



Projet de Scoring

Sara LAVAL-JEANTET
Chahla TARMOUN
Minh Nhat Thy HUYNH
Salma BENMOUSSA

Encadré par : M. Ibrahim TOURE

Master 2 Modélisations statistiques économiques et financières

Année Scolaire : 2024-2025

Table des matières

1	Introduction	1
2	Présentation des données	2
3	Analyse exploratoire des données	2
3.1	Analyse univariée	2
3.1.1	Distribution des variables catégorielles	2
3.1.2	Exploration des distributions et hypothèses initiales	2
3.2	Analyse Bivariée	3
3.2.1	Analyse des distributions conditionnelles	3
3.2.2	Les interactions complexes : variables et taux de défaillance	3
3.2.3	Vers une compréhension plus fine des relations	4
4	Comment trouver des modèles potentiels ?	6
4.1	Objectif	6
4.2	Comment identifier les modèles potentiels ?	6
4.2.1	Méthode 1 : Prédiction des modèles potentiels en se basant sur l'analyse du chevauchement (overlapping) à travers des pairplots	6
4.2.2	Méthode 2 : Prédiction des modèles potentiels en utilisant la bibliothèque LazyPredict	8
5	Modélisation avec les modèles choisis	10
5.1	Régression Logistique	10
5.1.1	Prétraitement Général aux Modèles	10
5.1.2	Prétraitement Spécifique aux Modèles	11
5.2	XGBoost	15
5.2.1	XGBoost version 1 : Prétraitement optimisé avec GridSearchCV	15
5.2.2	XGBoost Version 2 : Prétraitement optimisé avec GridSearchCV combiné à SMOTE	16
5.2.3	XGBoost Version 3 : Prétraitement optimisé avec GridSearchCV combiné à l'utilisation de class-weight	16
5.3	LightGBM	17
5.3.1	LightGBM version 1 : Prétraitement optimisé avec GridSearchCV	17
5.3.2	LightGBM Version 2 : Prétraitement optimisé avec GridSearchCV combiné à SMOTE	18
5.3.3	LightGBM Version 3 : Prétraitement optimisé avec GridSearchCV combiné à l'utilisation le class-weight	18
6	Évaluation et comparaison des modèles	19
6.1	Régression logistique	19
6.2	Modèles Gradient Boosting	20
7	Grille de score	20
8	Conclusion	21
A	Annexe 1 : Analyse descriptive	i
A.1	Analyse catégorielle	i
A.2	Visualisations des distributions	ii
A.3	Taux de défaillance et caractéristiques	ii
A.4	Taux de défaillance : lissage par moyenne mobile	iv

B Annexe 2 : Pre-processing	vi
C Annexe 3 : Analyse des features importantes dans les modèles Gradient Boosting	vi

1 Introduction

Dans le contexte économique actuel, la gestion du risque de crédit est devenue l'une des préoccupations majeures pour les banques et les autres institutions financières. L'octroi de crédits, en particulier ceux garantis par l'équité domiciliaire, nécessite une analyse rigoureuse des comportements de remboursement des emprunteurs. L'objectif principal de ce rapport est de développer un modèle prédictif permettant d'identifier les caractéristiques influençant la probabilité de défaut de paiement sur des prêts hypothécaires garantis par l'équité domiciliaire.

Afin d'atteindre cet objectif, nous utiliserons un ensemble de données contenant des caractéristiques et des informations de délinquance sur des prêts hypothécaires. En étudiant les relations entre la variable cible, mesurée par si un emprunteur a fait défaut ou non, et diverses caractéristiques disponibles, nous chercherons à déterminer les composants du risque de défaut. Notre analyse reposera sur des modèles statistiques tels que la régression logistique, ainsi que d'autres techniques avancées de modélisation.

Le rapport sera divisé en plusieurs parties, chacune correspondant à des aspects particuliers de l'analyse : la présentation des données, l'analyse exploratoire des données, le prétraitement et la modélisation de données ainsi que l'évaluation des performances de ces modèles. Le choix de ces analyses statistiques offrira aux banques une meilleure compréhension de la situation et une vision réaliste des conséquences qui en découleront.

En fin de compte, nous élaborerons une grille de score qui permettra d'apporter des suggestions pertinentes pour les décideurs financiers sur l'octroi de crédits en vue d'une gestion appropriée des risques dans les institutions financières.

2 Présentation des données

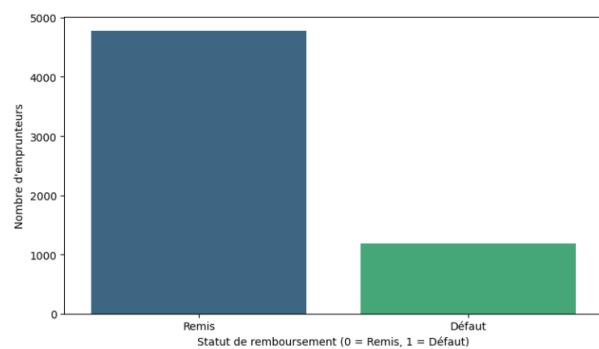
Le jeu de données utilisé pour cette analyse est le **HMEQ**, qui regroupe des caractéristiques sur 5 960 prêts hypothécaires garantis par l'équité domiciliaire. Chaque ligne du jeu de données représente un prêt et les colonnes fournissent diverses informations pertinentes pour l'évaluation du risque de crédit.

3 Analyse exploratoire des données

3.1 Analyse univariée

3.1.1 Distribution des variables catégorielles

Distribution de la variable cible



Nous observons un déséquilibre marqué entre les classes, avec une majorité d'emprunteurs remboursant leur prêt ($TARGET = 0$). Ce déséquilibre, présent dans les jeux d'entraînement et de test (voir Annexe A.1), peut poser des défis pour la modélisation, car le modèle pourrait ignorer les cas de défaillance. Des stratégies d'ajustement seront donc nécessaires pour gérer ce biais.

FIGURE 1 – Distribution de la variable cible

Distribution des raisons et catégories professionnelles

Concernant la distribution des métiers (voir Annexe A.1), la catégorie "Other" domine avec près de 40 %, ce qui réduit la granularité des données. Des métiers comme "Mgr" ou "Sales" sont présents mais sous-représentés, limitant la capacité du modèle à capturer des différences spécifiques. Les prêts pour la consolidation de dettes sont plus fréquents que ceux pour l'amélioration de la maison (voir Annexe A.1), suggérant une plus grande vulnérabilité financière chez les emprunteurs concernés. Enfin, la similarité des distributions retrouvées dans l'annexe A.1 entre les jeux de test et d'entraînement assure une bonne représentativité des données.

3.1.2 Exploration des distributions et hypothèses initiales

Avant de passer à l'analyse globale, il est essentiel de bien comprendre les distributions des variables pour déterminer les ajustements nécessaires avant la modélisation (voir Annexe A.2).

Montant du prêt : un indicateur de risque potentiel La distribution montre que la majorité des emprunts sont inférieurs à 30 000, alors qu'une petite proportion contracte des prêts plus élevés. Hypothèse : les emprunteurs ayant des montants plus importants présentent potentiellement un risque moindre, ces montants reflétant possiblement des revenus ou des garanties plus élevés.

Valeur de la propriété et montant dû : protection ou vulnérabilité La plupart des emprunteurs possèdent des biens modestes, mais les distributions révèlent une proportion notable de propriétés de grande valeur. Ces dernières pourraient offrir une protection contre le défaut, bien que les grandes hypothèques puissent exposer certains emprunteurs à des risques similaires à ceux ayant des biens de moindre valeur.

Ancienneté dans l'emploi : un facteur de stabilité ? La distribution d'ancienneté au travail montre une forte concentration d'emprunteurs avec moins de 10 ans, ainsi qu'un groupe significatif au-delà de 20 ans. Hypothèse : une ancienneté élevée est souvent un facteur de stabilité, bien que cela puisse dépendre du secteur.

Rapports dérogatoires et défauts de paiement : des signaux de vulnérabilité La majorité des emprunteurs ne présente pas de rapports dérogatoires ni de lignes de crédit en défaut, mais ceux ayant des antécédents font face à un risque de défaut élevé, renforçant l'hypothèse d'un risque accru pour ces profils.

Ancienneté et nombre de lignes de crédit : gestion ou risque accru ? La distribution montre que la plupart des emprunteurs ont une longue expérience de crédit, mais une minorité gère un nombre élevé de lignes, au-delà de 50. Cette gestion agressive pourrait augmenter leur risque de défaut en raison de la dette cumulée.

3.2 Analyse Bivariée

3.2.1 Analyse des distributions conditionnelles

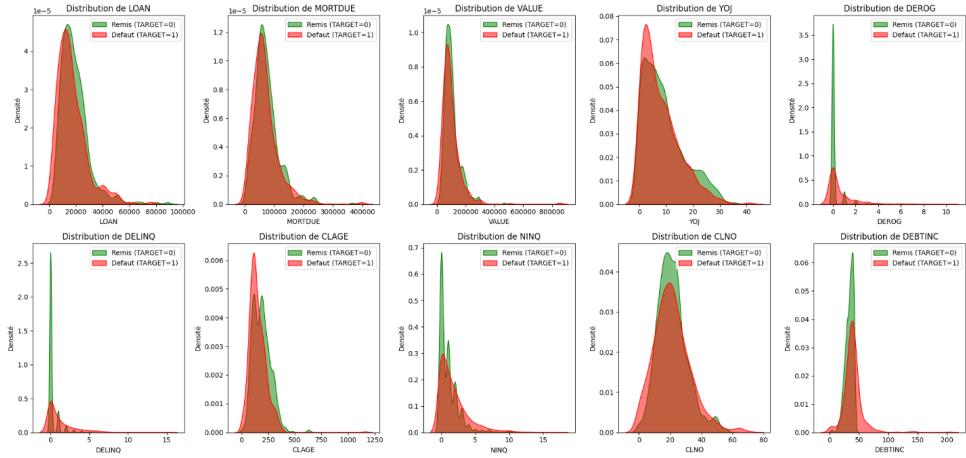


FIGURE 2 – Distributions conditionnelles selon la variable cible

L'analyse des distributions selon la variable cible révèle que les emprunteurs en défaut présentent fréquemment un grand nombre de rapports dérogatoires, ce qui témoigne de difficultés financières antérieures. De plus, ceux qui ont des lignes de crédit plus récentes semblent plus exposés au risque de défaillance, contrairement à ceux avec un historique de crédit plus long, qui paraissent mieux gérer leurs dettes. Enfin, un ratio dette/revenu élevé souligne une pression financière immédiate chez les emprunteurs en difficulté.

3.2.2 Les interactions complexes : variables et taux de défaillance

Maintenant que nous avons examiné les distributions des variables du modèle, il est temps de passer à la deuxième phase de l'analyse : croiser ces distributions avec les taux de défaillance observés. C'est ici que les hypothèses émises précédemment vont être testées et nuancées, en prenant en compte des interactions plus complexes entre les variables.

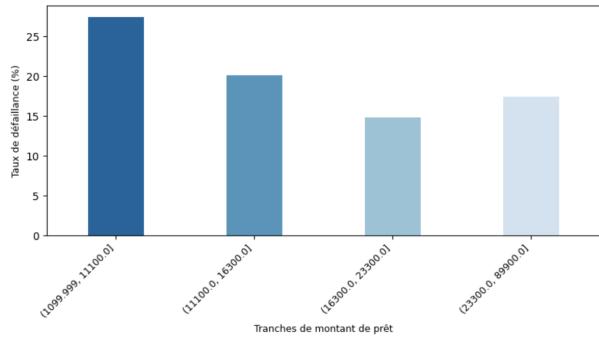


FIGURE 3 – Taux de défaillance par tranche de montant de prêt

Montant du prêt et taux de défaillance : tendance confirmée ? Les petits emprunteurs ont un taux de défaillance élevé tandis que ceux empruntant entre 16 000 et 23 000 sont plus stables. Cependant, pour les montants au-delà, le risque augmente à nouveau. Cette relation non linéaire montre que les montants moyens sont plus sûrs, alors que les emprunts très bas ou très élevés sont plus risqués.

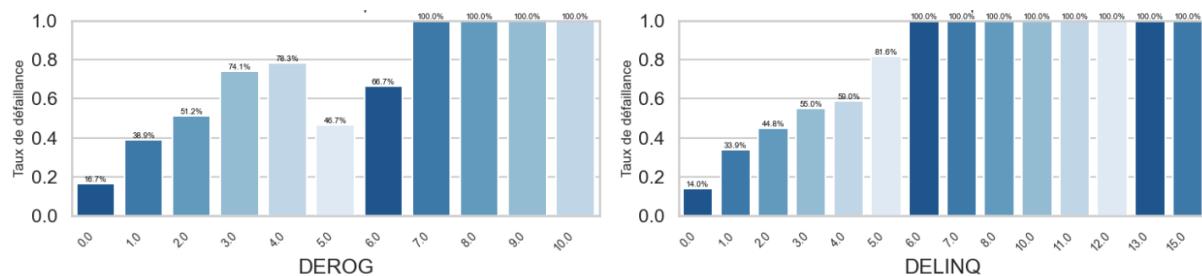


FIGURE 4 – Taux de défaillance par DELINQ et DEROG

Dérogations et lignes de crédit en défaut : Des indicateurs de risque systématique
Plus un emprunteur a de rapports dérogatoires ou de lignes de crédit en défaut, plus son risque de défaillance est élevé. Avec 6 antécédents ou plus, les taux de défaillance approchent 100%, confirmant qu'un historique difficile est un fort indicateur de risque. De même, en Annexe A.3, le risque de défaillance augmente avec les demandes de crédit récentes : il reste modéré jusqu'à 3 demandes, dépasse 50% à partir de 6 et atteint 100% après 11. Les demandes de crédit anticipent souvent des difficultés, en faisant un indicateur de vulnérabilité financière.

