MOBILIZE FINANCIAL SERVICES

MASTER ESA
Université d'Orléans

MODÉLISATION D'UN SCORE D'OCTROI DE CRÉDIT

BARRAUD Lorenzo
MIRZA Simon
VIEIRA DE BARROS Mathias

SOMMAIRE

Résumé non technique

Le score d'octroi, également connu sous les termes "score de crédit" ou "score de risque", revêt une importance cruciale dans le domaine de la finance, de la banque et de la gestion du crédit. Il occupe une place centrale dans la gestion des risques financiers.

Son objectif principal consiste à évaluer le risque associé à l'octroi de crédit à des individus, des entreprises ou des entités. Plus précisément, il remplit diverses fonctions essentielles, notamment :

- Prédire le risque de défaut en évaluant la probabilité qu'un emprunteur ne puisse pas honorer sa dette.
- Faciliter la prise de décision quant à l'octroi ou au refus d'un prêt, ainsi que les conditions qui lui seront associées.
- Optimiser les taux d'intérêt : les emprunteurs à faible risque bénéficient de taux d'intérêt plus avantageux.
- Justifier, auprès des organismes de réglementation, l'accord d'un crédit à un individu.

Ce score est élaboré à l'aide de techniques statistiques. Des modèles sont "entraînés" sur les données historiques des clients afin d'analyser leur comportement en matière de remboursement. Cela permet de créer un modèle prédictif qui relie les caractéristiques des individus aux probabilités de défaut.

L'un des avantages majeurs de ce score réside dans sa capacité à permettre aux prêteurs de prendre des décisions rapides et objectives. Il repose sur des modèles statistiques, éliminant ainsi les jugements subjectifs. De plus, il permet de classer les clients en catégories de "bons" et "mauvais" emprunteurs, ce qui présente un intérêt financier indéniable pour les institutions bancaires.

Néanmoins, la modélisation d'un score d'octroi peut présenter certaines difficultés. Elle dépend de la qualité des données, peut être sujette à des biais pouvant entraîner des discriminations envers certains groupes, et peut être compliquée par un faible nombre de clients ayant fait défaut, ce qui rend l'apprentissage du modèle plus complexe. De plus, certains modèles peuvent sembler opaques, compliquant la compréhension des raisons derrière une décision d'octroi de crédit.

Dans le cadre de ce projet, nous développerons un score d'octroi pour les demandes financées par le groupe Mobilize. Après avoir effectué un nettoyage de nos données et sélectionné les prédicteurs pertinents, nous mettrons en œuvre plusieurs méthodes de modélisation, à savoir une régression logistique et deux techniques d'apprentissage automatique. Nous utiliserons la régression logistique dans le logiciel SAS et les deux méthodes d'apprentissage automatique avec Python. Ensuite, nous comparerons leurs performances afin de conclure sur les avantages et les inconvénients de chacune d'entre elles.

Abstract

The credit score, also known as a "lending score" or "risk score", holds significant importance in the field of finance, banking, and credit management. It plays a central role in financial risk management.

Its primary objective is to assess the risk associated with extending credit to individuals, businesses, or entities. More specifically, it serves various essential functions, including:

- Predicting the risk of default by evaluating the probability that a borrower may fail to honor their debt.
- Facilitating decision-making regarding loan approval or denial and the associated terms.
- Optimizing interest rates: low-risk borrowers benefit from more favorable interest rates.
- Justifying, to regulatory bodies, the granting of credit to an individual.

This score is developed using statistical techniques. Models are "trained" on historical customer data to analyze their repayment behavior. This enables the creation of a predictive model that links individual characteristics to default probabilities.

One of the major advantages of this score is its ability to enable lenders to make quick and objective decisions. It relies on statistical models, thus eliminating subjective judgments. Furthermore, it allows the categorization of customers into "good" and "bad" borrowers, which is financially advantageous for banking institutions.

However, modeling a lending score may come with certain challenges. It depends on data quality, may be subject to biases that could lead to discrimination against certain groups, and can be complicated by a small number of customers who have defaulted, making model learning more complex. Additionally, some models may appear opaque, making it difficult to understand the reasons behind a credit approval decision.

Abstract

In the context of this project, we will develop a lending score for applications funded by the Mobilize Group. After cleaning our data and selecting relevant predictors, we will implement various modeling methods, namely logistic regression and two machine learning techniques. We will use logistic regression in SAS software and the two machine learning methods with Python. Subsequently, we will compare their performance to draw conclusions regarding the advantages and disadvantages of each.

I. Introduction

1. Présentation de la base

Le groupe Mobilize nous a fourni un ensemble de données comprenant 464 170 observations sur leurs clients qui ont bénéficié d'un financement sur la période allant de 2017 à 2021.

La base de données est constituée de 57 variables, mais seules 37 d'entre elles disposent d'une description et ont été conservées pour notre analyse. Parmi ces variables sélectionnées, on trouve 22 variables quantitatives, dont 2 correspondent respectivement aux identifiants du contrat et du client, ainsi que 10 variables qualitatives et 2 variables de type date.

Notre variable d'intérêt, *WE12c*, également appelée variable cible, est binaire et prend la valeur 1 si le client est tombé en défaut, et 0 dans le cas contraire. Ici, le défaut signifie que le contrat du client est devenu défaillant dans les 12 mois qui suivent son entrée en gestion.

Il est important de noter que, selon les données de l'entreprise, sur la période de janvier 2017 à janvier 2020, les demandes financées en défaut représentent en moyenne 1,79 % de l'ensemble des clients.

2. Exploration de la base

Afin de bien comprendre notre base de données et de nous projeter dans notre travail, il est essentiel de mener une analyse descriptive de nos variables. Cette étape nous permettra d'évaluer la qualité des données et de déterminer le niveau de données manquantes.

Nous avons utilisé diverses mesures pour analyser nos prédicteurs quantitatifs, notamment la moyenne, la médiane, l'écart-type et l'étendue[1]. Quant aux variables qualitatives, nous les avons représentées à l'aide de la répartition de leurs modalités[2] et de boxplots.

[1] Résumé des variables quantitatives en annexe (Table 1)

[2] Cf annexe pour repartition des variables qualitatives (Table 2)

Pour éviter d'alourdir notre rapport, nous avons choisi de ne pas afficher les statistiques descriptives pour l'ensemble des variables, à l'exception de la répartition de la variable cible qui est pertinente pour le reste du projet.

Voici comment se présente notre variable d'intérêt :

WE12c	Fréquence	Pourcentage
0	455845	98,20
1	8334	1,80

Nous remarquons que la fréquence de 1 est très faible, cela est similaire aux taux annoncés par Mobilize à la période de 2017 à 2020.

Dans l'ensemble, les observations manquantes sont rares dans notre base de données (moins de 0,0001 %), à l'exception des variables *nb_imp_tot* et *nb_imp_an_0*, qui présentent près de 70% de valeurs manquantes[3]. Cette situation s'explique par la nature de ces variables, car les valeurs manquantes correspondent aux nouveaux clients.

Par ailleurs, nous évaluer la présence de valeurs extrêmes pour chaque variable quantitative en se basant sur le critère interquartile. En additionnant les outliers identifiés pour chaque prédicteur, puis en les divisant par le nombre total d'observations, nous avons pu calculer la proportion de valeurs considérées comme extrêmes[4]. Ces analyses nous ont conduit à la conclusion que les valeurs aberrantes ne sont pas significatives, et qu'il ne sera probablement pas nécessaire de les traiter.

3. Modifications effectuées

Après un aperçu approfondi de notre base nous pouvons effectuer des modifications. Tout d'abord, les prédicteurs quantitatifs de notre base de données étant exprimés en centimes, cela paraît pertinent de les convertir en euros en les divisant par 100.

Nous avons également essayé de créer une variable que nous avons nommé la *part_finance_rev*. La variable est la division de *rev_tot* par *mt_finance* que l'on multiplie par cent. De ce fait nous obtenons la part du revenu mensuel par rapport au montant financé. Par exemple, si la variable était égale à 10, nous pouvions interpréter cela de la façon suivante : le revenu mensuel de l'individu représente 10% du montant financé. La variable *mt_finance* divisée par *rev_tot* n'était pas

[3] Part des valeurs manquantes en annexe (Table 3)

[4] Cf annexe pour la part des outliers selon le critère interquartile (Table 4)

possible dus aux revenus mensuels égaux à zéro. Cependant nous nous sommes aperçus par la suite que ce prédicteur n'était pertinent.

Comme indiqué ci-dessus, les observations présentant des valeurs manquantes sont peu nombreuses. Les prédicteurs traités étant qualitatifs, nous avons choisi d'imputer ces observations par leurs modes respectifs. Les deux variables *nb_imp_an_0* et *nb_imp_tot* citées précédemment ont néanmoins nécessité un remplacement manuel : Les valeurs manquantes ont ici été remplacées par « -1 », caractérisant le nouveau client.

Suite aux statistiques descriptives de nos variables, nous remarquons que certaines modalités peuvent et doivent être regroupées[5]. En effet, certaines classes contiennent moins de 5% des observations de la variable et/ou ne respectent pas un taux d'ascendance. La variable *region_* est par exemple regroupée en 5 grands territoires qui sont Nord-Est, Nord-Ouest, Sud-Est, Sud-Ouest et Centre.

Nous avons également décidé d'effectuer une discrétisation à l'aide du test du khi-deux normée permettant le regroupement des variables quantitatives en classe. Les intérêts de ces classes sont nombreux, elles permettent de simplifier les interprétations, sont moins coûteuses et d'intégrer les valeurs aberrantes dans les groupes extrêmes.

[5] Exemple de modalités à regrouper en annexe (Table 5)

II. Modélisation

1. Sélection des variables

La sélection des variables dans un modèle de crédit scoring est une étape fondamentale qui implique de choisir judicieusement les prédicteurs qui auront un impact significatif sur la variable cible, à savoir le risque de défaut de paiement. En économétrie, cette sélection repose sur plusieurs concepts et méthodes clés. Il faut noter qu'au-delà de l'aspect purement statistique, la banque doit pouvoir justifier auprès du client et du régulateur, tout choix vis-à-vis du modèle.

Analyse de la Significativité Statistique

Dans cette phase, il est crucial d'identifier les variables qui ont un pouvoir prédictif significatif pour expliquer la variation de la variable cible. Cela se fait en appliquant des tests statistiques appropriés à chaque prédicteur. Les tests de Wald, les tests du rapport de vraisemblance, ou les tests du score sont couramment utilisés pour évaluer la significativité de chaque variable. Ces tests comparent la variation expliquée par le modèle en présence de la variable à celle du modèle sans la variable, et déterminent si cette variation supplémentaire est statistiquement significative.

Les prédicteurs dont la significativité statistique est confirmée peuvent être inclus dans le modèle final. Cependant, il est important de prendre en compte le seuil de significativité, généralement défini à un niveau de confiance de 95% (valeur-p inférieure à 0,05), bien que ce seuil puisse varier en fonction des besoins spécifiques de l'analyse.

Critères d'Information

Outre la significativité, les critères d'information sont utilisés pour évaluer la pertinence des variables. Ces critères tiennent compte de la complexité du modèle en pénalisant les modèles qui incluent un grand nombre de variables. Deux critères

d'information couramment utilisés sont le critère d'information d'Akaike (AIC) et le critère d'information bayésien (BIC).

L'AIC et le BIC combinent la qualité de l'ajustement du modèle (la vraisemblance) avec une pénalité basée sur le nombre de variables. En général, un modèle avec un AIC ou un BIC plus bas est préféré, car il indique un meilleur équilibre entre ajustement et parcimonie. Cela signifie qu'il explique bien les données avec un nombre minimal de variables, ce qui facilite l'interprétation du modèle.

La validation croisée

L'objectif ici est d'entrainer notre modèle à l'aide de multiples échantillons d'apprentissage pour s'assurer de sa robustesse. Nous développerons la notion de validation croisée lors de l'utilisation du Machine Learning.

Méthodes de Machine Learning

La théorie de l'apprentissage statistique fournis des outils de sélection de variables tels que l'eslastic-net ou bien encore le modèle Penalised Logistic Tree Regression (PLTR). Ces approches n'ont pas été retenus dans la régression logistique car nous devions séparer l'économétrie « classique » avec les méthodes Machine Learning.

A) Sélection entre prédicteurs

Il est essentiel de rappeler l'objectif fondamental de la régression logistique : créer des classes de risques pour classifier les individus en fonction de leurs caractéristiques. La grille de score exige l'utilisation de variables qualitatives, et la discrétisation permet de satisfaire cette exigence, transformant ainsi notre ensemble de variables en qualitatives.

De plus, les méthodes de sélection telles que Stepwise, Backward et Forward ne seront pas employées dans l'estimation de notre modèle. Ces méthodes ont été initialement conçues pour des variables continues. L'élimination de certains prédicteurs par ces méthodes pourrait d'ailleurs compliquer davantage l'explication que l'utilisation d'une approche de sélection plus conventionnelle.

Les statistiques non paramétriques se révèlent être la solution idéale pour cette situation spécifique. Leur utilité réside dans l'analyse des relations entre les différents prédicteurs. En effet, l'intégration de prédicteurs fortement corrélés dans notre régression logistique pourrait engendrer un biais dans les résultats.

Afin de quantifier la force des relations entre les prédicteurs qualitatifs, nous adopterons le V de Cramer. Le V de Cramer est une mesure statistique de l'association entre deux variables catégorielles, largement utilisée pour analyser les données tabulées dans un tableau de contingence. Il repose sur le coefficient de contingence, qui évalue l'association entre les catégories de ces variables. Le résultat de cette mesure se situe dans une plage de 0 à 1. Plus le V de Cramer s'approche de 1, plus l'association entre les variables catégorielles est forte, indiquant ainsi une relation étroite. À l'inverse, un V de Cramer proche de zéro révèle une association faible, indiquant des relations moins marquées entre les variables.

Tableau d'interprétation :

Valeur du V de Cramer	**Intensité de la relation**
$0 \leq V < 0.1$	Très faible
$0.1 \leq V < 0.3$	Modérée
$0.3 \leq V < 0.5$	Assez forte
$0.5 \leq V$	Forte

Rappelons que notre base de données initiale comporte 37 variables dont la variable cible « WE12c ». Cette dernière ne sera donc pas considérée pour la première phase de sélection.

Tout d'abord, avant d'effectuer quelconques mesures, on s'aperçoit qu'il est inutile de garder certaines variables. C'est par exemple le cas de *no_cnt_crypte* et *no_par_crypte*. En effet, il y a peu d'intérêt d'avoir une variable qui permet d'identifier le client. Néanmoins dans certains cas ces variables peuvent être très utiles et pourraient permettre d'effectuer des fusions de table avec d'autres données. On aurait pu par exemple ajouter des données micro-économiques telles qu'une note comportementale, si par exemple le client est souvent à découvert ou bien encore prendre en compte la variation de ses flux de solde de son compte bancaire.

