



**Implémenter un  
modèle de Scoring :**

**Note Méthodologique**



# Contexte et Objectifs

---



La société financière Prêt à Dépenser propose des crédits à la consommation destinés aux individus ayant peu ou pas du tout d'historique de prêt.

Dans le cadre de l'implémentation d'un modèle de scoring, l'entreprise vise à mettre en place un outil de notation de crédit capable de calculer la probabilité de remboursement d'un client et de classer la demande en crédit accordé ou refusé. À cette fin, elle envisage de développer un algorithme de classification en exploitant diverses sources de données telles que les données comportementales et celles provenant d'autres institutions financières. Les données originales sont disponibles en téléchargement sur Kaggle à l'adresse spécifiée.

Pour répondre à une préoccupation exprimée par les chargés de relation client concernant la demande croissante de transparence de la part des clients, l'entreprise décide de créer un tableau de bord interactif. Ce tableau de bord, réalisé avec Streamlit, a pour objectif d'offrir aux chargés de relation client un moyen transparent d'expliquer les décisions d'octroi de crédit. Il permet également aux clients d'accéder facilement à leurs informations personnelles et de les explorer.

Le dashboard final, accessible en cliquant [ici](#) (Streamlit Cloud) ou [ici](#) (Heroku), ainsi que l'ensemble des programmes utilisés, sont disponibles sur [Github](#)



# Sommaire

I Les étapes préalables à la modélisation

II La méthodologie d'entraînement du modèle

III La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation

IV L'interprétation globale et locale du modèle

V Les limites et les améliorations possibles



# Note Méthodologique

---

Cette note méthodologique expose les techniques utilisées dans l'élaboration du modèle de scoring, ainsi que les outils déployés pour en faciliter l'interprétation. Pour atteindre cet objectif, nous aborderons brièvement les points suivants :

1. La méthodologie d'entraînement du modèle.
2. La fonction coût métier, l'algorithme d'optimisation, et la métrique d'évaluation.
3. L'interprétation globale et locale du modèle.
4. Les limites identifiées et les pistes d'amélioration envisageables.

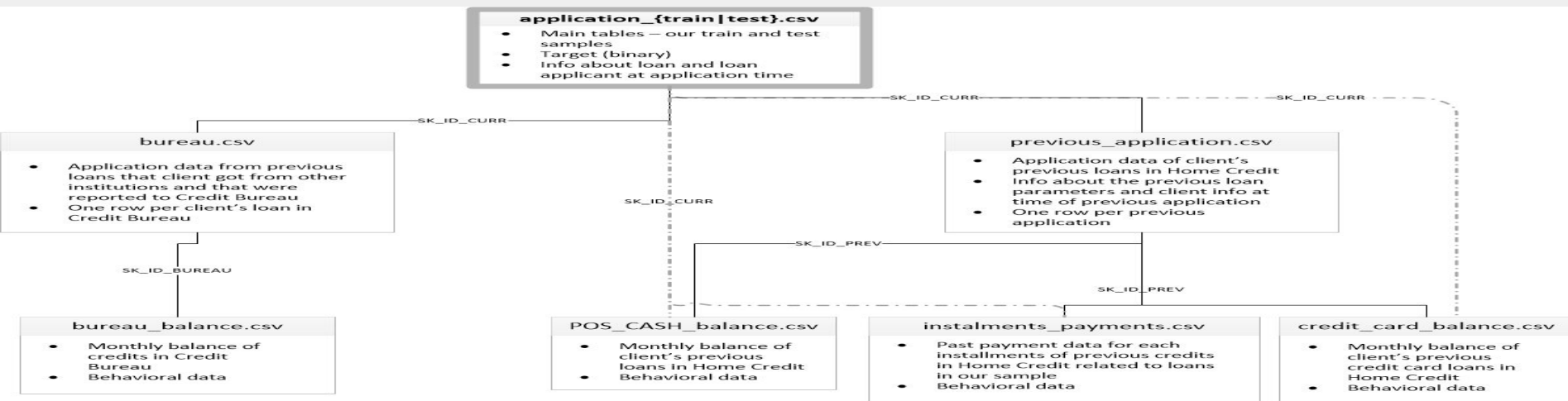
# Les étapes préalables à la modélisation


Les données sont réparties dans sept fichiers interconnectés selon le schéma ci-dessous. La table "application" consolide les informations personnelles des clients actuels ainsi que les détails liés aux crédits qu'ils sollicitent.

Cette table est segmentée en deux ensembles de données distincts : l'application "train" regroupe **307 511 clients** pour lesquels la décision de "Prêt à Dépenser" sur l'octroi du crédit est connue (variable "Target"). En parallèle, l'application "test" rassemble des données pour lesquelles cette décision est inconnue.

Les autres fichiers renferment les données historiques de crédit pour ces mêmes clients :

- Les tables "bureau" et "balance bureau" comprennent les informations relatives aux crédits antérieurs obtenus auprès d'institutions autres que "Prêt à Dépenser".
- La table "Previous\_application" compile les données des crédits précédents accordés par "Prêt à Dépenser".





Bien que l'on puisse penser qu'un grand volume de données est suffisant pour obtenir un algorithme performant, les données dont nous disposons ne sont souvent pas directement adaptées. Il est donc nécessaire de les traiter préalablement, ce qui constitue l'étape de prétraitement.

