LIVE` ou si la cardinalité[b] de la variable est égale à 2 → encodage ordinal
    - Si la variable est multidimensionnelle → encodage OneHot.

Les variables des tables secondaires sont ajoutées et prétraitées dans une fonction à part (`add_secondary_table_features` disponible dans le script de preprocessing)

3. Modélisation du risque de défaut

La modélisation s'est effectuée en plusieurs étapes successives :

1. Les données d'entraînement ont été **rééquilibrées**.
2. **Différents algorithmes** d'apprentissage ont été **testés** ; leur performance a été évaluée.
3. Les **hyper-paramètres** du modèle choisi ont été **optimisés** pour maximiser la performance.
4. Le **seuil de décision** du modèle a été **adapté** au moyen d'une modélisation « naïve » du profit de HomeCredit sur chaque prêt.

3.1. Métriques d'évaluation de la performance

Déterminer si un modèle de classification est performant consiste à trouver une mesure qui décrit à quel point les prédictions qu'on lui demande de faire sont conformes à la réalité. Pour cela il faut déterminer ce que l'on cherche à savoir grâce au modèle.

Dans le cadre de la prévention du risque chez HomeCredit, la variable que l'on cherche à prédire décrit la présence (classe 1) ou l'absence (classe 0) d'un défaut de paiement pour un crédit. On a donc 2 classes à prédire, mais plusieurs approches pour évaluer la qualité de la prédiction.

	Prédit comme classe 0	Prédit comme classe 1
Est de classe 0	Vrai négatif (VN)	Faux positif (FP)
Est de classe 1	Faux négatif (FN)	Vrai positif (VP)

- Exactitude = (VN + VP) / (VN + FN + VP + FP)
  = le nombre de prédictions correctes sur l'ensemble des prédictions
- Précision (classe 1) = VP / (FP + VP)
  = le nombre de crédits qui font véritablement défaut parmi les crédits prédits comme tel
- Rappel (classe 1) = VP / (FN + VP)
  = parmi tous les crédits qui font défaut, combien ont-ils été prédits comme tel ?

On peut regarder si le modèle effectue des prédictions correctes toutes classes confondues (exactitude), ou au contraire s'intéresser à la qualité de la prédiction d'une des deux classes.

Dans le cadre du projet, on s'intéresse davantage à la prédiction des **défauts de paiement** que des non-défauts de paiement. C'est pourquoi l'algorithme de machine learning utilisé a été sélectionné selon la précision et le rappel de la classe 1 ; la classe des défauts.

Privilégier la détection des défauts de paiement peut néanmoins impacter négativement la qualité de la prédiction des non-défauts de paiement, et donc engendrer beaucoup de faux-positifs dans les prédictions. Cela peut avoir une incidence financière néfaste si beaucoup de prêts sont refusés « à tort ». Pour minimiser le manque à gagner du aux faux positifs, on ajustera sur le seuil final d'évaluation du modèle (voir partie 3.5).

3.2. Rééquilibrage des données

Le défaut de paiement est un phénomène minoritaire dans l'ensemble des crédits accordés. Sur l'échantillon considéré, seul 8 % des prêts présentaient un défaut ou un retard de paiement. Ce déséquilibre dans le jeu de données a un impact négatif sur la performance de prédiction des défauts. C'est pourquoi le jeu de données d'entraînement a été rééquilibré avant la sélection du type de modèle de machine learning, via un sous-échantillonnage des données originales. Cela signifie que des prêts sans défaut ont été exclus de manière à ce que les classes 0 et 1 soient de tailles équivalentes dans les données qui ont servi à entraîner les modèles.

En raison du grand nombre de prêts (environ 300.000), les méthodes d'élimination basées sur le voisinage des points de données comme SMOTE ou les liens de Tomek ont été écartées car elles sont trop gourmandes en temps comme en ressources de calcul. Un sous-échantillonnage aléatoire leur a été préféré.

3.3. Sélection du meilleur algorithme

Cinq types d'algorithme de machine learning ont été testés pour modéliser le risque de crédit :

Gradient stochastique
Arbre de décision
Random Forest
AdaBoost
Gradient Boosting (LightGBM)

Ces algorithmes ont été passés dans un même pipeline de traitement :

Sous-échantillonnage pseudo-aléatoire (la graine d'aléa a été figée)
Prétraitement des données (cf. partie 2.3.)
Normalisation des données (*Standard-scaling*) pour l'algorithme du gradient stochastique.

Pour tous les modèles l'hyper-paramètre qui gère l'aléatoire a été fixé avec une même graine.

Chaque modèle a été entraîné sur 80 % des données d'entraînement et évalué sur les 20 % restants.

Sur les données d'évaluation, on a cherché à prédire les étiquettes de défaut de paiement avec chaque modèle. Les critères d'évaluation furent dans l'ordre : maximiser le score f1[c] (1), maximiser le rappel (2) et maximiser la précision (3).

