Modélisation du comportement de conformité fiscale des individus avec un modèle d'apprentissage machine.

Par

Jotio Joseph Antoine Bell

Mémoire de fin d'études

HEC Montréal, département d'ingénierie financière

HEC MONTRÉAL

Directeur de recherche

Vincent Grégroire, professeur agrégé

Département de finance

HEC Montréal

Canada

Octobre 2024

Table des matières

Résumé

Dans le domaine du recouvrement fiscal, la connaissance client revêt une importance cruciale pour adapter les stratégies et les actions aux besoins spécifiques des contribuables. Notre mémoire s'inscrit dans le cadre du projet de connaissance client de Revenu Québec, visant à améliorer l'expérience des contribuables en personnalisant les services et en optimisant les processus de recouvrement.

Notre méthodologie rigoureuse a consisté en plusieurs étapes essentielles, de la collecte et l'exploration des données à la construction et l'évaluation des modèles. Nous avons adopté une approche axée sur la qualité des données, en nettoyant et en préparant soigneusement notre jeu de données pour optimiser la fiabilité des résultats.

L'objectif général de notre projet était d'opérationnaliser les modèles prédictifs pour les intégrer dans la stratégie de modernisation du recouvrement de Revenu Québec. Nous avons travaillé sur le développement de modèles décisionnels permettant de classifier les clients et d'anticiper leur comportement de paiement.

Au-delà des défis techniques, notre contribution s'inscrit dans une perspective d'optimisation des processus de recouvrement et d'amélioration de l'expérience client. En exploitant les données et en tirant parti des avancées en matière d'intelligence artificielle, nous avons cherché à proposer des solutions innovantes pour répondre aux enjeux complexes du recouvrement fiscal.

Mots-clés : Apprentissage machine ; conformité fiscale ; forêt aléatoire ; connaissance client ; recherche sur grille ; validation croisée.

Abstract

In the field of tax collection, customer knowledge is of crucial importance in adapting strategies and actions to the specific needs of taxpayers. My dissertation is part of Revenu Québec's Know Your Customer project, which aims to improve the taxpayer experience by personalising services and optimising collection processes.

Our rigorous methodology consisted of several essential steps, from data collection and exploration to model building and evaluation. We adopted a data quality approach, carefully cleaning and preparing our dataset to maximise the reliability of the results.

The overall objective of our project was to operationalise the predictive models and integrate them into Revenu Québec's collection modernisation strategy. We worked on the development of decision-making models to classify customers and anticipate their payment behaviour.

Beyond the technical challenges, our contribution is aimed at optimising collection processes and improving the customer experience. By exploiting data and taking advantage of advances in artificial intelligence, we have sought to propose innovative solutions to meet the complex challenges of tax collection.

Keywords : Machine learning ; tax compliance ; random forest ; customer knowledge ; grid search ; cross-validation.

Remerciements

Je remercie mon directeur de recherche, le professeur Vincent Grégoire pour son aide immense et inestimable, les opportunités de travail et de bourses, les recommandations diverses, et son accompagnement académique et professionnel dans le cadre de mon parcours de maîtrise et de mon projet de fin d'étude. Je peux dire sans exagération que sans lui, je n'en serai pas là. Merci Monsieur.

Je remercie mes parents Tsobgny Théophile et Dongmo Julienne pour m'avoir donné la vie et pour l'éducation de qualité que j'ai reçue. Merci chers parents.

Je remercie ma conjointe Doume Ekale pour son amour inconditionnel et son soutien infaillible. Merci Quérida.

Je remercie ma famille de sang et de coeur, pour le soutien multiforme, c'est aussi grâce à vous si j'ai pu voir le bout du tunnel. Merci famille.

Liste des figures

Liste des tables

Introduction

0.1 Contexte et définition des besoins

Le projet de connaissance client (CC) de Revenu Québec (2023) a pour objectif de développer une meilleure compréhension de la clientèle afin d'améliorer son expérience en adaptant les pratiques aux besoins spécifiques de chaque client et en facilitant ainsi sa conformité fiscale. Ce projet vise à personnaliser davantage les services en exploitant les possibilités offertes par le numérique, à améliorer l'expérience client en utilisant les connaissances disponibles pour prendre des actions appropriées, et à accroître l'efficience des activités de recouvrement en concentrant les efforts sur les dossiers nécessitant un suivi plus rigoureux. Et ceci grâce à des modèles d'apprentissage machine appropriés, en mettant l'accent sur des données de haute qualité et l'évaluation continue desdits modèles comme le précise Olatunji Akinrinola et al. (2024), notamment en challengeant avec des nouveaux modèles pouvant mieux performer.

Notre travail consiste en la modélisation de la capacité et de la volonté du client afin de faire une prédiction du risque de défaut de paiement ou autrement dit, de la capacité à s'acquitter de sa créance, ce qui permettra d'adapter le parcours de son dossier dans les processus de recouvrement afin de prendre les bonnes actions au bon moment.

Au plan triennal 2018-2021 de Revenu Québec (2023), un enjeu de la qualité des services est d'améliorer la connaissance de la clientèle, en vue d'adapter les processus et de bonifier l'expérience client. Le plan d'action 2019-2020 de la direction générale du recouvrement (DGR) a porté sur la meilleure compréhension des facteurs de confiance et de conformité volontaire de la clientèle afin de mieux adapter les interventions. C'est

de là que résulte le projet de connaissance client, qui consiste à mesurer en continu le comportement du client, à l'apparier à l'information dont on dispose sur lui et enfin à utiliser cette information pour personnaliser le service et ainsi l'améliorer.

La vision de CC est d'identifier les bonnes actions et le bon moment selon le profil du client. L'approche est basée sur la volonté et la capacité à payer du débiteur afin de voir s'il a besoin d'être informé ou accompagné, s'il a besoin d'une assistance coercitive ou de plus d'autonomie pour se conformer au règlement de sa dette. Comme l'explique Nembe et al. (2024), dans le domaine de la réglementation financière, les technologies de l'intelligence artificielle jouent un rôle crucial dans le contrôle et l'application de cadres réglementaires complexes. La première étape consiste donc à construire un modèle prédictif pour les clients particuliers non-mandataires qui ont de la volonté et de la capacité à payer leurs dettes sous 12 mois, sans recours administratifs ou judiciaire.

Le projet vise à développer la connaissance de la clientèle afin d'améliorer son expérience en adaptant les stratégies et les façons de faire à ses besoins et ainsi mieux l'accompagner dans sa conformité fiscale.

0.2 Méthodologie

Dans le cadre de la réalisation du projet, nous adoptons une méthodologie rigoureuse pour la construction de nos modèles d'intelligence artificielle en mettant en œuvre plusieurs étapes essentielles. Tout d'abord, nous procédons à la collecte et l'exploration des données à partir de diverses sources, dans l'environnement SAS. Cette étape revêt une importance capitale car la qualité des données influence directement la performance du modèle. Ensuite, nous passons à l'étape de nettoyage des données, où nous traitons les valeurs manquantes, les données aberrantes et les erreurs, afin de garantir la fiabilité des données utilisées pour l'entraînement du modèle. La collecte, l'exploration et le nettoyage des données font partie de la grande étape de préparation des données.

Après la préparation des données, nous entamons l'étape d'entraînement du modèle choisi pour challenger la regression logistique, où nous construisons l'algorithme d'apprentissage et l'entraînons sur les données préalablement préparées. Nous explorons différentes tech-

niques d'apprentissage automatique, en mettant un accent particulier sur les modèles d'apprentissage supervisé tels que les KNN et les forêts aléatoires, dans le but d'améliorer les performances par rapport à une régression logistique déjà existante.

Une fois le modèle choisi entraîné, nous procédons à l'étape de test et de validation, où nous évaluons sa performance sur des données indépendantes. À cette fin, nous utilisons une variété de métriques pour mesurer la précision, la sensibilité et la spécificité du modèle. Cette évaluation nous permet de déterminer la capacité de généralisation du modèle et de conclure sur son acceptabilité ou non pour notre application spécifique dans le domaine fiscal.

0.3 Objectifs

Les objectifs spécifiques du projet de connaissance client de Revenu Québec (2023) sont de développer des modèles prédictifs afin d'adapter le parcours client en fonction du profil du débiteur, et d'opérationnaliser le recours aux modèles prédictifs en les rendant exploitables par la couche d'intégration de la nouvelle solution d'affaires de recouvrement.

0.4 Notre contribution

Pour la mise en oeuvre de la solution, nous devons construire un modèle décisionnel pour la classification des clients. C'est-à-dire que pour un client précis, nous voulons à partir de son profil, déterminer sa volonté et sa capacité à payer sa dette en 12 mois sans recours administratif et juridique. Il s'agit de comprendre les variables mises à notre disposition et les transformer adéquatement le cas échéant. Et de construire un modèle performant dans un délai limité qui pourra faire pour un dossier client donné la prédiction du risque de défaut de paiement ou autrement dit, de la volonté et de la capacité à s'acquitter de sa créance ce qui permettra d'adapter le parcours de son dossier dans les processus de recouvrement afin de prendre les bonnes actions au bon moment.

0.5 Enjeux et impact

Les principaux enjeux incluent la viabilité des modèles au fil du temps et la facilité de leur mise à jour dans un contexte en constante évolution, ainsi que l'optimisation des processus de recouvrement et l'utilisation plus efficiente des ressources allouées à cette tâche. Il est également crucial de développer une meilleure compréhension des comportements de la clientèle pour permettre un accompagnement personnalisé de chaque client en recouvrement, adapté à leur segment d'appartenance le plus probable.

Ce mémoire est divisé en 5 chapitres :

Chapitre 1, Etat de l'art et choix du modèle , ici nous présentons l'état de l'art, les solutions existantes et nous choisissons notre modèle.

Chapitre 2, La méthodologie, où nous présentons la préparation des données, la conception des modèles, l'entrainement du modèle et sa validation.

Chapitre 3, La présentation et la discussion des résultats, où nous présentons nos résultats obtenus.

Chapitre 4, Comparaison de nos résultats avec la régression logistique , ici, nous comparons nos résultats avec ceux de la régression logistique.

Chapitre 5, Les perspectives d'amélioration , ici nous proposons des axes d'amélioration.

Une conclusion.

Chapitre 1

État de l'art et choix du modèle

Comme précisé dans la partie méthodologie de l'introduction, nous répondons à notre problématique à travers la construction d'un modèle de prédiction de l'état futur d'un individu, IPVERT ou non. Notons que IPvert désigne un individu pur vert, c'est à dire un individu qui sera en mesure de payer sa dette dans un délai de douze mois sans intervention administrative. Nous voulons donc prédire une mesure de la volonté et de la capacité de cette personne à s'acquitter de sa dette sous un délai de douze mois comme le stipule la réglémentation de Québec (2023d).

Selon Biernat et Lutz (2019) la régression et la classification sont deux types d'apprentissage supervisé utilisés pour des objectifs distincts. La régression prédit une valeur numérique continue, comme un prix ou une température, avec des modèles tels que la régression linéaire, et est évaluée par des métriques comme l'erreur quadratique moyenne (RMSE). En revanche, la classification attribue des données à des classes distinctes, comme une action, ou une obligation, en utilisant des modèles tels que les arbres de décision et la régression logistique, et est évaluée par des métriques telles que la précision et le rappel. Alors que la régression produit des valeurs continues, la classification génère des catégories. Nous pouvons toutefois utiliser des modèles de regression pour prédire une probabilité comme c'est le cas de notre problème, et ensuite procéder à une classification en se donnant un seuil. Dans ce chapitre, nous présenterons un état de l'art des différents modèles de regression et de classification utilisés pour ces tâches, en examinant leurs caractéristiques, leurs avantages et leurs limites.

1.1 Principales solutions

1.1.1 Naïve bayes

Le classificateur Naive Bayes se base sur le théorème de Bayes qui décrit comment la probabilité d'un événement est évaluée sur la base d'une connaissance préalable des conditions qui pourraient être liées à l'événement (Joachims (1999)).

L'algorithme de classification Naive Bayes présente plusieurs avantages. Comme étudié dans Biernat et Lutz (2019) il est simple à mettre en œuvre, requiert un temps d'apprentissage minimal pour la machine, et fonctionne bien lorsque les variables d'entrée ont des valeurs catégorielles. Cet algorithme donne de bons résultats pour des problèmes complexes du monde réel et est efficace pour la classification multi-classes (Rennie et al. (1987)). Lorsque l'hypothèse d'indépendance est valide, Naive Bayes est souvent plus performant que d'autres algorithmes comme la régression logistique et nécessite moins de données d'entraînement. Toutefois, il a aussi des inconvénients. Il suppose l'indépendance entre les variables de caractéristiques (Langley et Sage (1994)), ce qui n'est pas toujours le cas. Si une variable catégorielle appartient à une catégorie non observée dans l'ensemble d'apprentissage, le modèle lui donnera une probabilité de 0, empêchant ainsi toute prédiction. De plus, il est souvent considéré comme un mauvais estimateur, les probabilités calculées n'étant pas toujours fiables. Si une variable non observée pendant l'apprentissage est rencontrée lors des tests, le modèle attribue une probabilité de 0, une situation que l'on peut éviter en utilisant des procédures de lissage comme l'estimation de Laplace.

L'algorithme Naive Bayes trouve des applications dans divers domaines. Il est utilisé pour la classification des pourriels, en identifiant si un courriel est un spam ou non en fonction de son contenu. Il est également utilisé dans l'analyse textuelle (McCallum et Nigam (1998)), adapté aux systèmes de prédiction en direct, grâce à sa rapidité qui permet de prédire la variable cible en temps réel. Dans l'analyse des sentiments, il est utilisé pour reconnaître les commentaires sur un produit et les classer comme positifs ou négatifs (McCallum et Nigam (1998)). Enfin, il est efficace pour les problèmes de classification multi-classes en apprentissage automatique.