Les annexes en A.3 apportent des informations complémentaires sur le risque de défaut. L'analyse des petites hypothèques montre un risque élevé, tandis que les grandes hypothèques, souvent accordées sous des critères plus stricts, sont associées à une meilleure stabilité financière. Quant à la raison du prêt, la consolidation de dettes et les améliorations domiciliaires présentent des risques similaires. Enfin, les professions stables, comme les cadres, présentent un risque de défaut réduit, contrairement aux indépendants et vendeurs, davantage exposés à l'instabilité.

3.2.3 Vers une compréhension plus fine des relations

Dans cette section, nous allons examiner des résultats plus complexes, qui croisent plusieurs variables continues et utilisent des moyennes mobiles pour lisser les évolutions. À travers ces analyses, nous pourrons affiner nos hypothèses.



FIGURE 5 – Taux de défaillance selon LOAN et CLAGE

Rôle de l'ancienneté de crédit dans la gestion du risque : Un facteur atténuant
L'analyse du taux de défaillance en fonction du montant du prêt et de l'ancienneté des lignes de crédit montre que les emprunteurs avec des lignes récentes (Tranches 1-2) présentent un risque élevé, surtout pour de petits montants, en raison d'un historique de crédit limité. À l'inverse, les emprunteurs avec des lignes plus anciennes (Tranches 3-4) sont moins risqués, même pour des montants élevés, mais le risque augmente pour des prêts très élevés. Une longue expérience de crédit est donc protectrice, sauf si elle est associée à des antécédents financiers négatifs.

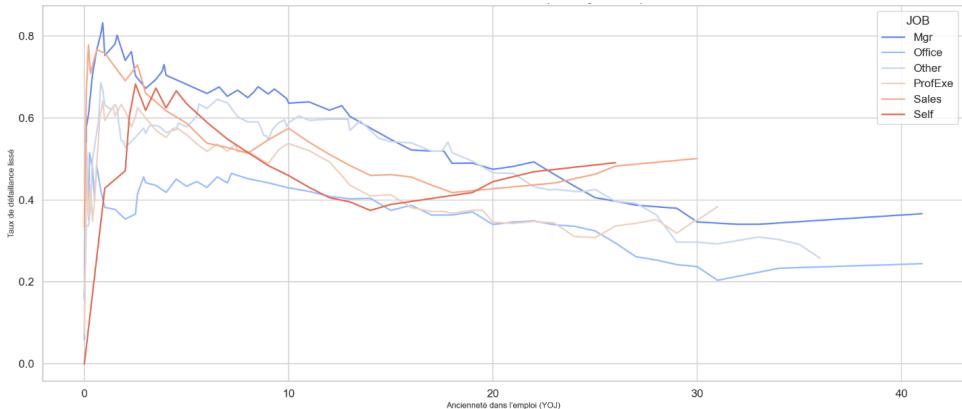


FIGURE 6 – Taux de défaillance par emploi et ancienneté

Impact de l'ancienneté dans l'emploi : Un amortisseur selon la catégorie professionnelle Les emprunteurs avec une ancienneté faible (moins de 5 ans) présentent un taux de défaillance élevé, indépendamment de leur secteur d'activité, mais cette tendance diminue avec l'ancienneté. Les emprunteurs occupant des postes de cadres ou de gestion (Mgr, ProfExe) affichent des taux de défaillance plus bas que ceux ayant des métiers précaires ou instables.

Les annexes en A.4 précisent d'autres facteurs de risque. L'analyse de l'interaction entre le ratio dette/revenu et la raison du prêt montre que la consolidation de dettes est associée à des taux de défaillance élevés, surtout pour les ratios supérieurs à 40 %, tandis que les emprunts pour des améliorations domiciliaires sont plus stables. De même, l'étude des rapports dérogatoires et de la valeur de la propriété souligne qu'un historique de crédit négatif maintient un risque élevé, quelle que soit la valeur du bien, alors qu'un bon historique réduit le risque pour les biens de grande valeur. Enfin, le nombre de lignes de crédit en défaut augmente pour les emprunteurs en difficulté, ce qui indique que des défauts concentrés sur certaines lignes sont un prédicteur clé de défaut généralisé.

4 Comment trouver des modèles potentiels ?

4.1 Objectif

Afin de bien orienter et optimiser la phase de modélisation, il est essentiel de définir clairement la métrique sur laquelle nous devons nous concentrer à ce stade. Dans la majorité des projets de scoring, nous observons fréquemment que le ROC AUC est une métrique privilégiée. Cependant, ROC AUC et F1 pondéré sont des métriques "pondérées" :

- Elles offrent généralement de bonnes performances sur des ensembles de données où les classes 0 et 1 sont équilibrées.
- Il suffit que la classe majoritaire soit bien prédite pour que les résultats de ROC AUC et de F1 soient relativement satisfaisants, cela ne reflète pas une information complète sur le ratio entre le nombre de faux positifs et de faux négatifs.

Cependant, dans notre cas particulier, où le dataset est fortement déséquilibré, une analyse plus détaillée des taux de precision-recall révèle un écart significatif, avec des performances extrêmement faibles pour le recall, souvent inférieures à 30% sur plusieurs modèles. En pratique, le choix entre les stratégies basées sur le ROC AUC, la precision et le recall dépend en grande partie de la stratégie adoptée par l'entreprise. Par exemple, une entreprise peut chercher à attirer un large éventail de nouveaux clients, même ceux présentant un risque élevé, tandis qu'une autre peut privilégier une stricte gestion des prêts, en ne les accordant qu'aux candidats les moins risqués.

Dans le cadre de notre dataset, marqué par la complexité des faux négatifs, nous avons choisi de prioriser la métrique **recall** comme objectif principal. Cette métrique sera utilisée comme base pour optimiser les performances et évaluer, ainsi que comparer, les différents modèles dans toutes les phases ultérieures. Notre objectif est d'atteindre les meilleures performances possibles en recall, tout en adoptant une stratégie rigoureuse de contrôle des risques financiers afin de mieux gérer les flux de prêts.

4.2 Comment identifier les modèles potentiels ?

En pratique, le choix des modèles constitue l'une des tâches qui exige non seulement une expertise technique, mais aussi une solide connaissance du métier. Ainsi, déterminer quel modèle à adopter est un défi considérable. En général, il existe deux approches courantes pour rendre le choix du modèle plus clair.

4.2.1 Méthode 1 : Prédiction des modèles potentiels en se basant sur l'analyse du chevauchement (overlapping) à travers des pairplots

Dans le pairplot ci-dessus, nous pouvons observer un chevauchement (overlapping) assez important entre les deux groupes de données. Cela indique que la séparation des données entre les deux classes n'est pas aisée en se basant uniquement sur des relations linéaires simples ou sur quelques variables isolées.

- Dans les scatter plots entre paires de variables, les points sont souvent mélangés et difficilement séparables de manière claire. Cela est particulièrement visible dans les paires de variables continues comme 'LOAN' avec 'VALUE', ou 'DELINQ' avec 'CLAGE'.
- Dans les graphiques KDE (Kernel Density Estimate) sur la diagonale, les distributions des deux groupes de données se chevauchent dans la plupart des variables, ce qui montre que se baser sur une seule variable ne suffira pas pour une classification efficace.

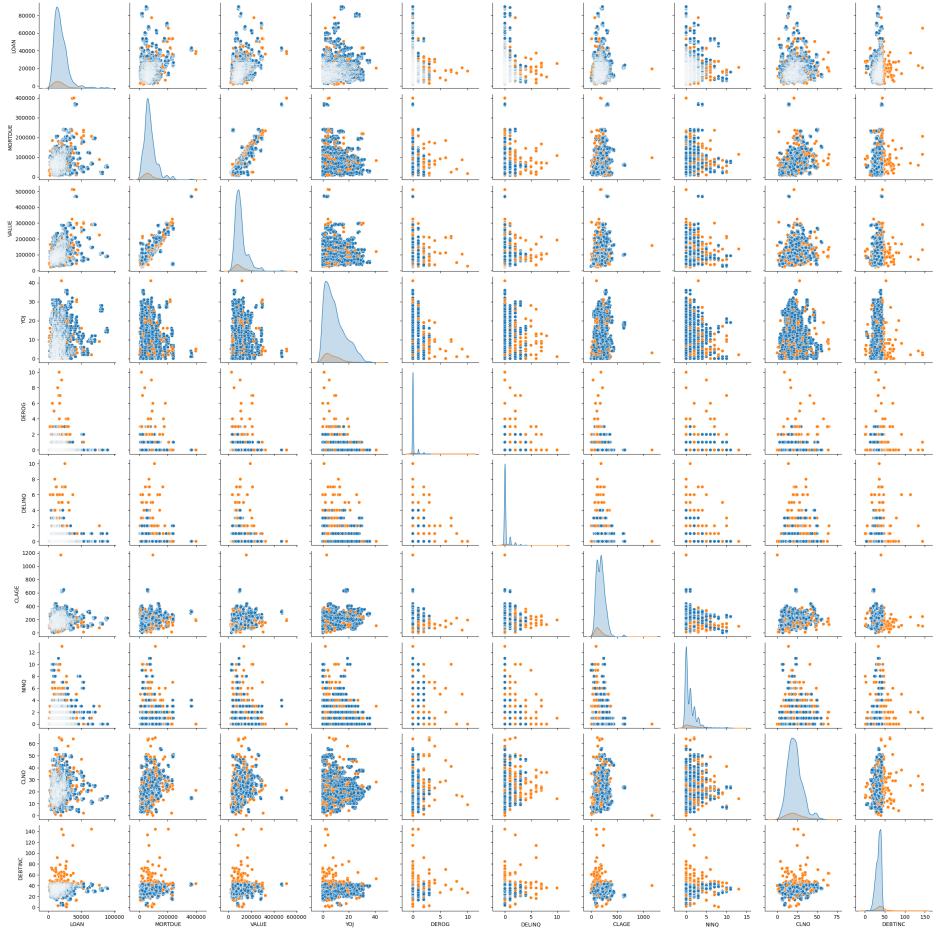


FIGURE 7 – Pairplot des variables

A partir de ces informations, on peut prédire les modèles pouvant être utilisés :

Régression logistique	Elle pourrait ne pas être le meilleur choix car le niveau de chevauchement entre les deux classes est élevé, rendant difficile la séparation linéaire des données.
Arbres de décision / Forêts aléatoires	Les arbres ou forêts aléatoires pourraient être des options viables, car ils ont la capacité de traiter des relations complexes entre les variables et de trouver des seuils optimaux.
Gradient Boosting Machines (GBM) / XGBoost	Les modèles GBM ou XGBoost sont particulièrement prometteurs dans ce cas, car ils combinent des modèles d'arbres de décision et sont capables de bien gérer des jeux de données où les classes se chevauchent.
Support Vector Machines (SVM)	Les SVM pourraient être envisagés, surtout avec un kernel non linéaire comme le kernel RBF, car ils peuvent créer des frontières de séparation plus efficaces entre les deux groupes de données ayant un chevauchement important.
K-Nearest Neighbors (KNN)	KNN pourrait être une méthode de secours, mais son efficacité dépendra du bon choix du paramètre k et de la capacité du modèle à traiter les variables dépendantes linéaires.

4.2.2 Méthode 2 : Prédiction des modèles potentiels en utilisant la bibliothèque LazyPredict

Avec l'évolution des bibliothèques disponibles, il est désormais possible d'évaluer rapidement et de sélectionner certains candidats les plus pertinents en fonction de la métrique choisie. Pour ce faire, j'ai utilisé la bibliothèque `LazyPredict`, qui permet une comparaison préliminaire efficace des performances de différents modèles.

Il s'agit des points importants à noter avec la bibliothèque `LazyPredict`. Elle adopte une approche extrêmement simplifiée et minimale pour le traitement des données, dont le seul but est de permettre au modèle de s'exécuter (sans véritable optimisation des étapes de preprocessing comme nous le faisons manuellement). Nous savons que différents types de prétraitement (preprocessing) conduisent à des résultats différents. Par conséquent, l'idée est d'utiliser les résultats de `LazyPredict` comme une base de comparaison "par défaut", puis de voir comment l'optimisation manuelle des étapes de preprocessing avec différentes techniques impacte les métriques de performance. Sur le plan technique, dans la bibliothèque `LazyPredict` ([Lien Github de LazyPredict](#)) :

- Les variables numériques sont traitées via un pipeline en deux étapes : `SimpleImputer (strategy=mean)` et `StandardScaler`.
- Les variables catégorielles avec une faible cardinalité (moins de 11 modalités) sont traitées par un pipeline comprenant : `SimpleImputer (strategy=constant, fill_value=missing)` et `OneHotEncoder`.
- Les données catégorielles avec une haute cardinalité sont traitées différemment avec un pipeline utilisant : `SimpleImputer (strategy=constant, fill_value=missing)`, puis `OrdinalEncoder`.
- Elle présente une faiblesse majeure : elle ne fait pas de distinction claire entre les variables ordinaires et nominales.

Un autre facteur crucial dans la sélection des modèles est le temps d'exécution. Cet aspect est souvent négligé dans un cadre académique, mais dans la réalité des entreprises, la facilité de déploiement est primordiale, et un modèle léger, rapide et performant de manière stable est souvent préféré. Par conséquent, le modèle retenu doit non seulement optimiser la métrique choisie, mais aussi présenter un temps d'exécution raisonnable.

