École d'Économie de la Sorbonne
Master 2 Modélisations Statistiques, Économiques et Financières
MoSEF - Data science

Scoring & Risque de défaut

Abdoul Aziz BERRADA · Amira Slimene

Encadrés par M. Ibrahim TOURE

21 Novembre, 2021
Paris, France

Table des matières

ÉCOLE D'ÉCONOMIE
DE LA SORBONNE

Scoring et Risque de défaut

Abdoul Aziz BERRADA · Slimene Amira

Abstract :
Dans cet article, nous allons développer un modèle de risque de crédit piloté par les données en Python pour prédire les probabilités de défaut (PD) et attribuer des scores de crédit aux emprunteurs existants ou potentiels. Nous déterminerons les scores de crédit à l'aide d'une Grille de score hautement interprétable, facile à comprendre et à mettre en œuvre.

Mots clés : Risque de crédit, Probabilité de défaut, Grille de score.

Correspondance : amira.slimane@outlook.fr, Abdoul-Aziz-Dit-Moun.Berrada@etu.univ-paris1.fr

Tous les auteurs partagent la même responsabilité pour ce travail.

1 Introduction

Cette analyse définit le scoring des clients sur la base d'une analyse statistique des caractéristiques des emprunteurs passés au lieu d'utiliser des règles de jugement. Il est prouvé que les modèles statistiques améliorent la précision des décisions de crédit et rendent les prêts plus rentables. Ils aident également les entreprises à prendre des décisions clées tout au long du cycle de vie de leur relation avec un client.

On pense parfois que l'évaluation statistique du crédit est trop coûteuse ou difficile, ou qu'on ne dispose pas du type de données nécessaires pour la mettre en œuvre. Cependant, la principale donnée nécessaire pour ce type de modélisation est : les historiques de remboursement des clients.

Le credit scoring aide ainsi les institutions financières à développer leurs portefeuilles en réduisant le coût du service aux clients à faibles revenus et en augmentant la qualité du service et la satisfaction client. Un modèle de scoring de crédit est un outil de gestion des risques qui évalue la solvabilité d'un demandeur de prêt en estimant sa probabilité de défaillance sur la base de données historiques. Il utilise des outils numériques pour classer les cas en utilisant des données intégrées dans une seule valeur qui tente de mesurer le risque ou la solvabilité.

Le scoring statistique s'appuie sur des caractéristiques quantifiées de l'historique du portefeuille du client, enregistrées dans une base de données. Il utilise un ensemble de règles et de techniques statistiques pour prévoir le risque sous forme de probabilité.

2 Description des données

Nous utilisons l'ensemble de données sur le crédit immobilier HMEQ rapportant les caractéristiques et les informations de remboursement des clients. Le jeu de données contient les observations de 5 960 demandeurs de prêts hypothécaires.
En effet, un prêt sur valeur nette immobilière est un prêt pour lequel le débiteur utilise la valeur nette de son logement comme garantie sous-jacente.
Pour chaque demandeur, 12 variables d'entrée ont été enregistrées.

1. LOAN : Montant du prêt
2. MORTDUE : Montant dû sur l'hypothèque existante
3. VALUE : Valeur du bien actuel
4. REASON : DebtCon = consolidation de dettes, HomeImp = amélioration du bien
5. JOB : Catégories professionnelles
6. YOJ : Nombre d'années dans l'emploi actuel
7. DEROG : Nombre de cas dérogatoires majeurs
8. DELINQ : Nombre de lignes de crédit en défaut de paiement
9. CLAGE :Âge de la ligne de crédit la plus ancienne en mois
10. NINQ : Nombre d'enquêtes de crédit récentes
11. CLNO : Nombre de lignes de crédit
12. DEBTINC : Ratio dette/revenu

2.1 Analyse univariate

L'ensemble de données présente les caractéristiques suivantes : La cible (BAD) est une variable binaire indiquant si le client a payé son prêt ou s'il est en défaut de paiement. Cette issue défavorable s'est produite dans 1 189 cas (20 %).

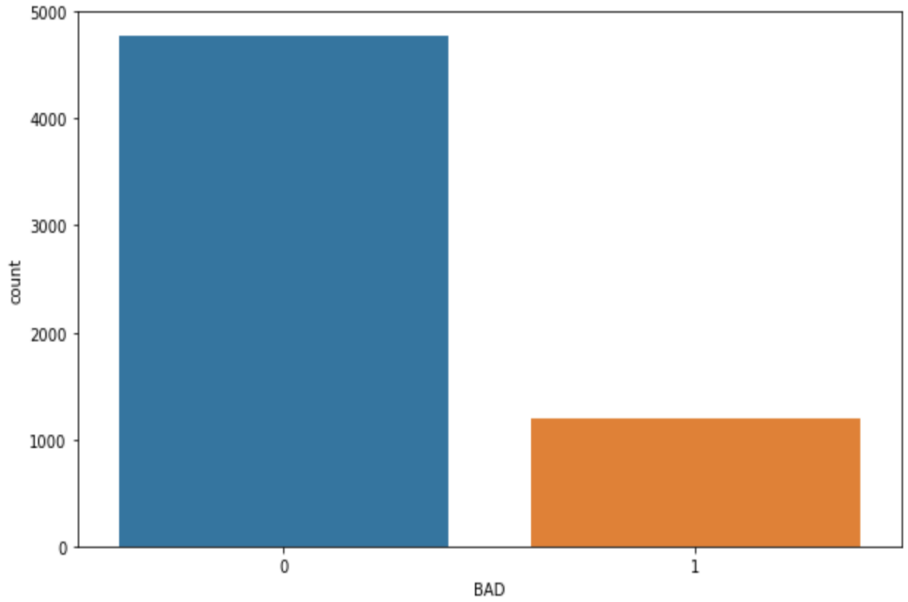

FIGURE 1 – Répartition des modalités de la target

On a 2 variables qualitatives (REASON, JOB) et 10 variables quantitatives.
Une distribution désaxée vers la droite ; une moyenne donc supérieure à la médiane distingue toutes les variables quantitatives de DEBTINC. La médiane est une meilleure mesure de tendance centrale lorsque les distributions sont désaxées.
Toutes les variables quantitatives ont un écart à la moyenne faible donc elles sont bien centrées sauf pour LOAN, MORTDUE, VALUE qui ont un écart type important.
Concernant les 2 seules variables qualitatives, on voit que REASON qui indique le motif de demande du prêt a 2 modalités et la plus fréquente est *DebtCon* (Consolidation de dettes) à hauteur de 68.8% des cas. Les demandeurs de prêts font rouler leurs prêts, ils s'endettent de nouveau pour payer leurs anciens prêts. La variable JOB , elle distingue 6 types de demandeurs de prêts, dans l'ordre d'importance on a : *Other, ProfExe, Office, Mgr, Self et Sales.*

2.2 Analyse multivariate

On remarque à partir de la Figure 2 que les montants prêtés augmentent au fur et à mesure que le nombre d'année d'ancienneté augmente et d'autres part avec la hausse de la valeur du bien. La valeur du prêt est positivement corrélée à l'expérience professionnelle et à la valeur du logement. On remarquera aussi que les prêts en défaut sont majoritairement ceux des clients les moins expérimentés (- de 20 ans d'ancienneté) et des individus avec un montant d'hypothèque inférieur à 200.000 $ avec des prêts inférieurs à 40.000 $.

