UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Scoring

CHEKALINA Alisa, CRISTEA Carmen, GASPARIN Lia, VO Nguyen Thao Nhi

Master 2 MoSEF 2024 - 2025

Table des matières

1 Introduction

L'intérêt provenant des prêts sur valeur domiciliaire joue un rôle important dans la rentabilité d'une banque. Cependant, ces prêts peuvent être risqués, et il est essentiel pour une banque d'être judicieuse dans l'évaluation des profils de clients avant d'accepter et de procéder à un prêt quelconque. Le fait de reconnaître des clients fiables, ce qui nécessite que les experts financiers examinent manuellement les profils (ou portefeuilles) des clients et analysent ceux qui sont prometteurs et ceux qui présentent un risque, consomme énormément de temps et de ressources financières. Ce processus peut également être sujet à des erreurs de jugement en raison des erreurs humaines et des biais. Grâce aux avancées de la science des données et aux algorithmes de machine learning, la vitesse et la capacité d'analyse de grandes bases de données sont améliorées, réduisant ainsi les biais humains.

2 Jeu de données

La base de données "Home Equity" (HMEQ) contient des informations sur les prêts sur valeur domiciliaire récents. L'objectif de ce projet est de construire un modèle de classification afin de prédire si un demandeur fera défaut et de déterminer quels sont les facteurs à considérer avant que la banque n'accepte la demande de prêt. Il y a 5960 observations et 13 variables enregistrées, dont 1 variable cible et 12 variables explicatives. Parmi les variables explicatives, on trouve 2 variables qualitatives et 10 variables quantitatives.

On constate qu'il n'y a pas de doublons dans ce jeu des données.

La variable cible "BAD" est une variable binaire où "1" signifie que le demandeur a fait défaut, c'est-à-dire qu'il était en retard de paiement, et "0" signifie qu'il a remboursé le prêt. Il y a donc 1189 cas de défaut, ce qui constitue 19,9% des observations.

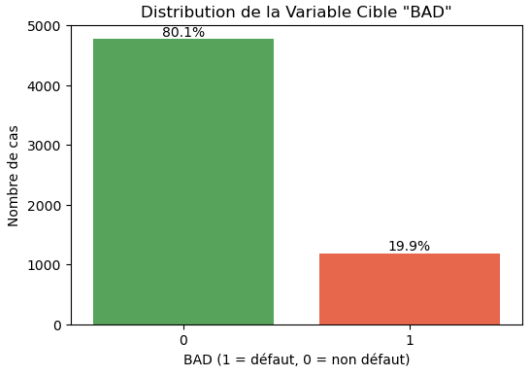


FIGURE 1 – Distribution de la Variable Cible "BAD"

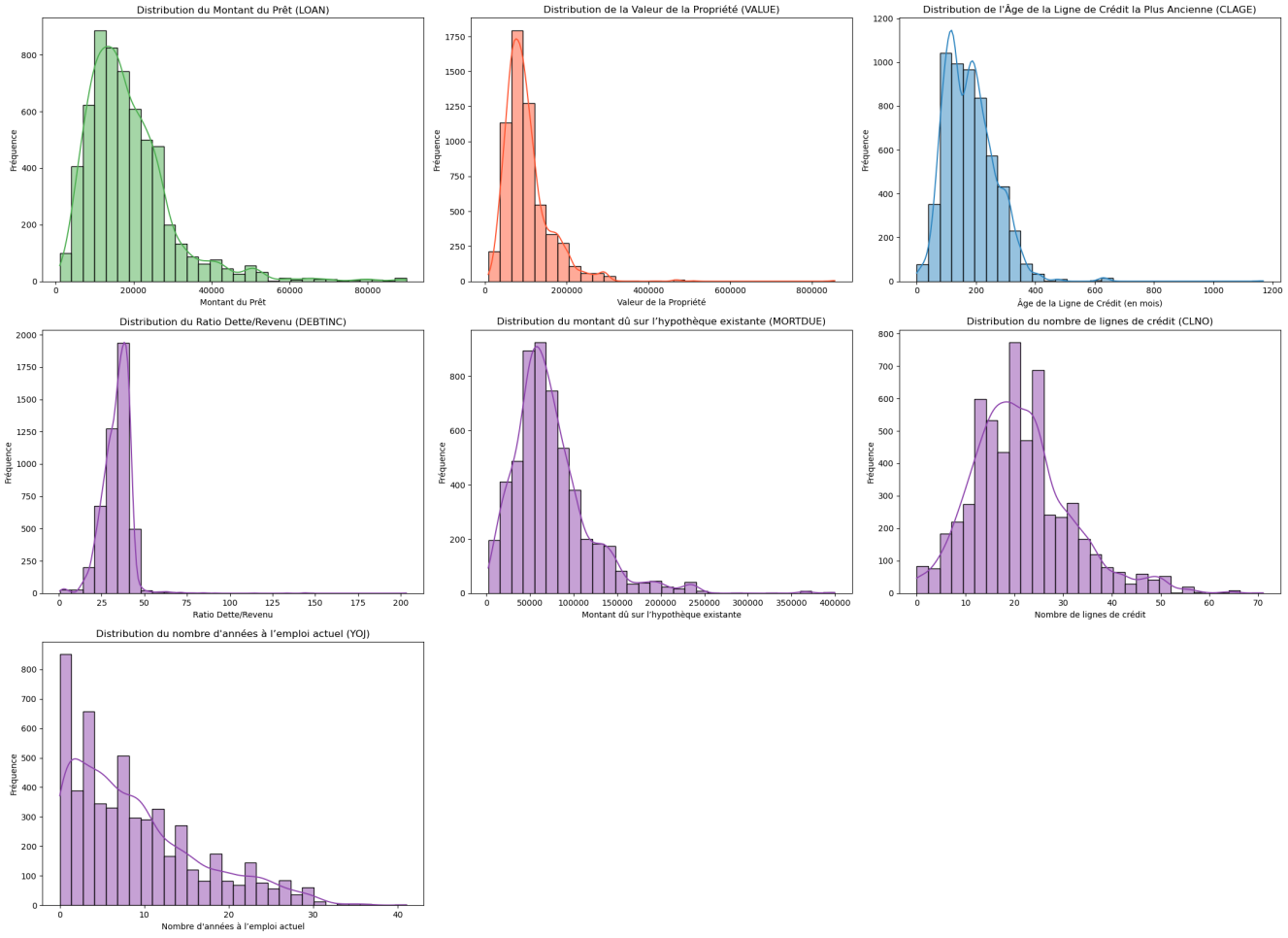
Il y a 10 variables quantitatives présentes dans la base de données : LOAN(Montant du prêt), VALUE (Valeur de la propriété), CLAGE (Age de la Ligne de Crédit (en mois)), DEBTINC (Ratio Dette/Revenu), MORTDUE (Montant dû sur l'hypothèque existante), CLNO (Nombre de lignes de crédit), YOJ (Nombre d'années à l'emploi actuel), DEROG (Nombre de Rapports Dérogatoires), DELINQ (Nombre de lignes de crédit en retard), NINQ (Nombre de demandes de crédit récentes).

	count	mean	std	min	25%	50%	75%	max
BAD	5960.0	0.199497	0.399656	0.000000	0.000000	0.000000	0.000000	1.000000
LOAN	5960.0	18607.969799	11207.480417	1100.000000	11100.000000	16300.000000	23300.000000	89900.000000
MORTDUE	5442.0	73760.817200	44457.609458	2063.000000	46276.000000	65019.000000	91488.000000	399550.000000
VALUE	5848.0	101776.048741	57385.775334	8000.000000	66075.500000	89235.500000	119824.250000	855909.000000
YOJ	5445.0	8.922268	7.573982	0.000000	3.000000	7.000000	13.000000	41.000000
DEROG	5252.0	0.254570	0.846047	0.000000	0.000000	0.000000	0.000000	10.000000
DELINQ	5380.0	0.449442	1.127266	0.000000	0.000000	0.000000	0.000000	15.000000
CLAGE	5652.0	179.766275	85.810092	0.000000	115.116702	173.466667	231.562278	1168.233561
NINQ	5450.0	1.186055	1.728675	0.000000	0.000000	1.000000	2.000000	17.000000
CLNO	5738.0	21.296096	10.138933	0.000000	15.000000	20.000000	26.000000	71.000000
DEBTINC	4693.0	33.779915	8.601746	0.524499	29.140031	34.818262	39.003141	203.312149

FIGURE 2 – Statistiques descriptives des variables quantitatives

Le montant moyen du prêt est d'environ 18 608, et 50% des clients demandent un montant de prêt d'au moins 16 300. Les variables LOAN, VALUE, CLAGE, DEBTINC et MORTDUE varient largement, ce qui indique une diversité dans les demandes de prêt des clients. Le nombre d'années d'emploi actuel est en moyenne d'environ 9 ans. 50% des clients ont au moins 20 lignes de crédit au total, et le nombre de demandes de crédit récentes varie de 0 à 17, avec 50% des clients ayant au moins 1 demande de crédit récente. Le nombre de rapports dérogatoires varie de 0 à 10, avec au moins 50% des clients qui n'en ont pas. Le nombre de lignes de crédit en retard varie de 0 à 15, avec au moins 50% des clients qui n'en ont pas.

Nous pouvons remarquer qu'il y a une distribution asymétrique vers la droite pour toutes les variables quantitatives sauf DEBTINC, avec les moyennes sont plus grandes que les médianes.