Nous avons créé quatre versions de l'ensemble de données avec quatre méthodes de prétraitement différentes (pour évaluer la réaction de chaque approche de preprocessing) sur divers modèles selon trois métriques : F1, recall et ROC-AUC. Les données utilisées dans les quatre cas présentés ici ont été épurées de la colonne MORTDUE, en raison de la multicolinéarité avec la colonne VALue et la colonne VALUE a été conservée car elle contient moins de valeurs manquantes.

Dans la première version : les données sont introduites sans traitement préalable, et la bibliothèque `LazyPredict` s'occupe uniquement du minimum requis pour que les modèles puissent être exécutés.

Dans la deuxième version : les colonnes contenant des données continues ont été épurées de leurs outliers à l'aide de la méthode utilisant l'IQR (avec un seuil par défaut de 1,5 pour l'IQR).

Dans la troisième version : les données ont été traitées avec les meilleurs paramètres trouvés grâce à `GridSearchCV` en optimisant le recall pour le modèle de régression logistique.

Dans la quatrième version : les données ont été traitées de la même manière que dans la troisième version, avec en plus l’application de **SMOTE** (Synthetic Minority Over-sampling Technique), une technique d’oversampling couramment utilisée dans les cas de déséquilibre de classes pour améliorer la répartition des données entre les classes. Les étapes de SMOTE sont les suivantes : il sélectionne un échantillon minoritaire, identifie ses k plus proches voisins, puis génère un nouvel exemple synthétique en interpolant entre l’échantillon sélectionné et l’un de ses voisins.

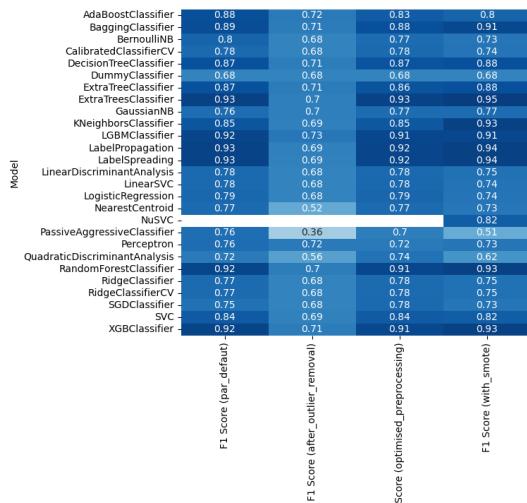


FIGURE 8 – Heatmap des F1 Scores

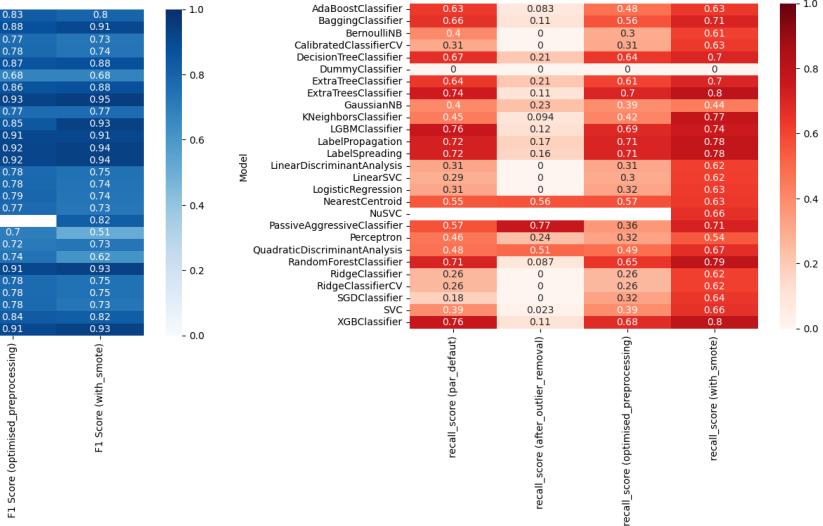


FIGURE 9 – Heatmap des Recall Scores

A partir des graphiques, nous pouvons remarquer que :

- L’utilisation de SMOTE combinée aux paramètres optimisés (colonne 4) constitue une approche de prétraitement efficace, améliorant de manière significative la performance de la majorité des modèles, en particulier ceux sensibles aux déséquilibres de classe. Globalement, SMOTE contribue à l’amélioration du F1-score pour la plupart des modèles, bien que l’augmentation ne dépasse souvent que quelques pourcents. La ROC-AUC montre une légère augmentation pour certains modèles de gradient boosting ou arbres de décision dans cette colonne. Il est important de noter que lorsque le recall augmente fortement grâce à SMOTE, la précision diminue inévitablement en raison du compromis entre ces deux métriques.
- L’élimination des valeurs outliers a rendu les modèles moins aptes à apprendre des exemples « atypiques », ce qui a globalement réduit les performances.
- Sur la heatmap de recall, dans la colonne 3, le recall de la régression logistique ainsi que celui des autres modèles linéaires s’améliore (car l’optimisation a été axée sur le recall). Cependant, les performances des modèles non linéaires n’ont pas montré d’amélioration notable et ont même parfois diminué, indiquant que les paramètres optimisés et le prétraitement appliqué ne conviennent pas à tous les modèles.
- Les modèles de gradient boosting ont montré une robustesse exceptionnelle, fonctionnant de manière stable malgré les valeurs manquantes et nécessitant seulement un prétraitement minimal pour atteindre des performances satisfaisantes.

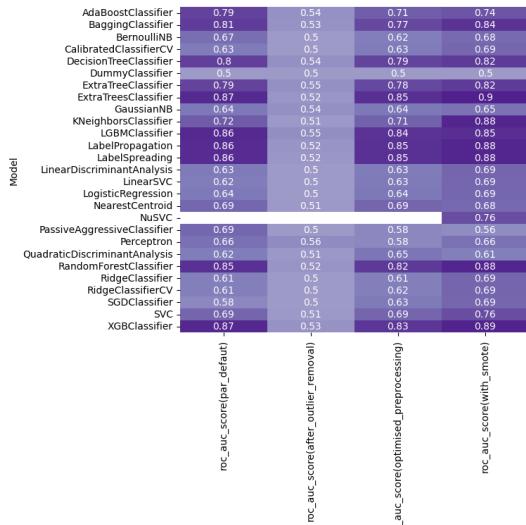


FIGURE 10 – Heatmap de ROC-AUC

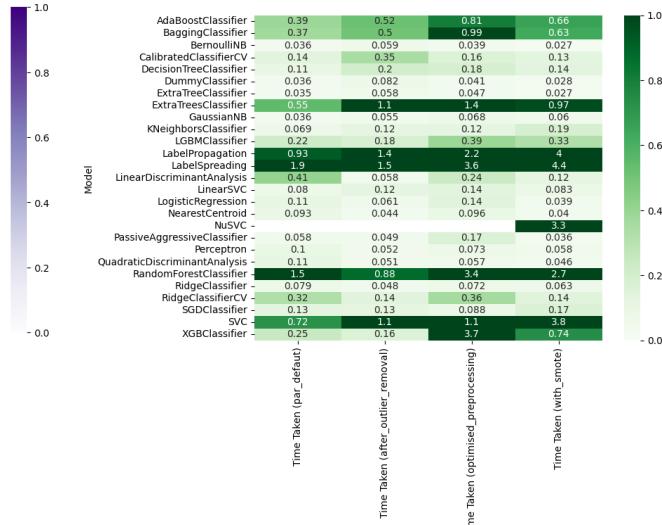


FIGURE 11 – Heatmap du Temps d'exécution

Ainsi, à partir de l'analyse ci-dessus, nous avons sélectionné un modèle de base, **la régression logistique (LogisticRegression)**, ainsi que deux modèles de gradient boosting **XGBOOST** et **LightGBM** comme trois candidats pour optimiser les performances dans la section suivante et comparer leurs résultats. Bien que les forêts aléatoires (Random Forests) soient parmi les modèles offrant de bonnes performances, elles nécessitent des ressources de calcul importantes, notamment lorsque le nombre d'arbres ou la taille du jeu de données augmente. La construction de centaines d'arbres dans une forêt aléatoire peut consommer beaucoup de temps et de mémoire, rendant la phase de test particulièrement lourde. Dans les situations où un traitement rapide est nécessaire, des modèles plus légers et optimisés en termes de temps d'exécution seraient des options plus pragmatiques et adaptées.

5 Modélisation avec les modèles choisis

5.1 Régression Logistique

Dans cette partie, nous allons examiner le prétraitement appliqué aux données de manière générale, puis les prétraitements spécifiques selon les types de modèles essayés. Nous commencerons par les transformations générales, puis nous aborderons 4 cas spécifiques à la régression logistique :

1. Un premier cas où aucun feature engineering n'est effectué.
2. Un deuxième cas où nous traitons les déséquilibres de classes.
3. Un troisième cas où nous utilisons la discrétilisation avec traitement des déséquilibres.
4. Un quatrième cas avec combinaison de certaines variables et traitement des déséquilibres.

5.1.1 Prétraitement Général aux Modèles

Traitement des valeurs abberantes : En analysant les valeurs minimales et maximales des variables, nous pouvons constater que la majorité d'entre elles paraissent plausibles et logiques dans le contexte d'un prêt immobilier. Pour exemple, les valeurs minimales et maximales pour LOAN (allant de 1100 à 89900) et VALUE (allant de 8000 à 855909) sont réalistes pour des montants de prêts et des valeurs immobilières. Cependant, ce ratio est possible, donc nous choisissons de le conserver.

	BAD	LOAN	MORTDUE	VALUE	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
count	5960.000000	5960.000000	5442.000000	5648.000000	5445.000000	5252.000000	5380.000000	5652.000000	5450.000000	5738.000000	4693.000000
mean	0.199497	18607.969799	73760.817200	101776.048741	8.922268	0.254570	0.449442	179.766275	1.186055	21.296096	33.779915
std	0.399656	11207.480417	44457.609458	57385.775334	7.573982	0.846047	1.127266	85.810092	1.728675	10.138933	8.601746
min	0.000000	1100.000000	2063.000000	8000.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.524499
25%	0.000000	11100.000000	46276.000000	66075.500000	3.000000	0.000000	0.000000	115.116702	0.000000	15.000000	29.140031
50%	0.000000	16300.000000	65019.000000	89235.500000	7.000000	0.000000	0.000000	173.466667	1.000000	20.000000	34.818262
75%	0.000000	23300.000000	91488.000000	119824.250000	13.000000	0.000000	0.000000	231.562278	2.000000	26.000000	39.003141
max	1.000000	89900.000000	39950.000000	855909.000000	41.000000	10.000000	15.000000	1168.233561	17.000000	71.000000	203.312149

FIGURE 12 – Statistiques descriptives des données

De même, pour CLAGE (âge de la plus ancienne ligne de crédit), la valeur maximale de 1168 mois (soit environ 97 ans) semble peu probable et pourrait être une erreur de saisie, car elle impliquerait que le client avait une ligne de crédit ouverte depuis près d'un siècle. Le box plot de la distribution de CLAGE ci-dessous révèle en effet deux observations avec une ancienneté de crédit d'environ 97 ans, ce qui justifie leur suppression.

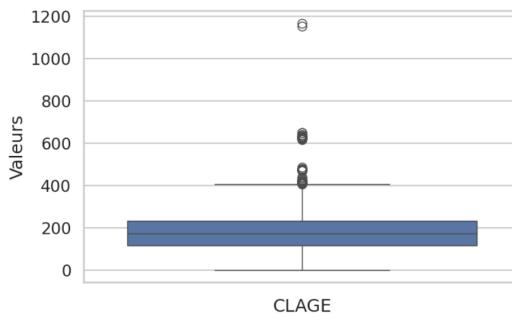


FIGURE 13 – Box plot de la variable CLAGE

Traitement des valeurs manquantes : Le jeu de données présente un taux de valeurs manquantes de 6,8 % (comme illustré en Annexe 2), ce qui nécessite un traitement pour éviter tous biais dans l'analyse. L'analyse des coefficients d'asymétrie (skewness) a révélé une forte asymétrie dans la distribution de la majorité des variables. Par conséquent, la méthode d'imputation la plus appropriée pour les variables numériques est l'imputation par la médiane, car elle est plus robuste aux valeurs extrêmes que la moyenne. Pour les variables catégorielles, nous utilisons l'imputation par le mode (valeur la plus fréquente). Pour maintenir la cohérence et éviter le data leakage, les valeurs d'imputation (médianes et modes) sont calculées uniquement sur l'ensemble d'entraînement, puis réutilisées pour imputer les valeurs manquantes dans l'ensemble de test. Cette approche garantit qu'aucune information de l'ensemble de test n'influence le processus d'imputation.

5.1.2 Prétraitement Spécifique aux Modèles

Premier et Deuxième Cas : Modèle Sans Feature Engineering + avec équilibre de classes Pour ce premier cas, nous avons commencé par vérifier la linéarité des variables continues avec la variable cible (BAD) à l'aide du test de Box-Tidwell. Les résultats ont montré que certaines variables ne respectaient pas l'hypothèse de linéarité nécessaire pour la régression logistique. Ainsi, nous avons appliqué des transformations log appropriées :

1. **Test de linéarité initial :** Le test de Box-Tidwell a révélé que des variables telles que LOAN, MORTDUE, VALUE, DEBTINC n'avaient pas une relation linéaire avec la cible ($p < 0,05$). En revanche, YOJ et CLAGE étaient déjà linéaires ($p > 0,05$). Pour ces variables qui ont montré un manque de linéarité, des transformations appropriées ont été appliquées afin de mieux respecter l'hypothèse de linéarité, essentielle pour le modèle de régression logistique.