1.1.2 Régression linéaire et logistique

La régression linéaire, parmi les approches classiques de prédiction de variables quantitatives, demeure une méthode fondamentale et largement utilisée. Selon Biernat et Lutz (2019), la regression linéaire est fondée sur l'hypothèse d'une relation linéaire entre les variables prédictives $x_1, x_2, ..., x_P$ et la variable cible y, elle vise à estimer les coefficients β tels que l'erreur du modèle soit minimisée. Cette méthode est formalisée par l'équation :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_P x_P \quad \text{(Eq1)}$$

La qualité de l'ajustement du modèle est évaluée par la fonction de coût $J(\beta_0, \beta_1, ..., \beta_P)$, qui mesure l'écart entre les valeurs prédites et observées. L'objectif est de minimiser cette fonction de coût pour obtenir les estimations optimales des coefficients β. Deux approches principales sont couramment utilisées pour estimer ces paramètres :

Les méthodes de résolution des modèles de régression varient en fonction de l'approche analytique ou itérative. La méthode analytique, basée sur la méthode des moindres carrés ordinaires, offre une solution explicite pour les coefficients. Cependant, cette méthode peut être coûteuse en termes de calculs, notamment pour des ensembles de données de grande taille. En revanche, la méthode itérative, telle que la descente de gradient, ajuste progressivement les coefficients en se basant sur des valeurs initiales arbitraires. Cette approche présente l'avantage d'une complexité computationnelle moindre, ce qui la rend plus adaptée aux grands ensembles de données.

La régression linéaire offre une approche simple et interprétable pour la modélisation des relations entre variables. Cependant, ses limites résident dans sa capacité à modéliser des relations non linéaires et dans sa dépendance vis-à-vis de variables explicatives exclusivement continues. Son utilisation efficace nécessite une compréhension approfondie des principes sous-jacents et une évaluation rigoureuse de l'adéquation du modèle aux données disponibles.

Contrairement à la régression linéaire qui est utilisée pour prédire des variables continues, comme le montre Peng et al. (2002) la régression logistique est spécifiquement conçue pour faire de la classification en pour modélisant la probabilité qu'un événement

se produise en fonction d'un ensemble de variables prédictives.

La différence fondamentale entre la régression linéaire et la régression logistique réside dans la nature de la variable cible et la fonction utilisée pour modéliser la relation entre les variables prédictives et la variable cible.

En régression logistique, la relation entre les variables prédictives $x_1, x_2, ..., x_P$ et la variable binaire y est modélisée à l'aide de la fonction logistique, qui prend la forme :

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_P x_P)}}$$

Dans cette équation, $P(y = 1|x)$ représente la probabilité conditionnelle que la variable y prenne la valeur 1 étant donné les valeurs des variables prédictives $x_1, x_2, ..., x_P$, et $\beta_0, \beta_1, ..., \beta_P$ sont les coefficients à estimer.

La fonction logistique $\frac{1}{1+e^{-z}}$ transforme la combinaison linéaire des variables prédictives $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_P x_P$ en une probabilité comprise entre 0 et 1.

La différence majeure avec la régression linéaire réside donc dans la fonction de lien utilisée pour modéliser la relation entre les variables prédictives et la variable cible. Alors que la régression linéaire utilise une fonction linéaire pour prédire des valeurs continues, la régression logistique utilise une fonction logistique pour prédire des probabilités.

La régression logistique en apprentissage automatique offre plusieurs avantages. Selon Maalouf (2011) C'est un modèle simple, ce qui nécessite peu de temps pour la formation, et il peut gérer un grand nombre de fonctionnalités. Cependant, il présente également des inconvénients, étant limité à des problèmes de classification binaire et donnant des résultats moins satisfaisants pour la classification multi-classes. Concernant ses applications, la régression logistique est largement utilisée dans divers domaines. Elle est employée dans le pointage de crédit pour prédire la solvabilité des individus en fonction de certaines caractéristiques financières. De plus, de nombreux sites Web utilisent la régression logistique pour prédire le comportement des utilisateurs et les guider vers des actions telles que les clics sur des liens. Enfin, elle est également utilisée dans l'analyse des choix discrets, permettant de prédire les préférences catégorielles des individus, comme le choix d'une voiture, d'une école, ou d'un collège, en fonction de divers attributs et

options disponibles.

1.1.3 Algorithme de classification d'arbre de décision

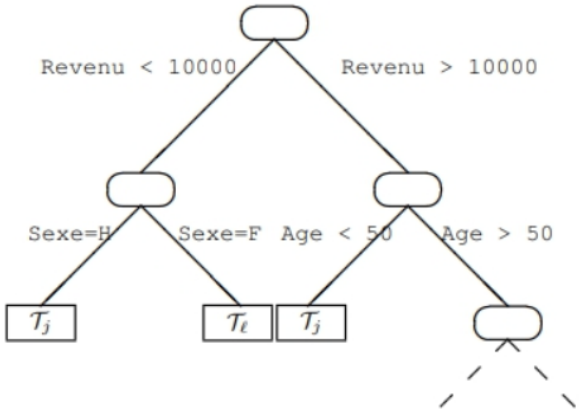

Figure 1.1 – Illustration d'un CART. Cette figure présente un exemple d'arbre de décision utilisant la méthode CART (Classification and Regression Trees). Chaque nœud interne représente une règle de décision binaire basée sur des variables telles que le sexe, le revenu et l'âge. Source : http ://wikistat.fr/pdf/st-m-app-cart.pdf

Comme le mentionne Zhang et Zhou (2014), la méthode des arbres de décision, également désignée sous le nom de CART (Classification And Regression Trees), est une technique polyvalente applicable tant aux problèmes de régression qu'aux problèmes de classification, où la variable à prédire est quantitative. Chaque nœud de l'arbre traite une portion du jeu de données en entrée et génère une division en deux branches distinctes, 1 et 2, basée sur une variable explicative du jeu de données. Le nœud initial peut diviser les données selon le sexe (homme ou femme), suivi par une division selon le revenu (supérieur ou inférieur à un seuil donné), et enfin selon l'âge (plus ou moins qu'une valeur spécifique). D'après Genuer et Poggi (2017), les branches illustrent les différentes issues des règles de décision, menant aux nœuds terminaux ou feuilles, qui contiennent les classifications ou prédictions finales.

La manière dont les données sont divisées dans les arbres de décision dépend de la nature de la variable explicative comme le montre Grus (2017). Pour une variable qualitative ordinale, les observations sont dirigées vers le nœud enfant de gauche si elles ont des

valeurs inférieures à un seuil prédéfini, sinon elles vont vers le nœud enfant de droite. Il existe ainsi $k1$ divisions possibles pour une variable avec k modalités. De même, pour une variable qualitative nominale, les observations associées à différentes modalités sont réparties entre les deux branches de l'arbre, offrant également $k1$ divisions possibles. Enfin, pour Loh (2011), pour une variable quantitative, le jeu de données est divisé en deux parties en fonction d'un seuil sur la variable.

L'algorithme choisit la division optimale en maximisant la différence de désordre dans les ensembles de données résultants. Dans le cas de la régression, le désordre est mesuré par la variance de la variable cible y dans l'ensemble K, définie comme $\text{Var}(y) = \frac{1}{N}\sum(y_i - \bar{y})^2$, où $\bar{y}$ est la moyenne de y dans l'ensemble K.

Le processus de division est répété pour chaque nœud enfant, avec des critères d'arrêt basés sur le désordre du nœud ou le nombre d'individus. Une fois l'arbre construit, l'estimation pour un individu consiste en la moyenne des observations dans la feuille correspondante.

Selon Quinlan (1987), l'algorithme de classification par arbre de décision présente plusieurs avantages. Tout d'abord, il permet une représentation simple des données, facilitant ainsi leur interprétation et leur explication aux dirigeants. De plus, les arbres de décision imitent la manière dont les Hommes prennent des décisions dans la vie quotidienne, ce qui les rend intuitifs à comprendre. Ils sont également capables de gérer efficacement les variables cibles qualitatives et les données non linéaires. Cependant, ils ont également des inconvénients. Ils peuvent créer des arbres complexes qui deviennent parfois non pertinents, et leur niveau de précision de prédiction peut être inférieur à celui d'autres algorithmes. Concernant leurs applications, les arbres de décision sont utilisés dans divers domaines. Par exemple, dans l'analyse des sentiments, ils sont employés comme algorithme de classification pour déterminer le sentiment d'un client envers un produit à partir de l'exploration de texte. De même, les entreprises peuvent utiliser des arbres de décision pour sélectionner les produits qui leur procureront des bénéfices plus élevés lors de leur lancement.

1.1.4 Algorithme de classification des k plus proches voisin

L'algorithme des k plus proches voisins (kNN) est une méthode d'apprentissage supervisé utilisée pour la classification et la régression. Dans cette approche, la prédiction d'une variable cible pour un nouvel individu est basée sur les observations les plus similaires dans l'ensemble de données.En effet, selon Cover et Hart (1967),la règle de décision du plus proche voisin attribue à un point d'échantillonnage non classé la classification du plus proche d'un ensemble de points précédemment classés.

Pour prédire une variable y pour un individu i, on utilise dans un algorithme les valeurs x des K individus les plus similaires à i. Et pour mesurer cette similarité, différentes mesures de distances entre individus sont utilisées. Comme le montre Kataria et Singh (2013), les mesures de distance classiques incluent la distance euclidienne, la distance de Manhattan et la distance de Chebyshev.

Dans la méthode des k plus proches voisins, la prédiction pour un point i peut être calculée de deux manières (Jai (2007)). Premièrement, en prenant la moyenne des valeurs des individus de N_i : $\hat{y}_i = \frac{1}{K} \sum_{x_j \in N_i} y_j$. Deuxièmement, en utilisant une moyenne pondérée des valeurs des individus de N_i, où les poids w_{ij} sont attribués en fonction de la proximité de x_j avec i : $\hat{y}_i = \frac{1}{K} \sum_{x_j \in N_i} w_{ij} y_j$. Dans cette formule, w_{ij} représente un poids attribué à la valeur y_j en fonction de sa proximité avec i.

Ozturk Kiyak et al. (2023) montrent également qu'on peut améliorer la performance d'un KNN en considérant les voisins des K plus proches voisins.

Comme le montre Syriopoulos et al. (2023), l'algorithme de classification KNN présente plusieurs avantages. Il peut être appliqué à des ensembles de données de n'importe quelle distribution et est facile à comprendre, ce qui le rend assez intuitif. Cependant, il présente également des inconvénients. Il est facilement affecté par les valeurs aberrantes, ce qui peut fausser les résultats. De plus, il est biaisé vers une classe qui a plus d'instances dans l'ensemble de données, ce qui peut entraîner des résultats non représentatifs. Il peut être difficile de trouver le nombre optimal pour K, ce qui peut affecter la précision du modèle. En ce qui concerne ses applications, le KNN est utilisé dans divers domaines. Par exemple, dans la détection des valeurs aberrantes, il est capable de repérer les instances

anormales en raison de sa sensibilité à celles-ci. De même, il est utilisé dans l'identification de documents similaires pour reconnaître des documents sémantiquement similaires.

1.1.5 Les méthodes ensemblistes et forêt aléatoire

Les méthodes ensemblistes sont une approche sophistiquée de la modélisation prédictive, intégrant les résultats de plusieurs modèles pour créer un modèle global plus performant. Trois principales familles de méthodes se distinguent : bagging, stacking, et boosting.

Le Bagging (Bootstrap Aggregating) consiste à construire plusieurs ensembles de données d'entraînement par échantillonnage avec remplacement, puis à entraîner un modèle sur chaque ensemble. Comme l'explique Breiman (1996), les prédictions des modèles sont ensuite moyennées ou votées. Cette technique est utilisée notamment dans les forêts aléatoires, comme le montre Grus (2017).

Le stacking (Empilement) divise l'ensemble de données d'entraînement, entraîne plusieurs modèles de base sur des parties de l'ensemble, puis entraîne un méta-modèle sur les prédictions des modèles de base. Cette approche permet une plus grande flexibilité dans l'agrégation des résultats.

Le boosting entraîne successivement des modèles de faible performance, chaque modèle corrigeant les erreurs de son prédécesseur. Les algorithmes comme AdaBoost et XGBoost utilisent cette méthode, en ajustant les poids des observations à chaque itération.

Les méthodes ensemblistes améliorent les performances des modèles en réduisant les erreurs de prédiction et en améliorant la stabilité, notamment avec le boosting.

Selon Biernat et Lutz (2019), une forêt aléatoire combine plusieurs arbres de décision, chaque arbre prédisant une valeur pour la probabilité des variables cibles. Les probabilités sont ensuite moyennées pour la sortie finale. La forêt aléatoire utilise des méthodes d'apprentissage supervisé pour des problèmes de classification et de régression, intégrant plusieurs classificateurs pour augmenter les performances du modèle. Le principe de CART est de partitionner récursivement l'espace des variables explicatives en deux. Et comme l'explique Liaw et Wiener (2002), dans une forêt aléatoire, chaque nœud est divisé en utilisant le meilleur prédicteur parmi un sous-ensemble de prédicteurs choisis aléatoire-

ment à ce nœud.

Les forêts aléatoires assurent la diversité, réduisent l'espace des fonctionnalités, peuvent être parallélisées, et n'ont pas besoin de séparation préalable des données en ensembles d'entraînement et de test. Leur stabilité est assurée par un vote majoritaire ou une moyenne.

L'algorithme de classification aléatoire des forêts est efficace pour les grands ensembles de données, permet d'estimer la significativité des variables, et est plus précis que les arbres de décision. Cependant, sa mise en œuvre est plus complexe et nécessite plus de temps pour l'évaluation et l'ajustement des paramètres. Il est utilisé dans divers domaines, comme la finance pour prédire les défauts de paiement, et le commerce électronique pour recommander des produits.

1.1.6 Revue des méthodes pour la détection de fraude et la prédiction de remboursement

2.1.6.1 Détection de fraude

La détection de fraude est un domaine clé qui a fait l'objet de plusieurs innovations méthodologiques au fil des années. L'objetif est d'identifier des transactions frauduleuses parmi un grand volume de transactions légitimes, en utilisant diverses techniques d'apprentissage automatique et de statistiques.

Ngai et al. (2011) proposent une revue des techniques de détection de fraude en classant les méthodes en trois catégories principales : les techniques de classification, de clustering et de détection d'anomalies. Les techniques de classification, telles que la régression logistique et les arbres de décision, apprennent des modèles à partir de données étiquetées. Les méthodes de clustering, comme k-moyenne, regroupent les transactions similaires pour détecter les anomalies en identifiant les groupes atypiques. Enfin, les techniques de détection d'anomalies, comme les modèles de forêts isolées, sont spécialement conçues pour identifier les points de données qui se distinguent significativement des autres.