En effet, des erreurs d'acquisition, qu'elles soient imputables à des erreurs humaines ou techniques, peuvent altérer notre ensemble de données et introduire des biais dans le processus d'entraînement. Parmi ces erreurs, citons les informations incomplètes, les valeurs manquantes ou incorrectes, ainsi que les interférences parasites associées à l'acquisition des données. Par conséquent, l'établissement d'une stratégie de prétraitement des données, également appelée preprocessing des données, à partir de nos données brutes est souvent essentiel. Cela nous permet d'obtenir des données exploitées qui contribueront à la création d'un modèle plus performant.

## Data Cleaning

La première étape consiste à procéder au nettoyage des données incorrectes, incomplètes, ou manquantes. Cette procédure a été effectuée de manière indépendante pour chaque table.

Dans un premier temps, nous avons éliminé toutes les variables comportant plus de 0,1 % de valeurs manquantes. Ensuite, pour les variables présentant moins de 0,1 % de données manquantes, nous avons opté pour leur imputation. Les données numériques ont été complétées en utilisant la médiane, tandis que pour les variables catégorielles, nous avons substitué les valeurs manquantes par la modalité la plus fréquente, également appelée le mode.



## Data Transformation

Cette étape de prétraitement englobe les modifications apportées à la structure intrinsèque des données. Ces transformations sont alignées sur les définitions mathématiques des algorithmes et la manière dont ces derniers traitent les données, le tout dans le but d'optimiser les performances.

Nous avons principalement utilisé deux techniques :

- **Discrétisation des variables continues** : Cette méthode, réalisée par le découpage en intervalles, vise à réduire le nombre de modalités d'une variable tout en éliminant les valeurs aberrantes potentielles.
- **Normalisation et standardisation des données** : Ces approches visent à ramener les données numériques à une échelle plus restreinte (par exemple, entre -1 et 1). Elles permettent également de centrer la moyenne et de réduire la variance des données.



## Data Reduction

**Lors de la création** d'un modèle prédictif, notre ensemble de données est souvent composé de nombreuses fonctionnalités ou variables. Bien que ces fonctionnalités soient présentes dans nos données, cela ne garantit pas leur pertinence pour notre modèle, et il est donc crucial de choisir judicieusement celles à inclure.

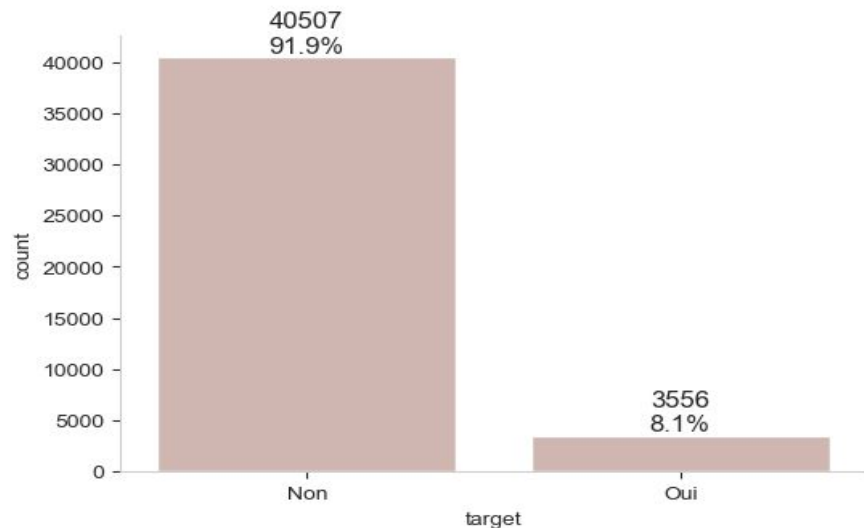
C'est là que la sélection des caractéristiques entre en jeu. Il s'agit simplement d'un processus visant à réduire le nombre de variables d'entrée pour ne conserver que les plus importantes.

Il existe trois catégories de méthodes de sélection de fonctionnalités, en fonction de la manière dont elles interagissent avec la variable à prédire, à savoir les méthodes de filtrage, d'encapsulation et intégrées. Nous avons recouru aux trois catégories de méthodes pour affiner notre modèle.

*En fin de compte, nous disposons d'un ensemble de données d'entraînement comprenant 307 511 clients, avec 56 variables descriptives.*

### *Le client est-il en difficulté de paiement ?*

*A-t-il eu un retard de paiement de plus de X jours sur au moins une des Y premières échéances du crédit ?*

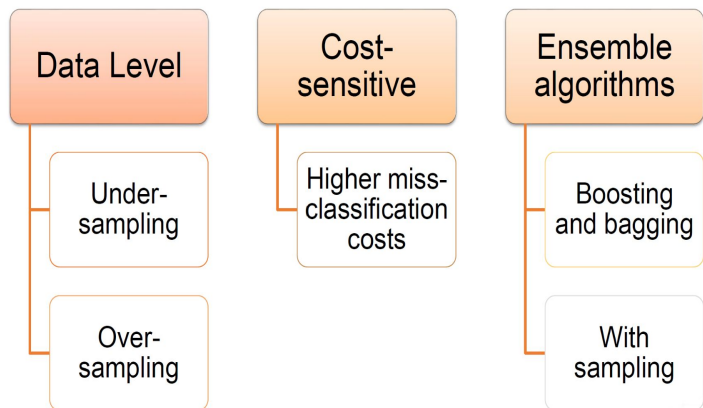




# La méthodologie d'entraînement du modèle

Les clients en situation de difficulté de paiement sont nettement sous-représentés, ne représentant que 8,1 % des données d'entraînement. Cependant, les méthodes de machine learning classiques ne sont pas toujours adaptées à la **classification sur des ensembles de données déséquilibrés**. Elles produisent souvent des résultats médiocres et, dans certains cas, peuvent même être trompeuses en fournissant des scores excessivement optimistes. L'un des facteurs contribuant à ces échecs est que les points de la classe minoritaire sont parfois perçus comme des valeurs aberrantes ne contenant aucune information.