Examinons maintenant les corrélations entre les variables explicatives.

Table	Cramers_V	Chisq	p_value	Cramers_V_abs
CSP_classe * rev_men_autr2	0,6006657	334950,911	<.0001	,60067
appo_cptt_cnt2 * no_nat_prod2	0,6324753	185683,182	<.0001	,63248
REV_TOT2 * rev_men_autr2	0,7071141	464188,591	<.0001	,70711
diag_cli_rnva * nb_imp_tot2	0,7152119	474881,138	<.0001	,71521
diag_cli_rnva * nb_imp_an_0_2	0,7173528	477728,356	<.0001	,71735
mt_charges2 * tx_end_syex2	0,7440912	257002,755	<.0001	,74409
age_indv2 * anc_emp_indv2	0,7887155	288752,86	<.0001	,78872
anc_emp_indv2 * rev_men_autr2	0,8237262	314957,002	<.0001	,82373
nb_imp_an_0_2 * nb_imp_tot2	0,8286799	637513,054	<.0001	,82868
cpt_pai2_2 * nb_imp_an_0_2	0,8928043	739993,718	<.0001	,89280
CSP_classe * anc_emp_indv2	0,9693104	436125,221	<.0001	,96931

Le tableau ci-contre nous présente seulement les interactions entre les variables explicatives fortement corrélées (V de Cramer > 0.6).

Lorsque deux variables explicatives sont corrélées, il est recommandé de supprimer celle qui présente la moindre corrélation avec la variable cible, afin de respecter le principe de parcimonie.

B) Sélection des prédicteurs selon la variable cible

Lors de l'évaluation de la corrélation entre les variables explicatives et la variable cible, nous avons rencontré un problème : aucune variable ne présentait de corrélation significative avec la variable cible[6]. Par conséquent, nous avons opté pour la sélection des variables en fonction de critères tels que l'AUC, le Gini, le 10/X, ainsi que la significativité des paramètres associés aux modalités d'une variable, telle qu'elle est fournie par la régression logistique.

Certaines variables, comme *CSP_PERPHY*, n'ont pas réussi à démontrer leur significativité au sein de toutes leurs modalités. Nous avons donc pris la décision de les supprimer afin d'optimiser la parcimonie de notre modèle.

Les variables explicatives retenues dans notre modèle final sont :

- REV_TOT : **montant du revenu mensuel**
- part_ech : **part de l'échéance en pourcentage**
- mt_ttc_vech : **prix du véhicule**
- mt_finance : **montant financé**
- my_charges : **montant des charges**
- anc_adr_indv : **ancienneté de l'adresse**
- age_indv : **âge du client**
- nb_imp_an_0 : **nombre d'impayés**
- cd_natl_indv : **nationalité CEE ou hors CEE**
- etat_civ_prtc : **état civil**
- appo_cptt_cnt : **pourcentage d'apport**

[6] Table des corrélations en annexe (Table 6)

2. Méthode 1 : Régression logistique

A) Théorie

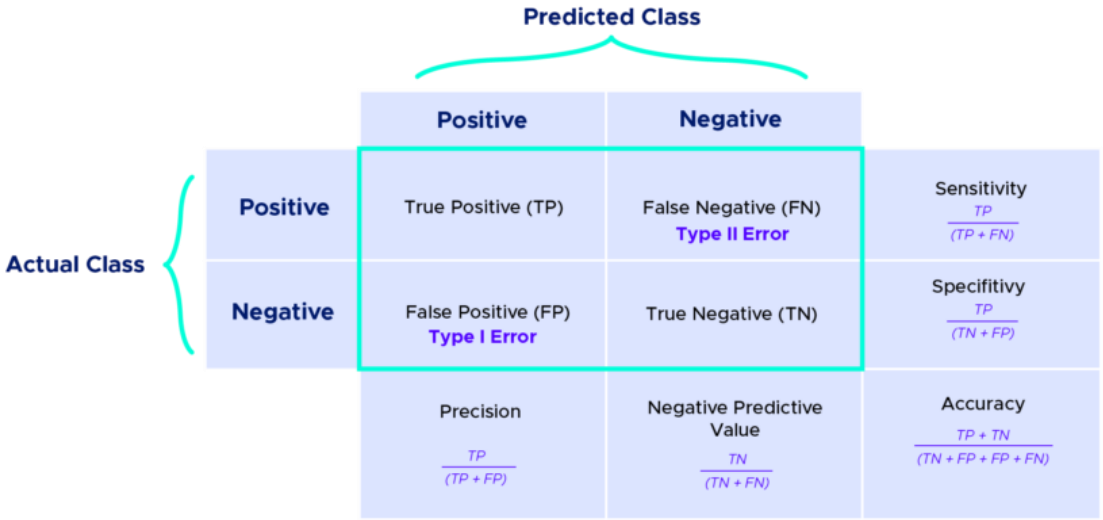
La régression logistique

Comme indiqué dans notre introduction, l'objectif de ce projet est de comparer un modèle classique à une approche basée sur le Machine Learning. Parmi les modèles classiques dont nous disposons, nous avons choisi d'utiliser la régression logistique, qui appartient à la catégorie des modèles probabilistes pour la réponse binaire.

Le principe de la régression logistique est simple : elle estime des coefficients associés aux variables explicatives en utilisant la méthode du maximum de vraisemblance. Ces coefficients, combinés aux valeurs des variables explicatives, permettent à la régression logistique de générer un score. Pour obtenir une probabilité, ce score est ensuite introduit dans la fonction de répartition logistique. Il est important de noter que la probabilité estimée ne dicte pas de règle de décision ; c'est à la banque de déterminer le seuil à partir duquel elle considère si un client est susceptible de faire défaut ou non.

Matrice de confusion

Dans les problèmes de classification, un modèle prédit des résultats que l'on doit comparer à la réalité pour mesurer son degré de performance. On utilise généralement la matrice de confusion, appelée aussi tableau de contingence. Elle mettra non seulement en valeur les prédictions correctes et incorrectes mais nous donnera surtout un indice sur le type d'erreurs commises.



On classe les résultats en 4 catégories :

- True Positive (TP) : la prédiction et la valeur réelle sont positives. Exemple : Une personne malade et prévu malade.
- True Negative (TN) : la prédiction et la valeur réelle sont négatives. Exemple : Une personne saine et prévu saine.
- False Positive (FP) : la prédiction est positive alors que la valeur réelle est négative. Exemple : Une personne saine et prévu malade.
- False Negative (FN) : la prédiction est négative alors que la valeur réelle est négative. Exemple : Une personne malade et prévu saine.

On retrouve ci-dessous les manières les plus communes de tirer des informations intéressantes de ce genre de tableau, on appelle ces indicateurs des métriques :

- Accuracy **:** proportion d'individus correctement prédit par le modèle
- Specificity **:** proportion d'individus n'ayant pas connu l'événement correctement identifiés par le modèle
- Sensitivity **:** proportion d'individus ayant connu l'événement correctement identifiés par le modèle

Il est essentiel de préciser que la matrice de confusion est établie en fonction d'un seuil de décision. Chaque individu est associé à une probabilité de présenter un défaut, et en fonction de ce seuil, nous décidons si l'individu sera classé comme présentant un défaut ou non. Par exemple, si le seuil de décision est fixé à 0,3 et que la probabilité d'un individu de présenter un défaut est de 0.38, notre prédiction serait qu'il présentera un défaut.

Le choix du seuil (ou cut-off) dans un modèle de scoring peut être déterminé de manière statistique ou exogène. En général, les banques peuvent sélectionner leur cut-off en fonction de trois principaux objectifs :

- **Objectif de part de marché :** En utilisant la courbe de sélection, la banque peut décider du nombre d'individus qu'elle souhaite inclure dans son portefeuille de prêt. Cela lui permet de mesurer la part de défaut qu'elle sera en mesure de capturer parmi ces emprunteurs. Ainsi, elle peut équilibrer le volume d'octroi de crédit avec son appétit pour le risque.
- **Objectif de taux de sélection :** Toujours en se basant sur la courbe de sélection, la banque peut fixer un seuil pour la proportion de défauts qu'elle souhaite identifier. En ajustant le cut-off en conséquence, elle peut observer la part de marché associée à cette sélection. Cette approche permet de

cibler spécifiquement les emprunteurs à risque tout en maintenant un certain volume de prêts.

- **Fonction objectif :** Dans ce cas, le choix du cut-off ne repose plus uniquement sur des critères statistiques, mais intègre également des considérations de rentabilité. L'objectif est de maximiser la rentabilité du portefeuille de prêt. Cela signifie que la banque peut accepter des emprunteurs dont le risque de défaut est plus élevé, tant que le rendement généré par ces prêts compense les pertes attendues.

Il est essentiel de souligner que l'objectif premier d'une entreprise, y compris une institution financière, est généralement de maximiser sa rentabilité. Par conséquent, il est possible de définir une fonction de rentabilité qui prend en compte à la fois les revenus attendus des prêts et les pertes attendues en raison des défauts. En utilisant cette fonction de rentabilité, une banque peut construire un programme de sélection de seuil visant à maximiser ses profits tout en contrôlant le risque de crédit.

En fin de compte, le choix du cut-off dépend des priorités stratégiques de la banque, de son appétit pour le risque, et de ses objectifs commerciaux, tout en respectant les contraintes réglementaires en matière de risque et de conformité.

On peut poser un programme visant à maximiser cette rentabilité :

$$\mathcal{A}^* = \arg\max_{\mathcal{A}} R(\mathcal{A})$$

$$\begin{aligned} R = {} & gP[1_{\mathcal{A}}(X) \mid Y = 0]P[Y = 0] \\ & -c_1 P[1_{\mathcal{A}}(X) \mid Y = 1]P[Y = 1] \\ & -c_0 P\left[1_{\overline{\mathcal{A}}}(X) \mid Y = 0\right] P[Y = 0] \end{aligned}$$

Ici, la fonction de rentabilité repose sur la notion de gain résultant d'une décision judicieuse, auquel sont soustraites les pertes découlant de décisions erronées. On considère un gain lorsque l'on accorde un crédit à un client qui ne fera pas défaut ($gP[1_{\mathcal{A}}(X) \mid Y = 0]P[Y = 0]$), tandis que des pertes surviennent lorsque l'on accorde un crédit à un client qui finira par faire défaut ($c_1 P[1_{\mathcal{A}}(X) \mid Y = 1]P[Y = 1]$), ou lorsque l'on refuse un crédit à un client qui aurait pu honorer son engagement ($c_0 P[1_{\overline{\mathcal{A}}}(X) \mid Y = 0]P[Y = 0]$). La région d'acceptation des clients, notée $\mathcal{A}$, dépend donc de cette fonction de rentabilité.

Dans notre cas, Mobilize Financial Service n'a pas fourni les informations nécessaires pour déterminer le seuil de cut-off qui maximiserait la rentabilité. Cependant, il est possible d'examiner plusieurs points de vue concernant les pertes.

Tout d'abord, on peut considérer qu'accorder un crédit à un client qui finira par faire défaut représente un coût plus élevé que le coût de ne pas accorder de crédit à un bon client. Dans le premier cas, la banque subit une perte nette qu'elle ne pourra pas récupérer, tandis que dans le second cas, bien qu'elle ne gagne rien, elle ne subit aucune perte.

Cependant, il est important de prendre du recul. Dans le cas où l'on ne fournit pas de crédit à un client qui aurait finalement remboursé, on parle de coût d'opportunité. Ce coût doit également être pris en compte, car la banque se prive d'un bénéfice potentiel, qui est quantifiable. Il pourrait s'agir des intérêts que le client aurait versés ou encore de la fidélisation de la clientèle que l'octroi du crédit aurait pu engendrer.

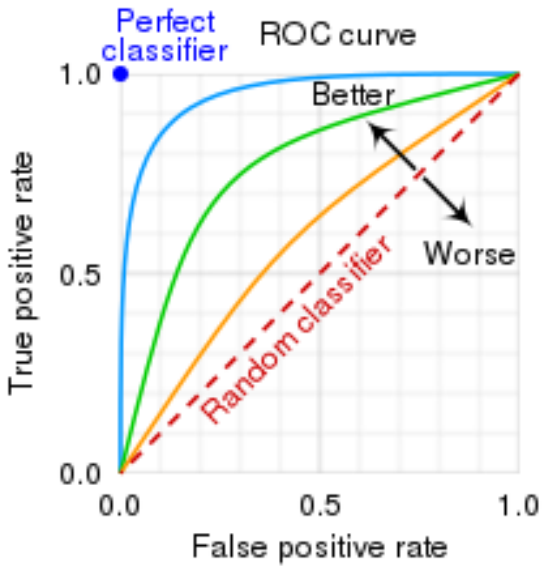
De plus, les banques ont recours à des assurances pour se prémunir contre le risque de défaut. Il est donc essentiel de ne pas surestimer ce coût. Cependant, notamment dans le cas des crédits hypothécaires, les assureurs sont de plus en plus réticents à couvrir ce risque, en raison de la multiplication des événements climatiques et d'autres facteurs.

Dans notre analyse, nous avons utilisé des matrices de confusion pour évaluer la performance du modèle en utilisant un seuil qui permet d'obtenir un nombre égal de clients correctement classés comme non-défaillants et de clients correctement classés comme défaillants. Statistiquement, ce seuil optimal a été déterminé sur notre modèle de régression logistique en équilibrant la sensibilité et la spécificité. Ce seuil servira de point de comparaison pour les méthodes de machine learning que nous allons examiner.

ROC (Receiver Operating Characteristic curve)

La courbe ROC est un instrument essentiel pour évaluer et comparer les capacités prédictives de différents modèles. Elle représente toutes les combinaisons de valeurs de (1-Spécificité, Sensitivité) en fonction du seuil de classification. Sur l'axe horizontal, nous avons la 1-spécificité, tandis que sur l'axe vertical, nous représentons la sensitivité.

Le modèle qui est associé à la bissectrice correspond à un modèle "aléatoire". Comme représenté sur le graphique ci-dessus, plus la courbe du modèle se rapproche du coin supérieur gauche, plus le modèle classe correctement les individus.



AUC

L'indicateur discriminant entre différentes courbes ROC est l'AUC (Area Under the Curve), qui représente l'aire sous la courbe. Comme nous l'avons mentionné précédemment, plus la courbe ROC se rapproche du coin supérieur gauche, meilleure est la performance du modèle. Par conséquent, il est souhaitable d'obtenir la plus grande valeur possible pour l'AUC. Un modèle parfait se traduit par une valeur de l'AUC égale à 1, tandis qu'un modèle purement "aléatoire" donnera une valeur de 0,5. Selon la littérature, un modèle est généralement considéré acceptable lorsque son AUC dépasse 0,65 en moyenne.