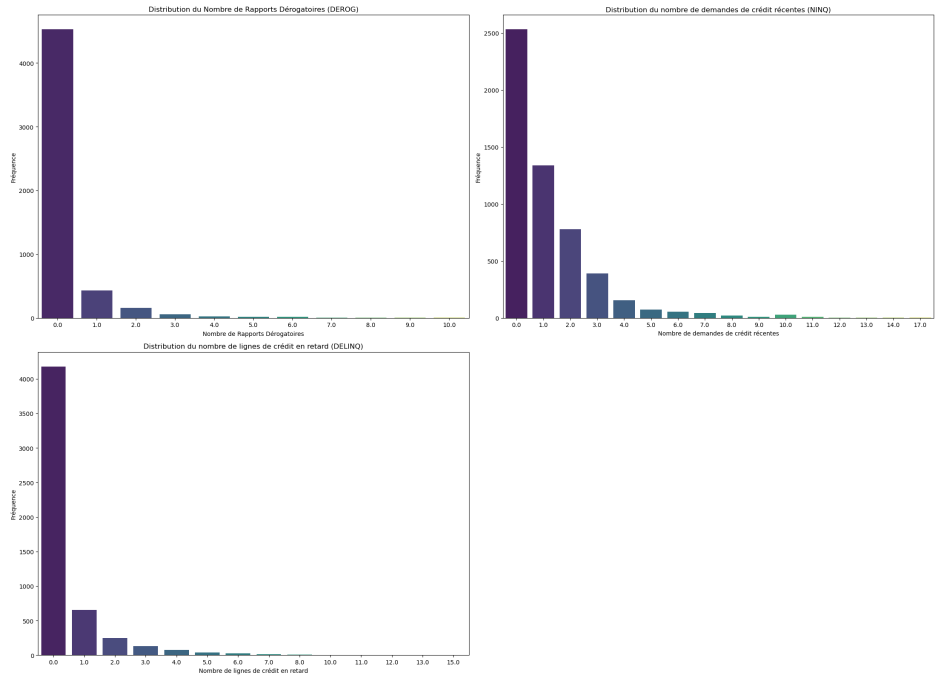


FIGURE 3 – Distribution des variables quantitatives

Il y a 2 variables qualitatives présentes dans la base de données.

- REASON : Raison du prêt. Il y a deux modalités : HomeImp (amélioration de la maison) et DebtCon (consolidation de dettes, ce qui signifie demander un nouveau prêt pour rembourser d'autres dettes personnelles)
- JOB : Catégories professionnelles. Il y a 6 modalités : Sales (vendeur), Self (indépe-ndant), Mgr (manager), Office (employé de bureau), ProfExe (professionnel/ cadre) et Other (autre)

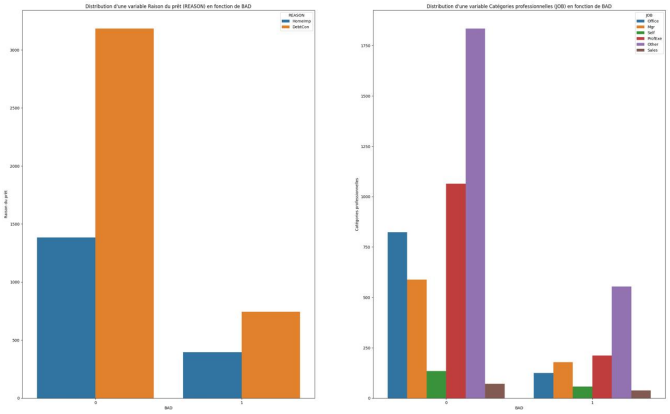


FIGURE 4 – Distribution des variables qualitatives en fonction de BAD

Au sujet de la raison du prêt, pour les clients sans défaut, la plupart des prêts sont destinés à la consolidation de dettes, tandis que l'amélioration de la maison est moins fréquente. Pour les clients en défaut, la tendance reste similaire, mais avec une proportion nettement inférieure dans les deux catégories.

En ce qui concerne les catégories professionnelles, pour les clients sans défaut, la catégorie 'Other' est la plus fréquente, suivie de 'ProfExe', tandis que les autres catégories ont une distribution plus faible. Pour les clients en défaut, la catégorie 'Other' reste prédominante, mais avec une fréquence réduite, et les autres catégories ont une présence minimale.

Concernant les relations entre les variables quantitatives, il existe une forte corrélation entre la valeur de la propriété (VALUE) et le montant dû sur l'hypothèque existante (MORTDUE), ce qui est logique, car la valeur d'une propriété est souvent liée à la taille de l'hypothèque. De plus, les clients ayant un nombre d'années d'emploi plus élevé (YOJ) et une ancienneté de ligne de crédit plus importante (CLAGE) tendent à avoir un ratio dette/revenu (DEBTINC) plus bas. Cette situation reflète une plus grande stabilité financière et une gestion plus prudente des dettes.

En ce qui concerne les variables les plus corrélées avec la variable cible BAD, qui indique un défaut de prêt, on observe que le nombre de rapports dérogatoires (DEROG) et le nombre de lignes de crédit en souffrance (DELINQ) sont particulièrement liés à un risque accru de défaut. Cela est intuitif, car les incidents dérogatoires constituent souvent des signes directs d'une mauvaise gestion du crédit, augmentant ainsi la probabilité de ne pas honorer les obligations financières à l'avenir. De même, lorsqu'un client a plusieurs paiements en retard sur ses lignes de crédit, cela révèle des difficultés dans la gestion de ses finances, augmentant ainsi la probabilité de défaut sur ses engagements financiers.

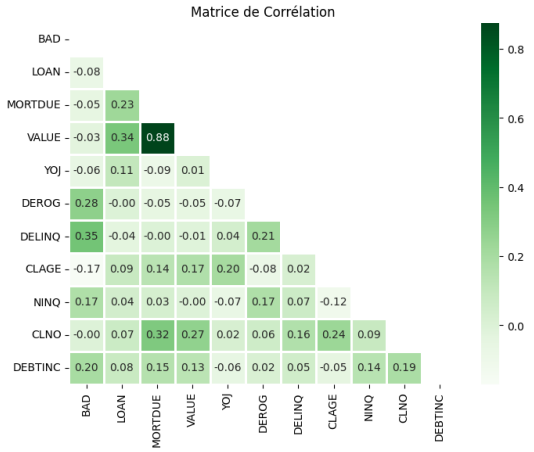


FIGURE 5 – Matrice de Corrélation

3 Preprocessing

3.1 Traitement des valeurs manquantes

Le traitement des données manquantes est une étape indispensable dans la préparation des données, car l'absence de certaines valeurs peut grandement affecter les performances de certains modèles prédictifs. Si ces données ne sont pas correctement prises en charge, cela peut entraîner des erreurs ou des dégradations significatives des résultats des modèles. Dans notre analyse, nous avons opté pour l'imputation des valeurs manquantes à l'aide de la régression via le modèle RandomForestRegressor, qui présente de nombreux avantages.

Tout d'abord, cette méthode permet de prendre en compte les relations complexes entre les différentes variables. Contrairement aux méthodes plus simples telles que l'imputation par la moyenne ou la médiane, qui ne tiennent pas compte des interactions entre les colonnes, RandomForestRegressor exploite toutes les variables disponibles pour prédire les valeurs manquantes. Cela permet de générer des imputations plus précises et plus représentatives des données réelles, car elles reposent sur les corrélations et les interactions présentes dans l'ensemble de données.

En outre, RandomForestRegressor est un modèle extrêmement flexible et robuste, capable de traiter efficacement les relations non linéaires et les interactions complexes entre les variables. Contrairement aux modèles de régression linéaire qui présupposent une relation linéaire, RandomForest capte des schémas plus subtils dans les données, ce qui améliore la précision des imputations, surtout dans des ensembles de données complexes et hétérogènes.

Un autre avantage significatif de RandomForest est sa capacité à gérer les valeurs aberrantes et la variabilité des données. Contrairement aux méthodes d'imputation basées sur la moyenne ou la médiane, souvent influencées par la présence de valeurs extrêmes, RandomForest est moins sensible aux anomalies. Grâce à sa structure fondée sur plusieurs arbres de décision, chacun utilisant un sous-échantillon des données, ce modèle offre des imputations plus fiables, même en présence d'outliers.

Cependant, il est important de noter que notre base de données contient certaines colonnes numériques dont les valeurs sont de type entier (integer). L'imputation par régression ayant généré des valeurs continues pour ces variables, nous avons pris soin d'arrondir ces valeurs afin de préserver la nature discrète de ces variables et d'obtenir une représentation plus fidèle des données originales.

Enfin, pour évaluer la qualité de l'imputation, nous avons comparé les distributions des variables avant et après l'imputation des valeurs manquantes.