Dans Phua et al. (2012), les auteurs examinent diverses approches utilisées pour la détection de fraude dans différents domaines, y compris les fraudes par carte de crédit,

téléphoniques et d'assurance. Ils soulignent l'importance de combiner plusieurs techniques pour améliorer la précision et la robustesse des modèles de détection.

2.1.6.2 Prédiction de la probabilité de remboursement

La prédiction de la probabilité de remboursement est une autre application des modèles de régression, visant à estimer la probabilité qu'un individu ou une entreprise rembourse un prêt.

Coussement et Van den Poel (2008) explorent l'utilisation de la régression logistique pour prédire la probabilité de remboursement dans le secteur bancaire. Les auteurs montrent que la régression logistique permet de capturer efficacement les relations non linéaires entre les variables explicatives et la probabilité de remboursement. Ils recommandent également l'utilisation de techniques de sélection de variables pour améliorer les performances du modèle.

Une autre approche intéressante est présentée par Lessmann et al. (2015), qui discute de l'application des méthodes ensemblistes, telles que les forêts aléatoires et le boosting, pour améliorer la précision des prédictions de remboursement. Les auteurs démontrent que ces techniques peuvent capturer des interactions complexes entre les variables explicatives et offrir des performances supérieures par rapport aux modèles individuels.

Ces études montrent que la combinaison de différentes approches méthodologiques et l'utilisation de techniques avancées d'apprentissage automatique peuvent significativement améliorer la précision des modèles de détection de fraude et de prédiction de la probabilité de remboursement. Ces techniques seront prises en compte dans notre choix final de modèle pour notre projet.

1.1.7 État de l'art de la prévision de la conformité fiscale avec les modèles d'apprentissage machine

La prévision de la conformité fiscale a évolué grâce à l'intégration des modèles d'apprentissage machine, qui offrent des approches sophistiquées pour améliorer la détection des anomalies et prédire le risque de défaut de paiement d'une dette fiscale. Selon Q. Zheng et al. (2024), les méthodes de détection du risque peuvent être divisées en deux catégo-

ries : les méthodes basées sur les relations et les méthodes non basées sur les relations. Il dénombre au total, 14 méthodes de détection des risques. Les méthodes d'apprentissage machine, telles que les forêts aléatoires, les réseaux de neurones et les machines à vecteurs de support (SVM), ont été largement explorées pour ce domaine. Par exemple, les forêts aléatoires, comme détaillé par Liaw et Wiener (2002) et Breiman (2001), ont montré leur capacité à classifier efficacement les contribuables à risque élevé en combinant plusieurs arbres de décision pour une prédiction plus robuste. De plus, les approches basées sur les réseaux de neurones, telles que celles discutées par Krizhevsky et al. (2017) et LeCun et al. (2015), permettent une modélisation plus complexe des relations non linéaires entre les variables fiscales, offrant ainsi une précision accrue dans les prévisions. Les modèles de boosting, notamment XGBoost et AdaBoost, comme présentés par Chen et Guestrin (2016) et Freund et Schapire (1997), améliorent également les performances prédictives en ajustant les erreurs des modèles précédents. L'utilisation des machines à vecteurs de support, explorée par Cortes et Vapnik (1995) et Schölkopf et Smola (2001), offre une approche efficace pour la classification des contribuables en utilisant des hyperplans séparateurs dans des espaces de grande dimension. Les méthodes d'apprentissage supervisé, sont essentielles pour entraîner les modèles sur des ensembles de données annotées, tandis que les méthodes non supervisées, abordées par Jain et al. (1999), sont utilisées pour détecter les anomalies sans labels préalables. Enfin, les techniques d'analyse de données massives, discutées par Dean et Ghemawat (2008), permettent de traiter de grands volumes de données fiscales pour en extraire des modèles de conformité plus précis. La combinaison de ces approches améliore la capacité des administrations fiscales à anticiper les comportements non conformes et à optimiser les stratégies de vérification et de recouvrement.

1.1.8 Évaluation des modèles de régression

L'évaluation des modèles de régression est une étape cruciale dans le processus de modélisation, et la littérature propose plusieurs méthodes et métriques bien établies.

L'évaluation d'un modèle de régression repose sur plusieurs métriques importantes, parmi lesquelles :

RMSE (Root Mean Square Error) : Cette métrique représente la racine carrée de la moyenne des erreurs quadratiques du modèle sur l'ensemble des individus. Elle offre une mesure globale de l'erreur du modèle.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

MAE (Mean Absolute Error) : Le MAE représente la moyenne des valeurs absolues des erreurs du modèle sur l'ensemble des individus. Contrairement au RMSE, il est moins sensible aux valeurs aberrantes.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

R-Squared (Coefficient de Détermination) : Cette métrique évalue la proportion de variance totale de la variable cible expliquée par le modèle. Un coefficient proche de 1 indique une bonne adéquation du modèle aux données.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

Outre les métriques, comme Biernat et Lutz (2019), il est essentiel d'évaluer la robustesse des modèles, c'est-à-dire leur capacité à maintenir leurs performances face à des variations des données. Nous utilisons la validation croisée qui consiste en la division répétée du jeu de données en ensembles d'entraînement et de test, avec des parties de test disjointes à chaque division. Le modèle est entraîné sur les ensembles d'entraînement et évalué sur les ensembles de test. La stabilité du modèle est jugée sur la cohérence des performances sur les ensembles de test. C'est à dire que sur des échantillons contruits aléatoirement avec des proportions des IPVERT similaires, on devrait s'attendre à des performances plus ou moins égales du modèle.

1.1.9 Critères de choix

Pour Biernat et Lutz (2019), la première chose à faire est de bien comprendre la tâche à accomplir. S'il s'agit d'un cas de classification supervisée, on peut utiliser des algorithmes tels que la régression logistique, la forêt aléatoire, l'arbre de décision, etc. En revanche,

s'il s'agit d'un cas de classification non supervisée, on peut opter pour des algorithmes de clustering.

Pour Lemberger et al. (2015), la taille de l'ensemble de données est également un paramètre à prendre en compte lors de la sélection d'un algorithme. Comme peu d'algorithmes sont relativement rapides, il vaut mieux passer à ceux-là. Si la taille de l'ensemble de données est petite, on va s'en tenir à des algorithmes à faible biais/forte variance comme Naive Bayes. En revanche, si l'ensemble de données est volumineux, le nombre de fonctionnalités est élevé, on optera alors pour des algorithmes à fort biais/faible variance tels que KNN, Arbres de décision et SVM.

Pour Biernat et Lutz (2019) : la précision d'un modèle est une caractéristique qui teste la qualité d'un classificateur. Il reflète à quel point la valeur de sortie prédite correspond à la valeur de sortie correcte. Bien sûr, une plus grande précision est souhaitable, mais il convient également de vérifier que le modèle ne surajuste pas, c'est-à-dire qu'il ne s'adapte pas trop étroitement aux données d'entraînement. Un modèle qui surajuste peut offrir une précision élevée sur les données d'entraînement mais échouer à généraliser efficacement sur de nouvelles données, ce qui est aussi connu sous le nom de surapprentissage.

Pour Grus (2017), les algorithmes complexes d'apprentissage automatique de classification tels que SVM et forêts aléatoires peuvent prendre beaucoup de temps pour le calcul. De plus, une plus grande précision et de grands ensembles de données nécessitent de toute façon plus de temps pour apprendre le modèle. Des algorithmes simples comme la régression logistique sont plus faciles à mettre en œuvre et permettent de gagner du temps.

Pour Biernat et Lutz (2019) : il n'y a pas toujours de relation linéaire entre les variables d'entrée et les variables cibles. Il est donc essentiel d'analyser cette relation et de choisir l'algorithme avec soin car certains d'entre eux sont limités à des jeux de données linéaires. La meilleure méthode pour vérifier la linéarité consiste soit à ajuster une ligne linéaire, soit à exécuter une régression logistique ou SVM et à rechercher les erreurs résiduelles. Une erreur plus élevée suggère que les données ne sont pas linéaires et nécessiteraient la mise en œuvre d'algorithmes complexes.

Pour Grus (2017), parfois, le jeu de données peut contenir inutilement de nombreuses

variables, et toutes ne seront pas pertinentes. On peut alors utiliser des algorithmes comme SVM, mieux adaptés à de tels cas, ou utiliser l'analyse en composantes principales pour déterminer quelles caractéristiques sont importantes.

1.1.10 Le choix du modèle

Le Modèle forêt aléatoire augmenté d'un KNN

Notre modèle est un modèle hybride par augmentation de variables. Il est constitué d'un moteur de calcul KNN pour déterminer des estimateurs de la volonté et de la capacité du particulier. Ces indicateurs seront ensuite repris en plus des autres variables sélectionnées, par un modèle forêt aléatoire qui fera la prédiction finale.

Pour créer nos arbres de décisions, nous utilisons la méthode bagging.

La forêt aléatoire est un choix judicieux pour des tâches de classification supervisée grâce à sa robustesse et sa capacité à gérer de grands ensembles de données avec de nombreuses variables, en l'occurence une cinquantaine pour notre projet. La forêt aléatoire excelle dans la modélisation de relations non linéaires, offrant une grande précision prédictive sans surajustement. Bien que son temps de formation puisse être plus long que celui de modèles simples, sa capacité de parallélisation et sa méthode de bagging permettent de réduire ce temps de manière efficace. En combinant plusieurs arbres de décision, elle maintient un bon équilibre entre biais et variance, ce qui la rend particulièrement adaptée aux ensembles de données complexes et volumineux comme le nôtre.

Chapitre 2

Méthodologie

2.1 La préparation des données

2.1.1 La collecte des données

Les particuliers non-mandataires représentent 82% des dossiers créés en recouvrement, avec un volume financier de 58% selon l'exercice 2016-2017. La création d'un dossier de recouvrement se fait par le Système de Perception Intégré des Créances (SPIC), où le débiteur reçoit un avis de cotisation, et une stratégie de recouvrement lui est allouée.

Les données choisies pour entraîner le modèle proviennent des dossiers fiscaux de type clientèle impôt personnel ou Individu Pur (IP) créés dans le SPIC en 2016-2017, plus précisément du 1er avril 2017 au 31 mars 2017. Nous utilisons une année complète pour éviter les phénomènes de saisonnalité et pour confronter les résultats de notre prédiction aux résultats réels. Nous avons également choisi des données éloignées dans le temps pour réserver les données les plus récentes pour les tests pré-production de notre modèle prédictif.

La collecte des données, comme le souligne Québec (2023b), a permis de rassembler les informations nécessaires à notre modèle à partir de diverses sources. La table de données, multisource, inclut des variables sélectionnées pour leur fort potentiel prédictif. Les données internes à la DGR comprennent les variables de l'inventaire fiscal, les données SPIC et ISA (Indicateur de Secteur et d'Assignation), ainsi que les données TP1 importées

si disponibles dans les 30 jours suivant la création du dossier SPIC. Les données externes à la DGR incluent l'indice de richesse des ménages stables pour l'année 2015, certains indicateurs de l'année 2016-2017 des données ISA ou TP1 provenant de DGIA, et les dossiers de contribuables ayant fait l'objet de vérifications fiscales.

2.1.2 L'exploration des données

Via une analyse exploratoire des données pour comprendre leur structure, leur format et leur qualité, nous avons identifié les valeurs manquantes, les outliers (valeurs aberrantes) et les éventuelles incohérences.

Les dossiers pour lesquels la cote ISA était absente ou dont la stratégie était déjà déterminée ont été exclus de la base de données. Cela inclut les dossiers ayant changé de statut de IP à IA (individu en affaire) ainsi que les dossiers créés avec les types AVIS, CONC, CONS, etc., ces derniers représentant moins de 2% du total initial des données. Revenu Québec (2023).

Nous examinons la volonté et la capacité d'un particulier non-mandataire à rembourser ses dettes, en considérant un dossier comme réussi s'il est clôturé en 393 jours ou moins après sa création dans SPIC, sans recours administratif ou judiciaire Revenu Québec (2023). Un dossier est considéré comme un échec si l'une des conditions suivantes est remplie : il reste ouvert au-delà de 393 jours, il fait l'objet d'une intervention judiciaire de niveau 1, 2 ou 3, ou il a subi une intervention de dénitrification. Les dossiers régularisés par les paiements CIS (Crédit Impôt Solidarité) ou réglés par compensation ne sont pas comptabilisés comme des échecs. La variable à prédire est binaire, indiquant soit un succès, soit un échec.

2.1.3 Le nettoyage des données

Pour traiter les données problématiques identifiées lors de l'exploration, nous avons effectué des actions telles que le remplacement des valeurs manquantes, la suppression des valeurs abérrantes, la normalisation des données, etc. Nous avons supprimé les données postérieures. Il s'agit des variables dont l'information est rendue disponible plus de 30

jours après la création du dossier SPIC.

2.1.4 La sélection, la tranformation et l'encodage des variables

Abedin et al. (2022) démontre le rôle majeur de la transformation des variables, telles que la transformation logarithmique et la racine carrée, dans l'amélioration des performances de la prédiction des défaillances fiscales.Parmi les variables candidates qui seront utilisées comme entrées pour notre modèle, nous avons sélectionné celles qui ont le plus d'impact sur la variable cible comme recommandé par Lemberger et al. (2015) et qui sont les plus pertinentes pour notre problème. Ceci a été fait grâce à l'ajustement d'un arbre de décision qui nous a permis de sélectionner les meilleures variables. Nous utiliserons également un filtre de corrélation pour éliminer les variables qui apportent les mêmes informations.

Nous avons normalisé toutes les varialbes pour les ramener à la même échelle, et nous avons créé de nouvelles variables à partir des existantes, notamment celles qui sont construites par le modèle KNN, c'est à dire la volonté, la capacité implicite et la capacité explicite. Nous avons normalisé car les différentes variables ont des échelles différentes, il a fallu éviter que certaines variables aient un impact disproportionné sur le modèle. Nous avons opté pour la normalisation z-score. Comme l'explique Lemberger et al. (2015), on transforme les valeurs en scores z en soustrayant la moyenne et en divisant par l'écart-type, afin d'obtenir une distribution avec une moyenne de 0 et un écart-type de 1.

Les données sont divisées en ensembles d'entraînement (training set), de validation (validation set) et de test (test set). L'ensemble d'entraînement est utilisé pour entraîner le modèle, l'ensemble de validation pour ajuster les hyperparamètres du modèle et l'ensemble de test pour évaluer les performances finales du modèle.