Indice de Gini

Un second indicateur lié directement avec l'AUC est l'indice de Gini, qui permet de mesurer le niveau d'inégalité dans la distribution d'une variable. Il se situe entre 0 et 1, où 0 représente une répartition parfaitement égale (tous les individus ont la même part) et 1 représente la plus grande inégalité possible (un seul individu détient tout). Ce dernier est calculé de manière suivante :

Indice de Gini = (2*AUC) - 1

Indice 10/X

L'indice 10/X est calculé à l'aide d'une courbe qui représente le pourcentage de défauts identifiés en fonction du classement décroissant des individus en fonction de leurs scores. L'objectif est de déterminer le pourcentage de défauts parmi les 10% d'individus ayant les scores les plus bas. Par exemple, lorsque x est égal à 10% et que la courbe atteint un niveau de y égal à 65%, cela signifie que les 10% d'individus avec les scores les plus bas expliquent 65% des défauts, ce qui se traduit par un indice de 10/65. Par conséquent, il est souhaitable d'obtenir la valeur la plus élevée possible pour les défauts parmi les 10% d'individus ayant les scores les moins élevés.

B) Évaluation du modèle

Afin d'évaluer les performances de notre modèle, nous avons divisé notre ensemble de données en deux parties : un échantillon d'apprentissage et un échantillon de test. Ces échantillons sont stratifiés, car nous avons une sous-représentation des cas de défaut, et il est essentiel d'avoir suffisamment de cas de défaut à la fois pour l'estimation de notre modèle et pour l'évaluation de ses performances. Il est important de noter que, sur l'ensemble du projet, ces échantillons resteront les mêmes, que ce soit pour la régression logistique ou pour l'approche basée sur le Machine Learning. Notre objectif est de comparer les deux méthodes dans des conditions d'apprentissage et d'évaluation des performances similaires.

Echantillon d'apprentissage stratifié

WE12c	Fréquence	Pourcentage	Fréquence cumulée	Pourcentage cumulé
0	319092	98,20	319092	98,20
1	5834	1,80	324926	100,00

Echantillon test stratifié

WE12c	Fréquence	Pourcentage	Fréquence cumulée	Pourcentage cumulé
0	136753	98,20	136753	98,20
1	2500	1,80	139253	100,00

De cette manière, nous disposons de la même proportion de défaut sur les deux échantillons.

Nous avons donc l'ensemble des coefficients pour chaque modalité associée à leurs variables effectives :

Analyse des valeurs estimées du maximum de vraisemblance						
Paramètre	**Modalité**	**DDL**	**Estimation**	**Erreur type**	**Khi-2 de Wald**	**Pr > khi-2**
Intercept		1	-4,0374	0,1578	654,2131	<.0001
REV_TOT2	**[0.04;130000]**	1	1,2868	0,0641	403,1392	<.0001
REV_TOT2	**]130000;169266]**	1	0,9610	0,0570	284,0699	<.0001
REV_TOT2	**]169266;199200]**	1	0,8044	0,0589	186,4931	<.0001
REV_TOT2	**]199200;218000]**	1	0,7163	0,0666	115,6824	<.0001
REV_TOT2	**]218000;250000]**	1	0,6782	0,0583	135,1852	<.0001
REV_TOT2	**]250000;295000]**	1	0,6480	0,0558	134,8768	<.0001
REV_TOT2	**]295000;335000]**	1	0,4777	0,0590	65,6523	<.0001
REV_TOT2	**]335000;400900]**	1	0,2476	0,0586	17,8520	<.0001
part_ech2	**[0.01;1.08]**	1	-0,4746	0,0694	46,7394	<.0001
part_ech2	**]1.08;1.23]**	1	-0,5850	0,0641	83,2668	<.0001
part_ech2	**]1.23;1.36]**	1	-0,5240	0,0494	112,6000	<.0001
part_ech2	**]1.36;1.52]**	1	-0,4202	0,0372	127,3792	<.0001
part_ech2	**]1.52;1.66]**	1	-0,2756	0,0372	55,0353	<.0001
mt_ttc_veh2	**[199898;2509976]**	1	-0,2721	0,0401	45,9413	<.0001
mt_finance2	**[199898;799951.8]**	1	-0,4118	0,1162	12,5582	0,0004
mt_finance2	**]1120069;1312850]**	1	0,1757	0,0573	9,3971	0,0022
mt_finance2	**]1312850;23597912]**	1	0,4811	0,0466	106,3673	<.0001
mt_charges2	**[0;25000]**	1	0,2722	0,0462	34,6720	<.0001
anc_adr_indv2	**[0;107]**	1	0,3868	0,0309	156,5036	<.0001
age_indv2	**[17;26]**	1	0,8875	0,0607	213,7544	<.0001
age_indv2	**]26;34]**	1	0,7542	0,0563	179,3173	<.0001
age_indv2	**]34;38]**	1	0,7312	0,0631	134,4476	<.0001
age_indv2	**]38;44]**	1	0,6879	0,0554	154,0039	<.0001

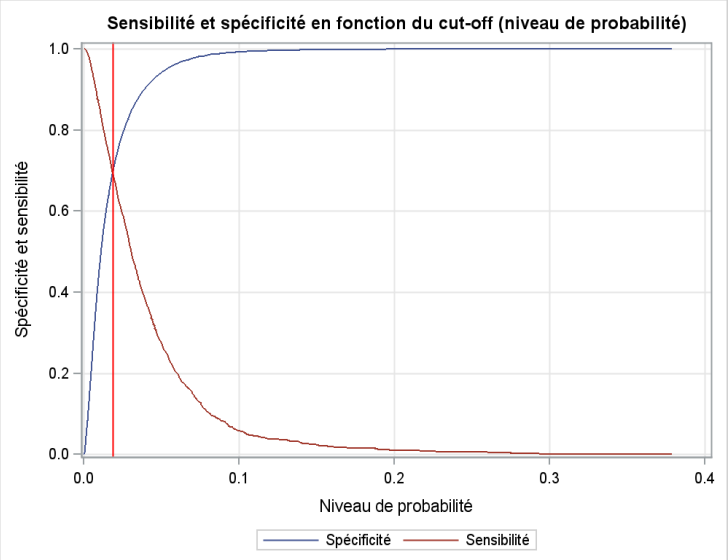
age_indv2	**]44;48]**	1	0,4725	0,0604	61,2832	<.0001
age_indv2	**]48;55]**	1	0,2496	0,0543	21,1404	<.0001
age_indv2	**]55;66]**	1	0,1177	0,0511	5,3060	0,0213
nb_imp_an_0_2	**0**	1	-0,3109	0,0350	78,8602	<.0001
nb_imp_an_0_2	**>1**	1	1,3250	0,0540	601,8330	<.0001
cd_natl_indv2	**CEE**	1	-0,8223	0,1258	42,7342	<.0001
eta_civ_prtc2	**Celibataire**	1	0,2695	0,0481	31,3629	<.0001
eta_civ_prtc2	**Marié**	1	-0,2050	0,0499	16,8894	<.0001
eta_civ_prtc2	**Séparé/Divorcé/Veuf**	1	0,3986	0,0557	51,1926	<.0001
appo_cptt_cnt2	**]0.57;8.47]**	1	-0,2239	0,0439	26,0037	<.0001
appo_cptt_cnt2	**]11.28;17.46]**	1	-0,5220	0,0559	87,0847	<.0001
appo_cptt_cnt2	**]17.46;28.74]**	1	-0,5732	0,0606	89,6215	<.0001
appo_cptt_cnt2	**]28.74;50]**	1	-0,5320	0,0694	58,7210	<.0001
appo_cptt_cnt2	**]50;97.62]**	1	-1,1110	0,1392	63,7403	<.0001
appo_cptt_cnt2	**]8.47;11.28]**	1	-0,2190	0,0623	12,3620	0,0004

L'ensemble des coefficients estimés sont significatifs. Leur p-value étant pour tous inférieure au seuil de 5%.

C) Indices de performances

Le cut-off sélectionné pour la matrice de confusion est celui qui égalise la sensibilité et la spécificité, on le considèrera comme étant « optimal » :

Ici, le cut-off optimal est environ égal à 0.0188.

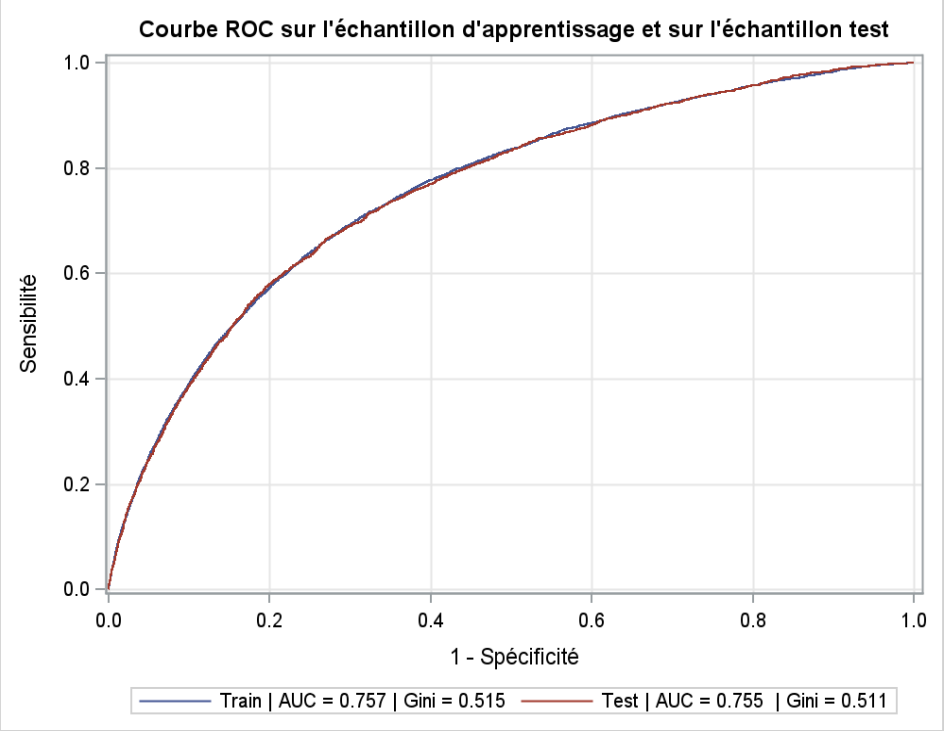


La matrice de confusion associée

Matrice de confusion associée à WE12C : cut-off 0.0188			
Valeurs observées	**Valeurs prédites**		
Fréquence Pourcentage Pct de ligne Pct de col	**0**	**1**	**Total**
0	94996 68.22 69.47 99.20	41757 29.99 30.53 96.01	136753 98.20
1	763 0.55 30.52 0.80	1737 1.25 69.48 3.99	2500 1.80
Total	95759 68.77	43494 31.23	139253 100.00

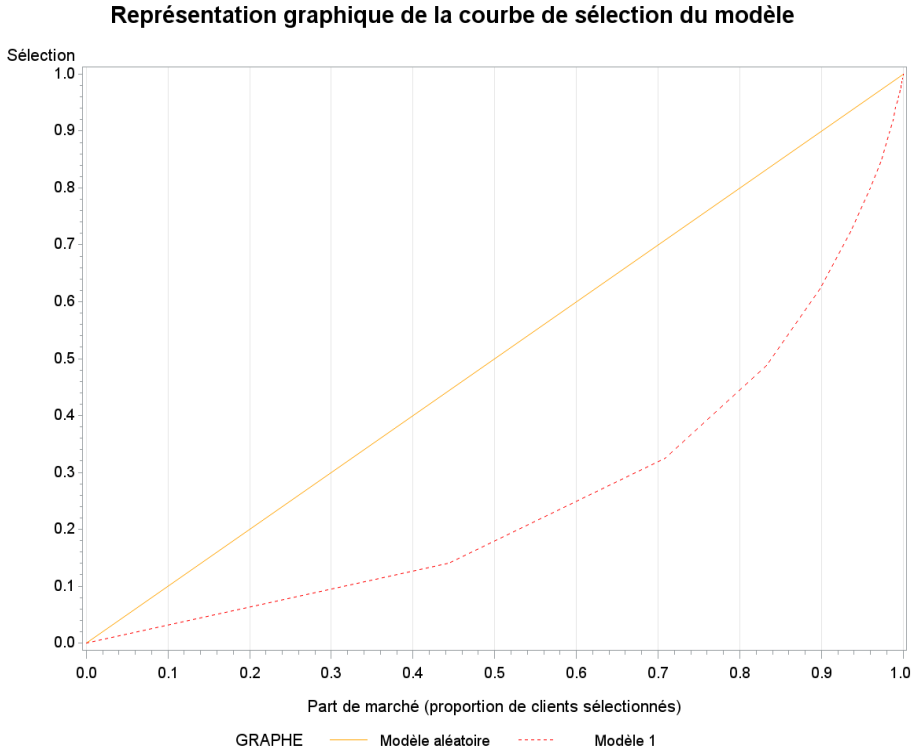
On observe qu'il y 69.47% de bonne prédictions pour les clients qui n'ont pas fait défaut et 69.48% pour ceux qui ont fait défaut.

Un autre moyen de visualiser la robustesse de notre modèle, est la ROC. Le graphique ci-contre compare la ROC, l'AUC et le Gini de notre échantillon d'apprentissage et du test.