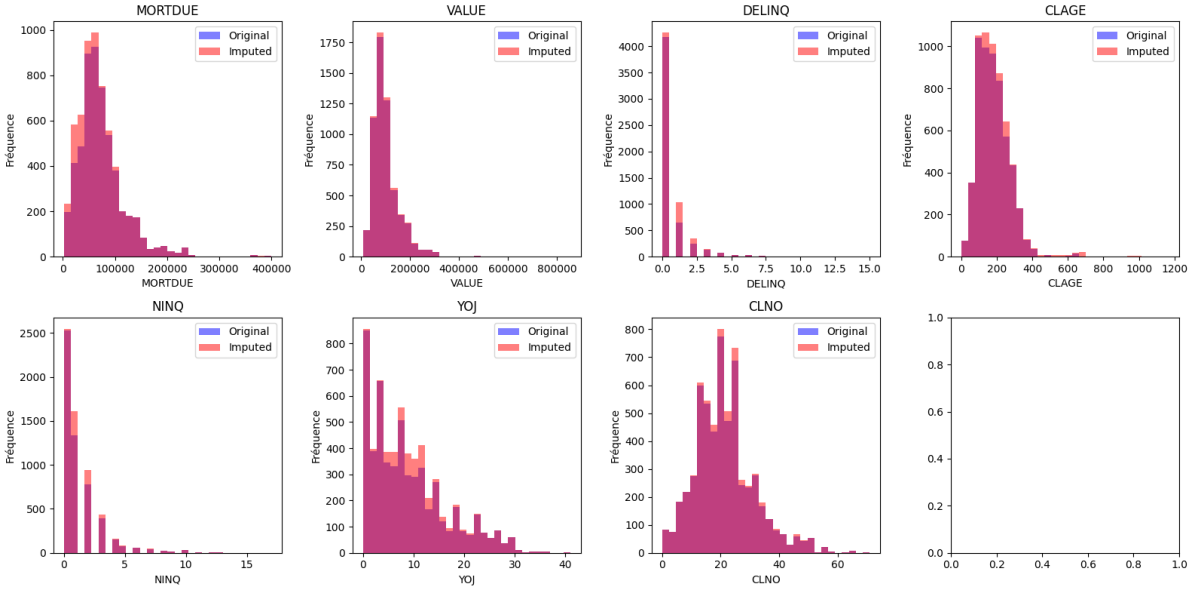


FIGURE 6 – Comparaison des distributions avant et après l'imputation

Les distributions des variables imputées sont globalement similaires aux distributions des données

originales, indiquant que l'imputation avec RandomForestRegressor a bien préservé la structure des données. Les légères différences observées, notamment aux extrêmes, sont attendues et acceptables. Les variables numériques discrètes, après arrondi, maintiennent une cohérence avec leurs distributions d'origine. Cela montre que l'approche d'imputation choisie est à la fois robuste et précise pour traiter les valeurs manquantes.

3.2 Outliers et Discrétisation

3.2.1 Outliers

Le traitement des outliers est également une étape essentielle dans la préparation des données, car la présence de valeurs aberrantes peut gravement compromettre la précision et la robustesse des modèles prédictifs. De plus, dans un modèle de scoring, il est souvent préférable de ne pas supprimer les outliers, car ils peuvent contenir des informations précieuses sur des comportements extrêmes qui sont essentiels pour l'évaluation du risque. Ces valeurs atypiques peuvent indiquer des situations de crédit à haut risque, et leur exclusion pourrait introduire un biais dans le modèle, rendant les résultats moins représentatifs de la réalité.

Nous avons remarqué qu'il y a des valeurs aberrantes pour la variable CLAGE. Précisé-ment, il y a deux observations dont l'âge est égal à 1168 et 1154 mois, soit 97 et 96 ans. Avoir des âges aussi extrêmes peut poser plusieurs inconvénients dans une analyse de données. Cela peut augmenter la moyenne d'âge de l'échantillon, donnant une image déformée de l'âge moyen des participants. De plus, les modèles statistiques peuvent devenir biaisés en raison de ces valeurs aberrantes, affectant les coefficients estimés et réduisant la fiabilité des prédictions.

3.2.2 Discrétisation

La discrétisation d'une variable dans un modèle de scoring, comme dans le cadre de la prédiction de défaut de prêt, présente plusieurs avantages, notamment pour la gestion des outliers ainsi que pour la stabilité et la simplification du modèle. Cette approche permet de limiter l'impact des outliers en regroupant les valeurs dans des classes, de sorte que leur influence est atténuée dans les calculs, rendant le modèle plus stable et moins sensible à ces valeurs rares, améliorant ainsi la robustesse face aux nouvelles données. Par ailleurs, le fait de diviser les variables en catégories, souvent ordinales, facilite la création d'une grille de score basée sur des règles fixes par catégorie, ce qui sera présenté par la suite.

Dans ce contexte, nous appliquons la discrétisation à la variable JOB, afin de transformer ses nombreuses catégories en groupes plus représentatifs par rapport à la cible. Pour vérifier la pertinence de ce regroupement, nous utilisons le coefficient de Tschuprow, qui mesure l'association entre deux variables qualitatives. Sa formule est donnée par

$$T = \sqrt{\frac{\chi^2}{n \cdot \sqrt{(r-1)(c-1)}}} \tag{1}$$

où :

— χ^2 est la statistique du chi-carré calculé à partir d'une table de contingence entre la variable discrétisée et la cible. Il mesure la différence entre les fréquences observées et attendues pour chaque combinaison de catégories.

— n est le nombre total d'observations dans l'échantillon.

— r est le nombre de lignes dans la table de contingence (ou le nombre de catégories de la variable discrétisée).

— c est le nombre de colonnes dans la table de contingence (en général, $c = 2$ si la variable cible est binaire, comme dans de nombreux modèles de scoring).

Avant la discrétisation, le coefficient de Tschuprow pour la variable JOB était de 0,071. Après la discrétisation, ce coefficient passe à 0,084, montrant une légère amélioration de l'association entre JOB et la variable cible grâce au regroupement des catégories. Bien que cette augmentation soit modeste, elle indique que la discrétisation a renforcé la corrélation entre JOB et le taux de défaut.

Cependant, même après discrétisation, la valeur du coefficient reste faible, ce qui révèle une corrélation limitée entre JOB et la variable cible. Cela peut indiquer que JOB n'a pas une grande influence sur la prédiction de la cible et qu'elle apporte peu d'information discriminante pour différencier les classes de défaut.

Pour les variables quantitatives, nous avons choisi d'utiliser la libraire OptimalBinning qui utilise des techniques d'optimisation avancées visant à maximiser la capacité de séparation d'une variable par rapport à la cible. Cette approche utilise les concepts de Weight of Evidence (WOE) et d'Information Value (IV). Le WOE permet de transformer des variables continues en classes discrètes en fonction de la similarité de la distribution de la variable cible, c'est-à-dire le nombre d'événements et de non-événements. Cela améliore la robustesse du modèle en réduisant l'impact des valeurs aberrantes. Il est calculé pour chaque catégorie d'une variable discrétisée en utilisant la formule suivante :

$$\text{WOE} = \ln\left(\frac{\%\text{ de bons}}{\%\text{ de mauvais}}\right) \tag{2}$$

— % de bons : proportion de clients qui n'ont pas fait défaut dans une catégorie donnée.

— % de mauvais : proportion de clients qui ont fait défaut dans cette même catégorie.

De plus, l'IV mesure la capacité d'une variable à discriminer entre les classes, ce qui aide à sélectionner celles qui sont les plus pertinentes pour le scoring.

$$\text{IV} = \sum_{i=1}^{n} (\%\text{ de bons}_i - \%\text{ de mauvais}_i) \times WOE_i \tag{3}$$

Alors, OptimalBinning construit des intervalles qui minimisent la perte d'information tout en améliorant la robustesse du modèle, grâce à une discrétisation qui capte la relation entre la variable et la cible. Pourtant, chaque classe doit avoir suffisamment d'observations (donc au moins 5% de la taille totale de l'échantillon) pour que les estimations soient stables et représentatives, afin d'éviter les problèmes d'indétermination qui pourraient survenir si une classe est trop petite ou vide. Cette méthode impose des contraintes pour chaque bin, comme un nombre minimum d'observations ou une taille minimale pour chaque intervalle, afin de s'assurer que les bins ont une représentativité statistique suffisante. Cela permet de réduire l'impact des valeurs rares ou des outliers, qui peuvent autrement nuire à la robustesse du modèle.

4 Modélisation

Dans le cadre du projet, nous avions pour objectif de construire un modèle permettant d'expliquer le défaut de paiement (« BAD ») pour des prêts garantis par l'équité des biens immobiliers à partir des autres variables. Comme dit précédemment, la base de données HMEQ comporte 5 960 observations de prêts immobiliers, avec des informations détaillées sur les emprunteurs et leurs antécédents financiers. L'objectif est donc de modéliser cette variable cible « BAD » (1 = défaut et 0 = remboursement sans incident) en fonction des variables explicatives comme le montant du prêt, l'encours hypothécaire, la valeur du bien, le ratio dette/revenu, l'ancienneté des lignes de crédit, entre autres.

Pour atteindre cet objectif, nous avons implémenté et comparé plusieurs modèles d'appr-entissage supervisé. Chacun de ces modèles présente des caractéristiques uniques qui peuvent être adaptées aux données complexes et aux relations non linéaires présentes dans ce notre problème.

