Université de Caen
Normandie

IUT Grand Ouest Normandie

Bachelor Universitaire de Technologie

Science des Données

Campus de Lisieux

Science des Données 2 - RES 5-03

Mise en œuvre d'un processus de Datamining

Thématique

Implémentez un modèle de scoring crédit

Auteurs

BELOIN Lucas
GAYE Youssouf
DIOP Mandir

Année universitaire 2025–2026

Table des matières

1 Introduction et Contexte

1.1 Le Contexte Métier

La distribution de prêts fait partie des activités de base des banques. Pour un groupe financier par exemple comme **Home Credit Group**, la problématique consiste essentiellement à bien évaluer la *capacité de remboursement* de ses clients, c'est-à-dire leur **solvabilité**.

Néanmoins, beaucoup de clients potentiels ont peu voire pas d'histoire de crédit classique. Il s'agit d'utiliser différents types de données afin de pouvoir prédire le risque de défaut de paiement :

— Données comportementales ;
— Données financières ;
— Données sociodémographiques.

L'objectif de ce projet est de développer un algorithme de Machine Learning capable d'automatiser cette décision en classant les demandes de prêt en deux catégories :

— **Crédit Accordé** (Client fiable) ;
— **Crédit Refusé** (Client à risque) ;

1.2 La Problématique : L'équilibre Risque / Rentabilité

Il est crucial de distinguer les deux types d'erreurs que le modèle peut commettre, car elles n'ont pas le même impact financier :

Le Risque de Crédit (Faux Négatifs)
C'est le cas où le modèle prédit qu'un client est **fiable**, alors qu'il ne remboursera pas.
Conséquence : **Perte du capital prêté** (très coûteux).

Le Risque Commercial (Faux Positifs)
C'est le cas où le modèle **refuse** un client par sécurité, alors qu'il aurait remboursé.
Conséquence : **Manque à gagner** sur les intérêts (moins coûteux, mais dommageable pour la croissance).

Conclusion stratégique : Notre mission n'est donc pas seulement d'avoir un modèle "juste" (exactitude), mais un modèle **rentable** qui minimise avant tout les pertes de capital.

1.3 Présentation des Données

Les données utilisées proviennent de la compétition Kaggle "Home Credit Default Risk". Il s'agit d'un jeu de données complexe et relationnel comprenant plusieurs fichiers

Le volume et la diversité de ces données nécessitent une étape majeure de nettoyage et de préparation (Feature Engineering) avant toute modélisation.

Nom	Taille	Compressé	Type	Modifié	CRC32
..			Dossier de fichiers		
HomeCredit_columns_description.csv	37 163	?	Fichier CSV Micros...	12/12/2025 11:09	CFDE7116
application_train-LFS.txt	166 133 370	432 315 622	Document texte	12/12/2025 11:10	EDD0B71A
bureau-LFS.txt	170 016 717	?	Document texte	12/12/2025 11:12	8C620F57
bureau_balance-LFS.txt	375 592 889	?	Document texte	12/12/2025 11:13	5B0F0F5C
POS_CASH_balance-LFS.txt	392 703 158	?	Document texte	12/12/2025 11:10	F98D9C37
previous_application-LFS.txt	404 973 293	?	Document texte	12/12/2025 11:18	F5B0DA6B
credit_card_balance-LFS.txt	424 582 605	?	Document texte	12/12/2025 11:14	C8CB85DC
installments_payments-LFS.txt	723 118 349	?	Document texte	12/12/2025 11:16	81EDDFA6

Fig. 1. Données utilisées pour le projet

2 Préparation et Nettoyage des Données (Feature Engineering)

2.1 Traitement et Agrégation des Données

Les données étaient brutes et réparties dans plusieurs fichiers (tables relationnelles). Un client (`SK_ID_CURR`) correspond à une ligne dans le fichier principal, mais peut correspondre à plusieurs lignes dans les fichiers historiques (ex : 5 crédits demandés, 12 relevés de compte, etc.).

Il a fallu que nous fassions une agrégation sur ces fichiers secondaires avant de les *merge*.

Méthode : Pour chaque client, nous avons calculé des statistiques (Moyenne, Min, Max, Somme) sur ses interactions passées.