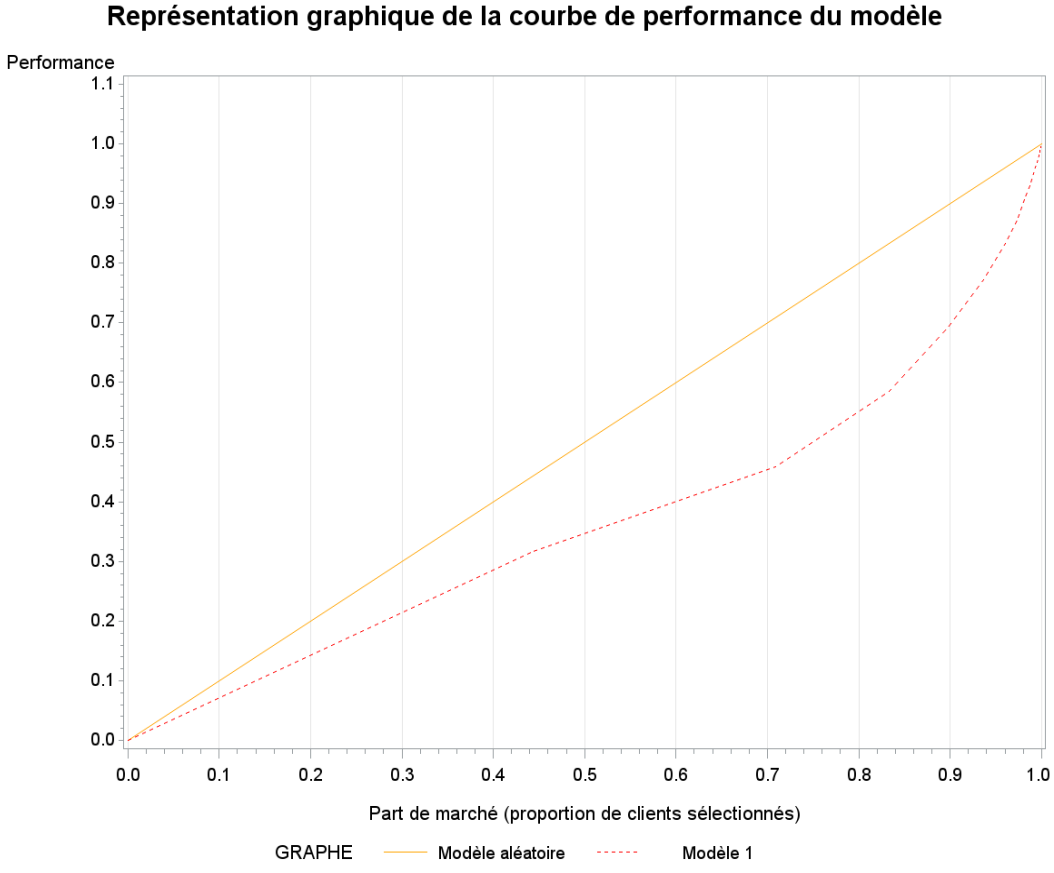
Les deux courbes ROC se superposent. Notre modèle a un très bon pouvoir de généralisation. En clair, sur des nouvelles données, il donne des performances similaires que lors de sa phase d'apprentissage.

Courbe de sélection

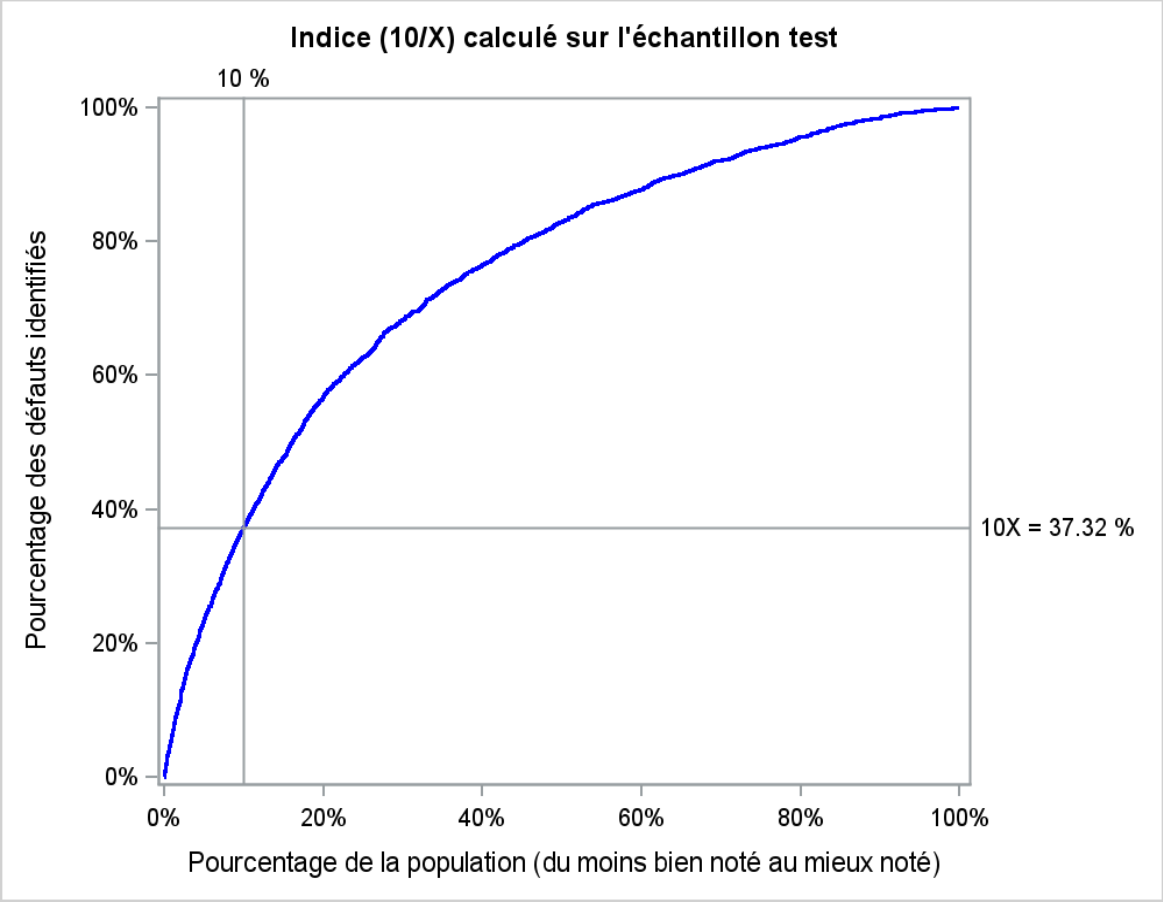


On peut interpréter de la sorte : si nous choisissons de sélectionner 50% des individus les plus mal notés, notre score sélectionnera moins de 20 % de défaut.

Courbe de performance



Courbe 10/X



Ainsi avec la régression logistique classique, dans les 10% de la population la plus mal notée, on arrive à capter 37,32 % de défauts.

D) Interprétation

Label	Variable	Modalité	Poids	Taux de défaut	Répartition	Contribution
Age	AGE_INDV2	[17;26]	0	4,45%	5,41%	11,83%
		]26;34]	15	3,48%	9,19%	
		]34;38]	17	2,68%	5,61%	
		]38;44]	23	2,10%	9,80%	
		]44;48]	49	1,74%	8,41%	
		]48;55]	76	1,52%	15,62%	
		]55;66]	93	1,16%	22,83%	
		]66;98]	107	1,00%	23,13%	

Ancienneté de l'adresse	ANC_ADR_INDV2	[0;107]	0	2,35%	53,12%	5,14%
		]107;110]	47	1,17%	46,88%	
Nationalité	CD_NATL_INDV2	Hors CEE	0	5,11%	0,41%	8,23%
		CEE	92	1,78%	99,59%	
Etat civil	ETA_CIV_PRTC2	Séparé/Divorcé/Veuf	0	2,27%	14,88%	7,37%
		Celibataire	16	3,12%	24,05%	
		Union libre	48	2,19%	9,83%	
		Marié	72	0,96%	51,24%	
Montant des charges	MT_CHARGES2	[0;25000]	0	3,12%	5,58%	3,01%
		]25000;1186200]	33	1,72%	94,42%	
Montant financé	MT_FINANCE2	]1312850;23597912]	0	2,16%	61,96%	21,13%
		]1120069;1312850]	45	1,92%	12,18%	
		]799951.8;1120069]	74	1,27%	14,93%	
		[199898;799951.8]	168	0,32%	10,93%	
Prix du véhicule	MT_TTC_VEH2	]2509976;25597041]	0	1,59%	16,95%	2,55%
		[199898;2509976]	26	1,24%	83,05%	
Nombre d'impayés	NB_IMP_AN_0_2	>1	0	9,77%	1,51%	12,47%
		nouveau client	162	1,88%	69,97%	
		0	195	1,17%	28,52%	
Part de l'échéance en pourcentage	PART_ECH2	]1.66;75.75]	0	2,90%	20,01%	14,19%
		]1.52;1.66]	32	2,31%	17,97%	
		]1.36;1.52]	54	1,77%	22,17%	
		]1.23;1.36]	79	1,50%	13,05%	
		]1.08;1.23]	103	0,85%	9,85%	
		[0.01;1.08]	105	0,76%	16,96%	
Revenu mensuel	REV_TOT2	[0.04;130000]	0	3,21%	5,02%	14,09%
		]130000;169266]	40	3,00%	9,97%	
		]169266;199200]	60	2,42%	8,96%	
		]199200;218000]	70	2,28%	6,06%	
		]218000;250000]	75	1,89%	10,08%	
		]250000;295000]	78	1,80%	11,93%	
		]295000;335000]	98	1,62%	10,81%	
		]335000;400900]	125	1,32%	13,96%	
		]400900;76061300]	155	0,93%	23,21%	

3. Méthode 2 : Régression logistique avec SMOTE

Comme nous l'avons remarqué précédemment, nous avons une sous-représentation des défauts (1.8% d'effectif). Nous allons donc augmenter l'effectif des individus défaillants (la classe minoritaire) afin d'améliorer l'apprentissage, et donc les performances du modèle.

A) Théorie

En effet, les données déséquilibrées dans les problèmes de classification posent de nombreuses difficultés, notamment lors de la modélisation. L'une des solutions pour traiter les données déséquilibrées est de les "rééquilibrer".

Ce type d'approches se décline sous 2 formes principales :

1) Le sous-échantillonnage (undersampling). Parmi les individus majoritaires, on en retire une partie afin d'accorder plus d'importance aux individus minoritaires. Cette approche permet de diminuer la redondance des informations apportées par le grand nombre d'individus majoritaires.

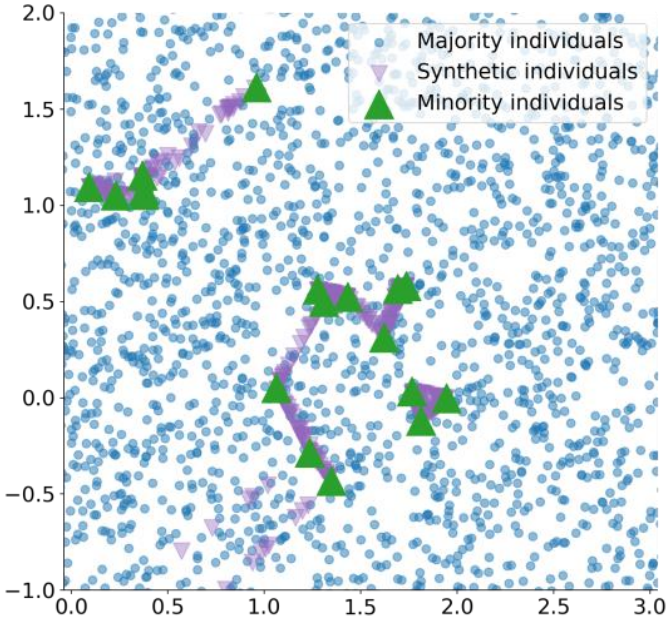
2) Le sur-échantillonnage (oversampling). Le nombre d'individus minoritaires est augmenté en les répliquant pour qu'ils aient plus d'importance lors de la modélisation. Différentes solutions sont possibles, notamment le SMOTE.

   - Le SMOTE (Synthetic Minority Oversampling Technique) est une méthode d'oversampling, mais ce qui la distingue, c'est que de nouvelles observations ne sont pas simplement dupliquées, mais plutôt générées en tant qu'individus distincts. Ces derniers sont créés de manière à présenter des similitudes avec les individus les plus proches dans l'espace, en se basant sur leurs k voisins les plus proches. Bien sûr, il est essentiel de spécifier l'effectif souhaité pour la classe minoritaire que l'on cherche à atteindre. Pour le choisir, il est nécessaire de tenir compte de la nature du problème, des données, et des besoins spécifiques de votre tâche d'apprentissage automatique. En bref, l'objectif de ces techniques est d'apporter le plus d'information à l'ensemble d'apprentissage permettant ainsi au modèle d'acquérir une compréhension plus approfondie des cas au sein de la classe minoritaire, évitant ainsi la sous-estimation de ces occurrences.

Pour créer un individu synthétique, les étapes définies dans l'algorithme du SMOTE sont les suivantes :

a. Sélectionner aléatoirement une observation minoritaire "initiale".
b. Identifier ses k plus proches voisins parmi les observations minoritaires (où k est un paramètre défini par l'utilisateur).
c. Choisir aléatoirement l'un des k plus proches voisins (par défaut k=5).
d. Générer aléatoirement un coefficient 0< alpha <1
e. Créer un nouvel individu entre l'observation initiale et le plus proche voisin choisi, selon la valeur du coefficient. Par exemple, si alpha=0.5, le nouvel individu sera positionné à mi-chemin entre l'observation initiale et le plus proche voisin choisi.

Voici ci-dessous une représentation graphique d'un SMOTE :



Application du SMOTE à un couple de variables numériques. Les points bleus sont les individus majoritaires, les triangles verts les individus minoritaires et les triangles violets les individus synthétiques générés par SMOTE.

Cette illustration montre que les individus "synthétiques" sont créés de manière à ce qu'ils soient proches de vrais individus en termes de valeur au niveau des variables explicatives.

L'utilisation du SMOTE peut améliorer considérablement l'entraînement du modèle, mais il faut l'utiliser avec prudence lors de la validation et du test. Pour évaluer un modèle de Machine Learning, divisez les données en ensembles d'entraînement, de validation et de test. La validation garantit que le modèle fonctionnera avec des données futures. Ne pas appliquer le SMOTE aux données de validation et de test, car cela peut donner des performances artificielles. En effet, tester notre modèle sur des données qui ont été créées artificiellement peut être risqué. En général, les métriques de classification étant sensibles au déséquilibre de classes, les données de validation doivent refléter les données réelles pour des performances réalistes.