4.1 Gestion du déséquilibre de la variable cible

La variable cible "BAD" est déséquilibrée, avec une proportion bien plus élevée de non-défauts que de défauts, comme observé dans l'analyse exploratoire des données. Contrairement aux pratiques courantes en machine learning, le rééchantillonnage est rarement utilisé en scoring de crédit, car il peut altérer la distribution naturelle des classes et fausser les résultats. Nous avons donc choisi de travailler avec cette distribution réelle sans appliquer de rééchantillonnage. Pour évaluer les performances de nos modèles dans ce contexte, nous nous concentrons sur des métriques adaptées aux données déséquilibrées, telles que l'AUC-ROC, la Log Loss et le rappel. L'AUC-ROC est une mesure de discrimination qui permet d'évaluer la capacité du modèle à distinguer les cas de défaut des non-défauts, ce qui est essentiel dans un contexte de scoring de crédit. La Log Loss, quant à elle, mesure la précision des probabilités prédites et pénalise fortement les erreurs de classification, surtout lorsqu'un risque élevé est incorrectement attribué. Enfin, le rappel nous aide à minimiser les faux négatifs, en s'assurant que le modèle détecte efficacement les cas de défaut. Ces métriques offrent une vision complète des performances et de la précision du modèle dans un contexte de données déséquilibrées.

4.2 Choix des variables

Notre approche de sélection de variables combine deux concepts statistiques importants : le Variance Inflation Factor (VIF) et la corrélation linéaire entre les variables explicatives et la variable cible. L'objectif est de conserver les variables les plus pertinentes tout en évitant les effets indésirables dus à la colinéarité. Le VIF est une mesure qui quantifie l'ampleur de la multicolinéarité d'une variable explicative au sein d'un ensemble de variables. Plus précisément, il indique dans quelle mesure la variance estimée d'un coefficient de régression est augmentée en raison de la corrélation avec les autres variables du modèle. Une valeur de VIF supérieure à un seuil (typiquement 5) indique une forte colinéarité, ce qui peut nuire à l'interprétation des coefficients de régression. Le calcul du VIF repose sur la régression de chaque variable explicative sur les autres, et l'inverse de l'indice de tolérance (1 - R^2) est utilisé pour estimer cette inflation de variance.

$$\mathrm{VIF}(X_i) = \frac{1}{1 - R_i^2} \tag{4}$$

R_i^2 est le coefficient de détermination de la régression de X_i sur les autres variables.

L'inverse du VIF est connu sous le nom de tolérance. Le VIF ou la tolérance peuvent être utilisés pour détecter la multicolinéarité, selon les préférences personnelles. La corrélation linéaire, quant à elle, mesure la relation entre deux variables, ici entre chaque variable explicative et la variable *BAD*. Le coefficient de corrélation varie entre -1 et 1. Une corrélation proche de 1 ou -1 indique une forte relation linéaire positive ou négative respectivement, tandis qu'une valeur proche de 0 indique une faible relation. Le calcul de la corrélation permet d'identifier les variables ayant le plus d'influence directe sur la variable cible. La méthode mise en place pour sélectionner les variables repose sur une combinaison des concepts précédents. Le processus s'effectue de la manière suivante :

1. Calcul du VIF pour chaque variable : Cela permet d'identifier les variables potentiellement problématiques à cause de la multicolinéarité.
2. Calcul de la corrélation de chaque variable avec la variable cible : Cette étape permet d'identifier les variables qui influencent le plus fortement la cible.
3. Fusion des informations VIF et corrélation : Ces deux indicateurs sont combinés afin d'évaluer à la fois la multicolinéarité et la pertinence des variables pour la prédiction.
4. Suppression des variables avec VIF élevé et faible corrélation : Dans une boucle, on supprime successivement les variables ayant le VIF le plus élevé, en priorisant celles qui ont la plus faible corrélation avec la variable cible. Cela permet de réduire la redondance des informations sans sacrifier la qualité prédictive du modèle.

Ce processus est itératif car après chaque suppression d'une variable, les VIFs et les corrélations sont recalculés pour ajuster le modèle en fonction des nouvelles relations entre les variables restantes.

Pour la base de données discrétisée, plusieurs variables ont été éliminées en raison de valeurs de VIF élevées et de corrélations faibles ou modérées. La variable VALUE binned a été retirée en premier, avec un VIF de 22,34 et une corrélation de -0,0850. Ensuite, la variable YOJ binned a été supprimée suivie de CLNO binned. La variable LOAN binned a également été éliminée, présentant un VIF de 8,08 et une corrélation de -0,1136. Par la suite, MORTDUE binned a été retirée avec un VIF de 6,25 et une corrélation de -0,0791 ainsi que la variable DEROG binned avec un VIF de 6,02 et une corrélation de 0,2436. Enfin, DEBTINC binned a été éliminée en raison d'un VIF de 5,01 et d'une corrélation de 0,2754.

Pour la base de données avec les variables continues standardisées, moins de variables ont été éliminées, mais celles qui l'ont été présentaient des problèmes de multicolinéarité plus marqués. La variable CLAGE YEAR a été supprimée en premier en raison d'un VIF infini et d'une corrélation de -0,1089, indiquant une dépendance très forte avec d'autres variables (notamment la variable CLAGE dont elle découle). Ensuite, la variable VALUE a été retirée avec un VIF de 5,05 et une corrélation de -0,0195, ce qui montre qu'elle contribuait également à la multicolinéarité du modèle, bien qu'à un degré moindre.

4.3 Choix des modèles à explorer

Pour expliquer la variable cible binaire "BAD" (qui indique si le demandeur a fait défaut ou s'il est en retard de paiement ou pas) et après un traitement approfondi de notre base de données, nous avons sélectionné plusieurs modèles de classification. L'objectif est double : d'une part, être en

mesure de prédire avec précision si un demandeur, en fonction de ses caractéristiques, fera défaut ou non ; d'autre part, identifier des patterns statistiques qui permettent une meilleure interprétation de la relation entre la variable cible (le défaut) et les facteurs explicatifs. En combinant ces deux objectifs, nous cherchons à créer un modèle de scoring de crédit performant tout en assurant une compréhension approfondie des variables qui influencent le risque de défaut.

Les modèles choisis pour cette analyse incluent la régression logistique, le Naive Bayes, les k-plus proches voisins (KNN), les arbres de décision, les forêts aléatoires, le Gradient Boosting, et les machines à vecteurs de support (SVM). Cette sélection de modèles nous permet d'explorer plusieurs approches afin de répondre à nos objectifs et d'exploiter au mieux les données disponibles. Dans un premier temps, nous avons analysé les modèles paramétriques. Cette catégorie inclut la régression logistique, un modèle de référence pour les variables binaires, et le Naive Bayes. Après, nous avons étudié le Machine à vecteurs de support (SVM). Bien que paramétrique, le SVM se distingue par l'utilisation de noyaux (ou kernels) qui permettent de gérer les séparations non linéaires entre les classes.

Par la suite, nous nous sommes penchés sur les modèles non paramétriques.

Dans cette catégorie, on retrouve le K-Nearest Neighbors (KNN) et les arbres de décision. Ces modèles sont également bien adaptés pour notre classification binaire, car ils permettent de capturer des relations non linéaires entre les variables explicatives et la cible. Nous avons également exploré les modèles d'ensemble qui se basent sur des techniques de combinaison de plusieurs modèles pour améliorer la robustesse. Les forêts aléatoires (Random Forest) et le Gradient Boosting en sont des exemples.

L'utilisation de ces modèles est justifiée par leur complémentarité. Les modèles paramé-triques offrent une interprétation directe et permettent d'estimer l'impact de chaque facteur explicatif sur la probabilité de défaut. D'autre part, les modèles non paramétriques et les modèles d'ensemble permettent de capturer des relations plus complexes et non linéaires entre les variables, ce qui est souvent nécessaire pour une meilleure précision dans la prédiction. Enfin, le SVM apporte une approche différente pour discriminer les classes, tout en étant adapté aux données déséquilibrées. Pour rester concis dans ce rapport, nous nous concentrerons sur les modèles de référence suivants : la régression logistique, retenue pour son interprétabilité, ainsi que le Random Forest et le Gradient Boosting, choisis pour leurs hautes performances prédictives.

4.3.1 Naive Bayes

Le modèle Naïve Bayes est une approche populaire pour la classification, notamment dans des cas où l'on souhaite prédire un événement binaire comme le défaut de paiement ou non. L'avantage principal de ce modèle réside dans sa simplicité et sa rapidité de calcul, même avec un grand nombre de variables. Néanmoins, il repose sur l'hypothèse d'indépendance conditionnelle des variables, c'est-à-dire qu'il considère que chaque variable explicative contribue de manière indépendante à la probabilité d'occurrence de l'événement (hypothèse qui rend ce modèle « naïf »). Il s'agit d'une hypothèse qui simplifie énormément les calculs mais qui est rarement vérifiée.

Le modèle calcule la probabilité conditionnelle pour chaque variable donnée la classe (BAD = 1 ou BAD = 0), puis applique le théorème de Bayes pour obtenir la probabilité totale. Par exemple,

il va estimer la probabilité qu'une personne ayant une dette élevée (DEBTINC élevé) et plusieurs lignes de crédit délinquantes (DELINQ) soit en défaut. En supposant que chaque variable est indépendante des autres, on multiplie toutes ces probabilités conditionnelles pour déterminer la probabilité que l'individu soit en défaut.

Le théorème de Bayes s'écrit de la manière suivante :

$$P(C_k \mid X) = \frac{P(X \mid C_k) \cdot P(C_k)}{P(X)} \tag{5}$$

Ici, $P(C_k \mid X)$ représente la probabilité d'appartenance à la classe C_k (par exemple, BAD = 1 pour défaut) donnée les variables explicatives X sélectionnées. $P(X \mid C_k)$ est la probabilité d'observer ces variables explicatives dans la classe C_k, tandis que $P(C_k)$ est la probabilité a priori de cette classe. Enfin, $P(X)$ est la probabilité des variables X.