Résultat : Nous sommes passés d'une structure relationnelle complexe à un **Tableau Unique** (*Master Table*) contenant toutes les informations par client.

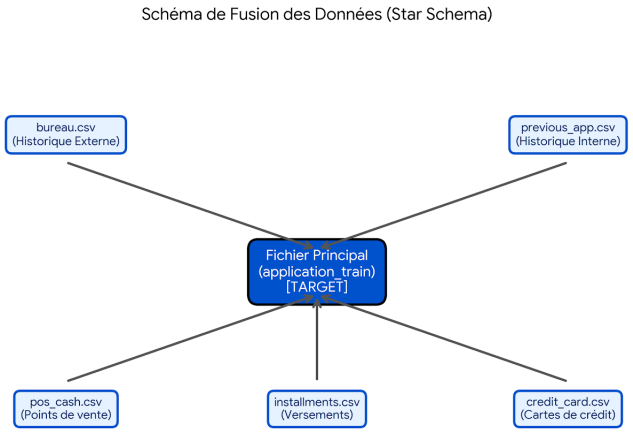

Fig. 2. Plusieurs fichiers -> 1 fichier

2.2 Nettoyage et Encodage

Une fois les données fusionnées, nous avons traité deux problèmes majeurs inhérents aux données brutes :

Valeurs Manquantes (NaN) : De nombreuses colonnes contenaient des trous. Nous avons opté pour une stratégie de remplissage par zéro (`fillna(0)`) après analyse, considérant que l'absence d'information (ex : pas de retard de paiement noté) équivaut souvent à une absence d'incident.

Transformation des colonnes qualitatives : Comme les algorithmes de machine learning ne traitent que des variables numériques, nous devions coder les modalités des variables qualitatives (ex : Sexe : "Homme", "Femme", Statut : "salarié", "indépendant"...). Pour cela, nous avons opté pour le *One-Hot Encoding*, qui produit une variable binaire pour chaque modalité.

2.3 Sélection des Variables (Réduction de dimension)

Après fusion et encodage, notre jeu de données a explosé pour atteindre plus de 1600 colonnes. Entraîner un modèle sur autant de variables pose deux problèmes majeurs :

— **Risque de sur-apprentissage** (*Overfitting*) ;
— **Lenteur de calcul** et consommation excessive de mémoire.

Nous avons appliqué une sélection rigoureuse pour ne conserver que les **50 meilleures variables**.

Critère : Nous avons analysé la corrélation de chaque variable avec la cible (`TARGET`).

Bénéfice : Cette réduction drastique (de 1600 à 50) a permis d'accélérer l'entraînement tout en conservant l'essentiel de l'information prédictive.

2.4 Rééquilibrage des Classes

L'analyse exploratoire a révélé un fort déséquilibre : environ 92% des clients remboursent leur crédit (**Classe 0**), contre seulement 8% de défauts de paiement (**Classe 1**). Un modèle entraîné sur ces données brutes aurait tendance à ignorer la classe minoritaire.

Nous avons corrigé ce biais via une technique de *Random Oversampling* :

Méthode : Nous avons dupliqué aléatoirement les exemples de « mauvais payeurs » dans le jeu d'entraînement.

Résultat : Le modèle apprend sur une base équilibrée (50% de bons / 50% de mauvais), ce qui le force à mieux détecter les fraudes.