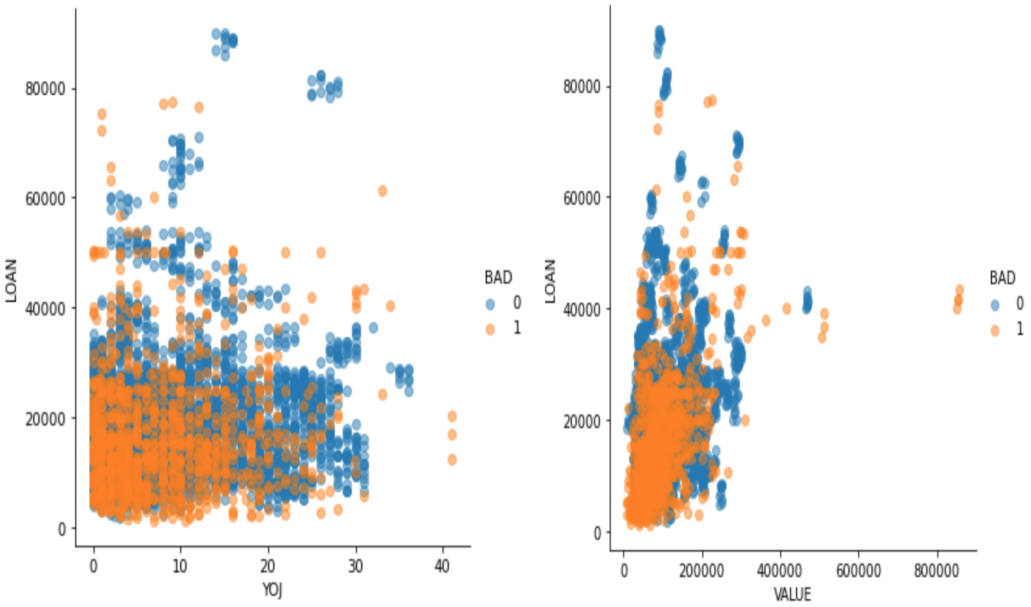

FIGURE 2 – Montant du prêt en fonction de l'expérience et de la valeur du logement

Le corrélogramme ci-dessous nous montre qu'il y a une corrélation entre MORTDUE et VALUE ce qui est logique étant donné que la première variable dénote le montant dû sur l'hypothèque existante et VALUE la valeur actuelle de ce même bien. En revanche, les autres variables sont très faiblement corrélées ce qui est un bon indicateur pour avoir des coefficients de régression significatifs en absence de corrélations entre les variables explicatives.

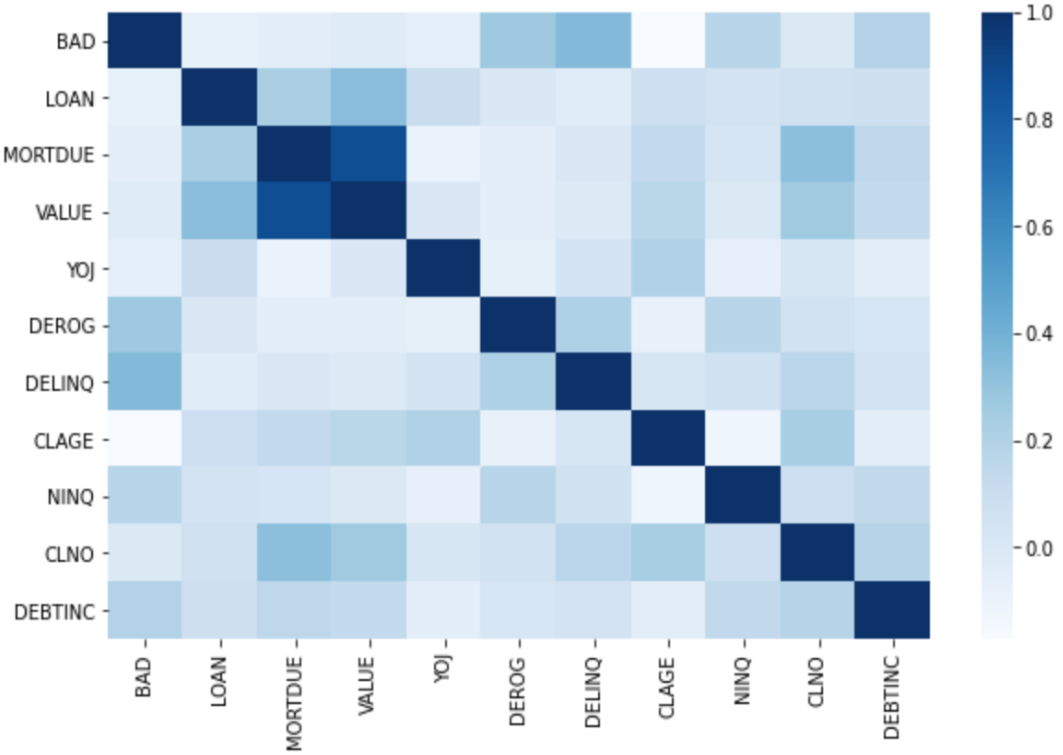

FIGURE 3 – Corrélations entre les variables

3 Méthodologie

Nous voulons automatiser le processus de prise de décision pour l'approbation des lignes de crédit hypothécaires. Pour ce faire, nous allons créer un modèle de notation de crédit empirique et statistiquement fiable. Le modèle sera basé sur des données recueillies auprès de demandeurs récents auxquels un crédit a été accordé par le biais du processus actuel de souscription de prêts. Le modèle sera construit à partir d'outils de modélisation prédictive, et sera suffisamment interprétable pour fournir une raison pour toute action défavorable (rejet).

3.1 Répartition des données en train et test

On va répartir maintenant nos données dans les ensembles suivants : entraînement (80%) et test (20%). Nous allons effectuer des manipulations sur la partie entraînement pour évaluer préliminairement notre modèle, tandis que l'ensemble de test nous servira pour l'évaluation finale du modèle. Notons que les transformations effectuées dans le train set seront aussi effectuées dans le test set.
Le fractionnement de nos données avant tout nettoyage des données ou imputation des valeurs manquantes empêche toute fuite de données de l'ensemble de test vers l'ensemble d'apprentissage et permet une évaluation plus précise du modèle.

3.2 Nettoyage des données

Il est important de comprendre pourquoi on a des données manquantes pour une colonne donnée dans un ensemble de données. Parmi les raisons possibles de l'absence de données on cite :

— Ce champ n'est pas applicable à tous les clients

— Ce champ contient des informations facultatives que les clients ne sont pas tenus de fournir

— L'information n'est pas disponible parce que elle n'a pas pas été demandée au client ou parce qu' elle a été demandée et que le client ne l'a pas fournie

— Les informations pour le champs n'ont pas été collectées avant ou après une date spécifique.

Selon la nature et la raison pour laquelle l'information est manquante, certaines colonnes de données doivent être traitées avec des techniques permettant de traiter des variables manquantes.
La méthodologie employée pour imputer les données manquantes est la suivante : on considère qu'au delà de 40% de valeurs manquantes, une stratégie d'imputation risque d'introduire un biais dans l'analyse donc on va pas utiliser une telle variable par la suite. Ce n'est pas le cas dans notre jeux de données puisque pour toutes les variables ont a un pourcentage de données manquantes inférieur à ce seuil.
On distinguera le traitement selon le type de la variable :

1. Pour les variables quantitatives : On va se baser sur la distribution de la variable, si sa distribution est asymétrique, on va imputer par la médiane et sinon par la moyenne.
   — Les variables asymétriques sont :
      (a) Dans le Train set : MORTDUE, VALUE, DEROG, DELINQ, CLAGE, NINQ, DEBTINC

(b) Les mêmes dans le Test set

— Les variables symétriques sont :

(a) Dans le Train set : YOJ et CLNO

(b) Dans le Test set : CLAGE et CLNO

2. Pour les variables qualitatives : On va considérer le pourcentage de données manquantes, si celui-ci est inférieur à 15%, on va imputer par la valeur la plus fréquente (donc le mode), sinon on sera amené à créer une nouvelle classe nommée "autres".

— Les variables avec - de 15% de Nan sont :

(a) Dans le Train set : REASON et JOB

(b) Les mêmes dans le Test set

— Aucune variable n'a + 15% de Nan ni dans le train ni dans le test set

Les valeurs imputées dans le train set seront répercutées dans le test set.

3.3 Features Engineering

Afin de construire notre grille de score, on a décidé de discrétiser toutes nos variables explicatives. Il y a beaucoup de justifications à la discrétisation des variables, une grille de score doit généralement être facilement interprétable, étant donnés les coûts élevés des erreurs de classifications monétaires et non monétaires. Cet objectif est facilement atteignable par un tableau de bord qui ne comporte aucune variable continue, toutes les variables étant discrétisées. Les raisons des scores faibles ou élevés peuvent être facilement comprises et expliquées aux tiers. Tous ces éléments permettent aux grilles de scores d'obtenir plus facilement l'adhésion des utilisateurs finaux contrairement aux modèles plus complexes. Une grille de score doit être capable de séparer les observations à faible et à haut risque.
Mais comment allons nous discrétiser nos variables ? On va utiliser les WoE (Weights of Evidence) et IV (Information Value). Les deux concepts sont très utilisés dans les étapes de features engineering et features selection dans le domaine de l'évaluation du crédit.