Modèle	Score F1	Rappel	Précision
Gradient Stochastique	0,24	**0,68**	0,15
Arbre de décision	0,19	0,60	0,11
Random Forest	0,25	0,65	**0,16**
Ada-Boost	0,25	0,67	**0,16**
LightGBM	**0,26**	**0,68**	**0,16**

Tableau 1 : Métriques d'évaluation pour les différents modèles testés

L'algorithme LightGBM qui maximise les trois indicateurs, a été retenu pour la modélisation.

3.4. Optimisation des hyper-paramètres

Le lightGBM est un algorithme de gradient boosting basé sur les arbres de décision. Un algorithme de gradient boosting entraîne des classifieurs peu performants à devenir très performants en augmentant par étapes successives le poids des points de données difficiles à classer.

Optimiser les hyper-paramètres d'un modèle consiste à trouver pour les différentes variables extérieures aux données d'entrée les valeurs qui vont maximiser la performance du modèle.

Le lightGBM possède une vingtaine d'hypers-paramètres. Deux étapes de recherche successives ont été nécessaires pour trouver les hyper-paramètres optimaux :

1. une recherche aléatoire : 100 combinaisons d'hyper-paramètres piochées parmi 100.000.000 de combinaisons possibles.
2. Une recherche en grille : sur les hyper-paramètres importants, 40 combinaisons testées.

Une optimisation des hyper-paramètres a permis d'améliorer les scores de précision et de rappel.

LightGBM	Score-F1	Rappel	Précision
Avant optimisation	0,26	0,68	0,16
Après optimisation	**0,28**	**0,71**	**0,17**

Tableau 2 : Métriques d'évaluation du lightGBM avant et après optimisation des hyper-paramètres.

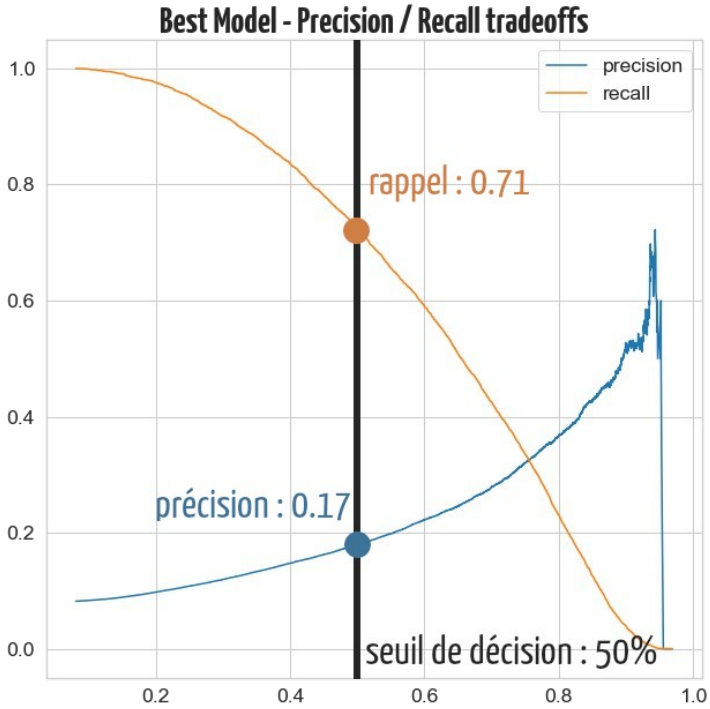
3.5. Ajustement du seuil de décision

Le modèle obtenu après les étapes de sélection & d'optimisation présente un bon niveau de rappel mais un faible niveau de précision. En d'autres termes, le modèle détecte assez bien les défauts de paiement (71% de ceux présents dans le jeu de test ont été détectés), mais – en contrepartie – le modèle génère beaucoup de faux positifs.

Se baser sur un modèle prédictif avec un rappel élevé mais une précision faible pour accepter ou refuser des demandes de prêt peut engendrer un manque à gagner préjudiciable à HomeCredit (cf. partie 3.1). Pour minimiser ce risque, on va donc ajuster le seuil de décision du modèle de manière à maximiser le profit réalisé sur l'ensemble des prêts.

3.5.1. Probabilité d'appartenance et seuil de décision.

Figure 3 : Graphique qui illustre différents niveaux de précision et de rappel sur la classe 1 pour le lightGBM optimisé en fonction du seuil de décision



Le modèle de machine learning qui a été construit ne prédit pas directement une classe mais des probabilités d'appartenance à chacune des classes de la variable-cible. Si la probabilité d'appartenance à la classe 1 est > 50 %, le modèle considère que l'étiquette à prédire doit être 1. Le seuil de décision par défaut pour notre modèle est donc de 50 %.

On peut modifier ce seuil de décision pour l'adapter aux intérêts de l'organisme de crédit ; cela aura pour conséquence de modifier la précision et le rappel de notre modèle pour la classe 1[d]. Pour trouver le seuil de décision idéal, il nous faut une modélisation du profit réalisé sur les prêts.