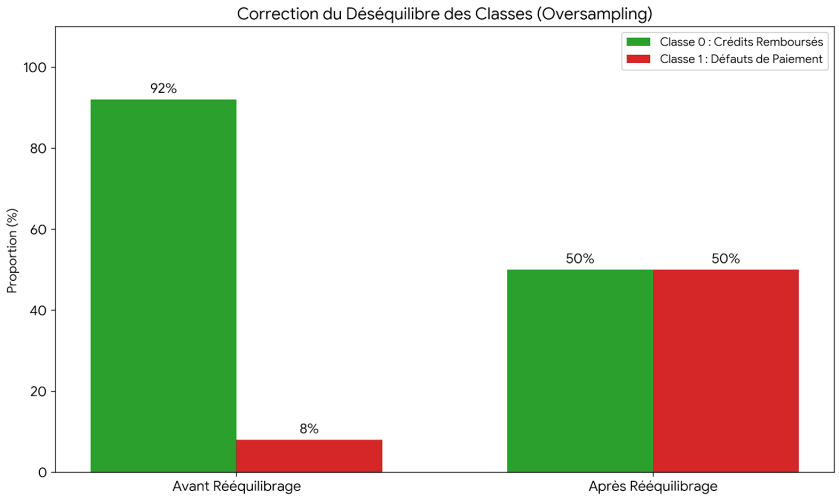

Fig. 3. Avant équilibrage -> Après équilibrage

3 Modélisation et Comparaison

3.1 Protocole Expérimental

Pour évaluer nos modèles de manière impartiale, nous avons appliqué un protocole strict.
Nous avons divisé le jeu de données en deux parties :

— **80% pour l'Entraînement** (*Train set*) : Utilisé pour que les algorithmes apprennent les motifs de fraude.

— **20% pour le Test** (*Test set*) : Conservé intact (« caché » aux modèles) pour simuler l'arrivée de nouveaux clients et vérifier la performance réelle.

3.2 Métrique d'évaluation : L'AUC (Area Under the Curve)

Plutôt que la simple « Précision » (trompeuse ici car 92% des clients sont honnêtes), nous avons choisi l'**AUC-ROC**.

Définition : Cette métrique mesure la capacité du modèle à bien classer un bon payeur devant un mauvais payeur.

Interprétation : Un score de **0.5** équivaut au hasard (pile ou face), tandis qu'un score de **1.0** est parfait.

3.3 Modèles Évalués

Nous avons mis en compétition trois algorithmes de complexité croissante :

Régression Logistique (*Baseline*)
Il s'agit d'un modèle linéaire basique et interprétable. Il sert de benchmark. Si ce modèle ne performe pas suffisamment, cela indique que le problème ne peut pas être résolu de manière linéaire

Random Forest (Forêt Aléatoire)
Un modèle d'ensemble qui combine des centaines d'arbres de décision. Il s'agit généralement d'un modèle plus robuste qui peut saisir des relations non-linéaires.

XGBoost (*eXtreme Gradient Boosting*)
Le modèle de référence pour les données tabulaires. Il apprend au travers de phases **séquentielles** où chaque nouvel arbre corrige les erreurs des précédents.

3.4 Analyse des Performances (Résultats)

Après entraînement sur les données rééquilibrées, voici les résultats obtenus sur le jeu de test :

Régression Logistique (AUC = 0.61)
Elle montre ses **limites**. Les relations entre les variables (âge, scores externes, historique) sont trop complexes pour une simple ligne droite.

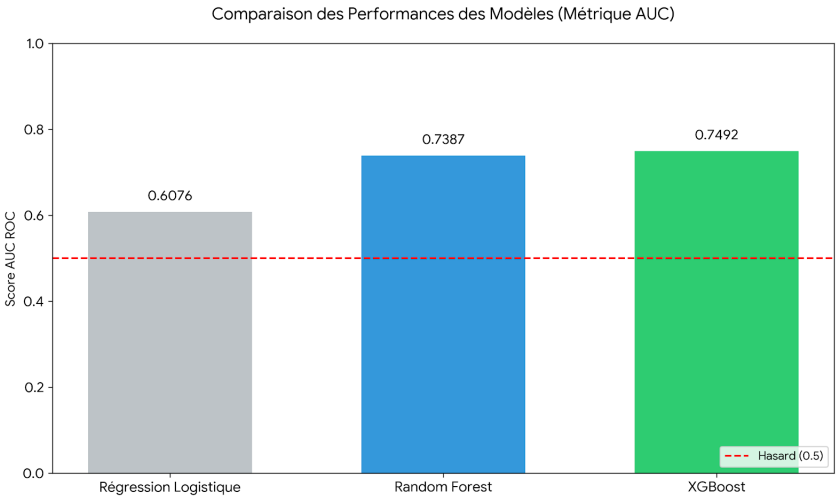

Fig. 4. Résultats des performances

Random Forest (AUC = 0.74)
Un **bond significatif** en performance. Il parvient à capter la complexité du problème.

XGBoost (AUC = 0.75)
C'est le plus **performant**. Il offre la meilleure performance prédictive, tout en étant optimisé pour la vitesse d'exécution.