3.3.1 Weight of Evidence (WoE)

La création de nouvelles caractéristiques catégorielles pour toutes les variables numériques et catégorielles basées sur le WoE est l'une des étapes les plus critiques avant le développement d'un modèle de risque de crédit. On génère un binning optimal pour les valeurs numériques, factorielles et catégorielles en utilisant des méthodes comprenant la segmentation arborescente ou le chi carré pour fusionner des modalités. C'est une mesure du pouvoir prédictif d'une variable indépendante par rapport à la variable cible. Elle indique dans quelle mesure une caractéristique spécifique peut différencier les classes cibles, dans notre cas : les bons et les mauvais clients.
La formule pour calculer le WoE est la suivante :

$$WoE = \ln(\frac{\%defaut}{\%non-defaut}) \tag{1}$$

Un WoE positif signifie que la proportion de bons clients est supérieure à celle des mauvais clients et vice versa pour une valeur WoE négative.

Nous discrétisons les variables en classes distinctes (chacune avec un WoE différent), ce qui donne plusieurs catégories, ainsi les nouveaux emprunteurs potentiels seraient classés dans l'une des catégories en fonction de leurs caractéristiques.
L'objectif ultime est de faire un binning qui implique une monotonicité de la proportion de clients en défaut dans chaque classe créée (voir courbe bleue dans la Figure 4), toutefois elle n'est respectée que pour quelques variables dans notre cas JOB, NINQ, CLAGE, DELINQ, REASON et DEROQ, ci-dessous des illustrations.

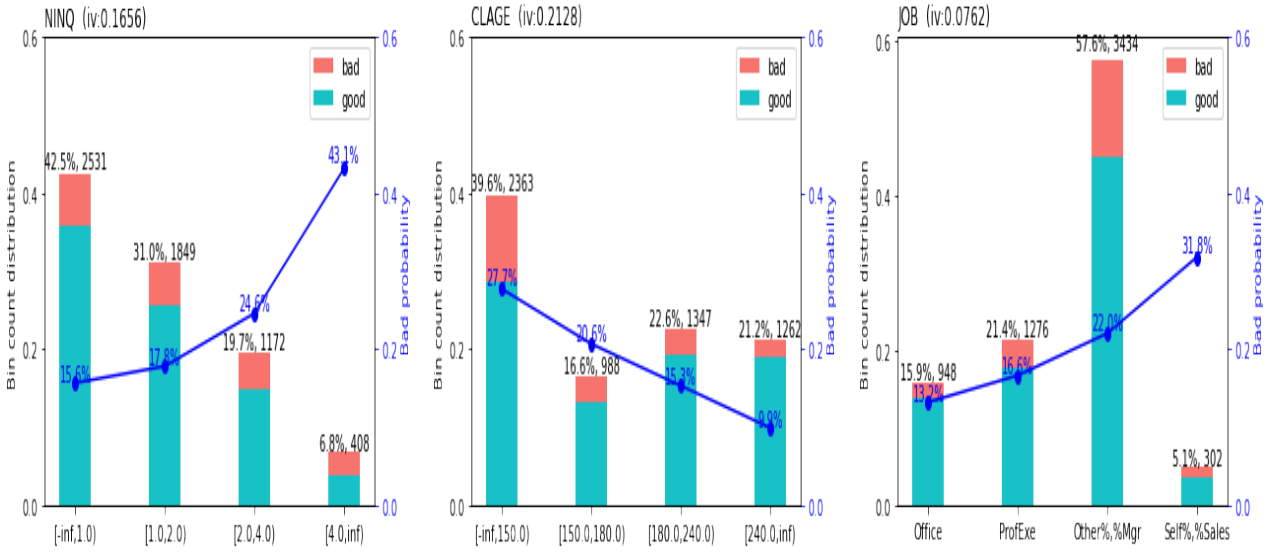

FIGURE 4 – Discrétisation des variables NINQ & CLAGE & JOB

On attend de l'algorithme de classement qu'il divise un ensemble de données d'entrée en classes de telle sorte que si vous passez d'une classe à l'autre dans la même direction, il y ait un changement monotone de l'indicateur de risque de crédit, c'est-à-dire qu'il n'y ait pas de sauts soudains dans le score de crédit si on passe d'une catégorie socio-professionnelle à une autre par exemple.

3.3.2 Information Value (IV)

L'IV aide à classer les variables explicatives (les caractéristiques des clients) en fonction de leur importance relative. Les analyses WOE et IV permettent de :

1. Considérer la contribution indépendante de chaque variable au résultat.
2. Détecter les relations linéaires et non linéaires
3. Classer les variables en fonction de leur force prédictive univariée
4. Visualiser les corrélations entre les variables et la valeur cible binaire.

La formule pour calculer la IV est la suivante :

$$IV = \sum(\ \%\ \text{of good customers}\ -\ \%\ \text{of bad customers})\ \times\ \text{WoE} \tag{2}$$

Voici un exemple de sortie de ces deux fonctions lorsqu'elles sont appliquées à nos données.

Information Value	Pouvoir predictif
Moins de 0.02	Non prédictif
0.02 à 0.1	Faible
0.1 à 0.3	Moyen
0.3 à 0.5	Fort
Plus de 0.5	Suspect

TABLE 1 – Valeurs possibles de l'IV

VALUE							
Classes	**Count**	**Count_distr**	**Good**	**Bad**	**Badprob**	**WoE**	**Bin_IV**
]-inf, 50000[	624	0.104698	428	196	0.314103	0.608435	0.045715
[50000,70000[	1114	0.186913	925	189	0.169659	-0.198603	0.006936
[70000, 85000[	891	0.149497	704	187	0.209877	0.063774	0.000620
[85000, 90000[	461	0.077349	295	166	0.360087	0.814456	0.063349
[90000, 125000[	1548	0.259732	1338	210	0.135659	-0.462380	0.048007
[125000, 175000[	733	0.122987	579	154	0.210095	0.065093	0.000531
[175000, inf[	589	0.098826	502	87	0.147708	-0.363249	0.011642
Somme	**5960**	**1**	**4771**	**1189**	**0.249**	**0.54**	**0.1768**

TABLE 2 – WOE & IV de la variable VALUE

Une fois que nous avons calculé et visualisé les valeurs de WoE et de IV, nous allons sélectionner les variables explicatives à abandonner en fonction de leur IV.
On remarque que certaines variables ne sont pas monotones mais on a fait le choix de les garder car on estime qu'elles sont pertinentes pour expliquer la probabilité de défaut telles que DEBTINC, LOAN et MORTDUE.
Pour la suite, les classes créées dans chaque variable explicative seront substituées à leur WoE dans le modèle, par exemple la classe]-inf, 50000[de la variable VALUE sera remplacée en entrée par 0.608 et ainsi de suite pour toutes les variables retenues.

3.3.3 Oversampling

Dans la base de données, on a eu un déséquilibre au niveau de la target qui est la variable BAD. En effet on a la répartition suivante : 80% de 0 (non défaut) et 20% de 1 (défaut).
On a ainsi décidé de faire du ré-échantillonage à l'aide du SMOTE pour rééquilibrer notre training set. Le SMOTE (Synthetic Minority Over-sampling Technique) est une méthode de sur-échantillonage qui permet de rééquilibrer la base de données en créant de nouvelles observations de la classe minoritaire par la technique des plus proches voisins KNN (k nearest neighbors).

