|  |  |
| --- | --- |
|  | **Credit Scoring with Machine Learning**  Approche de modélisation et intégration au processus d’octroi de prêts |

Cette notice a pour objectif de développer spécifiquement :

Table des matières

[Introduction : Use Case 1](#_Toc58878510)

[Use Case détaillé 2](#_Toc58878511)

[Donnée cible : le défaut de paiement 3](#_Toc58878512)

[Données sources 4](#_Toc58878513)

[Panel des classificateurs binaires 5](#_Toc58878514)

[Méthodologie d'entraînement du modèle 5](#_Toc58878515)

[Fonction coût, algorithme d'optimisation et métrique d'évaluation 5](#_Toc58878516)

[Meilleur modèle et simplification 7](#_Toc58878517)

[Interprétabilité du modèle 9](#_Toc58878518)

# Introduction : Use Case

Le projet a pour cadre la décision d’octroyer ou non un prêt bancaire, dont l’Use Case primaire peut être décrit comme suit :



Figure 1 : Use Case primaire

Un client ***« Applicant »*** soumet une demande de prêt à l’organisme : la demande est analysée et donne lieu au refus ou à l’accord de prêt.

L’organisme collecte alors le motif du **refus** et, dans le cas de prêts **accordés**, l’ensemble des données de remboursement.

L’objectif du projet est double :

1. Construire l’approche Machine Learning permettant de prédire, à partir des données collectées, une future difficulté de remboursement,
2. Intégrer cette approche dans le processus d’octroi de prêt de l’organisme par la fourniture d’un tableau de bord interactif.

Pour cela, commençons par détailler notre use case.

# Use Case détaillé

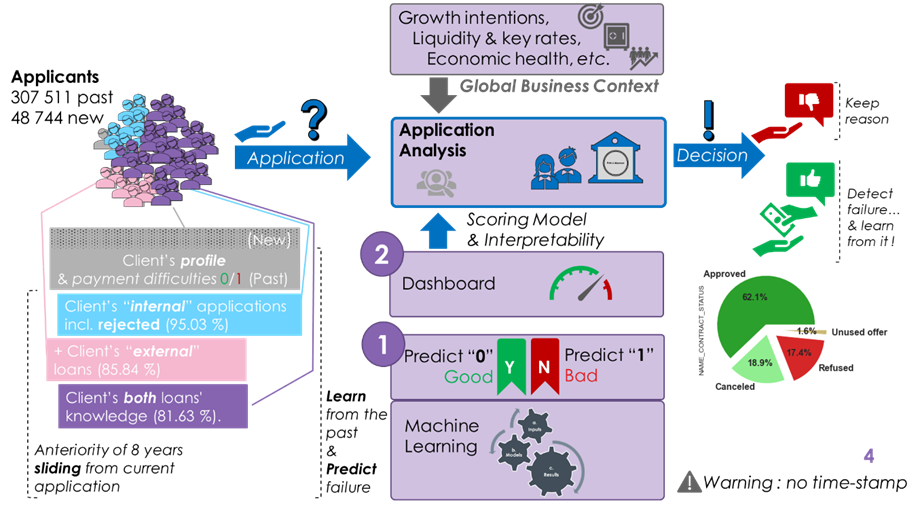


Figure 2 : Use Case détaillé

En accumulant au fil du temps le suivi des remboursements des prêts, l’organisme a construit un indicateur cible identifiant une difficulté de remboursement. Cette cible ***« TARGET »*** est connue pour 307 511 dossiers et consiste en une information binaire qui vaut 1 que lorsqu’un client a connu un défaut de payement ***« failure »*** durant plus de X jours pendant les Y premiers remboursements.

L’organisme dispose donc des caractéristiques du prêt demandé, des données de profil du client ainsi que **8 années** d’antériorité (glissantes) des emprunts internes ou externes contractés par le client.

Nous reviendrons ultérieurement sur l’analyse des données disponibles.