Nous avons retenu XGBoost pour la mise en production

4 Interprétabilité du Modèle

4.1 Sortir de l'effet "Boîte Noire"

Dans le secteur bancaire, on n'a pas le droit de dire « non » sans argumenter. La réglementation (et la logique commerciale) exigent la **clarté**. Ce n'est pas suffisant d'avoir un bon modèle *performant*, il doit être **interprétable**.

Nous avons analysé l'importance des variables (*Feature Importance*) sur le modèle **XGBoost** pour comprendre quels sont les critères les plus lourds dans la décision finale d'octroi ou de refus.

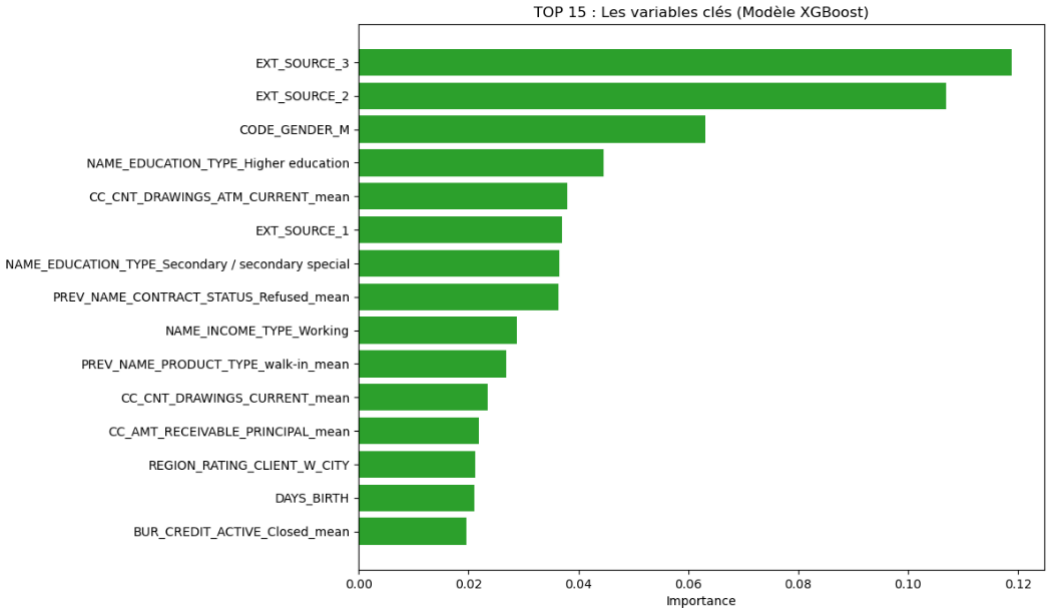

Fig. 5. Classement des 15 variables les plus influentes pour le modèle XGBoost.

4.2 Analyse des Facteurs de Risque

Le 1er graphique démontre l'importance du modèle que nous avons construit pour notre métier. L'intelligence artificielle ne s'appuie pas uniquement sur des signes faibles ou des corrélations bizarres, mais sur trois piliers :

1. La Réputation Externe (Les juges de paix)
Variables : `EXT_SOURCE_3`, `EXT_SOURCE_2`, `EXT_SOURCE_1`.
Analyse : Ce sont, de très loin, les variables les **plus importantes**. Il s'agit de scores de solvabilité provenant d'organismes externes (*Credit Bureau*).
Interprétation : Le modèle accorde une immense confiance à l'historique bancaire global. Si le client est bien noté ailleurs, notre modèle lui fera confiance.

2. Le Profil Sociodémographique
Variables : `DAYS_BIRTH` (Âge), `CODE_GENDER` (Genre), `EDUCATION`.
Analyse :
— *L'Âge :* C'est un facteur déterminant. Les données montrent que la stabilité financière s'accroît avec l'âge (les profils seniors sont moins risqués que les très jeunes actifs).
— *L'Éducation :* Un niveau d'études supérieures est corrélé positivement à la capacité de remboursement.