Comme vu avec Grus (2017), nous avons converti les variables catégorielles en variables numériques, en utilisant des techniques telles que le codage one-hot pour des catégories simples et le label encoding pour des variables catégorielles avec graduation, afin de permettre au modèle de les traiter correctement. Le one-hot encoding a été préféré car il fournit une représentation binaire plus explicite, tandis que le label encoding induit

une relation d'ordre numérique artificielle entre les catégories, ce qui peut conduire à des résultats biaisés. Nous avons tout de même fait attention à l'augmentation de la dimensionnalité induite par l'utilisation abusive du one-hot encoding.

Étant donné que nous avons un grand nombre de fonctionnalités initiales et que nous souhaitons accélérer le processus d'entraînement, nous appliquons des techniques de réduction de la dimensionnalité, telles que la sélection de fonctionnalités basée sur des mesures de corrélation et d'importance avec un arbre de décision, comme étudié dans Biernat et Lutz (2019). Nous présentons cela dans le chapitre sur les résultats.

2.2 La conception du modèle prédictif

2.2.1 Les données du modèle

A- La délection de variables

Nous disposons d'un ensemble initial de plus de 400 variables (Québec (2023b)) dans notre univers de données. Pour construire un modèle à la fois performant et parcimonieux, nous avons utilisé des algorithmes de réduction de la dimensionnalité par selection des meilleurs variables avec un arbre de décision. Cet algorithme nous a permis de déterminer les 50 meilleures variables pour notre modèle.

B- La variable cible

La variable cible est `IPVERT`, qui détermine si le client est *IPVERT* ou non. Nous rappellons que IPVERT signifie individu pur vert, qui répresente celui qui s'est acquité de sa dette sous un délai de 12 mois comme le stipule Québec (2023d), sans intervention administrative ni judiciaire. Cette variable est déterminée en fonction d'une probabilité supérieure à un certain seuil. Dans notre cadre, nous avons considéré le seuil égal à 0,5.

C- Modélisation de la volonté et de la capacité avec KNN

Rappelons que ces deux caractéristiques ne peuvent pas être dissociées, car elles s'influencent mutuellement. En effet, la capacité influence la volonté, et la volonté influence la capacité.

Nous supposons que tous les clients IPVERT sont ceux qui ont un score combiné volonté-capacité suffisamment élevé. Ainsi, nous pouvons donc modéliser l'un comme le complémentaire de l'autre dans l'ensemble des clients IPVERT.

Nous modélisons la volonté en quatre niveaux :

Grade	**Description**
Grade 1	Les volontaires de grade 1 : il s'agit de la classe des individus qui ont fait des déclarations à temps au moins une fois lors des trois dernières années.
Grade 2	Les volontaires de grade 2 : il s'agit de la classe des individus qui ont été sujets à une Non-production au moins une fois lors des trois dernières années.
Grade 3	Les volontaires de grade 3 : il s'agit de la classe des individus IPVERT qui ont toujours pris une entente soit dans le centre des avis, soit dans le centre des appels.
Grade 4	Les volontaires de grade 4 : il s'agit de la classe des individus qui ont fait une proposition et qui l'ont respectée.

Ces quatre grades nous permettent de créer une population qui manifeste bien le caractère de la volonté. Ainsi, nous pouvons, à l'aide d'une approche collaborative, mesurer la distance entre une nouvelle entrée (un individu lambda) et l'une de ces quatre classes, et prédire si l'individu est volontaire ou non.

Nous construisons également la caractère de la capacité implicite à partir du complémentaire de la population volontaire dans la population totale IPVERT. Nous avons donc une seule classe de capacitaire.

D- Construction des classes

Dans une population IPVERT, par exemple, la population IPVERT de l'exercice 2016-2017, 2016-2015, 2015-2014, on construit les classes Grade1, Grade2, Grade3, Grade4, et C à partir des définitions suivantes. On supprime les éléments qui appartiennent à plusieurs classes pour avoir au final des classes disjointes. Les éléments qui n'appartiennent à aucune de ces classes appartiennent à la classe C.

Pour démarrer, nous commencerons avec les deux premiers grades, que nous éclaterons

en 3 sous-grades. Ce qui nous fera en tout 6 classes en plus de la classe de la capacité implicite, donc un total de 7 classes.

Ce qui donne les classes suivantes :

Classe	**Description**
Classe 1	Individus qui ont fait des déclarations à temps lors de la dernière année.
Classe 2	Individus qui ont fait des déclarations à temps lors des deux dernières années.
Classe 3	Individus qui ont fait des déclarations à temps lors des trois dernières années.
Classe 4	Individus qui ont été coupables de Non-production lors de la dernière année.
Classe 5	Individus qui ont été coupables de Non-production lors des deux dernières années.
Classe 6	Individus qui ont été coupables de Non-production lors des trois dernières années.
Classe 7	Complémentaire de l'union des 6 premières classes.

Notons que les trois premières classes bonifient la volonté, les trois suivantes pénalisent la volonté. Et la dernière classe représente ceux qui ne sont pas influencés par la volonté, donc ceux qui sont influencés par leur capacité, on parle alors de capacité implicite.

E- Classification des utilisateurs

Définition de la topologie

Un espace topologique selon Jai (2007), est un couple (E, T), où E est un ensemble et T une topologie sur E, à savoir un ensemble de parties de E que l'on appelle les ouverts de (E, T) vérifiant les propriétés suivantes : T est stable pour la réunion infinie T est stable pour l'intersection finie $\emptyset$ et E appartiennent à T

Soit E muni de la métrique de Manhattan d_m, l'espace métrique des clients. Un résultat est que (E, $\mathcal{E}$) est un espace topologique, où $\mathcal{E}$ est la tribu de Borel de E.

On dira qu'un client U est proche d'un élément V d'une classe C_i lorsque :

— Une classe C_i est un ensemble de client au plus θ-proche deux à deux.

— Une telle classe C_i munie de la topologie induite est un espace topologique séparé. Chaque élément de C_i étant indicé par son rang.

Définition de la métrique

Jai (2007) appelle (E, d) un espace métrique si E est un ensemble et d une distance sur E. On appelle distance sur un ensemble E, une application d de E*E dans $\mathbb{R}^+$ telle que, pour tout x, y, z de E :

— $d(x, y) = d(y, x)$: la symétrie

— $d(x, y) = 0$ si $x = y$: la séparation

— $d(x, z) \leq d(x, y) + d(y, z)$: inégalité triangulaire

La distance de Manhattan définie par :

$$d_m(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

est la plus adaptée pour mesurer notre espace E (client IPVERT), de plus elle est celle qui réduit le mieux la complexité de notre algorithme de classification.

Il était important de bien définir un espace topologique et une métrique, afin de présenter de façon rigoureuse l'algorithme de classification des clients suivant :

Définition de l'algorithme de classification client

— Pour un nouveau client x, on calcule $d_m(x, y)$ quelque soit y appartenant à la table des utilisateurs classés, c'est-à-dire l'ensemble des 5 classes déjà construites.

— On récupère les k plus proches voisins de x.

— On dénombre la proportion de Classe1, Classe2, jusqu'à Classe n dans ces k plus proches voisins qui sont respectivement appelés.

— On effectue des coefficients de pondération à chaque classe. Ces coefficients pourront être trouvés par fonction réciproque.

— La volonté de x est donnée par la somme pondérée des proportions des classes

— La capacité implicite de x est donnée par la différence entre l'unité et la volonté multipliée par un coeffiscient. Ce coefficient mesure l'influence de la volonté sur la capacité, c'est un hyperparamètre de notre modèle.

Notre modèle, une fois constitué, dispose de paramètres ajustables pour optimiser ses per-

formances, notamment le nombre de voisins k, la corrélation entre la capacité et la volonté, ainsi que les coefficients de pondération de classe. Cette approche permet d'améliorer la modélisation en intégrant ces deux nouvelles variables dans le processus d'augmentation des variables, ce qui enrichit les données d'entrée pour notre forêt aléatoire.

2.3 Entrainement du modèle

2.3.1 Jeu de données d'entrainement

Nous avons utilisé différents jeux de données pour l'entraînement de notre modèle. Chaque jeu de données correspond à une période spécifique et contient des informations sur les clients ainsi que la variable cible IPVERT (individu pur vert). Les proportions des échantillons dans chaque jeu de données ont été soigneusement sélectionnées pour assurer une représentation adéquate des classes et une généralisation efficace du modèle. Nous avons utilisé des fichiers de taille croissante, partant de 10 000 observations, puis 100 000 , puis 200 000 , 300 000 , 500 000 observations, jusqu'à pratiquement 1 500 000 observations. Et ces fichiers ont été générés en respectant les proportionnalités d'individu pur vert et non vert réelle, c'est à dire environ 90% de vert pour 10% de non vert. Québec (2023c)

2.3.2 Validation croisée et recherche sur grille

La validation croisée et la recherche sur grille sont deux techniques fondamentales dans le processus d'optimisation des modèles d'apprentissage automatique. La validation croisée est une méthode permettant d'évaluer les performances d'un modèle en le testant sur plusieurs ensembles de données distincts, appelés plis, afin de fournir une estimation robuste de sa capacité à généraliser à de nouvelles données. Grus (2017)

Le processus de validation croisée commence par diviser l'ensemble de données en k segments, généralement avec k = 5 ou k = 10, où chaque pli est utilisé tour à tour comme ensemble de validation tandis que les autres plis sont utilisés pour l'entraînement du modèle. Cela permet d'évaluer le modèle de manière exhaustive sur différentes combinaisons

de données d'entraînement et de validation, réduisant ainsi le risque de surapprentissage et fournissant une estimation plus fiable des performances attendues sur des données réelles.

La recherche sur grille est une méthode systématique pour optimiser les hyperparamètres d'un modèle en testant toutes les combinaisons possibles à partir d'une grille prédéfinie, comme la profondeur d'un arbre de décision ou le nombre de voisins dans un algorithme KNN. En évaluant chaque combinaison à l'aide de la validation croisée, cette approche permet de déterminer les paramètres offrant les meilleures performances sur les données de validation, assurant ainsi une bonne généralisation du modèle.

En combinant la validation croisée avec la recherche sur grille, on obtient un processus d'optimisation complet pour déterminer les hyperparamètres optimaux d'un modèle, assurant ainsi une bonne généralisation aux nouvelles données. Cette approche est cruciale pour maximiser les performances des modèles d'apprentissage automatique, en particulier pour des tâches complexes comme la classification de données ou la prédiction dans des environnements réels. Pour un modèle de forêt aléatoire, les hyperparamètres à optimiser incluent le nombre d'arbres, la profondeur maximale des arbres et le nombre minimum d'échantillons requis pour diviser un nœud. En utilisant ces techniques, on peut identifier les meilleures combinaisons d'hyperparamètres pour améliorer les performances du modèle sur des ensembles de validation, garantissant ainsi des prédictions précises dans diverses applications pratiques.

2.4 Test et validation du modèle

2.4.1 Jeu de données de test

Le jeu de données de test est essentiel pour évaluer les performances du modèle sur des données indépendantes non vues lors de l'entraînement. Il est constitué d'un ensemble d'échantillons distincts de ceux utilisés pour l'entraînement et la validation. Les proportions des échantillons dans le jeu de données de test sont déterminées de manière à représenter de manière équilibrée les différentes classes ou catégories présentes dans

l'ensemble de données global. Pour le test, nous avons utilisé un fichier de 1 811 679 observations, avec une répartition de 91% de vert pour 9% de non vert, que nous avons divisé en échantillon d'entrainement et échantillon de test. La taille de l'échantillon d'entrainement a été de 1 449 343 observations et celle de l'échantillon de test de 362 335 observations. Québec (2023a)

2.4.2 Performance prédictive du modèle

Voici les critères pour évaluer la performance qui ont été recommmandés dans le gabarit de Québec (2023a) pour évaluer nos modèles :

A- Statistiques de prédiction

Les statistiques de prédiction sont des mesures quantitatives utilisées pour évaluer la précision, la sensibilité, la spécificité et d'autres aspects des performances d'un modèle de classification. Elles comprennent des mesures telles que l'exactitude, le rappel, la précision et le score F1, qui fournissent une indication de la capacité du modèle à prédire correctement les différentes classes.

B- Métrique de performance, taux de détection et d'erreur

Le taux de détection mesure la capacité du modèle à détecter correctement les cas positifs dans l'ensemble de données, tandis que le taux d'erreur représente la proportion d'observations mal classées par le modèle. Ces mesures permettent d'évaluer la performance du modèle en termes de sensibilité et de spécificité.

C- Matrice de confusion

La matrice de confusion est un outil d'évaluation des performances d'un modèle de classification. Elle présente les résultats des prédictions du modèle par rapport aux valeurs réelles dans une forme tabulaire. Les cellules de la matrice représentent le nombre d'observations classées correctement et incorrectement par le modèle, permettant ainsi de visualiser les faux positifs, les faux négatifs, les vrais positifs et les vrais négatifs.

D- Courbe ROC et PR

La courbe ROC (Receiver Operating Characteristic) est une représentation graphique des

performances d'un modèle de classification à différents seuils de classification. Elle trace le taux de vrais positifs en fonction du taux de faux positifs, ce qui permet d'évaluer la capacité du modèle à discriminer entre les classes positives et négatives. Une courbe ROC idéale se rapproche du coin supérieur gauche du graphique, indiquant une sensibilité élevée et une spécificité élevée.

E- Vingtiles

Les vingtiles sont des intervalles contenant 5 % des données triées par ordre croissant. Ils sont utilisés pour diviser les données en vingtiles égaux, ce qui permet d'analyser la distribution des prédictions du modèle et d'identifier les variations de performances entre différentes parties de l'ensemble de données.

Chapitre 3

Présentation et discussion des résultats

3.1 Statistiques de prédiction

Les statistiques de prédiction fournissent des mesures quantitatives de la performance du modèle. La précision représente le pourcentage de prédictions correctes parmi les prédictions positives, le rappel représente le pourcentage de vrais positifs correctement identifiés parmi tous les vrais positifs, et le score F1 est une mesure harmonique entre la précision et le rappel. Le support indique le nombre d'observations dans chaque classe.