Les 2 objectifs du projet sont illustrés ci-dessus 1. ***Learn & Predict***, 2. ***Dashboard***.

Ils consistent à fournir et étayer pour chaque dossier parmi les 48 744 nouvelles demandes, une **prédiction de la cible**, **interprétable** comme un risque futur de défaut de paiement par le client.

On note que **17,4 % des demandes sont refusées**, et 62,1 % des demandes accordées sont effectivement mise en place. Cela ouvre un espace entre la « sévérisation » des refus et le risque de défaut de paiement réellement observé : plus de 2 fois plus de dossiers sont considérés porter un risque suffisamment élevé pour refuser la demande.

Le projet s’inscrit dans un contexte élargi des KPIs de l’organisme comme les stratégies de croissance, ainsi que des paramètres externes tels que les taux directeurs ou la santé économique.

Ces éléments de contexte étant non seulement inconnus mais eux-mêmes fluctuants au cours du temps (et nous agissons en temps relatif), nous pouvons au mieux considérer qu’il existe des facteurs externes qui pilotent l’activité d’analyse/décision et proposer à ce titre une logique d’interprétation et un tableau de bord ouvert.

# Donnée cible : le défaut de paiement

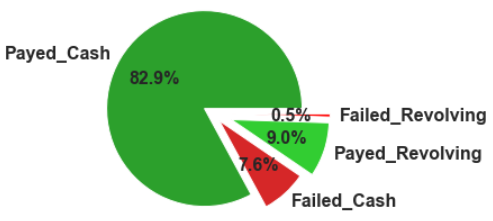
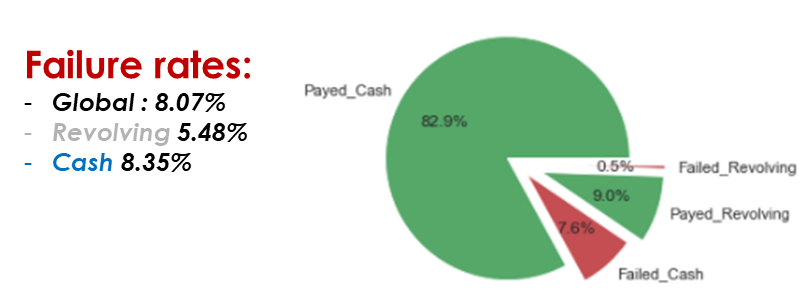


Figure 3 : taux de défauts de paiements

Les taux de défauts (failure rates) concernent environ 10% des cas, avec une prévalence des prêts classiques dits *‘cash loans’* sur les prêts dits *‘revolving loans’* ou renouvelable : lorsque l’organisme met à disposition une somme d’argent réutilisable au fur et à mesure du remboursement, pour effectuer des achats non prédéfinis.

Le risque lié à un défaut de remboursement semble moindre dans le cas de prêts renouvelables, en effet, ces derniers engagent des moindres sommes, et offrent une possibilité d’interruption permettant de limiter les dommages financiers.