Cependant, il présente aussi des limites importantes, notamment à cause de cette hypothèse d'indépendance. Dans notre cas, plusieurs variables ne sont pas réellement indépendantes. Par exemple, la valeur de la propriété actuelle (VALUE) est liée au montant de l'hypothèque (MORTDUE). Ignorer ces relations peut réduire la précision du modèle. Cette limite nous a donc poussé à réaliser une sélection de variables précise au travers d'une combinaison de technique telle que l'identification des variables colinéaires à l'aide du VIF et le calcul des corrélation entre chaque variable explicative et la variable cible « BAD ».

Pour finir, les valeurs manquantes dans les données posent un problème au modèle Naïve Bayes. Si des données sont absentes, il est difficile de calculer correctement les probabilités conditionnelles car Naïve Bayes n'a pas de méthode intégrée pour gérer ces valeurs manquantes. Il faut donc les imputer avant d'entraîner le modèle, au risque d'introduire des biais dans les résultats.

4.3.2 Régression Logistique

La régression logistique est un modèle de classification qui modélise la probabilité qu'un événement se produise. La régression logistique repose sur une transformation de la régression linéaire. Elle utilise la fonction sigmoïde pour contraindre la sortie à être une probabilité entre 0 et 1.

Soit X un vecteur de variables explicatives (ici les variables sélectionnées pour notre modélisation), β le vecteur des coefficients associés à ces variables, $P(Y = 1 \mid X)$ la probabilité que l'événement $Y = 1$ se produise, c'est-à-dire que la personne fasse défaut, et $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$, où β_0 est l'intercept, et $\beta_1, \beta_2, \ldots, \beta_n$ sont les coefficients des variables explicatives $X_1, X_2, \ldots, X_n$.

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-z}} \tag{6}$$

La régression logistique est idéale pour des problèmes de scoring. Elle est facile à interpréter et à implémenter : les coefficients associés aux variables indépendantes peuvent être traduits en termes de chances relatives d'occurrence de l'événement BAD (défaut ou non). Cependant, cette simplicité a des limites. La première que nous avons identifiée, comme pour le modèle Naive Bayes, concerne la gestion des valeurs manquantes. La régression logistique ne peut pas directement traiter des données incomplètes. Or, dans notre cas, plusieurs variables explicatives avaient des valeurs manquantes. La gestion de ces valeurs manquantes était donc un point essentiel de notre projet

car une mauvaise gestion aurait pu biaiser les prédictions. Pour finir, cette méthode est sensible à la présence de valeurs extrêmes qui peuvent fausser l'estimation des coefficients et dégrader les performances du modèle. La discrétisation des variables nous a permis de réduire l'impact des valeurs extrêmes qui peuvent affecter négativement notre modèle en influençant de manière disproportionnée les probabilités. Ces valeurs sont moins problématiques une fois que les valeurs sont regroupées dans des catégories.

4.3.3 Support Vector Machine (SVM)

Le SVM est une méthode de classification qui cherche à séparer nos deux classes (défaut/non-défaut) par un hyperplan optimal c'est-à-dire une frontière de décision qui sépare au mieux les deux classes dans un espace de caractéristiques. L'objectif est de maximiser la marge entre les classes. Le SVM est particulièrement efficace pour les problèmes où les classes sont difficiles à séparer avec une frontière linéaire. Dans notre cas, les facteurs influençant le défaut de paiement peuvent avoir des relations complexes non linéaires, ce qui rend le SVM adapté avec des noyaux comme le noyau gaussien (RBF). La première limite que nous constatons dans l'utilisation de ce modèle est son interprétabilité limitée contrairement aux modèles vus précédemment. Dans le cas du SVM, il est difficile d'interpréter directement la contribution de chaque variable, car le modèle se concentre sur la maximisation de la marge entre les classes, et les variables ne sont pas utilisées de manière additive. De plus, les SVM peuvent rencontrer des difficultés à traiter des problèmes où les classes sont fortement déséquilibrées. C'est notre cas puisque la proportion d'individus en défaut (classe BAD = 1) est beaucoup plus faible que celle des individus qui remboursent correctement leur crédit (classe BAD = 0). En effet, le modèle peut avoir tendance à favoriser la classe majoritaire (ici, ceux qui ne sont pas en défaut) au détriment de la classe minoritaire (les individus en défaut), ce qui peut conduire à des erreurs de classification, surtout sur les cas les plus intéressants (les mauvais payeurs). Pour finir, le SVM est, comme les deux premiers modèles, sensible aux outliers et aux valeurs manquantes.

4.3.4 K-plus proches voisins (KNN)

Le principe fondamental du KNN consiste à rechercher, pour chaque nouvelle observation à classer, les k voisins les plus proches parmi les exemples de l'ensemble d'apprentissage. Le modèle s'appuie ensuite sur les caractéristiques de ces voisins pour déterminer la prédiction finale. Il s'appuie sur une mesure de distance pour identifier les voisins les plus proches d'un point donné et prédire sa classe en fonction de la majorité des voisins. Le KNN est simple à implémenter et ne fait aucune hypothèse sur la distribution des données (non paramétrique). Il peut être utile pour capturer des relations locales complexes entre les variables. Toutefois, sa performance peut être dégradée par les outliers. En effet, si un ou plusieurs voisins proches sont des outliers ou des points aberrants, ils peuvent fausser la prédiction.

4.3.5 Gradient Boosting

Le gradient boosting est une méthode d'ensemble qui construit des modèles faibles de décision successifs, chaque nouveau modèle étant entraîné pour corriger les erreurs du modèle précédent (approche itérative). Le processus se poursuit jusqu'à ce que les erreurs soient suffisamment réduites. On commence par entraîner un premier modèle faible (souvent un arbre de décision) qui fait une première prédiction des valeurs cibles. À chaque itération suivante, un nouveau modèle est construit pour prédire les résidus de la prédiction précédente. Chaque nouveau modèle apprend

donc à corriger les erreurs des prédictions faites jusqu'à ce point, ce qui permet d'améliorer progressivement la performance globale du modèle. Le gradient boosting est capable de produire des modèles très précis et performants, car il affine les prédictions étape par étape en se concentrant sur les erreurs à chaque itération. En ajustant les erreurs de manière itérative, le gradient boosting peut bien fonctionner avec des jeux de données où les classes sont déséquilibrées. Cependant, le gradient boosting est très sensible aux hyperparamètres, comme le nombre d'arbres, la profondeur des arbres, et le taux d'apprentissage, pour obtenir de bonnes performances sans surapprentissage (une technique de grid search est donc préférable).

4.3.6 Arbre de Décision

Les arbres de décision sont des algorithmes d'apprentissage utilisés pour les tâches de classification et de régression. Ils fonctionnent en segmentant l'espace des caractéristiques en sous-ensembles de plus en plus petits, sur la base de règles de décision simples. Chaque nœud interne de l'arbre correspond à un test sur une caractéristique et chaque feuille de l'arbre représente une prédiction finale (dans le cas de la classification, cela peut être une classe comme 0 ou 1). Dans le cadre de notre classification binaire, l'arbre de décision essaie de diviser les données en deux groupes : les personnes qui font défaut et celles qui ne font pas défaut. Le critère de division le plus couramment utilisé pour la classification est l'entropie ou l'indice de Gini, qui mesure la pureté d'un sous-ensemble de données après la division. Un arbre profond peut capturer des relations complexes dans les données tout en restant intuitif et interprétable, mais cela peut aussi entraîner un surapprentissage où l'arbre s'adapte trop bien aux données d'entraînement mais généralise mal aux données nouvelles.

4.3.7 Forêt Aléatoire (Random Forest)

Les forêts aléatoires sont une extension des arbres de décision conçue pour remédier à certains des problèmes des arbres individuels, notamment le surapprentissage. Une forêt aléatoire est un ensemble d'arbres de décision indépendants entraînés sur différentes sous-parties des données et/ou avec des sous-ensembles de caractéristiques. L'idée clé est de générer plusieurs arbres de décision sur des échantillons aléatoires des données, puis d'agréger leurs prédictions pour prendre une décision finale. Dans notre cas, chaque arbre va prédire soit défaut, soit non-défaut, et la forêt aléatoire prendra la classe majoritaire (la prédiction la plus fréquente parmi les arbres). Ce processus de vote permet d'obtenir un modèle plus robuste et moins sujet au surapprentissage. Elle est particulièrement adaptée aux bases de données hétérogènes comme celle-ci, et fournit également des informations sur l'importance des différentes variables dans la prédiction du défaut de paiement.

4.3.8 Boosting d'Arbres de Décision (XGBoost, LightGBM)

Le boosting est une méthode d'ensemble qui construit un modèle puissant en combinant plusieurs modèles faibles (ici souvent des arbres de décisions peu profonds). Contrairement au bagging (utilisé dans les forêts aléatoires), qui entraîne plusieurs modèles en parallèle, le boosting les entraîne séquentiellement. Chaque modèle successif tente de corriger les erreurs faites par le modèle précédent. Dans notre cas, nous avons mis en place un boosting d'arbres de décision à l'aide du GradientBoostingClassifier. Le Gradient Boosting est une forme particulière de boosting qui fonctionne en minimisant une fonction de perte. Il applique un gradient de descente pour ajuster les poids des arbres successifs.