3.4 Métriques utilisées

Afin de juger des performances de notre modèle, on sera amené à juger les valeurs des critères objectifs qui sont les suivants :

3.4.1 Accuracy

L'accuracy représente le pourcentage de bonne prédictions. Elle calcule ainsi le ratio du nombre de clients (en défaut ou pas) bien classés sur le nombre total de clients.
L'accuracy entre les données de test et les données prédites se détermine de la façon suivante :

$$
\begin{aligned}
Accuracy &= \frac{1}{N}\sum_{i=0}^{N-1} 1(y_{pred}i = y_i) && (3)\\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{|y_i \cap y_{pred}i|}{|y_i \cup y_{pred}i|} && (4)\\
&= \frac{TN + TP}{TN + TP + FP + FN} && (5)
\end{aligned}
$$

3.4.2 Précision

La précision minimise le taux d'erreurs parmi les exemples prédits positifs par le modèle. Elle renseigne alors sur la proportion des prédictions positives correctes. Sa valeur est comprise entre 0 et 1, plus elle est proche de 1 mieux est la précision, l'objectif étant de diminuer l'erreur du 1^{er} type.

$$
\begin{aligned}
Precision &= \frac{1}{N}\sum_{i=1}^{N} \frac{|y_i \cap y_{pred}i|}{|y_{pred}i|} && (6)\\
&= \frac{TP}{TP + FP} && (7)
\end{aligned}
$$

3.4.3 Recall

Le recall aussi connu sous le nom de sensitivité détermine la probabilité de détection d'un prédicteur. Il permet de connaître sur tous les clients en défaut, le pourcentage de clients que le prédicteur a effectivement classé en défaut (True Positive Rate). Sa valeur est comprise entre 0 et 1, plus elle est proche de 1 mieux est le recall, l'objectif étant de diminuer l'erreur du 2^{nd} type.

$$
\begin{aligned}
Recall &= \frac{1}{N}\sum_{i=1}^{N} \frac{|y_i \cap y_{pred}i|}{|y_i|} && (8)\\
&= \frac{TP}{TP + FN} && (9)
\end{aligned}
$$

3.4.4 F1 score

Dans la majeure partie des cas, le recall et la précision ne suffisent pas pour valider un modèle de classification. En effet il est difficile de choisir entre 2 modèles si l'un a une grande précision et un faible recall ou vice versa. Il est pertinent de chercher une autre métrique pour trancher, alors intervient le F1 score. Il représente une moyenne harmonique de la précision et du recall. Il pénalise les valeurs

extrêmes, si le recall ou la précision est faible alors le F1 score tend vers 0.

$$
\begin{aligned}
F1 - score &= 2\frac{precision * Recall}{precision + Recall} && (10)\\
&= \frac{TP}{TP + \frac{1}{2}(FP + FN)} && (11)\\
&= \frac{2 * TP}{2 * TP + (FP + FN)} && (12)
\end{aligned}
$$

3.4.5 ROC curve - AUC

La ROC (Receiver Operating Characteristic) curve est une mesure de performance d'un classificateur binaire. Elle trace le taux de vrais positifs (TPR ou Recall) en fonction du taux de faux positifs TFP pour chaque seuil possible. La meilleure courbe est celle qui longe l'axe des ordonnées et atteint le point extrême au Nord-Ouest, le coin supérieur gauche.
Elle facilite la comparaison de plusieurs modèles en fournissant un indicateur unique qui est l'AUC, cette métrique ne nécessite pas non plus de spécifier un seuil de décision . L'AUC est invariante à l'échelle, elle renseigne à quel point les prédictions sont classées, plutôt que leurs valeurs absolues. Le meilleur modèle a son AUC qui est proche de 1, le prédicteur aléatoire a son AUC = 0.5 et sa ROC curve est une bissectrice. Un modèle avec son AUC = 0 est un modèle qui prédit toujours Y=0 lorsque Y=1 et Y=1 lorsque Y=0.

3.4.6 Lift curve - Cumulative Gain Chart

La Lift curve mesure la performance d'un modèle de classification par rapport à un classificateur aléatoire. Elle fournit une synthèse visuelle de l'information apportée par un modèle statistique dans la prévision d'une variable de sortie binomiale. Plus précisément, la courbe synthétise les gains auxquels on peut s'attendre en utilisant le modèle prédictif respectif, par rapport à l'utilisation de l'information de référence uniquement.
La lift curve est dérivée de la Cumulative Gains Chart (La courbe des gains). Cette courbe indique le taux des totaux positifs (Y=1) ou négatifs (Y=0) en pourcentage par rapport au pourcentage de dénombrements totaux.

Afin de comparer 2 modèles de classification, on considère que le meilleur modèle est celui qui a la plus grande valeur de Lift.

$$Lift = \frac{\%Gain}{\%population} \quad (13)$$

.

3.4.7 Kolmogorov-Smirnov

La courbe de Kolmogorov-Smirnov montre la différence entre la distribution des «good» (bons clients - sans défaut) et des «bad» (clients en défaut). La différence maximale entre la distribution de ses séries connue sous le nom de valeur Kolmogorov-Smirnov, est souvent utilisée avec la valeur Gini pour évaluer la qualité de la grille de score.

$$KS = Maximum|F_0(X) - F_r(X)| \tag{14}$$

3.4.8 Gini

Le coefficient de Gini est une métrique qui indique le pouvoir discriminatoire du modèle, à savoir l'efficacité du modèle à différencier les "mauvais" emprunteurs, qui feront défaut à l'avenir, des "bons" emprunteurs, qui ne feront pas défaut à l'avenir. Cette métrique est souvent utilisée pour comparer la qualité de différents modèles et évaluer leur pouvoir de prédiction. On peut calculer le coefficient de gini grâce à l'AUC tel que :

$$Gini = 2 * AUC - 1 \tag{15}$$

4 Modèles

4.1 Features selection

Pour sélectionner les meilleures variables à mettre en input dans notre modèle, on a jugé nécessaire d'appliquer différents algorithmes de sélection autonmatiques de variables. Ces algorithmes en question sont la Recursive Feature Elimination, la Sequential Feature Selector et Exhaustive Feature Selector. On a alors appliqué une régression logistique à chaque algorithme en maximisant l'AUC pour au final obtenir les variables les plus pertinentes qui seront sélectionnées.

— **Recursive Feature Elimination - RFE :** c'est une méthode de sélection des caractéristiques qui ajuste un modèle et élimine la ou les caractéristiques les plus faibles jusqu'à ce que le nombre spécifié de caractéristiques soit atteint. L'élimination récursive des caractéristiques requiert un nombre spécifique de caractéristiques à conserver, mais il est souvent impossible de savoir à l'avance combien de caractéristiques sont valides.

— **Sequential Feature Selector - SFS :** c'est un algorithme qui sélectionne plusieurs caractéristiques (variables) à partir de l'ensemble des caractéristiques et les évalue par itérations du modèle entre les différents ensembles en réduisant et en améliorant le nombre de caractéristiques afin que le modèle puisse atteindre les performances et les résultats optimaux.

— **Exhaustive Feature Selector - EFS :** c'est une approche enveloppante pour l'évaluation brute de sous-ensembles de caractéristiques ; le meilleur sous-ensemble est sélectionné en optimisant une mesure de performance spécifiée pour un régresseur ou un classificateur arbitraire. Par exemple, si le classificateur est une régression logistique et que l'ensemble de données est constitué de 4 caractéristiques, si min-features=1 et max-features=4, l'algorithme évaluera 15 (4 + 6 + 4 + 1) combinaisons, ie chaque variable à la fois, ensuite les variables combinées 2 à 2 puis leurs combinaisons 3 à 3 et enfin les 4 variables à la fois. L'algorithme sélectionnera celle qui donne les meilleures performances du modèle de régression logistique.

Au final, la régression logistique avec la méthode Recursive Feature Elimination a eu la valeur d'AUC la plus élevée et a sélectionné 11 variables : LOAN, MORTDUE, DEBTINC, DELINQ, VALUE, NINQ,

CLNO, CLAGE, YOJ, JOB, DEROG ; la variable REASON est supprimée. Nous allons alors considérer que ces 11 variables pour la suite de notre modèle.

4.2 Model selection and validation

Afin de valider le modèle choisi, on va appliquer notre régression logistique et puis faire une analyse en 2 parties : d'abord juger des critères d'information du modèle dans le train et puis déterminer sa capacité à généraliser.

```
Optimization terminated successfully.
         Current function value: 0.313470
         Iterations 7
                         Results: Logit
================================================================
Model:              Logit            Pseudo R-squared: 0.362
Dependent Variable: BAD              AIC:              3013.2533
Date:               2021-11-16 22:08 BIC:              3090.8895
No. Observations:   4768             Log-Likelihood:   -1494.6
Df Model:           11               LL-Null:          -2344.3
Df Residuals:       4756             LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     7.0000
----------------------------------------------------------------
                Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
----------------------------------------------------------------
const          -1.4371   0.0497 -28.9167 0.0000 -1.5345 -1.3397
JOB_woe         0.8457   0.1709   4.9488 0.0000  0.5108  1.1807
DEROG_woe       0.6953   0.0773   8.9961 0.0000  0.5438  0.8468
YOJ_woe         0.9129   0.1835   4.9763 0.0000  0.5534  1.2725
DEBTINC_woe     0.9207   0.0405  22.7237 0.0000  0.8413  1.0001
MORTDUE_woe     0.4662   0.2056   2.2674 0.0234  0.0632  0.8693
CLNO_woe        1.0641   0.1511   7.0403 0.0000  0.7679  1.3603
NINQ_woe        0.5895   0.1114   5.2940 0.0000  0.3713  0.8077
CLAGE_woe       0.9279   0.1066   8.7023 0.0000  0.7189  1.1368
LOAN_woe        0.6240   0.1088   5.7353 0.0000  0.4107  0.8372
VALUE_woe       0.6420   0.1192   5.3883 0.0000  0.4085  0.8756
DELINQ_woe      0.9309   0.0653  14.2651 0.0000  0.8030  1.0588
================================================================
```

TABLE 3 – Régression logistique

On remarque que toutes les variables explicatives ont une p-value inférieure à .000 sauf MORTDUE-woe qui reste significative au seuil de 5%. En outre, e modèle est donc statistiquement significatif avec une p-value du Logarithme de Vraisemblance nulle.