2. Transformations appliquées :

- **LOAN, MORTDUE, VALUE, DEBTINC** : Ces variables, qui représentent des montants financiers ou des ratios, ont été transformées en utilisant une transformation logarithmique (log1p). Cette transformation est couramment utilisée pour réduire l'impact des valeurs extrêmes et améliorer la linéarité avec la variable cible.
- **DEROG, DELINQ, NINQ** : Ces variables sont des données de comptage discrètes, avec des valeurs relativement faibles. Aucune transformation n'a été effectuée, car les transformations logarithmiques ou autres ne sont pas toujours bénéfiques pour des variables discrètes de faible ampleur.
- **CLNO** : Cette variable, bien qu'étant une donnée de comptage, présente un grand éventail de valeurs. Par conséquent, une transformation par racine carrée a été appliquée pour atténuer l'effet des valeurs extrêmes.
- **YOJ, CLAGE** : Ces variables ont déjà une relation linéaire suffisante avec la cible, et ont donc été conservées sans transformation.

3. Analyse des Corrélations :

Une matrice de corrélation a été construite pour vérifier la corrélation entre les variables. Une forte corrélation a été constatée entre certaines variables, comme VALUE et MORTDUE.

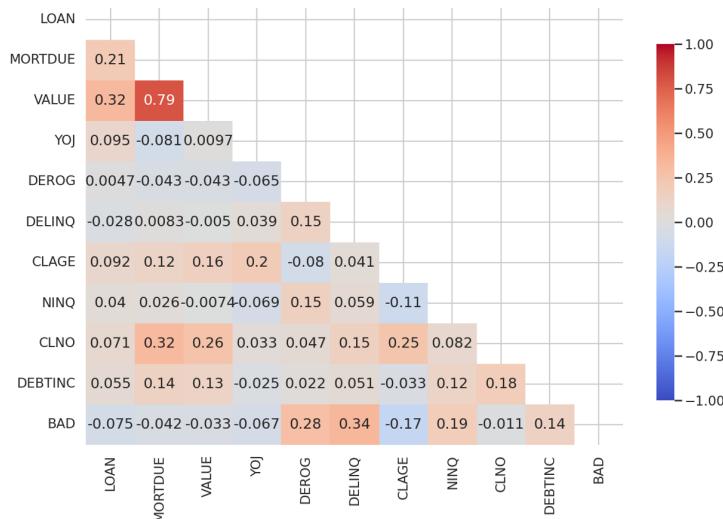


FIGURE 14 – Matrice de corrélation entre les variables numériques et la variable cible

4. **Multicolinéarité (VIF) entre les variables numériques :** Les valeurs VIF (Variance Inflation Factor) ont été calculées pour évaluer la multicolinéarité entre les variables continues. Les variables présentant des VIF élevés, par exemple, VALUE (1018.066131), MORTDUE (633.251983) ou encore LOAN (288.978784) ont été marquées comme problématiques pour la stabilité du modèle.
5. **Multicolinéarité (test de Cramer) entre les variables catégorielles :** Le V de Cramer a été calculé pour évaluer les associations entre les variables catégorielles. Le résultat du test entre les variables JOB et REASON a donné 0.1449. La valeur de V de Cramer n'a dépassé 0,3, ce qui indique qu'il n'y a pas de forte association entre les variables catégorielles. Cela suggère que les variables REASON et JOB sont relativement indépendantes et qu'il est peu probable qu'une multicolinéarité importante soit présente entre elles.
6. **Multicolinéarité (ANOVA) entre variables numériques et catégorielles :** Pour évaluer les relations entre les variables numériques et les variables catégorielles, un test

ANOVA a été utilisé ainsi que l'eta carré (eta-squared) pour mesurer l'effet de chaque variable catégorielle sur chaque variable numérique. Les résultats principaux sont :

- **VALUE et JOB** ont une relation modérée (eta-squared = 0,1352), ce qui suggère que la catégorie d'emploi influence modérément la valeur de la propriété.
- **MORTDUE et JOB** montrent également une relation modérée (eta-squared = 0,0870).
- **LOAN et REASON** présentent une relation modérée (eta-squared = 0,0601).

Dans l'ensemble, la majorité des relations présentent un effet faible, ce qui indique une faible dépendance entre les variables catégorielles et numériques. Cela peut guider la sélection des caractéristiques et la réduction de la complexité des modèles.

7. **Standardisation des Données** : Etant donné que les données ne suivent pas une loi normale, nous avons choisi de standardiser les données en utilisant le MinMaxScaler. Cette transformation permet d'assurer une échelle uniforme pour toutes les variables, facilitant ainsi leur interprétation et leur utilisation dans des algorithmes sensibles à l'échelle, tel que la régression logistique. De plus, cela permet de réduire l'impact des valeurs extrêmes sur les analyses, ce qui contribue à une meilleure performance de nos modèles de machine learning.
8. **Encodage des variables catégorielles** : Les variables catégorielles ont été encodées via un one-hot encoding, excluant la première modalité pour éviter la multicolinéarité. Cela a permis de convertir ces variables en un format numérique exploitable pour la régression logistique.

Ces transformations et analyses ont été appliquées au jeu de données d'entraînement et de test pour assurer la cohérence des analyses et améliorer les performances des modèles prédictifs, tels que la régression logistique, Lasso, Ridge et Elastic Net. Nous avons également testé chaque modèle en appliquant un traitement des déséquilibres des classes à l'aide de SMOTE et de class weights.

Troisième Cas : Modèle avec Discrétisation Dans le cadre de notre analyse, nous avons également testé l'impact de la discrétisation des variables continues sur la performance des modèles.

1. **Discrétisation des variables numériques** : La discrétisation constitue une méthode alternative à la standardisation et pourrait s'avérer plus efficace avec la régression logistique. Nous allons comparer les deux approches dans le cadre de notre analyse.
 - **Variables Monétaires (LOAN, MORTDUE, VALUE)** : Ces variables ont été discrétisées en cinq classes à l'aide de la méthode des quantiles, afin de capturer les différences dans les niveaux de valeur tout en réduisant l'influence des valeurs extrêmes. Par exemple, la variable LOAN a été discrétisée avec des seuils allant de 1100 à 89900, définissant ainsi cinq intervalles distincts.
 - **Variables de Comptage (DEROG, DELINQ, NINQ)** : Ces variables ont été regroupées en trois classes (0, 1, 2+). Cette simplification permet de rendre l'analyse plus robuste et de réduire la complexité, surtout lorsque les variables ont des valeurs faibles.
 - **Variables de Durée (YOJ, CLAGE) et autres variables numériques (CLNO, DEBTINC)** : Ces variables ont été discrétisées en quatre ou cinq classes, selon leur distribution. Par exemple, CLAGE a été divisée en quatre intervalles afin de mieux représenter l'ancienneté des lignes de crédit.

La discrétisation a été choisie pour rendre les modèles plus robustes aux variations et réduire l'impact des valeurs extrêmes. Cette approche sera testée dans les prochaines étapes de modélisation pour évaluer son impact sur la performance.

2. Test de Cramer pour les variables catégorielles : La discréétisation nous a permis de réduire les associations fortes entre certaines variables continues, notamment entre MORTDUE et VALUE, qui avaient une forte corrélation avant discréétisation (Cramer's V de 0.79) et qui a été réduite à 0.54 après discréétisation. Cette réduction des corrélations permet de limiter les problèmes de multicolinéarité, améliorant ainsi la stabilité et la robustesse du modèle.

	LOAN	MORTDUE	VALUE	Y0J	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
LOAN	0.0000	0.1495	0.1958	0.0773	0.0502	0.0647	0.0806	0.0484	0.1078	0.1093
MORTDUE	0.1495	0.0000	0.5458	0.0921	0.0479	0.0683	0.1146	0.0726	0.1953	0.1122
VALUE	0.1958	0.5458	0.0000	0.0545	0.0661	0.0616	0.1367	0.0579	0.1953	0.1070
Y0J	0.0773	0.0921	0.0545	0.0000	0.0498	0.0597	0.1064	0.0679	0.0774	0.0597
DEROG	0.0502	0.0479	0.0661	0.0498	0.0000	0.1337	0.0624	0.1028	0.0390	0.0928
DELINQ	0.0647	0.0683	0.0616	0.0597	0.1337	0.0000	0.0446	0.0661	0.1019	0.1154
CLAGE	0.0806	0.1146	0.1367	0.1064	0.0624	0.0446	0.0000	0.0778	0.1572	0.0465
NINQ	0.0484	0.0726	0.0579	0.0679	0.1028	0.0661	0.0778	0.0000	0.1117	0.1417
CLNO	0.1078	0.1953	0.1953	0.0774	0.0390	0.1019	0.1572	0.1117	0.0000	0.0902
DEBTINC	0.1093	0.1122	0.1070	0.0597	0.0928	0.1154	0.0465	0.1417	0.0902	0.0000

FIGURE 15 – Associations avec Cramer's V

3. Encodage des Variables : Pour préparer les données discréétisées à l'entraînement des modèles, nous avons appliqué des techniques d'encodage Ordinal, Utilisé pour les variables numériques discréétisées, permettant de conserver l'ordre des catégories, et le One-Hot Encoding appliqué aux variables catégorielles d'origine, afin de représenter chaque catégorie par une colonne binaire distincte. Cette méthode permet de mieux capter l'information sans introduire de relation d'ordre artificielle.

Ces prétraitements spécifiques ont été appliqués aux jeux de données d'entraînement et de test, en vue de tester leur impact sur les performances des modèles tels que la régression logistique, le Ridge, le Lasso, et ElasticNet, et en évaluant leur efficacité avec des méthodes comme SMOTE et *class weights* pour traiter les déséquilibres de classes.

Quatrième Cas : Modèle avec Combinaison de variables Pour ce dernier cas, nous avons décidé de créer des variables combinées afin de capturer des informations plus complexes dans les données, réduire la multicolinearité et ainsi améliorer les performances de nos modèles.

1. Nouvelles variables : Voici les principales transformations effectuées :

- **Weighted Risk Score :** Calculé en combinant les variables DEROG, DELINQ, et NINQ avec des pondérations différentes (3, 2, 1), puis normalisé. Cette variable permet de mesurer le risque global associé à un client, en donnant plus de poids aux incidents les plus graves (DEROG).
- **Debt to Income (DEBTINC) :** Conservé tel quel car il s'agit d'une mesure standard qui est couramment utilisée pour évaluer la capacité de remboursement d'un emprunteur.
- **Modified Credit History Score :** Calculé en divisant le nombre de lignes de crédit ouvertes (CLNO) par l'âge de la ligne de crédit la plus ancienne (CLAGE), puis ajusté par le nombre d'incidents de crédit (DEROG, DELINQ). Cette variable vise à capturer la qualité de l'historique de crédit d'un client.
- **Modified Loan Stability Ratio :** Calculé en prenant le logarithme du montant du prêt (LOAN), divisé par une fonction du nombre d'années d'ancienneté dans l'emploi (Y0J). Cela permet de mieux évaluer la stabilité financière d'un client en tenant compte de son expérience professionnelle.

- **Value Coverage Ratio** : Calculé en divisant la valeur de la propriété (VALUE) par le montant du prêt (LOAN), avec une transformation logarithmique pour réduire l'impact des valeurs extrêmes. Cela permet d'évaluer la couverture de la dette par la valeur du bien immobilier.

Ces transformations ont été appliquées afin de mieux capturer les relations non linéaires et de réduire l'impact des valeurs extrêmes sur les modèles. Par exemple, l'utilisation de transformations logarithmiques aide à atténuer l'effet des valeurs très élevées qui pourraient biaiser l'entraînement des modèles. De plus, la création de variables combinées permet de synthétiser plusieurs informations en une seule, facilitant ainsi l'interprétation des modèles.

2. **Corrélations** : Une analyse des corrélations entre les nouvelles variables et la variable cible a révélé que **WEIGHTED RISK SCORE** avait la corrélation la plus forte avec **BAD** (0.443), suivie de **DEBT TO INCOME** (0.148). Cela suggère que ces nouvelles variables apportent une valeur informative significative pour prédire le risque de défaut.

Les valeurs du **Variance Inflation Factor (VIF)** ont également été calculées pour évaluer la multicolinéarité. Les résultats montrent que certaines variables, comme **DEBT TO INCOME** et **VALUE COVERAGE**, présentent des valeurs de VIF relativement élevées (respectivement 10.27 et 9.30), ce qui indique une possible redondance d'information. Toutefois, **WEIGHTED RISK SCORE** et **CREDIT QUALITY** ont des valeurs de VIF faibles, suggérant qu'elles apportent des informations distinctes et utiles au modèle.

3. **Standardisation** : Ces nouvelles variables ont été standardisées à l'aide de MinMaxScaler pour garantir une échelle uniforme, ce qui est essentiel pour des modèles tels que la régression logistique, Ridge, Lasso, et ElasticNet.

Nous avons ensuite testé ces modèles sur les données transformées, et avons également évalué leur performance avec des techniques de traitement des déséquilibres de classes, telles que SMOTE et *class weights*, pour maximiser la capacité des modèles à détecter les cas de défaut.