## Solutions for Imbalanced Datasets



Les méthodes pour surmonter le déséquilibre des classes se répartissent en deux catégories distinctes.

- D'une part, les méthodes au niveau des données (data-level) impliquent des transformations appliquées aux données d'entraînement.
- D'autre part, les méthodes au niveau de l'algorithme (algorithm-level), comprenant les approches coût-sensibles et les algorithmes d'ensemble, reposent sur des modifications apportées aux modèles afin de les rendre plus adaptés à la résolution de ce problème.

## ✓ Méthodes au niveau des données (data-level)

Les approches data-level partagent une idée fondamentale : transformer les données d'entraînement du modèle pour atténuer le déséquilibre. On recourt fréquemment à des techniques d'échantillonnage pour augmenter la représentation de la classe minoritaire et/ou réduire celle de la classe majoritaire.

### **Sous-échantillonnage aléatoire :**

La première approche consiste à sous-échantillonner la classe majoritaire en réduisant sa taille de manière aléatoire. Les points sont retirés de manière naïve, simplement de manière aléatoire.

### **Tomek Links :**

Une approche plus élaborée de sous-échantillonnage de la classe majoritaire utilise les Tomek Links. Cette méthode propose une sélection plus fine des points à éliminer. Elle identifie les points de la classe majoritaire proches d'un point de la classe minoritaire (Tomek Link). Dans chaque Tomek Link, le point de la classe majoritaire est retiré. L'identification des Tomek Links s'effectue de manière similaire à celle des plus proches voisins dans l'algorithme KNN. Cette approche s'avère plus efficace que le sous-échantillonnage aléatoire, réduisant la variance de la classe majoritaire et éliminant d'éventuels outliers, sources potentielles de confusion.

### **SMOTE & ADASYN :**

D'autres techniques courantes d'échantillonnage incluent SMOTE (Synthetic Minority Oversampling Technique, ou suréchantillonnage minoritaire synthétique) et ADASYN (ADaptive SYNthetic sampling, ou échantillonnage synthétique adaptatif). Ces méthodes sont cette fois des techniques de suréchantillonnage. Au lieu de réduire la taille de la classe majoritaire, l'objectif est d'augmenter celle de la classe minoritaire. Pour ce faire, des points de la classe à augmenter sont sélectionnés, et de nouveaux points sont générés.

**Il est également important de noter que certains algorithmes sont moins sensibles au problème des données déséquilibrées. Les arbres de décision, les forêts aléatoires et le gradient boosting fonctionnent souvent bien avec de tels ensembles de données.**



## Méthodes au niveau de l'algorithme (algorithm-level)

La deuxième catégorie d'approches englobe les méthodes dites "algorithm-level", qui impliquent des adaptations des modèles classiques de machine learning pour mieux traiter le déséquilibre.

### **Apprentissage sensible aux coûts :**


Une méthode algorithm-level possible consiste à attribuer un poids plus important à la classe minoritaire, également connue sous le nom d'apprentissage sensible aux coûts. Concrètement, cela signifie spécifier à notre modèle que la classification correcte d'un point de la classe minoritaire est plus cruciale que celle d'un point de la classe majoritaire. Ainsi, une erreur de classification sur la classe minoritaire est considérée comme plus grave par le modèle, incitant l'algorithme à accorder une plus grande importance à cette classe et à améliorer les prédictions correspondantes.

### **Apprentissage à une classe :**

Une autre approche pouvant offrir de bons résultats dans certains cas est l'apprentissage à une classe. Plutôt que de former un modèle sur les deux classes, un détecteur est entraîné sur la classe majoritaire. Ce détecteur peut ensuite prédire si un point appartient ou non à cette classe. Bien que les modèles d'apprentissage à une classe ne soient pas spécifiquement conçus pour contrer le déséquilibre des classes, ils peuvent fournir d'importantes améliorations de performances.

Avant d'appliquer ces différentes méthodes pour remédier au déséquilibre des classes, le choix du type de modèle de prédiction le plus adapté à nos données est crucial.

Pour ce faire, nous avons divisé notre jeu d'entraînement en deux : un sous-échantillon d'entraînement représentant 70 % des données et un sous-échantillon de validation. Chaque sous-échantillon maintient un mélange équilibré d'exemples par classe, soit environ 92 % de la classe 0 et 8 % de la classe 1. Sur le sous-échantillon d'entraînement, nous avons évalué les modèles candidats à l'aide d'une validation croisée stratifiée répétée k-fold.



La **procédure de validation croisée k-fold** offre une estimation générale de la performance du modèle qui est relativement moins biaisée par l'optimisme par rapport à une seule séparation train-test.