Ensuite, voici les différentes métriques calculées par validation croisée de 10 Folds dans le train et test sets :

	0	1	2	3	4	5	6	7	8	9
test_auc	0.893	0.878	0.876	0.887	0.890	0.884	0.884	0.889	0.879	0.889
train_auc	0.884	0.888	0.888	0.885	0.885	0.886	0.886	0.885	0.887	0.885
est_accuracy	0.816	0.802	0.806	0.808	0.811	0.803	0.799	0.820	0.810	0.804
train_accuracy	0.805	0.810	0.806	0.806	0.806	0.808	0.808	0.807	0.803	0.807
test_recall	0.818	0.784	0.817	0.812	0.827	0.801	0.801	0.837	0.822	0.798
train_recall	0.809	0.816	0.812	0.810	0.808	0.815	0.813	0.808	0.809	0.812
test_precision	0.815	0.813	0.800	0.805	0.801	0.804	0.798	0.809	0.803	0.808
train_precision	0.803	0.806	0.803	0.804	0.804	0.804	0.806	0.806	0.799	0.803
test_f1	0.816	0.798	0.808	0.809	0.814	0.803	0.799	0.823	0.812	0.803
train_f1	0.806	0.811	0.807	0.807	0.806	0.809	0.809	0.807	0.804	0.808

TABLE 4 – Validation du modèle

Ce tableau nous permet de voir que le modèle est capable de généraliser ses performances. En effet les performances dans le train et le test sont très proches, le modèle n'overfitte pas.

4.3 Résultats et performances

Comment déterminer les prêts à approuver et à rejeter ? Quelle est la capacité de détection de notre modèle de classification ? Quel est le seuil idéal pour classer un client en défaut ou pas ?
Pour trouver ce seuil, nous devons déterminer la statistique de Youden. Cette statistique permet de maximiser le Taux de Vrais Positifs tout en minimisant le Taux de Faux Positifs. Toutefois en appliquant ce cutoff de 0.09 très très faible, notre modèle était très déséquilibré. En effet on a un recall très élevé et une précision très faible. Même si notre problème est orienté recall, on préfère garder un certain équilibre entre ces 2 métriques afin d'avoir un f1 score appréciable.

Dans notre cas, on a procédé par tâtonnement et puis fixé ce cutoff à 0,45. Toutes les observations dont la probabilité prédite est supérieure à ce seuil doivent être classées dans la catégorie Défaut (classe 1) et Non Défaut (Classe 0) sinon.

Les résultats sont les suivants :

	Predicted	
Actual	TN = 713	FP = 214
	FN = 54	TP = 211

TABLE 5 – Matrice de confusion

Un aperçu des principales métriques nous renseigne sur la capacité prédicative de notre régression logistique. On considère que notre modèle est capable de généraliser grâce notamment à son accuracy qui est sensiblement égale à celle du train set, et qui est très élevée car supérieure à la nouvelle répartition issue du ré-échantillonage (qui était de 50-50) . Notre problème est orienté recall, avec un cutoff arbitrairement fixé à 0.45, on a un recall de 0.80 ce qui dépasse largement nos attentes. Pour

Cutoff = 0.45	Precision	Recall	F1 score	Accuracy	AUC	Gini
Training set				0.81	0.89	0.77
Test set				0.80	**0.88**	**0.75**
0	0.93	0.77	0.61			
1	**0.50**	**0.80**	**0.61**			
Macro avg	0.71	0.78	0.73			
Weighted avg	0.83	0.78	0.79			