B) Evaluation du modèle

L'effectif à faire atteindre à la classe minoritaire étant un paramètre à calibrer nous avons choisi de réaliser un SMOTE à 5% et un à 10% pour la raison suivante :

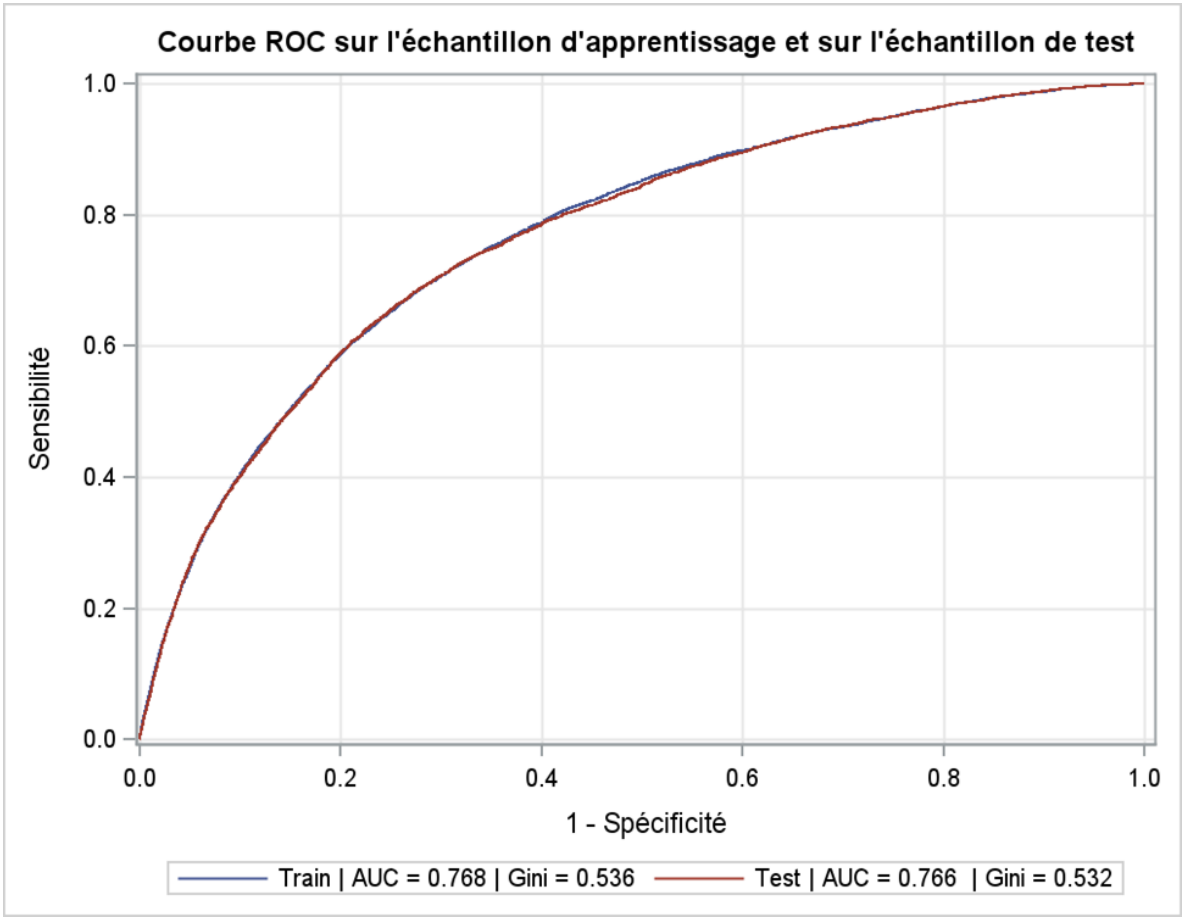
Puisque nos choix ont été pris sur le dataset déséquilibré, qui reflète la réalité, cela sera plus judicieux de calibrer notre modèle sur un dataset qui se rapproche de la réalité. En effet, la proportion de défaut dans le dataset équilibré ne changera pas de manière significative avec un effectif de 5 ou 10%. Équilibrer un jeu de données de manière stricte en adoptant un ratio de 50/50 en fonction du défaut risquerait d'introduire un biais dans la représentation de la réalité, car il est largement reconnu que les taux de défaut dans le secteur bancaire sont généralement faibles.

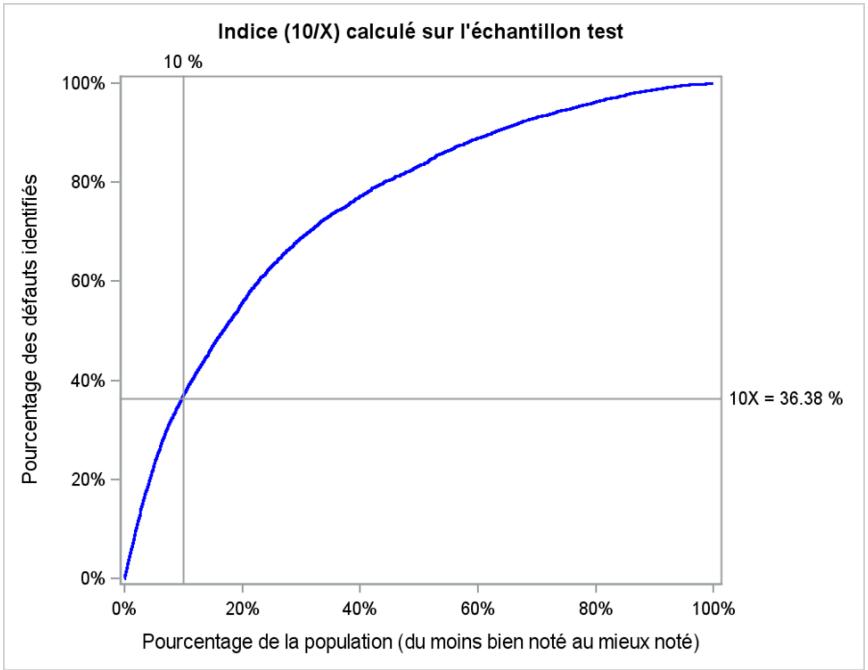
La base de données contient 464 179 individus dont 8 334 défaillants et 455 845 non défaillants.

L'objectif d'un SMOTE à 5% (resp. 10%) est de faire atteindre la classe minoritaire à 5% (resp. 10%) de l'effectif total de la classe majoritaire.

- SMOTE 5% : 478 637 individus dont 455 845 non défaillants et 22 792 défaillants
- SMOTE 10% : 501 429 individus dont 455 845 non défaillants et 45 584 défaillants

SMOTE 5%





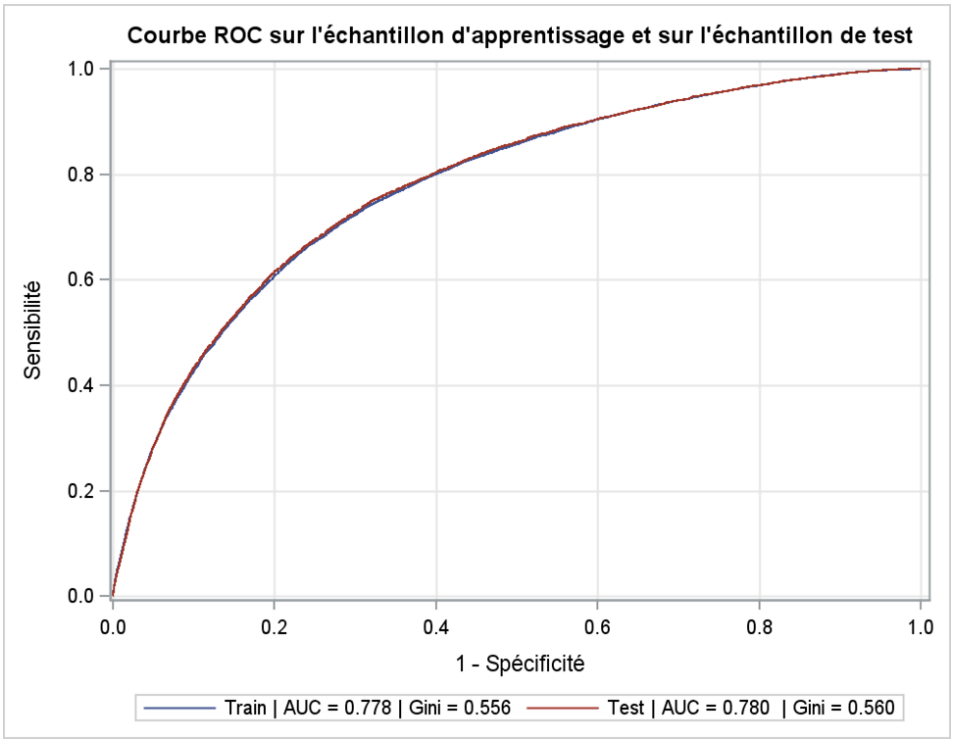
D'après la courbe ROC et Gini, les performances sont légèrement meilleures par rapport à la régression classique.

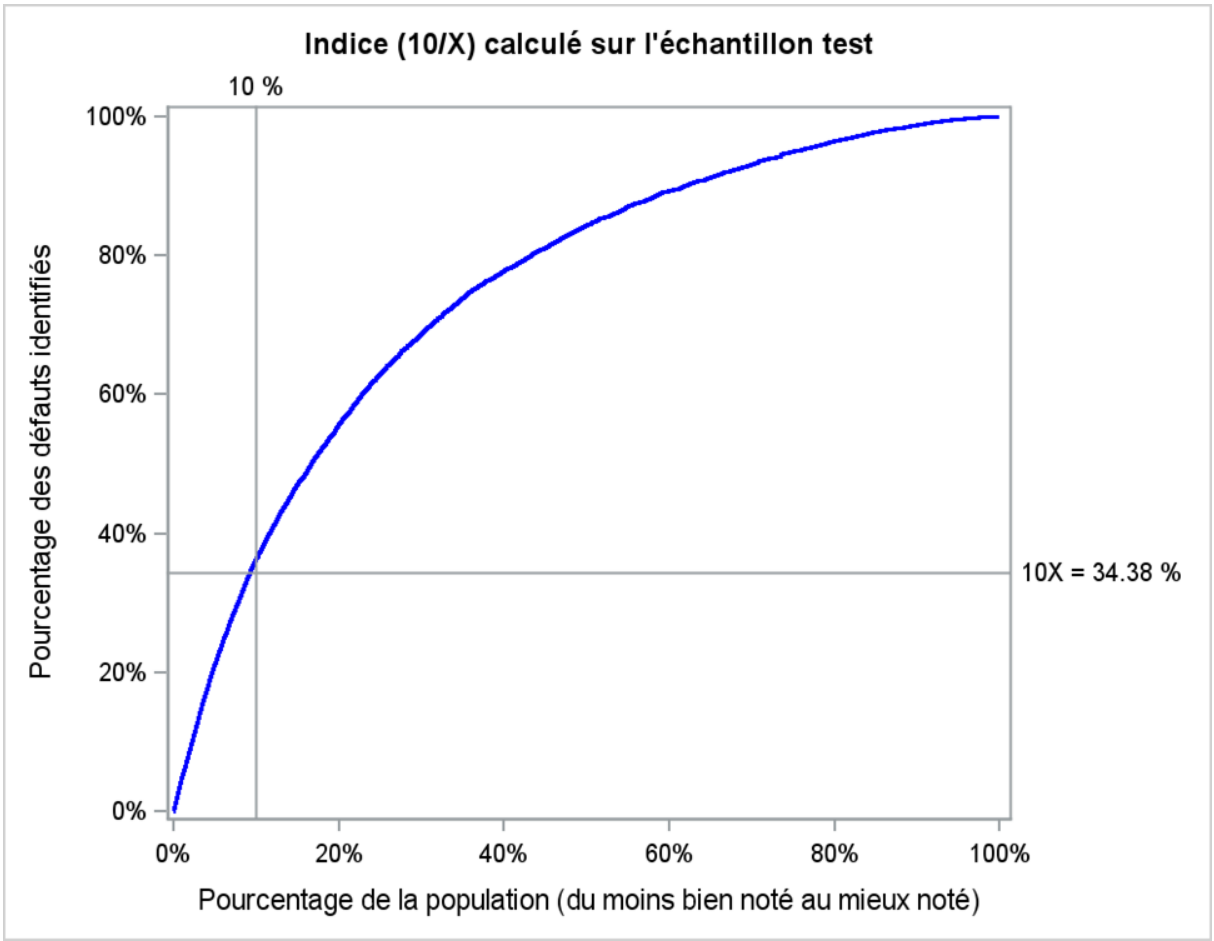
En revanche du côté du 10/X, nous perdons près de 1 point de pourcentage par rapport à la régression classique.

SMOTE 10%

On a ici une légère amélioration de l'AUC passant de 0.766 à 0.78 pour l'échantillon test.

Nous apercevons néanmoins une baisse de l'indice 10/X





Nous constatons que l'AUC, l'indice de Gini et le 10/X sont pratiquement identiques à ceux obtenus par la régression logistique sans l'utilisation du SMOTE. Cela suggère que le recours au SMOTE n'entraîne pas d'amélioration significative du modèle de prédiction de la probabilité de défaut d'un individu. En d'autres termes, il semble que le modèle ait déjà atteint son niveau optimal d'apprentissage en utilisant les cas disponibles, et l'interpolation de ces cas supplémentaires grâce au SMOTE ne contribue pas à enrichir les informations obtenues.

C) Conclusion

Le SMOTE est une technique précieuse pour rééquilibrer les données en entrée d'un modèle de Machine Learning. Elle prévient le surapprentissage en enrichissant uniformément les données minoritaires, et ses effets sur les performances du modèle peuvent varier. Comme mentionné précédemment, l'utilisation du SMOTE n'a pas eu d'impact positif sur les performances de notre modèle.

4. Méthode 3 : Random Forest

A) Introduction au Random Forest

Le Random Forest, un algorithme puissant de Machine Mearning, s'avère être un outil de prédiction précieux. Il doit son nom à sa structure complexe qui consiste en une "forêt" d'arbres de décision. Contrairement à un unique arbre de décision, le Random Forest combine les prédictions de plusieurs arbres, d'où le terme "Forest," pour améliorer la précision et la robustesse de la prédiction.

Le fonctionnement du Random Forest repose sur le principe de l'agrégation d'arbres de décision. Chaque arbre individuel est construit en utilisant un sous-ensemble aléatoire des données d'apprentissage et en utilisant une partie des caractéristiques (variables) disponibles. Cela permet de réduire la variance et d'éviter au mieux le surajustement, car chaque arbre se spécialise dans une partie différente de l'espace des caractéristiques. Lors de la phase de prédiction, chaque arbre émet sa propre prédiction, et le résultat final est obtenu en agrégeant ces prédictions (généralement par un vote majoritaire pour la classification ou une moyenne pour la régression).

Dans le cadre de ce projet, nous avons utilisé la bibliothèque scikit-learn de Python, qui offre des outils puissants pour implémenter le Random Forest.

B) Modélisation

Les données utilisées et la modification des variables

Afin de comparer nos résultats, nous avons importé l'échantillon d'apprentissage et de test similaire à celui utilisé pour la méthode de régression logistique.

Il était cependant nécessaire de procéder à des transformations sur les données. En effet, nos données comprenant des variables discrétisées et donc qualitatives, nous avons dû convertir les variables catégorielles en variables binaires (dummy variables) pour qu'elles puissent être prises en compte par le modèle. Cette étape est cruciale pour que le Random Forest puisse fonctionner, car il s'agit d'une méthode basée sur des arbres de décision qui nécessite des données numériques.