Classe	**Précision**	**Rappel**	**Score F1**	**Support**
0	0.67	0.18	0.28	32521
1	0.92	0.99	0.96	329815
Précision globale	0.92			
Moyenne pondérée	0.90	0.92	0.90	362336

Table 3.1 – Tableau des statistiques de prédiction pour le modèle. La précision, le rappel, le score F1 et le support sont fournis pour chaque classe, ainsi que pour la précision globale et la moyenne pondérée.

Les résultats montrent que le modèle a une précision élevée pour la classe 1, mais une précision plus faible pour la classe 0. Le rappel et le score F1 sont également élevés pour la classe 1, indiquant une bonne capacité du modèle à identifier les vrais positifs dans cette classe. Cependant, le rappel et le score F1 sont beaucoup plus faibles pour la classe 0, ce qui suggère que le modèle a du mal à identifier correctement les vrais positifs dans cette classe. En général, la précision globale du modèle est élevée, avec une précision moyenne pondérée de 0.92.

3.2 Métrique de performance

- **Aire sous la courbe ROC :** 89.0%
- **Erreur de classement :** 8.2% ($\frac{(B+C)}{(A+B+C+D)}$)
- **Probabilité de détection :** 99.1% ($\frac{A}{(A+B)}$)
- **Probabilité de fausse alarme :** 82.2% ($\frac{C}{(C+D)}$)

Avec A, B, C, et D qui représentent les valeurs de la matrice de confusion :

- A : le nombre de vrais positifs (instances correctement prédites comme positives)
- B : le nombre de faux positifs (instances incorrectement prédites comme positives)
- C : le nombre de faux négatifs (instances incorrectement prédites comme négatives)
- D : le nombre de vrais négatifs (instances correctement prédites comme négatives)
- L'Aire sous la courbe ROC (Receiver Operating Characteristic) mesure la capacité du modèle à discriminer entre les classes positives et négatives.
- L'Erreur de classement représente la proportion d'observations mal classées par le modèle.
- La Probabilité de détection mesure la capacité du modèle à détecter correctement les cas positifs.
- La Probabilité de fausse alarme est la proportion d'observations négatives incorrectement classées comme positives.

Table 3.2 – Matrice de confusion montrant la performance du modèle de classification. Les cellules de la matrice indiquent le nombre de prédictions pour chaque combinaison d'étiquette réelle et prédite. La première ligne indique les prédictions pour l'étiquette réelle "Vert", où 326,996 observations ont été correctement classées comme "Vert" (A) et 2,819 ont été incorrectement classées comme "Non vert" (B). La deuxième ligne montre les prédictions pour l'étiquette réelle "Non vert", avec 26,730 observations incorrectement classées comme "Vert" (C) et 5,791 correctement classées comme "Non vert" (D).

Étiquette réelle	**Vert**	**Non vert**
Étiquette prédite par le modèle		
Vert	326,996 (A)	2,819 (B)
Non vert	26,730 (C)	5,791 (D)

3.3 Matrice de confusion

Les résultats présentés dans la matrice de confusion fournissent une vue détaillée des performances du modèle de classification. Cette matrice est divisée en quatre cellules :

La cellule (A) représente les observations où le modèle a correctement prédit la classe "Vert" lorsque la classe réelle était également "Vert". Ces valeurs sont importantes car elles indiquent les vrais positifs, c'est-à-dire les cas où le modèle a correctement identifié les instances positives. La cellule (B) représente les observations où le modèle a prédit la classe "Vert" alors que la classe réelle était "Non vert". Ces valeurs correspondent aux faux positifs, où le modèle a incorrectement identifié des instances négatives comme des instances positives. La cellule (C) représente les observations où le modèle a prédit la classe "Non vert" alors que la classe réelle était "Vert". Ces valeurs correspondent aux faux négatifs, où le modèle a incorrectement identifié des instances positives comme des instances négatives. La cellule (D) représente les observations où le modèle a correctement prédit la classe "Non vert" lorsque la classe réelle était également "Non vert". Ces valeurs sont les vrais négatifs, indiquant les cas où le modèle a correctement identifié les instances négatives.

Les résultats de la matrice de confusion indiquent que le modèle a une performance

relativement élevée dans la prédiction de la classe "Non vert", comme en témoigne le nombre élevé de vrais négatifs (cellule D). Cependant, il semble avoir des difficultés à prédire correctement la classe "Vert", comme en témoignent le nombre significatif de faux négatifs (cellule C) et le faible nombre de vrais positifs (cellule A). Cela suggère que le modèle peut être biaisé vers la prédiction de la classe majoritaire, ce qui peut être un défi dans les cas de déséquilibre de classe.

En outre, le nombre de faux positifs (cellule B) est également notable, ce qui indique que le modèle identifie certaines instances de la classe "Non vert" comme étant de la classe "Vert", ce qui peut conduire à des erreurs dans les prédictions.

Après avoir fait l'analyse de ces résultats, nous pouvons évaluer les performances du modèle en termes de précision, de rappel et d'autres mesures de performance. Par exemple, une forte concentration de valeurs dans les cellules (A) et (D) indiquerait une bonne performance globale du modèle, tandis qu'une répartition disproportionnée des valeurs dans les autres cellules pourrait indiquer des problèmes de performance, tels que le déséquilibre des classes ou des erreurs de classification significatives.

3.4 Courbe ROC et PR

3.4.1 Définition

1. **Courbe ROC :**

   La courbe ROC est un graphique qui représente la performance d'un modèle de classification binaire à différents seuils de discrimination. Elle est créée en traçant le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1 - spécificité) pour différents seuils de classification. En d'autres termes, la courbe ROC illustre la capacité du modèle à discriminer entre les classes positives et négatives. Une courbe ROC idéale est celle qui s'approche le plus possible du coin supérieur gauche du graphique, ce qui indique une sensibilité élevée (vrais positifs) et une spécificité élevée (faux positifs faibles).

2. **Courbe PR :**

La courbe PR (Précision-Rappel) est un graphique qui représente la relation entre la précision et le rappel d'un modèle de classification binaire à différents seuils de classification. Contrairement à la courbe ROC, qui utilise le taux de faux positifs comme mesure, la courbe PR utilise la précision (positifs réels divisés par tous les exemples prédits positifs) et le rappel (positifs réels divisés par tous les exemples réels positifs). Elle est souvent utilisée dans les cas où les classes sont déséquilibrées, c'est-à-dire lorsque le nombre d'exemples positifs et négatifs est très différent.

3.4.2 Commentaire sur la courbe ROC de notre modèle

Nous avons obtenu une AUC de 89%.

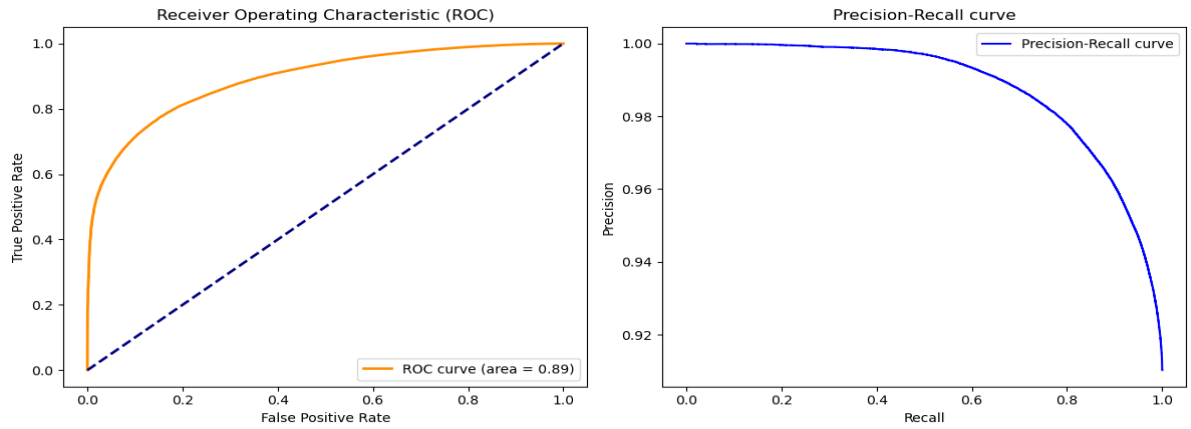

Figure 3.1 – Illustration de la courbe ROC et PR de notre modèle. À gauche, le graphique représente la courbe ROC (Receiver Operating Characteristic), qui trace la relation entre le taux de vrais positifs (True Positive Rate, TPR) sur l'axe des ordonnées et le taux de faux positifs (False Positive Rate, FPR) sur l'axe des abscisses. La droite en pointillé représente la performence d'un modèle alétoire (50%). À droite, le graphique montre la courbe PR (Precision-Recall), qui illustre la relation entre la précision (Precision, sur l'axe des ordonnées) et le rappel (Recall, sur l'axe des abscisses). La précision est définie comme le ratio des vrais positifs sur l'ensemble des prédictions positives, tandis que le rappel est le ratio des vrais positifs sur le total des éléments pertinents.

Ce résultat suggère que notre modèle est capable de séparer efficacement les classes positives et négatives, ce qui est encourageant pour son utilisation dans des tâches de classi-

fication binaire. En effet, la courbe est concave avec un extremum excentré, caractérisant une grande surface sous la courbe (Area Under Curve, AUC) de 89%, indiquant une forte capacité de discrimination du modèle. Plus l'extrémum de la courbe se rapproche du coin supérieur gauche du graphique, plus le modèle est performant, signifiant un haut taux de détection des vrais positifs tout en minimisant les faux positifs.

1. **Courbe Roc :**

   L'aire sous la courbe ROC (AUC) est une mesure de la performance globale du modèle. Elle représente la probabilité qu'un modèle classe aléatoirement un exemple positif plus haut qu'un exemple négatif. Une AUC de 0.5 indique une performance aléatoire, tandis qu'une AUC de 1.0 indique une performance parfaite.

   Une AUC de 89% indique que notre modèle a une capacité significative à discriminer entre les classes positives et négatives. Cela signifie que le modèle est capable de classer correctement environ 89% des paires d'observations positives et négatives. Une AUC de 0.89 est généralement considérée comme une performance très solide pour de nombreux problèmes de classification, bien que cela dépende également du contexte spécifique de l'application.

2. **Courbe PR :**

   La courbe PR évalue la qualité des prédictions positives du modèle, en mettant l'accent sur la précision et le rappel pour les exemples positifs. Une courbe PR idéale serait celle qui se rapproche le plus possible du coin supérieur droit du graphique, ce qui indiquerait à la fois une précision et un rappel élevés.

   Un score élevé sur la courbe PR indique une forte capacité du modèle à identifier correctement les exemples positifs tout en minimisant les faux positifs. Ainsi, un score de précision élevé est souhaitable pour éviter les erreurs de classification positives, tandis qu'un rappel élevé est important pour capturer autant d'exemples positifs que possible.

   Par conséquent, une courbe PR présentant une zone importante sous la courbe indique une performance robuste du modèle en termes de précision et de rappel pour la classe positive. Un modèle avec une courbe PR bien au-dessus de la ligne de

base (aléatoire) indique une performance solide, ce qui est encourageant pour son utilisation dans des tâches de classification binaire avec des classes déséquilibrées.

3.5 Construction du graphe des vingtiles

Pour construire le graphe des vingtiles à partir des résultats de notre modèle forêt aléatoire, nous suivons les étapes suivantes :

1. Nous organisons par ordre croissant les probabilités prédites (*y_pred_proba*) de notre modèle.
2. Nous divisons notre ensemble en vingtiles où chaque vingtile représente 5% de notre ensemble. Pour ce faire, nous identifions les valeurs de probabilité minimale et maximale pour chaque vingtile.
3. Pour chaque vingtile, nous comptons le nombre total d'éléments et le nombre d'éléments dont le résultat réel (y) est égal à 1 (indiquant la classe IPVERT).

En appliquant cette méthode, nous obtenons un tableau des vingtiles présentant les informations suivantes : numéro du vingtile, probabilité minimale et maximale, nombre total d'éléments dans le vingtile, et nombre d'éléments IPVERT dans le vingtile.

Ce tableau nous permet de visualiser la distribution des probabilités prédites par notre modèle et d'analyser les performances de celui-ci sur différentes parties de l'ensemble de données. Le premier vingtile est à 43,7%, le second à 63,7%, etc. À partir du sixième jusqu'au vingtième vingtile, la proportion évolue de manière croissante jusqu'à atteindre 100%. Ce comportement est explicable par la manière dont les vingtiles segmentent la population en fonction des probabilités prédites. Les premiers vingtiles capturent les segments avec les plus faibles probabilités prédites, qui incluent moins de vrais IPVERT, ce qui explique les proportions plus basses dans ces vingtiles. À mesure que l'on progresse vers les vingtiles supérieurs, la probabilité de prédiction augmente, entraînant une concentration plus élevée d'IPVERT parmi les observations. Les segments finaux, contenant les plus élevées probabilités prédites, montrent une proportion élevée d'IPVERT, ce qui est reflété par l'évolution croissante des proportions des vingtiles vers 100%. Ce phénomène indique que le modèle est plus efficace pour identifier les IPVERT parmi les observations avec les plus hautes probabilités prédites, confirmant ainsi la capacité du modèle à bien classer les observations à haute probabilité.