3. Le Comportement Bancaire (Signaux d'alerte)
Variables : `CC_CNT_DRAWINGS_ATM` (Retraits Cash), `PREV...REFUSED` (Refus passés).
Analyse : Le modèle sanctionne les comportements de « trésorerie tendue ».
— Un client qui utilise fréquemment sa carte de crédit pour retirer du liquide (ATM) est vu comme **risqué**.
— Un client ayant déjà essuyé des refus de prêt par le passé est **fortement pénalisé**.

4.3 Conclusion sur la cohérence

Cette étude montre que le modèle **XGBoost** a compris des règles tout à fait logiques et conformes à ce que ferait un analyste financier.

Ce n'est pas une *« boîte noire »* folle, mais un **assistant expert** qui suit des règles de gestion très strictes et qui, du fait de son automatisation, peut les appliquer de manière tout à fait **cohérente et rigoureuse** sur des milliers de dossiers.

5 Optimisation Métier (Approche Business)

5.1 La Limite du Modèle Technique

Jusqu'à présent, notre modèle XGBoost prenait une décision binaire standard :
— Si la probabilité de défaut est $< 50\% \rightarrow$ **Accordé**.
— Si la probabilité de défaut est $> 50\% \rightarrow$ **Refusé**.
Cependant, dans le monde bancaire réel, cette symétrie n'a pas de sens économique. Se tromper a des coûts très différents selon le type d'erreur :

Refuser un bon client (Faux Positif) :
C'est un **manque à gagner** (perte des intérêts du prêt), estimé arbitrairement à **1**.

Accorder un crédit à un mauvais payeur (Faux Négatif) :
C'est une **perte sèche** (perte du capital prêté), estimée à **10**.

Le défaut de paiement coûte **10 fois plus cher** que le refus d'un client. Le seuil standard de 0.50 n'est donc pas adapté.

5.2 Recherche du Seuil Optimal (Fonction de Coût)

Afin de maximiser le profit de la banque, nous avons défini une « fonction de coût » qui nous est propre. Nous avons testé tous les seuils de décision possibles (de 0 à 1) afin de déterminer celui qui minimise la perte totale de la banque.

5.3 Résultats et Impact Stratégique

La courbe ci-dessus montre clairement que le seuil par défaut (0.50) n'est pas le point le plus bas (le moins coûteux).

Seuil Optimal Trouvé : 0.53 (53%)

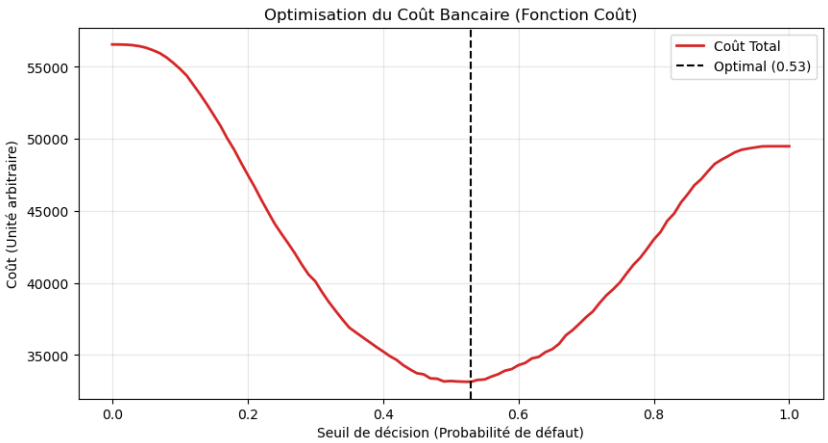

Fig. 6. Minimisation du coût bancaire en fonction du seuil de décision.

Interprétation :

— Le modèle était naturellement trop « prudent » ou « sévère ».

— En déplaçant le curseur à 0.53, nous acceptons de prendre un risque très légèrement supérieur.

Le Gain : Cela nous permet de récupérer un grand nombre de clients honnêtes (qui avaient une note entre 0.50 et 0.53) que nous refusions à tort auparavant.

5.4 Conclusion Business

En fixant ce seuil optimal à **0.53**, nous faisons passer notre modèle de la simple modélisation du risque à un véritable **levier de rentabilité**.

Nous ne nous contentons plus de prédire le risque mais nous permettons de piloter le risque/rentabilité pour minimiser le niveau de perte de l'établissement.