5.2 XGBoost

5.2.1 XGBoost version 1 : Prétraitement optimisé avec GridSearchCV

Le modèle XGBOOST utilisé dans cette section est la version **XGBClassifier**. Pour optimiser le prétraitement, nous avons employé GridSearchCV en combinaison avec une validation croisée de k-fold ($k=5$) afin de trouver les meilleures paramètres et hyperparamètres adaptés. À noter que cette recherche a été effectuée uniquement sur l'ensemble d'entraînement, sans utiliser les données de test, afin d'éviter toute fuite de données (*data leakage*). Voici les résultats obtenus : le modèle XGBClassifier est optimisé avec un learning rate de 0,1, une profondeur maximale de 7 et 200 arbres, assurant un apprentissage équilibré entre performance et complexité. Pour le prétraitement :

- Les valeurs manquantes des variables catégorielles sont imputées par la modalité la plus fréquente (SimpleImputer).
- Les variables continues utilisent un IterativeImputer avec régression bayésienne (BayesianRidge) et sont normalisées par un StandardScaler.
- Les valeurs manquantes des variables ordinaires sont comblées par un KNNImputer selon la similarité des observations.

```
Rapport de classification :
precision    recall   f1-score   support
          0       0.92      0.98      0.95      927
          1       0.92      0.71      0.80      265

   accuracy        0.92      0.92      0.92      1192
 macro avg       0.92      0.84      0.87      1192
weighted avg     0.92      0.92      0.92      1192

Matrice de confusion :
[[910  17]
 [ 78 187]]
```

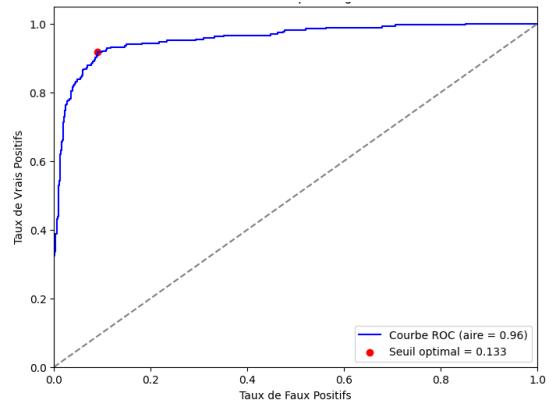


FIGURE 16 – Rapport de la version 1

FIGURE 17 – Courbe ROC de la version 1

(L’analyse des features importantes de XGBoost sera détaillée dans l’annexe.)

À partir des résultats ci-dessus, globalement, ce modèle affiche une performance satisfaisante avec des indicateurs élevés. Cependant, puisque notre objectif initial est d’optimiser le recall, nous allons tester une version supplémentaire intégrant SMOTE ainsi qu’une version utilisant le class weight pour observer l’impact de ces différentes approches sur le recall.

5.2.2 XGBoost Version 2 : Prétraitement optimisé avec GridSearchCV combiné à SMOTE

En utilisant le même ensemble de meilleurs paramètres que dans la version 1, nous ajoutons l’étape de SMOTE afin d’observer l’amélioration potentielle du recall.

```
Rapport de classification (avec SMOTE) :
precision    recall   f1-score   support
          0       0.94      0.97      0.96      927
          1       0.89      0.78      0.83      265

   accuracy        0.92      0.93      0.93      1192
 macro avg       0.92      0.88      0.89      1192
weighted avg     0.93      0.93      0.93      1192

Matrice de confusion (avec SMOTE) :
[[902  25]
 [ 59 206]]
```

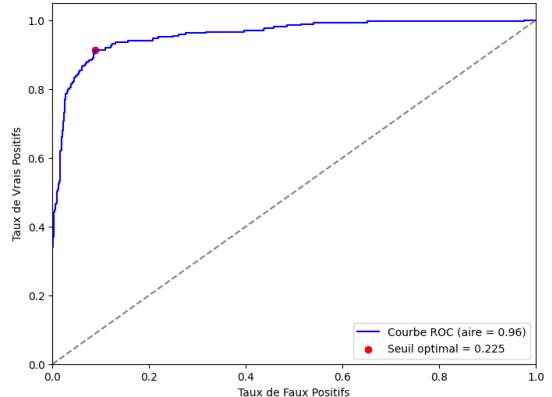


FIGURE 18 – Rapport de la version 2

FIGURE 19 – Courbe ROC (avec SMOTE) de la version 2

5.2.3 XGBoost Version 3 : Prétraitement optimisé avec GridSearchCV combiné à l’utilisation de class-weight

En utilisant le même ensemble de meilleurs paramètres que dans la version 1, nous n’appliquons pas SMOTE mais utilisons à la place l’option `class_weight`, une technique pour gérer les problèmes de déséquilibre des classes dans les jeux de données de classification. La valeur `scale_pos_weight` ajuste le poids de la classe minoritaire. Une bonne valeur de départ pour `scale_pos_weight` est le rapport d’instances entre les classes majoritaire et minoritaire dans le jeu de données :

$$\text{scale_pos_weight} = \frac{\text{nombre d'échantillons de la classe majoritaire}}{\text{nombre d'échantillons de la classe minoritaire}}$$

```
Rapport de classification (avec class_weight) :
precision    recall   f1-score   support
          0       0.94      0.97      0.96      927
          1       0.89      0.80      0.84      265

   accuracy          0.92
   macro avg       0.92      0.89      0.90     1192
weighted avg       0.93      0.93      0.93     1192

Matrice de confusion (avec class_weight) :
[[900 27]
 [ 53 212]]
```

FIGURE 20 – Rapport de la version 3

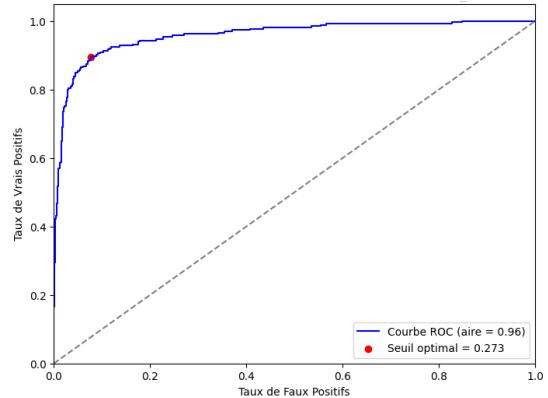


FIGURE 21 – Courbe ROC (avec class_weight) de la version 3

5.3 LightGBM

5.3.1 LightGBM version 1 : Prétraitement optimisé avec GridSearchCV

De manière similaire au modèle XGBClassifier, dans le cas de LGBMClassifier, pour la version 1, nous commencerons par utiliser GridSearchCV en combinaison avec une validation croisée par k-fold ($k=5$) afin d'identifier les meilleurs paramètres et hyperparamètres adaptés. Les meilleurs paramètres sont les suivants :

- Le learning rate est de 0,1, et la profondeur maximale des arbres est fixée à 7.
- Le modèle utilise 200 arbres. Pour les données catégorielles, les valeurs manquantes sont imputées avec la valeur la plus fréquente via SimpleImputer.
- Pour les variables continues, l'imputation est réalisée par IterativeImputer avec une régression bayésienne (BayesianRidge), et la mise à l'échelle utilise RobustScaler.
- Les variables ordinaires manquantes sont complétées avec KNNImputer.

```
Rapport de classification :
precision    recall   f1-score   support
          0       0.92      0.98      0.95      927
          1       0.92      0.69      0.79      265

   accuracy          0.92
   macro avg       0.92      0.84      0.87     1192
weighted avg       0.92      0.92      0.91     1192

Matrice de confusion :
[[911 16]
 [ 81 184]]
```

FIGURE 22 – Rapport de la version 1

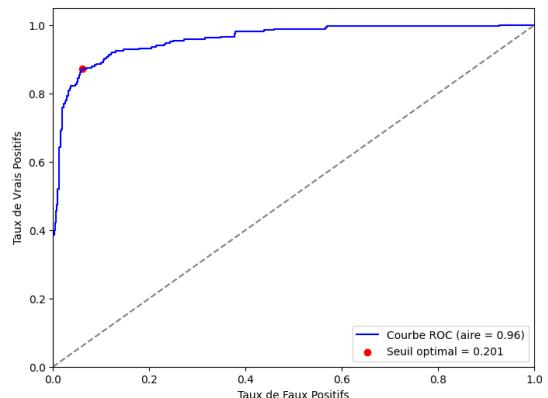


FIGURE 23 – Courbe ROC de la version 1

(L'analyse des features importantes de LightGBM sera détaillée dans l'annexe.)

De manière similaire à XGBClassifier, si nous utilisons les métriques F1-score ou ROC-AUC pour évaluer le modèle, la performance semble globalement satisfaisante à ce stade. Cependant, dans la mesure où nous avons choisi le recall comme métrique la plus importante, nous allons continuer en appliquant SMOTE et l'option class-weight pour examiner leur effet sur l'amélioration du recall.

5.3.2 LightGBM Version 2 : Prétraitement optimisé avec GridSearchCV combiné à SMOTE

Comme dans le cas de XGBoost, nous ajoutons l'étape de SMOTE et utilisons le même ensemble de meilleurs paramètres que dans la version 1, afin d'observer l'amélioration potentielle du recall.

```
Rapport de classification (avec SMOTE) :
precision    recall   f1-score  support
0            0.94     0.97     0.95      927
1            0.88     0.78     0.83      265

accuracy          0.93
macro avg       0.91     0.88     0.89      1192
weighted avg    0.93     0.93     0.93      1192

Matrice de confusion (avec SMOTE) :
[[898 29]
 [ 57 208]]
```

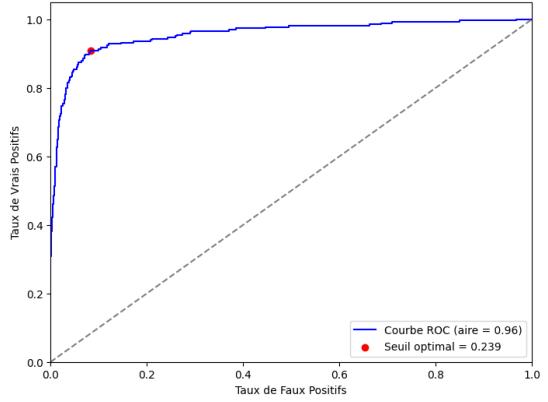


FIGURE 24 – Rapport de la version 2

FIGURE 25 – Courbe ROC de la version 2

5.3.3 LightGBM Version 3 : Prétraitement optimisé avec GridSearchCV combiné à l'utilisation le class-weight

De même manière en XGBClassifier, nous n'appliquons pas SMOTE mais utilisons à la place l'option `class_weight`, pour gérer les problèmes de déséquilibre des classes dans notre jeu de données (La formule de calcul reste la même que dans le cas de XGBoost).

```
Rapport de classification (avec class_weight) :
precision    recall   f1-score  support
0            0.95     0.96     0.95      927
1            0.85     0.81     0.83      265

accuracy          0.93
macro avg       0.90     0.88     0.89      1192
weighted avg    0.92     0.93     0.92      1192

Matrice de confusion (avec class_weight) :
[[889 38]
 [ 51 214]]
```

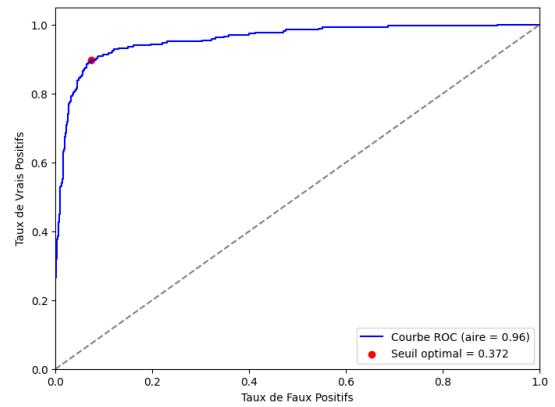


FIGURE 26 – Rapport de la version 3

FIGURE 27 – Courbe ROC de la version 3

6 Évaluation et comparaison des modèles

6.1 Régression logistique

Pour chaque modèle (régression logistique de base, Ridge, Lasso, Elastic Net), nous avons utilisé une recherche par grille (GridSearchCV) pour identifier les meilleurs hyperparamètres. Cette méthode permet de tester plusieurs combinaisons d'hyperparamètres et de sélectionner celle qui maximise la performance du modèle, mesurée ici par le score F1.

Les hyperparamètres optimisés pour chaque modèle incluent les paramètres de régularisation tels que le coefficient de pénalité (C) et, dans le cas d'ElasticNet, le ratio l1. Ces hyperparamètres ont une influence directe sur la capacité du modèle à éviter le surapprentissage et à gérer la variance.

Les résultats ont montré des différences significatives en termes de scores F1, ROC AUC, précision et rappel, selon les méthodes de traitement appliquées (sans équilibrage, avec *class weights* et avec SMOTE).

Le score F1 est choisi comme métrique parce qu'il offre une performance globale. En effet, le recall avait déjà de bons résultats avec la régression logistique de base, notamment lorsque nous avons utilisé SMOTE pour traiter les déséquilibres de classe. C'est pourquoi nous nous sommes focalisés sur l'amélioration du score F1, qui combine la précision et le rappel pour mieux équilibrer les performances globales du modèle, tout en minimisant les faux négatifs.