Nous avons opté pour **k = 10**, ce qui implique que chaque pli contient environ **9 225 échantillons (clients)**, résultant de la division de **92 254 par 10**.

L'utilisation du terme "stratifié" indique que chaque pli maintient un mélange identique d'exemples par classe, avec environ **92 % de la classe 0 et 8 % de la classe 1**.

En ce qui concerne le terme "répété", il souligne que le processus d'évaluation est répété plusieurs fois pour atténuer les résultats aléatoires et mieux capturer la variance du modèle. Dans notre cas, nous avons effectué trois répétitions, ce qui signifie qu'un seul modèle sera ajusté et évalué **30 fois (10 × 3)**. La moyenne et l'écart type de ces essais seront ensuite calculés.

**Nous avons évalué sept modèles distincts**, comprenant un classificateur naïf (**Dummy**), une régression logistique, deux classificateurs basés sur la forêt aléatoire, un modèle de Support Vector Machine, ainsi que deux classificateurs utilisant des méthodes de descente de gradient : le classificateur de la librairie **LightGBM** et celui de la librairie **XGBoost**.

Le classificateur naïf (**DummyClassifier**) génère des prédictions ignorant les entités d'entrée et sert de référence simple pour la comparaison avec les autres classificateurs plus complexes.

# La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation



## Choisir judicieusement les métriques pour évaluer les performances des modèles

En machine learning, la sélection appropriée des métriques de performance revêt une importance cruciale, une nécessité d'autant plus fondamentale lorsqu'on travaille avec des données déséquilibrées. Bien que cela ne résolve pas directement le problème, le choix adéquat des métriques évite des évaluations trompeuses.

Dans le contexte de la classification binaire, l'usage courant consiste à recourir au pourcentage de bonnes prédictions comme métrique. Toutefois, ce pourcentage peut être élevé même si une grande partie des points de la classe minoritaire est mal classifiée. De plus, cette métrique peut être biaisée par le déséquilibre des classes, pouvant donner une impression erronée de la qualité du modèle, particulièrement s'il est efficace uniquement pour la classe majoritaire. Pour surmonter ce problème, il est préférable de se tourner vers des métriques moins sensibles à la mauvaise répartition des classes.

**Le recall (sensibilité) ou le F-Score sont des exemples pertinents dans ce contexte.**

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
		<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$

## La matrice de confusion dans la classification

Dans le domaine de la classification, la **matrice de confusion** représente l'indicateur principal de la performance d'un modèle. Il s'agit d'un tableau à double entrée qui établit une correspondance entre les classes réelles et les classes prédites par le modèle. La matrice de confusion constitue la base de tous les calculs d'indicateurs de performance.

Comme l'accuracy ne fait pas de distinction entre les types d'erreurs commises, elle est souvent complétée par deux indicateurs : le rappel (recall en anglais) et la précision.

**Le rappel** représente la proportion de la classe positive détectée, et il varie entre 0 et 1. Un rappel élevé indique que la plupart des cas de la classe positive ont été correctement détectés, par exemple, que la majorité des patients réellement malades ont été classés comme tels. Un rappel fort est associé à moins d'erreurs de type II, qui sont généralement les plus graves.

**La précision**, quant à elle, représente la proportion de vrais positifs parmi tous les positifs détectés et varie également entre 0 et 1. Elle permet d'évaluer la proportion réelle de positifs parmi les cas identifiés comme tels par le modèle, aidant à estimer le nombre d'erreurs de type I (faux positifs).

**En fonction des exigences métier**, l'un de ces deux indicateurs peut avoir un impact plus significatif que l'autre. Il est donc pertinent de choisir un modèle en fonction du rappel et/ou de la précision plutôt que de l'accuracy.

**Le F1-score** (ou "score F1") vise à fournir un indicateur unique qui prend en compte à la fois le rappel et la précision, tout en évitant les pièges de l'accuracy. Sa formule est la moyenne harmonique de la précision et du rappel, soit :  $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$ . Sa valeur varie de 0 à 1, avec un score de 1 indiquant une précision et un rappel de 100 %. Étant donné que les erreurs de type I ou de type II n'ont généralement pas le même impact, il est possible de calculer le F $\beta$ -score en ajustant la valeur de  $\beta$  pour refléter le coût relatif des erreurs de type II par rapport aux erreurs de type I. Le symbole  $\beta$  est souvent remplacé par la valeur relative du rappel par rapport à la précision.

Le score F1 est un cas particulier où le rappel et la précision ont la même importance.

**Un score F2**, par exemple, indique que le rappel a deux fois plus d'impact que la précision, soulignant qu'une erreur de type II est deux fois plus coûteuse qu'une erreur de type I.

## Choix de la fonction de coût

La focalisation de cette tâche est orientée vers la classe positive (clients éprouvant des difficultés de paiement), avec la précision et le rappel comme points de départ.

- Maximiser la précision permet de minimiser les faux positifs, c'est-à-dire lorsque le client n'a pas de difficultés de paiement mais est prédit comme tel. Ceci évite une perte de client pour la société de crédit (erreur de Type I).
- Maximiser le rappel permet de minimiser les faux négatifs, c'est-à-dire lorsque le client a des difficultés de paiement mais est prédit comme n'en ayant pas. Ceci évite une perte de chiffre d'affaires pour la société de crédit (erreur de Type II).