3.5.2. Fonction naïve de profit sur un prêt

Pour modéliser le profit de manière simple, plusieurs présupposés ont été établis :

- Le profit réalisé sur un prêt correspond au seul montant des intérêts sur le prêt.
- Le prix du bien (`AMT_GOODS_PRICE`) et le principal du prêt sont une même chose.
  - Le montant des intérêts = le montant du crédit (`AMT_CREDIT`) moins le prix du bien.
  - Toutes les entrées où `AMT_GOODS_PRICE` > `AMT_CREDIT` sont des erreurs ; ces valeurs doivent être corrigées et imputées.
- Les clients qui font défaut le font sur l'ensemble du prêt. On considère qu'ils n'ont jamais rien remboursé et ne rembourseront jamais le prêt. HomeCredit perd tout le principal.
- Une demande de prêt dont la probabilité de défaut est supérieure au seuil de décision est refusée. Le profit sur ce prêt est « logiquement » de 0 pour HomeCredit. Cela a pour conséquence un manque à gagner égal aux intérêts si le prêt est remboursable.

	Prédit comme classe 0	Prédit comme classe 1
Est de classe 0	Profit = + montant des intérêts	Profit = 0 (manque à gagner = montant des intérêts)
Est de classe 1	Profit = - montant du principal	Profit = 0 (coûts évité = montant du principal)

Tableau 3 : Profit réalisé sur un prêt HomeCredit selon la fonction de profit

À partir de ces hypothèses, le profit total a été modélisé plusieurs fois, en changeant pour chaque scénario le seuil de décision du modèle. 62 seuils de décision ont été testés. Parmi ces différentes situations, le profit maximum est atteint avec un seuil de décision d'environ 68 %.

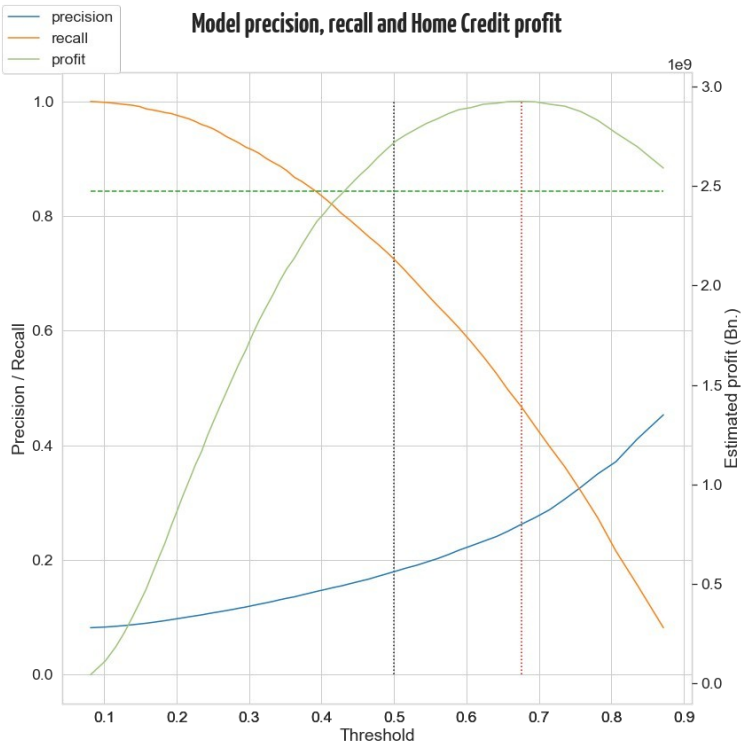


Figure 4 : Modélisation du profit de HomeCredit (en vert) en fonction du seuil de décision, avec la précision (bleue) et le rappel (orange) associé à chaque seuil.
La ligne verticale grise donne le niveau de profit, précision et rappel avec un seuil de 50 %.
La ligne verticale rouge donne le niveau de profit, précision et rappel avec un seuil d'~68 %.
La ligne horizontale correspond au profit sans prédictions, si tous les prêts avaient été acceptés.

4. Conclusion et limites de la modélisation

Au regard du dernier graphique, on peut conclure que suivre les prédictions de notre modèle de machine learning pour prévenir le risque de défaut ferait gagner de l'argent à HomeCredit, même avec un seuil de décision de 50 %.

Il convient de noter que la manière dont le profit a été modélisé est basé sur des hypothèses imparfaites, voire naïves, ce qui limite la portée des conclusions que l'ont peut tirer sur les bénéfices financiers éventuels.

Cette démarche d'évaluation des coûts économisés reste néanmoins valable si on remplace cette fonction naïve de profit par une modélisation réaliste du profit sur un crédit, qu'il soit entièrement remboursé, en retard de paiement, en défaut partiel ou en défaut total.

En conclusion, le machine learning est une approche pertinente pour anticiper le risque de défaut.

Notes de fin de document :

a Par exemple pour une variable qui donne la moyenne d'un indicateur pour un groupe donné, les valeurs manquantes sont remplacées par valeur moyenne de cette variable.
b La cardinalité d'une variable correspond au nombre de valeurs possibles pour cette variable.
c Le score f1 est la moyenne harmonique de la précision et du rappel pour une classe donnée.
d Plus le seuil de décision est élevé (proche de 1) plus la précision est forte et plus le rappel est faible.