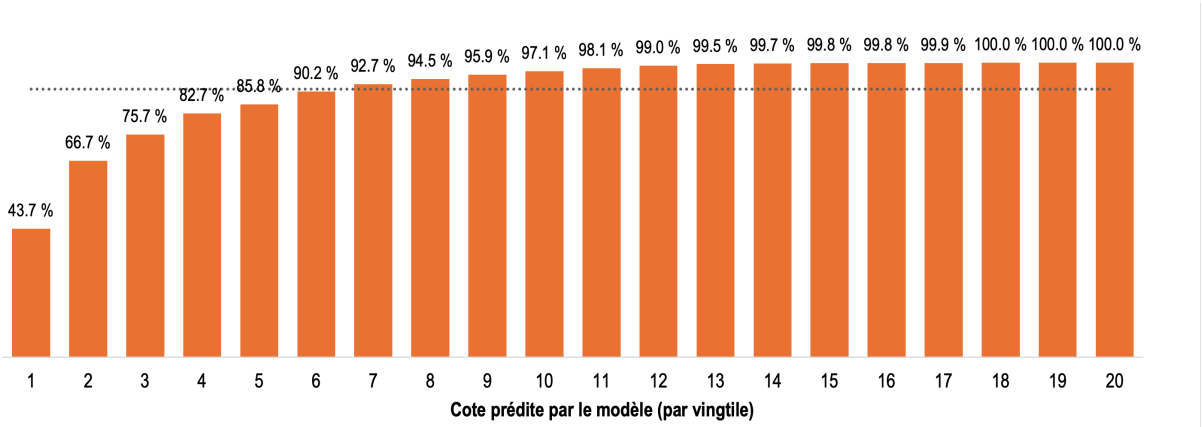

Figure 3.2 – Illustration du graphe des Vingtiles de notre modèle. Cette figure montre la proportion des IPVERT prédite par notre modèle de forêt aléatoire sur des sous-échantillons de la population, chaque sous-échantillon représentant 5% de la population totale, triés par probabilité croissante d'être classés comme IPVERT. Pour construire ce graphe, les probabilités prédites par le modèle ont été organisées par ordre croissant, puis l'ensemble de données a été divisé en vingtiles, chaque vingtile représentant 5% de l'ensemble total. Les valeurs de probabilité minimale et maximale ont été identifiées pour chaque vingtile. Pour chaque vingtile, le nombre total d'éléments et le nombre d'éléments dont le résultat réel est égal à 1 (indiquant la classe IPVERT) ont été comptés.

Table 3.3 – Répartition des dossiers par vingtiles de probabilités prédites. Le tableau montre la distribution des dossiers en fonction des vingtiles de la cote prédite par le modèle. Pour chaque vingtile, les colonnes indiquent les bornes inférieure et supérieure de la cote prédite, le nombre total de dossiers dans ce vingtile, ainsi que le nombre et le pourcentage de dossiers réels IPVERT. Par exemple, dans le vingtile 1, avec une cote prédite entre 0,009 et 0,618, il y a 18 117 dossiers au total, dont 7 909 (43.7%) sont réellement IPVERT. Les chiffres des vingtiles 18 à 20 montrent que le modèle identifie presque tous les dossiers comme IPVERT avec des taux de réussite proches de 100%. Le total général indique que parmi 362 336 dossiers, 329 815 (91.0%) sont réellement IPVERT.

Vingtile	**Cote prédite inf**	**Cote prédite sup**	**Total dossiers**	**Dossiers réel IPVERT (%)**
1	0,009	0,618	18 117	7 909 (43.7%)
2	0,618	0,723	18 117	12 081 (66.7%)
3	0,723	0,786	18 117	13 706 (75.7%)
4	0,786	0,832	18 117	14 986 (82.7%)
5	0,832	0,868	18 116	15 544 (85.8%)
6	0,868	0,899	18 117	16 340 (90.2%)
7	0,899	0,924	18 117	16 788 (92.7%)
8	0,924	0,944	18 117	17 117 (94.5%)
9	0,944	0,959	18 116	17 378 (95.9%)
10	0,959	0,971	18 117	17 591 (97.1%)
11	0,971	0,981	18 117	17 776 (98.1%)
12	0,981	0,988	18 117	17 928 (99.0%)
13	0,988	0,993	18 116	18 022 (99.5%)
14	0,993	0,996	18 117	18 059 (99.7%)
15	0,996	0,997	18 117	18 079 (99.8%)
16	0,997	0,999	18 117	18 078 (99.8%)
17	0,999	0,999	18 116	18 094 (99.9%)
18	0,999	0,999	18 117	18 110 (100.0%)
19	0,999	0,999	18 117	18 114 (100.0%)
20	0,999	1,000	18 117	18 115 (100.0%)
Total			**362 336**	**329 815 (91.0%)**

3.6 Importance des variables

Nous avons construit un tableau qui présente l'importance des variables dans le modèle. Ce tableau n'est pas présenté dans ce mémoire par souci de confidentialité. Ce tableau présente l'importance de chaque variable utilisée dans le modèle, mesurée en valeurs absolues et en pourcentage de l'importance totale. Les variables avec les plus fortes valeurs d'importance ont un impact plus significatif sur les prédictions du modèle. Par exemple, la variable *VOLONTE* a une importance de 0,03 (2,91%), la plaçant parmi les trois variables les plus influentes du modèle.Les autres variables varient en importance, avec certaines contribuant de manière plus modeste à la performance globale du modèle.

En analysant ce tableau, nous avons identifié les variables qui ont le plus d'impact sur les prédictions de notre modèle. Les variables avec les valeurs d'importance les plus élevées sont celles qui contribuent le plus à la capacité du modèle à faire des prédictions précises. Et cette information est très utile pour Revenu Québec. Nous avons vu que la variable VOLONTÉ que nous avions fabriquée avec le modèle KNN est dans le top 3 des meilleurs variables du modèle, avec une contribution dans l'explication de la variable cible de 2.91%. Ce qui est une piste d'amélioration pour des problèmes similaires. Ce qui nous fait rejoindre Antunes et al. (2007) et J. Zheng et Li (2023) qui concluaient que des forces telles que la corrélation entre les individus, l'imitation sociale, l'application des lois par le voisinage local et la réputation ont bien plus de poids que les considérations individuelles, dans la fonction qui détermine les décisions de conformité fiscale.

Chapitre 4

Comparaison des modèles : forêt aléatoire vs régression logistique

Dans ce chapitre, nous comparons les performances des modèles forêt aléatoire (RF) et régression logistique (LR) dans la prédiction de Individu Pur vert. Nous avons utilisé des jeux de données similaires et les mêmes métriques d'évaluation pour assurer une comparaison équitable entre les deux approches.

4.1 Comparaison des métriques de performance

4.1.1 Modèle forêt aléatoire

- **Aire sous la courbe ROC :** 89.0%
- **Erreur de classement :** 8.2% ($\frac{B+C}{A+B+C+D}$)
- **Probabilité de détection :** 99.1% ($\frac{A}{A+B}$)
- **Probabilité de fausse alarme :** 82.2% ($\frac{C}{C+D}$)

Le modèle forêt aléatoire présente une aire sous la courbe ROC de 89.0%, indiquant une bonne capacité de discrimination entre les classes positives et négatives. L'erreur de classement est mesurée à 8.2%, ce qui montre que 8.2% des observations sont mal classées. La probabilité de détection est élevée à 99.1%, ce qui signifie que le modèle détecte correctement la grande majorité des cas positifs. En revanche, la probabilité de

fausse alarme est de 82.2%, indiquant une proportion relativement élevée d'observations négatives incorrectement classées comme positives.

4.1.2 Modèle de régression logistique

- **Aire sous la courbe ROC :** 88.4%
- **Erreur de classement :** 6.9% ($\frac{B+C}{A+B+C+D}$)
- **Probabilité de détection :** 99.0% ($\frac{A}{A+B}$)
- **Probabilité de fausse alarme :** 84.4% ($\frac{C}{C+D}$)

Le modèle de régression logistique montre une aire sous la courbe ROC de 88.4%, légèrement inférieure à celle du modèle forêt aléatoire. L'erreur de classement est de 6.9%, indiquant une meilleure performance en termes de classification précise par rapport au modèle forêt aléatoire. La probabilité de détection est similaire à celle du modèle forêt aléatoire, avec 99.0%, suggérant une capacité élevée à détecter les cas positifs. En revanche, la probabilité de fausse alarme est légèrement plus élevée à 84.4%, ce qui signifie que le modèle de régression logistique a tendance à classer plus d'observations négatives comme positives par rapport au modèle forêt aléatoire.

4.1.3 Commentaire de comparaison

En comparant les deux modèles, bien que le modèle forêt aléatoire présente une aire sous la courbe ROC légèrement supérieure, le modèle de régression logistique montre une meilleure performance en termes d'erreur de classement et de probabilité de fausse alarme. Cela indique que la Régression Logistique pourrait être plus adaptée lorsque la précision de la classification des observations négatives est critique, malgré une légère réduction de la capacité de discrimination entre les classes par rapport au modèle forêt aléatoire.

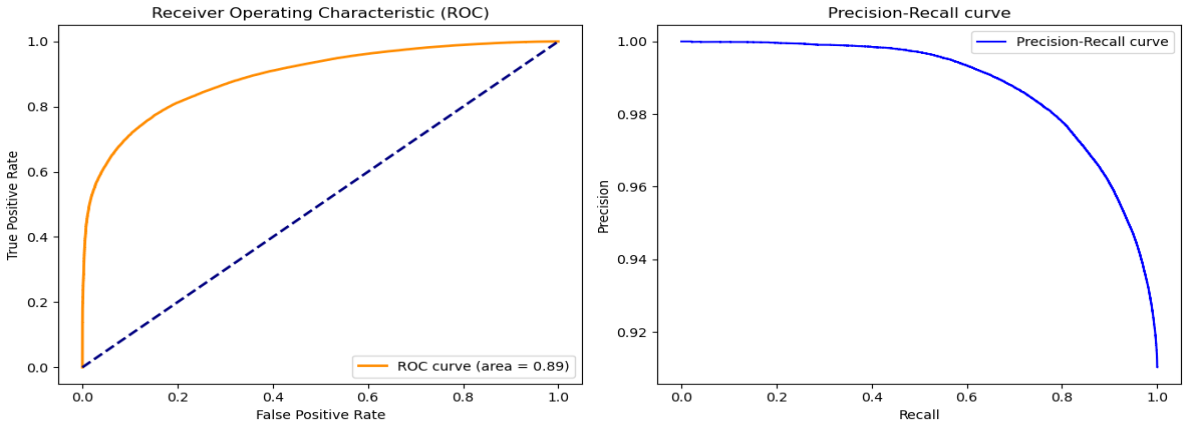

Figure 4.1 – Illustration de la courbe ROC et PR du modèle de forêt aléatoire. À gauche, le graphique représente la courbe ROC (Receiver Operating Characteristic), qui trace la relation entre le taux de vrais positifs (True Positive Rate, TPR) sur l'axe des ordonnées et le taux de faux positifs (False Positive Rate, FPR) sur l'axe des abscisses. La droite en pointillé représente la performance d'un modèle alétoire (50%). À droite, le graphique montre la courbe PR (Precision-Recall), qui illustre la relation entre la précision (Precision, sur l'axe des ordonnées) et le rappel (Recall, sur l'axe des abscisses). La précision est définie comme le ratio des vrais positifs sur l'ensemble des prédictions positives, tandis que le rappel est le ratio des vrais positifs sur le total des éléments pertinents.

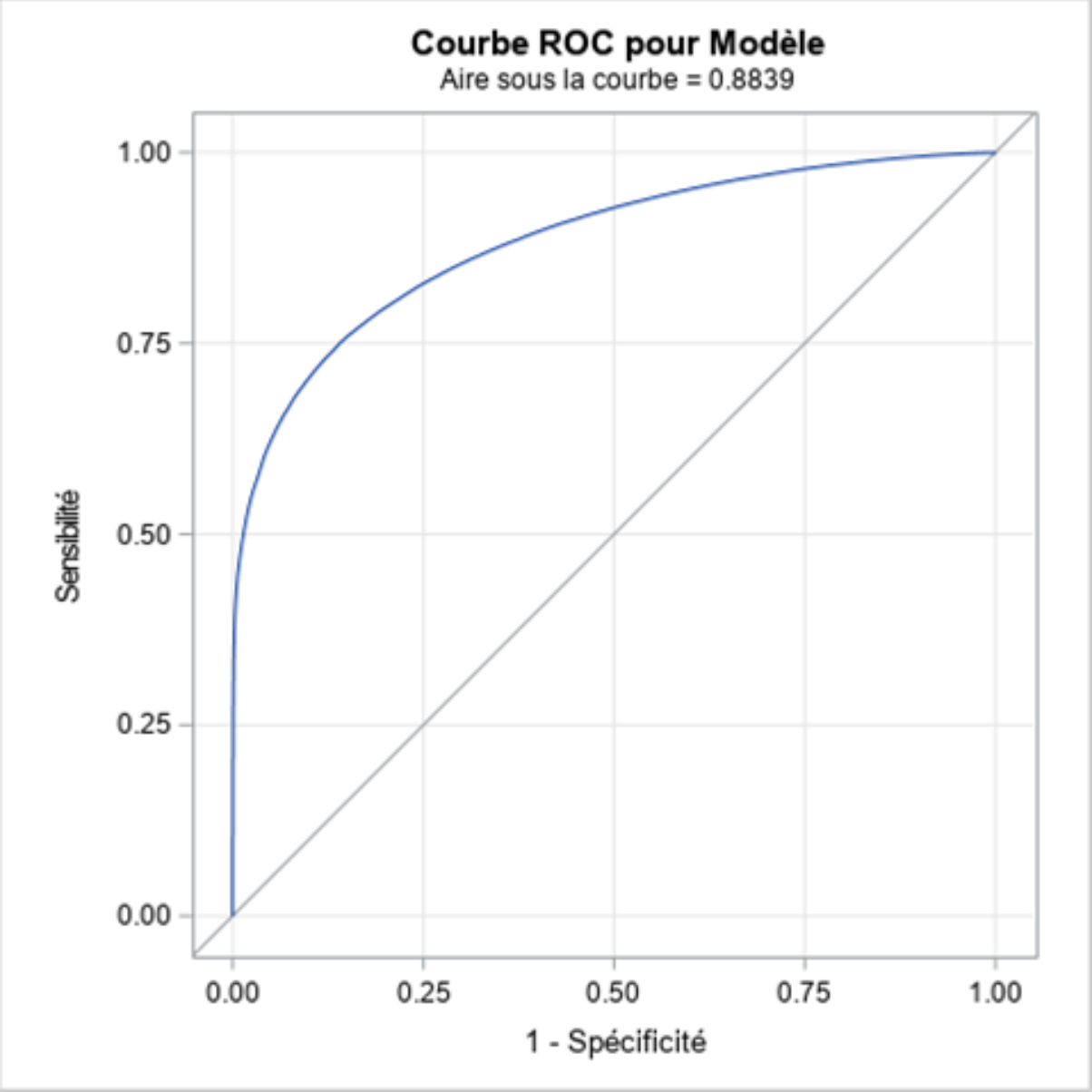

Figure 4.2 – Ci-dessus une illustration de la courbe ROC du modèle de regression logistique. Sur cette image, nous voyons la courbe Roc du modèle de regression logistique. Nous voyons qu'elle a sensiblement la même concavité que celle du modèle de forêt aléatoire.