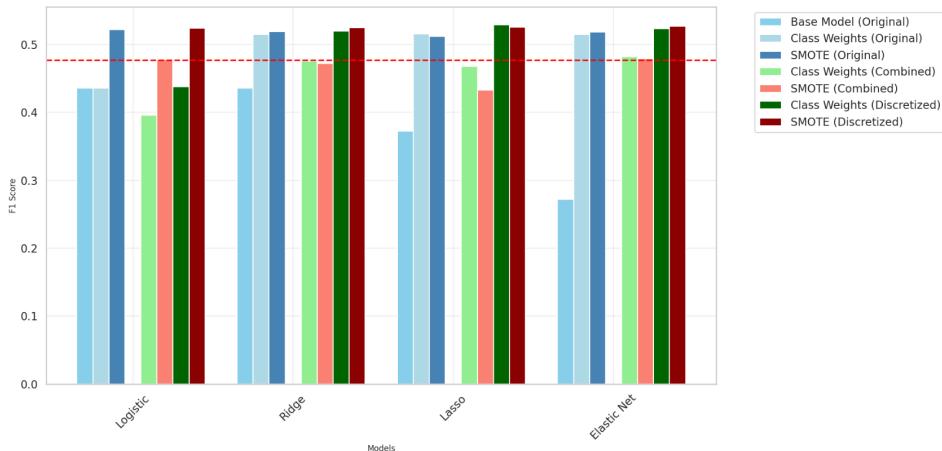


FIGURE 28 – Comparaison des scores F1 des différents modèles et méthodes de traitement des déséquilibres

Les résultats montrent que l'approche utilisant la discrétisation des variables et le Lasso avec des poids de classe a fourni le meilleur score F1 global (0.5287), surpassant les autres méthodes en termes de compromis entre précision et rappel. Cela suggère que la discrétisation a aidé à mieux capturer la distribution des variables et à réduire la multicolinéarité, ce qui a amélioré la robustesse du modèle.

L'approche sans feature engineering, mais avec SMOTE, a également produit de bons résultats (F1 de 0.5217), ce qui montre que la balance des classes est cruciale pour améliorer la capacité de détection des cas de défaut. De même, les caractéristiques combinées, bien que moins performantes que les autres, ont apporté une valeur informative, mais pourraient bénéficier d'une optimisation supplémentaire.

En conclusion, les prétraitements tels que la discrétisation et la création de nouvelles variables ont montré un impact significatif sur les performances des modèles, en particulier lorsqu'ils sont combinés avec des méthodes de traitement des déséquilibres. Le choix du Lasso avec la discrétisation et les poids de classe est particulièrement efficace pour ce jeu de données, démontrant une bonne capacité à gérer à la fois la variance et le biais, tout en minimisant les erreurs critiques.

6.2 Modèles Gradient Boosting

À l'analyse des résultats de XGBClassifier et LGBMClassifier, nous constatons qu'une évaluation reposant uniquement sur les métriques ROC-AUC et accuracy ne montre presque aucune différence significative entre les différentes versions, les variations étant minimales et négligeables. En revanche, la version utilisant class-weight apporte un ajustement optimal, en maintenant un équilibre approprié entre précision et recall, ce qui entraîne même une légère amélioration du F1-score. Par conséquent, la version 3, intégrant class-weight, se révèle la plus appropriée pour atteindre notre objectif principal : optimiser le recall.

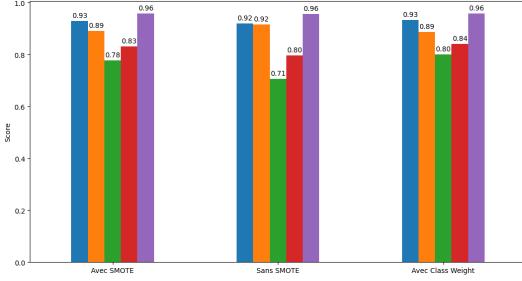


FIGURE 29 – Comparaison des métriques entre XGBClassifier avec SMOTE, sans SMOTE et avec Class Weight

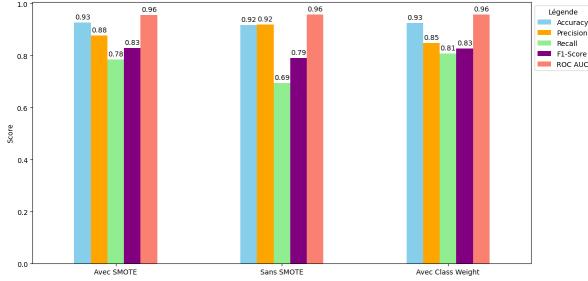


FIGURE 30 – Comparaison des métriques entre LightGBM avec SMOTE, sans SMOTE et avec Class Weight

La performance ne montre quasiment aucune différence significative, bien que XGBoost affiche un léger avantage (mais très marginal) par rapport à LightGBM sur la plupart des métriques, notamment en termes de précision et de F1-score (à l'exception du recall). Cela signifie que XGBoost constitue un modèle particulièrement adapté lorsque des exigences de précision plus élevées sont prioritaires et que les ressources ne sont pas une contrainte. À l'inverse, LightGBM présente un avantage en termes de rapidité et de gestion des ressources. Ainsi, si le temps d'entraînement et l'optimisation des ressources sont cruciaux (par exemple, pour les jeux de données volumineux), LightGBM s'avère être un choix plus approprié.

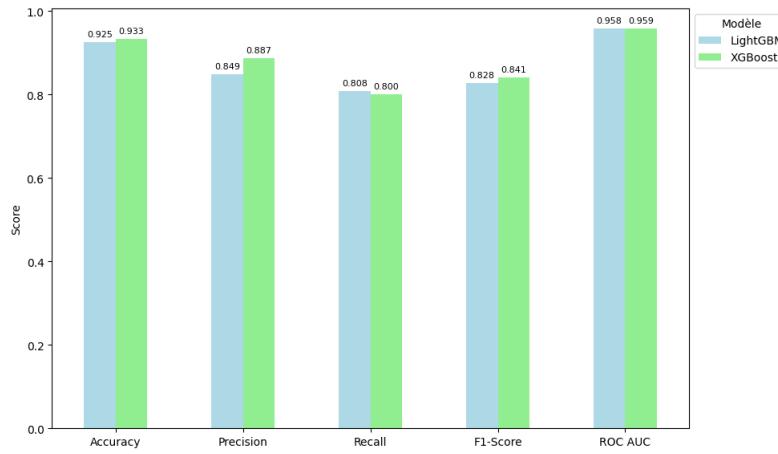


FIGURE 31 – Comparaison des métriques entre XGBoost et LightGBM (Avec Class Weight)

7 Grille de score

Nous avons également construit une grille de score qui permet d'évaluer le risque d'un nouveau demandeur de crédit selon son profil et identifier les segments les plus risqués, ce qui

permet d'ajuster la politique de crédit. Nous avons utilisé deux modèles différents (XGBoost et LightGBM) pour faire ces grilles, ce qui a révélé une concordance remarquable dans leurs résultats. Cette convergence entre les deux approches renforce significativement la robustesse et la fiabilité de nos conclusions en matière d'évaluation des risques de crédit.

Nous constatons que les profils les plus risqués sont les vendeurs (Sales) avec un taux de défaut de base de 22.95% pour les prêts de consolidation de dette (DebtCon) et 100% pour les prêts d'amélioration de l'habitat (HomeImp), les personnes dont la catégorie est Other avec des taux de défaut de 12.87% (DebtCon) et 22.81% (HomeImp) et les travailleurs indépendants (Self) avec 17.65% pour DebtCon. Ainsi, les profils les plus fiables sont les employés de bureau (Office) avec 7.76% de défaut pour DebtCon sans incident et seulement 1.18% pour HomeImp sans incident (meilleur score global) et les professions libérales/cadres (ProfExe) qui montrent une bonne stabilité : 9.41% pour DebtCon et 9.85% pour HomeImp.

En ce qui concerne les incidents de crédit, nous constatons que le premier incident (DEROG=1) augmente significativement le risque de défaut. Ensuite, dès qu'on atteint DEROG=2 ou DELINQ>2, le taux de défaut monte souvent à 100%. Ainsi, nous observons que la présence simultanée de DEROG et DELINQ augmente exponentiellement le risque.

Globalement, les prêts HomeImp sont généralement plus risqués que DebtCon.

Les effectifs montrent que la catégorie Other représente une part importante des emprunteurs (16.95% pour DebtCon), les professions libérales et managers représentent environ 15% des emprunteurs pour DebtCon et ont des taux de défaut modérés. De plus, les segments à 100% de défaut ont généralement des effectifs très faibles (<0.1%).

Cette grille de score permet de définir une politique de crédit claire favorisant les employés de bureau et professions libérales, recommandant le refus systématique des dossiers avec plus de 2 incidents et portant une attention particulière aux prêts HomeImp qui sont globalement plus risqués.

8 Conclusion

En conclusion, notre analyse a démontré l'importance du choix des modèles et des méthodes de prétraitement pour optimiser les performances de scoring dans le cadre de la prédiction du défaut de paiement. Lorsque des chevauchements complexes (overlapping) sont fortement présents dans les données, on peut aisément prévoir que la régression logistique sera moins performante que des modèles basés sur le gradient boosting ou des arbres de décision/forêts aléatoires. Cela a été démontré par la diversification des approches de prétraitement et les régularisations pour encourager l'apprentissage sur la classe minoritaire dans des cas de données déséquilibrées. Toutefois, malgré ces efforts, les performances d'un modèle simple comme la régression logistique restent insuffisantes pour des analyses complexes lorsque les relations dans les données sont enchevêtrées.

De plus, dans le contexte bancaire, la facilité de déploiement est primordiale ; un modèle léger, rapide et stable est souvent préféré. C'est pourquoi les deux modèles candidats XGBoost et LightGBM figurent parmi les choix les plus utilisés en pratique, grâce à leur adaptabilité aux données complexes tout en restant efficaces en termes de temps et de ressources. Ce projet, en phase avec un ensemble de données réaliste et fortement applicable, nous a permis d'explorer des méthodes de traitement efficaces pour les données déséquilibrées, tout en consolidant les modèles de prédiction adaptés aux environnements réels.