4.4 Résultats

4.4.1 Régression logistique

La régression logistique est entraînée sur un jeu de données discrétisé train, qui inclut des variables ordinales ainsi que des variables binaires. Nous avons utilisé scikit-learn pour entraîner la régression logistique, avec une recherche d'hyperparamètres. Nous avons recherché les meilleures combinaisons de pénalité, solveurs et valeurs de régularisation 'C' afin d'optimiser le modèle et d'obtenir des prédictions. Après, nous avons prédit sur notre jeu de test discrétisé en utilisant le meilleur modèle trouvé : 'C' : 0.1, 'penalty' : 'l2', 'solver' : 'lbfgs'. Ces hyperparamètres témoignent d'une régularisation modérée qui aide à prévenir le surapprentissage, en maintenant les coefficients des variables sous contrôle pour éviter des valeurs extrêmes. Le choix de la pénalité « l2 » indique une régularisation de type Ridge, qui est efficace pour gérer des variables corrélées, tandis que le solver « lbfgs » est un algorithme d'optimisation robuste, adapté aux problèmes de régression logistique et performant sur des volumes de données relativement importants.

Interprétation des métriques : Nous avons configuré notre modèle de manière à minimiser le risque de faux négatifs, c'est-à-dire les cas où le modèle prédirait un non-défaut pour des personnes qui feraient finalement défaut. Pour cette raison, nous évaluerons principalement la performance du modèle sur le jeu de test en nous concentrant sur le Recall, l'AUC-ROC et la Log Loss, en tenant compte du déséquilibre dans notre variable cible.

- Accuracy (0.718) : Le modèle est correct dans 71.8% des prédictions, mais ce n'est pas la meilleure mesure dans un contexte de données déséquilibrées, nous n'allons pas l'utiliser pour juger de la performance du modèle.
- Précision (0.381) : Seulement 38.1% des observations prédites comme "défaut" (classe 1) sont réellement des défauts. Cette précision assez faible indique que le modèle a tendance à produire des faux positifs (prédit défaut alors que c'est un non-défaut).
- Recall (0.659) : Le rappel montre que le modèle détecte 65.9% des cas de défauts réels. Cette métrique est importante dans le contexte de détection de défauts, car elle montre la capacité du modèle à éviter les faux négatifs (c'est-à-dire les défauts non détectés).
- F1-Score (0.483) : Légèrement supérieur à la précision, ce score F1 prend en compte à la fois la précision et le rappel. Il montre que le modèle a un équilibre modéré entre la détection des défauts et la précision de cette détection, bien que la performance reste moyenne.
- ROC AUC (0.744) : ça indique une capacité de discrimination correcte du modèle pour séparer les classes "défaut" et "non-défaut".
- Log Loss (0.593) : La Log Loss mesure la qualité des probabilités prédictives. Un score plus faible est préférable, et ici, la valeur de 0.593 montre que le modèle n'est pas parfaitement calibré, mais reste acceptable pour ce type de problème.

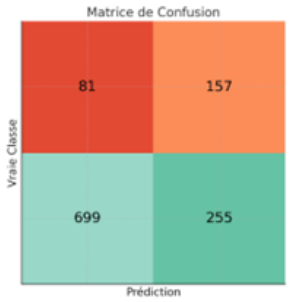


FIGURE 7 – Matrice de confusion de la régression logistique

— Vrais négatifs (699) : Les non-défauts correctement classés par le modèle.

— Faux positifs (255) : Les cas de non-défauts que le modèle a classés à tort comme défauts.

— Faux négatifs (81) : Les défauts que le modèle n'a pas réussi à détecter.

— Vrais positifs (157) : Les défauts correctement détectés par le modèle.

Au vu de ces résultats, le modèle présente une bonne capacité de rappel, ce qui signifie qu'il est relativement efficace pour détecter les cas de défauts. On pourrait qualifier ce modèle de prudent. Cependant, la précision relativement faible indique qu'il génère un nombre important de faux positifs, ce qui pourrait être problématique dans un contexte de scoring de crédit, mais dans une moindre mesure. Globalement, les résultats montrent que le modèle est assez robuste pour distinguer les défauts des non-défauts, mais il pourrait être amélioré pour réduire les faux positifs. Nous avons exploré la piste de modifier le seuil de décision pour améliorer davantage notre modèle. Cependant, les gains en performance obtenus n'étaient pas suffisamment significatifs pour justifier ce choix, qui reste en partie arbitraire.

Est-ce une performance satisfaisante ? L'objectif de cette démarche était de vérifier si notre modèle est réellement pertinent et significatif, ou s'il obtient ses résultats simplement par hasard. Pour le savoir, il nous fallait un point de comparaison pour évaluer sa performance « réelle ».

— Nous avons donc créé un modèle de référence aléatoire en permutant les valeurs de la cible (le label « défaut » = 1 ou « non-défaut » = 0) dans le jeu d'entraînement. Cela signifie que nous avons rompu toute association entre les caractéristiques des demandeurs et le risque de défaut, entraînant ainsi le modèle sur des données sans lien réel avec la cible. En répétant ce processus sur plusieurs itérations (dans notre cas n_iterations = 10 en raison de capacité limitée de calcul, sinon on aurait reconstruit la distribution empirique de chaque métrique dans le cas aléatoire), nous avons calculé la moyenne des métriques. Ces résultats, obtenus à partir de données aléatoires, servent de référence de « performance aléatoire ». En comparant les résultats de notre modèle optimisé avec ces métriques, nous pouvons juger de sa véritable efficacité : si les performances de notre modèle optimisé dépassent clairement celles du modèle entraîné aléatoirement, cela prouve qu'il capte bien des relations pertinentes dans les données et qu'il n'est pas simplement « chanceux ».

— La comparaison montre des performances nettement supérieures du modèle original, le modèle de régression logistique entraîné et calibré, par rapport au modèle aléatoire sur toutes les métriques, notamment en précision, F1-Score et AUC, ce qui indique que notre modèle capture des relations significatives dans les données.

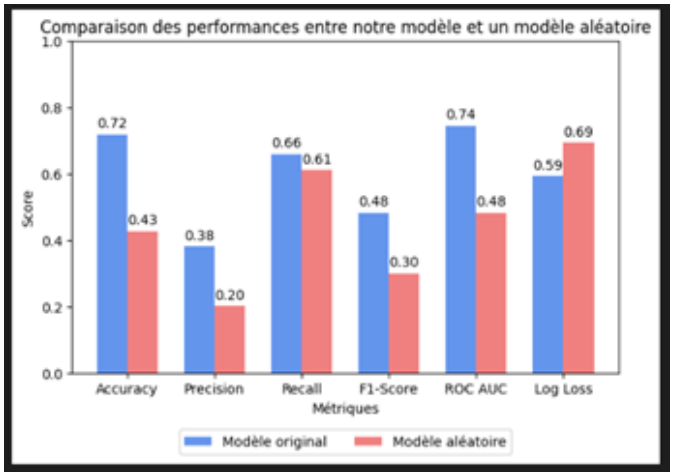


FIGURE 8 – Comparaison des performances entre notre modèle et un modèle aléatoire

Est-ce que notre modèle calibré présente de l'overfitting ? Pour évaluer si la régression logistique présente de l'overfitting, nous avons comparé les métriques de performance calculées sur le jeu d'entraînement avec celles obtenues sur le jeu de test. Si les performances sur X_train sont nettement meilleures que celles sur X_test, cela indique que le modèle s'adapte trop aux spécificités du jeu d'entraînement et généralise mal sur de nouvelles données, ce qui est un signe d'overfitting. Cette méthode permet de détecter l'overfitting en vérifiant la capacité du modèle à maintenir un niveau de performance similaire entre les deux ensembles de données.

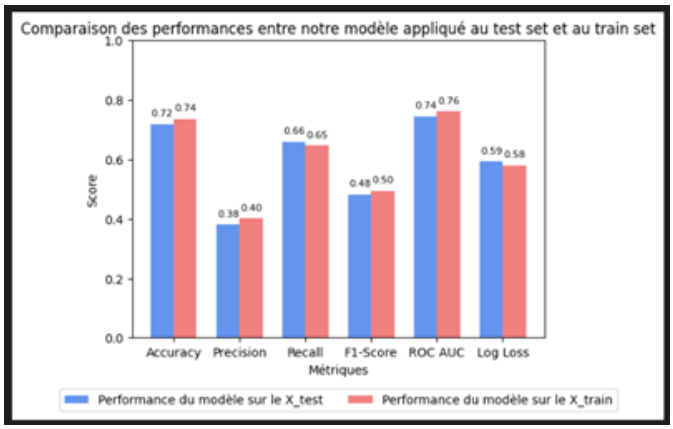


FIGURE 9 – Comparaison des performances de notre modèle entre test set et train set