TABLE 6 – Principales métriques

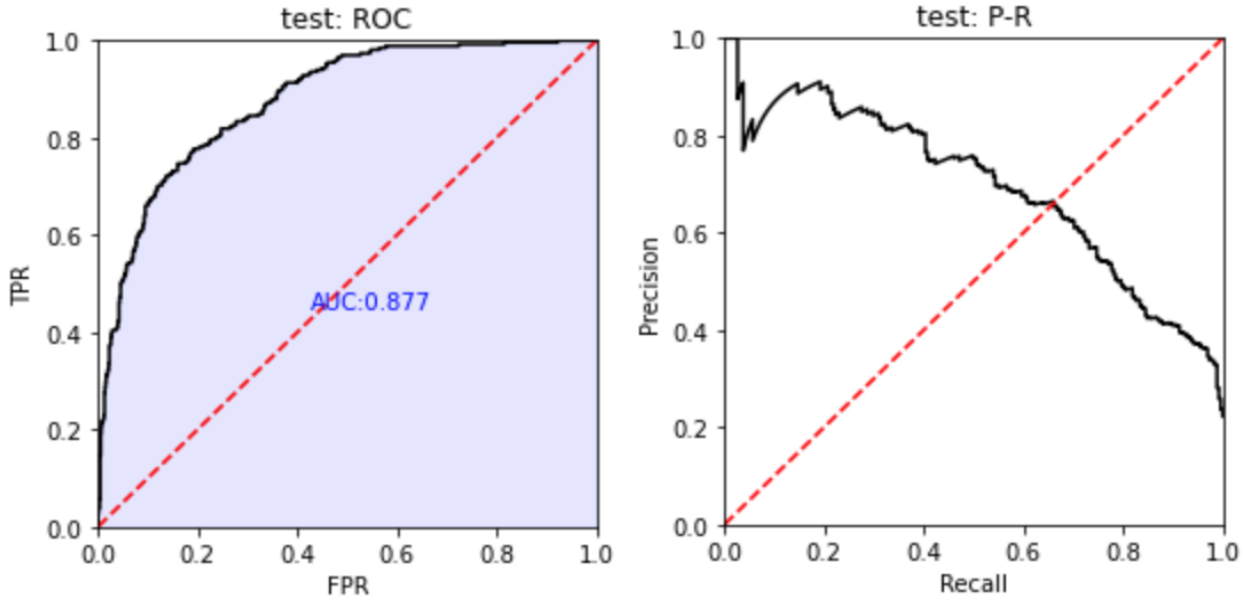

FIGURE 5 – ROC Curve & P-R Curve

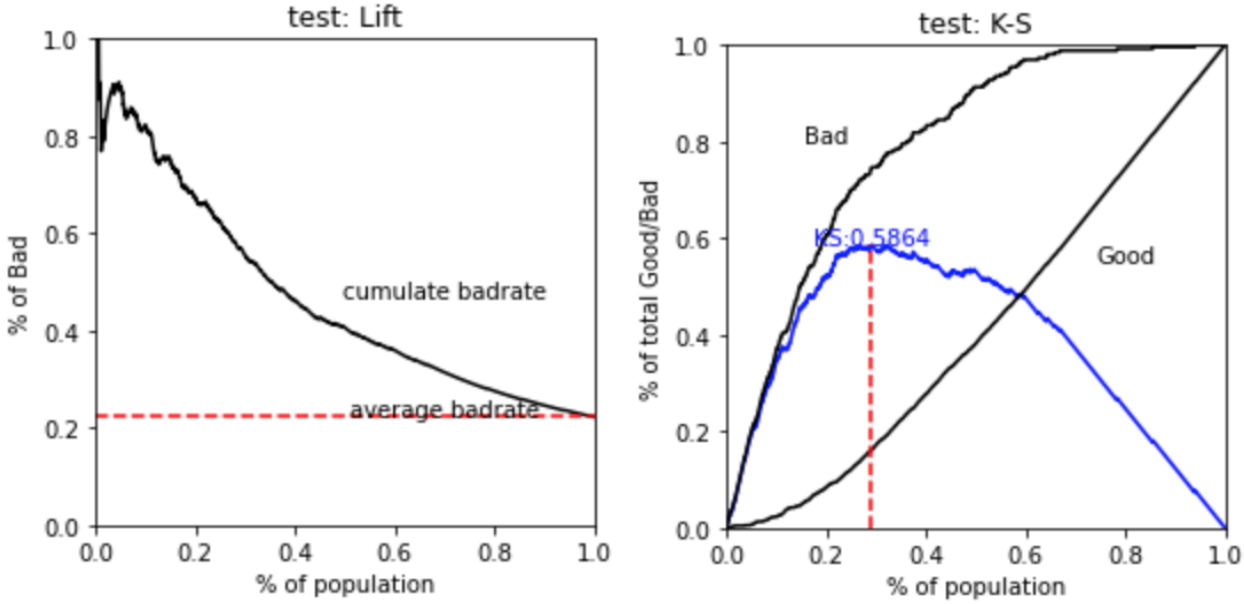

FIGURE 6 – Lift Curve & K-S Curve

couronner le tout, notre AUC à hauteur de 0.88 nous garantit de bonnes prédictions globales invariantes

au cutoff.
La PR curve (Precision - Recall curve) montre l'arbitrage entre la précision et le recall pour différents seuils. L'espace important sous la courbe représente à la fois un rappel et une précision élevés, où une précision élevée correspond à un faible taux de faux positifs, et un recall élevé correspond à un faible taux de faux négatifs.
Elle montre les courbes d'analyse de la proportion d'instances de données vrai positives par rapport au seuil du classificateur ou au nombre d'instances que nous classons comme positives. En prenant seulement les 20% des individus ayant les scores (ou les probabilités de faire défaut) les plus faibles, on identifie environ 65% des défauts totaux.
Dans notre cas présent, notre K-S stats est égale à 0.5864, ce qui nous permet de constater une différence nette entre les deux distributions.

5 Grille de score

5.1 Construction de la Grille de Score

Notre Grille de score est calibrée sur 1000 points, plus le score est élevé, moins le client a de risque de faire défaut. On fixe les Odds à $\frac{1}{500}$ (500 :1) et les PDO (Points Double the Odds) à 30 points.
Ainsi une personne qui a un score 1000 points a une probabilité de défaut de 1/500 alors qu'une personne qui a un score de 1030 (1000 + 30) elle a une une probabilité de défaut égale à $\frac{1}{1000}$ (1000 :1 = 2*500 :1).
En pratique, la Grille de score est déterminée de la façon suivante :

— On détermine la probabilité de défaut $p(y = 1)$ par :

$$p(y = 1) = \frac{1}{1 + exp(-Z)} \tag{16}$$

avec z qui est la combinaison linéaire des WOE de toutes les variables sélectionnées en entrée

$$Z = b_0 + b_1 * WOE_1 + b_2 * WOE_2 + ... + b_k * WOE_k \tag{17}$$
$$Z = log(\frac{p}{1-p}) \tag{18}$$

Mais comme on sait que :

$$Odds = \frac{p}{1-p} \tag{19}$$

donc

$$Z = log(Odds) \tag{20}$$

— Ensuite on va directement déterminer les points par les formules suivantes :

$$Points = Shift + Slope * Z \tag{21}$$
$$Points + PDO = Shift + Slope * log(2 * Odds) \tag{22}$$

RQ : Le PDO est positif pour garantir le fait les scores les plus élevés engendrent les risques les

plus faibles.

La contribution de variable j pour l'individu i est donnée par :

$$Points_j i = Slope * (b_j * WOE_j(i)). \tag{23}$$

Par défaut, les points de base ne sont pas équi-distribués ; dans notre cas on a décidé que les points de base soient également distribués à chaque variable donc :

$$Points_j i = \frac{Shift + Slope * b_0}{k + Slope * (b_j * WOE_j(i))} \tag{24}$$

avec k le nombre de variables dans le modèle et $Basepoints = Shift + Slope * b_0$.

5.2 Test Stabilité de la Grille de Score

Nous avons testé la stabilité des deux distributions de scores dans le train et le test set, on obtient un PSI=0.0162 (PSI : Population Stability Index). L'indice de stabilité de la population (ISP) est utilisé pour mesurer l'applicabilité d'un modèle en mesurant le changement des variables indépendantes ou dépendantes.Il compare la distribution d'une variable de notation (probabilité prédite) dans un ensemble de données de notation à un ensemble de données d'entraînement qui a été utilisé pour développer le modèle. L'idée est de vérifier "comment le score actuel est comparé à la probabilité prédite à partir de l'ensemble de données du train set."
Pour chaque variable, on a déterminé la stabilité du taux de défaut pour toutes les classes précédemment créées dans le test et le train. Cette stabilité permet de bien pouvoir généraliser les résultats.

Variables	**PSI**
CLAGE	0.000723
DEROG	0.000166
MORTDUE	0.001799
DELINQ	0.011429
NINQ	0.003350
LOAN	0.009875
DEBTINC	**0.801477**
VALUE	0.007331
JOB	0.001790
YOJ	0.003823
CLNO	0.009945
SCORE	**0.016228**

TABLE 7 – Population Stability Index

Toutefois on a vu que pour la variable DEBTINC, on n'a absolument pas de stabilité dans les points délivrés par chaque classe. Mais la Grille de score étant très stable, on considérera cet instabilité comme étant sans grande incidence.