Le choix des hyperparamètres

Le choix des hyperparamètres est une étape fondamentale pour tirer le meilleur parti du Random Forest. Pour parvenir à cette optimisation, nous avons eu recours à la validation croisée (cross-validation). Cette technique implique la division des données d'apprentissage en plusieurs ensembles plus petits, connus sous le nom de "folds," puis l'entraînement du modèle sur plusieurs combinaisons de ces plis, tout en évaluant la performance sur un pli de validation distinct. Cela nous permet de déterminer la combinaison d'hyperparamètres offrant la meilleure performance globale.

Les hyperparamètres considérés dans notre étude étaient le nombre d'arbres dans la forêt (70, 90, 110 ou 130) et la profondeur maximale des arbres (libre, 8, 10, 12, 14). Le choix de ces hyperparamètres est crucial car ils influencent directement la complexité du modèle et son aptitude à s'adapter aux données. Un nombre d'arbres plus élevé augmente la capacité du modèle à capturer les nuances des données, mais peut entraîner un surajustement. De même, la profondeur des arbres détermine le niveau de détail des décisions prises par chaque arbre, mais une profondeur excessive peut conduire à un surajustement. Il est donc impératif de trouver un équilibre entre ces deux hyperparamètres pour optimiser la performance du modèle tout en évitant l'overfitting.

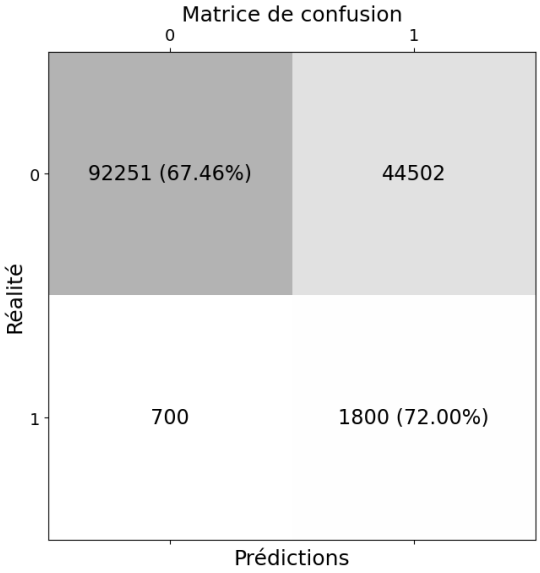
La création du modèle

Une fois la grille des hyperparamètres définie, nous avons entraîné plusieurs modèles Random Forest en utilisant différentes combinaisons de ces paramètres. Le modèle optimal a été obtenu en sélectionnant la configuration qui a produit la meilleure AUC moyenne[7]. Dans notre cas, le modèle optimal a été obtenu avec un nombre d'arbres égal à 90 et une profondeur maximale de 10. Ce modèle a ensuite été utilisé pour effectuer des prédictions sur l'échantillon de test, générant des probabilités prédictives.

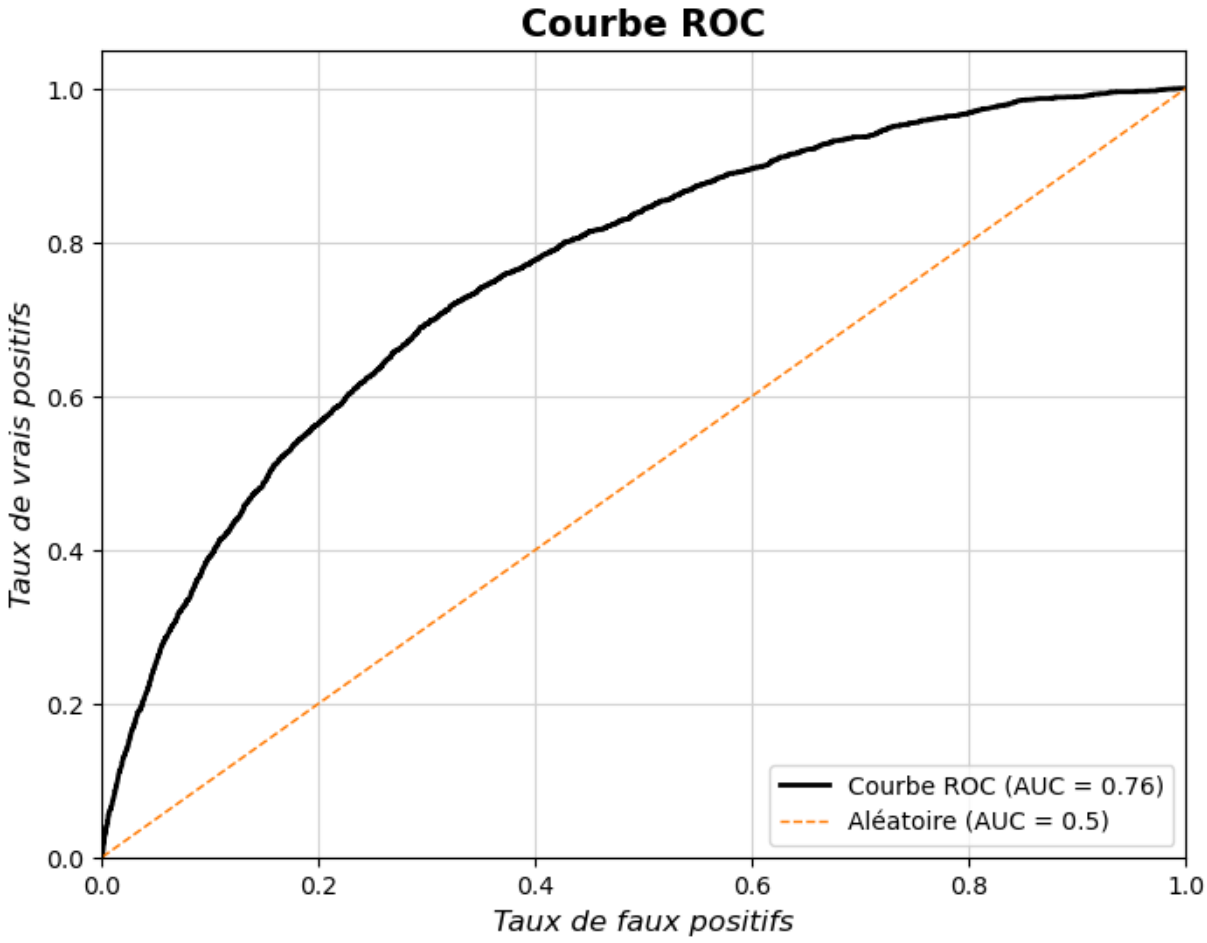
[7] Cf annexe (Table 7)

C) Performances

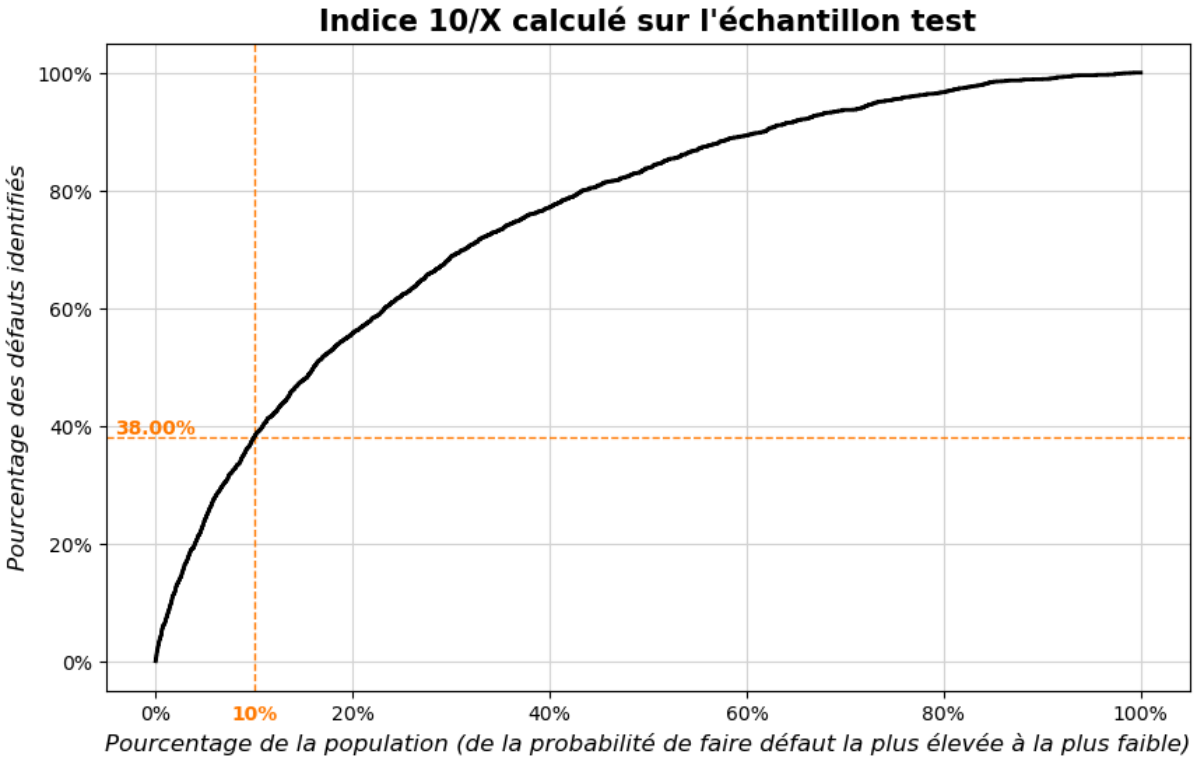
Pour évaluer les performances du modèle, nous avons utilisé plusieurs mesures. Tout d'abord, nous avons construit une matrice de confusion pour évaluer la qualité des prédictions. Le seuil de coupure (cut-off) choisi pour la classification a été déterminé en reprenant la valeur estimée dans la méthode précédente, garantissant ainsi une comparaison équitable des résultats. Il a donc été fixé à 0.188 et les résultats de la matrice de confusion ont révélé un taux de vrais négatifs de 67.46% et un taux de vrais positifs de 72%.



En outre, nous avons évalué la performance du modèle en utilisant la courbe ROC et le coefficient Gini. La courbe ROC nous a fourni un AUC de 0.76, tandis que le coefficient Gini a atteint 0,52.



A l'instar de la régression logistique, nous avons également estimé l'indice 10/X :



Ces mesures démontrent la capacité du modèle Random Forest à discriminer de manière efficace entre les clients à risque et les clients sains, confirmant ainsi son utilité dans notre contexte.

5. Méthode 4 : XGBoost

A) Introduction à XGBoost

XGBoost, ou "eXtreme Gradient Boosting," est une méthode de Machine Learning puissante qui s'est imposée comme un outil incontournable dans la modélisation du risque de crédit. C'est une méthode d'ensemble qui repose sur le Boosting, la régularisation et la séquentialité et se distingue par sa capacité à produire des modèles de prédiction précis et robustes, tout en minimisant le risque de surajustement. Pour comprendre en détail comment fonctionne XGBoost, il est essentiel de se plonger dans les mécanismes internes de cette méthode.

Au cœur de XGBoost se trouve le concept du Boosting, une technique d'ensemble qui combine plusieurs modèles de base pour créer un modèle global plus puissant. Dans le cas de XGBoost, ces modèles de base sont généralement des arbres de décision faibles. Le processus de construction du modèle XGBoost se déroule en plusieurs étapes clés :

- Boosting séquentiel : XGBoost construit des arbres de décision de manière séquentielle, un par un. Chaque nouvel arbre est conçu pour corriger les erreurs de prédiction faites par les arbres précédents. Cela signifie que les exemples mal classés par les arbres précédents reçoivent plus d'attention dans la construction du nouvel arbre, permettant ainsi d'améliorer progressivement les performances du modèle.
- Gradient Boosting : La technique de "gradient boosting" est utilisée pour minimiser une fonction de coût. Cette fonction mesure à quel point les prédictions du modèle s'éloignent des valeurs réelles. XGBoost calcule les gradients de cette fonction de coût par rapport aux prédictions actuelles du modèle, puis ajuste les prédictions en suivant ces gradients pour minimiser la perte.
- Régularisation : Pour prévenir le surajustement, XGBoost intègre des techniques de régularisation. Deux types de régularisation, L1 (Lasso) et L2 (Ridge), sont couramment utilisés pour pénaliser les modèles trop complexes en réduisant les poids des caractéristiques moins importantes.
- Sélection des caractéristiques : XGBoost effectue automatiquement la sélection des caractéristiques en évaluant l'importance de chaque caractéristique pour la tâche de prédiction. Les caractéristiques les plus importantes sont privilégiées dans la construction des arbres, ce qui contribue à la robustesse du modèle.

- Combinaison des arbres : Une fois que tous les arbres faibles sont construits, XGBoost les combine pour former un modèle global robuste. Les prédictions finales sont obtenues en agrégeant les prédictions de tous les arbres.

Pour exploiter pleinement la puissance de XGBoost, il est essentiel de prendre en compte les hyperparamètres, tels que le taux d'apprentissage, le nombre d'arbres ou la profondeur maximale des arbres. L'optimisation de ces hyperparamètres est cruciale pour obtenir les meilleures performances du modèle, tout en évitant les pièges du surajustement.

Nous avons utilisé la bibliothèque XGBoost sur Python afin de réaliser notre modèle de Machine Learning.

B) Modélisation

Les données utilisées et les changements nécessaires

De la même manière que notre autre méthode de Machine Learning, nous avons utilisés le même échantillon d'apprentissage et test que ceux de la régression logistique. Avant de pouvoir modéliser avec XGBoost, il était encore une fois impératif de convertir ces variables catégorielles en variables binaires. Cette étape est nécessaire pour que le modèle puisse traiter ces variables catégorielles de manière appropriée, car XGBoost ne gère également que des données numériques.