Les résultats montrent que les performances du modèle sont très similaires entre le jeu d'entraînement et le jeu de test sur toutes les métriques, avec des différences minimes. Cela suggère que le modèle généralise bien et qu'il n'y a pas de signe d'overfitting. Le modèle semble donc avoir une bonne capacité à maintenir des performances constantes sur des données nouvelles, ce qui est un indicateur de sa robustesse.

```
Optimization terminated successfully.
         Current function value: 0.425361
         Iterations 6
<Wald test (chi2): statistic=[[100.78743892]], p-value=1.024036768773551e-23, df_denom=1>
                           Logit Regression Results
==============================================================================
Dep. Variable:                    BAD   No. Observations:                 4768
Model:                          Logit   Df Residuals:                     4761
Method:                           MLE   Df Model:                            6
Date:                Sat, 26 Oct 2024   Pseudo R-squ.:                  0.1487
Time:                        22:10:47   Log-Likelihood:                -2028.1
converged:                       True   LL-Null:                       -2382.3
Covariance Type:            nonrobust   LLR p-value:                9.498e-150
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const             -2.3430      0.158    -14.819      0.000      -2.653      -2.033
JOB_discret        0.3947      0.072      5.504      0.000       0.254       0.535
CLAGE_YEAR_binned -0.3785      0.033    -11.621      0.000      -0.442      -0.315
DELINQ_binned      0.7985      0.041     19.295      0.000       0.717       0.880
NINQ_binned        0.2445      0.031      7.873      0.000       0.184       0.305
CLNO_binned       -0.0508      0.030     -1.686      0.092      -0.110       0.008
REASON_HomeImp     0.3157      0.087      3.648      0.000       0.146       0.485
==============================================================================
```

FIGURE 10 – Résultats statistiques de la régression logistique

Interprétations statistiques

- Pseudo R-squared, qui est une approximation du R^2 dans le contexte de la régression logistique est égal à 14,87%. Cela indique qu'environ 14.87% de la variance de la variable cible est expliquée par le modèle. Ce résultat est typique pour des modèles logistiques sur des données complexes et il sert surtout à comparer des modèles entre eux.
- Test de Wald avec une statistique du Chi2 de 100.79 et une p-valeur associée très faible (1.02e-23), ce qui indique que le modèle global est significatif et qu'il existe au moins un coefficient différent de zéro. Cela signifie que le modèle est statistiquement valide et que les variables explicatives apportent une contribution significative à la prédiction du défaut.
- Toutes les variables sont significatives au seuil de 5%, et même de 1%, à l'exception de la variable CLNO_binned (variable discrétisée qui représente le nombre total de lignes de crédit). Celle-ci a une p-value de 9% et elle n'est pas statistiquement significative au seuil de 5%.
- Des valeurs plus élevées pour les variables DELINQ_binned (nombre accru de lignes de crédit en retard de paiement) et NINQ_binned (nombre élevé de demandes récentes de crédit), ainsi que le fait d'avoir souscrit un crédit pour des travaux d'amélioration de la maison, par opposition à une consolidation de dettes, augmentent la probabilité de faire défaut. Ces résultats sont logiques et intuitifs.
- Au contraire, des valeurs élevées pour CLAGE_YEAR_binned (ancienneté de la ligne de crédit la plus ancienne) sont associées à une probabilité réduite de faire défaut. Ce résultat est aussi intuitif.

	Odds Ratios
const	0.096040
JOB_discret	1.483951
CLAGE_YEAR_binned	0.684860
DELINQ_binned	2.222288
NINQ_binned	1.277002
CLNO_binned	0.950503
REASON_HomeImp	1.371196

FIGURE 11 – Odd Ratios

4.4.2 Random Forest

Nous avons entraîné un modèle Random Forest en ajustant certains hyperparamètres clés, notamment le nombre d'arbres dans la forêt (n_estimators), la profondeur maximale des arbres (max_depth), et le nombre minimum d'échantillons nécessaires pour diviser un nœud (min_samples_split). Ces hyperparamètres sont essentiels pour optimiser la performance et éviter le surajustement (overfitting). Il est aussi à noter que nous avons testé le Random Forest sur deux jeux de données différents : un dataset avec des variables discrétisées et un autre avec des variables quantitatives standardisées. Cela nous a permis d'évaluer l'impact de la transformation des données sur les performances du modèle et de choisir la meilleure configuration pour prédire efficacement le défaut. Sans surprise, le Random Forest fonctionne mieux sur les données quantitatives, car ce modèle utilise des seuils numvériques pour diviser les données, ce qui améliore ses performances sur des variables continues. Nous allons par la suite explorer les performances de notre modèle sur le jeu de données de test avec les variables continues standardisées.

Les résultats obtenus avec le modèle Random Forest optimisé (ajusté avec les hyperparamètres sélectionnés) montrent une amélioration notable par rapport à ceux de la régression logistique, qui est un modèle paramétrique. Sans détailler chaque métrique, on observe tout d'abord une précision de 0.993, ce qui suggère que la majorité des prédictions positives sont correctes, avec peu de faux positifs. En outre, le modèle parvient à détecter environ 61.76% des défauts (Recall = 0.618), ce qui est important pour éviter les faux négatifs, c'est-à-dire les personnes que le modèle aurait classées à tort comme « non-défaut » alors qu'elles présentent effectivement un risque de défaut. En se prémunissant contre les faux négatifs, le modèle aide à réduire le risque d'octroyer un crédit à des individus qui pourraient ne pas être en mesure de le rembourser. Cette amélioration se reflète également dans le F1-score, qui constitue une métrique cruciale pour évaluer la performance sur un ensemble de données déséquilibré. Par ailleurs, l'AUC-ROC atteint 0.9660, démontrant une excellente capacité de discrimination entre les classes "défaut" et "non-défaut". Enfin, avec une Log Loss de 0.2119, le modèle est bien calibré et fournit des probabilités fiables.

FIGURE 12 – Résultats de Random Forest

Pour conclure sur le modèle Random Forest, on observe d'excellentes performances prédictives, mais le modèle manque d'interprétabilité.

4.4.3 SVM

Après avoir analysé un modèle linéaire (régression logistique) et un modèle d'ensemble (Random Forest), nous avons exploré le SVM, un algorithme de classification qui sépare les données en

utilisant des hyperplans optimaux. Ce modèle est bien adapté à nos données continues standardisées et à notre variable cible binaire déséquilibrée, car il peut créer une frontière de décision efficace même en présence de classes déséquilibrées. Pour optimiser ses performances, nous avons ajusté les hyperparamètres clés, notamment le type de noyau, le paramètre de régularisation (C), et le gamma, afin de trouver un bon équilibre entre flexibilité et généralisation, tout en réduisant le risque de surajustement.

Nous portons une attention particulière au F1-score, à l'AUC ROC et à la Log Loss, car ces métriques sont particulièrement adaptées aux données déséquilibrées. Le F1-score de 0.807 montre un bon équilibre entre précision et rappel, ce qui est essentiel pour évaluer la performance sur les classes minoritaires. L'AUC ROC de 0.923 démontre une excellente capacité du modèle à distinguer entre les classes "défaut" et "non-défaut". Enfin, la Log Loss de 0.215 indique que le modèle est bien calibré et fournit des probabilités prédictives fiables, ce qui est crucial pour une interprétation probabiliste dans le cadre du scoring de crédit.

```
Entrainement SVM...
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Les meilleurs paramètres pour SVM: {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}
Résultats pour le modèle SVM :
    Metric     Score
  Accuracy  0.933725
 Precision  0.964912
    Recall  0.693277
  F1-Score  0.806846
   ROC AUC  0.923172
  Log Loss  0.214954
     Model       SVM
```

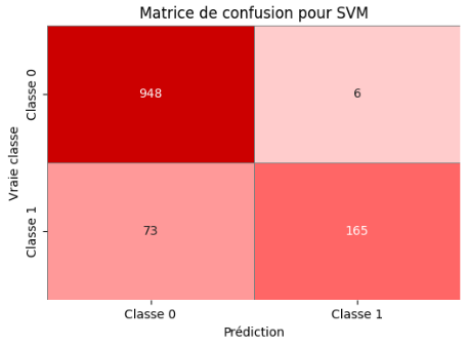


FIGURE 13 – Résultats de SVM

5 Grille de score

La grille de score dans un modèle de scoring de risque de crédit est essentielle pour évaluer la probabilité de remboursement d'un emprunteur. Cette grille permet d'attribuer une "note" ou un "score" basé sur divers critères financiers et personnels, permettant ainsi de mesurer le risque de défaut de paiement et de quantifier le niveau de risque associé à chaque profil. Nous avons utilisé la régression logistique pour construire la grille de score, car elle permet de modéliser la probabilité qu'un événement binaire se produise, ici, que un client fasse défaut ou non, en attribuant un coefficient à chaque variable qui montre son impact sur le risque.

Nous avons converti les coefficients du modèle en points de score pour chaque variable, afin d'évaluer l'impact de chacune d'elles sur le score final.

$$\text{Points}_{\text{variable}} = \frac{\beta_{\text{variable}}}{\ln(2)} \times \text{facteur} \tag{7}$$

Nous avons ensuite calculé le score total pour chaque client en additionnant le score de base aux points de chaque variable.

Pour notre grille de score, nous avons associé à chaque variable le point correspondant, sa contribution au modèle, sa fréquence ainsi que son taux de défaut. Pour évaluer la performance de la

grille de score, nous avons établi la grille pour l'échantillon d'apprentissage et celui de test, puis comparé la courbe ROC, l'indice de Gini et la distribution des scores. La courbe ROC trace le taux de vrais positifs contre le taux de faux positifs à différents seuils de classification, permettant de visualiser la performance de la grille de score à travers différents seuils, où les coûts des faux positifs et des faux négatifs peuvent être significatifs. L'indice de Gini est étroitement lié à l'AUC. Cela signifie qu'un bon Gini signifie également une bonne AUC, ce qui facilite la comparaison entre ces deux mesures.