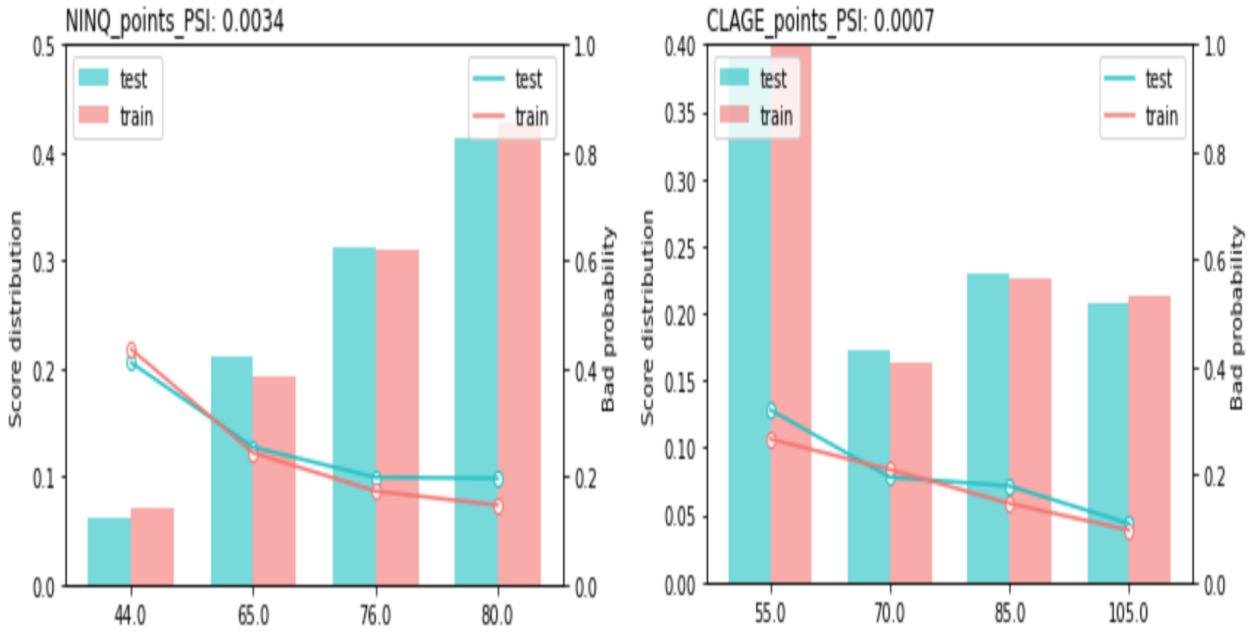

FIGURE 7 – NINQ & CLAGE PSI

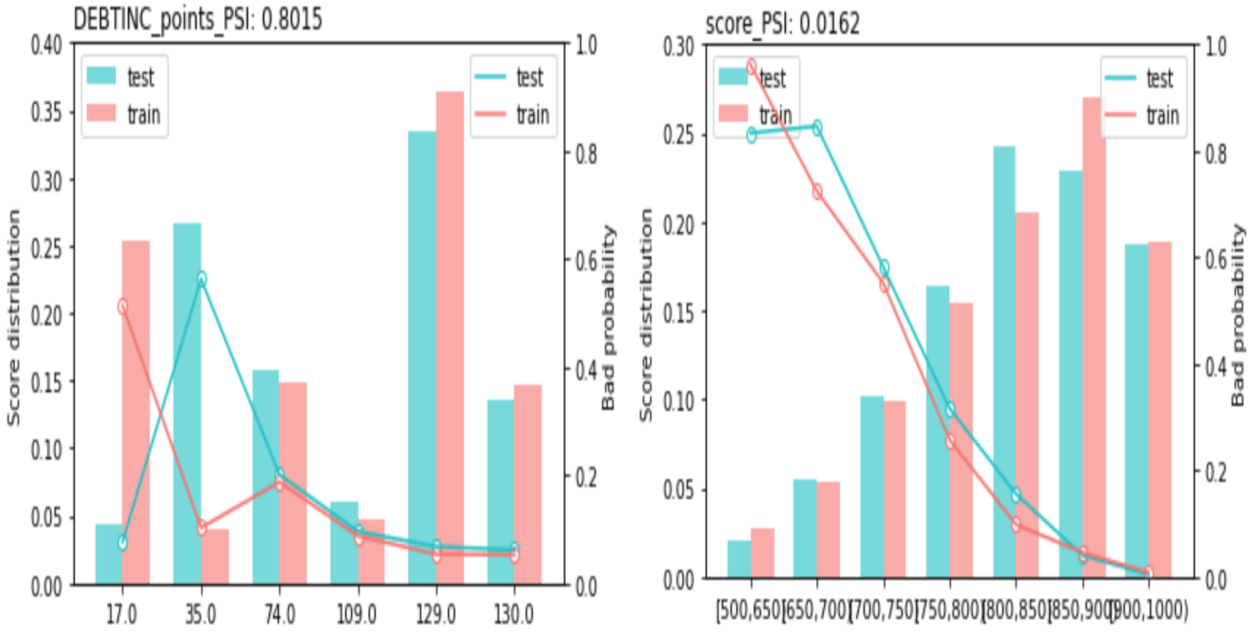

FIGURE 8 – DEBTINC & SCORE PSI

5.3 Grille finale

Après avoir transformé toutes nos variables en classes, sélectionné les pertinentes et puis choisi notre modèle, on a transformé les coefficients de la régression logistique en points. Ainsi chaque classe d'une variable donne au client qui y appartient un certain nombre de points, le cumul de tous les points pour toutes les classes et chaque individu donne le score final. Ce score permettra de décider s'il faut accorder ou non le crédit.

Cette grille est facilement interprétable et très intuitive, pour preuve les 2 exemples ci-dessous :

Variable	**VALUE**			
Classes	[-inf, 50000[	[50000, 70000[	[70000, 85000[	[85000, 90000[
-	-	-	-	-
Points	55	78	70	50
Classes	[90000, 125000[	[125000, 175000[	[175000, inf[	
-	-	-	-	
Points	85	70	82	
Variable	**JOB**			
Classes	Office	ProfExe	Other%,%Mgr	Self%,%Sales
-	-	-	-	-
Points	90	80	68	49

TABLE 8 – Exemple de points par classes VALUE & JOB

On voit alors que pour la variable VALUE, les clients qui ont une valeur de leur bien de 25K se voient attribués 55 points alors que ceux qui dont la valeur est de 55K ont 78 points.
Le corollaire pour la variable JOB donne aux clients qui travaillent dans l'administration 90 points, 80 points aux enseignants et 49 points aux auto-entrepreneurs et commerçants.

La table 10 nous donne notre grille de score finale, il est important de noter que chaque classe de risque correspond à une probabilité de défaut. En effet on observe comme prévu une baisse de la probabilité de défaut lorsque le score augmente. Les individus en bas (en rouge) ne devraient pas voir leur demande acceptée, ceux en haut(en vert) doivent avoir le prêt et ceux en ballottage (en orange) méritent une analyse métier plus approfondie afin de décider.

	Effectifs		**Probabilité de défaut**	
Classes de risques	**Test**	**Train**	**Test**	**Train**
[500, 650[	24	132	**0.833333**	**0.96212**
[650, 700[	65	254	**0.846154**	**0.724409**
[700, 750[	122	473	**0.581967**	**0.551797**
[750, 800[	196	738	**0.316327**	**0.257453**
[800, 850[	289	979	**0.155709**	**0.098059**
[850, 900[	273	1290	**0.040293**	**0.045736**
[900, 1000]	223	902	**0.004484**	**0.007761**

TABLE 9 – Classes de risques - Grille de score

Sur les 1192 individus formant le test set, 75% d'entre eux ont un score inférieur à 886 points, la moyenne étant à 824 et la médiane à 830.5 points. Le plus grand score est de 988 et le plus petit 596. Notons que la volatilité du score est de 78 points.
Dans la Figure 9, on voit que les distributions des scores dans le train et test ont les mêmes tendances, ce qui permet de confirmer sur sa stabilité.

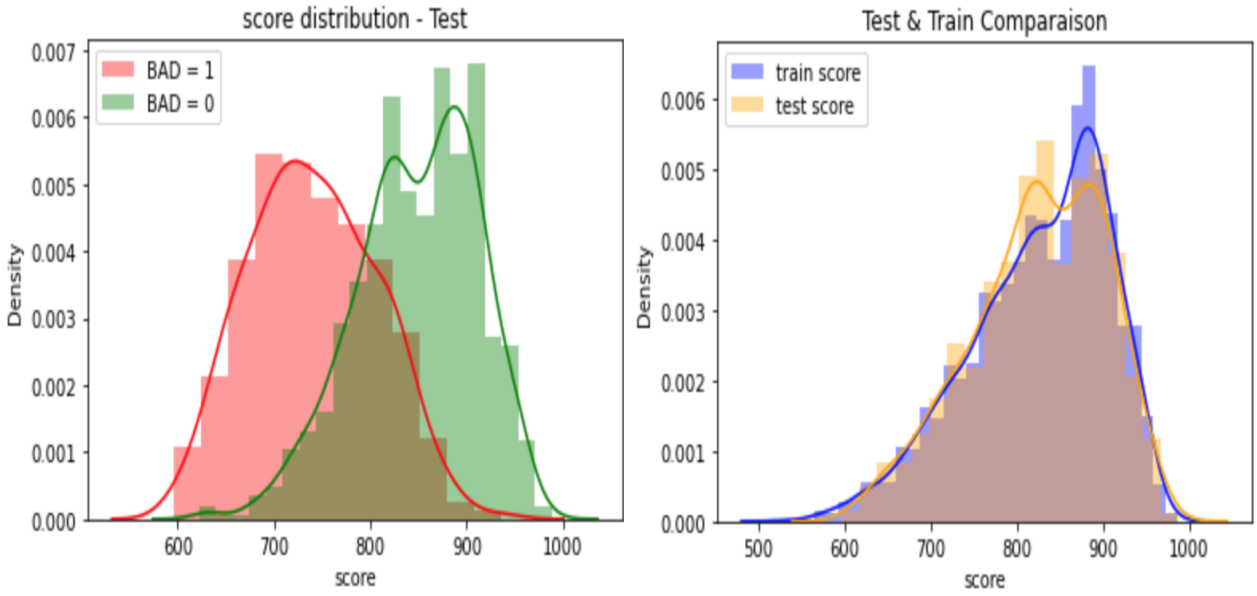

FIGURE 9 – Distribution des scores

6 Analyse comparative avec le Machine Learning

Nous avons tenté de challenger les résultats du modèle de scoring «classique» précédemment présenté par différents algorithmes de Machine Learning. Il s'agit du RandomForestClassifier, DecisionTreeClassifier, GradientBoostingClassifier, LinearDiscriminantAnalysis, StackingClassifier, VotingClassifier et KNeighborsClassifier. Afin d'optimiser au maximum nos résultats, on a présenté des jeux de paramètres à un algorithme de sélection le GridSearchCV qui par validation croisée va choisir pour chaque classificateur les paramètres qui maximisent l'AUC.
Les différents algorithmes de Machine Learning testés sont les suivants :

— **Random Forest** : est un algorithme d'apprentissage supervisé . La «forêt» qu'elle construit est un ensemble d'arbres de décision, généralement formés avec la méthode «ensachage». L'idée générale de la méthode d'ensachage est qu'une combinaison de modèles d'apprentissage, augmente le résultat global. En termes simples : une forêt aléatoire construit plusieurs arbres de décision et les fusionne pour obtenir une prédiction plus précise et plus stable.