La complexité réside dans le fait que **les faux négatifs sont plus préjudiciables que les faux positifs**. Sur cet ensemble de données, les faux négatifs se traduisent par des cas où un mauvais client est considéré comme un bon client et obtient un prêt, tandis que les faux positifs sont des cas où un bon client est perçu comme un mauvais client, et la société de crédit lui refuse un prêt. Les faux négatifs représentent donc un coût plus élevé : **Coût(Faux Négatifs) > Coût(Faux Positifs)**.

En d'autres termes, l'attention se porte sur la **F-mesure**, qui résume la capacité d'un modèle à minimiser les erreurs de classification pour la classe positive, en **privilégiant les modèles qui réduisent davantage les faux négatifs que les faux positifs**. Cela peut être accompli en utilisant une version de la F-mesure qui calcule une moyenne harmonique pondérée de la précision et du rappel, mais qui donne plus de poids aux scores de rappel qu'aux scores de précision. C'est ce qu'on appelle la mesure **F-beta**, une généralisation de la F-mesure, où beta est un paramètre déterminant le poids respectif des deux scores.

**La valeur de beta dépend ainsi indirectement de la fonction de coût que nous souhaitons appliquer à chacun des deux types d'erreurs : les faux positifs et les faux négatifs.**

**Un beta de 2 accorde plus d'importance au rappel qu'à la précision, ce qui donne naissance à la mesure F2.**

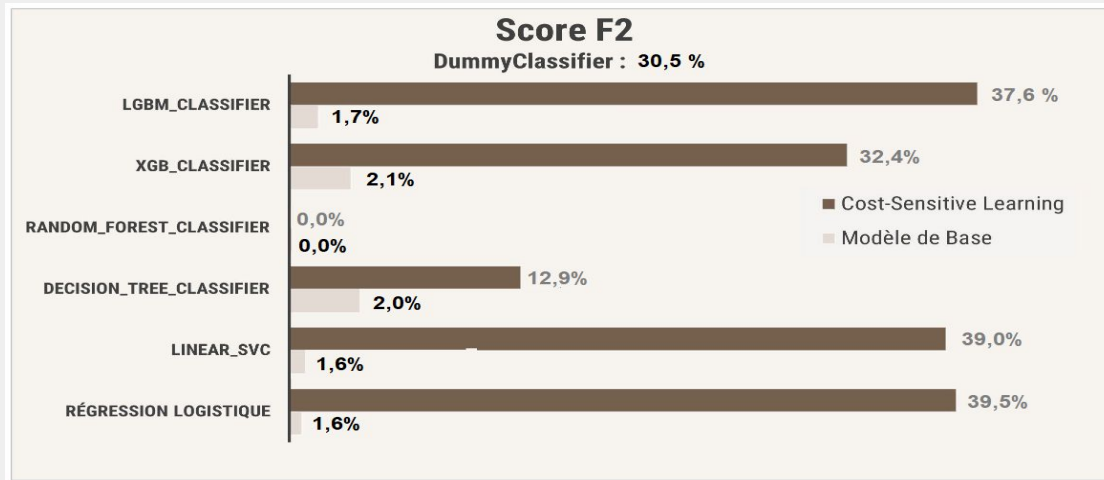


## Prise en compte du déséquilibre des classes : Apprentissage Sensible aux Coûts

L'apprentissage sensible aux coûts vise à minimiser le coût d'un modèle sur l'ensemble de données d'apprentissage en supposant que différents types d'erreurs de prédiction ont une valeur de coût associée différente et connue. **Il existe ainsi un lien étroit entre la classification déséquilibrée et l'apprentissage sensible aux coûts.**

**Plus précisément, un problème d'apprentissage déséquilibré peut être résolu efficacement en utilisant des techniques d'apprentissage sensible aux coûts.** Contrairement aux méthodes de suréchantillonnage et de sous-échantillonnage, les méthodes de pondération équilibrée ne modifient pas le rapport entre les classes minoritaires et majoritaires. Au lieu de cela, elles pénalisent les erreurs de prédiction sur la classe minoritaire en accordant plus de poids à la fonction de perte.

Un point de départ judicieux pour les tâches de classification déséquilibrées consiste à **attribuer des coûts basés sur la distribution inverse des classes.**



Nous constatons que trois des modèles testés affichent une mesure F2 supérieure à la valeur par défaut de prédiction de la classe majoritaire dans tous les cas (30,5%).

Les trois meilleurs scores F2 ont été obtenus avec les modèles **LightGBM (41%)**, suivi de la régression logistique (39,5%) et du LinearSVC (38,9%).

Les deux meilleurs modèles ont ensuite été réentraînés en utilisant différentes méthodes de gestion du déséquilibre, notamment l'échantillonnage par RandomOverSampler, SMOTE, ADASYN et BorderlineSMOTE. Cependant, ces techniques n'ont pas amélioré les résultats par rapport à l'approche du "Cost-Sensitive Learning". Un inconvénient général de cette approche est que des exemples synthétiques sont créés sans tenir compte de la classe majoritaire, pouvant conduire à des exemples ambigus en cas de fort chevauchement entre les classes.