A Annexe 1 : Analyse descriptive

A.1 Analyse catégorielle

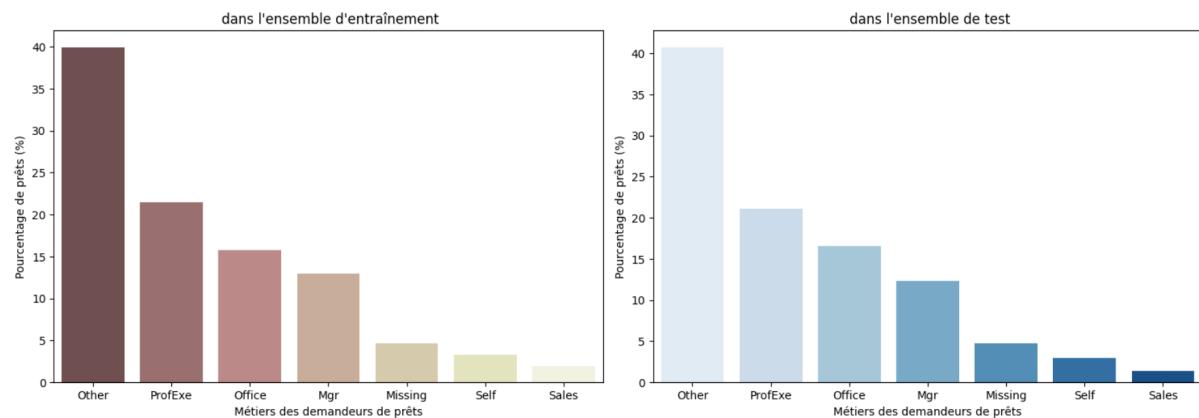


FIGURE 32 – Répartition des métiers dans les jeux de test et d'entraînement

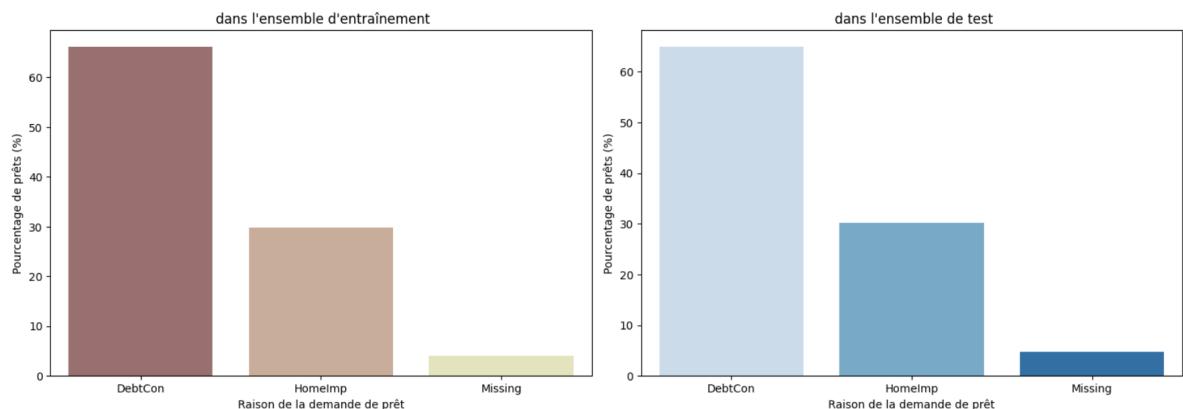


FIGURE 33 – Répartition des raisons dans les jeux de test et d'entraînement

A.2 Visualisations des distributions

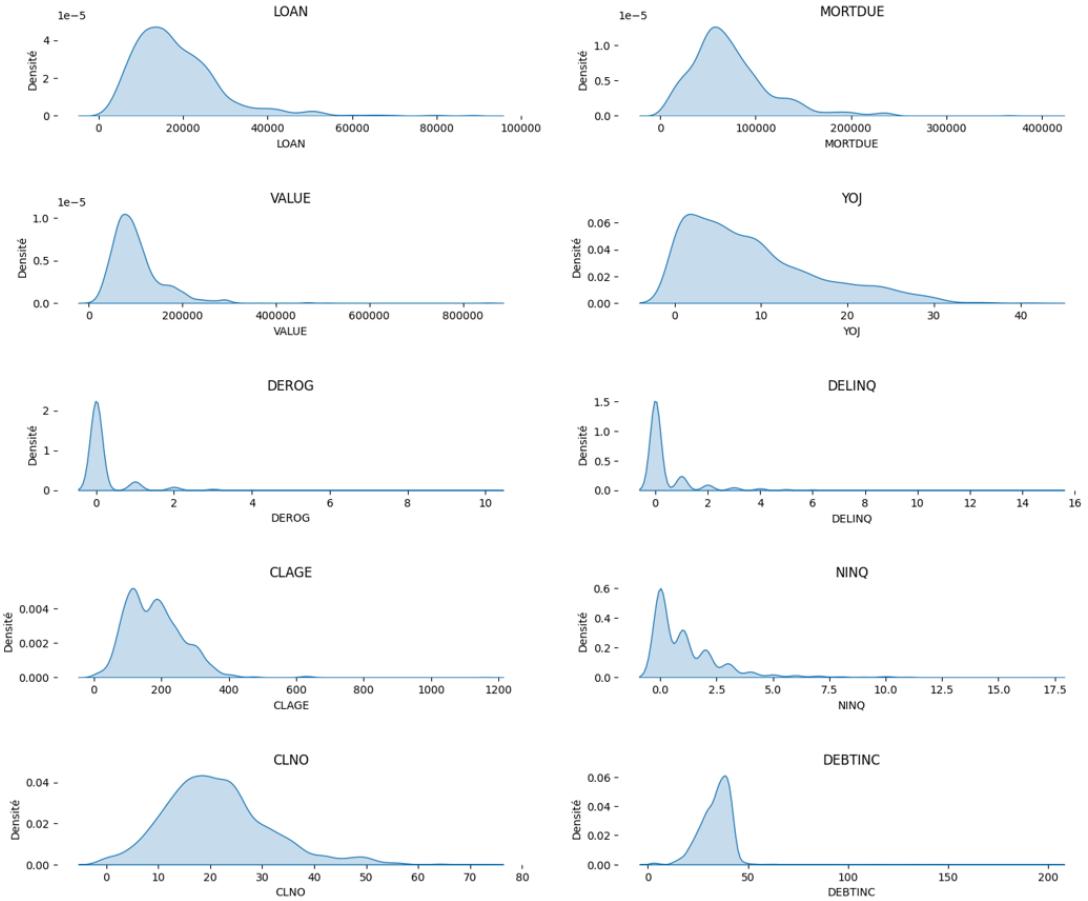


FIGURE 34 – Distribution des variables continues

A.3 Taux de défaillance et caractéristiques

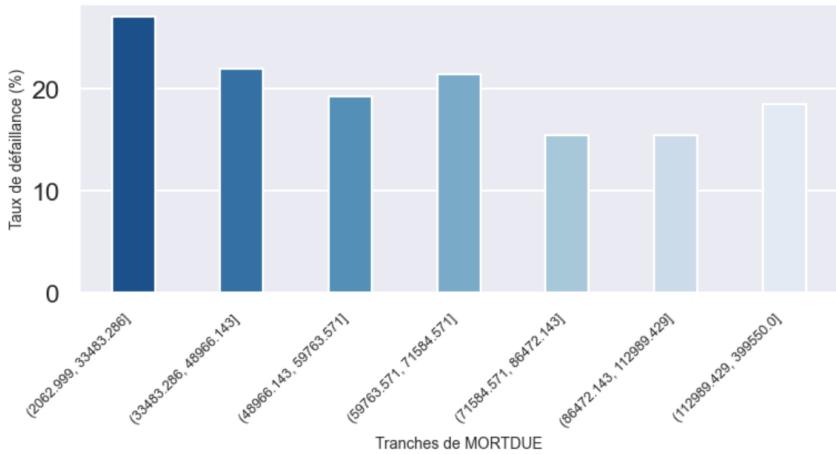


FIGURE 35 – Taux de défaillance par tranche de Mortdue

Dette Hypothécaire et Risque de Défaillance : Une Réalité Nuancée Le taux de défaillance le plus élevé chez les emprunteurs avec de petites hypothèques suggère un profil

financier fragile. À l'inverse, pour les hypothèques supérieures, le risque diminue, indiquant des ressources plus solides et une meilleure gestion des engagements. Cela révèle une corrélation entre solidité financière et capacité à gérer des dettes élevées, probablement renforcée par des critères de prêt plus stricts pour les grandes hypothèques.

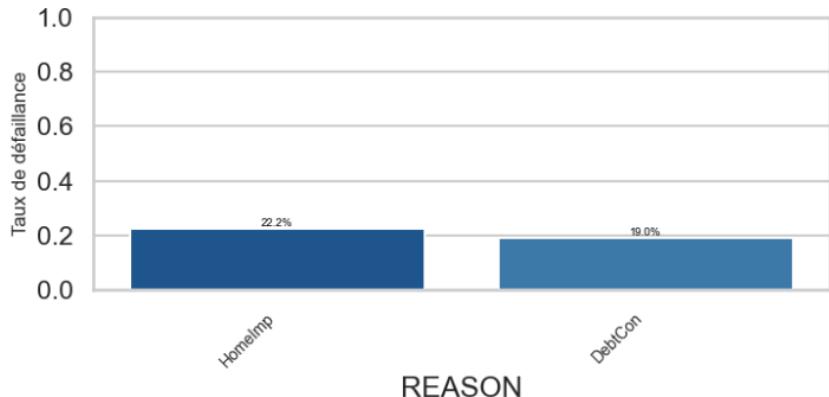


FIGURE 36 – Taux de défaillance par raison

Raison du prêt et défaillance : La consolidation de dettes, un facteur aggravant ?
Les emprunteurs consolidant leurs dettes (DebtCon) ont un taux de défaillance similaire à ceux empruntant pour des améliorations domiciliaires (HomeImp), remettant en question l'idée d'un risque accru pour la consolidation. Bien que déjà endettés, ils cherchent peut-être à stabiliser leur situation à court terme. Les prêts pour améliorations, bien qu'investissements, peuvent causer un surendettement si les coûts sont mal estimés. Une analyse avec le ratio dette/revenu pourrait approfondir ces tendances.

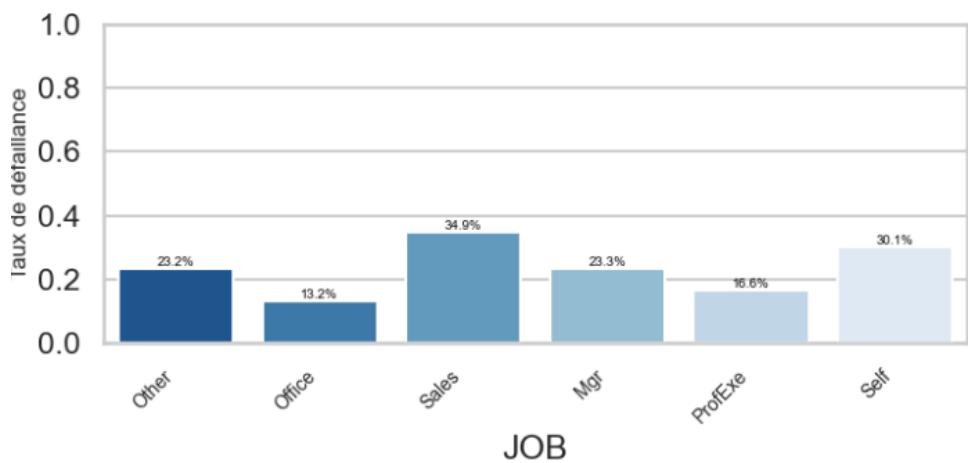


FIGURE 37 – Taux de défaillance par emploi

Catégories socioprofessionnelles : Un indicateur de stabilité financière ? Les cadres (ProfExe) et employés de bureau (Office) ont des taux de défaut bas, grâce à une meilleure stabilité d'emploi et des ressources solides. En revanche, les vendeurs (Sales) et indépendants (Self) montrent des taux plus élevés, dus à une instabilité financière et des revenus volatils. Une analyse de la durée d'activité pourrait affiner ces résultats, car les indépendants avec une longue expérience sont souvent plus résilients.

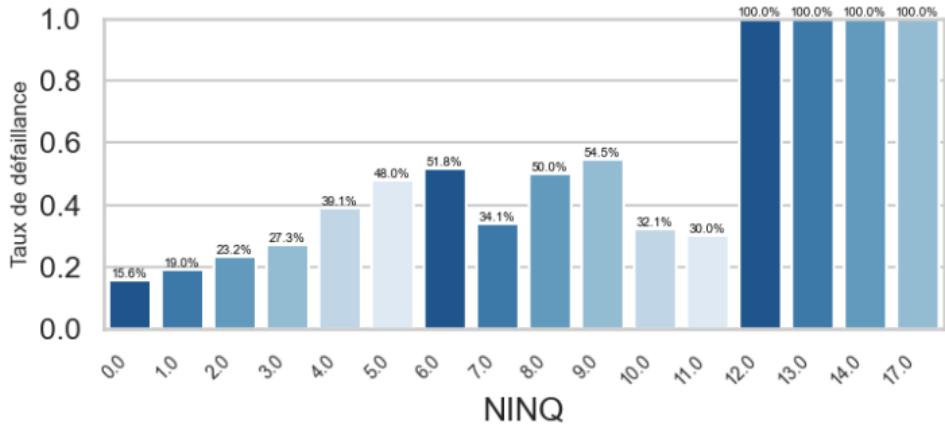


FIGURE 38 – Taux de défaillance par NINQ

A.4 Taux de défaillance : lissage par moyenne mobile

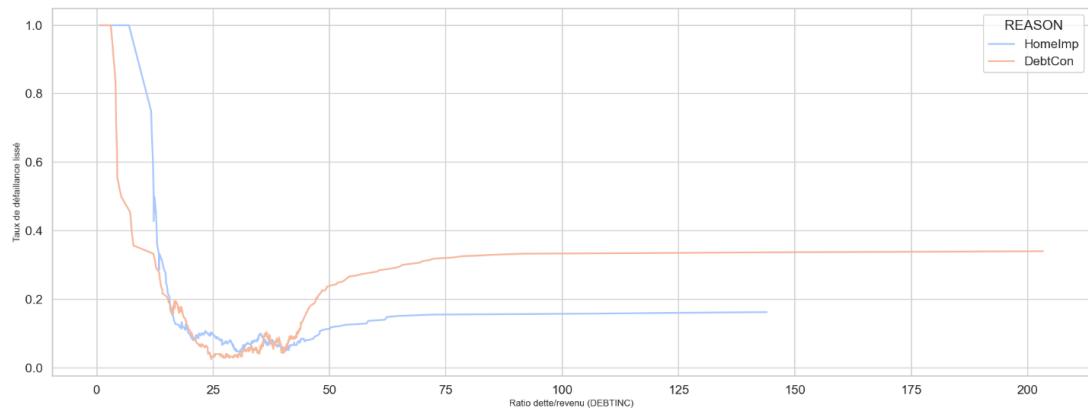


FIGURE 39 – Taux de défaillance selon REASON et DEBTINC

Interaction entre le ratio dette/revenu et la raison du prêt : Une autre clé du risque
L'analyse du ratio dette/revenu et de la raison du prêt montre que les emprunteurs consolidant leurs dettes ont des taux de défaillance élevés, surtout si leur ratio dépasse 40 %, suggérant que la consolidation ne fait que retarder les difficultés. En revanche, ceux empruntant pour des améliorations domiciliaires sont plus stables, même avec des ratios élevés, car ils investissent dans un actif renforçant leur sécurité financière. Cette analyse souligne l'importance de prendre en compte la raison du prêt et le niveau d'endettement pour évaluer le risque de défaillance.

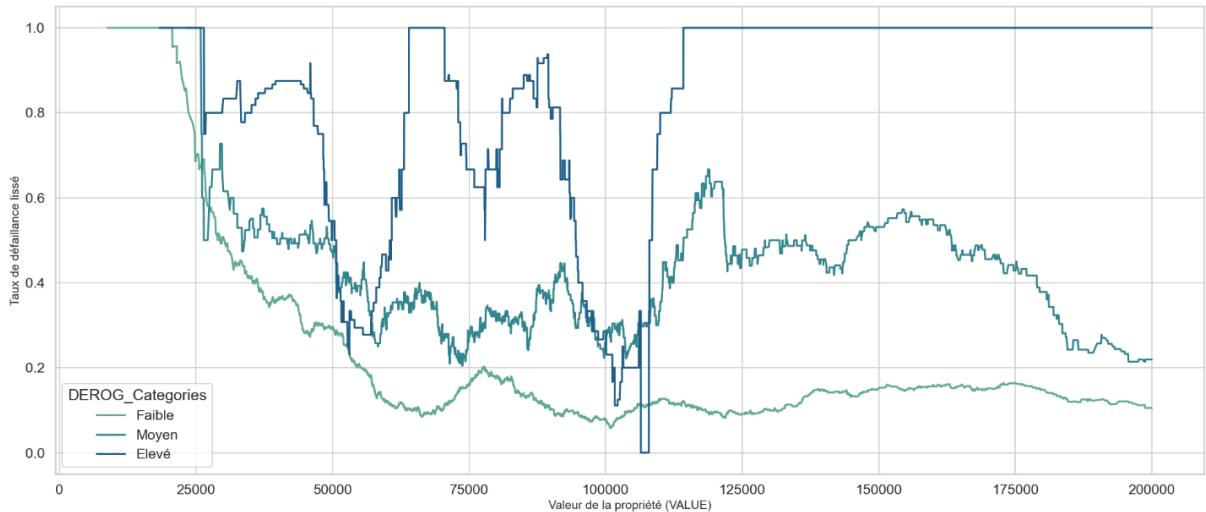


FIGURE 40 – Taux de défaillance selon DEROG et VALUE

Effet combiné des rapports dérogatoires et de la valeur de la propriété Pour les emprunteurs avec de nombreux rapports dérogatoires, le taux de défaillance reste élevé, peu importe la valeur de leur propriété, soulignant que les antécédents de crédit négatifs sont un prédicteur majeur du défaut. En revanche, pour ceux avec peu ou pas de rapports dérogatoires, la relation est plus linéaire : les emprunteurs avec des biens de grande valeur (plus de 150 000) ont des taux de défaillance très bas. Cela confirme que la possession d'actifs de valeur réduit significativement le risque de défaut, surtout pour ceux avec un historique de crédit sain.