6 Mise en Production

6.1 De l'Algorithme à l'Outil Métier

Un modèle de Machine Learning de high quality, cela reste quelque chose de peu compréhensible pour les clients FinTech et autres conseillers bancaires. Ils ont besoin d'une interface simple, claire, intuitive et surtout rapide, pour qu'ils puissent prendre LA bonne décision tout de suite, à l'instant L, face au client.

C'est pourquoi nous avons développé une application interactive en utilisant le framework **Streamlit**.

6.2 Fonctionnalités du Dashboard

Notre outil, baptisé "Outil d'Octroi de Crédit", permet de simuler le parcours complet d'une demande de prêt :

Sélection du Client : Le conseiller choisit un identifiant client (`SK_ID_CURR`) dans la base de données.

Visualisation du Profil : Les informations clés (Âge, Ancienneté, Scores externes) sont pré-remplies automatiquement.

Simulation (*What-If Analysis*) : Le conseiller peut modifier manuellement certaines valeurs (ex : ajuster l'âge ou le score externe) pour voir comment le modèle réagit.

Décision Automatisée : En cliquant sur « Lancer l'analyse », l'application interroge le modèle **XGBoost** en arrière-plan.

6.3 Intégration de la Stratégie Métier

L'application ne se contente pas de fournir une probabilité brute (ex : 62%). Elle intègre en fait directement notre stratégie d'optimisation financière définie précédemment :

— Elle compare la probabilité calculée au seuil optimal de **53%**.

— Elle affiche une décision claire en code couleur :

— **VERT** (Crédit Accordé) si le risque est inférieur à 53%.

— **ROUGE** (Crédit Refusé) si le risque dépasse 53%.

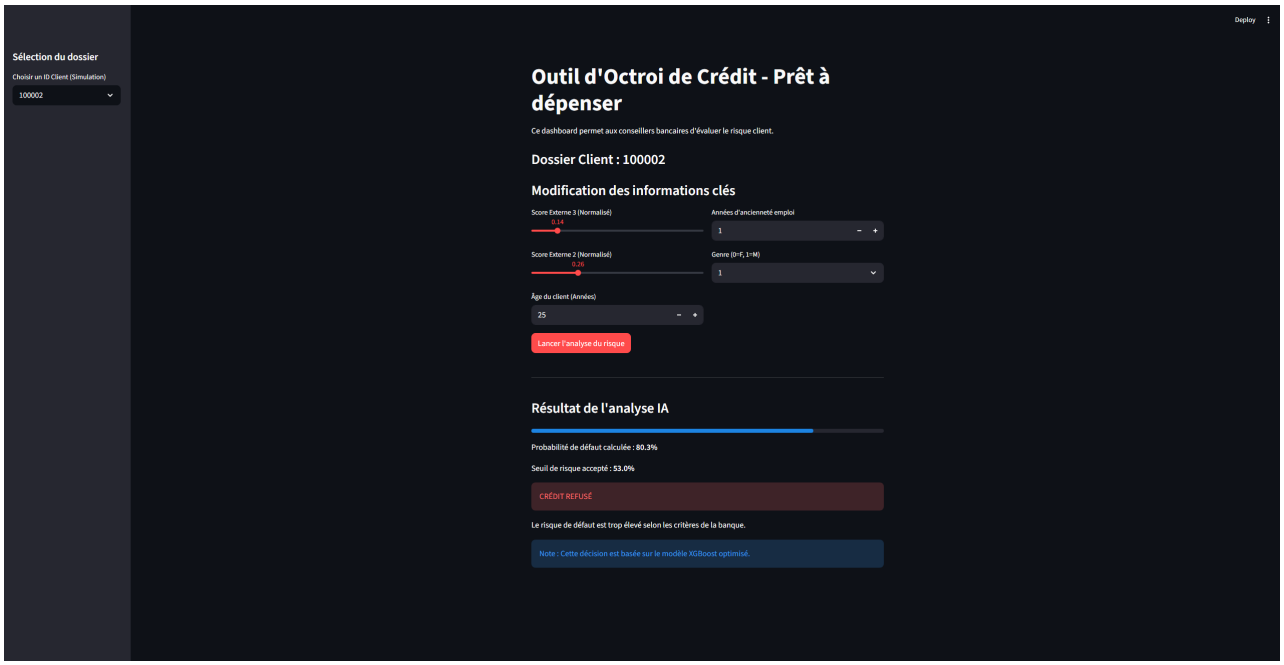

Fig. 7. Test et interface du site web

Cela garantit que tous les conseillers appliquent la politique de risque la plus **rentable** pour la banque, sans avoir à faire de calculs manuels.