**Le modèle final sélectionné est le Light Gradient Boosted Machine, ou LightGBM en abrégé.**

### Optimisation de l'algorithme LightGBM :

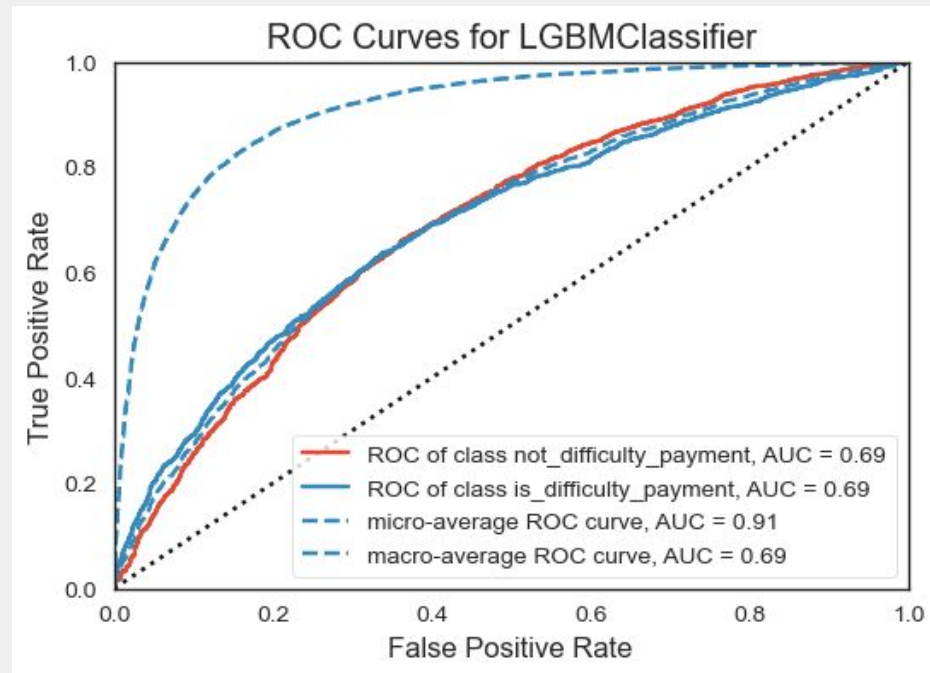
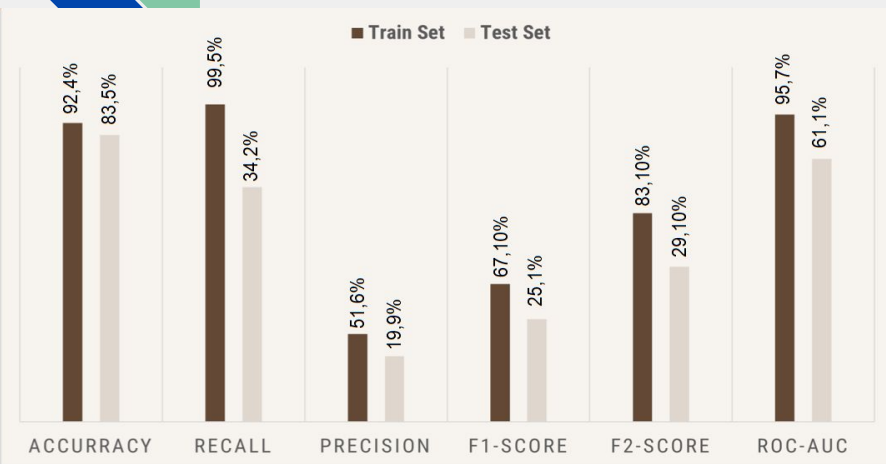
Il existe de nombreux hyperparamètres à examiner pour LightGBM, regroupés en quatre catégories :

- Paramètres qui affectent la structure et l'apprentissage des arbres de décision.
- Paramètres qui affectent la vitesse d'entraînement.
- Paramètres pour une meilleure précision.
- Paramètres pour lutter contre le surajustement.

Ces catégories se chevauchent souvent, et **ajuster manuellement un paramètre peut affecter d'autres aspects**. Ainsi, l'utilisation de frameworks tels qu'**Optuna**, qui peuvent trouver le juste équilibre entre ces catégories en automatisant le processus, est préférable.

Nous avons réalisé 300 essais avec comme objectif la minimisation de la perte logarithmique (Log-loss), où une probabilité de prédiction éloignée de la valeur réelle entraîne une perte logarithmique plus élevée.

Les paramètres optimisés nous procurent les performances suivantes :



Le F2-score a progressé de 40,7 % à 52,9 % sur notre ensemble d'entraînement.

# L'interprétation globale et locale du modèle



L'interprétation des modèles revêt une importance cruciale en apprentissage automatique, en raison notamment de la nécessité d'expliquer et de justifier les décisions prises par le modèle prédictif dans de nombreux domaines.

La question de l'identification des variables pertinentes demeure essentielle, ne serait-ce que pour comprendre le mécanisme d'affectation sous-jacent au modèle. L'importance des variables mesure l'impact global de chaque descripteur dans le modèle.

Cette importance peut être évaluée soit lors de la modélisation sur l'échantillon d'apprentissage, soit lors de la prédiction sur l'échantillon de test.