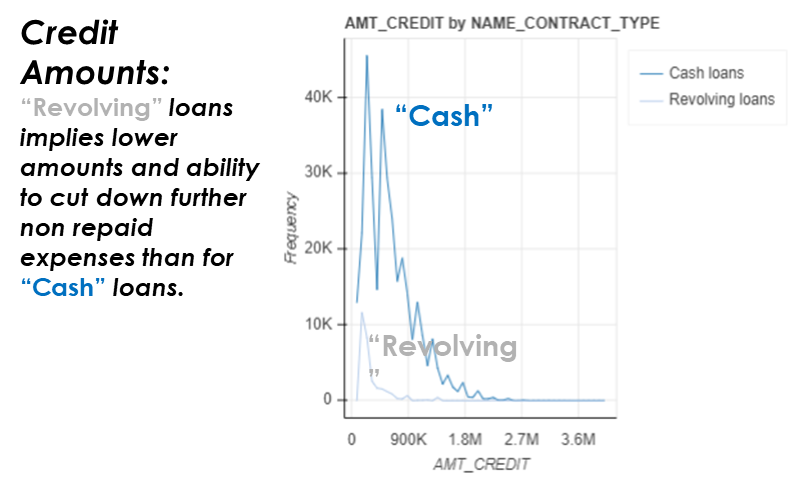


Figure 4 : montants des crédits

Dans notre étude, nous allons néanmoins considérer que le couple {X, Y}, qui traduit les données de paiement en cible binaire, a été décliné de façon à équilibrer pour les 2 types de prêts cette notion de risque associé à un défaut (target = 1).

* **Une perspective serait toutefois d’étudier de façon distincte ces 2 cas de figures.**

# Données sources

7 jeux de données sources détaillent le « profil » du client et la description du prêt qu’il sollicite et l’antériorité interne et externe éventuellement disponible.

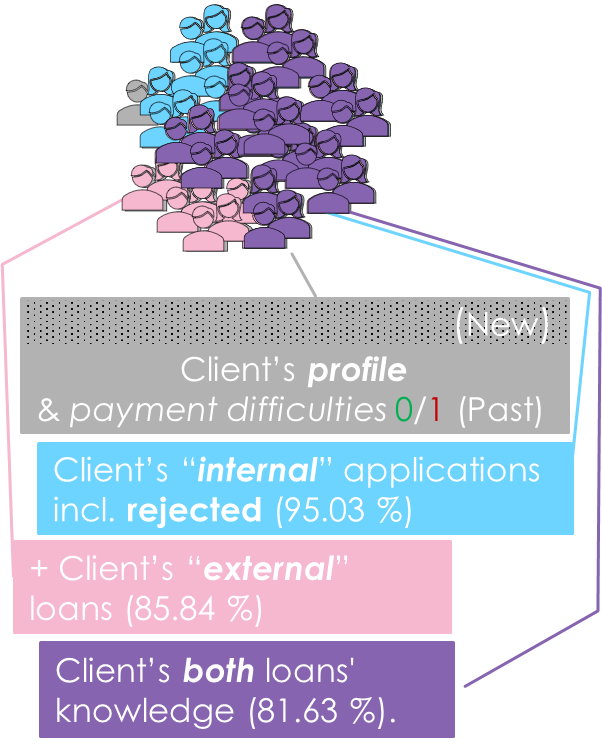


Figure 5 : données de profil et d’antériorité client

Comme l’illustre la figure ci-dessus, nous disposons à divers degrés de l’antériorité client, atteignant le cumul de sources externes et internes pour plus **80 % des dossiers**, sans présager du nombre et de l’ancienneté des prêts antérieurs disponibles.

A cette étape du projet, il est demandé de réemployer un kernel Kaggle. Si cela ouvre un gain de temps substantiel pour avancer dans le projet, cela peut néanmoins conduire à handicaper la bonne marche du projet.

En particulier, tandis que les compétiteurs se focalisent sur l’obtention du meilleur score par tous les moyens, notre objectif est de construire une démarche **interprétable** et même nous efforcer de rendre **transparente** l’exploitation du Machine Learning.

En ce sens, le Feature Engineering et l’étape de Feature Selection font parties intégrantes de l’approche de modélisation et nous avons engagé une approche **Top-Down** pour l’exploration des données combinée à une approche **Bottom-Up** de Feature Engineering (manuel et automatique), pour aboutir à l’agrégation des données.

Les problématiques soulevées lors de cette phase préliminaire sont nombreuses et didactiques, sur les façons de préserver l’intégrité des features, de remédier au déséquilibre des classes ou encore d’imputer les valeurs manquantes de façon adéquate. Le tout étant **interdépendant**.

Une étape indépendante des modèles (dite model-agnostique) a également été menée à l’aide du package python Boruta.