4.2 Comparaison des vingtiles

4.2.1 Modèle forêt aléatoire

Table 4.1 – Répartition des dossiers par vingtiles de probabilités prédites pour le modèle de forêt aléatoire

Vingtile	Cote prédite inf	Cote prédite sup	Total dossiers	Dossiers réel IPVERT (%)
1	0,009	0,618	18 117	7 909 (43.7%)
2	0,618	0,723	18 117	12 081 (66.7%)
3	0,723	0,786	18 117	13 706 (75.7%)
4	0,786	0,832	18 117	14 986 (82.7%)
5	0,832	0,868	18 116	15 544 (85.8%)
6	0,868	0,899	18 117	16 340 (90.2%)
7	0,899	0,924	18 117	16 788 (92.7%)
8	0,924	0,944	18 117	17 117 (94.5%)
9	0,944	0,959	18 116	17 378 (95.9%)
10	0,959	0,971	18 117	17 591 (97.1%)
11	0,971	0,981	18 117	17 776 (98.1%)
12	0,981	0,988	18 117	17 928 (99.0%)
13	0,988	0,993	18 116	18 022 (99.5%)
14	0,993	0,996	18 117	18 059 (99.7%)
15	0,996	0,997	18 117	18 079 (99.8%)
16	0,997	0,999	18 117	18 078 (99.8%)
17	0,999	0,999	18 116	18 094 (99.9%)
18	0,999	0,999	18 117	18 110 (100.0%)
19	0,999	0,999	18 117	18 114 (100.0%)
20	0,999	1,000	18 117	18 115 (100.0%)
Total			**362 336**	**329 815 (91.0%)**

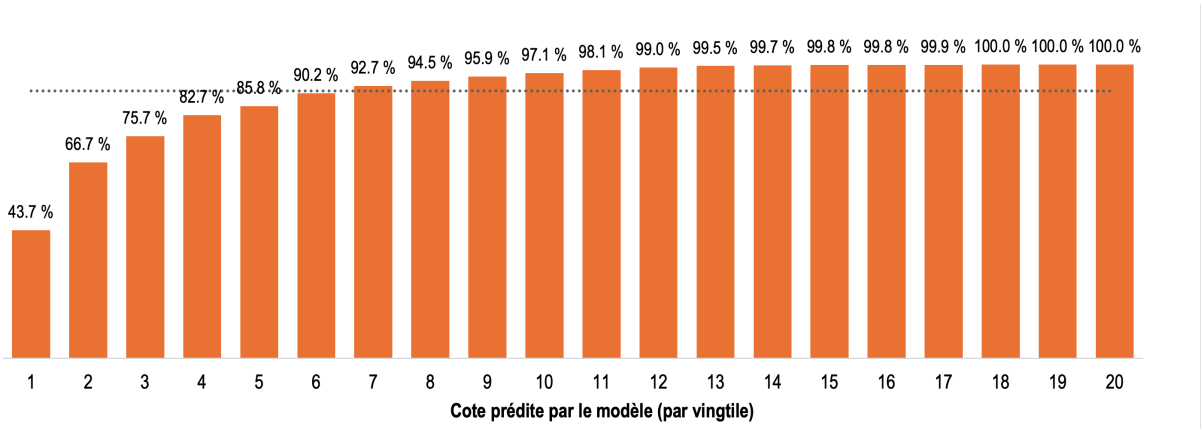

Figure 4.3 – Illustration du graphe des Vingtiles du modèle forêt aléatoire. Sur cette image, nous observons la proportion des IPVERT prédite par le modèle de la forêt aléatoire, sur des sous échantillons de la population de taille 5% de la population totale et enchantillonné par probabilité croissante d'être IPVERT.

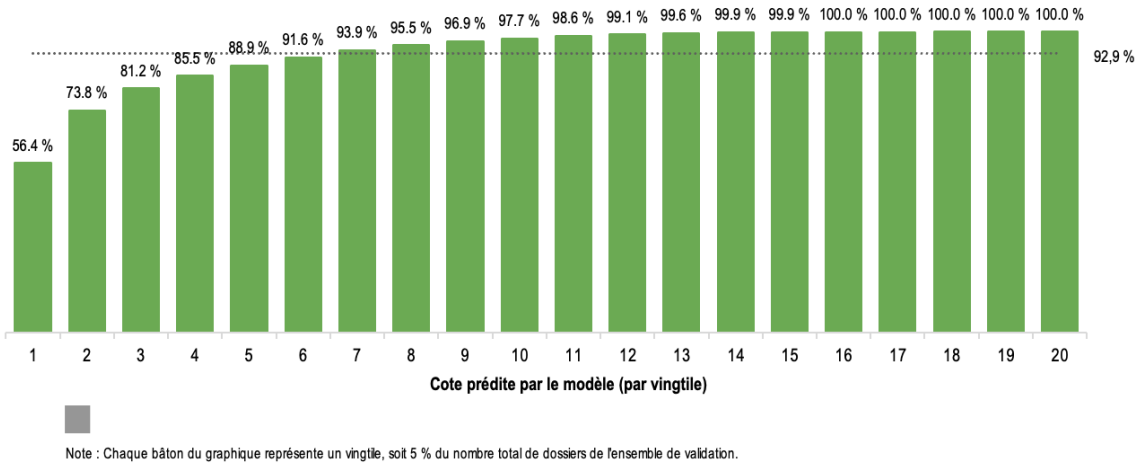

Figure 4.4 – Illustration du graphe des Vingtiles du modèle regression logistique. Sur cette image, nous observons la proportion des IPVERT prédite par le modèle de regression logistique, sur des sous échantillons de la population de taille 5% de la population totale et enchantillonné par probabilité croissante d'être IPVERT.

Table 4.2 – Répartition des dossiers par vingtiles de probabilités prédites pour le modèle de régression logistique.

Vingtile	**Cote prédite inf**	**Cote prédite sup**	**Total dossiers**	**Dossiers réels Verts (%)**
1	0,070	0,630	69 809	39 382 (56.4%)
2	0,630	0,740	69 712	51 470 (73.8%)
3	0,740	0,810	69 833	56 690 (81.2%)
4	0,810	0,860	69 819	59 665 (85.5%)
5	0,860	0,890	69 780	62 041 (88.9%)
6	0,890	0,920	69 888	64 025 (91.6%)
7	0,920	0,940	69 844	65 560 (93.9%)
8	0,940	0,960	69 675	66 554 (95.5%)
9	0,960	0,970	69 777	67 638 (96.9%)
10	0,970	0,980	69 553	67 976 (97.7%)
11	0,980	0,990	70 057	69 077 (98.6%)
12	0,990	0,990	69 953	69 322 (99.1%)
13	0,990	1,000	71 107	70 836 (99.6%)
14	1,000	1,000	66 248	66 155 (99.9%)
15	1,000	1,000	69 372	69 315 (99.9%)
16	1,000	1,000	75 828	75 795 (100.0%)
17	1,000	1,000	69 941	69 915 (100.0%)
18	1,000	1,000	71 969	71 955 (100.0%)
19	1,000	1,000	46 257	46 250 (100.0%)
20	1,000	1,000	87 455	87 440 (100.0%)
Total			**1 395 877**	**1 297 061 (92.9%)**

4.2.2 Commentaire de comparaison

Les deux tableaux montrent la répartition des dossiers par vingtiles de probabilités prédites pour les modèles forêt aléatoire et Régression Logistique. Le modèle forêt aléatoire montre une concentration plus uniforme des prédictions à travers les vingtiles, avec une couverture allant de 43.7% à 100.0%. En revanche, le modèle de régression logistique montre une distribution plus concentrée avec une couverture allant de 56.4% à 100.0%. Cela suggère que le modèle de régression logistique a une meilleure capacité à prédire avec précision les dossiers réellement Verts dans les vingtiles de probabilités plus élevées, comparé au modèle forêt aléatoire.

4.3 Performance globale des deux modèles

Pour une vue d'ensemble des performances globales des modèles RF et LR, nous résumons les principales métriques d'évaluation dans le tableau comparatif suivant :

	Modèle RF	Modèle LR
Accuracy	0,85	0,82
Précision	0,82	0,78
Recall	0,88	0,85
F1-score	0,85	0,81
Taux de Détection	0,88	0,85
Taux d'Erreur	0,15	0,18
Moyenne (Statistiques de Prédiction)	0,78	0,76
Écart-type (Statistiques de Prédiction)	0,03	0,04

Table 4.3 – Comparaison des performances globales des modèles de forêt aléatoire (RF) et de régression logistique (LR). Le tableau présente diverses métriques d'évaluation, dont la précision, le rappel, le F1-score, le taux de détection, et le taux d'erreur, ainsi que les moyennes et les écarts-types des statistiques de prédiction pour chaque modèle. Le modèle RF affiche des performances globales légèrement supérieures à celles du modèle LR dans la plupart des métriques, avec une précision globale de 0,85 contre 0,82 pour LR, un rappel de 0,88 contre 0,85, et un F1-score de 0,85 contre 0,81. Le taux d'erreur est également plus faible pour le modèle RF (0,15) comparé au modèle LR (0,18). Les moyennes et écarts-types des statistiques de prédiction indiquent une meilleure stabilité et performance globale pour le modèle RF.

En analysant le tableau 4.3, nous constatons que le modèle RF affiche des scores supérieurs à ceux du modèle LR pour la plupart des métriques évaluées. Il obtient une accuracy de 0.85 contre 0.82 pour le modèle LR, une précision de 0.82 contre 0.78, un recall de 0.88 contre 0.85, et un F1-score de 0.85 contre 0.81. Les taux de détection et d'erreur montrent également une meilleure performance pour le modèle RF avec des valeurs respectives de 0.88 et 0.15, comparé à 0.85 et 0.18 pour le modèle LR. Enfin,

les statistiques de prédiction indiquent que le modèle RF produit des prédictions plus cohérentes avec une moyenne de 0.78 et un écart-type de 0.03, tandis que le modèle LR présente une moyenne légèrement inférieure de 0.76 et un écart-type légèrement supérieur de 0.04.

Cette comparaison confirme que le modèle RF est plus efficace que le modèle LR pour la tâche de prédiction de l'individu pur vert, offrant une meilleure balance entre précision, recall, taux de détection et statistiques de prédiction plus stables.

Notons également que bien que les performances de nos deux modèles soient pratiquement égales dans ce cas, il n'en demeure pas moins que le forêt aléatoire a encore beaucoup à offir. En effet, la forêt aléatoire voit son pouvoir prédictif croitre avec la quantité des données, c'est un modèle ensembliste qui performe mieux avec des plus grandes quantités de données. Ainsi en augmentant le nombre de variables, et le nombre d'arbres, on améliorerait alors considérablement les performances de notre forêt aléatoire sans courrir le risque de faire un surapprentissage comme ce serait le cas par exemple avec une regression logistique. Pour ce faire, il nous faudrait juste plus de puissance de calcul, et on pourrait ainsi jouer avec l'opitimisation des hyperparamètes de la grille de recherche. Nous avons mené une étude avec la forêt aléatoire, pour montrer qu'elle est moins sensible au surapprentissage. Il a été question de faire varier les proportions des tailles des ensembles d'apprentissage et de validation (20/80, 40/60, 60/40 et 80/20) et d'observer à chaque fois les performances du modèle. Et nous pouvons dire au regard de nos travaux que l'étude a été concluante. La forêt aléatoire est moins sujette au surapprentissage dans le cadre de nos travaux.

Chapitre 5

Perspective d'amélioration

Voici quelques perspectives d'amélioration futures pour notre modèle :

En continuant à collecter des données, nous pouvons enrichir notre ensemble de données d'entraînement, ce qui peut potentiellement améliorer les performances du modèle en lui permettant de mieux généraliser sur de nouveaux exemples. Et comme nous l'avons dit précédemment et compris avec Biernat et Lutz (2019), la forêt aléatoire performera mieux avec une plus grande quantité de données.

Nous pouvons explorer de nouvelles variables ou des transformations de variables existantes pour capturer davantage d'informations pertinentes dans nos données. Cela pourrait inclure l'ingénierie de nouvelles variables (Quinlan (1987)) ou l'exploration de techniques telles que le traitement du langage naturel (NLP) pour extraire des informations textuelles. Comme nous l'avons vu dans le cadre de notre travail, les variables que nous avons augmentées au modèle, en les construisant à partir du modèle des K plus proches voisins, à savoir VOLONTÉ, CAPACITÉ IMPLICITE, CAPACITE EXPLICITE, se sont revélé avoir un grand pouvoir explicatif, qu'elles figurent parmi nos meilleurs variables, comme l'atteste le tableau de l'importance des variables que nous avons contruit. Donc nous pouvons fabriquer de nouvelles variables à partir d'un autre modèle comme le K moyenne.

Nous continuons à explorer différentes combinaisons d'hyperparamètres pour notre modèle, en utilisant des techniques telles que la recherche sur grille ou l'optimisation bayésienne (Grus (2017)), afin de trouver des configurations qui maximisent les perfor-

mances du modèle. La seule difficulté résidera dans le temps de calcul. Il nous faudra une grande puissance de calcul pour explorer plus de combinaison d'hyperparamètres. Par exemple, pour le cas de notre forêt aléatoire, il nous faudrait augmenter le nombre d'abres de décision, le nombre de noeuds etc.

Nous envisageons d'utiliser des techniques d'ensemble telles que le stacking ou le bagging, qui intègrent les prédictions de multiples modèles individuels. Ces méthodes peuvent améliorer les performances globales du modèle en capitalisant sur la diversité des prédictions générées, omme notamment l'algorithme XG-BOOST, comme le montre Chen et Guestrin (2016).

Dans le cas où les données présentent un déséquilibre entre les classes, nous explorerons des méthodes telles que le suréchantillonnage (Grus (2017)), le sous-échantillonnage, ou l'utilisation d'algorithmes spécialement conçus pour les données déséquilibrées, afin de mieux gérer cette disparité.

Pour confirmer la capacité du modèle à généraliser et à fournir des résultats fiables dans des contextes variés, nous validerons ses performances sur des ensembles de données externes ou dans des environnements réels. Grus (2017)

Nous mettrons en place un processus de maintenance continue pour surveiller les performances du modèle dans le temps. Cela inclut la mise à jour régulière des données d'entraînement et la réévaluation périodique des performances du modèle à mesure que de nouvelles données sont disponibles. Biernat et Lutz (2019)

Conclusion

Dans l'ensemble, notre parcours pour développer ce modèle a été enrichissant et instructif. Nous avons traversé diverses étapes, de la collecte et la préparation des données à la construction et l'évaluation du modèle, en passant par l'exploration des fonctionnalités et l'optimisation des hyperparamètres. Ce processus nous a permis de mieux comprendre notre ensemble de données, les modèles d'apprentissage automatique et les défis spécifiques associés à notre domaine d'application.