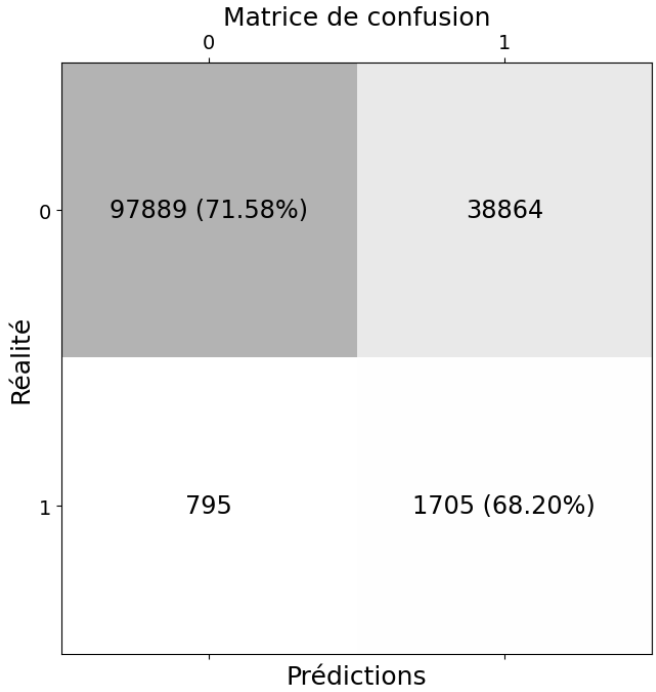
Le choix des hyperparamètres

Le choix des hyperparamètres est crucial pour optimiser les performances du modèle XGBoost. Pour déterminer les meilleurs hyperparamètres, nous avons opté pour une approche de validation croisée avec 10 folds. Nous avons défini une grille de paramètres contenant différentes combinaisons possibles, notamment le nombre d'arbres (70, 90, 110, ou 130), la profondeur (libre, 8, 10, 12, 14), et le taux d'apprentissage (0.1, 0.01). Cette grille a été conçue pour rester raisonnable en termes de temps de calcul, tout en explorant un espace d'hyperparamètres significatif. Le choix de ces hyperparamètres a été effectué avec soin, car ils influencent directement la capacité du modèle à généraliser et à éviter le surajustement.

La création du modèle

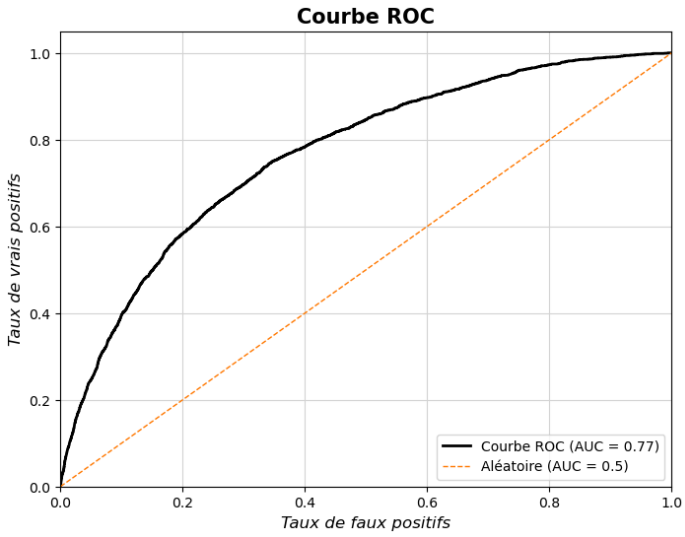
Une fois les hyperparamètres déterminés, nous avons construit le modèle XGBoost en utilisant la combinaison qui a donné la meilleure moyenne de l'aire sous la courbe ROC lors de la validation croisée[8]. Dans notre cas, le modèle final avait 90 arbres et une profondeur de 10. Ce modèle a été utilisé pour effectuer des prédictions sur l'échantillon de test, fournissant ainsi les probabilités prédites de défaut de crédit.

C) Performances



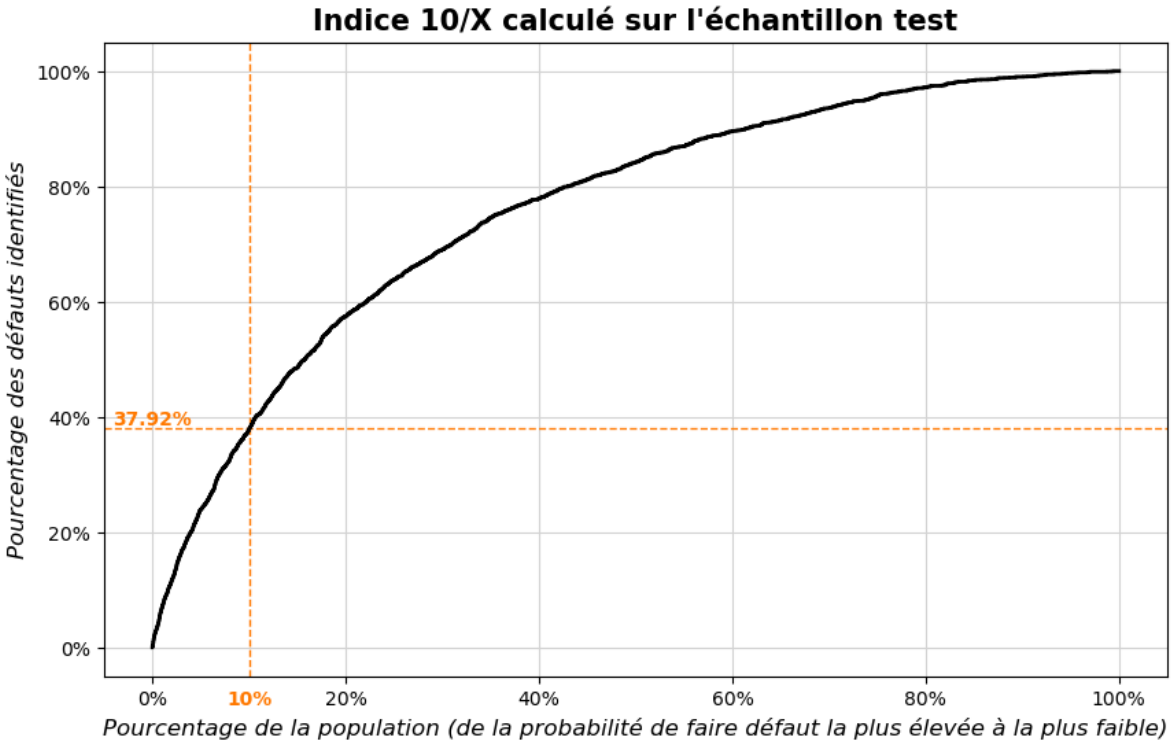
Pour évaluer les performances de notre modèle, nous avons suivi les mêmes approches. Tout d'abord, nous avons créé une matrice de confusion pour évaluer le pouvoir discriminant du modèle. Pour maintenir la cohérence, nous avons utilisé le même seuil que celui estimé lors de nos méthodes précédentes, ce qui nous a permis de comparer les résultats de manière équitable. Avec ce cut-off, nous avons obtenu 71.58% de vrais négatifs et 68.20% de vrais positifs.

Nous avons ensuite utilisé la courbe ROC pour évaluer la capacité du modèle à discriminer entre les classes. Dans notre cas, le modèle XGBoost a obtenu un AUC de 0.77. Le coefficient de Gini, calculé à partir de l'AUC, s'est lui élevé à 0.54, confirmant ainsi la capacité du modèle à bien classer les observations.



[8] Cf annexe (Table 8)

Enfin, voici une nouvelle fois l'indice 10/X estimé :



En conclusion, XGBoost s'est avéré être une méthode de machine learning puissante. Grâce à une sélection minutieuse, nous avons obtenu un modèle performant, avec une bonne capacité de discrimination. Nos résultats renforcent la pertinence de l'utilisation de XGBoost.

III. Conclusion

1) Comparaison entre les modèles

	Régression logistique	Machine Learning
Avantages	Interprétabilité : La régression logistique fournit des coefficients pour chaque variable, ce qui permet de comprendre l'impact de chaque variable sur la décision de crédit. Cela peut être important pour des raisons de conformité réglementaire. Moins de données requises : La régression logistique peut fonctionner efficacement avec des ensembles de données relativement petits par rapport à de nombreuses techniques de Machine Learning. Temps de calcul plus courts : La régression logistique est généralement plus rapide à entraîner et à déployer que de nombreuses méthodes de Machine Learning.	Capacité à modéliser des relations complexes : Les méthodes de Machine Learning, comme Random Forest et XGBoost, peuvent capturer des relations non linéaires complexes entre les variables, ce qui est souvent le cas dans le crédit scoring. Haute précision : Ces méthodes ont généralement une meilleure précision de prédiction par rapport à la régression logistique, en particulier lorsque le modèle est correctement paramétré. Capacité à gérer de grandes quantités de données : Les méthodes de Machine Learning sont souvent plus efficaces pour traiter de grandes quantités de données.
Inconvénients	Linéarité : La régression logistique suppose une relation linéaire entre les variables indépendantes et la variable dépendante. Si la relation est complexe, la régression logistique peut ne pas être en mesure de la modéliser efficacement.	Moins interprétables : Les modèles de Machine Learning, en particulier Random Forest et XGBoost, sont moins interprétables que la régression logistique. Ils ne fournissent pas de coefficients de variable directement interprétables.

	Moins adaptée à de grandes quantités de données : Pour de très grandes quantités de données avec des relations non linéaires complexes, la régression logistique peut être moins précise que les méthodes de Machine Learning.	Plus de temps de calcul et de ressources : L'entraînement de modèles de Machine Learning complexes peut prendre plus de temps et nécessiter plus de ressources informatiques que la régression logistique. Risque de surajustement : Les méthodes de Machine Learning peuvent être plus sensibles au surajustement, ce qui signifie qu'elles peuvent s'adapter trop étroitement aux données d'entraînement et avoir une moins bonne généralisation sur de nouvelles données.

2) Conclusion Générale / Ouverture

La construction d'un modèle de scoring est en effet un processus complexe et exigeant, nécessitant une approche rigoureuse et critique. Il est essentiel de maintenir une perspective globale sur les données que nous utilisons, en vérifiant constamment qu'elles soient cohérentes avec la réalité et conformes aux aspects réglementaires. De nombreux éléments, notamment ceux liés au contexte socio-économique et au comportement individuel, ne peuvent être pleinement compris qu'en prenant en compte les connaissances humaines.

En plus des données observables classiques, il est important de reconnaître l'existence de nombreuses variables inobservables qui pourraient améliorer la précision de notre modèle de scoring. Ces données sont souvent difficiles à quantifier car elles ne sont pas systématiquement enregistrées dans les rapports de crédit ou autres documents financiers. Par exemple, la fréquence des impayés pour le loyer ou les paiements réguliers, ainsi que la stabilité de l'emploi, peuvent offrir des informations cruciales sur la capacité de l'emprunteur à honorer ses engagements financiers sur le long terme.

Le comportement de consommation est un autre facteur à considérer. Les banques, grâce à l'analyse de données massives (big data), ont accès à une quantité considérable d'informations sur les habitudes de consommation des

individus. Par exemple, un client qui dépense la totalité de son salaire en début de mois peut présenter un risque de crédit plus élevé, car cela peut indiquer une gestion budgétaire moins prudente.

En outre, certaines entreprises de technologie financière (FinTech) explorent même l'utilisation de données provenant des médias sociaux pour évaluer le comportement financier des individus, telles que leurs fréquences de voyages ou leurs activités de loisirs. Cependant, il est important de noter que l'utilisation de telles données soulève des questions de protection de la vie privée et doit être conforme aux réglementations, notamment le Règlement Général sur la Protection des Données (RGPD) en Europe.

Un aspect essentiel dans la construction de modèles de crédit scoring est de garantir le "fairness". Cela signifie que les modèles ne doivent pas créer de discrimination indirecte envers certains groupes de la population. Cela peut survenir lorsque des facteurs apparemment non discriminatoires ont un impact disproportionné sur des groupes spécifiques, comme les minorités ethniques ou les femmes. Les réglementations et les directives, y compris le RGPD, imposent des exigences strictes pour minimiser ces biais et garantir l'équité dans les décisions de crédit.

De nombreux travaux sont actuellement en cours sur ce sujet, notamment Monsieur SAURIN qui poursuit actuellement une thèse sur le thème de l'équité algorithmique en finance.

Enfin face aux enjeux climatiques actuels, les institutions essayent de plus en plus d'intégrer les risques ESG à leurs modèles. En effet l'intégration de variables ESG dans un modèle de scoring d'octroi de prêt automobile chez un particulier pourrait à l'avenir être admise. La consommation de carburant est un facteur environnemental important, car un véhicule économe en carburant réduit les coûts, impactant positivement la capacité de remboursement. D'autre part selon le type de véhicule, on peut avoir une exposition plus ou moins forte aux nouvelles législations. Imaginons qu'un client achète un véhicule très polluant et que le gouvernement interdise sa circulation. Dans le cas où le client se retrouverai temporairement sans revenu, il verra sa probabilité de faire défaut augmenter dans la mesure où il a peu de chance de pouvoir revendre le véhicule pour rembourser son crédit.

Ainsi, il est indéniable que les banques doivent intensifier leurs efforts pour obtenir davantage de données en vue d'améliorer les modèles de scoring du futur.

IV. BIBLIOGRAPHIE

Gourieroux, C. (1984). *Econométrie des variables qualitatives*.

Annales d'économie et de statistique. (1992b).

Gouriéroux, C., & Jasiak, J. (2007). *The Econometrics of Individual Risk : credit, insurance, and marketing*. http://ci.nii.ac.jp/ncid/BA81309778

Vannieuwenhuyze, A. (2019). Intelligence artificielle vulgarisée : Le machine learning et le deep learning par la pratique.