* **Une perspective, au-delà des pistes en réponse aux problématiques soulevées, serait toutefois d’itérer une fois le cadre de modélisation sélectionné (interdépendance).**

# Panel des classificateurs binaires

Nous avons retenu le test d’un panel : modèle linéaire : **Logistic Regression** ; modèle ensembliste **Random Forest** « simple » ; et **LightGBM** « avancé » (au sens de la construction des arbres de décisions guidée par une méthode de gradient. Ces derniers permettent de valoriser des apprenants faibles et offrent des interprétations des « features importances ».

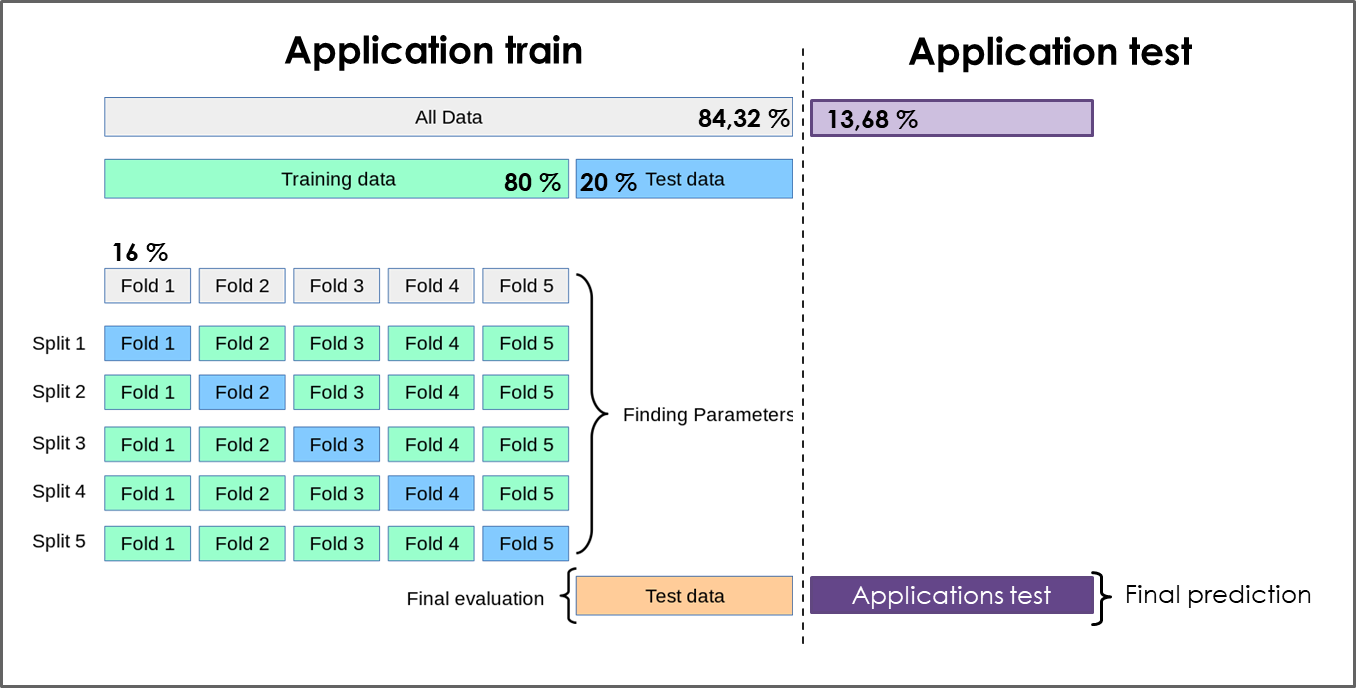
Nous avons également testé une méthode de remédiation au problème de déséquilibre des classes, SMOTE, basée sur le rééquilibrage via la création de données de synthèse, que nous avons testé contre le paramètre des modèles qui permet de remédier au déséquilibre par une méthode de poids des classes.