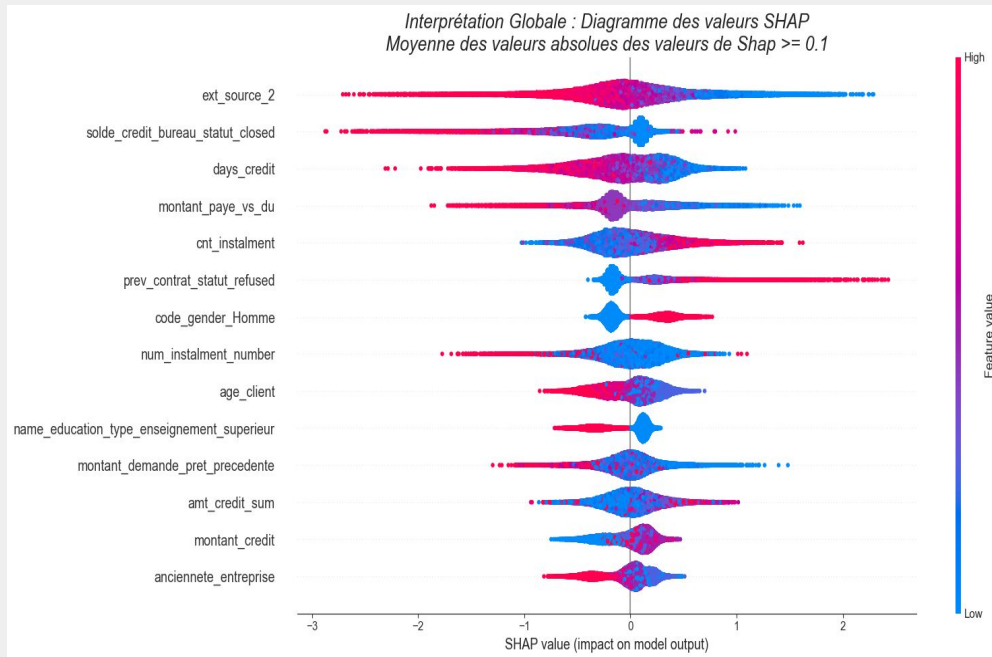
Dans les deux cas, les étapes principales demeurent les mêmes :

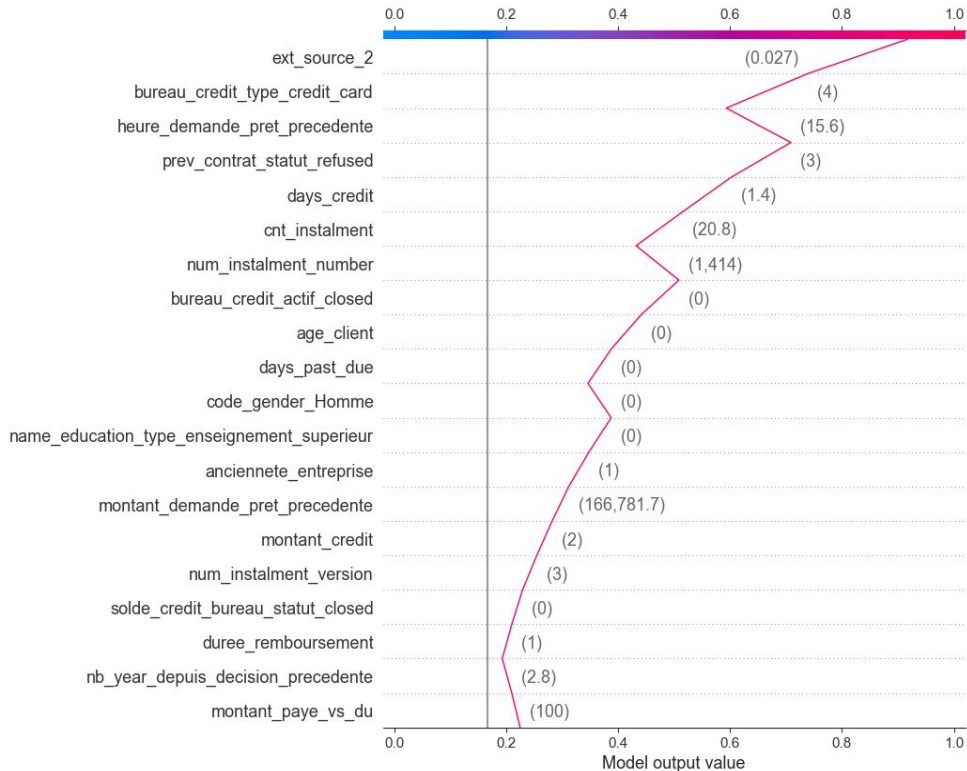
- Calculer le taux d'erreur de référence.
- Calculer ensuite le même indicateur en neutralisant tour à tour chaque variable prédictive.
- Former le ratio entre les deux valeurs.

L'échantillon d'apprentissage met en lumière le rôle des variables dans le processus de modélisation, tandis que l'échantillon de test éclaire sur l'influence de la variable sur les performances en généralisation, c'est-à-dire lorsque le modèle est déployé dans la population.