Admin_pix. (2023, 2 mars). *Déclarer la guerre aux données déséquilibrées : SMOTE*. Néosoft. https://www.neosoft.fr/nos-publications/blog-tech/techniques-augmentation-dataset-smote/

Benzaki, Y. (2018, 8 février). *Comment traiter les données manquantes en data science*. Mr. Mint : Apprendre le Machine Learning de A à Z. https://mrmint.fr/donnees-manquantes-data-science

Team, D. (2023, 12 octobre). *Algorithmes de boosting – AdaBoost, Gradient Boosting, XGBoost*. Formation Data Science | DataScientest.com. https://datascientest.com/algorithmes-de-boosting-adaboost-gradient-boosting-xgboost

V. ANNEXE

Table 1

Variable	Minimum	Maximum	Médiane	Quartile inférieur	Quartile supérieur	Ec-type
mt_finance	1998.98	235979.12	14884.14	11010.76	19800.00	6935.06
appo_cptt_cnt	0.00	97.62	7.24	0.00	25.00	22.73
nb_imp_an_0	0.00	26.00	0.00	0.00	0.00	0.56
nb_imp_tot	0.00	166.00	0.00	0.00	0.00	2.00
age_indv	17.00	98.00	54.00	42.00	66.00	15.52
anc_emp_indv	0.00	817.00	21.00	0.00	155.00	126.07
REV_TOT	0.00	760613.00	2880.00	2000.00	3923.00	2973.87
mt_charges	0.00	11862.00	0.00	0.00	480.00	400.66
part_ech	0.01	75.75	1.44	1.21	1.62	0.90
tx_end_syex	0.04	999.99	15.87	8.80	26.57	14.66
mt_ttc_veh	3000.00	255970.41	18181.76	13972.76	22695.76	6511.23
anc_adr_indv	0.00	1100.00	92.00	2.00	232.00	156.45
mt_alloc_pond	0.00	74611.20	0.00	0.00	0.00	233.32
NB_PERS_CHG	0.00	63.00	0.00	0.00	1.00	0.95
MT_SAL_MEN	0.00	760613.00	1400.00	0.00	2100.00	2391.18
MT_ALLOC_MEN	0.00	124352.00	0.00	0.00	0.00	307.46
REV_MEN_AUTR	0.00	265806.00	0.00	0.00	2000.00	1787.76
MS_CNT	13.00	72.00	49.00	49.00	60.00	9.86
MT_LOY_MEN_MENA	0.00	6000.00	0.00	0.00	0.00	229.58
MT_MEN_PRE_IMMO	0.00	11200.00	0.00	0.00	0.00	344.65
MT_MEN_ENG_MENA	0.00	8500.00	0.00	0.00	0.00	91.20
MT_ECH	0.90	11341.37	249.65	185.77	322.43	180.53
part_finance_rev	0.00	5596.12	18.89	13.21	28.98	29.09
ecart_dmd_gest	-1461.00	853.00	31.00	30.00	61.00	50.63

Table 2

REGION_				
region_	Fréquence	Pourcentage	Fréquence cumulée	Pourcentage cumulé
ARA	51117	11,01	51117	11,01
AUTR	50	0,01	51167	11,02
BFC	19710	4,25	70877	15,27
BRET	19406	4,18	90283	19,45
CORS	2353	0,51	92636	19,96
CVdL	20799	4,48	113435	24,44
HdF	57412	12,37	170847	36,81

IDF	58382	12,58	229229	49,38
NORM	28717	6,19	257946	55,57
NVA	49666	10,70	307612	66,27
OCC	43795	9,43	351407	75,71
PACA	46360	9,99	397767	85,69
PLOI	25336	5,46	423103	91,15
RGE	41076	8,85	464179	100,00
CD_NATL_INDV				
cd_natl_indv	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
C	4058	0,87	4058	0,87
F	458234	98,72	462292	99,60
H	1878	0,40	464170	100,00
Fréquence manquante = 9				
MOD_HABI_INDV				
mod_habi_indv	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
F	5022	1,08	5022	1,08
H	32848	7,08	37870	8,16
L	99417	21,42	137287	29,58
P	326856	70,42	464143	100,00
Fréquence manquante = 36				
ETA_CIV_PRTC				
eta_civ_prtc	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
C	112479	24,23	112479	24,23
D	32391	6,98	144870	31,21
M	237712	51,21	382582	82,42
S	12010	2,59	394592	85,01
U	45425	9,79	440017	94,80
V	24156	5,20	464173	100,00
Fréquence manquante = 6				
CSP_PERPHY				
csp_perphy	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
10	10	0,00	10	0,00
14	75	0,02	85	0,02
20	11295	2,43	11380	2,45
21	1949	0,42	13329	2,87
22	4079	0,88	17408	3,75
23	648	0,14	18056	3,89
30	40837	8,80	58893	12,69
31	4554	0,98	63447	13,67
32	6116	1,32	69563	14,99
33	11243	2,42	80806	17,41
34	402	0,09	81208	17,50
35	1675	0,36	82883	17,86

40	175754	37,86	258637	55,72
41	5260	1,13	263897	56,85
42	6001	1,29	269898	58,15
50	10629	2,29	280527	60,44
51	4566	0,98	285093	61,42
52	5	0,00	285098	61,42
53	32	0,01	285130	61,43
54	509	0,11	285639	61,54
55	32	0,01	285671	61,54
56	126	0,03	285797	61,57
57	199	0,04	285996	61,61
60	1144	0,25	287140	61,86
61	6128	1,32	293268	63,18
62	162173	34,94	455441	98,12
63	210	0,05	455651	98,16
64	2279	0,49	457930	98,66
65	3087	0,67	461017	99,32
66	34	0,01	461051	99,33
67	1656	0,36	462707	99,68
69	29	0,01	462736	99,69
70	1390	0,30	464126	99,99
77	47	0,01	464173	100,00
Fréquence manquante = 6				
SECTEUR_				
secteur_	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
AGR	3701	0,80	3701	0,80
ATR	305350	65,78	309051	66,58
BTP	12786	2,75	321837	69,33
CDD	9494	2,05	331331	71,38
CDG	6820	1,47	338151	72,85
EAE	10319	2,22	348470	75,07
FBC	6407	1,38	354877	76,45
FCP	14484	3,12	369361	79,57
FEM	5404	1,16	374765	80,74
HOP	19776	4,26	394541	85,00
LOA	433	0,09	394974	85,09
SCE	57685	12,43	452659	97,52
TRA	11520	2,48	464179	100,00
CPT_PAI2_				
cpt_pai2_	**Fréquence**	**Pourcentage**	**Fréquence cumulée**	**Pourcentage cumulé**
1	324516	69,91	324516	69,91
2	122260	26,34	446776	96,25
3	14036	3,02	460812	99,27
4	3367	0,73	464179	100,00
DIAG_CLI_RNVA				

diag_cli_rnva	Fréquence	Pourcentage	Fréquence cumulée	Pourcentage cumulé
B	126054	27,16	126054	27,16
I	324514	69,91	450568	97,07
M	13431	2,89	463999	99,96
R	180	0,04	464179	100,00
NO_NAT_PROD				
no_nat_prod	Fréquence	Pourcentage	Fréquence cumulée	Pourcentage cumulé
0	126234	27,20	126234	27,20
1	334558	72,08	460792	99,27
2	3387	0,73	464179	100,00

Table 3

Obs.	observations_total	observations	proportion_missing	name
1	464179	464179	,00000	No_cnt_crypte
2	464179	464179	,00000	No_par_crypte
3	464179	464179	,00000	date_gest
4	464179	464179	,00000	date_dmd
5	464179	464179	,00000	mt_finance
6	464179	464179	,00000	ms_cnt
7	464179	464179	,00000	appo_cptt_cnt
8	464179	464179	,00000	no_nat_prod
9	464179	464179	,00000	diag_cli_rnva
10	464179	139663	,69912	nb_imp_tot
11	464179	139663	,69912	nb_imp_an_0
12	464179	464179	,00000	cpt_pai2_
13	464179	464179	,00000	secteur_
14	464179	464179	,00000	age_indv
15	464179	464173	,00001	csp_perphy
16	464179	464173	,00001	eta_civ_prtc
17	464179	464143	,00008	mod_habi_indv
18	464179	464179	,00000	mt_sal_men
19	464179	464179	,00000	rev_men_autr
20	464179	464179	,00000	mt_alloc_men
21	464179	464179	,00000	nb_pers_chg
22	464179	464179	,00000	REV_TOT
23	464179	464179	,00000	mt_loy_men_mena
24	464179	464179	,00000	mt_men_pre_immo
25	464179	464179	,00000	mt_men_eng_mena
26	464179	464179	,00000	mt_charges
27	464179	464179	,00000	MT_ECH
28	464179	464179	,00000	mt_ttc_veh
29	464179	464179	,00000	part_ech
30	464179	464179	,00000	anc_emp_indv
31	464179	464179	,00000	tx_end_syex

32	464179	464179	,00000	anc_adr_indv
33	464179	464170	,00002	cd_natl_indv
34	464179	464179	,00000	DEPMT_HABI_INDV
35	464179	464179	,00000	mt_alloc_pond
36	464179	464179	,00000	region_

Table 4

Obs.	variable	part_outlier_var
1	ms_cnt	,04283
2	mt_fin	,01591
3	appo_c	,07517
4	nb_imp	,73661
5	nb_imp	,71430
6	age_in	,03572
7	nb_per	,04758
8	REV_TO	,03917
9	mt_loy	,21448
10	mt_men	,19876
11	mt_men	,05530
12	mt_cha	,02678
13	MT_ECH	,02312
14	mt_ttc	,01573
15	part_e	,04428
16	anc_em	,04319
17	tx_end	,00234
18	anc_ad	,00960
19	ecart_	,07011
20	mt_sal	,01978
21	rev_me	,02474
22	mt_all	,07618
23	mt_all	,07618
24	part_f	,07665

Table 5

Obs.	eta_civ_prtc	EFF_TOT	EFF_DEF	TD_EFF	PART_EFF
1	M	237718	2307	0,97%	,51213
2	V	24156	422	1,75%	,05204
3	U	45425	953	2,10%	,09786
4	D	32391	751	2,32%	,06978
5	C	112479	3504	3,12%	,24232
6	S	12010	397	3,31%	,02587

Table 6

Obs.	Variable	Cramers_V	Chisq	p_value
1	mod_habi_indv	0,0836	3244,3721	<.0001
2	cpt_pai2_2	0,0765	2716,9011	<.0001
3	nb_imp_an_0_2	0,0734	2504,0203	<.0001
4	age_indv2	0,0717	2388,9723	<.0001
5	eta_civ_prtc2	0,0690	2209,8516	<.0001
6	appo_cptt_cnt2	0,0644	1927,3936	<.0001
7	nb_imp_tot2	0,0623	1804,2429	<.0001
8	tx_end_syex2	0,0601	1677,7199	<.0001
9	part_ech2	0,0542	1361,3982	<.0001
10	CSP_classe	0,0483	1081,9221	<.0001
11	anc_adr_indv2	-0,0453	954,5919	<.0001
12	diag_cli_rnva	0,0429	853,1807	<.0001
13	anc_emp_indv2	0,0355	586,0988	<.0001
14	rev_men_autr2	0,0344	549,1923	<.0001
15	mt_ttc_veh2	0,0283	387,2584	<.0001
16	no_nat_prod2	0,0263	321,1919	<.0001
17	mt_charges2	-0,0263	320,4587	<.0001
18	secteur_2	0,0223	230,2806	<.0001
19	cd_natl_indv2	0,0169	133,2155	<.0001
20	region2	0,0135	84,7729	<.0001
21	MT_ECH2	-0,0124	71,3327	<.0001
22	REV_TOT2	0,0109	55,0079	<.0001
23	mt_finance2	0,0002	0,0183	0,8924

Table 7

AUC Moyenne	Paramètres
0,630629	{'max_depth': None, 'n_estimators': 70}
0,632172	{'max_depth': None, 'n_estimators': 90}
0,634357	{'max_depth': None, 'n_estimators': 110}
0,635175	{'max_depth': None, 'n_estimators': 120}
0,754235	{'max_depth': 8, 'n_estimators': 70}
0,754356	{'max_depth': 8, 'n_estimators': 90}
0,754387	{'max_depth': 8, 'n_estimators': 110}
0,754492	{'max_depth': 8, 'n_estimators': 120}

0,757302	{'max_depth': 10, 'n_estimators': 70}
0,757652	{'max_depth': 10, 'n_estimators': 90}
0,757533	{'max_depth': 10, 'n_estimators': 110}
0,75743	{'max_depth': 10, 'n_estimators': 120}
0,756071	{'max_depth': 12, 'n_estimators': 70}
0,756887	{'max_depth': 12, 'n_estimators': 90}
0,757133	{'max_depth': 12, 'n_estimators': 110}
0,757131	{'max_depth': 12, 'n_estimators': 120}
0,751852	{'max_depth': 14, 'n_estimators': 70}
0,752663	{'max_depth': 14, 'n_estimators': 90}
0,753282	{'max_depth': 14, 'n_estimators': 110}
0,753549	{'max_depth': 14, 'n_estimators': 120}

Table 8

AUC Moyenne	Paramètres
0,738588	{'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 90}
0,738957	{'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 100}
0,739648	{'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 110}
0,740265	{'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 120}
0,740748	{'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 130}
0,720159	{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 90}
0,721127	{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 100}
0,722308	{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 110}
0,723217	{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 120}
0,724843	{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 130}
0,738588	{'learning_rate': 0.01, 'max_depth': 6, 'n_estimators': 90}
0,738957	{'learning_rate': 0.01, 'max_depth': 6, 'n_estimators': 100}
0,739648	{'learning_rate': 0.01, 'max_depth': 6, 'n_estimators': 110}
0,740265	{'learning_rate': 0.01, 'max_depth': 6, 'n_estimators': 120}
0,740748	{'learning_rate': 0.01, 'max_depth': 6, 'n_estimators': 130}
0,746003	{'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 90}
0,746289	{'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 100}
0,746642	{'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 110}
0,746988	{'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 120}
0,747438	{'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 130}
0,747287	{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 90}

0,747759	{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 100}
0,748477	{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 110}
0,74891	{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 120}
0,749245	{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 130}
0,745509	{'learning_rate': 0.01, 'max_depth': 12, 'n_estimators': 90}
0,745449	{'learning_rate': 0.01, 'max_depth': 12, 'n_estimators': 100}
0,745866	{'learning_rate': 0.01, 'max_depth': 12, 'n_estimators': 110}
0,745899	{'learning_rate': 0.01, 'max_depth': 12, 'n_estimators': 120}
0,74604	{'learning_rate': 0.01, 'max_depth': 12, 'n_estimators': 130}
0,765223	{'learning_rate': 0.1, 'max_depth': None, 'n_estimators': 90}
0,765068	{'learning_rate': 0.1, 'max_depth': None, 'n_estimators': 100}
0,764772	{'learning_rate': 0.1, 'max_depth': None, 'n_estimators': 110}
0,76441	{'learning_rate': 0.1, 'max_depth': None, 'n_estimators': 120}
0,764114	{'learning_rate': 0.1, 'max_depth': None, 'n_estimators': 130}
0,763916	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 90}
0,764547	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}
0,764925	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 110}
0,765111	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 120}
0,765396	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 130}
0,765223	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 90}
0,765068	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 100}
0,764772	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 110}
0,76441	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 120}
0,764114	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 130}
0,759557	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 90}
0,758457	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 100}
0,757961	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 110}
0,756756	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 120}
0,755693	{'learning_rate': 0.1, 'max_depth': 8, 'n_estimators': 130}
0,752732	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 90}
0,750614	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 100}
0,748553	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 110}
0,746907	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 120}
0,745437	{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 130}
0,742139	{'learning_rate': 0.1, 'max_depth': 12, 'n_estimators': 90}
0,740181	{'learning_rate': 0.1, 'max_depth': 12, 'n_estimators': 100}
0,737441	{'learning_rate': 0.1, 'max_depth': 12, 'n_estimators': 110}
0,735005	{'learning_rate': 0.1, 'max_depth': 12, 'n_estimators': 120}
0,732632	{'learning_rate': 0.1, 'max_depth': 12, 'n_estimators': 130}