# Méthodologie d'entraînement du modèle

L’ensemble des dossiers ou « applications » ayant été regroupés dans l’étape de Feature Engineering et global Feature Selection, nous redissocions les dossiers « Application train » pour lesquels la cible est connue et les dossiers « Application test » pour lesquels nous réaliserons ultérieurement la prédiction finale que nous étaierons via le tableau de bord.

Pour l’entraînement et la validation de nos modèles, nous procédons à une nouvelle séparation en données d’apprentissage et de test, selon une méthode de split stratifié selon la cible.

Puis nous mettons en œuvre une cross-validation sur 5 Folds pour nous assurer de la robustesse des résultats obtenus.

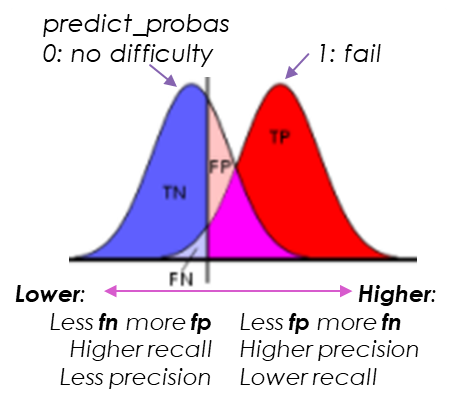
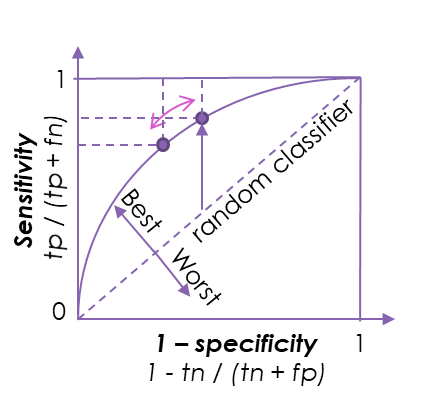


# Fonction coût, algorithme d'optimisation et métrique d'évaluation

Nous avons opté pour l’algorithme Hyperopt, afin d’explorer un large espace de recherche d’hyperparamètres des modèles, mettant en œuvre la cross-validation.

La fonction de coût retenue et appliquée en tant que ***« objective function »*** de notre algorithme Hyperopt est **le score AUC**, ou ***aire sur la courbe ROC.*** L’idée est d’obtenir un classificateur polyvalent, c’est-à-dire d’éviter d’optimiser nos paramètres pour un problème métier spécifique au détriment d’autres types d’optimisations que pourraient choisit le métier.

Nos modèles entrainés constituent une famille de classificateurs, représentés par leur **courbe ROC** illustrée ci-dessous (Receiver Operating Characteristic). Issue du traitement du signal, cette mesure est exploitée en classification binaire pour mesurer la performance du modèle : il est globalement d’autant plus performant qu’il maximise l’aire sous la courbe.



Figures 6a et 6b : Courbe ROC et notion de seuil de prédiction

La courbe ROC est obtenue en faisant varier le seuil de prédiction (ou discrimination, figure de droite), pour lequel la valeur de probabilité de classe 1 entraine une classe 1 prédite, et retourne les valeurs de sensibilité et d’antispécificité associées. Cela revient à moduler les effectifs de la matrice de confusion, illustrée ci-dessous.

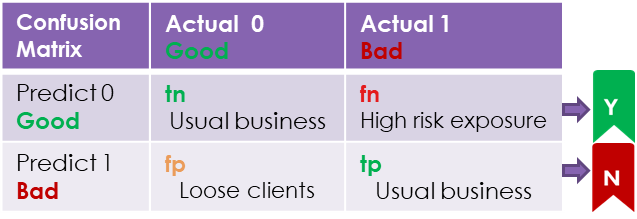


Figure 7 : matrice de confusion

Ainsi, en fonction de la définition de notre cible :

* On s’attend à faire confiance à une prédiction négative en cas de **tn : true negative**, mais elle nous expose en cas de **fn : false negative** aux risques de défaut en accordant le prêt.
* On s’attend à faire confiance à une prédiction positive en cas de **tp : true positive**, mais elle signifie en cas de **fp : fasle positive**, refuser le prêt et perdre un « bon » client potentiel.