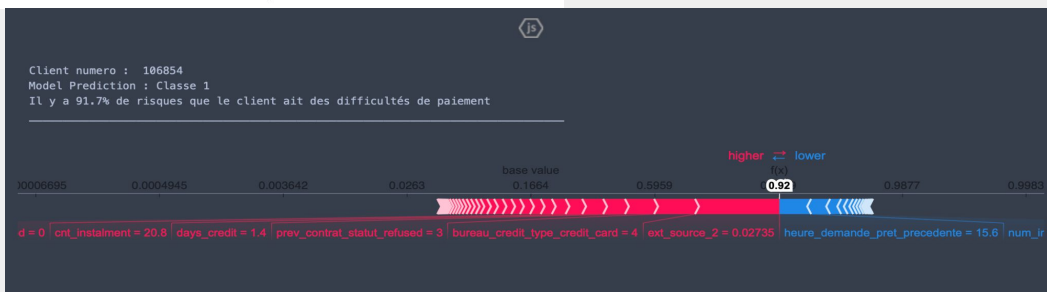
**Les valeurs de Shapley** calculent l'importance d'une variable en comparant les prédictions du modèle avec et sans cette variable. Toutefois, étant donné que l'ordre dans lequel un modèle prend connaissance des variables peut influencer ses prédictions, cette comparaison est effectuée dans tous les ordres possibles. Ainsi, les fonctionnalités sont équitablement évaluées, suivant une approche inspirée de la théorie des jeux.

**L'intelligibilité globale** vise à expliquer le modèle dans son ensemble, c'est-à-dire à identifier quelles sont les variables généralement les plus importantes pour le modèle. Par exemple, quelles caractéristiques influent sur le comportement global d'un modèle d'allocation de prêt ?





En revanche, **l'intelligibilité locale** consiste à expliquer la prévision  $f(x)$  d'un modèle pour un individu  $x$  spécifique. Par exemple, pourquoi la demande de prêt d'un client a-t-elle été approuvée ou rejetée ?



## MLflow :

l'intégration de **MLflow** dans notre processus de développement de modèles machine learning a considérablement amélioré la gestion et la traçabilité de nos expériences. **MLflow** offre une solution complète pour suivre et organiser chaque étape du cycle de vie des modèles, de l'expérimentation initiale à la mise en production.



Nous avons démontré comment **MLflow** facilite le suivi des hyperparamètres, la sauvegarde des métriques de performance telles que la précision, et l'enregistrement transparent des modèles avec toutes les informations nécessaires. La signature automatique du modèle et la possibilité de le charger ultérieurement ont simplifié la gestion des versions des modèles.

En utilisant **MLflow**, nous avons pu créer une expérience cohérente et reproductible, ce qui est essentiel pour la collaboration au sein de l'équipe. La transparence dans la gestion du cycle de vie des modèles nous a permis de prendre des décisions plus éclairées, de itérer rapidement sur les modèles, et d'assurer la traçabilité complète de chaque exécution.

En tant qu'outil open source robuste, **MLflow** a démontré sa valeur en tant que plateforme de gestion des modèles, et son intégration dans notre flux de travail continuera à jouer un rôle central dans nos futurs projets machine learning. Son extensibilité et sa facilité d'utilisation en font un choix optimal pour les équipes cherchant à améliorer leur pipeline de modélisation et de déploiement.




# Les limites et les améliorations possibles

Une première limite dans ce projet découle de ma **méconnaissance du milieu bancaire** et de la subtilité du vocabulaire spécifique. Mes sélections de variables ont été effectuées en fonction de ma compréhension des données et de leur impact potentiel, mais sans nécessairement appréhender pleinement les implications réelles. Il est ainsi possible que des choix inappropriés aient été faits lors de la sélection et de l'agrégation des variables, du point de vue métier.

Il aurait été bénéfique de disposer d'un véritable répertoire des variables et de pouvoir confronter ces choix avec l'expertise de « Prêt à dépenser » afin de **garantir la cohérence du prétraitement**.

De plus, **le choix et l'optimisation du modèle** de classification ont été effectués en se basant sur une hypothèse robuste concernant **la métrique d'évaluation** : le F Beta Score avec un Beta fixé à 2, suivant certaines hypothèses non confirmées par le domaine métier. Une amélioration majeure serait **de définir plus précisément la métrique d'évaluation et la fonction de coût en étroite collaboration avec les équipes métier**.

À cet égard, il aurait été essentiel d'engager des discussions avec « Prêt à dépenser » sur les points cruciaux de la métrique. En effet, il existe un compromis nécessaire entre le nombre de faux positifs et de faux négatifs. Le choix du seuil final d'acceptation ou de refus de crédit joue un rôle significatif dans ce compromis, car augmenter le seuil tend à accroître le nombre de crédits refusés, augmentant ainsi les faux positifs, tandis que le baisser a l'effet inverse. Ce seuil aurait dû être discuté et défini en collaboration avec le client, éventuellement en lien avec une analyse financière des pertes dues aux erreurs d'attribution.



Enfin, pour améliorer le **Dashboard**, il serait pertinent de développer une interface avec une page dédiée à la « banque » et une autre à « l'utilisateur ». Cette approche offrirait à l'équipe de « **Prêt à dépenser** » la possibilité d'accéder à des données spécifiques pour expliquer une décision à un client, sans toutefois révéler l'ensemble des informations sensibles. De plus, l'ajout d'une section interactive pourrait permettre au client d'explorer l'influence de différentes valeurs sur sa demande de crédit. Une page « **scénario** » pourrait ainsi être envisagée, offrant la possibilité de modifier une ou plusieurs variables du profil du client pour visualiser l'impact sur la réponse de la banque.

[CLIQUEZ ICI POUR VOUS RENDRE SUR LE DASHBOARD](#)