— **Decision Tree** : est une technique d'apprentissage supervisé qui peut être utilisée à la fois pour les problèmes de classification et de régression, mais elle est principalement préférée pour résoudre les problèmes de classification. Il s'agit d'un classificateur à structure arborescente, où les nœuds internes représentent les caractéristiques d'un ensemble de données, les branches représentent les règles de décision et chaque nœud feuille représente le résultat.

— **Gradient Boosting** : Dans le Gradient Boosting, chaque prédicteur tente d'améliorer son prédécesseur en réduisant les erreurs. Mais l'idée fascinante derrière ceci est qu'au lieu d'ajuster un prédicteur sur les données à chaque itération, il ajuste en fait un nouveau prédicteur aux erreurs résiduelles commises par le prédicteur précédent.

— **LinearDiscriminantAnalysis** :c'est un algorithme d'apprentissage automatique de classification linéaire.L'algorithme consiste à développer un modèle probabiliste par classe basé sur la distribution spécifique des observations pour chaque variable d'entrée. Un nouvel exemple est ensuite classé en calculant la probabilité conditionnelle de son appartenance à chaque classe et en sélectionnant la classe dont la probabilité est la plus élevée.

— **StackingClassifier** : L'empilage est une technique d'apprentissage d'ensemble permettant de combiner plusieurs modèles de classification via un méta-classifieur.Ce dernier peut être entraîné sur les étiquettes de classe prédites ou sur les probabilités de l'ensemble.

L'idée de l'empilage est d'apprendre plusieurs apprenants faibles différents et de les en formant un méta-modèle pour produire des prédictions basées sur les multiples prédictions renvoyées par ces modèles faibles. Ainsi, nous devons définir deux choses afin de construire notre modèle d'empilage : les L apprenants que nous voulons adapter et le méta-modèle qui les combine.

— **VotingClassifier** : C'est un modèle d'apprentissage automatique qui s'entraîne sur un ensemble de nombreux modèles et prédit une sortie (classe) en fonction de la probabilité la plus élevée de la classe choisie comme sortie. Il agrège simplement les résultats de chaque classificateur transmis au Voting Classifier et prédit la classe de sortie en fonction de la plus grande majorité des votes. L'idée est qu'au lieu de créer des modèles dédiés distincts et de déterminer la précision de chacun d'entre eux, nous créons un modèle unique qui s'entraîne avec ces modèles et prédit la sortie sur la base de leur majorité combinée de votes pour chaque classe de sortie.

— **KNeighborsClassifier** : La méthode des K-plus proches voisins (KNN) est une méthode d'apprentissage supervisé extrêmement simple à implémenter dans sa forme la plus élémentaire, et pourtant souvent performante pour des tâches de classification complexes. L'algorithme KNN ne s'entraîne sur aucune donnée, mais utilise à chaque fois toutes les données dont il dispose pour classifier une nouvelle donnée. Le principe est le suivant : une donnée de classe inconnue est comparée à toutes les données stockées. La classe à laquelle est attribuée la nouvelle donnée est la classe majoritaire parmi ses K plus proches voisins au sens d'une distance choisie. Par défaut, la distance utilisée par la classe KNeighborsClassifier est la distance euclidienne.

Les résultats se présentent ci-après :
On remarque que globalement le RandomForestClassifier donne des résultats très bons. Concernant l'AUC il détient la pôle avec le StackingClassifier. On voit que le DecisionTreeClassifier est le pire des algorithmes. Notre problème étant orienté recall, on se rend compte que ses performances comparées au KNeighborsClassifier ne sont pas très bonnes. Le modèle le plus équilibré est le VotingClassifier qui présente le F1 score le plus élevé et son AUC est aussi appréciable. Toutefois, même si les algorithmes de ML se débrouillent pas mal, on préférera la logistique pour construire notre Grille de score en raison de son interprétabilité et de sa capacité à être facilement explicable.

Metriques	Accuracy	Precision	Recall	F1 Score	AUC	Gini
Algorithmes	**Cutoff = 0.45**					
LogisticRegression	0.78	0.50	0.80	0.61	0.88	0.75
RandomForestClassifier	**0.87**	**0.88**	0.49	0.63	**0.92**	**0.85**
DecisionTreeClassifier	0.76	**0.46**	0.42	**0.44**	**0.66**	**0.29**
GradientBoostingClassifier	0.83	0.71	**0.39**	0.50	0.87	0.74
LinearDiscriminantAnalysis	0.78	0.51	0.79	0.62	0.88	0.75
KNeighborsClassifier	**0.74**	**0.46**	**0.84**	0.59	0.87	0.74
VotingClassifier	0.85	0.65	0.71	**0.68**	0.89	0.79
StackingClassifier	**0.87**	0.82	0.56	0.66	**0.92**	0.78

TABLE 10 – Métriques des algorithmes de ML

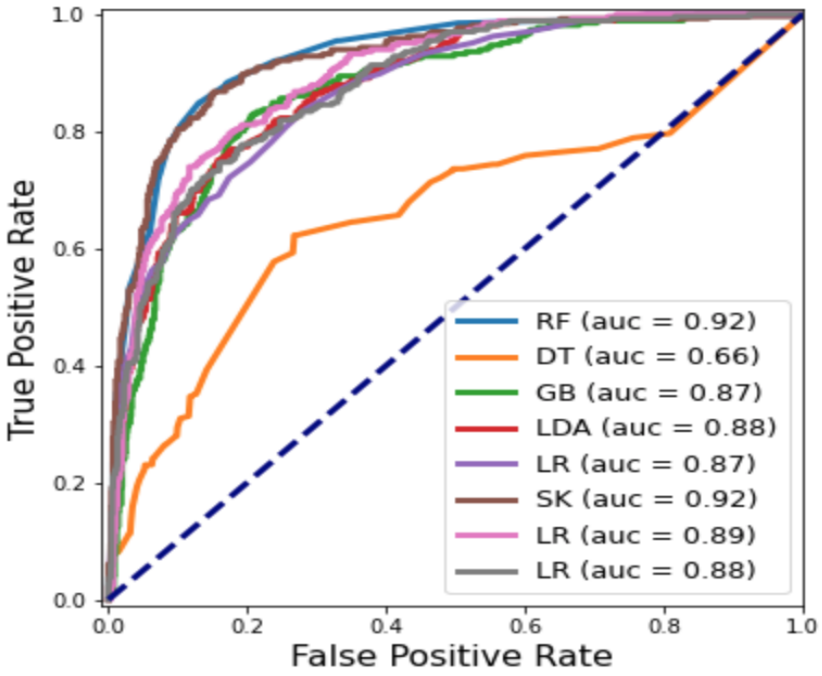

FIGURE 10 – ROC Curves algorithmes de ML

7 Conclusion

Ainsi, en se basant sur les données HMEQ et en appliquant notre méthodologie précise, nous avons pu créer notre grille de score répartie en classes de risque en fonction des probabilités de risque des individus. Ce modèle de scoring est fondé sur le meilleur modèle estimé que nous avons pu établir, utilisant l'algorithme le plus simple et le plus compréhensible qui est la régression logistique. Nous avons obtenu finalement une performance de prédiction importante et notamment un recall très intéressant. La qualité de notre grille scoring découle ainsi de la qualité et la performance de notre modèle de prévision.

Une fois notre modèle validé, il peut être converti en un outil d'aide à la décision. En effet, pour convertir le modèle statistique en un outil de prise de décision, l'organisation bancaire doit décider du score minimum requis pour accepter des clients. Ce dernier reflète le niveau de risque que l'organisation souhaite prendre, plus le score est bas et plus le nombre de clients recevant des services est élevé. Au

final, l'arbitrage est entre le volume du portefeuille et le profit par rapport au risque. Le seuil reflète les objectifs commerciaux de la banque , telles que la maximisation du profit, la réduction des coûts et l'amélioration de la productivité.

Table des figures

Liste des tableaux