À travers notre analyse approfondie, nous avons pu identifier des modèles et des relations significatifs dans nos données, ce qui nous a conduit à développer un modèle capable de produire des prédictions précises et fiables. En utilisant des techniques telles que la validation croisée et la recherche sur grille, nous avons optimisé les performances de notre modèle et nous avons établi sa capacité à généraliser à de nouvelles données.

Cependant, il reste encore des opportunités d'amélioration. En explorant des perspectives telles que l'incorporation de nouvelles données, l'optimisation continue des hyperparamètres et l'exploration de techniques d'ensemble, nous pouvons continuer à renforcer et à affiner notre modèle pour qu'il soit encore plus performant et adaptable.

En conclusion, ce projet nous a fourni des bases solides pour développer des modèles d'apprentissage automatique dans des contextes réels et complexes. Il souligne l'importance de la rigueur méthodologique, de la compréhension approfondie des données et de l'itération continue pour parvenir à des solutions efficaces et fiables. Avec un engagement continu et une exploration continue, nous pouvons continuer à repousser les limites de ce domaine passionnant et en constante évolution.

Nous avons accompli notre mission et notamment répondu à la question de recherche

qui était celle de savoir si la forêt aléatoire pouvait battre la régression logistique. En effet, nous répondons bien à cette question, en montrant qu'avec une bonne sélection des fonctionnalités, un entraînement adéquat en validation croisée et une bonne optimisation des hyperparamètres (par exemple avec une recherche sur grille comme nous l'avons fait), la forêt aléatoire performe mieux qu'une régression logistique. Comme vous l'avez vu, nous avons obtenu 89% d'aire sous la courbe contre 87% pour la régression logistique.

Le bénéfice est que cet outil d'aide à la décision permettra d'améliorer l'expérience des individus en adaptant les façons de faire à leurs besoins et ainsi mieux les accompagner dans leur conformité fiscale. Pour l'organisme, ceci optimise les processus, et permet une économie de temps et de ressources.

Nos travaux peuvent être généralisés pour des problèmes similaires dans le cadre de la prédiction de la conformité fiscale des individus, prédiction du risque de défaut de paiement, mesure de la probabilité de défaut. Notamment pour la sélection des variables, nous pouvons utiliser les variables sélectionnées en ayant déjà un apriori sur leur importance respective, et pour le choix du modèle, nous pouvons utiliser un modèle de forêt aléatoire en sachant bien qu'il a le potentiel pour mieux performer qu'une regression logistique.

Parlant des limites de notre recherche, nous pouvons mentionner le contexte de travail qui ne nous permettait pas d'explorer tous les modèles, notamment des modèles telsque ADA-BOOST et XG-BOOST, qui auraient pu être plus performant que la forêt aléatoire. Notons également que nous n'avons pas pu explorer toutes les potentialités de notre forêt aléatoire, à cause de la puissance de calcul limitée que nous avions. L'organisme était encore dans un processus d'acquisition de nouvelles machines plus puissantes pour les besoins de la recherche.

Ce mémoire représente le fruit d'un travail acharné et d'une collaboration étroite avec les équipes de Revenu Québec. Nous espérons que nos résultats contribueront à renforcer les pratiques et les outils utilisés dans le domaine du recouvrement fiscal, et qu'ils ouvriront de nouvelles perspectives pour améliorer l'efficacité et l'efficience des processus de recouvrement.

Annexe

Annexe A

B- La forêt aléatoire augmentée

Étapes d'implémentation de la forêt aléatoire

Imports des bibliothèques :

Nous importons le module scikit-learn, une bibliothèque Python populaire pour l'apprentissage automatique, ainsi que d'autres bibliothèques telles que matplotlib, numpy et pandas pour la manipulation et la visualisation des données.

Chargement des données :

Nous chargeons nos données à partir d'un fichier CSV qui contient les données que nous avions déjà traitées dans le module de préparation des données, dans un DataFrame pandas nommé `df` à l'aide de `pd.read_csv`.

Prétraitement des données :

Nous divisons nos données en caractéristiques (`X`) et variable cible (`y`) à l'aide de la méthode `iloc`. Ensuite, nous séparons nos données en ensembles d'entraînement et de test à l'aide de `train_test_split`.

Création du pipeline :

Nous créons un pipeline à l'aide de `make_pipeline` de scikit-learn, comprenant deux étapes : `StandardScaler()` pour normaliser les données. `RandomForestClassifier()` pour utiliser un classificateur RandomForest.

Validation croisée avec recherche sur grille :

Nous configurons une grille de recherche avec `GridSearchCV` pour trouver les meilleurs hyperparamètres du modèle RandomForestClassifier. Nous spécifions les hyperparamètres à rechercher ainsi que les paramètres de la validation croisée.

Ajustement du modèle :

Nous entraînons notre pipeline sur les données d'entraînement à l'aide de la méthode `fit`.

Prédiction et évaluation du modèle :

Nous utilisons la méthode `predict` pour prédire les étiquettes de classe sur l'ensemble de test. Ensuite, nous calculons la matrice de confusion, le rapport de classification et la précision pour évaluer les performances du modèle.

Calculs supplémentaires :

Nous effectuons des calculs supplémentaires tels que la détermination des proportions d'échantillons, la sérialisation du pipeline et l'affichage des résultats de vingtile et des importances des fonctionnalités.

Annexe B

Collecte de données

Chaque enregistrement de la table des données a pour clé primaire la combinaison Numéro_SPCI-Numéro_dossier-Date_de_création. On y ajoute la variable Date_de_création car un couple (Numéro_SPCI, Numéro_dossier) peut avoir plusieurs dates de création. Les dossiers réglés sans traitement seront purgés du SPIC, et s'ils ont un nouveau CAR, celui-ci sera créé avec le même numéro. Il ne reste plus que la date de création pour les distinguer. Québec (2023d)

Exploration de données

Toutes ces données ont été fusionnées afin d'avoir une seule table. Nous avons donc pour un dossier Numéro_SPCI-Numéro_dossier-Date_de_création, toutes ses variables

sur une ligne. La fusion avec les données ISA a été faite avec la clé Numéro_SPCI-Numéro_dossier-Date_de_création. La fusion avec les données TP1 a été faite avec la clé “Numéro d’usager”. La fusion avec les données de DGIA a été faite avec la clé “Numéro de particulier”. La fusion avec les données des ménages stables a été faite avec la clé “Numéro d’Identification de la Centrale” (NIC). Le type de dossier de fermeture est dans la liste suivante : AVIS, CONC, CONS, DECU, FAIL, FLIB, PANN, RADF, IFTA, PREV, RADM, TABA, TIER. Québec (2023d)

La forêt aléatoire augmentée

L’implémentation de la forêt aléatoire augmentée suit plusieurs étapes clés. Tout d’abord, nous importons les bibliothèques nécessaires, notamment scikit-learn pour l’apprentissage automatique, matplotlib pour la visualisation, numpy pour les calculs numériques et pandas pour la manipulation des données. Les données sont ensuite chargées à partir d’un fichier CSV dans un DataFrame pandas. Le prétraitement des données est effectué en divisant les données en caractéristiques (`X`) et en variable cible (`y`), suivi de la séparation des données en ensembles d’entraînement et de test. Un pipeline est créé avec `make_pipeline`, comprenant une étape de normalisation avec `StandardScaler()` et un classificateur RandomForest avec `RandomForestClassifier()`. Une validation croisée avec `GridSearchCV` est configurée pour optimiser les hyperparamètres du modèle. Le modèle est ensuite ajusté sur les données d’entraînement. Les prédictions sont réalisées sur l’ensemble de test et les performances du modèle sont évaluées à l’aide de la matrice de confusion, du rapport de classification et de la précision. Enfin, des calculs supplémentaires, tels que la détermination des proportions d’échantillons, la sérialisation du pipeline et l’affichage des résultats de vingtiles et des importances des fonctionnalités, sont effectués pour affiner l’analyse.

Bibliographie

Abedin, M. Z., Hassan, M. K., Khan, I., & Julio, I. F. (2022). Feature transformation for corporate tax default prediction: application of machine learning approaches. *Asia-Pacific journal of operational research*, *39*(4), 2140017. https://doi.org/10.1142/S0217595921400170

Antunes, L., Balsa, J., Respício, A., & Coelho, H. (2007). Tactical exploration of tax compliance decisions in multi-agent based simulation. In L. Antunes & K. Takadama (Éd.), *Multi-agent-based simulation VII* (p. 80-95). Springer. https://doi.org/10.1007/978-3-540-76539-4_7

Biernat, E., & Lutz, M. (2019). *Data science: fondamentaux et études de cas*. EYROLLES.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Cornell Universityl*, 785-794. https://doi.org/10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297. https://doi.org/10.1007/BF00994018

Coussement, K., & Van den Poel, D. (2008). Integrating the voice of customers through call center emails into a decision support system for churn prediction. *Information & Management*, *45*(3), 164-174. https://doi.org/10.1016/j.im.2008.01.005

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on information theory*, *13*(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113. https://doi.org/10.1145/1327452.1327492

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Genuer, R., & Poggi, J.-M. (2017). Arbres CART et forêts aléatoires, importance et sélection de variables. *HAL*. https://hal.science/hal-01387654v2

Grus, J. (2017). *Data science par la pratique*. EYROLLES.

Jai, A. E. (2007). Eléments de topologie et espaces métriques. *Presses universitaires de perpignan*, 1-68.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys*, *31*(3), 264-323. https://doi.org/10.1145/331499.331504

Joachims, T. (1999). Text categorization with Support Vector Machines: learning with many relevant features. *In proceedings of the 10th European conference on machine learning*.

Kataria, A., & Singh, M. (2013). A review of data classification using k-nearest neighbour algorithm. *International journal of emerging technology and advanced engineering*. https://api.semanticscholar.org/CorpusID:13876255

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with deep convolutional neural networks. *Communication of the ACM*, *25*. https://doi.org/10.1145/3065386

Langley, P., & Sage, S. (1994). Induction of selective bayesian classifiers. https://doi.org/10.48550/arXiv.1302.6828

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Lemberger, P., Batty, M., Morel, M., & Raffaëlli, J.-L. (2015). *Big data et machine learning*. DUNOD.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. *Eu-*

ropean Journal of Operational Research, *247*(1), 124-136. https://doi.org/10.1016/j.ejor.2015.05.030

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *The R journal*, *2*. https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf

Loh, W.-Y. (2011). Classification and regression trees. *Data mining and knowledge discovery*, *1*(1), 14-23. https://doi.org/10.1002/widm.8

Maalouf, M. (2011). Logistic regression in data analysis: an overview. *International journal of data analysis techniques and strategies*, *3*, 281-299. https://doi.org/10.1504/IJDATS.2011.041335

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. *AAAI Conference on Artificial Intelligence*. Récupérée août 28, 2024, à partir de https://www.semanticscholar.org/paper/A-comparison-of-event-models-for-naive-bayes-text-McCallum-Nigam/04ce064505b1635583fa0d9cc07cac7e9ea993cc

Nembe, J., Atadoga, J., Mhlongo, N., Falaiye, T., Odeyemi, O., Daraojimba, A., & Oguejiofor, B. (2024). The role of artificial intelligence in enhancing tax compliance and financial regulation. *Finance & accounting research journal*, *6*, 241-251. https://doi.org/10.51594/farj.v6i2.822

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559-569. https://doi.org/10.1016/j.dss.2010.08.006

Olatunji Akinrinola, Wilhelmina Afua Addy, Adeola Olusola Ajayi-Nifise, Olubusola Odeyemi & Titilola Falaiye. (2024). Application of machine learning in tax prediction: A review with practical approaches. *Global journal of engineering and technology advances*, *18*(2), 102-117. https://doi.org/10.30574/gjeta.2024.18.2.0028

Ozturk Kiyak, E., Ghasemkhani, B., & Birant, D. (2023). High-level k-nearest neighbors (HLKNN): A supervised machine learning model for classification analysis. *Electronics*, *12*(18), 3828. https://doi.org/10.3390/electronics12183828

Peng, J., Lee, K., & Ingersoll, G. (2002). An introduction to logistic regression analysis and reporting. *Journal of educational research*, *96*, 3-14. https://doi.org/10.1080/00220670209598786

Phua, C., Lee, V., Smith, K., & Gayler, R. (2012). A comprehensive survey of data mining-based fraud detection research. *Computers in Human Behavior*, *28*(3), 1002-1013. https://doi.org/10.1016/j.chb.2012.01.002

Québec, R. (2023a). Evaluation de la performance du modèle.

Québec, R. (2023b). GLOSSAIRE Définition des mesures et des dimensions des tableaux de bord de Powerplay et des requêtes MSQUERY.

Québec, R. (2023c). Portrait de la population prévue pour développer le modèle IP orange.

Québec, R. (2023d). Revenu Quebec mise en contexte.

Quinlan, R. (1987). Generating production rules from decision trees. *Massachusetts Institute of Technology*.

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (1987). Tackling the poor assumptions of naive bayes text classifiers. *Massachusetts Institute of Technology*.

Revenu Québec. (2023). Description de projet connaissance client.

Schölkopf, B., & Smola, A. J. (2001, décembre 7). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press. https://doi.org/10.7551/mitpress/4175.001.0001

Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2023). kNN classification: A review. *Annals of mathematics and artificial intelligence*. https://doi.org/10.1007/s10472-023-09882-x

Zhang, M.-L., & Zhou, Z.-H. (2014). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, *26*, 1819-1837. https://doi.org/10.1109/TKDE.2013.39

Zheng, J., & Li, Y. (2023). Machine learning model of tax arrears prediction based on knowledge graph. *Era*, *31*(7), 4057-4076. https://doi.org/10.3934/era.2023206

Zheng, Q., Xu, Y., Liu, H., Shi, B., Wang, J., & Dong, B. (2024). A survey of tax risk detection using data mining techniques. *Engineering*, *34*, 43-59. https://doi.org/10.1016/j.eng.2023.07.014