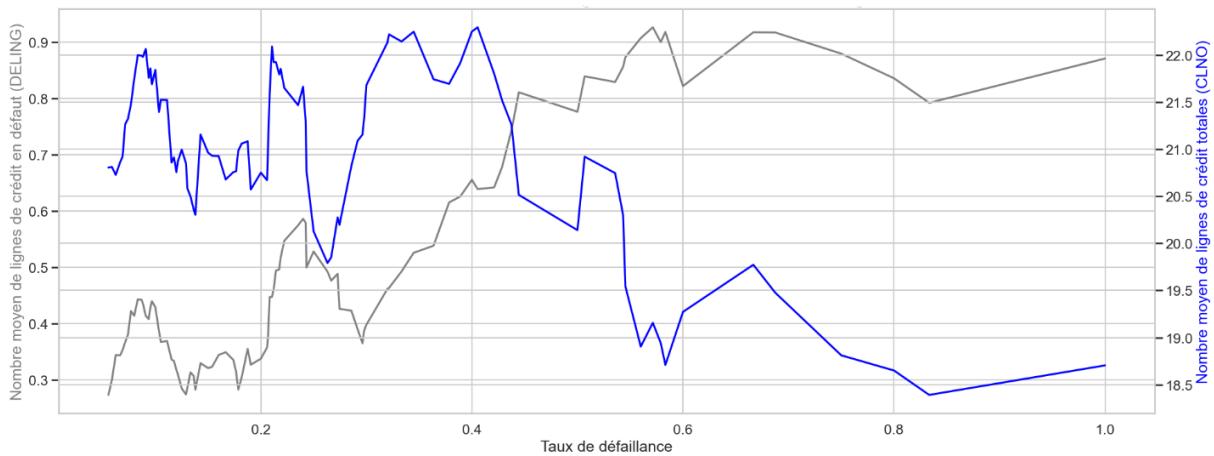


FIGURE 41 – Taux de défaillance, CLNO et DELINQ

Lignes de crédit et défaut : L'effet cumulatif de la dette Le nombre total de lignes de crédit diminue avec le taux de défaillance, suggérant que les emprunteurs en difficulté ont moins accès à de nouveaux crédits. En revanche, le nombre de lignes de crédit en défaut augmente, indiquant que ces emprunteurs voient leurs lignes existantes défaillantes. Ce résultat invite à réévaluer l'importance de la concentration des dettes dans les modèles de scoring : le nombre total de crédits n'est pas un indicateur optimal, mais les défauts partiels sur certaines lignes sont un fort prédicteur de défauts généralisés.

B Annexe 2 : Pre-processing

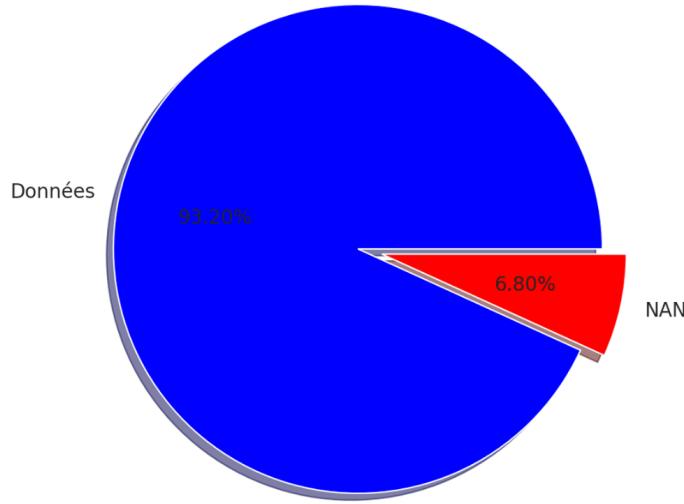


FIGURE 42 – Taux de remplissage

Taux de valeurs manquantes dans l'ensemble des données.

C Annexe 3 : Analyse des features importantes dans les modèles Gradient Boosting

Pour le modèle XGBoost, nous nous basons sur quatre mesures d'importance des features : Gain, Weight, Cover, et Total Gain. Chaque mesure offre une perspective différente sur la manière dont chaque feature contribue au modèle.

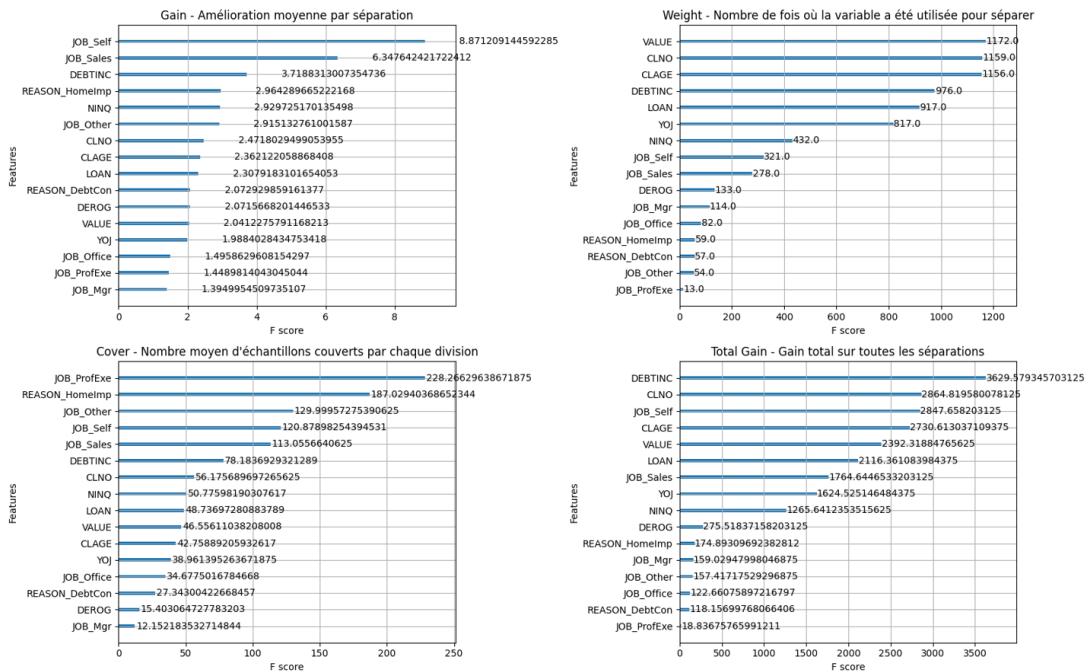


FIGURE 43 – Comparaison de l'importance des features dans XGBClassifier

— Gain - Amélioration moyenne par séparation

La feature **JOB_Self** présente la valeur de Gain la plus élevée (8.87), ce qui indique qu'elle apporte la plus grande amélioration moyenne pour le modèle lors de la construction des arbres. Les features suivantes comme **JOB_Sales**, **DEBTINC**, **REASON_HomeImp** et **NINQ** contribuent également de manière significative à l'amélioration de la précision des séparations. Un gain élevé signifie que ces features fournissent beaucoup d'informations et jouent un rôle important dans la prédiction de l'objectif (par exemple, le risque de défaut).

— Weight - Nombre de fois où la feature a été utilisée pour séparer

La feature **VALUE** est la plus souvent utilisée pour les séparations (1172 fois), suivie par **CLNO** (1159), **CLAGE** (1156), **DEBTINC** (976), et **LOAN** (917). Le Weight indique le nombre de fois que chaque feature a été sélectionnée pour séparer dans les arbres, c'est-à-dire la fréquence d'utilisation des features dans la construction du modèle. Des features comme **VALUE**, **CLNO**, et **CLAGE** ont un Weight élevé, ce qui montre qu'elles sont fréquemment utilisées et sont importantes pour la classification des échantillons.

— Cover - Nombre moyen d'échantillons couverts par chaque division

La feature **JOB_ProfExe** a le Cover le plus élevé (228), suivie par **REASON_HomeImp** (187), **JOB_Other**, et **JOB_Self**. Le Cover indique le nombre moyen d'échantillons que chaque séparation couvre. Les features ayant un Cover élevé couvrent généralement un grand nombre d'échantillons lors d'une séparation, ce qui aide le modèle à différencier de larges groupes dans les données. Les features comme **JOB_ProfExe** et **REASON_HomeImp** ont un Cover élevé mais un Weight faible, ce qui signifie qu'elles sont rarement choisies pour la séparation mais, lorsqu'elles le sont, elles affectent un grand nombre d'échantillons.

— Total Gain - Gain total sur toutes les séparations

La feature **DEBTINC** a le Total Gain le plus élevé (3629), suivie par **CLNO**, **JOB_Self**, **CLAGE**, et **VALUE**. Le Total Gain est la somme de toutes les valeurs de Gain pour chaque séparation. Il représente le bénéfice total de chaque feature dans l'ensemble du modèle. Une feature avec un Total Gain élevé contribue de manière significative au modèle à travers de nombreux arbres et de nombreuses séparations. Les features **DEBTINC**, **CLNO**, **JOB_Self**, et **CLAGE** avec un Total Gain élevé indiquent qu'elles ont un impact fort et aident constamment à améliorer le modèle sur l'ensemble des arbres.

— Conclusion dans le cas de XGBoost :

- Les features **DEBTINC**, **CLNO**, **JOB**, **CLAGE**, et **VALUE** sont les plus importantes dans le modèle XGBClassifier, comme le montrent les différentes mesures d'importance (Gain, Weight, et Total Gain). Cela montre qu'elles ont un impact majeur sur la capacité de prédiction du modèle.
- **JOB_Self** a le Gain le plus élevé, ce qui en fait une feature très importante pour améliorer la précision du modèle à chaque séparation, bien que son Weight ne soit pas aussi élevé que celui des features **VALUE** ou **CLNO**.
- **VALUE**, **CLNO**, et **CLAGE** ont le plus grand nombre de fois où elles sont sélectionnées pour la séparation (Weight), ce qui montre qu'elles sont des features précieuses pour segmenter les échantillons.
- **DEBTINC** n'a pas seulement le Total Gain le plus élevé, mais elle a également un poids important dans la détermination des groupes de risque, ce qui en fait une feature très cruciale pour le modèle.

De même manière, on fait un zoom dans le modèle LightGBM :

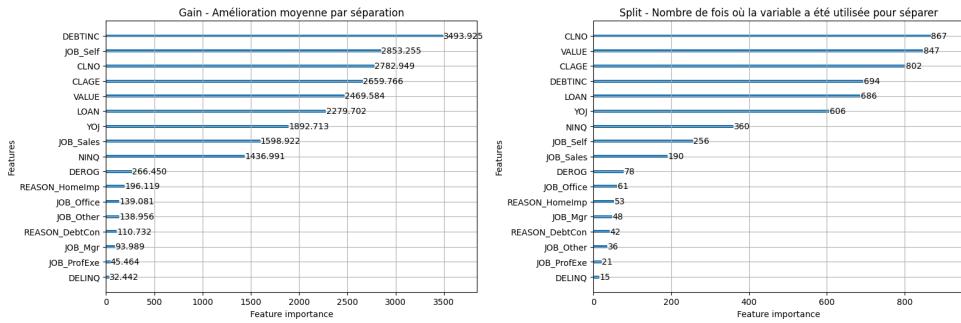


FIGURE 44 – Comparaison de l'importance des features dans LightGBM

— Gain - Amélioration moyenne par séparation

- La feature **DEBTINC** présente le gain le plus élevé (3493.925), ce qui signifie qu'elle apporte la plus grande amélioration moyenne pour la précision du modèle lors de la construction des arbres.
- Les features **JOB_Self**, **CLNO**, **CLAGE**, et **VALUE** suivent avec des gains élevés, indiquant qu'elles fournissent également une quantité importante d'information et jouent un rôle crucial dans les décisions de séparation.
- Les valeurs élevées de Gain pour ces features montrent qu'elles contribuent significativement à l'amélioration de la précision de prédiction du modèle.

— Split - Nombre de fois où la variable a été utilisée pour séparer

- La feature **CLNO** est celle qui est la plus souvent utilisée pour les séparations (867 fois), suivie de près par **VALUE** (847), **CLAGE** (802), **DEBTINC** (694), et **LOAN** (686).
- Le nombre de splits élevé pour ces features signifie qu'elles sont souvent préférées par le modèle pour diviser les données, ce qui en fait des éléments clés pour la classification des échantillons.
- Des features comme **NINQ** et **JOB_Self** apparaissent également fréquemment, bien que moins souvent, ce qui montre leur importance secondaire mais toujours significative dans le processus de séparation.

— Conclusion dans le cas de LightGBM

- Les features **DEBTINC**, **JOB_Self**, **CLNO**, **CLAGE**, **VALUE**, et **LOAN** se distinguent par leur importance élevée dans le modèle LightGBM. Elles ont soit un Gain élevé, indiquant leur contribution à l'amélioration de précision, soit un nombre de splits élevé, ce qui montre leur rôle essentiel dans les décisions de séparation.