Nous avons besoin de minimiser les erreurs de prédictions avec un plus grand intérêt porté aux **faux négatifs**, par conséquence, minimiser les **fn** augmente les **fp.** Nous avons donc décidé d’observer une métrique alternative Fbeta où la valeur de beta représente l’importance relative accordée aux fp par rapport aux fn, pouvant traduire une métrique du dommage financier.

Un autre aspect très important dans notre contexte, est que notre classificateur étant non parfait nous aurons recours à une métrique supplémentaire afin d’optimiser les résultats des prédictions.

# Meilleur modèle et simplification

Un **AUC score de 0.7646** est obtenu pour notre meilleur modèle : un **LGBMClassifier**, avec pour principaux paramètres : (boosting\_type='goss', class\_weight='balanced', learning\_rate=0.019…, max\_depth=23, estimators=610, num\_leaves=52).

L’utilisation du Fbeta score peut aider à identifier un meilleur seuil de prédiction, celui pour lequel le problème est optimal du point de vue métier.

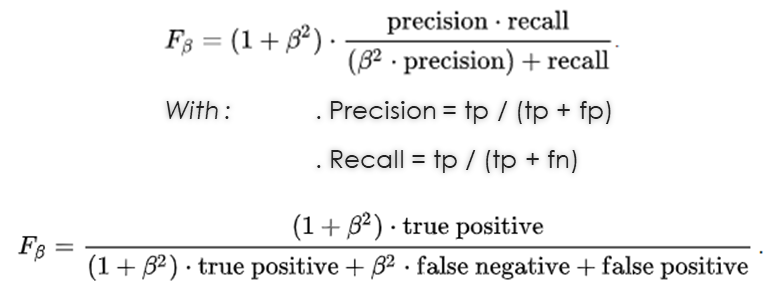


Figure 8 : Métrique additionnelle F-beta

Ci-dessous nous présentons les valeurs obtenues pour Fbeta (2) et le taux de défaut associés aux valeurs du seuil. Le seuil optimal au sens du Fbeta étant 0,47.

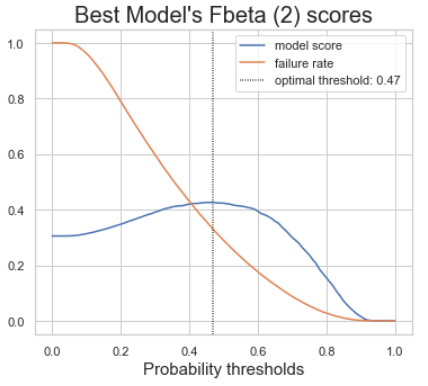


Figure 9 : Compromis sur le seuil de classification

Cet ajustement peut être réalisé dans des conditions réalistes, c’est-à-dire en surveillant la valeur du taux de dossiers refusés, entre l’observation réaliste des taux de défauts constatés, et le taux de refus des dossiers antérieurs. Ici le « meilleur » seuil représente plus de 30 % de dossiers rejetés : même en considérant une part de dossiers annulés requalifiés en défauts évités.

* **Une perspective serait de qualifier la métrique métier et confronter nos observations au contexte de la société.**

S’agissant de la simplification, nous avons retenus les features dont l’importance cumulée permet d’atteindre 80 % du total, cela permet de réduire de moitié le nombre de features requis (125) tout en conservant un score non dégradé : **l’AUC score obtenu est 0.764**.

* **D’autres techniques de features selection peuvent être introduites pour tenir compte de l’intérêt combiné des features.**
* **Une perspective pour le modèle serait de limiter des paramètres complexifiant son interprétabilité (de plus petits arbres de décisions, dans l’esprit surrogate model, etc).**

# Interprétabilité du modèle