Conclusion

6.4 Bilan du Projet

En ce sens, ce challenge a été pour nous l'occasion de :

Préparer : Transformer un jeu de données brutes, complexes et avec des tables relationnelles en un Dataset utilisable par un Data Scientist, appliquer un traitement des valeurs manquantes et des déséquilibres de classes (Oversampling).

Performer

- **Techniquement :** Aller plus loin qu'une régression Logistique traditionnelle en développant un modèle XGBoost qui surpasse cette dernière et atteint un AUC de 0.75.
- **Économiquement :** Faire la démonstration qu'il y a un intérêt à calibrer le seuil de décision (passer de 0.50 à 0.53) pour minimiser le coût du risque pour la banque.

Opérationnaliser : La mise en production de ce modèle via un Dashboard Streamlit permet une prise en main immédiate par les équipes métier.

6.5 Limites et Pistes Futures

Pour aller plus loin et industrialiser ce prototype, plusieurs pistes d'amélioration existent :

Explicabilité Locale (Ex : SHAP) : Afficher dans le Dashboard de monitoring des graphiques de SHAP Values qui explique pourquoi - selon le modèle - un client a été refusé (ex : "refusé(e) principalement à cause de votre historique de crédit").

Déploiement Cloud : Déployer l'application sur une plateforme Cloud (AWS, Azure ou Heroku) pour qu'elle soit accessible via une URL par tous les employés de la banque et plus seulement en local.

Monitoring (Data Drift) : Mettre en place du suivi pour détecter si le modèle devient obsolète dans le cas où le profil des clients change dans le futur.

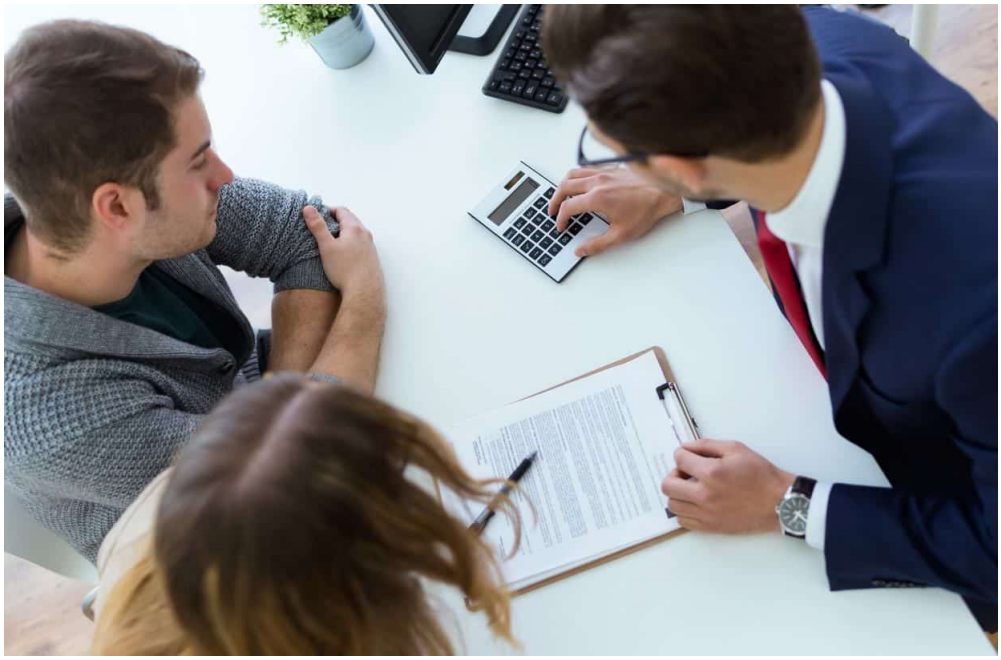

Fig. 8. Image de conclusion

Table des figures