Métrique	**Train**	**Test**
AUC	0.83	0.81
Indice de Gini	0.66	0.62

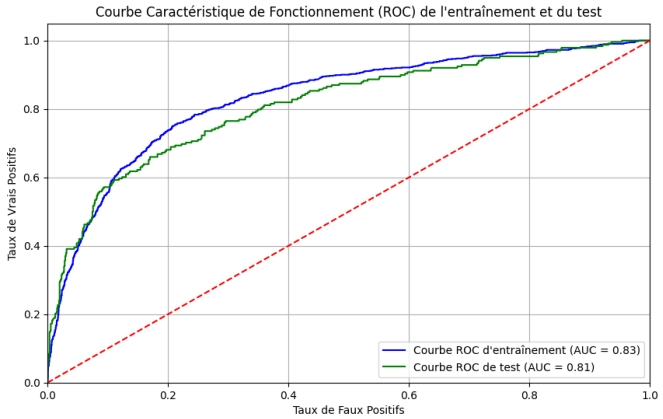


FIGURE 14 – Résultats de performance de la grille de score

Nous constatons qu'il n'y a pas de différence significative de performance entre l'échantillon d'apprentissage et celui de test en termes du score d'AUC et de l'indice de Gini. Ces deux métriques restent élevés pour les deux échantillons.

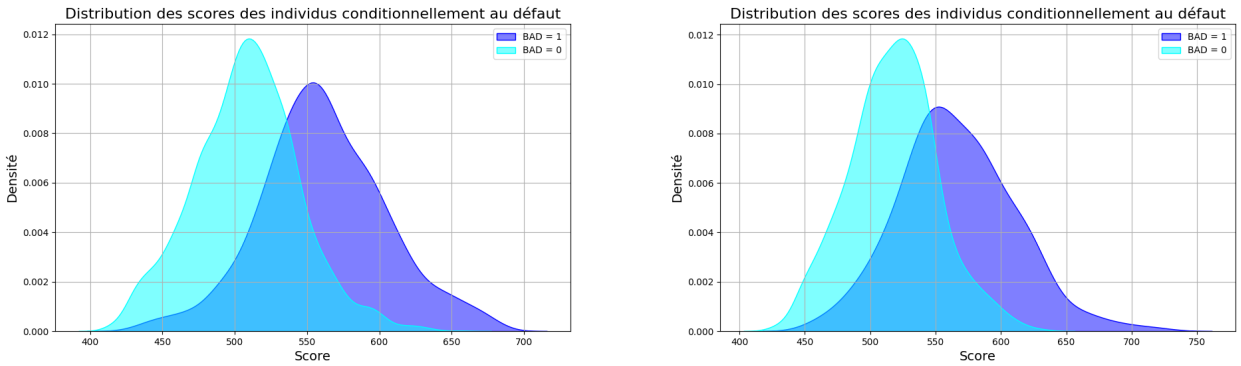


FIGURE 15 – Distribution des scores des individus conditionnellement au défaut pour Train et Test

Pour les deux échantillons, la distribution des scores des non-défauts se concentre autour de scores légèrement inférieurs, tandis que celle des individus en défaut est décalée vers la droite, autour de

550. On observe un certain éloignement entre les distributions des individus en défaut et de ceux sans défaut, mais cet écart n'est pas très marqué. Par conséquent, on constate que les individus sans défaut tendent vers des scores plus bas, tandis que les individus en défaut se regroupent vers des scores plus élevés, créant ainsi une séparation modérée mais non complète entre les deux groupes.

6 Conclusion

Pour conclure, la régression logistique, en tant que modèle linéaire, offre une bonne interprétabilité mais est surpassée en performance par les modèles d'ensemble et le SVM, notamment sur les métriques de rappel, F1-score, ROC AUC et Log Loss, essentielles pour traiter des données déséquilibrées. L'analyse des performances montre que les modèles d'ensemble, en particulier le Random Forest et le Gradient Boosting, affichent les meilleures performances en termes de précision, ROC AUC et Log Loss. Le SVM s'avère également performant, avec de bons résultats en Recall, F1-score et Log Loss.

Au final, le choix du meilleur modèle dépend de nos objectifs spécifiques. Si notre objectif principal est la prédiction, les modèles puissants comme le Random Forest, le Gradient Boosting et le SVM sont à privilégier. En revanche, si l'objectif est de comprendre les relations statistiques entre les variables et d'interpréter l'importance et l'influence des facteurs explicatifs, la régression logistique reste le choix le plus adapté.

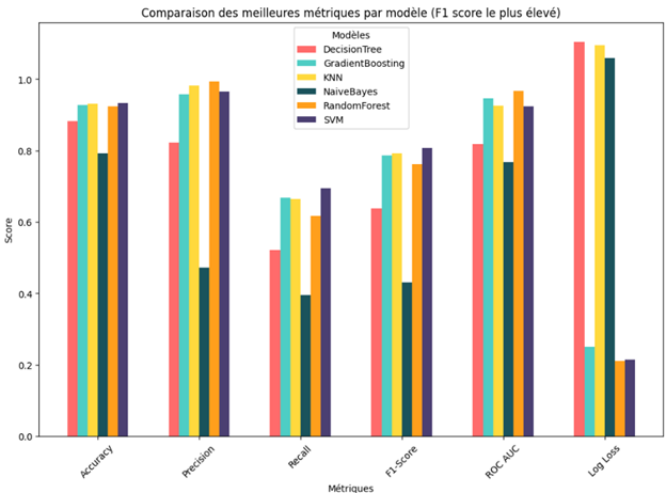


FIGURE 16 – Comparaison des meilleures métriques par modèle

Il est à noter que ce projet présente certaines limites, principalement liées à la nature des données. Premièrement, disposer d'une base de données plus vaste aurait permis d'améliorer la robustesse et la généralisation des résultats, car un échantillon plus grand est toujours souhaitable pour renforcer la fiabilité des modèles. Deuxièmement, pour améliorer la performance de nos modèles, l'accès à d'autres variables aurait été bénéfique. Notre base de données se concentre principalement sur les aspects de crédit et quelques données financières de base, mais elle ne couvre pas des éléments essentiels du profil financier, tels que le revenu total (salaire et autres sources de revenus) et les dépenses mensuelles. De plus, l'absence de données démographiques, comme l'âge, le statut familial ou le niveau d'éducation, limite la capacité du modèle à prendre en compte des facteurs

indirects de stabilité financière. Enfin, les informations disponibles sur l'historique de crédit sont également limitées. Des détails supplémentaires, comme le type de crédit contracté précédemment ou le comportement de paiement à long terme, auraient permis de mieux évaluer le risque de crédit. Les variables économiques et contextuelles, comme le taux de chômage régional ou la croissance économique, font aussi défaut, ce qui empêche le modèle de tenir compte de l'environnement économique du demandeur.

7 Annexe

Variable	Valeur	Coefficient	Points	Contribution (%)	Fréquence (%)	Taux de défaut (%)
Age de la Ligne de Crédit	1	-0.35	-10.14	0.31	5.89	0.40
	2			2.28	42.74	0.25
	3			0.45	8.45	0.17
	4			1.01	19.00	0.15
	5			1.27	23.91	0.11
Nombre de lignes de crédit	1	-0.11	-3.26	0.16	9.44	0.33
	2			0.92	53.71	0.18
	3			0.09	5.10	0.11
	4			0.12	7.15	0.15
	5			0.42	24.60	0.23
Ratio Dette/Revenu	1	0.92	26.62	2.72	19.44	0.09
	2			1.15	8.24	0.11
	3			8.68	62.00	0.19
	4			0.73	5.20	0.25
	5			0.72	5.12	0.76
Nombre de lignes de crédit en retard	1	0.72	20.76	7.86	71.96	0.14
	2			1.83	16.80	0.24
	3			0.59	5.41	0.47
	4			0.64	5.83	0.61
Nombre de Rapports Dérogatoires	1	0.73	21.09	9.09	81.92	0.16
	2			1.44	13.02	0.30
	3			0.56	5.05	0.61
Catégories professionnelles	0	0.28	8.19	1.62	37.50	0.16
	1			2.47	57.42	0.22
	2			0.22	5.08	0.32
Montant du prêt	1	-0.31	-8.87	0.26	5.66	0.49
	2			1.79	38.49	0.22
	3			1.67	35.91	0.16
	4			0.65	13.93	0.11
	5			0.28	6.02	0.26
Montant dû sur l'hypothèque existante	1	-0.26	-7.40	0.42	10.70	0.27
	2			0.56	14.45	0.22
	3			1.32	33.79	0.21
	4			0.24	6.06	0.18
	5			1.36	35.00	0.17
Nombre de demandes de crédit récentes	1	0.14	4.11	0.93	42.83	0.15
	2			0.59	27.16	0.19
	3			0.34	15.56	0.21
	4			0.16	7.24	0.26
	5			0.16	7.21	0.42
Raison du prêt: consolidation de dettes	0	-1.34	-38.60	6.00	29.57	0.23
	1			14.30	70.43	0.19
Raison du prêt: amélioration de la maison	0	-1.20	-34.57	12.81	70.43	0.19
	1			5.38	29.57	0.23
Valeur de la propriété	1	0.16	4.51	0.15	6.38	0.39
	2			0.95	39.85	0.21
	3			1.03	43.35	0.17
	4			0.13	5.41	0.10
	5			0.12	5.01	0.24
Nombre d'années à l'emploi actuel	1	-0.07	-2.00	0.07	6.88	0.13
	2			0.15	14.49	0.30
	3			0.18	17.37	0.24
	4			0.54	51.57	0.18
	5			0.10	9.69	0.12

FIGURE 17 – Grille de score